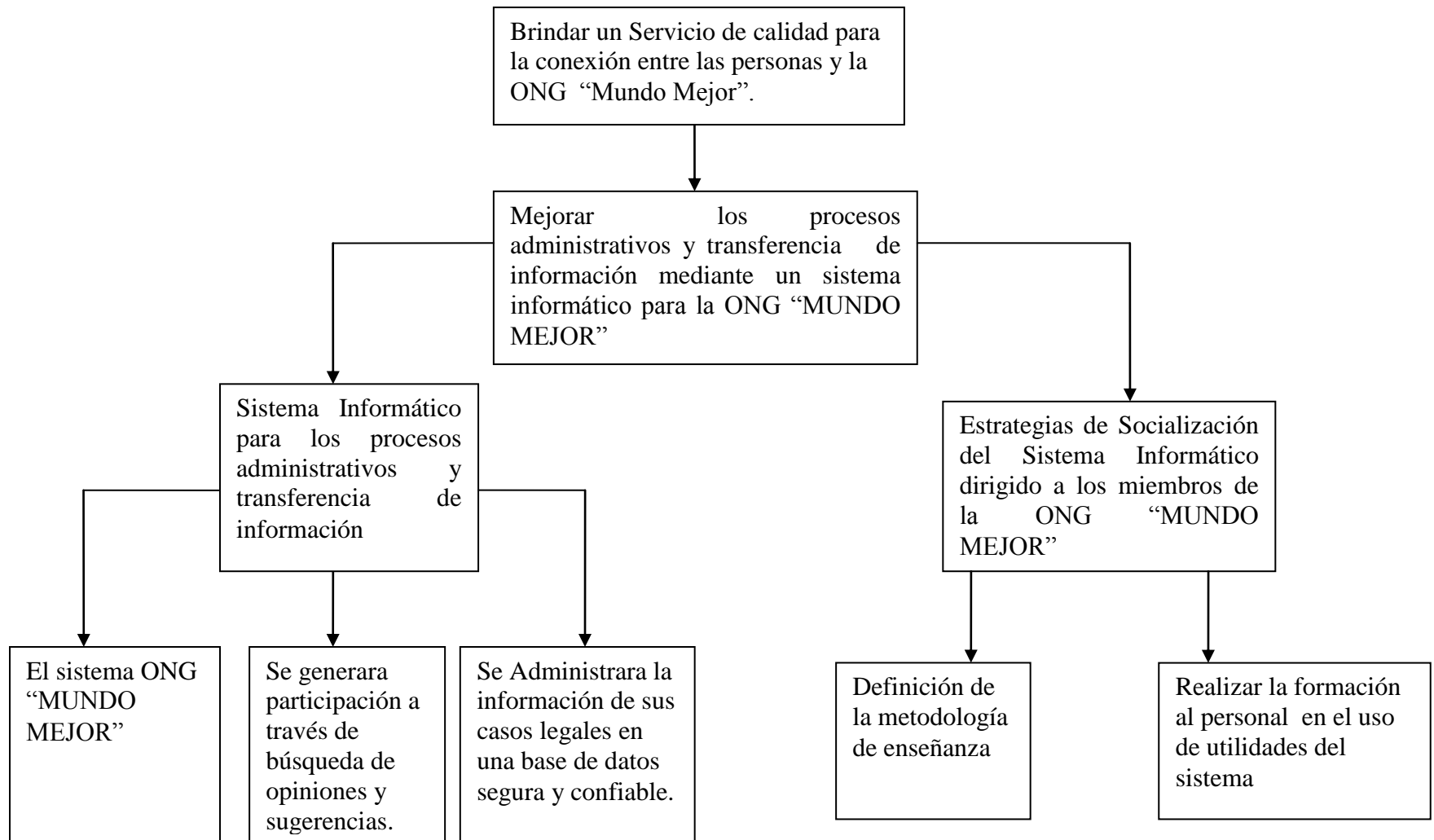


I.2.2.2 Análisis de Objetivos

Árbol de Objetivos



Por:

EDSON MARCELO PLAZA BARRIOS

Tarija – Bolivia

IDENTIFICACION DEL PROYECTO	
Titulo del proyecto	MEJORAMIENTO DE LA ADMINISTRACION DE CASOS LEGALES EN LA ONG “MUNDO MEJOR”
Carrera	ING. INFORMÁTICA
Facultad	CIENCIAS Y TECNOLOGÍA
Institución	ONG “MUNDO MEJOR”
Provincia	TARIJA- CERCADO
Duración del proyecto	8 MESES
Área	DERECHO DE FAMILIA

I.1. Personal Vinculado al Proyecto

I.1.1. Director de Proyecto

--	--	--	--

Plaza Apellido Paterno	Barrios Apellido Materno	Edson Marcelo Nombres	5049801 C.I.
Universitario Profesión	Informática Carrera ó Unidad	Ciencias y Tecnología Facultad:	
46641819 Telef. Oficina	77870644 Celular	jugadordeunbit@gmail.com Correo electrónico	Firma

I.1.2. Participantes equipo de Trabajo

Categoría	Nombres y Apellidos	Profesión	C.I.	Firma
Investigador	Edson Marcelo Plaza Barrios		5049801	

I.1.3. Equipo de Trabajo de: Institución cooperante

Nombre: ONG “Mundo Mejor”			
Dirección: C/ Colon N° 240		Telef. Oficina: 46668713	
Nombre y Apellidos	Cargo	C.I.	Firma
Carmen Rosa Cari Figueroa	Administrador	5011070	
Carolina Solares	Asesor Legal	5009450	
Henry Mamani Patzi	Asesor Legal	4014592	

I.1.4. Actividades previstas para los integrantes del equipo de investigación

Responsable	Actividades
Director	Globales del proyecto: <ul style="list-style-type: none"> Planificar y controlar del cronograma del proyecto.

	<ul style="list-style-type: none">• Asignar y gestionar recursos y prioridades a los distintos componentes y Actividades del proyecto.• Establecer un conjunto de prácticas que aseguren la calidad e integridad del proyecto.• Supervisar el desarrollo del proyecto. <p>Para los componentes:</p> <ul style="list-style-type: none">• Elaborar el análisis y diseño del sistema.• Elaborar la base de datos del sistema• Realizar la programación del sistema informático.
--	--

I.2. Descripción del Proyecto

I.2.1. Resumen Ejecutivo del Proyecto

El presenta proyecto tiene como finalidad implementar un Sistema Automatizado vía Web acorde a los requerimientos de la ONG “Mundo Mejor” que se encuentra en el departamento de Tarija- Cercado.

El Area que se tomara en cuenta para la realización del sistema es Derecho de Familia es el conjunto de normas e instituciones jurídicas que regulan las relaciones personales y patrimoniales de los miembros que integran la familia, entre sí y respecto de terceros. Tales relaciones se originan a partir del matrimonio y del parentesco.

Caracteres del Derecho de Familia

Sus normas son de carácter público Los fallos que resuelven conflictos del grupo familiar en la mayoría de los casos no alcanzan la calidad de cosa juzgada, son sentencias formales en su mayoría, no son sentencias materiales, es decir, las sentencias familiares son revisables ulteriormente.

La autonomía de la voluntad es restringida. No pueden extinguir o modificar una relación interpersonal del grupo familiar sin intervención del juez.

Sus normas son para regular la comunidad familiar.

Actualmente la información que existe se la maneja de forma manual, se busca implementar ayuda a las personas con algún problema ya sea en asistencia, orientación, patrocinio legal y capacitación, además pudiendo tener su espacio para opinar y brindar información de cualquier aspecto que pueda mejorar las alternativas de trabajo que tiene la organización, en el departamento de Tarija – Cercado para ello se propone desarrollar entornos amigables para la gestión y personalización del sistema.

El sistema permitirá dar información tanto a los funcionarios de la ONG “MUNDO MEJOR” como al Visitante y Beneficiario, se publicara temas de la actualidad, información de los distintos proyectos que se lleva a cabo en los diferentes lugares de la ciudad de Tarija.

1.2.2. Descripción, Fundamentación y justificación del proyecto

El sistema permitirá automatizar el control de todas sus actividades (asesoría legal, Noticias Relevantes, Actividades) donde el usuario contara con un acceso rápido y

sencillo a los datos, gracias a interfaces graficas sencillas y amigables para el manejo del sistema.

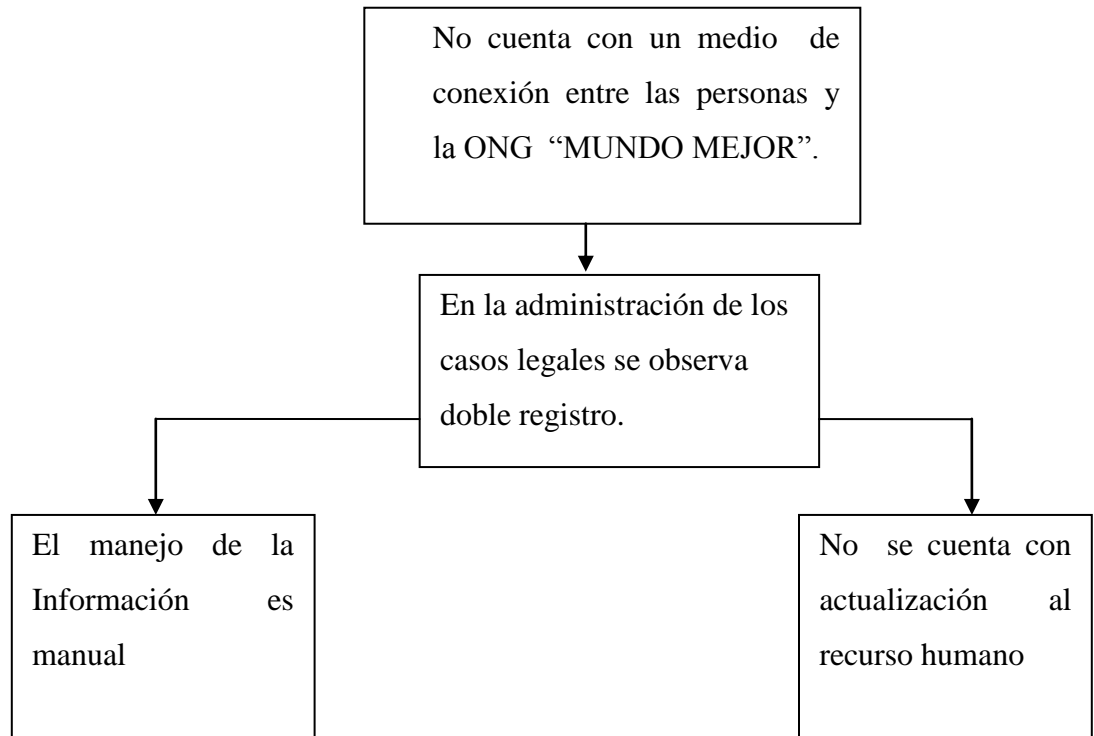
El administrador podrá imprimir reportes de los últimos casos legales, actualizar los datos desde computadora conectada al Internet, esta es una ventaja moderna y sofisticada para la administración de una organización.

Los Abogados podrán conocer más rápido la información que se le brinda el Visitante para un próximo caso legal.

Los Visitantes podrán verter opiniones de los proyectos que se llevan a cabo conocer las actividades ya realizadas, Noticias relevantes y entre lo mas importante brindar la información de algún tipo de problema de incidencia legal para la solución de su problema.

I.2.2.1 Análisis de causas de problemas

Árbol de Problemas



I.2.2.2 Análisis de Objetivos

Árbol de Objetivos

I.2.2.3 Situación planteada con el proyecto

Situación sin Proyecto	Situación con el Proyecto
Se cuenta con metodologías de comunicación a las personas de manera antigua pero no obsoleta y una documentación manual de actividades.	Población en general encontrara nuevas formas de interactuar y el manejo de sus actividades tendrán un respaldo de seguridad.

I.2.3. Objetivos

I.2.3.1. Objetivo General

Optimizar los medios para gestionar ayuda a las personas.

I.2.3.2. Objetivo Específico

- El sistema informático abarcara a toda la población sin distinción alguna
- Se generara participación a través de búsqueda de opiniones y sugerencias
- Se Administrara la información de sus actividades legales en una base de datos segura y confiable

I.2.4. Marco Lógico Del Proyecto

I.2.5. Metodología de Trabajo

Este Trabajo pretende contar con una herramienta de software que permita ayudar a buscar personas que necesiten ayuda ya sea educativos, informativos o de vía legal. Las características principales son:

- Uso de software.
- Generación del Sistema ONG “MUNDO MEJOR”.

- Incorporación de Noticias, Actividades, capacitaciones en diferentes áreas y dirigidas a sectores en formación.
- Navegación intuitiva.

Desarrollar el sistema ONG “MUNDO MEJOR” que involucre aspectos del desarrollo humano y búsqueda de personas que necesiten ayuda mediante la vía legal, capacitación y orientación.

El objetivo principal de este componente es contar con un espacio tecnológico basado en la Web, referido a la integración en línea buscando fortalecer el desarrollo humano, la opinión, ayuda y la seguridad de la información.

- Ámbito de la Información
- Actualidad
- Información Institucional
- Derechos Humanos
- Información sobre trámites en el área legal
- Herramientas de búsqueda
- Área que se trabaja.
- Incidencias Políticas
- Proyectos de Leyes
- Ámbito de participación
- Mensajería electrónica
- Ámbito tecnológico
- Uso del software.

a) Descripción y Relación con las estrategias con los objetivos

Creación del sistema vía Web para la	Mayor participación de las personas,

personas en general.	búsqueda incesante de personas que necesiten ayuda en la vía legal.
Información de noticias, actividades, quienes son como ONG “MUNDO MEJOR”.	Orientación a disposición para brindar ayuda a cualquier persona que así lo necesite.

I.2.6. Resultados Esperados

Sistema ONG “MUNDO MEJOR” que abarque a toda la población sin distinción alguna, Optimizar los Procesos para gestionar ayuda a las personas mediante Actividades, Noticias, opiniones respecto a lo que brinda de información.

Se generara participación a través de búsqueda de personas en ayuda.

Se Administrara la información de los casos legales en una base de datos segura y confiable.

I.2.7. Transferencia de Resultados

a) Medios y estrategias para la transparencia de resultados

En el presente proyecto se establecen 2 etapas que se realizarán de manera secuencial y mediante un convenio con la ONG “Mundo Mejor” las mismas que se detallan a continuación:

Diseñar el sistema que abarque a toda la población sin distinción alguna en las diferentes areas de trabajo que se brinda ayuda.

b) Grupo beneficiarios de los resultados

Los grupos que resultarán beneficiados con el presente proyecto son:

- Los Visitantes que requieran ayuda al conocernos mediante un sistema.
- Los Beneficiarios que forman parte de la ONG “MUNDO MEJOR”.
- El personal de Abogados y plantel administrativo de la ONG “MUNDO MEJOR”.

c) Resultados esperados

Este proyecto otorgara un desempeño importante en la ONG “MUNDO MEJOR”, presentando mayor facilidad en el manejo de los procesos legal, realizando transferencia

de información para minimizar el tiempo de esta manera adaptarse a una herramienta que les permite la automatización de los datos.

Se realizara Estrategias de socialización del Sistema Informático a los integrantes directos de la ONG “MUNDO MEJOR”.

I.2.8. Bibliografía

- CURSOSML www.sml.com
- MARCO LOGICO www.sistema-marco-logico-sml_files/navbar.htm
- CARE INTERNACIONAL www.careinternacional.org.ok
- DEFINICIONES www.definiciones.com
- MENU DE UTILIDADES <http://www.puravida.org/>
- INFORMACION www.procosi.org
- INFORMACION <http://www.luisrazeto.net>

I.3. Presupuesto / Justificación

ITEM	RUBROS	APORTE UNIVERSIDAD	OTRO APOORTE	TOTAL
20000	SERVICIOS PERSONALES NO		2800	2800
	21000. Servicios Básicos			

	22000. Servicios de transporte		2000	2000
	23000. Alquileres			
	24000. Mantenimiento y reparación		1500	1500
	25000. Servicios Profesionales y Comerciales		7,800	7,800
	Sub Total Rubro		14100	14100
30000	MATERIALES Y SUMINISTROS			
	31000. Alimentos y Productos		4000	4000
	32000. Productos de Papel, Cartón e Impresos		2000	2000
	33000. Textiles y Vestuario.		4000	4000
	34000. Productos Químicos, Combustibles y			

	Lubricantes			
	39000. Productos Varios.		500	500
	Sub total rubro		10,500	10,500
40000	ACTIVOS REALES			
	43000. Maquinaria y Equipo.		12,000	12,000
	46000. Descripción de estudios y proyectos para inversión			
	Sub Rubro Total		12,000	12000
	Total Presupuesto		36,600	36,600

1. Introducción

El presente documento es una Especificación de Requisitos de Software (ERS) para el Sistema Informático Vía WEB ONG “MUNDO MEJOR”. Todo su contenido ha sido elaborado en colaboración con los usuarios y responsables de la ONG “MUNDO MEJOR” que ayuda a personas de bajo recurso mediante asistencia legal y talleres de capacitación a varias personas.

- Área Legal.- comprende a los abogados que realizan seguimientos a todos los Casos Legales que están en proceso que son atendidos por su persona en la ONG “MUNDO MEJOR”
- Área Administrativa.- Comprende al administrador que tiene bajo su dependencia las Secciones de capacitaciones a distintos talleres y el manejo de la información de la ONG “MUNDO MEJOR”

El documento está dirigido al equipo de desarrolladores del Sistema, al equipo de Calidad, a los directivos del Ente Regulador y a los usuarios finales y tiene como objetivo principal documentar los requisitos funcionales y no funcionales del futuro Sistema.

Esta especificación está sujeta a revisiones por el grupo de usuarios, que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación por parte de los directivos del Ente Regulador, el grupo de Calidad y el grupo de usuarios. Una vez aprobado, servirá de base de desarrollo para la construcción del nuevo Sistema.

Esta especificación está estructurada según las directivas dadas por el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830-1998”.

1.1 Propósito

El propósito es definir de manera clara y precisa cuales son los requerimientos que debe tener un sistema en sus funcionalidades y restricciones para construirlo o desarrollarlo, que gestione los todos los procesos que genera la ONG “MUNDO MEJOR”.

El documento va dirigido a las personas encargadas del desarrollo e implementación del sistema, a la parte administrativa de la ONG “MUNDO MEJOR”, los Beneficiarios y todas las personas que visiten el sistema web. Este documento será el canal de comunicación entre las partes implicadas.

1.2 Alcance

El Sistema Informático Vía Web coadyuvara a la atención precisa, rápida, confiable y segura a todas las personas que requieran la ayuda en cualquier tipo de problema de tipo legal, orientación personal y Talleres que coadyuvan a la mejora de la vida cotidiana de cada persona

- Sistema Informático WEB.
- La parte de Administración de Usuarios: Se toma en cuenta con la creación de ingreso o modificaciones de los Usuarios y sus respectivos roles, donde solo el Administrador tiene permiso de realizar dichas acciones, esto incluye el nombre del usuario, apellidos, dirección, estado, email, login, clave y rol.
- La parte de Administración de Rol-Acceso: Se toma en cuenta las bajas y modificaciones de los Roles, donde solo el Administrador tiene permiso de realizar dichas acciones, esto incluye el nombre del rol, los módulos a los cuales tiene acceso y estado del mismo.
- La parte de Beneficiario Legal: Se toma en cuenta los ingresos del beneficiario, poder acceder a la ayuda legal, donde después algunos roles al cual se le de acceso tiene permiso de realizar dichas acciones, ver, guardar, subir, enviar.
- La parte de Visitante: El cual puede tener acceso a ver y opinar realizar consultas durante su sesión del sistema de algunas pantallas para después poder convertirse en beneficiario

El modo de acceso será a través de un navegador web, el Visitante tendrá ingreso al sistema sin validación, pero el administrador tanto como el y el Abogado y beneficiario ingresaran con previa validación de usuario y clave, pero a su vez también pueden ingresar como un visitante normal.

El número de usuarios que tendrán acceso para realizar modificaciones, dar bajas, incorporar nuevas aplicaciones y talleres de aprendizaje Administrativa del Sistema Informático Vía Web será reducido, entre 1 - 2 personas que trabajan en el área administrativa.

También los usuarios de la aplicación legal serán entre 4 – 6 personas.

Las personas visitantes del sistema no se tienen un número específico además que está abierto para las personas que necesitan la ayuda de la ONG “MUNDO MEJOR” el cual se tendrá un reporte.

1.3 Personal Involucrado

Nombre	Carmen Rosa Cari
Rol	Administrador de la ONG “MUNDO MEJOR”
Categoría profesional	Abogado
Responsabilidades	Dirigir, verificar y supervisar todas las actividades, talleres a la población en general.
Información de contacto	Abogado de Profesión
Aprobación	

Nombre	Jenrry Mamani
Rol	Encargado de Parte Legal
Categoría profesional	
Responsabilidades	Recibir los trámites de los beneficiarios ala

	parte legal.
Información de contacto	Abogado de profesión.
Aprobación	

1.4 Definiciones, acrónimos y abreviaturas

1.4.1 Definiciones

Abogado: Licenciado o doctor en derecho que ejerce profesionalmente la dirección y defensa de las partes en toda clase de procesos o el asesoramiento y consejo jurídico.

Beneficiarios: las personas que requieran ayuda en los problemas legales a un precio reducido o en algunas oportunidades gratis.

Administrador: Un Administrador de Sistema es aquella persona que se dedica a mantener y operar un sistema de cómputo.

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

Java (Java): Lenguaje de programación desarrollado por la empresa Sun para la elaboración de pequeñas aplicaciones exportables a la red (applets) y capaces de operar sobre cualquier plataforma a través, normalmente, de navegadores WWW. Permite dar dinamismo a las páginas web y aplicaciones de Escritorio.

Requerimientos funcionales: Se refiere a la funcionalidad o los servicios que se espera que el sistema provea.

Requerimientos no funcionales: Los requerimientos no funcionales tienen que ver con las características que de una u otra forma puedan limitar el sistema como son: el rendimiento (en tiempo y espacio), confiabilidad, interfaces, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

Racional Unified Process: Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Sistema Informático: Conjunto de partes (hardware y software) que funcionan relacionándose entre sí con un objetivo preciso. Los usuarios son parte del sistema informático.

Sistema Operativo: Un sistema operativo (SO) es un conjunto de programas o software destinado a permitir la comunicación del usuario con un ordenador y gestionar sus recursos de manera cómoda y eficiente. Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos. Ejemplos Windows, Linux, MacOS.

Usuario: Uno de las dos (o tres) personas autorizadas a usar las funcionalidades del sistema.

Administrador: Persona con acceso, no sólo a las funcionalidades, sino a las Interioridades de la BD.

1.4.2 Acrónimos

BBDD, BD: Bases de Datos, Base de Datos

Ficha de Antecedentes: se incorpora todos los detalles de un beneficiario de la ONG “MUNDO MEJOR”

Ficha caso legal: donde se ejecutara el caso legal.

1.4.3 Abreviaturas

ONG: Organizaciones no Gubernamental.

ERS: Especificación de Requerimientos Software

RUP: Racional Unified Process

TIC: Tecnologías de la Información y Comunicación

UML: Lenguaje Unificado de Modelado

1.5 Referencias

ANSI/IEEE Std. 830-1984 Guía del IEEE para la Especificación de Requerimientos Software.

1.6 Visión General del Documento

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo. En la sección 3 se definen con más detalle los requisitos que debe satisfacer el sistema.

2 Descripción General

En esta sección se presenta una descripción a alto nivel del sistema. Se presentarán las principales áreas a las cuales el sistema debe dar soporte, las funciones que el

sistema debe realizar, la información utilizada, las restricciones y otros factores que afecten al desarrollo del mismo.

2.1 Perspectiva del producto

El objetivo es que tanto los usuarios como el administrador puedan acceder al sistema desde un navegador web, facilidad de acceso a las funcionalidades del sistema. Lo ideal es que no dependa por lo general de terceras personas para hacer pequeños cambios a la BD.

El producto final se realizará utilizando la metodología RUP (Racional Unified Process), que mejora considerablemente la calidad de desarrollo del sistema, ya que la misma utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software.

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

El flujo de trabajo fundamental tiene los siguientes pasos:

- **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
 - Se localizan los problemas por lo que pasa la ONG “MUNDO MEJOR” según la información obtenida.
 - Se realizan cuestionarios
 - Se realizan entrevistas
 - Se realiza Diagrama de Casos de Uso del Negocio
- **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
 - Se realizan Diagramas de Casos de Uso del Sistema
 - Se realizan Diagramas de Clases
 - Se realizan Diagramas de Actividad
 - Se realizan Diagramas de Secuencia

- Se realiza Diagrama de Componentes
- Programación e Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
 - Se realiza la definición de la Arquitectura de Desarrollo
 - Se Realiza las Tecnologías a utilizar como:
 - Lenguaje de Programación
 - Tecnología Cliente / Servidor
 - Se realiza la Programación del Sistema
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
 - Se realiza la Verificación del Software
 - Pruebas de Caja Blanca
 - Pruebas de Caja Negra
 - Se realiza Manuales Instalación, Contingencia, Usuario.
 - Se realiza la Documentación Final del Sistema

La Aplicación Informática debe hacer conocer a las personas que existe una ONG “MUNDO” que ayuda a diferentes problemas de la sociedad con respecto a la Área Legal y las actividades que realiza como ONG mediante convenios en diferentes instituciones.

2.2. Funcionalidades del producto

En términos generales, el sistema deberá proporcionar soporte a las siguientes tareas:

- Consulta y actualización de información
- Seguimiento de la información durante un proceso
- Aplicación *online*, accesible mediante web
- Simplicidad y facilidad de manejo

- Base de Datos fácilmente extensible, de cara a la posible fusión con los datos

Modulo Administrador

- Incorporar a un nuevo Abogado
- Realizar actividades foros
- Realizar contenido de la ONG
- Informar del Área de trabajo
- Incorporar nuevas noticias
- Actualizar la información
- Actualizar el diccionario
- Confirmar los datos del beneficiario y actualizarlos
- Revisar todos los comentarios.

Modulo Visitante

- Ver toda la información del Abogado
- Ver todas las actividades de la ONG “MUNDO MEJOR”
- Ver todas la Área en las que trabaja la ONG
- Ver las Noticias actuales y actualizadas
- Generar una Consulta Legal
- Incorporar comentarios
- Informarse sobre los delitos que se tiene en Bolivia
- Acceso al diccionario

Modulo Beneficiario

- Ver toda la información del Abogado
- Ver todas las actividades de la ONG “MUNDO MEJOR”
- Ver todas las Áreas en las que trabaja la ONG
- Ver las Noticias actuales y actualizadas
- Incorporar comentarios
- Informarse sobre los delitos que se tiene en Bolivia

- Acceso al diccionario
- Según al delito donde se debe apersonarse
- Id y Clave
- Acceso a su caso legal
- Seguimiento de los antecedentes caso legal
- Enviar, guardar y subir información.

Modulo Abogado

- Acceso al caso legal
- Dirección actual donde se ubicara el caso legal
- Dirección del beneficiario
- Delito por el cual se está ejecutando el caso legal
- Actualizar su curriculum
- Acceso al diccionario
- Acceso a los antecedentes
- Acceso a la información del familiar del beneficiario
- Acceso a Guardar, Subir, Adicionar, Modificar y enviar.
- Acceso a imprimir

2.3 Características de los usuarios

Tipo de usuario	Visitante
Formación	Secundaria
Habilidades	Conocimientos de PC básicos
Actividades	Ver, registrarse,

	solicitud de ayuda, conexión, realizar consultas.
Tipo de usuario	Beneficiario
Formación	Secundaria
Habilidades	Conocimientos de PC básicos
Actividades	Ver, Adicionar, subir, guardar, realizar consultas, imprimir

Tipo de usuario	Administrador
Formación	Profesional según la ONG “MUNDO MEJOR”
Habilidades	Conocimientos de PC básicos
Actividades	Ver, Adicionar, Modificar, Borrar, guardar, subir, realizar consultas, incorporar noticias, actividades, generar resguardos, imprimir

Tipo de usuario	Abogado
Formación	Profesional según la ONG “MUNDO

	MEJOR”
Habilidades	Conocimientos de PC básicos
Actividades	Ver Adicionar, Modificar, algún tipo de dato del caso legal que se le fue asignado previamente cualquier modificación el sistema lo observara y guardar en el sistema, realizar consultas, incorporar noticias, actividades, acceso a las pantallas, imprimir

El sistema de información deberá ofrecer una interfaz de usuario estándar, fácil de aprender y sencillo de manejar. Lo deseable sería que un tenga conocimientos básicos de Windows a nivel usuario, con el único prerrequisito de ser una persona acostumbrada al uso de un navegador web, se familiarizase con el sistema en menos de una semana.

El Sistema deberá presentar un alto grado de usabilidad y entendimiento.

2.4 Restricciones

La aplicación se realizará con un lenguaje de programación Orientado a Objetos: JAVA software libre.

Se debe Contar con un Servidor Web que presente las características necesarias para la ejecución del Sistema.

La Base de datos a utilizar será SQL Postgres

El sistema será implementado en las plataformas Windows.

El sistema no soporta un módulo contable.

El sistema no visibilizara la documentación de los casos legales a los visitantes.

2.5 Suposiciones y dependencias

2.5.1 Suposiciones

En el documento se expresan los requisitos en términos de lo que el sistema debe proporcionar a los usuarios que acceden a él. No obstante, Se tendrá a alguien como administrador que será responsable de realizar una carga inicial de datos en la Base de Datos.

Interés de la institución para utilizar el Sistema.

Disponibilidad de Equipos de computación.

Disponibilidad de un dominio.

Disponibilidad de un espacio en la Web Hosting.

Disponibilidad de la Automatización de la Información.

2.5.2 Dependencias

El sistema funcionará, sin necesidad de comunicarse con otros sistemas similares, por lo que no hay dependencias respecto de otros sistemas.

El sistema se deberá actualizarse constantemente en las imágenes de presentación, documentación de los Abogados, talleres y noticias para una buena usabilidad del mismo para el cliente.

2.6 Evolución previsible del sistema

En un futuro se prevé incorporar nuevos módulos según el crecimiento de la ONG “MUNDO MEJOR” como por ej. Incorporar un modulo para psicología y modulo de reportes económicos.

3. Requisitos Específicos

En este apartado se presentan los requisitos funcionales y no funcionales que deberán ser satisfechos por el sistema.

Cada Requisito será identificado por un número, además presentaran un nombre breve en el caso de los requisitos funcionales; Cada requisito tendrá además asignado una prioridad de implementación, en la que consideraremos los siguientes niveles:

Prioridad Baja: Es para los requisitos que no son necesarias ahora, pero deberán incorporarse en el medio-largo plazo.

Prioridad Media: Es para los requisitos que necesitan más discusión.

Prioridad Alta: Es para los requisitos cuya ausencia no sería tolerable en el futuro sistema

Esto nos permitirá, probar fácilmente el cumplimiento de los mismos por parte del sistema y acompañar la evolución natural del Sistema ante los cambios en los requerimientos con las prioridades que estos presentan.

3.1 Requisitos comunes de los interfaces

3.1.1 Interfaces de usuario

Requerimiento (1) Prioridad: Alta

La Interfaz de Usuario tendrá a su disposición menús para obtener información de todo el sistema, accesibles a través de un navegador Web, una interacción fácil e intuitiva entre el usuario y el sistema.

3.1.2 Interfaces de hardware

No se han definido.

3.2.3 Interfaces de software

Requerimiento (2) Prioridad: Media

Se deberá proveer un mecanismo para poder acceder a la Base de Datos, en la cual esto nos permita acceder mediante el esquema de seguridad adoptado para la BD a los datos almacenados

3.1.4 Interfaces de comunicación

Requerimiento (3) Prioridad: Alta

Uso del espacio web, aun no se sabe si se va a comprar o se buscar una manera gratuita o buscar un financiador según la ONG “MUNDO MEJOR”.

3.2 Requisitos funcionales

3.2.1 Visitantes

Ver contenido de la página web

Llenado de antecedentes del beneficiario.

La ONG deberá cumplir con todos los requisitos para la ayuda a las personas.

La actividad económica no es lo más importante si no es la actividad social que pueda generar mediante el área legal

Solucionar el problema legal con la rapidez posible.

Seguimiento del caso constante.

3.2.2 Beneficiario de la ONG

Se determina el monto del costo de caso legal reducido o si es gratuito

Seguimiento del caso

Actualización de los datos del caso legal mediante el Abogado

Actualización de la localización del caso mediante el Abogado

Ver contenido de la página web (ver anexo)

3.2.3 Abogados

Determina el monto del costo de caso legal reducido o si es gratuito

Seguimiento del caso

Actualización de los datos del caso legal.

Actualización de la localización del caso

Ver estadísticas de casos legales

Generar reportes de casos legales

3.2.4 Administrador

Determinara las actualizaciones del sistema

Incorpora nuevas actividades dentro del sistema

Incorpora nuevas areas de trabajo de la ONG “MUNDO MEJOR”.

3.2.5 Consultas y Listados

Requerimiento (4) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de un listado simple de los Beneficiarios. Los atributos a mostrar por cada Usuario podrán ser elegidos por el usuario (Administrador) y solo podrán ver los usuario (abogado) asignados al caso, el administrador, se podrá buscarlo por (nombre, código).

Requerimiento (5) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de la identificación de la ONG “MUNDO MEJOR” mediante un lema que ellos tienen “Quienes Somos”, será incorporado por el (Administrador), será parte visible del sistema.

Requerimiento (6) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de un listado simple de las áreas de trabajo. Los atributos a mostrar serán por el usuario (Administrador) el tipo de área y su respectiva descripción, permitirá ver que tipo de áreas puede adecuar su caso el visitante.

Requerimiento (7) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de un listado simple de los Abogados que pertenecen a la ONG. Los atributos a mostrar serán por el usuario (Administrador) será abierto para cualquier persona que ingrese al sistema.

Requerimiento (8) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de un listado simple de las Noticias más relevantes que considera la ONG. Los atributos a mostrar serán por el usuario (Administrador) será abierto para cualquier persona que ingrese al sistema.

Requerimiento (9) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de comentarios que puedan brindar cualquier visitante a la página que pertenecen a la ONG. Los atributos a mostrar serán por el usuario (Administrador) será abierto para cualquier persona que ingrese al sistema.

Requerimiento (10) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla del curriculum que pertenece al abogado para poder permitir al visitante optar por esa persona a llevar algún caso que requiera. Los atributos a mostrar serán por el usuario (Administrador) será abierto para cualquier persona que ingrese al sistema.

Requerimiento (11) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de generar una consulta sobre algún delito. Los atributos a mostrar serán por el usuario (Administrador) será abierto para cualquier persona que ingrese al sistema.

Requerimiento (12) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de poder buscar en diccionario de alguna palabra en común con la ONG, ya se sobre algún delito o palabras que pertenezcan al área que se trabaja. Los atributos a mostrar serán por el usuario (Administrador) será abierto para la persona que ingrese al sistema.

Requerimiento (13) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de incorporar su dirección actual para el seguimiento del caso legal. Los atributos a mostrar serán por el usuario (Administrador) será abierto para la persona que ingrese al sistema.

Requerimiento (13) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de un código (id, clave) para el seguimiento del caso legal si en algún momento pueda ingresar para incorporar nuevos atributos. Será otorgado para la persona que ingrese al sistema y así lo requiera tendrá un tiempo de caducación.

Requerimiento (14) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de la ubicación actual a donde se debe apersonar para el seguimiento del caso legal. Los atributos a mostrar serán por el sistema y será abierto para la persona que ingrese al sistema.

Requerimiento (15) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de los antecedentes del Beneficiario para el seguimiento del caso legal. Los atributos a mostrar serán por el sistema y será abierto para el Beneficiario y para el abogado.

Requerimiento (16) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de Beneficiario con los datos personales del beneficiario para el seguimiento del caso legal. Los atributos a mostrar serán por el sistema y será abierto para el abogado que su adjudique el caso legal.

Requerimiento (16) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de Dirección con sus respectivos datos de la ubicación del beneficiario para el seguimiento del caso legal. Los atributos a mostrar serán por el sistema y será abierto para el abogado que su adjudique el caso legal.

Requerimiento (16) *Prioridad: Alta*

El sistema permitirá la presentación en pantalla de Familiar con los datos personales del parentesco del beneficiario para el seguimiento del caso legal. Los atributos a mostrar serán por el sistema y será abierto para el abogado que su adjudique el caso legal.

3.2.6 Mantenimiento de datos (ver, adicionar, borrar, modificar, buscar, opinar, consultar)

Requerimiento (17) *Prioridad: Alta*

El usuario (Visitante) del sistema tendrán acceso a una función básica de mantenimiento de los datos de (ver, buscar, opinar), en las pantallas que tiene acceso.

Requerimiento (18) *Prioridad: Alta*

El usuario (Beneficiario) del sistema tendrán acceso a una función básica de mantenimiento de los datos de (subir, descargar, ver, buscar, opinar, consultar), en las pantallas que tiene acceso.

Requerimiento (19) *Prioridad: Alta*

El usuario (Abogado) del sistema tendrán acceso a una función básica de mantenimiento de los datos de (ver, buscar, opinar, adicionar, borrar, modificar, consultar), en las pantallas que tiene acceso.

Requerimiento (19) *Prioridad: Alta*

El usuario (Administrador) del sistema tendrán acceso a una función básica de mantenimiento de los datos de (ver, buscar, opinar, adicionar, borrar, modificar, consultar), en las pantallas que tiene acceso.

3.2.7 Elaboración de Reportes – Estadísticas

Requerimiento (20) *Prioridad: Alta*

Los usuarios del sistema tendrán acceso a una función de estadística de los Visitantes del sistema web.

Requerimiento (21) *Prioridad: Alta*

Los usuarios del sistema tendrán acceso a una función de estadística de los Beneficiarios del sistema web.

Requerimiento (21) *Prioridad: Alta*

Los usuarios del sistema tendrán acceso a una función de reporte de los Visitantes que llegaron a convertirse Beneficiario del sistema web.

Requerimiento (22) *Prioridad: Alta*

Los usuarios del sistema tendrán acceso a una función de reporte de los Beneficiario que llegaron a concluir su caso con éxito mediante la información incorporada al sistema web.

Requerimiento (23) *Prioridad: Alta*

Los usuarios del sistema tendrán acceso a una función de reporte de los Beneficiario que se quedo su caso sin concluir mediante la información incorporada al sistema web.

Requerimiento (24) *Prioridad: Alta*

Los usuarios del sistema tendrán acceso a una función de reporte de beneficiario y visitantes dados de alta y baja.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

Requerimiento (26) *Prioridad: Alta.*

El tiempo de respuesta en las operaciones debe ser el esperable en cualquier aplicación Web, compra de un Dominio, aplicación de web Hosting, comparable al del Sistema Tariquia de la “UAJMS”, por ejemplo.

Requerimiento (27) *Prioridad: Media.*

El tiempo de respuesta a las consultas estadísticas no pueden superar los 45 segundos y las consultas de reportes no pueden superar los 60 segundos.

3.3.2 Requisitos de desarrollo

Requerimiento (28) *Prioridad: Alta.*

El ciclo de vida elegido para desarrollar el producto será el de RUP (Racional Unified Process), 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.

3.3.3 Requisitos tecnológicos

Requerimiento (29) *Prioridad: Alta.*

En cuanto a la infraestructura informática necesaria es claro que se requiere un servidor Web, con un Sistema de Gestión de Base de Datos: La opción encontrada es SQL Server, con tendencias a evaluar otras alternativas.

3.3.4 Requisitos de Eficiencia *Prioridad Alta*

En cuanto a la eficiencia se pretende implementar a una web hosting.

3.3.5 Seguridad

Requerimiento (30) *Prioridad: Alta.*

Cuando un usuario (Beneficiario, Abogado, Administrador) puede conectarse al sistema para tener acceso a otras alternativas de pantallas deberá introducir su id y clave de acceso y el sistema deberá comprobar que se trata de un usuario autorizado.

En cuanto al visitante será abierto el sistema sin introducir ningún tipo de id y clave.

Se pretende a lo largo del sistema y según su vida útil incorporar tendencias de las Aplicaciones SANS.

Requerimiento (31) *Prioridad: Alta.*

El acceso al sistema se realizará exclusivamente mediante conexión cifrada, de modo que se preserve la privacidad de los datos de nombre de id y clave.

Requerimiento (32) *Prioridad: Alta.*

El sistema de información tendrá distintos tipos de usuarios y a cada uno de ellos se le permitirá únicamente el acceso a las funciones que le correspondan.

Requerimiento (33) *Prioridad: Alta.*

El Sistema proporcionará además funciones de auditoria, registrando la fecha, hora y usuario de todo cambio realizado.

Requerimiento (34) *Prioridad: Alta.*

El procedimiento de recuperación luego de una caída debe estar documentado.

Requerimiento (35) *Prioridad: Alta.*

El procedimiento de copias de backup y su resguardo debe estar documentado.

3.3.6 Fiabilidad

Requerimiento (36) *Prioridad: Alta.*

Cualquier transacción finalizada por un usuario deberá ser procesada exitosamente con una tasa de error del 0%. En el caso de un error de procesamiento, la transacción no se deberá considerar finalizada.

3.3.7 Disponibilidad

Requerimiento (37) *Prioridad: Alta.*

Por determinar.

3.3.8 Mantenibilidad

Requerimiento (38) *Prioridad: Alta.*

La entrada de los datos se debe hacer continuamente para interacción con el usuario (Visitante y Beneficiario), para que permita la actualización de todos los datos estadísticos.

Requerimiento (39) *Prioridad: Media.*

Los datos reportes no deben necesitar de la intervención de un operador.

Requerimiento (40) *Prioridad: Alta.*

El sistema requerirá el mínimo posible de mantenimiento y, siempre que sea posible, las tareas de mantenimiento deberían realizarse vía Web o en tal caso en el servidor alojado en el que se encuentra el sistema.

3.3.9 Portabilidad

Requerimiento (41) Prioridad: Alta.

La totalidad del código desarrollado deberá ser compatible con las siguientes plataformas:

Windows o Linux

SQL Server 2008

3.4 Otros requisitos

3.4.1 Información histórica

Con el propósito de generar información se deberá almacenar de los años anteriores (formularios en papel de gestiones pasadas) los siguientes datos:

Beneficiarios

Visitantes

Casos concluidos con éxito

Casos concluidos sin éxito

3.4.2 Bases de Datos

En la base de datos se mantendrá información de toda la Area Legal de la ONG “MUNDO MEJOR”.

3.4.3 Operaciones

Al iniciar por primera vez se tendrá que incorporar algunos datos de gestión anterior para poder tener un dato estadístico para comparar como ah influido el sistema web ONG “MUNDO MEJOR”.

4. APÉNDICES

Información requerida y obtenida ONG “MUNDO MEJOR”.

MATRIZ DE MARCO LOGICO

Resumen Narrativo Del Proyecto	Indicadores	Medios de Verificación	Supuestos
Brindar un Servicio de calidad para la conexión entre las personas y la ONG “Mundo Mejor”	Al cierre de gestión 2012 la información obtenida por el sistema ayudara en un 70% para que las personas tengan una atención vía WEB.	La Base de datos del sistema entregado, su respectiva carta de aceptación por la ONG “Mundo Mejor”.	<ul style="list-style-type: none"> • El sistema ONG “Mundo Mejor” se utiliza con otros fines.
Mejorar los procesos administrativos y transferencia de información mediante un sistema informático para la ONG “MUNDO MEJOR”.	Al finalizar el proyecto la parte administrativa y transferencia de documentación será automatizada mejorando la búsqueda de información.	Carta firmada por la Directora de la ONG “Mundo Mejor”. Por la conformidad del Sistema.	<ul style="list-style-type: none"> • Voluntad y Predisposición de la ONG “MUNDO MEJOR”.
<p>Objetivos Específicos (Componentes)</p> <p>1. Sistema Informático para los procesos administrativos y transferencia de información</p>	<p>2.1. Al Finalizar el Proyecto el sistema para la administración y transferencia de información está desarrollado con sus aplicaciones.</p>	<p>Carta firmada por la Directora de la ONG “Mundo Mejor”. Por la conformidad del Sistema.</p>	<ul style="list-style-type: none"> • El personal hace uso del sistema. • El personal está dispuesto a capacitarse.

MATRIZ DE MARCO LOGICO

<p>2. Estrategias de Socialización del Sistema Informático dirigido a los miembros de la ONG “MUNDO MEJOR”</p>	<p>2.1. Al finalizar el proyecto se tendrá un taller de capacitación implementado en un 90% con los involucrados con la ONG “MUNDO MEJOR”.</p>	<p>2.1. Carta de solicitud para la socialización del sistema 2.2. Carta de solicitud del Laboratorio para la realización de la capacitación 2.3. Lista de Asistencia de la capacitación del Sistema. 2.4. Guías Positivas Utilizada en la Capacitación.</p>	<ul style="list-style-type: none"> • Los integrantes de la ONG “MUNDO MEJOR”. Tienen interés en el sistema.
<p>Actividades</p> <p>1. El sistema ONG “MUNDO MEJOR”</p> <ul style="list-style-type: none"> • Especificación de requerimientos. • Elaboración de los diagramas 		<p>5.1. Uso de Normas de modelado de los diagramas de análisis y diseño del sistema mediante la metodología de</p>	

MATRIZ DE MARCO LOGICO

<p>UML arquitectura del Sistema entre ellos Casos de Uso, Diagrama de clases y Diagrama secuencia.</p> <ul style="list-style-type: none"> • Diseño de la base de datos del Sistema. • Programación del Sistema. • Dominio Web. <p>Implementación del sistema en la Web hosting.</p> <p>2. Se generara participación a través de búsqueda de opiniones y sugerencias.</p> <p>3. Se Administrara la información de sus consultas de casos legales en una base de datos segura y confiable.</p> <p>4. Definición de la metodología</p>		<p>desarrollo de sistemas RUP.</p> <p>5.2.Documentación del Sistema.</p> <p>5.3.Manual de Usuario</p>	
--	--	---	--

MATRIZ DE MARCO LOGICO

<p>de enseñanza.</p> <p>5. Realizar la formación al personal en el uso de utilidades del sistema.</p>			
---	--	--	--

Componente 1

II.1.1. Introducción

Este componente busca una relación mas activa con las persona que tienen dificultades con algún caso legal que ellos mismos desconocen para ello el sistema brindara información actualizada y una interactividad acorde al conocimiento en informática y el Internet.

La atención se efectuara con prioridad a las personas de escasos recursos económicos.

II.1.2. Nombre

Sistema Informático para la ONG “MUNDO MEJOR” desarrollado

II.1.3. Indicador

Al Finalizar el Proyecto el sistema para la administración y transferencia de información está desarrollado con sus aplicaciones.

II.1.4. Actividades del Componente 1.

II.1.1.4.1. Especificación de requerimientos.

II.1.1.4.2. Elaboración de los diagramas UML arquitectura del Sistema entre ellos:

II.1.1.4.2.1. Casos de Uso

II.1.1.4.2.2. Diagrama de clases

II.1.1.4.2..3. Diagrama secuencia

II.1.1.4.2.4. Diseño de la base de datos

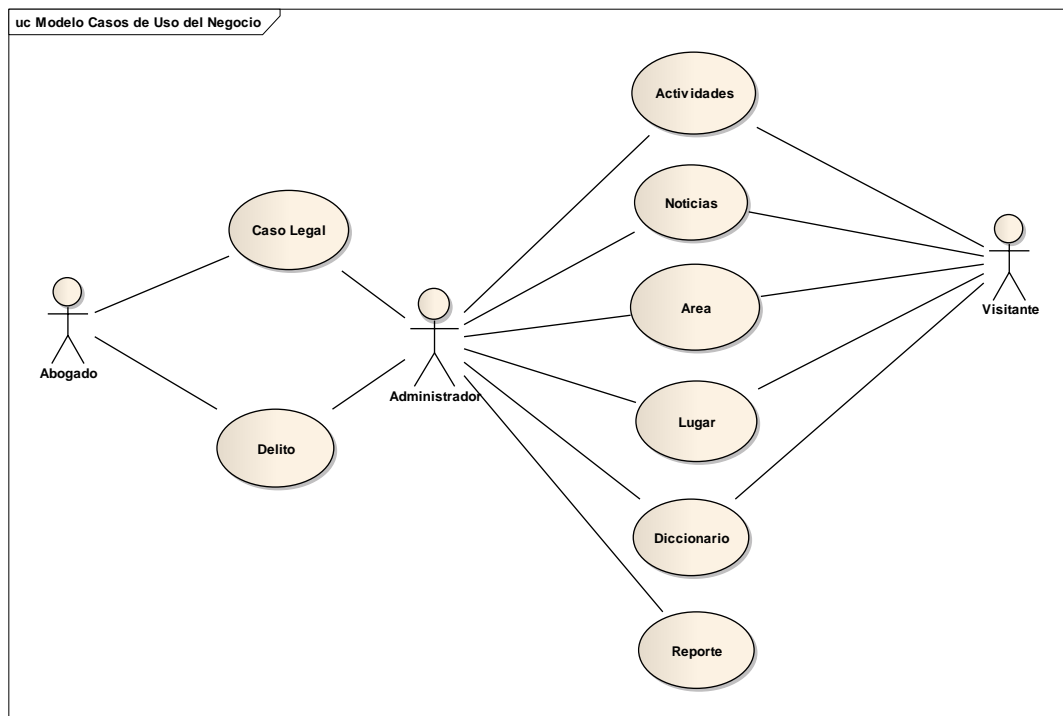
II.1.1.4.2.5. Programación del Sistema.

II.1.4.3. Se generara participación a través de búsqueda de opiniones y sugerencias.

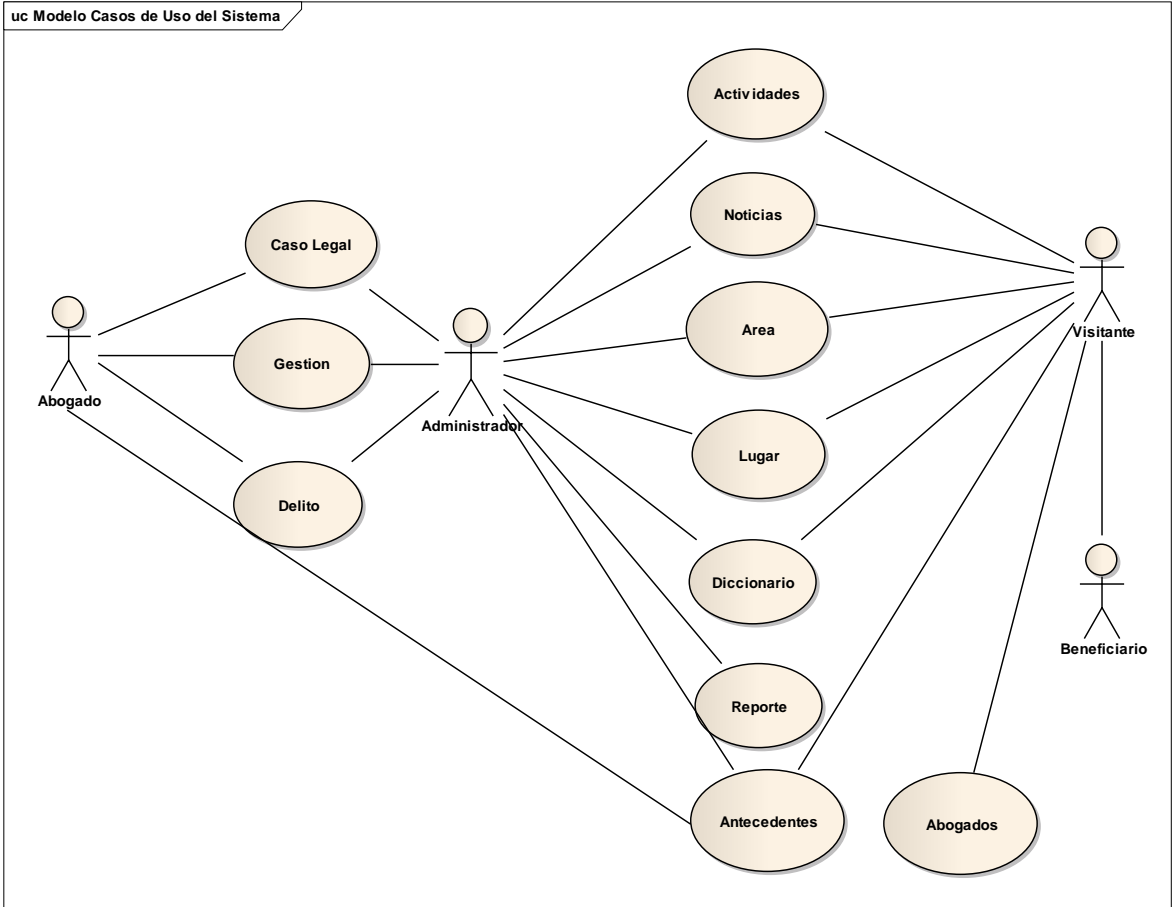
II.1.4.4. Se Administrara la información de sus consultas de casos legales en una base de datos segura y confiable.

II.1.5. Especificación de Requerimientos

II.1.5.1. Modelo de Casos de Uso del Negocio



II.1.5.2. Modelo de Casos de Uso del Sistema



II.1.5.2. Casos de Uso del Visitante

II.1.5.2.1. CASO DE USO INGRESANDO AL SISTEMA

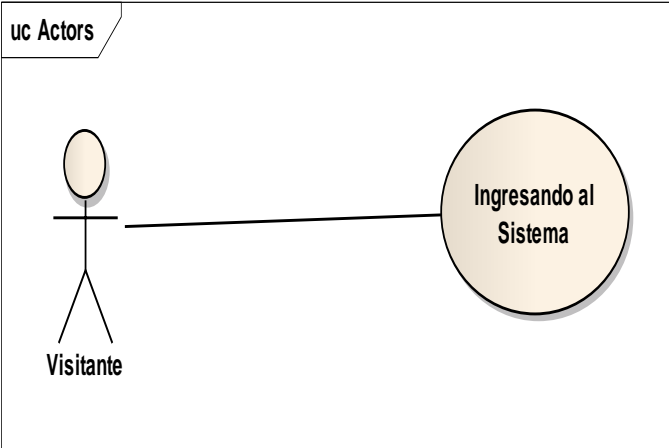


Figura 1.

II.1.5.2.2. CASO DE USO ABOGADO

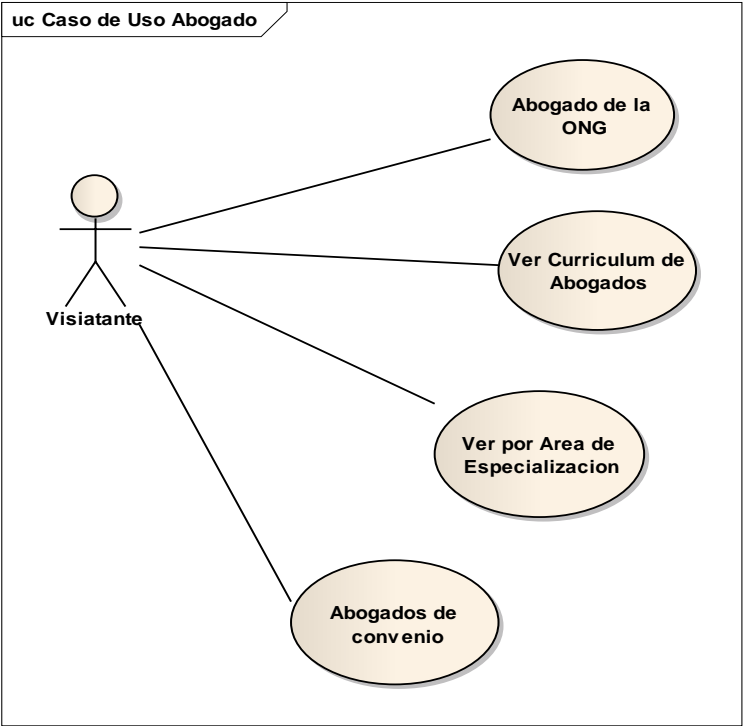


Figura 2.

II.1.5.2.3. CASO DE USO VER CURRICULO DEL ABOGADO

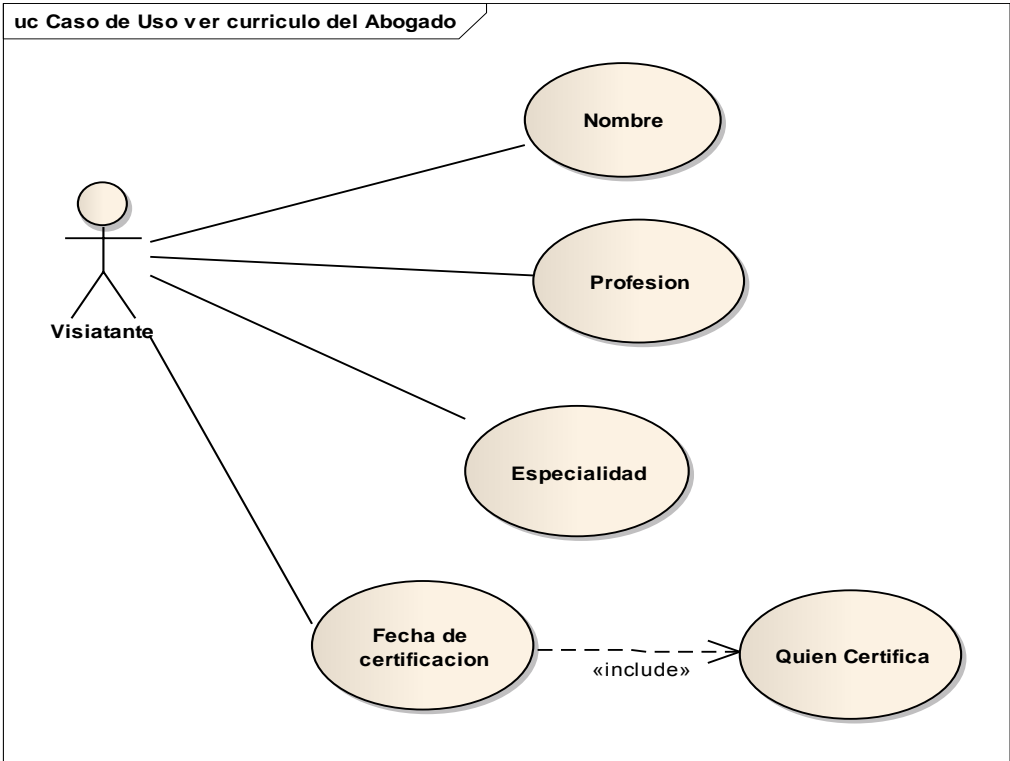


Figura 3.

II.1.5.2.4. CASO DE USO BUSCAR ABOGADO

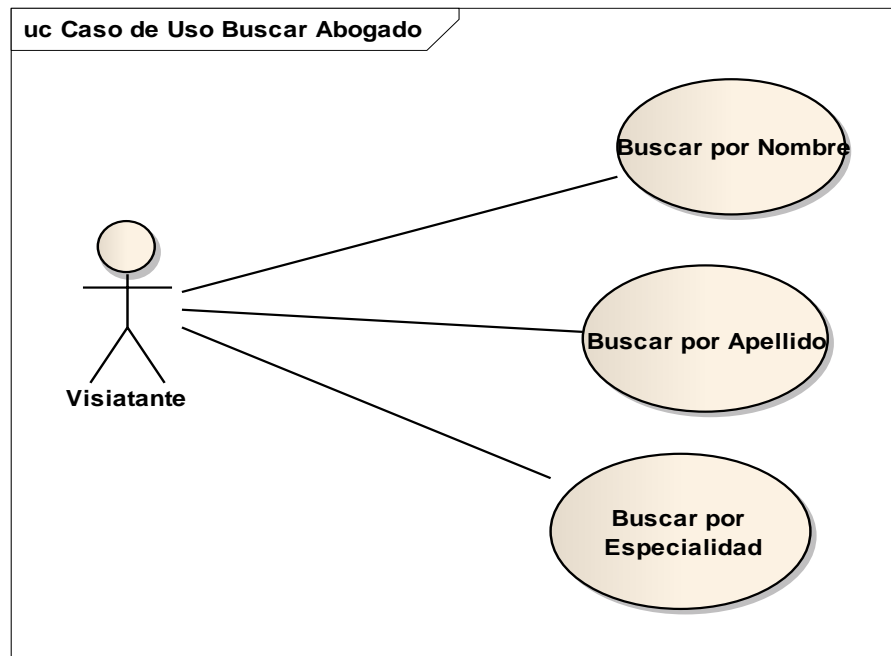


Figura 4.

II.1.5.2.5. CASO DE USO ACTIVIDADES DE LA ONG

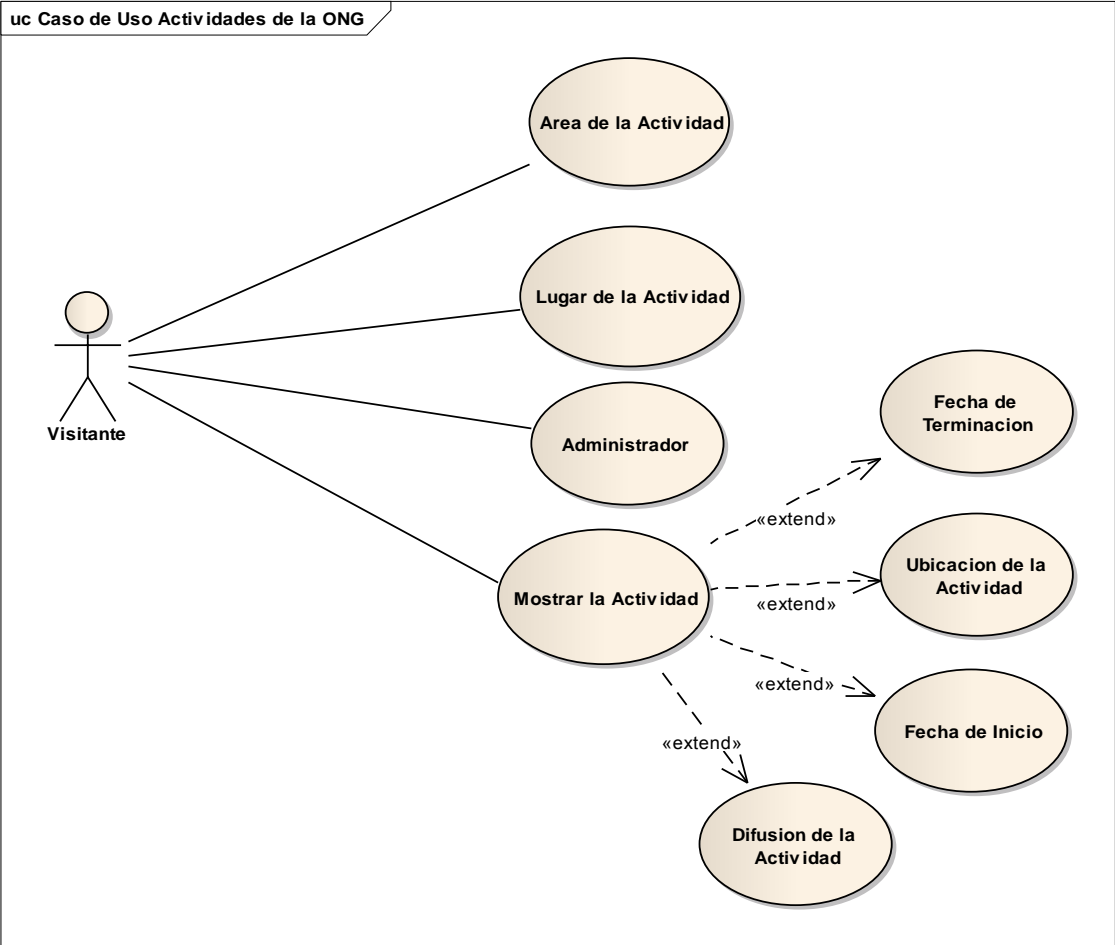


Figura 5.

II.1.5.2.6. CASO DE USO MOSTRAR ACTIVIDAD

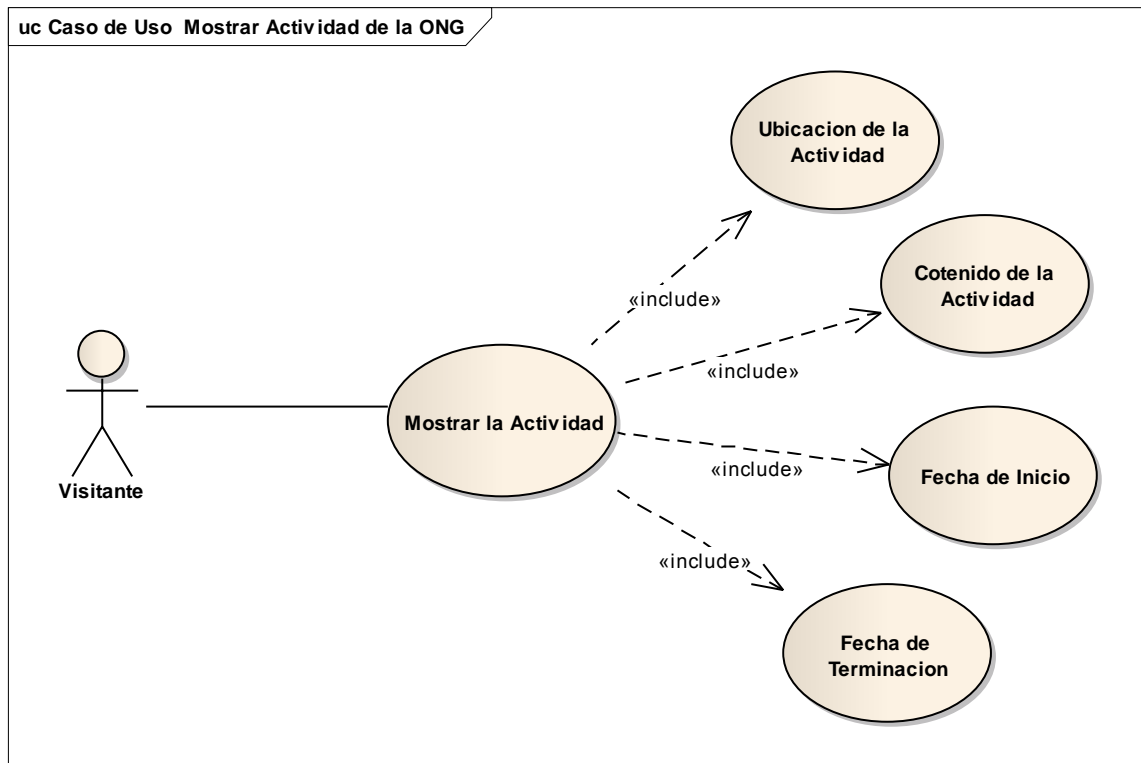


Figura 6.

II.1.5.2.7. CASO DE USO AREAS DE TRABAJO DE LA ONG

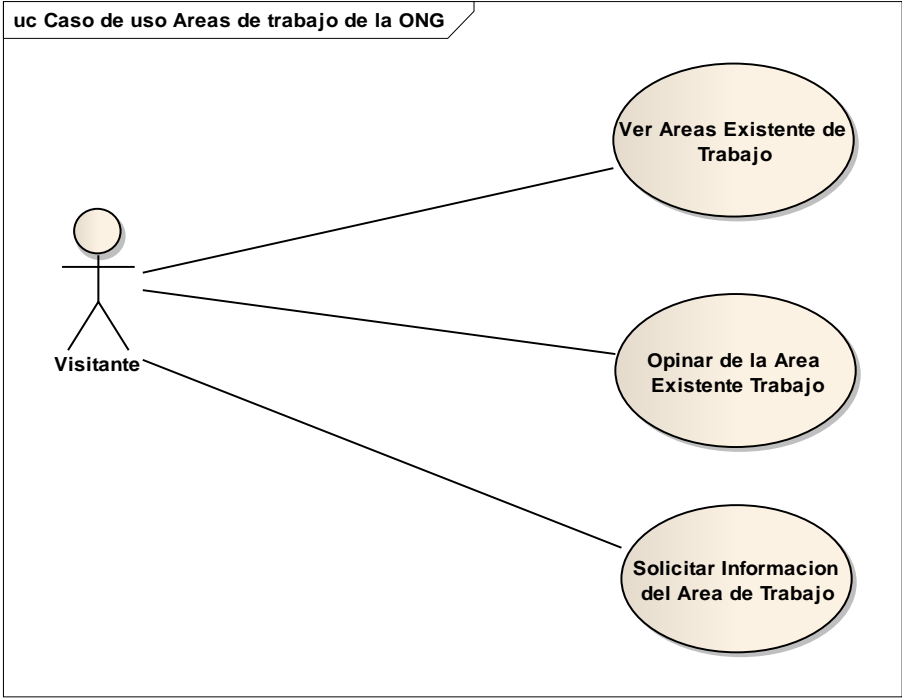


Figura 7.

II.1.5.2.8. CASO DE USO NOTICIA

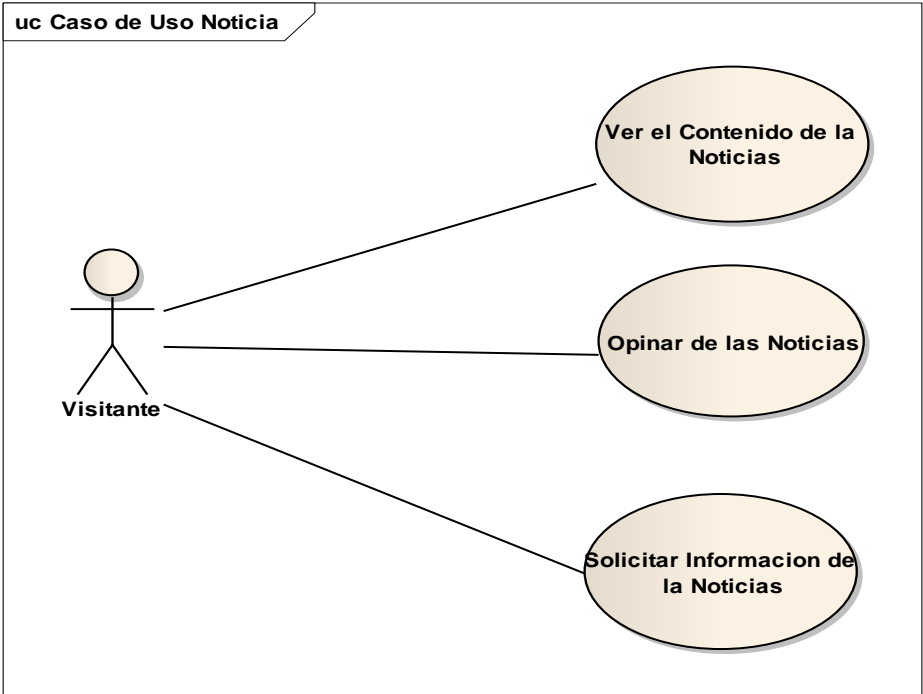


Figura 8.

II.1.5.2.9. CASO DE USO ANTECEDENTES DEL CASO

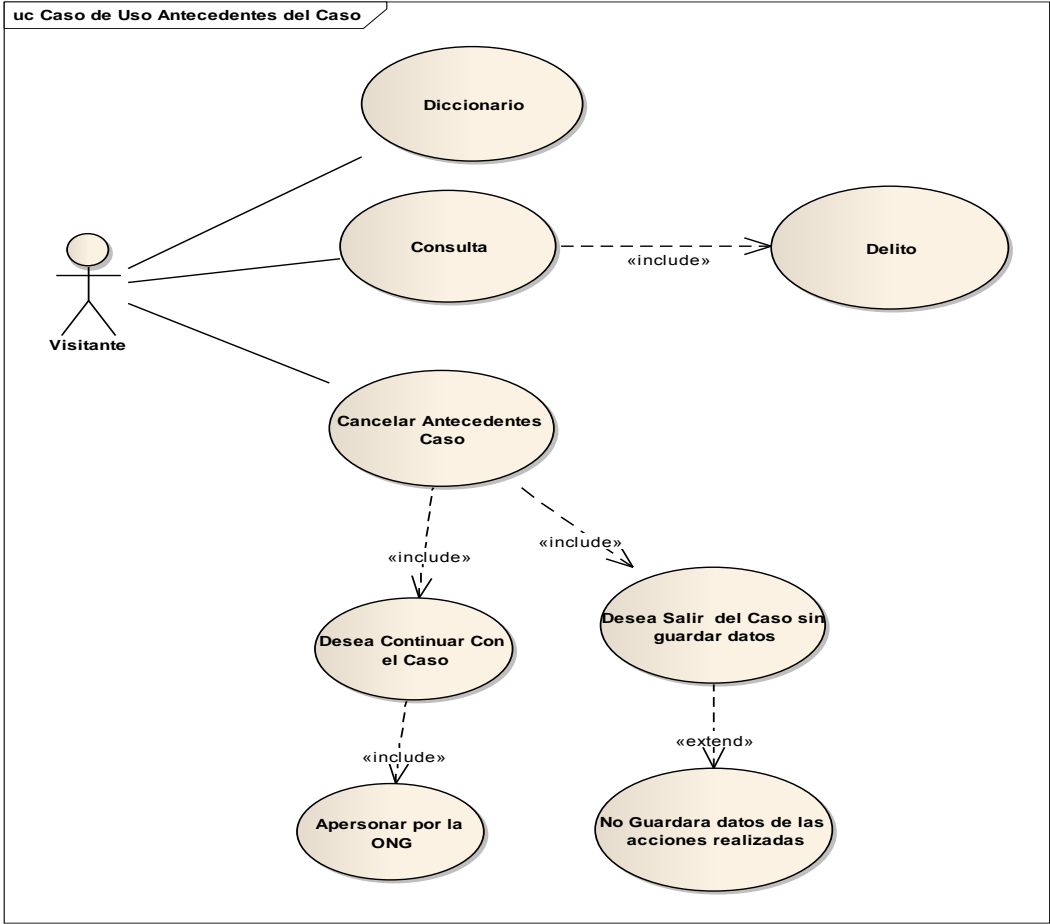


Figura 9.

II.1.5.2.10. CASO DE USO INGRESANDO AL SISTEMA ADMINISTRADOR

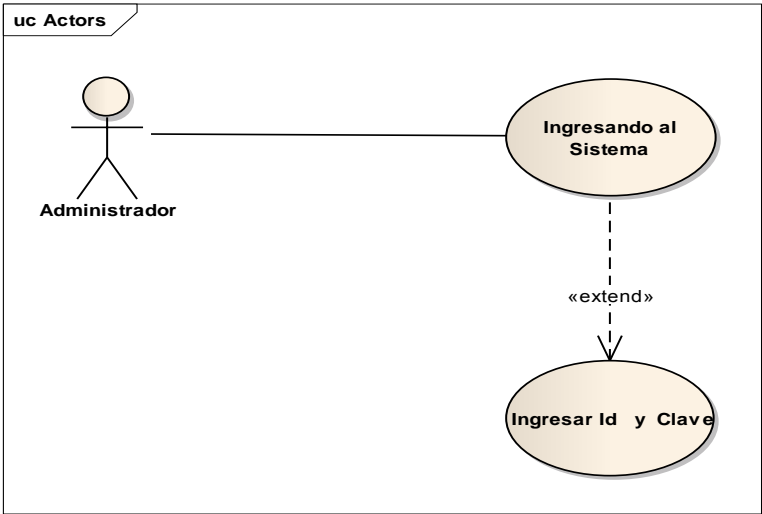


Figura 10.

II.1.5.2.11.- CASO DE USO ADMINISTRADOR ABOGADO

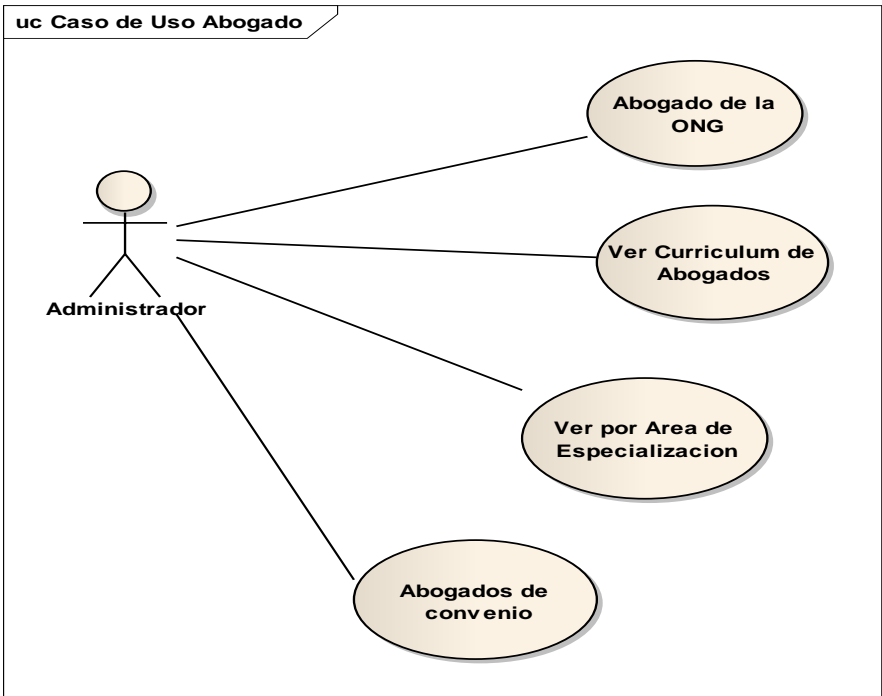


Figura 11.

II.1.5.2.12. CASO USO VER CURRICULO DEL ABOGADO

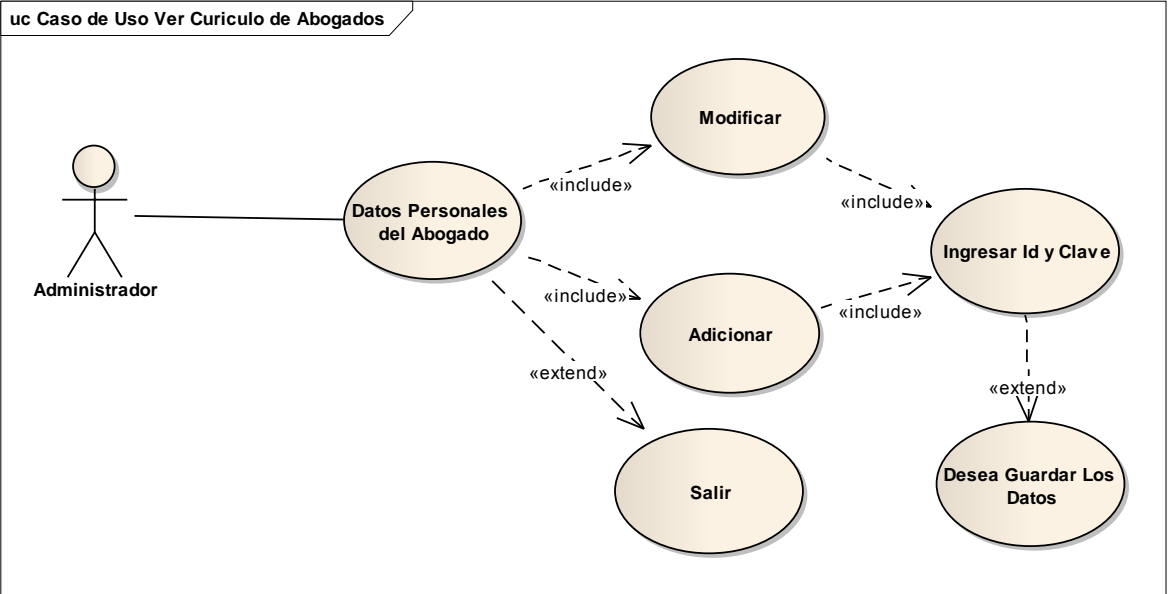


Figura 12.

II.1.5.2.13. CASO DE USO BUSCAR ABOGADO

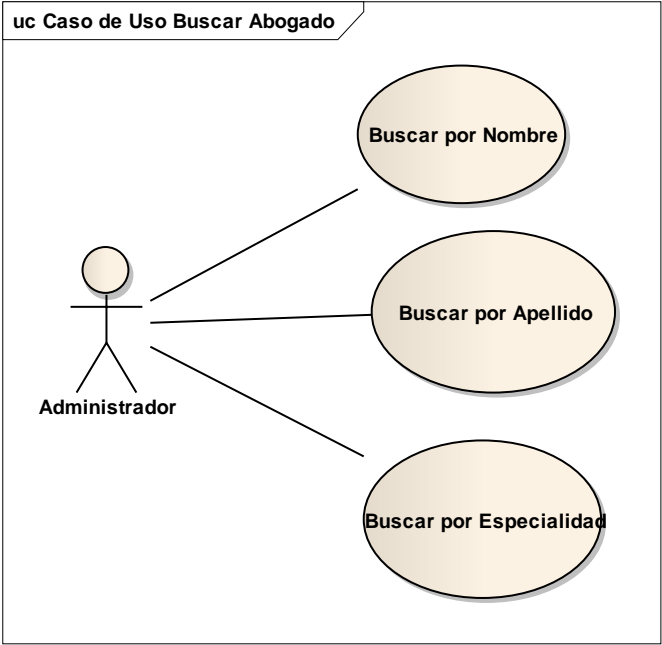


Figura 13.

II.1.5.2.14. CASO DE USO ACTIVIDAD DE LA ONG

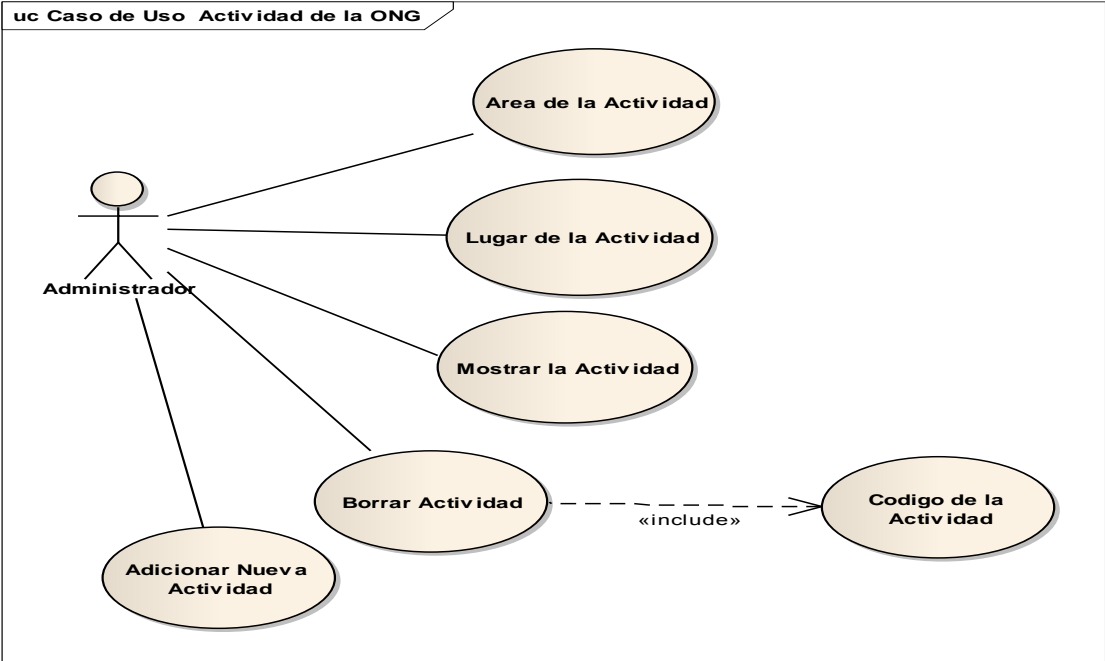


Figura 14.

II.1.5.2.15. CASO DE USO MOSTRAR ACTIVIDAD

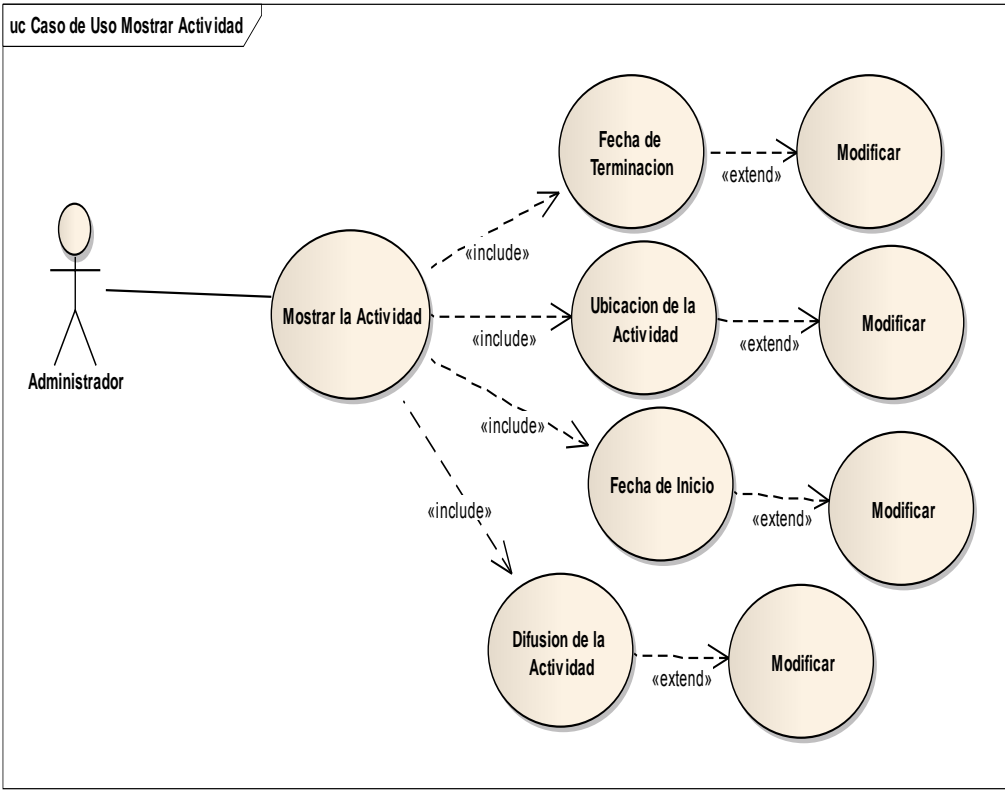


Figura 15.

II.1.5.2.16. CASO DE USO ADICIONAR NUEVA ACTIVIDAD

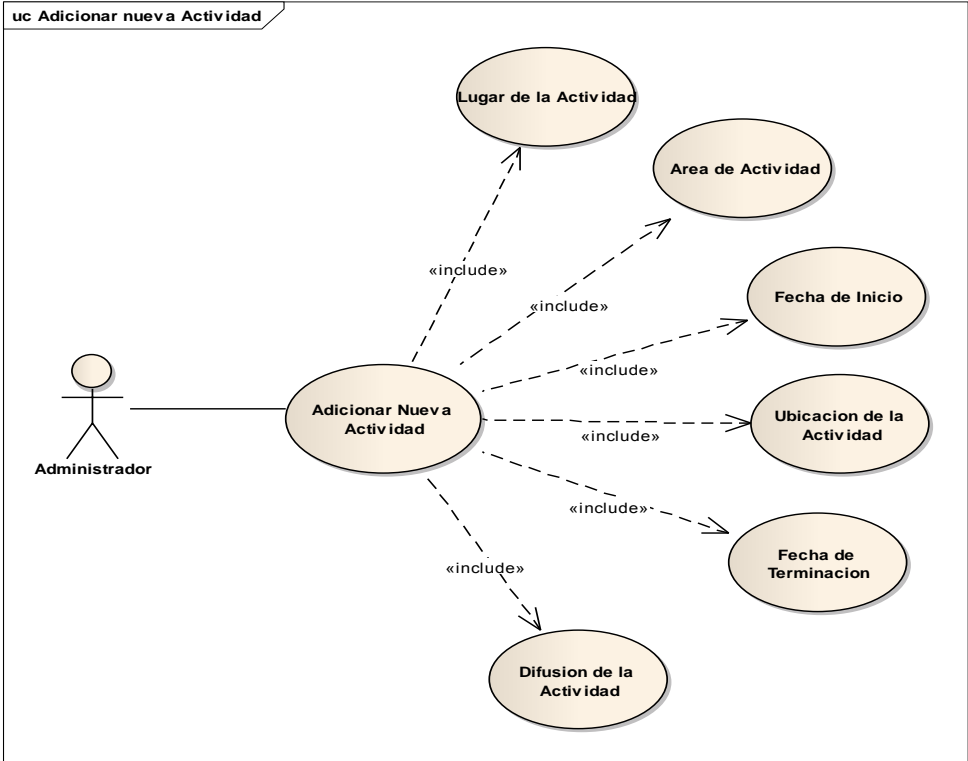


Figura 16.

II.1.5.2.17. CASO DE USO AREAS DE TRABAJO DE LA ONG

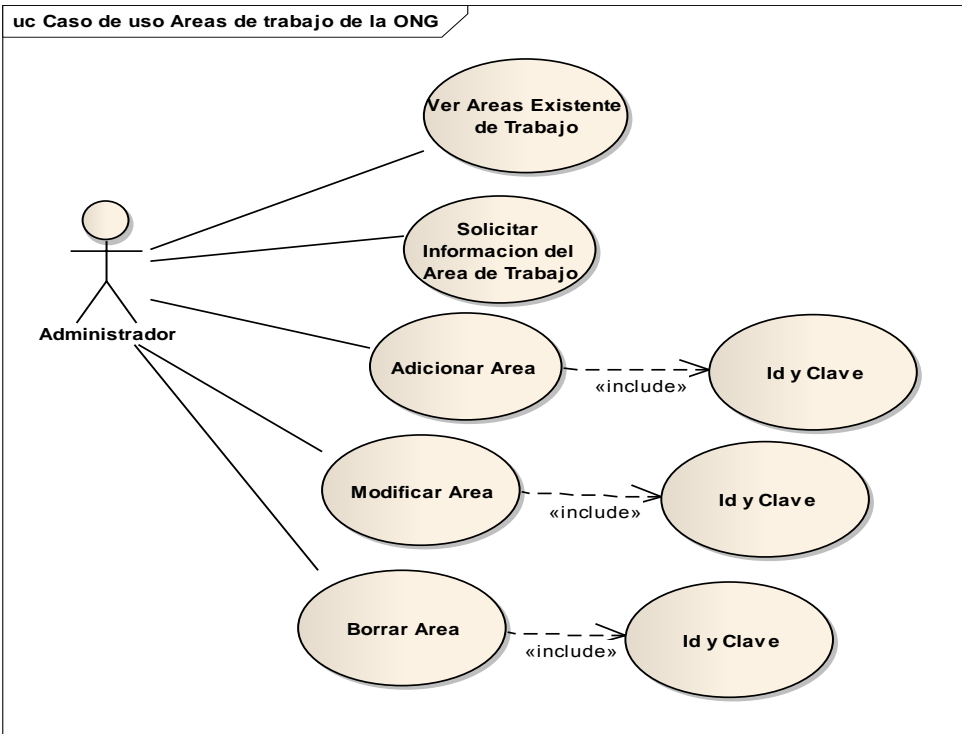


Figura 17.

II.1.5.2.18. CASO DE USO ADICIONAR AREA

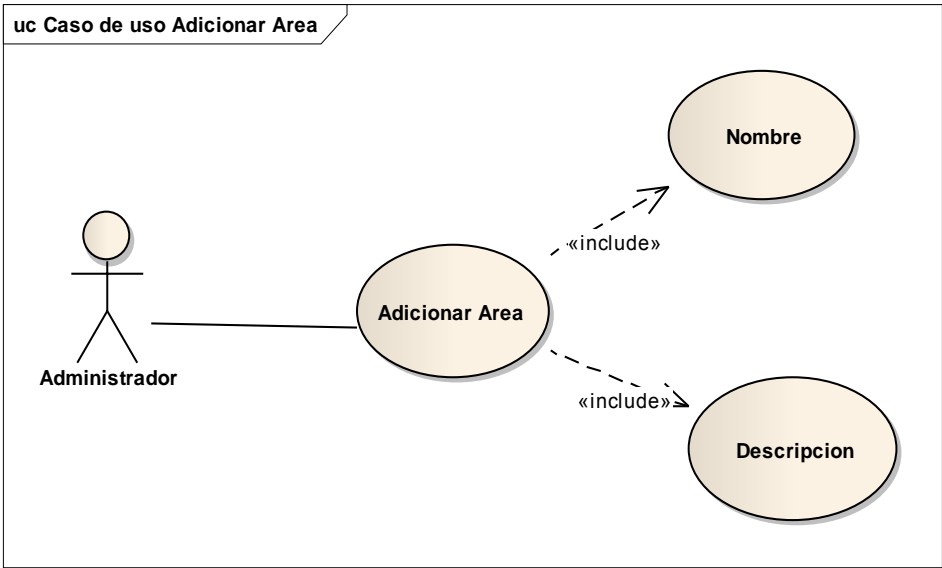


Figura 18.

II.1.5.2.19. CASO DE USO BORRAR AREA

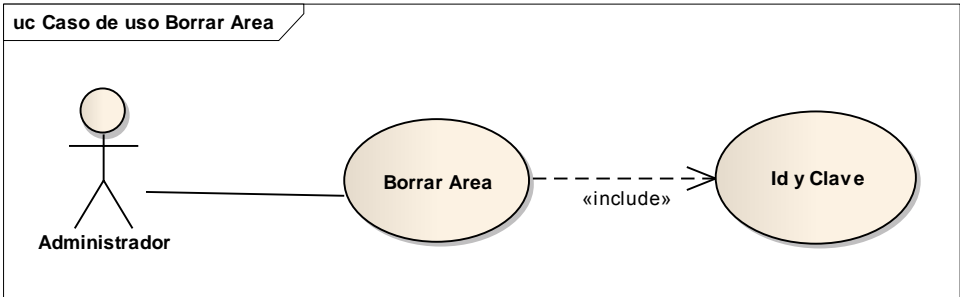


Figura 19.

II.1.5.2.20. CASO DE USO MODIFICAR AREA

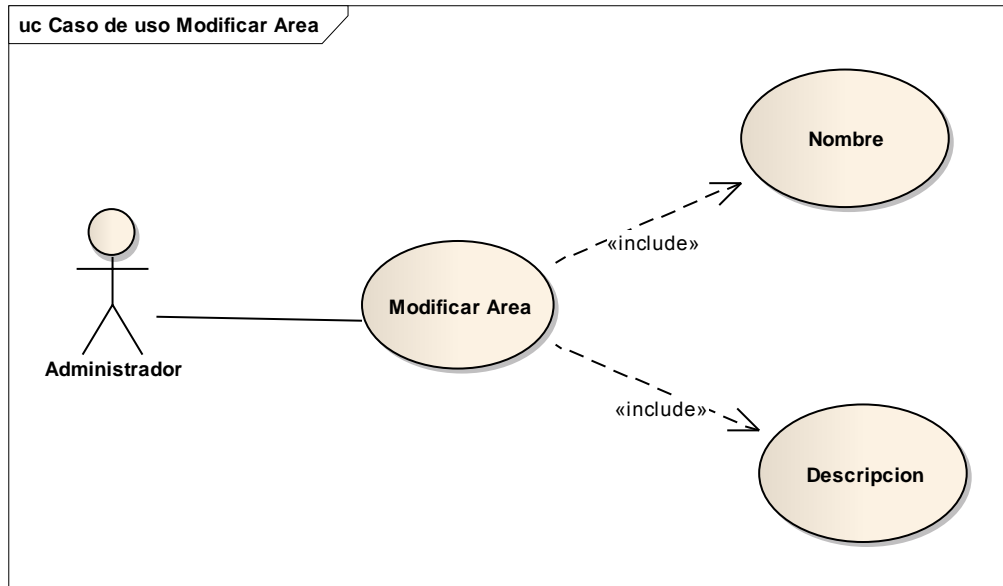


Figura 20.

II.1.5.2.21. CASO DE USO NOTICIA

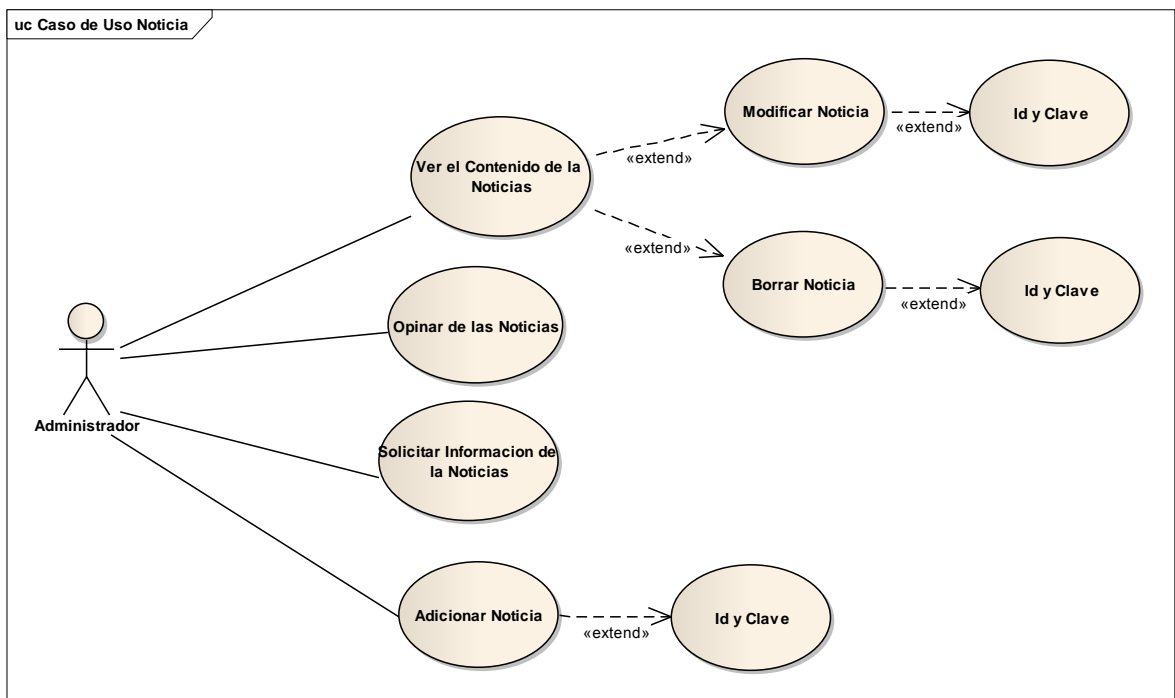


Figura 21.

II.1.5.2.22. CASO DE USO NOTICIA

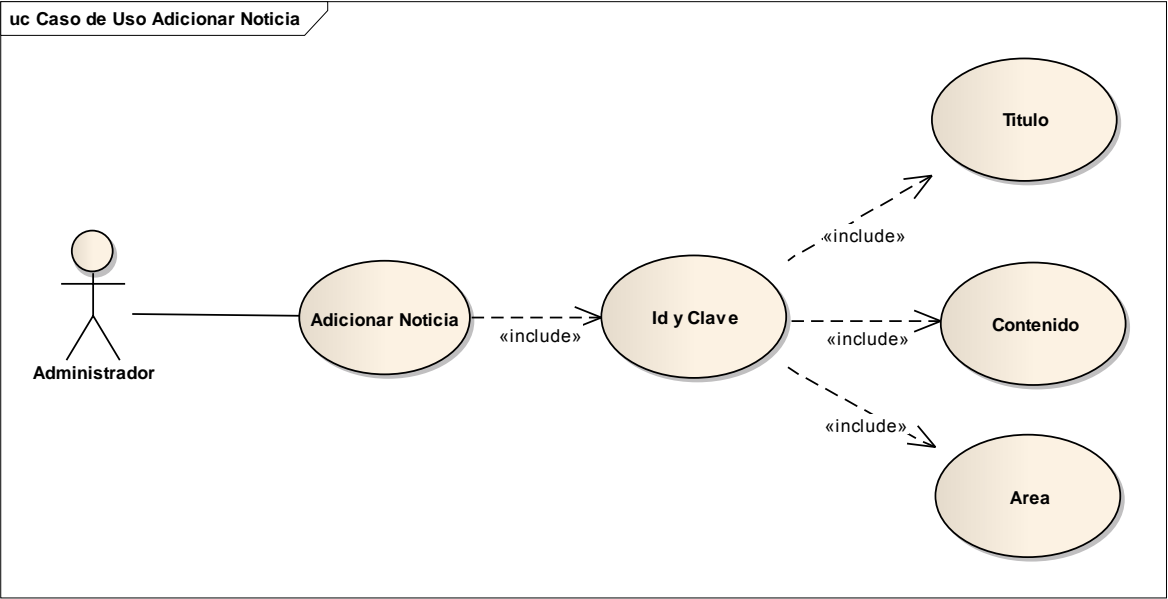


Figura 22.

II.1.5.2.23. CASO DE USO MODIFICAR NOTICIA

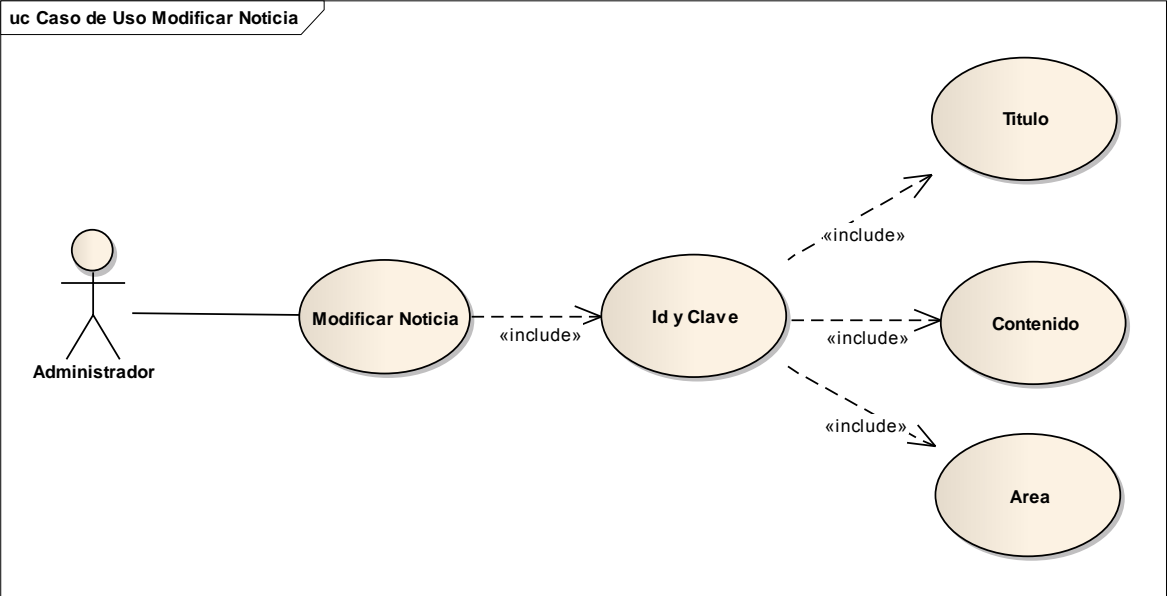


Figura 23.

II.1.5.2.24. CASO DE USO ANTECEDENTES DEL CASO

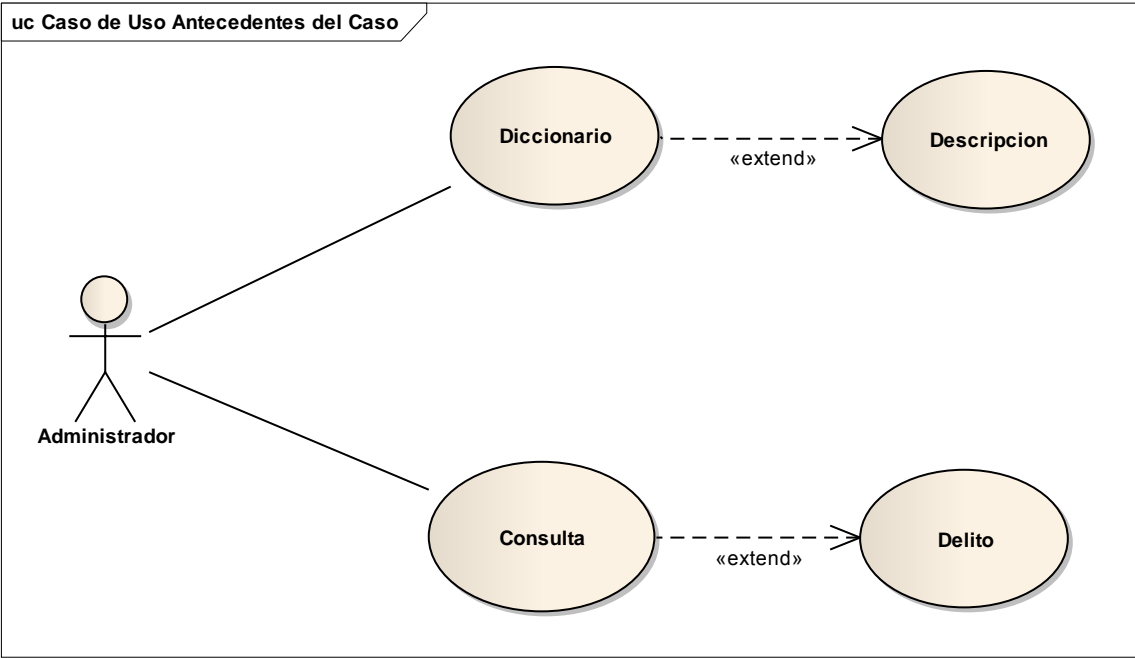


Figura 24.

II.1.5.2.25. CASO DE USO ANTECEDENTES DICCIONARIO

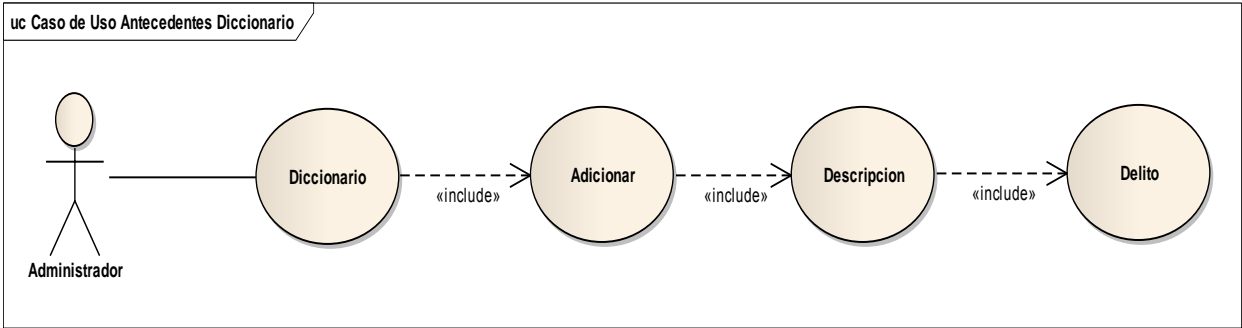


Figura25.

II.1.5.2.26. CASO DE USO ANTECEDENTES CONSULTA

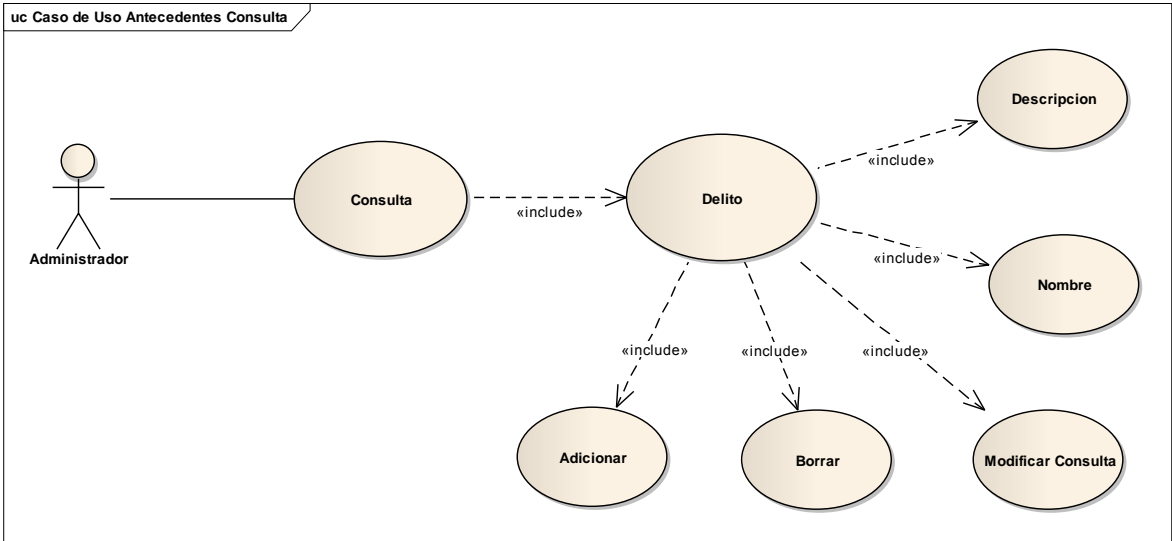


Figura 26.

II.1.5.2.27. CASO DE USO ABOGADO

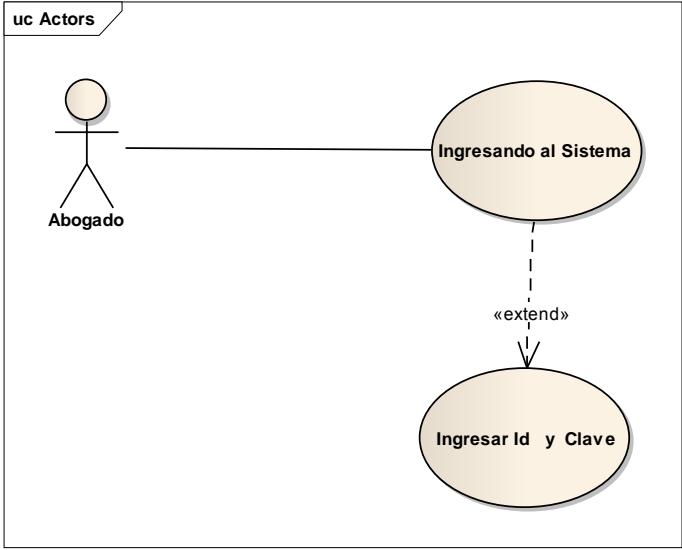


Figura 27.

II.1.5.2.28. CASO DE USO DESCRIPCION DEL CASO LEGAL

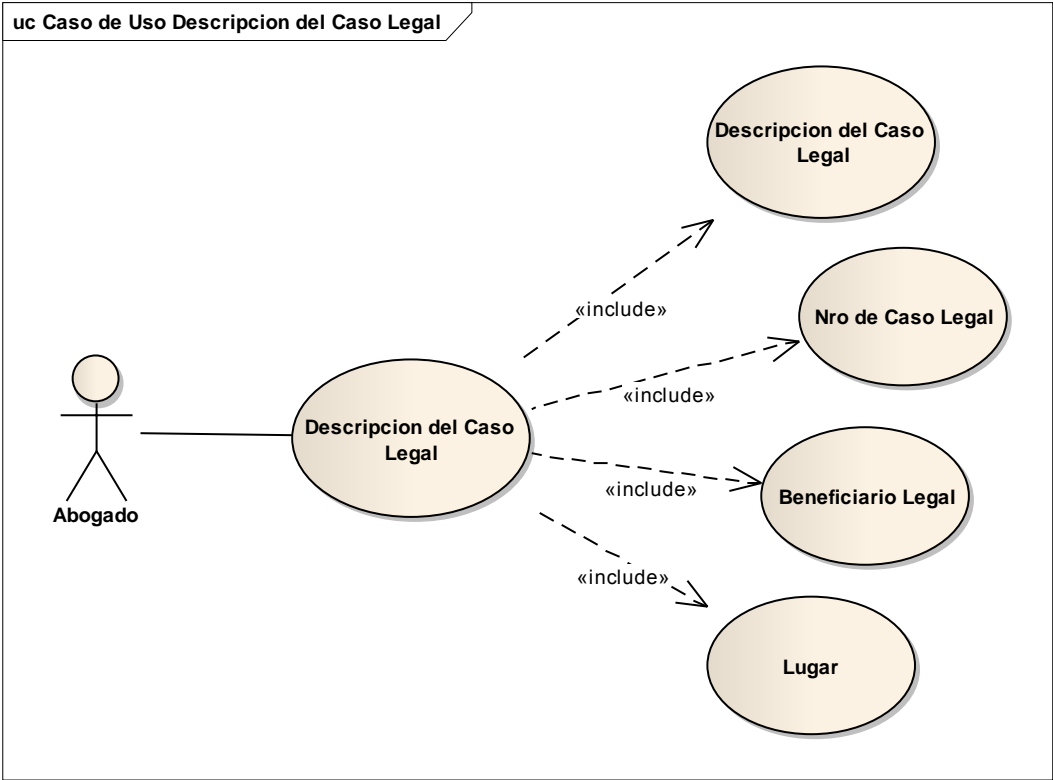


Figura 28.

II.1.5.2.29. CASO DE USO DESCRIPCION DEL CASO LEGAL MODIFICAR
ADICIONAR

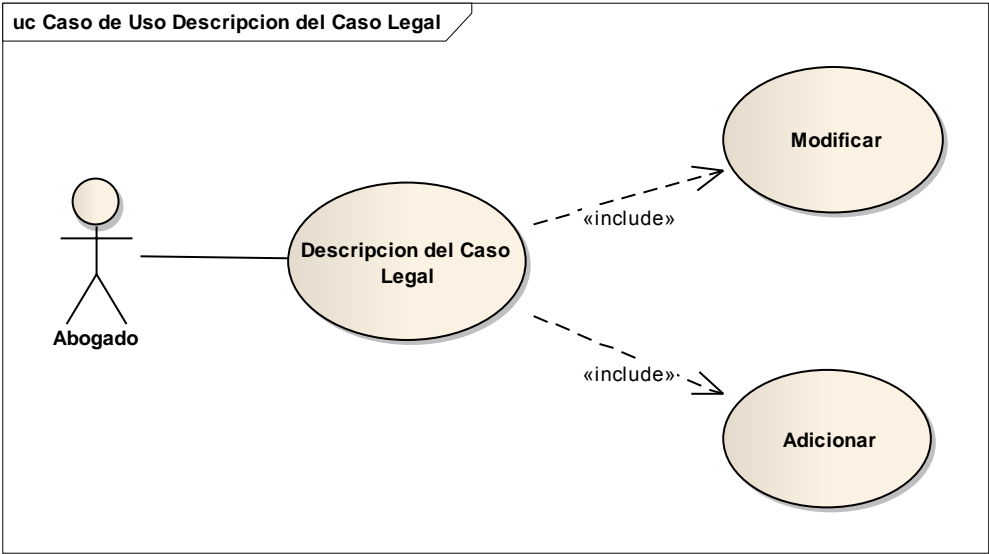


Figura 29.

II.1.5.2.30. CASO DE USO BENEFICIARIO LEGAL

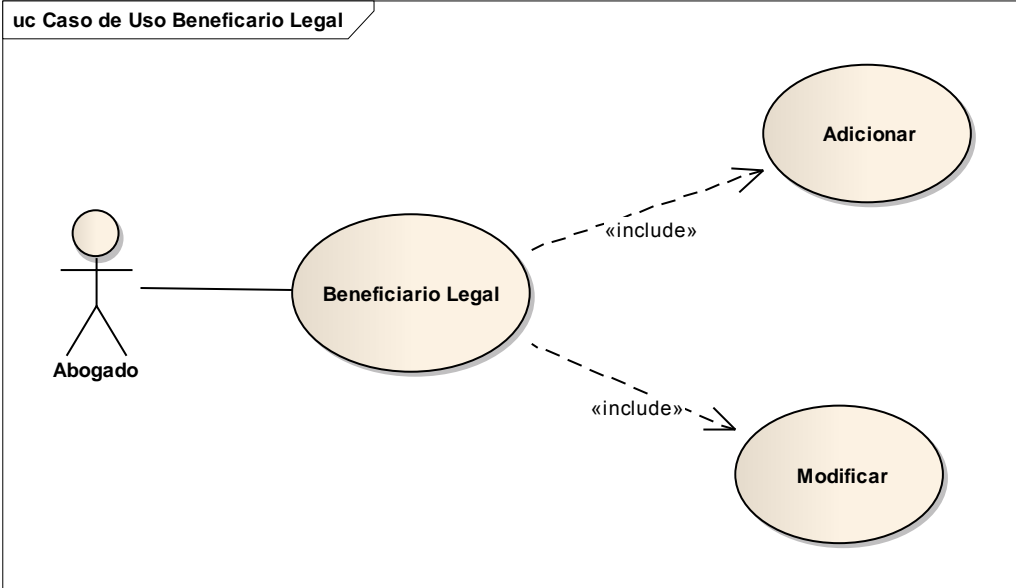


Figura 30.

II.1.5.2.31 CASO DE USO ANTECEDENTES DE CASO LEGAL BUSCAR

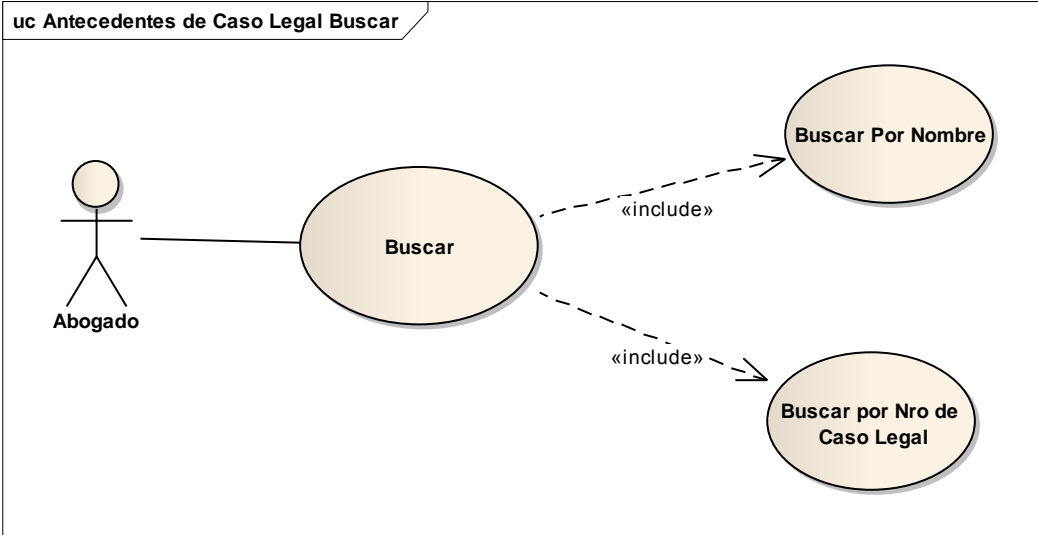


Figura 31.

II.1.5.2.32. CASO DE USO LUGAR

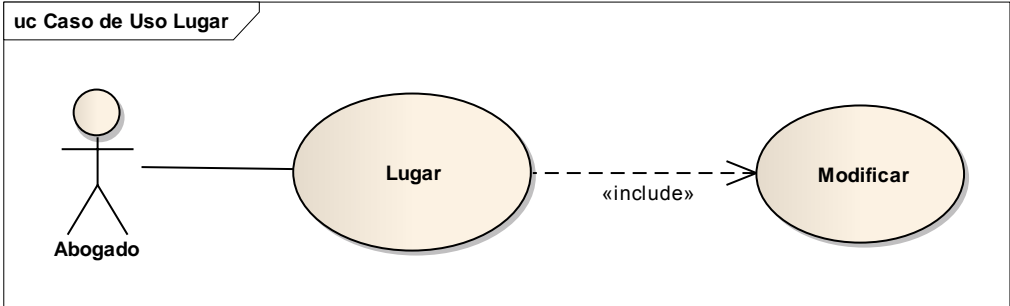


Figura 32.

II.1.5.2.33. CASO DE USO DATOS PERSONALES DE BENEFICIARIO

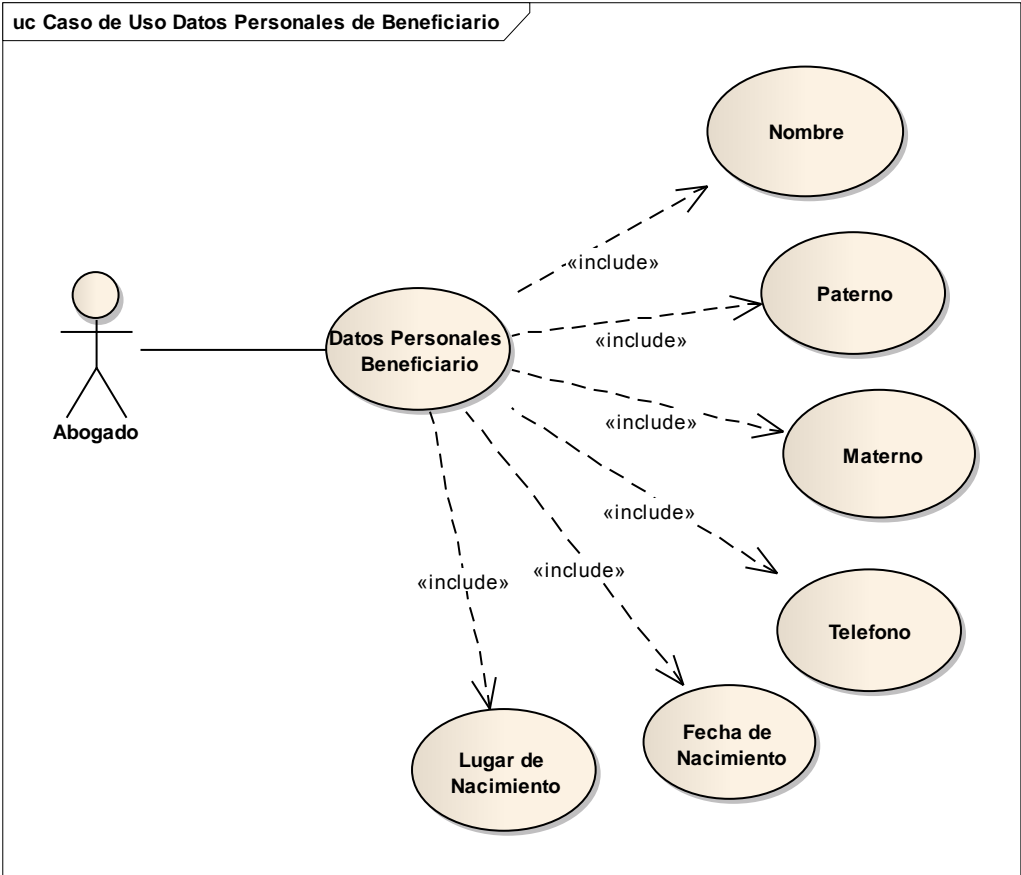


Figura 33.

II.1.5.2.34.- CASO DE USO DIRECCION DEL BENEFICIARIO

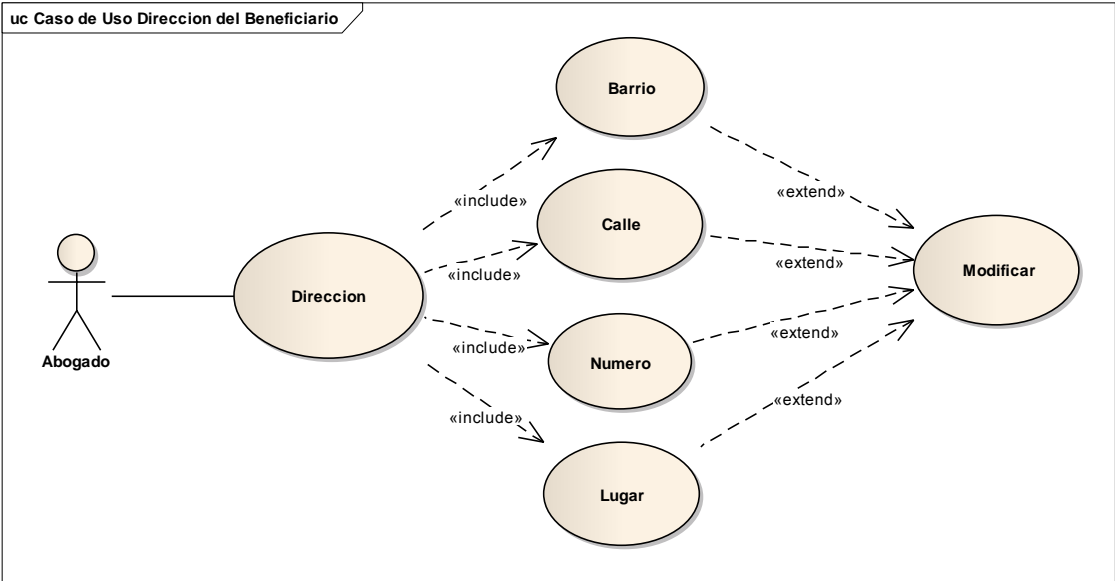


Figura 34.

II.1.5.2.35.- CASO DE USO CODIGO DE USUARIO

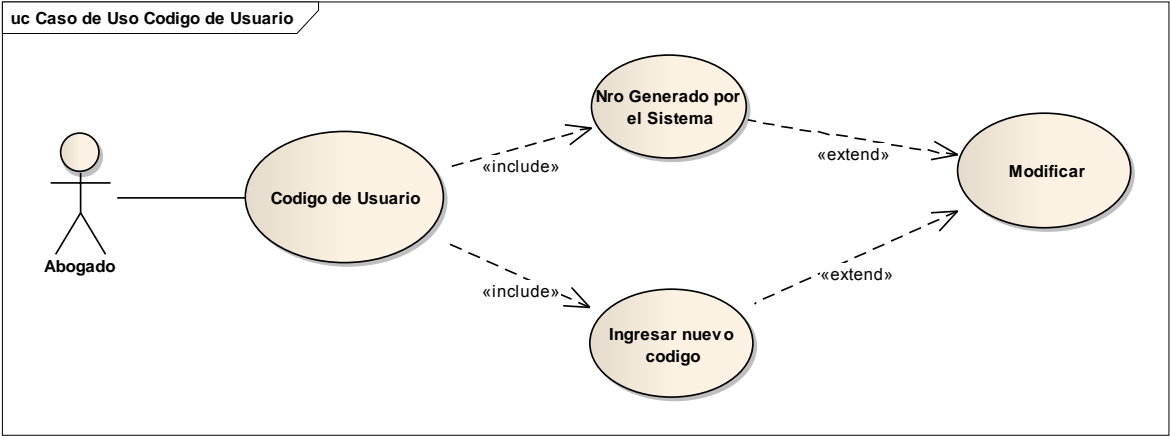


Figura 35.

II.1.5.2.36. CASO DE USO GENERAR REPORTE DE NOTICIAS

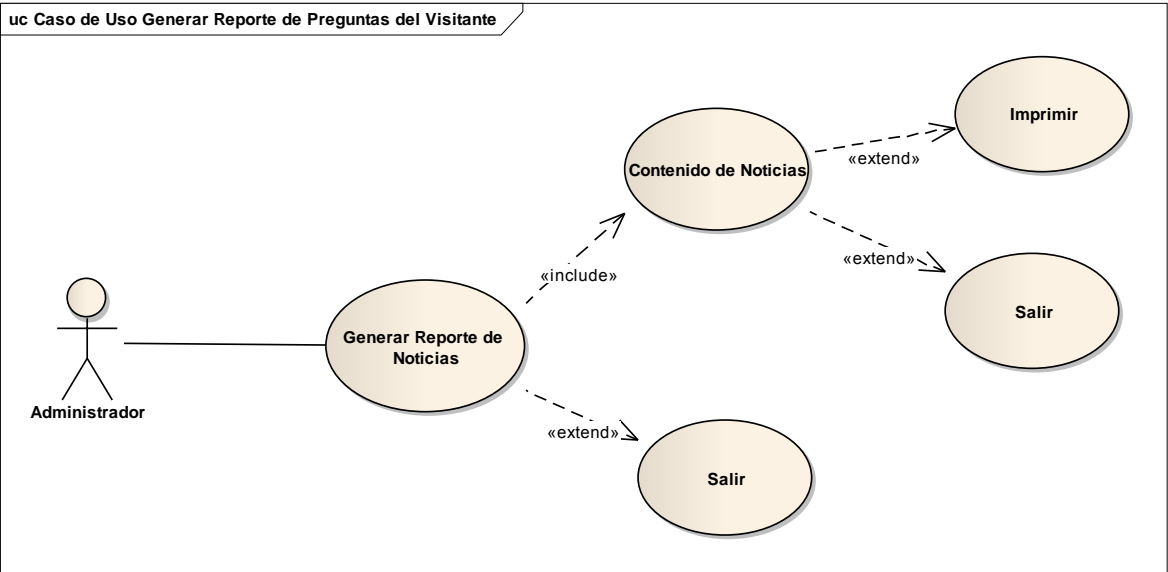


Figura 36.

II.1.5.2.37. CASO DE USO GENERAR REPORTE DE PREGUNTAS DEL VISITANTE

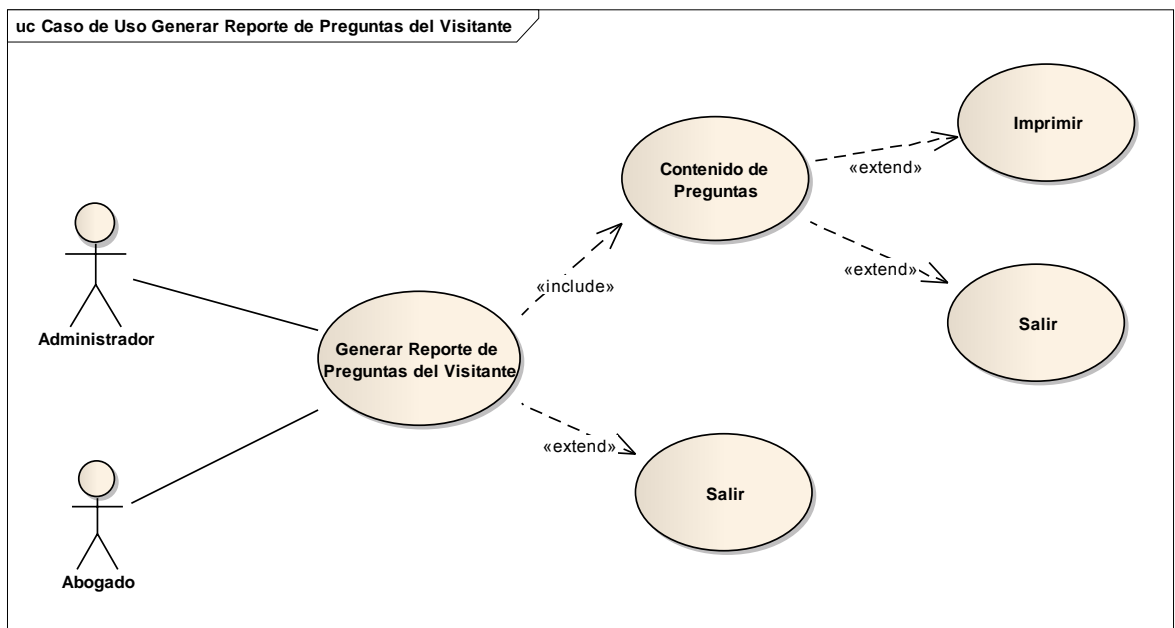


Figura 37.

II.1.5.3. Descripción del Casos de Uso

II.1.5.3.1. Descripción del Caso de Uso – Ingresando Al Sistema (Figura 1)

Caso de Uso Ingresando Al Sistema	
Descripción	<p>El Visitante ingresa al Sistema sin un Id y Clave por que el ingreso es libre, una vez que haya ingresado</p> <p>El Sistema lo tomara como un nuevo de tal manera que cada vez que alguien ingrese al sistema un identificador incrementara para almacenar en el sistema.</p>
Precondiciones	No hay precondiciones.
Poscondiciones	El visitante está identificado con un identificador.
Flujos de Evento	<p>a) El Sistema visualizara las categorías del menú principal según el rol, entonces podrá continuar con la visualización de todo el sistema.</p> <p>b) El usuario puede acceder directamente a salir en la barra superior.</p>
Excepciones	No Contiene.

II.1.5.3.2. Descripción del Caso de Uso – Abogado (Figura 2)

Caso de Uso Abogado	
Descripción	El Visitante una vez que haya visualizado la pantalla principal puede ingresar a lo que es la pantalla abogado, donde encontrara los abogados de la ONG, Curriculum de Abogados, Ver por Area de Especialización, Abogados de Convenio.
Precondiciones	El Visitante debe ingresar a la parte de Abogado para poder ingresar a los datos de la pantalla abogado.
Poscondiciones	No Hay poscondiciones.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de la pantalla Abogado según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El usuario puede acceder directamente a salir en la barra superior. c) El visitante puede acceder directamente a volver en la barra superior.
Excepciones	No contiene.

II.1.5.3.3. Descripción del Caso de Uso – ver Currículo de Abogado (Figura 3)

Caso de Uso ver Currículo de Abogado	
Descripción	El Visitante ingresa a ver currículum donde él puede ver datos correspondiente a un abogado de la ONG, la categoría de profesional que es.
Precondiciones	Ingresa previamente a abogado
Poscondiciones	El visitante está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de ver currículum de abogado según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Visitante puede acceder directamente a salir en la barra superior. c) El Visitante puede acceder directamente a volver se desplegara una pestaña para identificar a donde quiere volver exactamente.
Excepciones	No Contiene.

II.1.5.3.4. Descripción del Caso de Uso – Buscar Abogado (Figura 4)

Caso de Uso Buscar Abogado	
Descripción	El Visitante ingresa a Buscar Abogado, donde él puede buscar al abogado por su nombre, apellido y especialidad, como también algunos datos introducidos no encontrados en la búsqueda realizada.
Precondiciones	Haber ingresado a Abogado.
Poscondiciones	El visitante está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías del menú principal según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Visitante introducirá un dato para poder realizar la búsqueda que desea. c) Se mostrara algún dato después de la búsqueda realizada. d) El Visitante puede acceder directamente a salir en la barra superior. e) El Visitante puede acceder directamente a volver se desplegara una pestana para identificar a donde quiere volver

	exactamente.
Excepciones	La búsqueda se realiza solo con palabras dejando a un lado los números.

II.1.5.3.5. Descripción del Caso de Uso – Actividades de la ONG (Figura 5)

Caso de Uso Actividades de la ONG	
Descripción	El Visitante una vez que haya visualizado la pantalla principal puede ingresar a lo que es la pantalla Actividades, donde encontrara las actividades realizadas por la ONG, con sus respectivos características, área, lugar, administrador que impulso la actividad y mostrar actividad donde desenvolverá algunas datos relevantes de la actividad.
Precondiciones	El Visitante debe ingresar a la parte de pantalla principal.
Poscondiciones	El visitante está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de las Actividades según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Visitante podrá ingresar a la parte de mostrar actividad donde tiene sus respectivos datos.

	<p>c) El Visitante puede acceder directamente a salir en la barra superior.</p> <p>d) El Visitante puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	El sistema por defecto mostrar la Ultima actividad realizada.

II.1.5.3.6. Descripción del Caso de Uso – Mostrar Actividad de la ONG (Figura 6)

Caso de Uso Mostrar Actividades de la ONG	
Descripción	El Visitante una vez que haya ingresado a la pantalla Actividades, donde encontrara mostrar las actividades realizadas por la ONG, con sus respectivas características, lugar, administrador y mostrar actividad donde desenvolverá algunos datos relevantes de la actividad.
Precondiciones	El Visitante debe ingresar a la parte de pantalla principal.
Poscondiciones	El visitante está identificado con un identificador.
Flujos de Evento	<p>a) El Sistema visualizara las categorías de las Actividades</p>

	<p>según el rol, entonces podrá continuar con la visualización de todo el sistema.</p> <p>b) El Visitante podrá ingresar a la parte de mostrar actividad donde tiene sus respectivos datos.</p> <p>c) El Visitante puede acceder directamente a salir en la barra superior.</p> <p>d) El Visitante puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	El sistema por defecto mostrar la Ultima actividad realizada.

II.1.5.3.7. Descripción del Caso de Uso – Area de Trabajo de la ONG (Figura 7)

Caso de Uso Area de Trabajo de la ONG	
Descripción	El Visitante una vez que haya visualizado la pantalla principal puede ingresar a lo que es la pantalla Area de Trabajo de la ONG, donde encontrara las diferentes Area de trabajo la ONG, con sus respectivas características, Area existentes, Opinión respectiva de la Area, Solicitar información del Área de Trabajo.
Precondiciones	El Visitante debe ingresar a la parte de pantalla principal.
Poscondiciones	El visitante está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de las Area de la ONG según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Visitante podrá visualizar las diferentes area existentes en esta parte del sistema c) El Visitante podrá opinar con un determinado número de caracteres. d) El Visitante podrá solicitar la

	<p>información correspondiente de cada Area.</p> <p>e) El Visitante puede acceder directamente a salir en la barra superior.</p> <p>f) El visitante puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	<p>El sistema por defecto mostrara alguna area de trabajo.</p> <p>El sistema solo visualizara la parte de opinión y información si la solicitud está disponible.</p>

II.1.5.3.8. Descripción del Caso de Uso – Noticia (Figura 8)

Caso de Uso Noticia	
Descripción	El Visitante una vez que haya visualizado la pantalla principal puede ingresar a lo que es la pantalla de Noticia, donde encontrara las diferentes Noticias relevantes que propone la ONG.
Precondiciones	El Visitante debe ingresar a la parte de pantalla principal.
Poscondiciones	El visitante está identificado con un identificador. Al brindar una opinión este mensaje estará almacenado en la Base de Datos.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las Noticias mas relevantes que ocurran en nuestro medio pero se basara más que todo a lo que es las noticias con respecto a la parte Legal según el rol que tenga el que ingrese lo podrá ver, a su vez podrá continuar con la visualización de todo el sistema. b) El Visitante podrá opinar con un determinado número de caracteres. c) El Visitante podrá solicitar la información correspondiente de

	<p>cada Noticia.</p> <p>d) El Visitante puede acceder directamente a salir en la barra superior.</p> <p>e) El Visitante puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	<p>El sistema por defecto mostrara alguna area de trabajo.</p> <p>El sistema solo visualizara la parte de opinión y información si la solicitud está disponible.</p>

II.1.5.3.9. Descripción del Caso de Uso – Antecedentes del Caso (Figura 10)

Caso de Uso Antecedentes del Caso	
Descripción	El Visitante una vez que haya ingresado al sistema, encontrara Antecedentes del Caso, con sus respectivas características, Diccionario, Consulta, Cancelar.
Precondiciones	El Visitante debe ingresar a la parte de pantalla principal.
Poscondiciones	El Visitante está identificado con un identificador. Si el Visitante guarda los datos, el sistema le otorgara un Id para apersonarse a la ONG y de tal manera podrá convertirse en Beneficiario.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Antecedentes del Caso según su rol, entonces podrá continuar con la visualización de todo el sistema. b) El Visitante podrá ingresar a la parte de Consulta con un respectivo dato introducido. c) El Visitante podrá ingresar a la parte de Cancelar antecedentes de tal manera se le presentara la parte de guardar los datos introducidos después se le mostrara que tiene

	<p>que apersonarse a la ONG para poder tener un asesoramiento legal.</p> <p>d) El sistema guardara automáticamente la parte de los datos de búsqueda para poder saber que delitos son los más buscados para su respectivo asesoramiento</p> <p>e) Se le otorgara un Id correspondiente para poder presentarse a la ONG para una más rápida atención.</p> <p>f) El Visitante puede acceder directamente a salir en la barra superior.</p> <p>g) El Visitante puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	El sistema por defecto mostrara un espacio en blanco para realizar la consulta deseada.

II.1.5.3.10. Descripción del Caso de Uso – Ingresando al Sistema Administrador
(Figura 10)

Caso de Uso Ingresando Al Sistema Administrador	
Descripción	<p>El Administrador ingresa al Sistema con un Id y Clave para que se le de la atribución de su rol o simplemente ser un visitante, una vez que haya ingresado.</p> <p>Al ingreso del Sistema, cada modificación que realice el sistema guardara su Id para identificar qué tipo de alteración realizo en el Sistema.</p>
Precondiciones	Ingresar con Id y Clave
Poscondiciones	El Administrador está identificado con su propio identificador.
Flujos de Evento	<p>a) El Sistema visualizara las categorías del menú principal según el rol, entonces podrá continuar con la visualización de todo el sistema.</p> <p>b) El Administrador tendrá atribuciones distintas por el rol.</p> <p>c) El Administrador puede acceder directamente a salir en la barra superior.</p>
Excepciones	El Administrador también puede ser a su vez un visitante normal sin ingresar su Id y Clave.

II.1.5.3.11. Descripción del Caso de Uso – Administrador Abogado (Figura 11)

Caso de Uso Abogado	
Descripción	El Administrador una vez que haya visualizado la pantalla principal puede ingresar a lo que es la pantalla abogado, donde encontrara los abogados de la ONG, Currículo de Abogados, Ver por Area de Especialización, Abogados de Convenio, en el cual puede realizar alteraciones por rol que cumple.
Precondiciones	El Administrador debe ingresar a la parte de Abogado para poder ingresar a los datos de la pantalla abogado.
Poscondiciones	El Administrador está identificado con su propio identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de la pantalla Abogado según el rol que cumple, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador tendrá atribuciones distintas del Visitante como así del Abogado. c) El Administrador puede acceder directamente a salir en la barra superior. d) El Administrador puede acceder directamente a volver en la barra

	superior.
Excepciones	El Administrador también puede ser a su vez un visitante normal sin ingresar su Id y Clave.

II.1.5.3.12. Descripción del Caso de Uso – Ver Curriculum del Abogado (Figura 12)

Caso de Uso ver Currículo de Abogado	
Descripción	El Administrador ingresa a ver currículum donde él puede ver datos correspondiente del currículum de los abogados de la ONG, en si la categoría de profesional que es, el tendrá atribuciones distintas por el rol que cumple.
Precondiciones	Ingresar previamente a Ver Abogado
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de ver currículum de abogado según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador tendrá las atribuciones de modificar y Adicionar. c) Además tendrá que introducir su Id y Clave para realizar cualquier modificación que realice.

	<p>d) El Sistema guarda todo lo modificado, además el Id de quien modifíco el dato para cualquier problema que exista.</p> <p>e) Podrá salir Directamente.</p> <p>f) El Administrador puede acceder directamente a salir en la barra superior.</p> <p>g) El Administrador puede acceder directamente a volver se desplegara una pestana para identificar a donde quiere volver exactamente.</p>
Excepciones	Para cualquier dato que quisiera modificar el Administrador tendrá que introducir su Id y Clave.

II.1.5.3.13. Descripción del Caso de Uso – Buscar Abogado (Figura 13)

Caso de Uso Buscar Abogado	
Descripción	El Administrador ingresa a Buscar Abogado, donde él puede buscar al abogado por su nombre, apellido y especialidad, como también algunos datos introducidos no encontrados en la búsqueda realizada.
Precondiciones	Haber ingresado a Abogado.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías del menú principal según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador introducirá un dato para poder realizar la búsqueda que desea. c) Se mostrara algún dato después de la búsqueda realizada. d) El Administrador puede acceder directamente a salir en la barra superior. e) El Administrador puede acceder directamente a volver se desplegara una pestana para identificar a donde quiere volver

	exactamente.
Excepciones	La búsqueda se realiza solo con palabras dejando a un lado los números.

II.1.5.3.14. Descripción del Caso de Uso – Actividad de la ONG (Figura 14)

Caso de Uso Actividades de la ONG	
Descripción	El Administrador una vez que haya visualizado la pantalla principal puede ingresar a lo que es la pantalla Actividades, donde encontrara las actividades realizadas por la ONG, con sus respectivos características, área, lugar y mostrar actividad donde desenvolverá algunas datos relevantes de la actividad, por el rol que cumple tendrá la posibilidad de borrar la actividad y adicionar una actividad.
Precondiciones	El Administrador debe ingresar a la parte de pantalla principal.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	a) El Sistema visualizara las categorías de las Actividades según el rol, entonces podrá continuar con la visualización de todo el sistema.

	<ul style="list-style-type: none"> b) El Administrador podrá ingresar a la parte de mostrar actividad donde tiene sus respectivos datos. c) El Administrador tendrá la posibilidad de borrar o adicionar una nueva actividad para la realización de la misma. d) Al adicionar una nueva actividad el sistema le pedirá introducir su Id y Clave para su validación del mismo. e) El Administrador puede acceder directamente a salir en la barra superior. f) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	<p>El sistema por defecto mostrar la Ultima actividad realizada.</p> <p>Id se autentificara y se guardara en el sistema de tal manera se sabrá quién fue el que impulso la actividad.</p>

II.1.5.3.15. Descripción del Caso de Uso Mostrar Actividad (Figura 15)

Caso de Uso Mostrar Actividades de la ONG	
Descripción	El Administrador una vez que haya ingresado a la pantalla Actividades, donde encontrara mostrar las actividades realizadas por la ONG, con sus respectivas características, por su rol tendrá el privilegio de modificar cualquier detalle de la actividad.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Actividades.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de las Actividades según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador podrá ingresar a la parte de mostrar actividad donde tiene sus respectivos datos. c) El Administrador tendrá privilegios por el rol que cumple, modificar cualquier actividad. d) La difusión del contenido de la actividad podrá modificarla pero

	<p>su Id se quedara guardado internamente por el sistema.</p> <p>e) El Administrador puede acceder directamente a salir en la barra superior.</p> <p>f) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	<p>El sistema por defecto mostrar la Ultima actividad realizada.</p> <p>El sistema guardara el Id al modificar cualquier dato de la actividad.</p>

II.1.5.3.16. Descripción del Caso de Uso Adicionar nueva Actividad (Figura 16)

Caso de Uso Adicionar nueva Actividad	
Descripción	El Administrador una vez que haya ingresado a la pantalla Actividades, donde encontrara mostrar las actividades realizadas por la ONG, con sus respectivas características, por su rol tendrá el privilegio de Adicionar nueva actividad.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Actividades.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de las Actividades según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador podrá ingresar a la parte de Adicionar nueva actividad donde tiene sus respectivos datos. c) El Administrador tendrá privilegios por el rol que cumple, Adicionar nueva Actividad. d) El Sistema guardara el Id para

	<p>tener una referencia de quien guardo una nueva actividad de la ONG.</p> <p>e) El Administrador puede acceder directamente a salir en la barra superior.</p> <p>f) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	<p>El sistema por defecto mostrar la Ultima actividad realizada.</p> <p>El sistema guardara el Id al modificar cualquier dato de la actividad.</p>

II.1.5.3.17. Descripción del Caso de Uso – Area de Trabajo de la ONG. (Figura 17)

Caso de Uso Area de Trabajo de la ONG	
Descripción	El Administrador una vez que haya visualizado la pantalla principal puede ingresar a lo que es la pantalla Area, donde encontrara las Area donde trabaja la ONG, con sus respectivas características según al rol que cumple.
Precondiciones	El Administrador debe ingresar a la parte de pantalla principal.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de las Actividades según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador podrá ingresar a la parte Areas de Trabajo de la ONG donde tiene sus respectivos datos por el rol que cumple. c) El Administrador tendrá la posibilidad de adicionar, modificar y borrar una nueva Area donde la ONG desarrolla su trabajo. d) El Administrador puede acceder directamente a salir en la barra

	<p>superior.</p> <p>e) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	El sistema por defecto mostrara alguna area de trabajo de la ONG.

II.1.5.3.18. Descripción del Caso de Adicionar Area (Figura 18)

Caso de Uso Adicionar Area	
Descripción	El Administrador una vez que haya ingresado a la pantalla Area, donde encontrara Adicionar Area, con sus respectivas características, por su rol tendrá ese privilegio.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Area de la ONG.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<p>a) El Sistema visualizara las categorías de Adicionar Area por el rol que cumple, entonces podrá continuar con la visualización de todo el sistema.</p> <p>b) El Administrador podrá ingresar una nueva Area con el Nombre y</p>

	<p>su descripción.</p> <p>c) El Sistema guardara el Id para tener una referencia de quien guardo una nueva Area.</p> <p>d) El Administrador puede acceder directamente a salir en la barra superior.</p> <p>e) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	<p>El sistema por defecto mostrara alguna Area más Visitada.</p> <p>El sistema guardara el Id al Adicionar cualquier dato.</p>

II.1.5.3.19. Descripción del Caso de Borrar Area (Figura 19)

Caso de Uso Borrar Area	
Descripción	El Administrador una vez que haya ingresado a la pantalla Area, donde encontrara Borrar Area, con sus respectivas características, por su rol tendrá ese privilegio.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Areas de la ONG.
Poscondiciones	El Administrador está identificado con un identificador.

	Para borrar la Area tendrá que introducir su Id y Clave.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de Borrar Area por el rol que cumple, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador pora poder borrar una Area tendrá que introducir su Id y Clave para tener un respaldo para el sistema. c) El Administrador puede acceder directamente a salir en la barra superior. d) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	<p>El sistema por defecto mostrara alguna Area más Visitada.</p> <p>El sistema guardara el Id al borrar cualquier dato.</p>

II.1.5.3.20. Descripción del Caso de Uso – Modificar Area (Figura 20)

Caso de Uso Modificar Area	
Descripción	El Administrador una vez que haya ingresado a la pantalla Area, donde encontrara Modificar Area, con sus respectivas características, por su rol tendrá ese privilegio.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Areas de la ONG.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de Modificar Area por el rol que cumple, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador al modificar algún dato el sistema guardara su Id como seguridad. c) El Administrador puede acceder directamente a salir en la barra superior. d) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.

Excepciones	<p>El sistema por defecto mostrara alguna Area más Visitada.</p> <p>El sistema guardara el Id al modificar cualquier dato.</p>
-------------	--

II.1.5.3.21. Descripción del Caso de Uso – Noticia (Figura 21)

Caso de Uso Noticia	
Descripción	El Administrador una vez que haya visualizado la pantalla principal puede ingresar a lo que es la pantalla Noticia, donde encontrara las Noticias más relevantes publicadas por la ONG, con sus respectivas características según al rol que cumple.
Precondiciones	El Administrador debe ingresar a la parte de pantalla principal.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de las Actividades según el rol, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador podrá ingresar a ver el contenido de la Noticia donde el tendrá privilegios de borrar y modificar c) El Administrador tendrá el

	<p>privilegio de Adicionar Noticia.</p> <p>d) El Administrador puede acceder directamente a salir en la barra superior.</p> <p>e) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	<p>El sistema por defecto mostrara alguna area de trabajo de la ONG.</p> <p>Como es de alta prioridad la visualización de la Noticia se solicitara el Id y Clave para cualquier modificación, adicionar o borrar.</p>

II.1.5.3.22. Descripción del Caso de Uso – Noticia (Figura 22)

Caso de Uso Noticia	
Descripción	El Administrador una vez que haya ingresado a la pantalla Noticia, donde encontrara Adicionar Noticia, con sus respectivas características, por su rol tendrá ese privilegio.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Noticia de la ONG.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	a) El Sistema visualizara las

	<p>categorias de Adicionar Noticia por el rol que cumple, entonces podrá continuar con la visualización de todo el sistema.</p> <p>b) El Administrador podrá ingresar una nueva Noticia pero se solicitara su Id y Clave para identificación por tratarse de ser de alta prioridad, de tal manera podrá ingresar el titulo, contenido y Area.</p> <p>c) El Sistema guardara el Id para tener una referencia de quien guardo una nueva Noticia.</p> <p>d) El Administrador puede acceder directamente a salir en la barra superior.</p> <p>e) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	<p>El sistema por defecto mostrara alguna Noticia relevante más Visitada.</p> <p>El sistema guardara el Id al Adicionar cualquier dato.</p>

II.1.5.3.23. Descripción del Caso de Uso – Modificar Noticia (Figura 23)

Caso de Uso Modificar Noticia	
Descripción	El Administrador una vez que haya ingresado a la pantalla Noticia, donde encontrara Modificar Noticia, con sus respectivas características, por su rol tendrá ese privilegio.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Modificar Noticia.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de Modificar Noticia por el rol que cumple, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador podrá ingresar Modificar Noticia, tendrá que ingresar su Id y Clave después de su autenticación podrá modificar el titulo, contenido y area. c) El Sistema guardara el Id para tener una referencia de quien modifiko la Noticia. d) El Administrador puede acceder directamente a salir en la barra superior. e) El Administrador puede acceder

	directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	El sistema por defecto mostrara alguna Noticia más Visitada. El sistema guardara el Id al Modificar cualquier dato.

II.1.5.3.24. Descripción del Caso de Uso – Antecedentes del Caso (Figura 24)

Caso de Uso Antecedentes del Caso	
Descripción	El Administrador una vez que haya ingresado al sistema, encontrara Antecedentes del Caso, con sus respectivas características, Diccionario y Consulta.
Precondiciones	El Administrador debe ingresar a la parte de pantalla principal.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Antecedentes del Caso según su rol, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador podrá ingresar a la parte de Consulta con un respectivo dato introducido.

	<ul style="list-style-type: none"> c) El Administrador podrá ingresar a la parte de Diccionario. d) El Administrador podrá ingresar a la parte de Cancelar antecedentes e) El sistema guardara automáticamente la parte de los datos de búsqueda para poder saber que delitos son los más buscados para su respectivo asesoramiento f) El Administrador puede acceder directamente a salir en la barra superior. g) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	El sistema por defecto mostrara un espacio en blanco para realizar la consulta deseada.

II.1.5.3.25. Descripción del Caso de Uso – Antecedentes Diccionario (Figura 25)

Caso de Uso Antecedentes Diccionario	
Descripción	El Administrador una vez que haya ingresado a la pantalla Antecedentes de Caso, donde encontrara Diccionario y analizar si las palabras coinciden.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Antecedentes de Caso
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de Diccionario por el rol que cumple, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador accederá a la parte de diccionario y podrá adicionar una nueva palabra dentro del diccionario Con su respectiva descripción y al delito que corresponde. c) El Administrador puede acceder directamente a salir en la barra superior. d) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede

	elegir a donde ir.
Excepciones	El sistema mostrara algún contenido visitado.

II.1.5.3.26. Descripción del Caso de Uso – Antecedentes Consulta (Figura 26)

Caso de Uso Antecedentes Consulta	
Descripción	El Administrador una vez que haya ingresado a la pantalla Antecedentes de Caso, donde encontrara Consulta con su respectivo delito.
Precondiciones	El Administrador debe ingresar a la parte de pantalla Antecedentes de Caso
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las categorías de Antecedentes consulta por el rol que cumple, entonces podrá continuar con la visualización de todo el sistema. b) El Administrador tendrá atribuciones como de modificar, adicionar, borrar. c) El Administrador tendrá atribución de tener acceso de nombre y su descripción de delito. d) El Administrador puede acceder directamente a salir en la barra superior.

	e) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	El Sistema mostrara un espacio en blanco para acceder una consulta.

I.5.3.27. Descripción del Caso de Uso – Ingresando Abogado (Figura 27)

Caso de Uso Ingresando Abogado	
Descripción	El Abogado ingresa al Sistema con un Id y Clave para la atribución de su rol, una vez que haya ingresado. Al ingreso del Sistema, cada modificación que realice el sistema guardara su Id para identificar qué tipo de alteración realizo en el Sistema.
Precondiciones	Ingresar con Id y Clave
Poscondiciones	El Abogado está identificado con su propio identificador.
Flujos de Evento	<p>a) El Sistema visualizara las categorías del menú principal según el rol, entonces podrá continuar con la visualización de todo el sistema.</p> <p>b) El Abogado tendrá atribuciones distintas por el rol que tiene.</p> <p>c) El Abogado puede acceder</p>

	directamente a salir en la barra superior.
Excepciones	El Abogado también puede ser a su vez un visitante normal sin ingresar su Id y Clave.

II.1.5.3.28 Descripción del Caso de Uso – Descripción del Caso Legal (Figura 28)

Caso de Uso Descripción del Caso Legal	
Descripción	El Administrador una vez que haya ingresado al sistema, encontrara Descripción del Caso Legal, con sus respectivas características, Diccionario y Consulta.
Precondiciones	El Administrador debe ingresar a la parte de pantalla principal.
Poscondiciones	El Administrador está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Antecedentes del Caso según su rol, entonces podrá continuar con la visualización de todo el sistema. b) El Abogado podrá ingresar a la parte de descripción del caso para ver el tipo caso legal tendrá para un beneficiario. c) El sistema le brindara el Nro. de

	<p>Caso legal, Beneficiario Legal, Lugar.</p> <p>d) El Administrador podrá ingresar a la parte de Cancelar antecedentes</p> <p>e) El sistema guardara automáticamente la parte de los datos de búsqueda para poder saber que delitos son los más buscados para su respectivo asesoramiento</p> <p>f) El Administrador puede acceder directamente a salir en la barra superior.</p> <p>g) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	<p>El sistema por defecto mostrara un espacio en blanco para realizar la consulta deseada.</p> <p>Si el Abogado no ingresa con un Id y Clave será un visitante del Sistema.</p>

I.1.5.3.29. Descripción del Caso de Uso – Adicionar Modificar Caso Legal (Figura 29)

Caso de Uso Adicionar Modificar Caso Legal	
Descripción	El Abogado una vez que haya ingresado al sistema, encontrara Descripción del Caso Legal, con sus respectivas características.
Precondiciones	El Abogado debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Adicionar Modificar Caso Legal según su rol, entonces podrá continuar con la visualización de todo el sistema. b) El Abogado podrá ingresar a la parte de descripción del caso para poder modificar Adicionar caso legal que tendrá para un beneficiario. c) El sistema guardar Id de cualquier modificación o adición realiza. d) El Administrador puede acceder directamente a salir en la barra superior. e) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una

	pestanda despegable donde puede elegir a donde ir.
Excepciones	El Abogado tendrá que ingresar su Id y Clave para poder realizar cualquier modificación, si no será un Visitante.

II.1.5.3.30. Descripción del Caso de Uso – Beneficiario Legal (Figura 30)

Caso de Uso Beneficiario Legal	
Descripción	El Abogado una vez que haya ingresado al sistema, podrá entrar en beneficiario legal donde en algún momento poder modificar y adicionar directamente a un beneficiario.
Precondiciones	El Abogado debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Beneficiario Legal según su rol, entonces podrá continuar con la visualización de todo el sistema. b) El Abogado podrá ingresar a la parte de Beneficiario legal donde tendrá la atribución de Adicionar y Modificar. c) El sistema guardar Id de cualquier modificación o adición realiza.

	<p>d) El Abogado puede acceder directamente a salir en la barra superior.</p> <p>e) El Abogado puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	El Abogado tendrá que ingresar su Id y Clave para poder realizar cualquier modificación, si no será un Visitante.

II.1.5.3.31. Descripción del Caso de Uso – Antecedentes de Caso Legal Buscar
(Figura 31)

Caso de Uso Antecedentes de Caso Legal Buscar	
Descripción	El Abogado una vez que haya ingresado al sistema, podrá entrar en Buscar donde introduciendo un dato podrá buscar por el Numero de caso legal y por el nombre beneficiario.
Precondiciones	El Abogado debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<p>a) El Sistema visualizara las respectivas características de</p>

	<p>Antecedentes de Caso Legal</p> <p>Buscar según su rol, entonces podrá continuar con la visualización de todo el sistema.</p> <p>b) El Abogado podrá ingresar a la parte de Beneficiario legal donde tendrá la atribución de Buscar por el Nro. de Caso Legal, por el Nombre del Beneficiario.</p> <p>c) El sistema guardar Id de cualquier modificación o adición realiza.</p> <p>d) El Abogado puede acceder directamente a salir en la barra superior.</p> <p>e) El Abogado puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.</p>
Excepciones	El sistema mostrara un espacio en blanco para la introducción de un dato por el Abogado.

II.1.5.3.32. Descripción del Caso de Uso – Lugar (Figura 32)

Caso de Uso Lugar	
Descripción	El Abogado una vez que haya ingresado a Antecedentes de Caso, podrá entrar en Lugar donde podrá modificar el lugar de donde se va originar el caso legal.
Precondiciones	El Abogado debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de lugar para su modificación. b) El sistema guardar Id de cualquier modificación realiza. c) El Abogado puede acceder directamente a salir en la barra superior. d) El Abogado puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	Para cualquier modificación el sistema guardara el Id y Clave.

II.1.5.3.33. Descripción del Caso de Uso – Datos Personales del Beneficiario (Figura 33)

Caso de Uso Datos Personales Beneficiario	
Descripción	El Abogado una vez que haya ingresado al sistema, podrá entrar a Datos personales de Beneficiario.
Precondiciones	El Abogado debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de los Datos Personales de Beneficiario. b) El Abogado puede acceder directamente a salir en la barra superior. c) El Abogado puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	Para cualquier modificación el sistema guardara el Id y Clave.

II.1.5.3.34. Descripción del Caso de Uso – Direccion del Beneficiario (Figura 34)

Caso de Uso Direccion del Beneficiario	
Descripción	El Abogado una vez que haya ingresado al sistema, podrá entrar a Direccion de Beneficiario, por de ahí se determinara la dirección exacta del beneficiario para cualquier problema que suceda.
Precondiciones	El Abogado debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Direccion del Beneficiario de tal manera tener el privilegio de modificar algún dato dentro de Direccion. b) El Abogado puede acceder directamente a salir en la barra superior. c) El Abogado puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	Para cualquier modificación el sistema guardara el Id y Clave.

II.1.5.3.35. Descripción del Caso de Uso – Código de Usuario (Figura 35)

Caso de Uso Código de Usuario	
Descripción	El Abogado una vez que haya ingresado al sistema, tendrá privilegio de cambiar el código de usuario que le brinda el sistema al Visitante cuando desea convertirse en Beneficiario.
Precondiciones	El Abogado debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Código de Usuario de tal manera tener el privilegio de modificar algún o ingresar uno nuevo. b) El Abogado puede acceder directamente a salir en la barra superior. c) El Abogado puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	Para cualquier modificación el sistema guardara el Id y Clave.

II.1.5.3.36. Descripción del Caso de Uso – Generar Reporte de Noticias (Figura 36)

Caso de Uso Generar Reporte de Noticias	
Descripción	El Administrador una vez ingresado al sistema mediante su Id y Clave, tendrá el privilegio de generar reporte de noticias
Precondiciones	El Administrador debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Generar Reporte de Noticias, con la posibilidad imprimir o salir. b) El Administrador puede acceder directamente a salir en la barra superior. c) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	Para la impresión estará definido por el sistema.

II.1.5.3.37. Descripción del Caso de Uso – Generar Reporte de Preguntas del Visitante (Figura 37)

Caso de Uso Generar Reporte de Preguntas del Visitante	
Descripción	El Administrador una vez ingresado al sistema mediante su Id y Clave, tendrá el privilegio de generar reporte de Preguntas del Visitante.
Precondiciones	El Administrador debe ingresar a la parte de pantalla principal.
Poscondiciones	El Abogado está identificado con un identificador.
Flujos de Evento	<ul style="list-style-type: none"> a) El Sistema visualizara las respectivas características de Generar Reporte de Preguntas del Visitante. b) El Sistema mostrara el contenido de tal manera poder imprimir o salir. c) El Administrador puede acceder directamente a salir en la barra superior. d) El Administrador puede acceder directamente a volver en la barra superior donde se encontrara una pestana despegable donde puede elegir a donde ir.
Excepciones	Para la impresión estará definido por el sistema.

II.1.5.4. Diagramas de Secuencia

II.1.5.4.1. Diagrama de Secuencia Ingresar Sistema Visitante

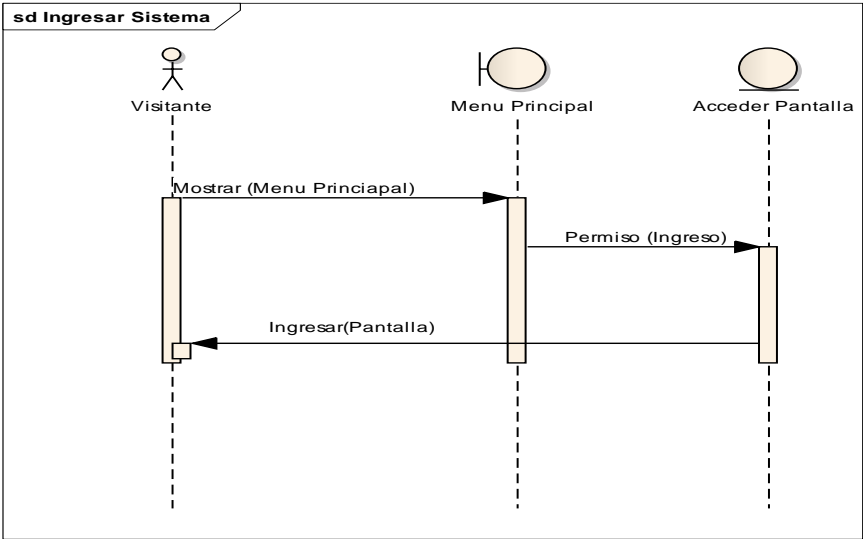


Figura 1.

II.1.5.4.2. Diagrama de Secuencia Abogados

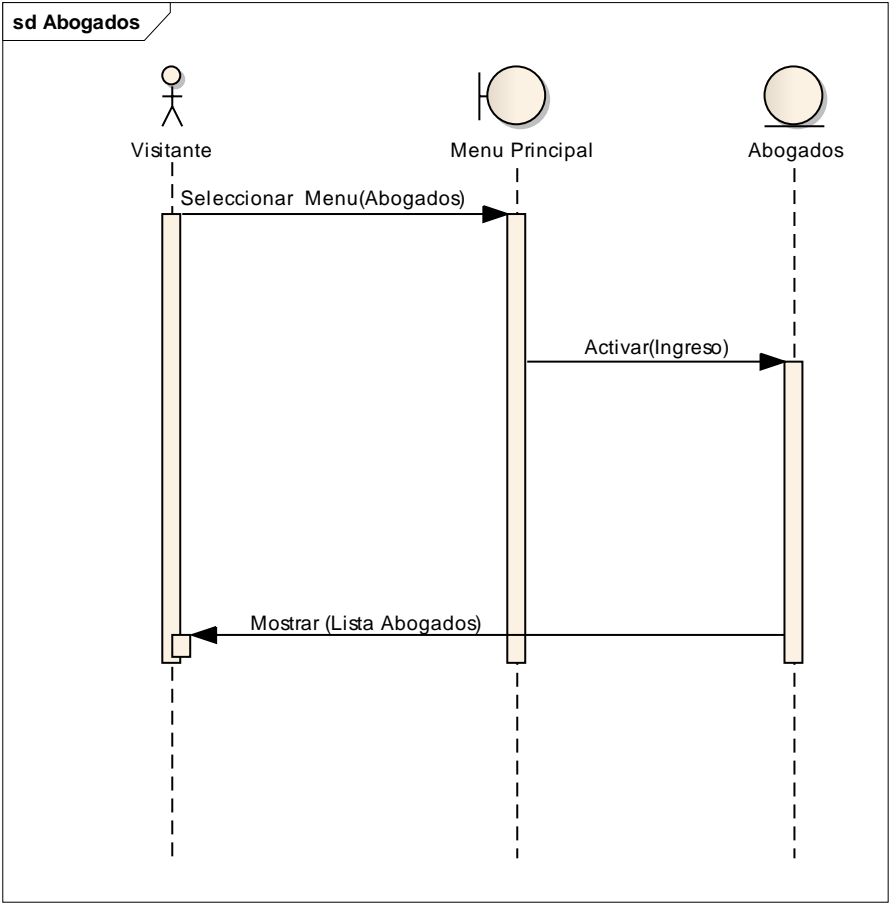


Figura 2.

II.1.5.4.3. Curriculum Ver

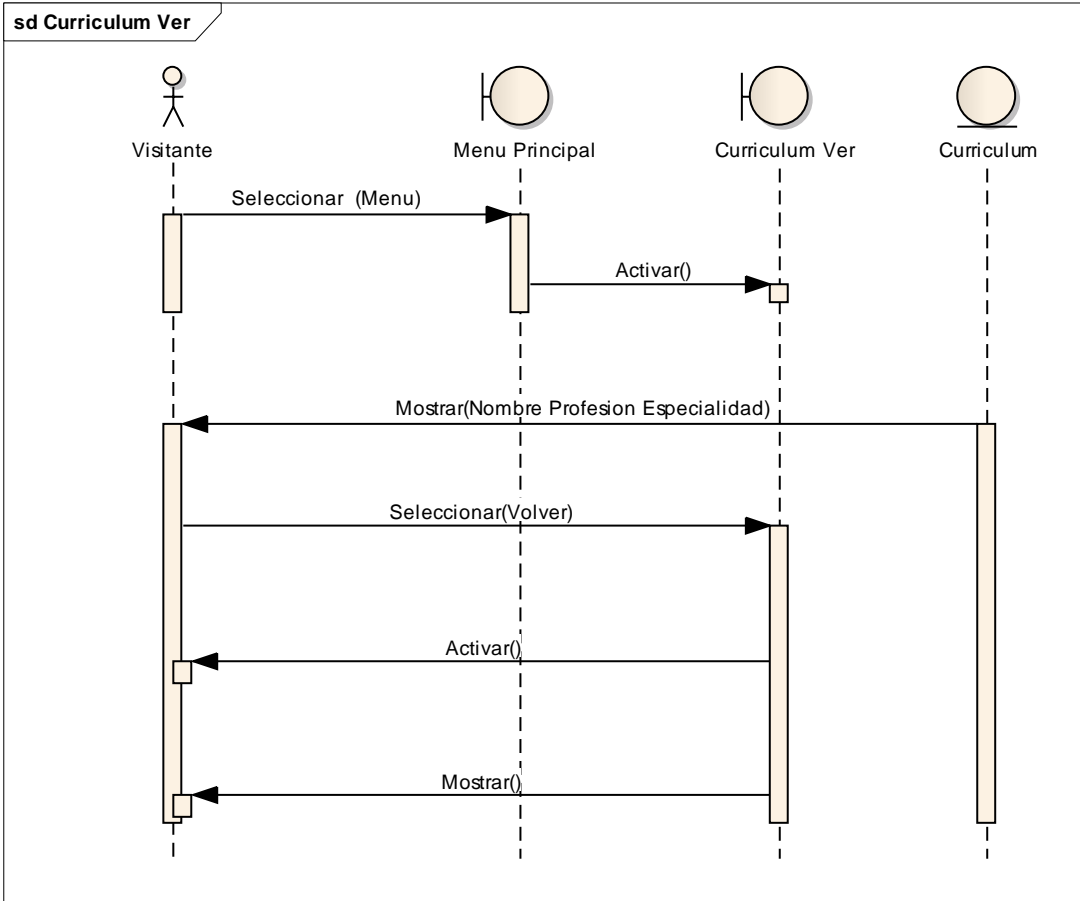


Figura 3.

II.1.5.4.4. Diagrama de Secuencia Buscar Abogado

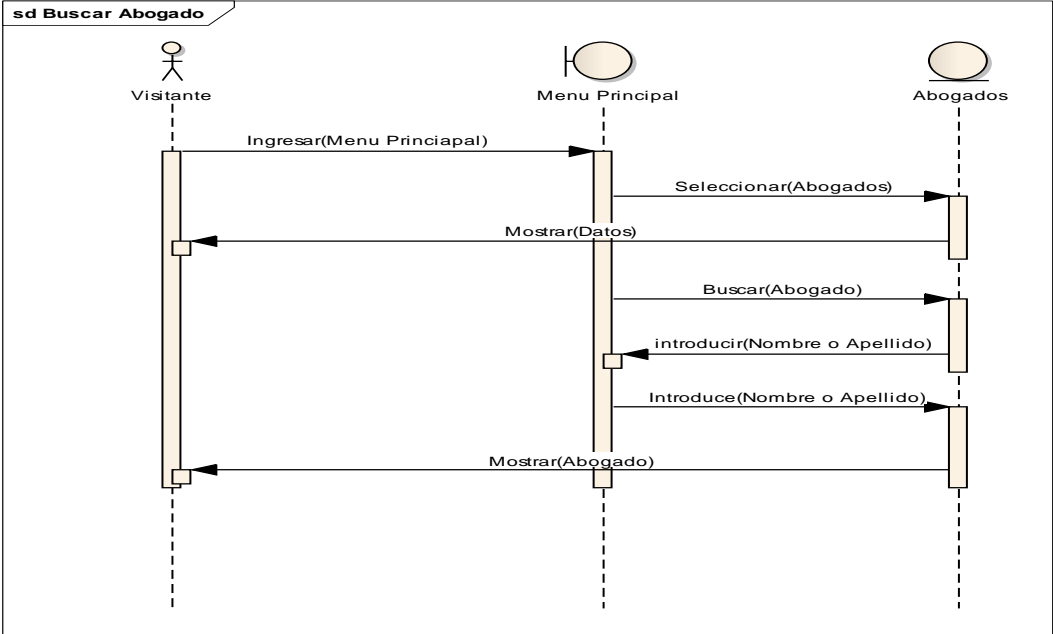


Figura 4.

II.1.5.4.5. Diagrama de Secuencia Actividades

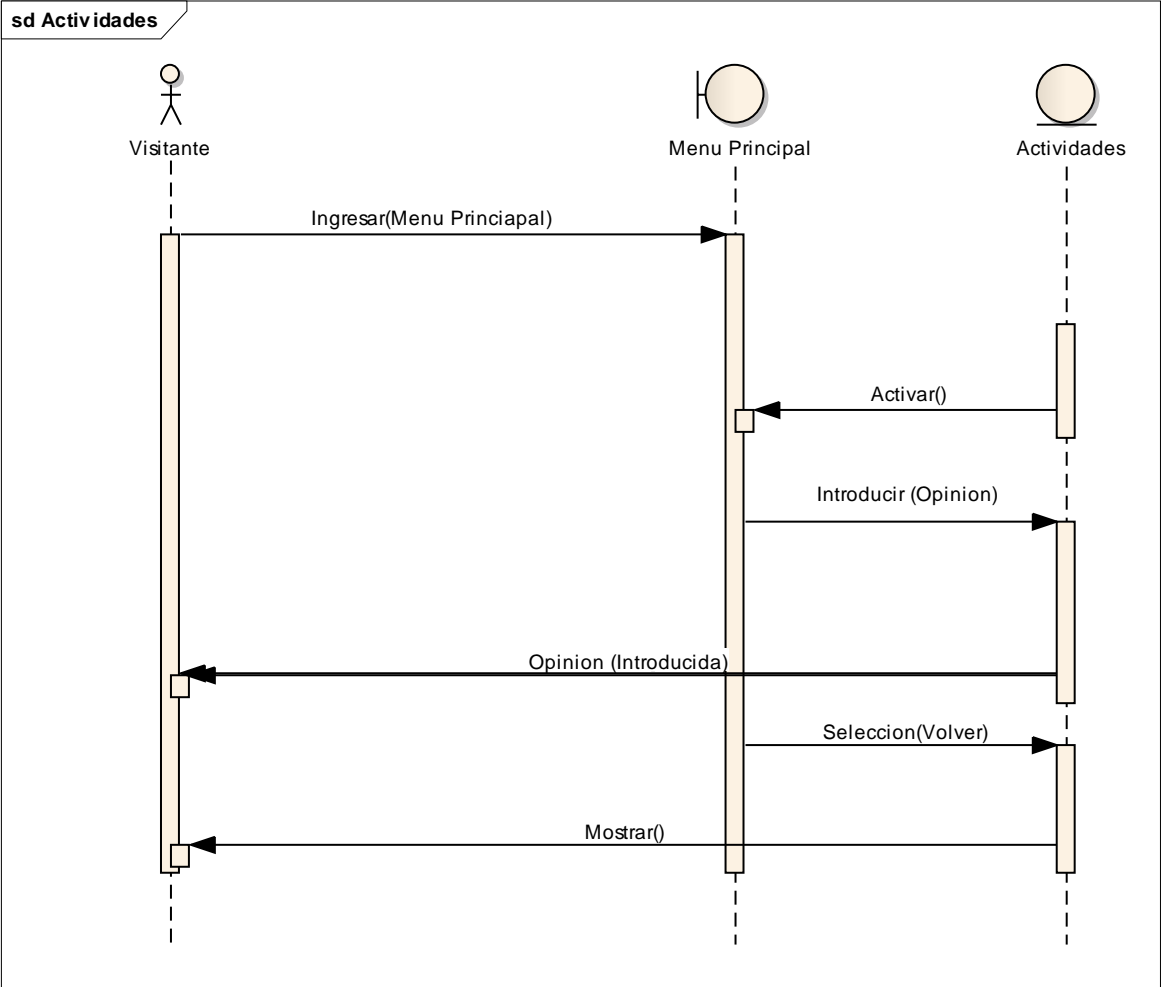


Figura 5.

II.1.5.4.6. Diagrama de Secuencia Ver Area

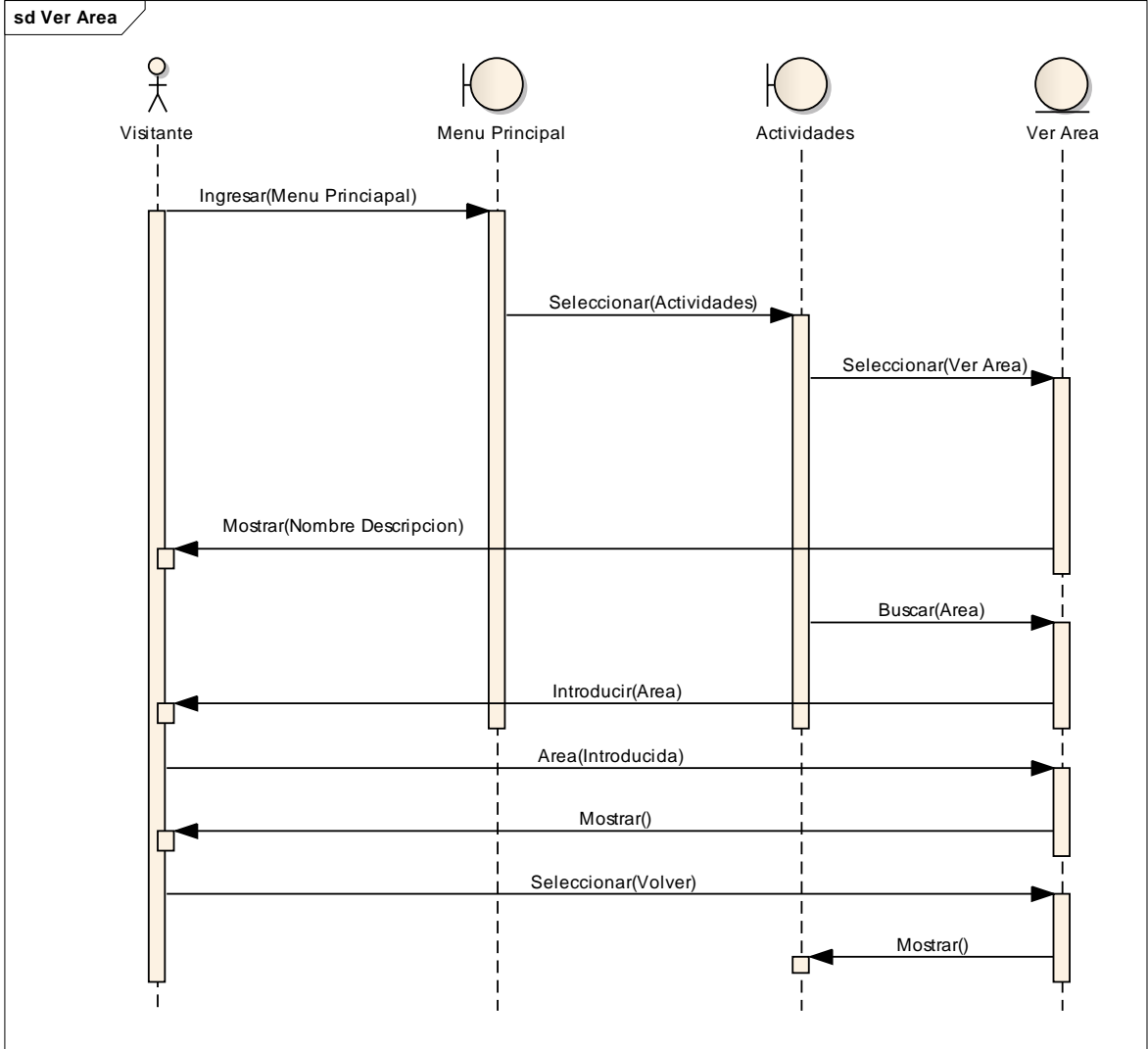


Figura 6.

II.1.5.4.7. Diagrama de Secuencia Ver Lugar

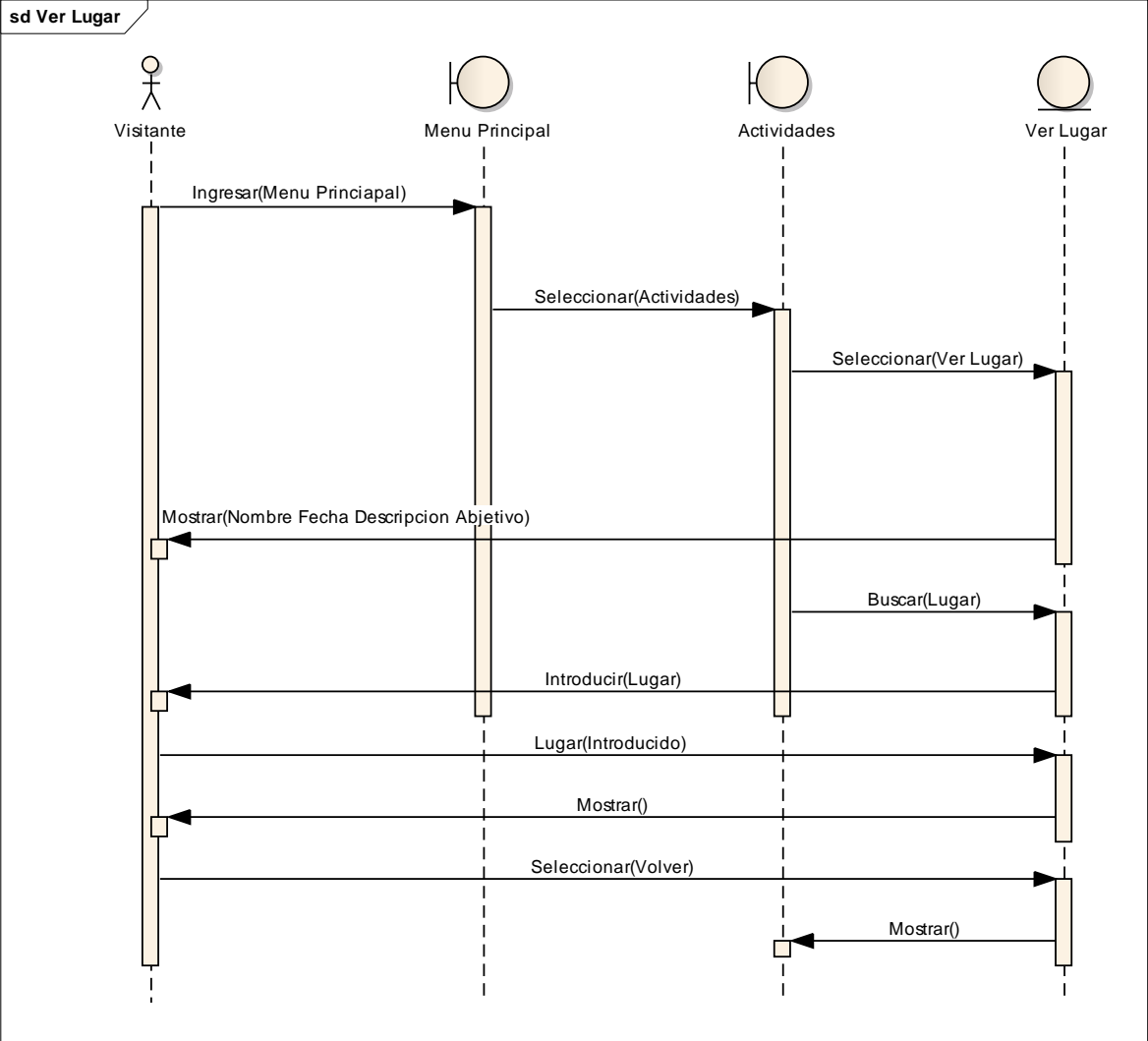


Figura 7.

II.1.5.4.8. Diagrama de Secuencia Area

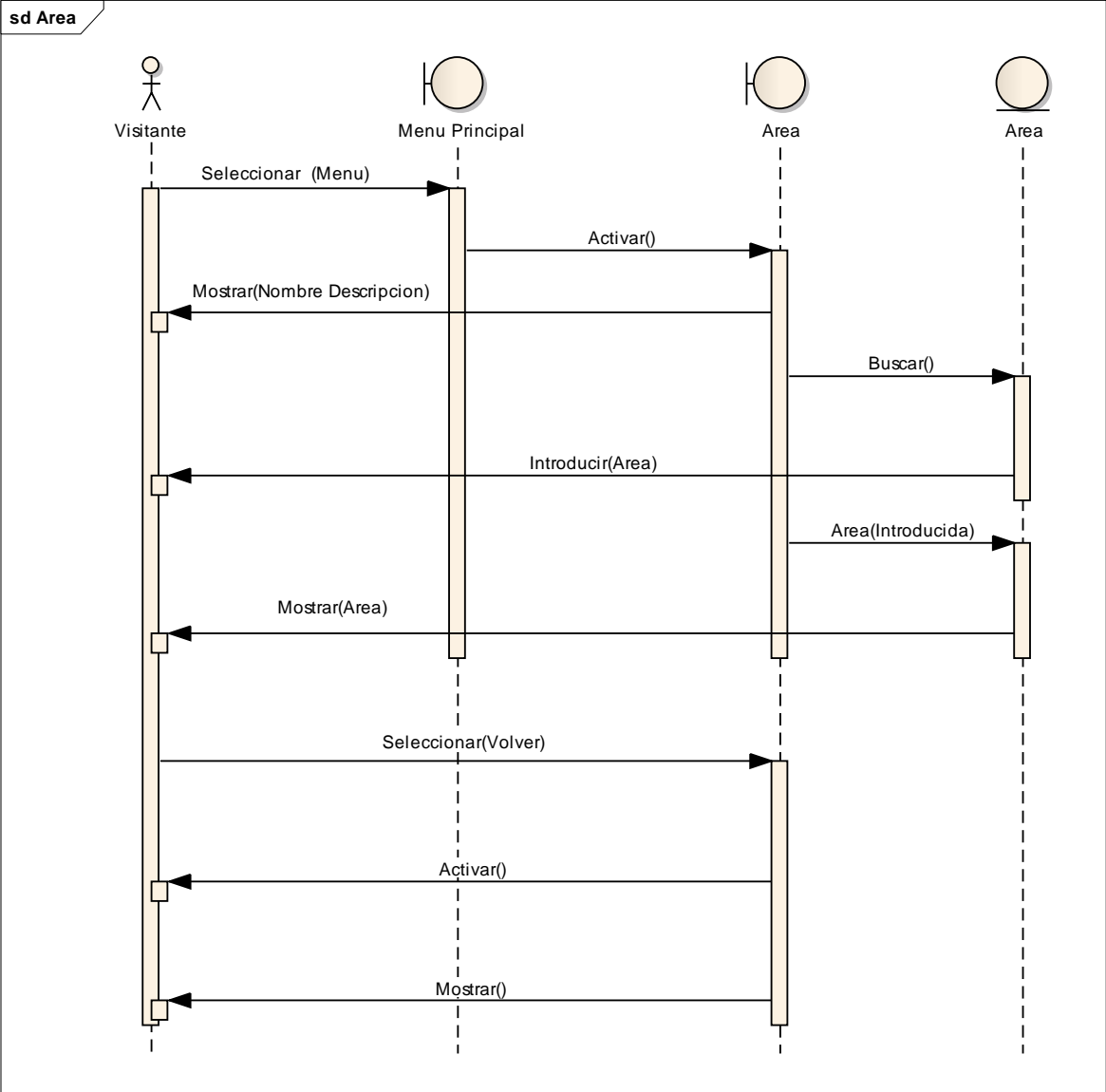


Figura 8.

II.1.5.4.9. Diagrama de Secuencia Noticia

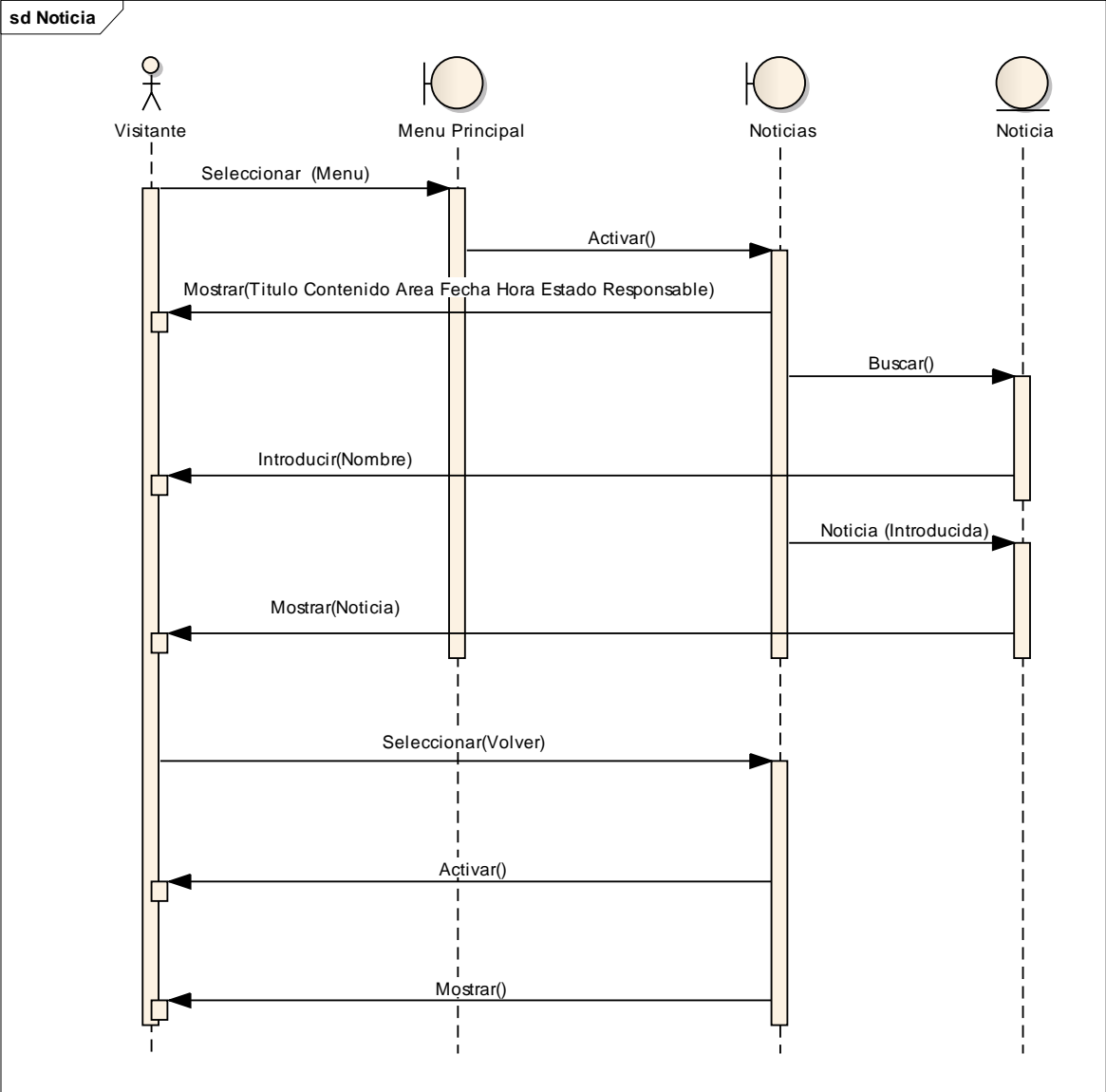


Figura 9.

II.1.5.4.10. Diagrama de Secuencia Caso Legal

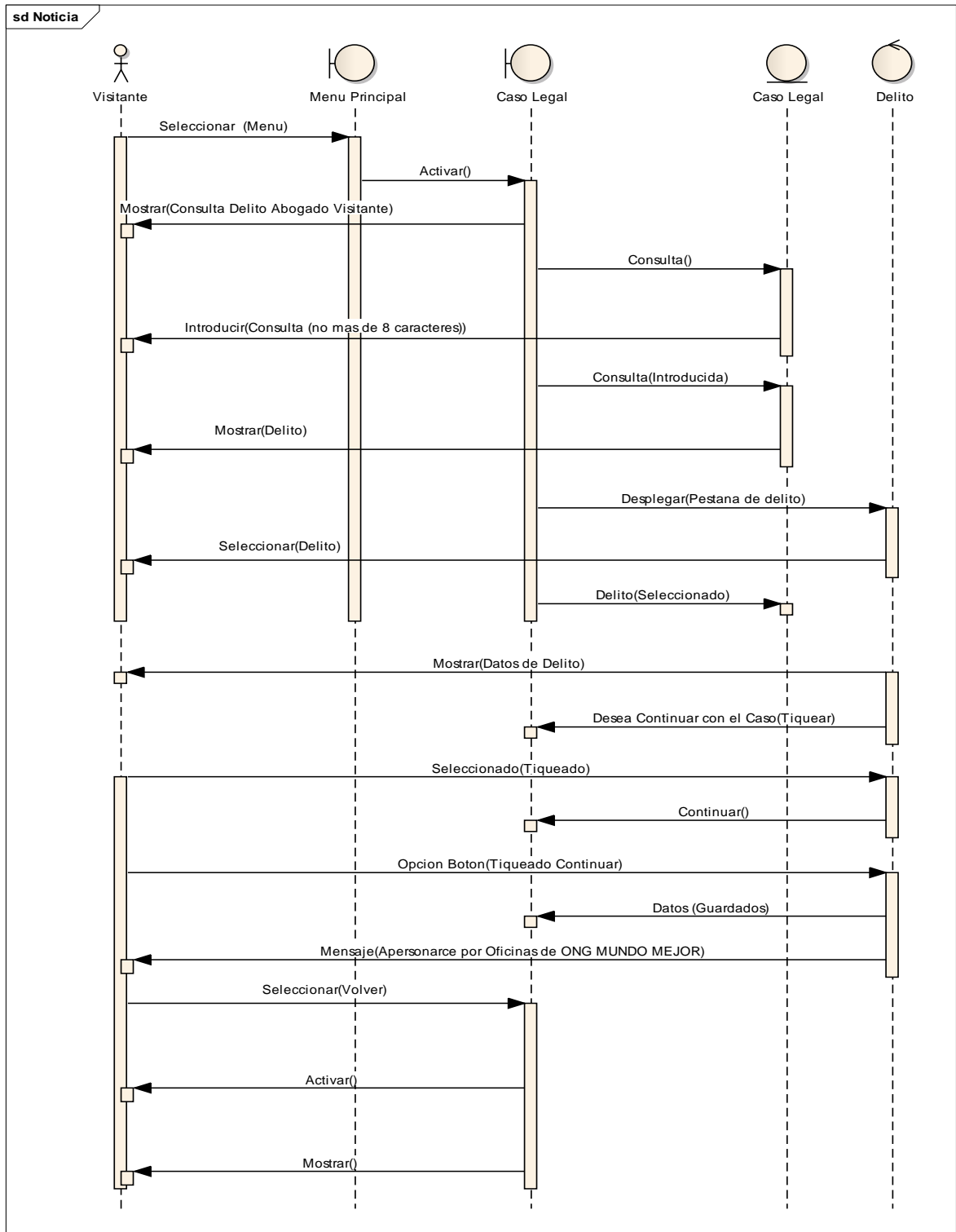


Figura 10.

II.1.5.4.11. Diagrama de Secuencia Noticia

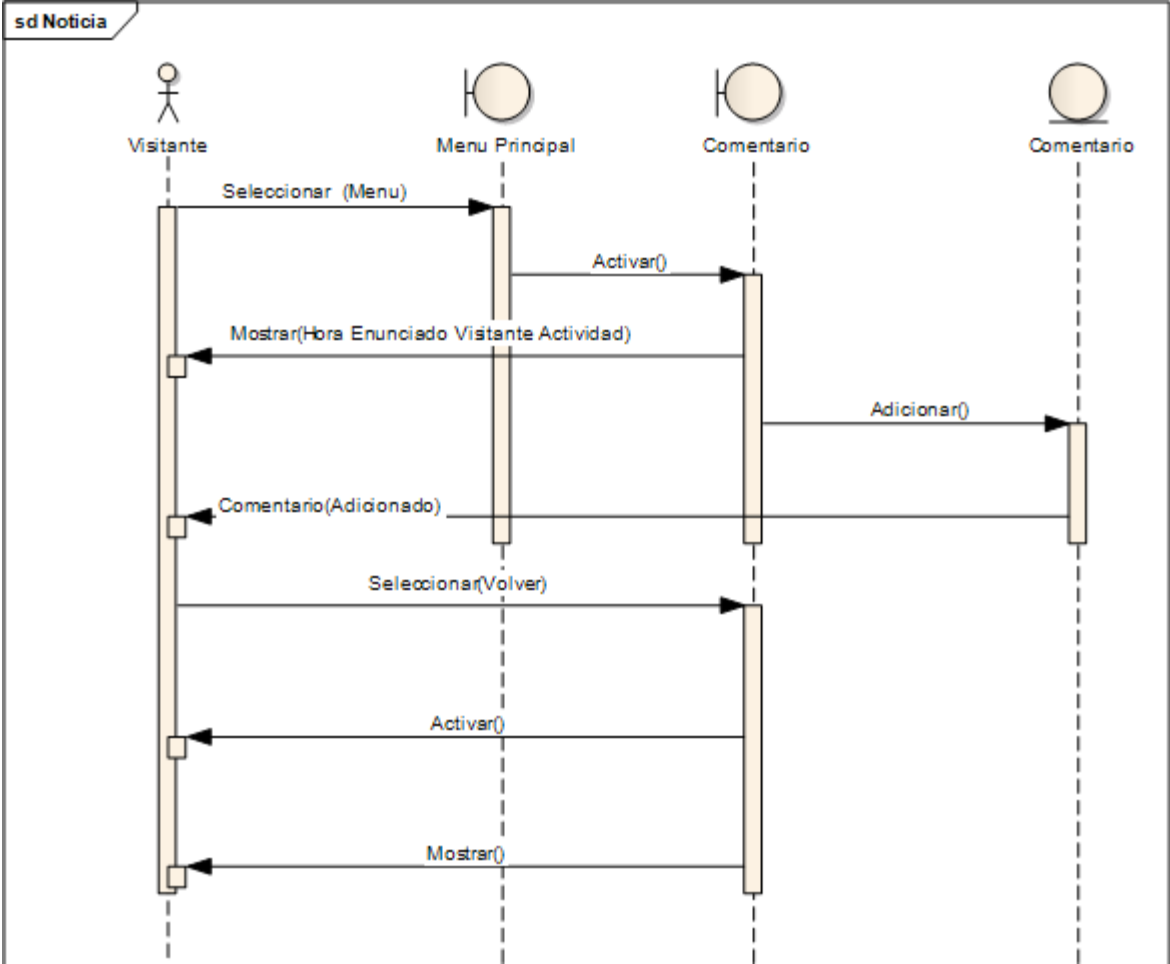


Figura 11.

II.1.5.4.12. Diagrama de Secuencia Ingresar Sistema Abogado

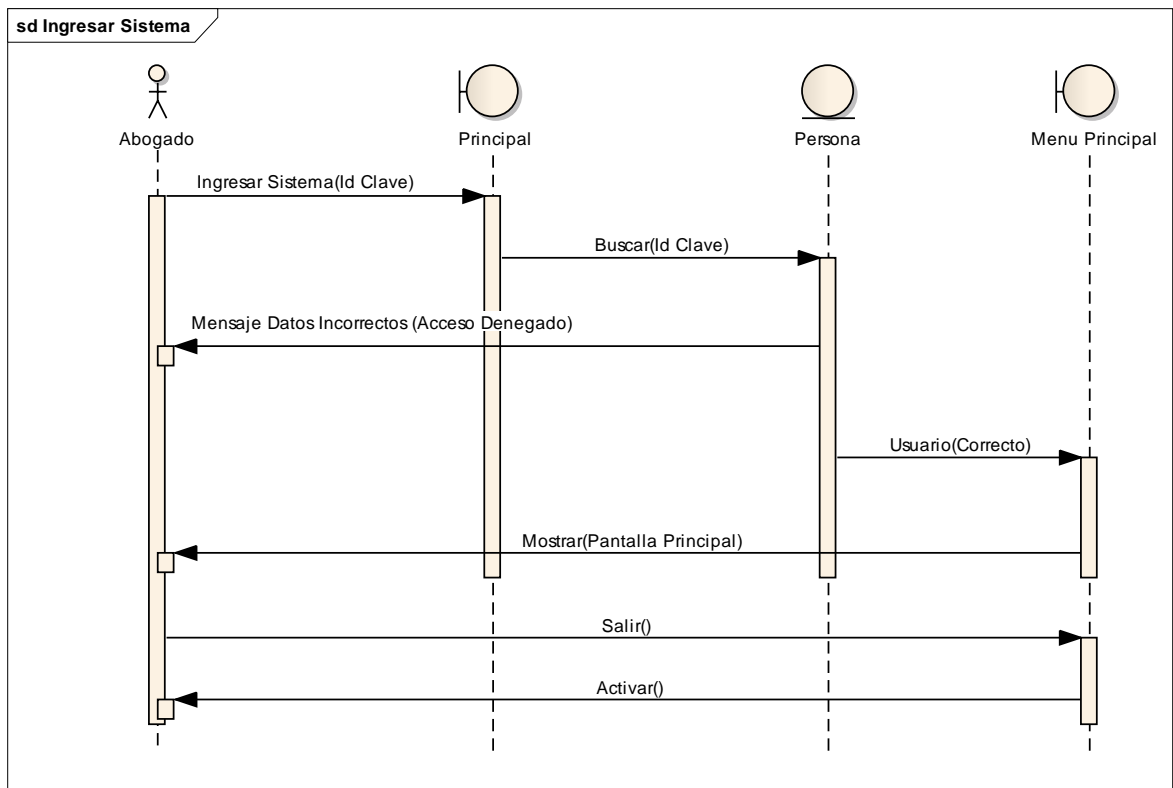


Figura 12.

II.1.5.4.13. Diagrama de Secuencia Beneficiarios

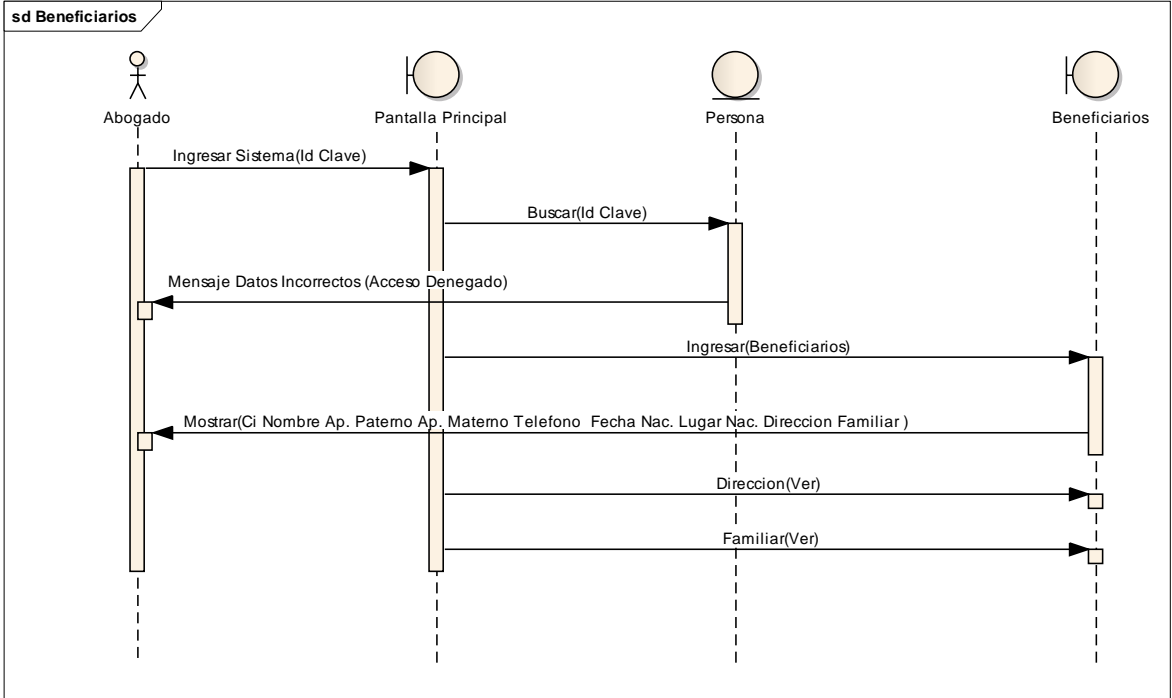


Figura 13.

II.1.5.4.14. Diagrama de Secuencia Beneficiario Dirección

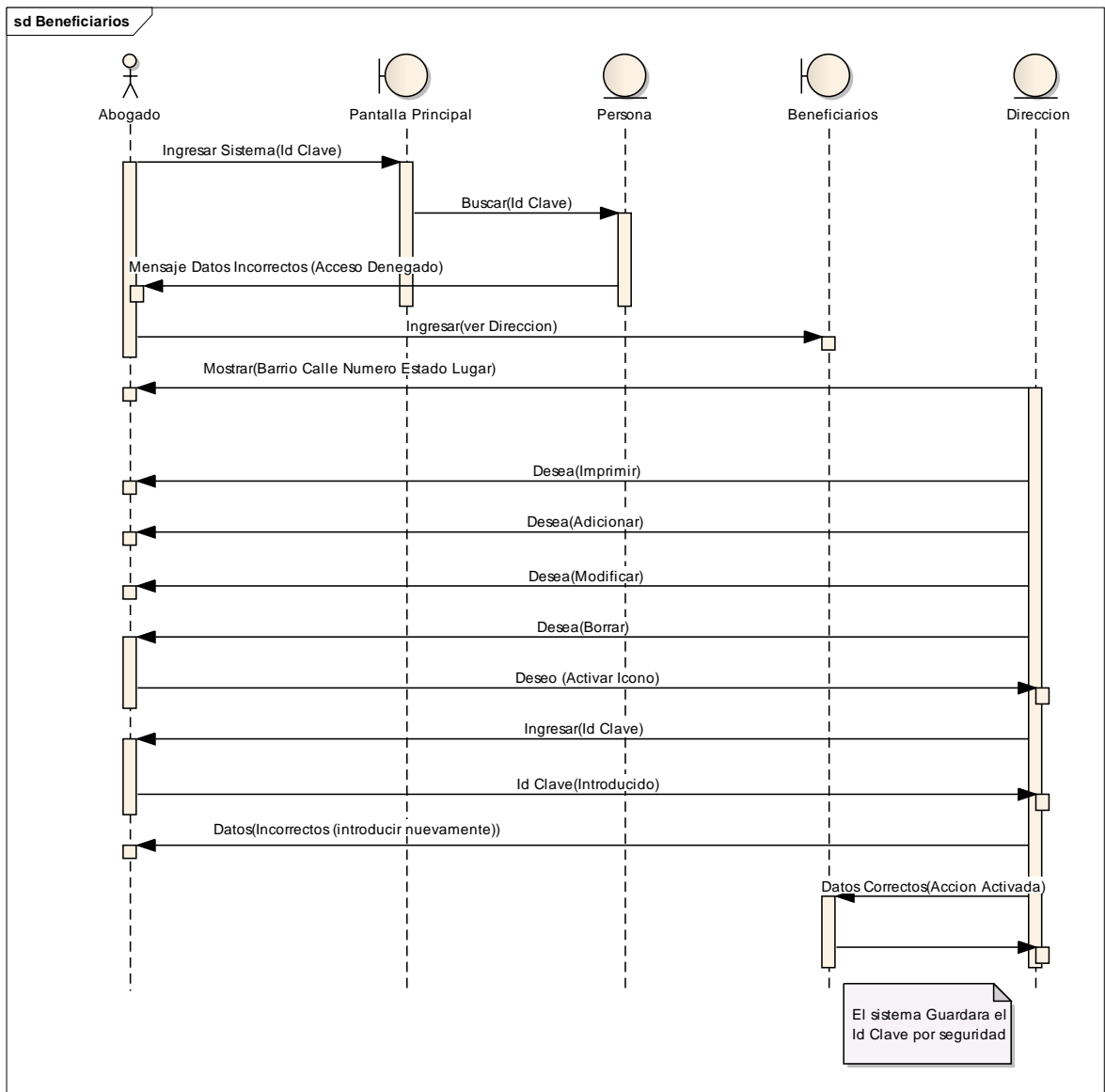


Figura 14.

II.1.5.4.15. Diagrama de Secuencia Beneficiarios Familiar

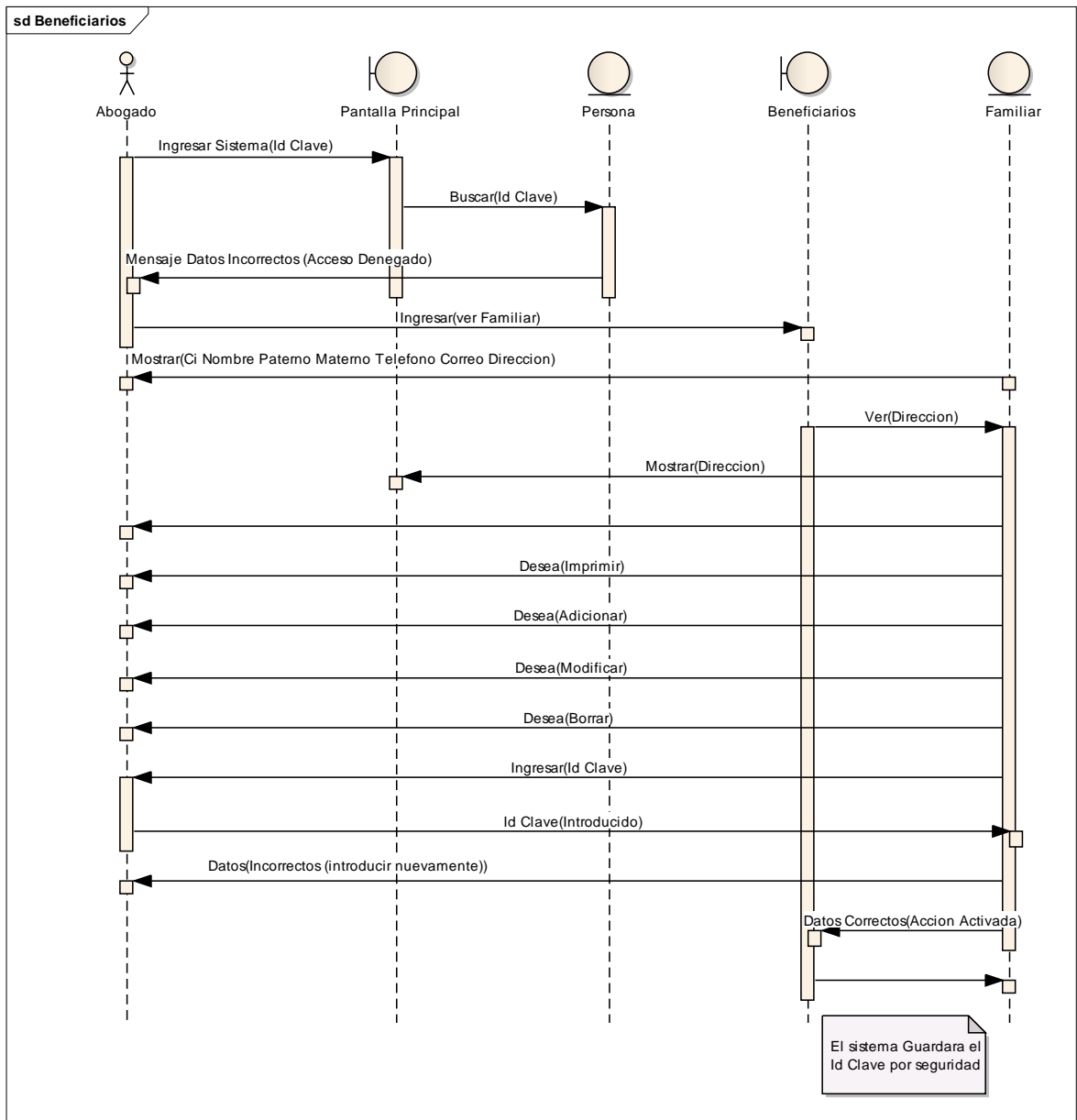


Figura 15.

II.1.5.4.16. Diagrama de Secuencia Ingresar Sistema Administrador

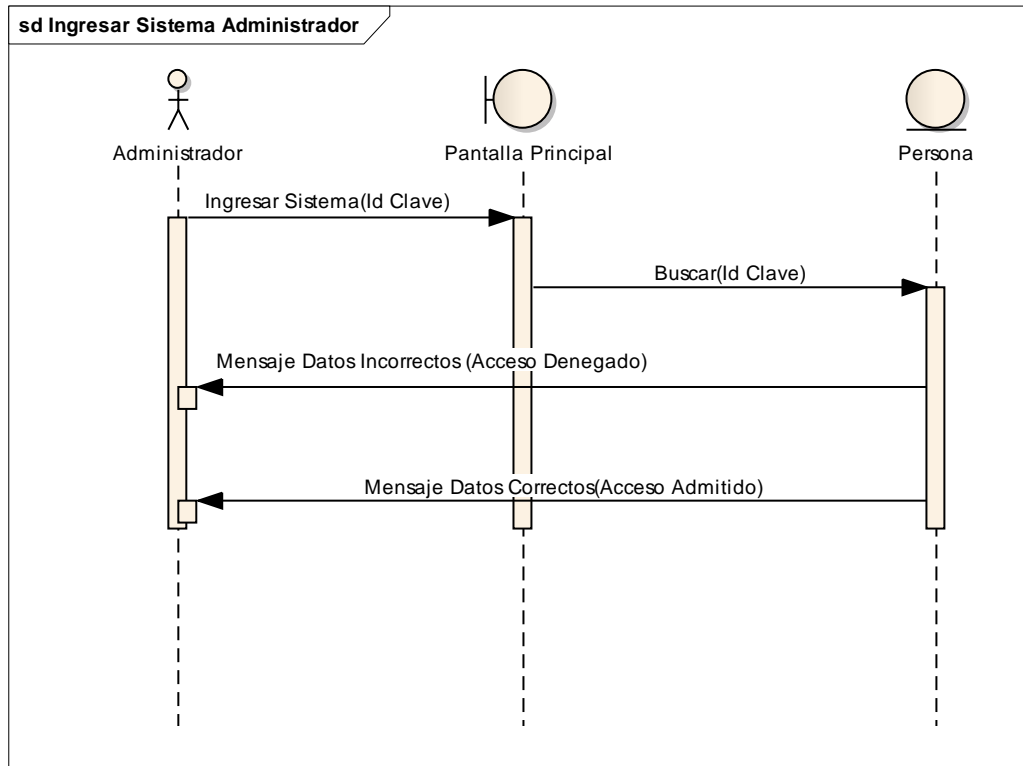


Figura 16.

II.1.5.4.17. Diagrama de Secuencia Administrador Adicionar Abogado

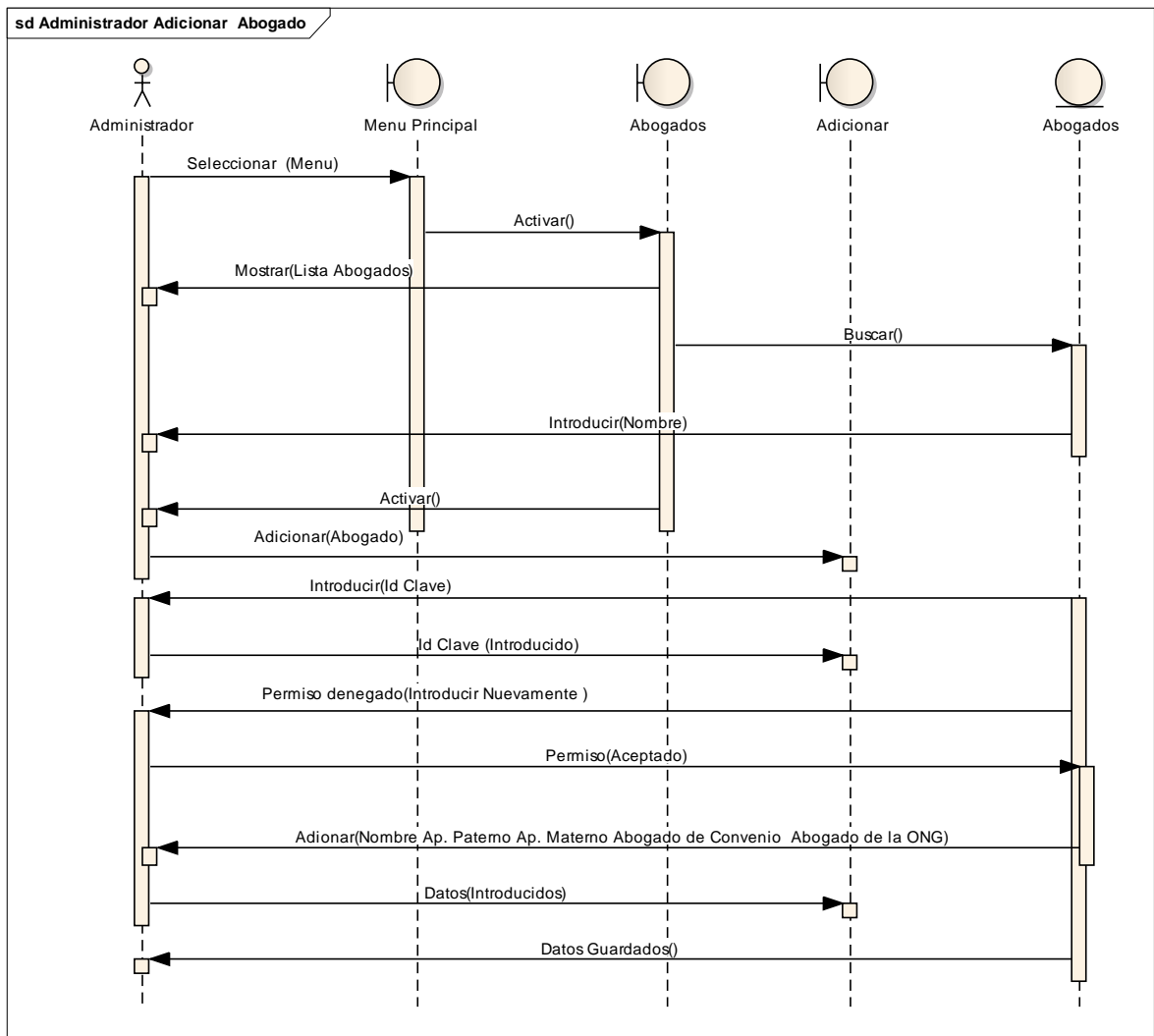


Figura 17.

II.1.5.4.18. Diagrama de Secuencia Administrador Modificar Abogado

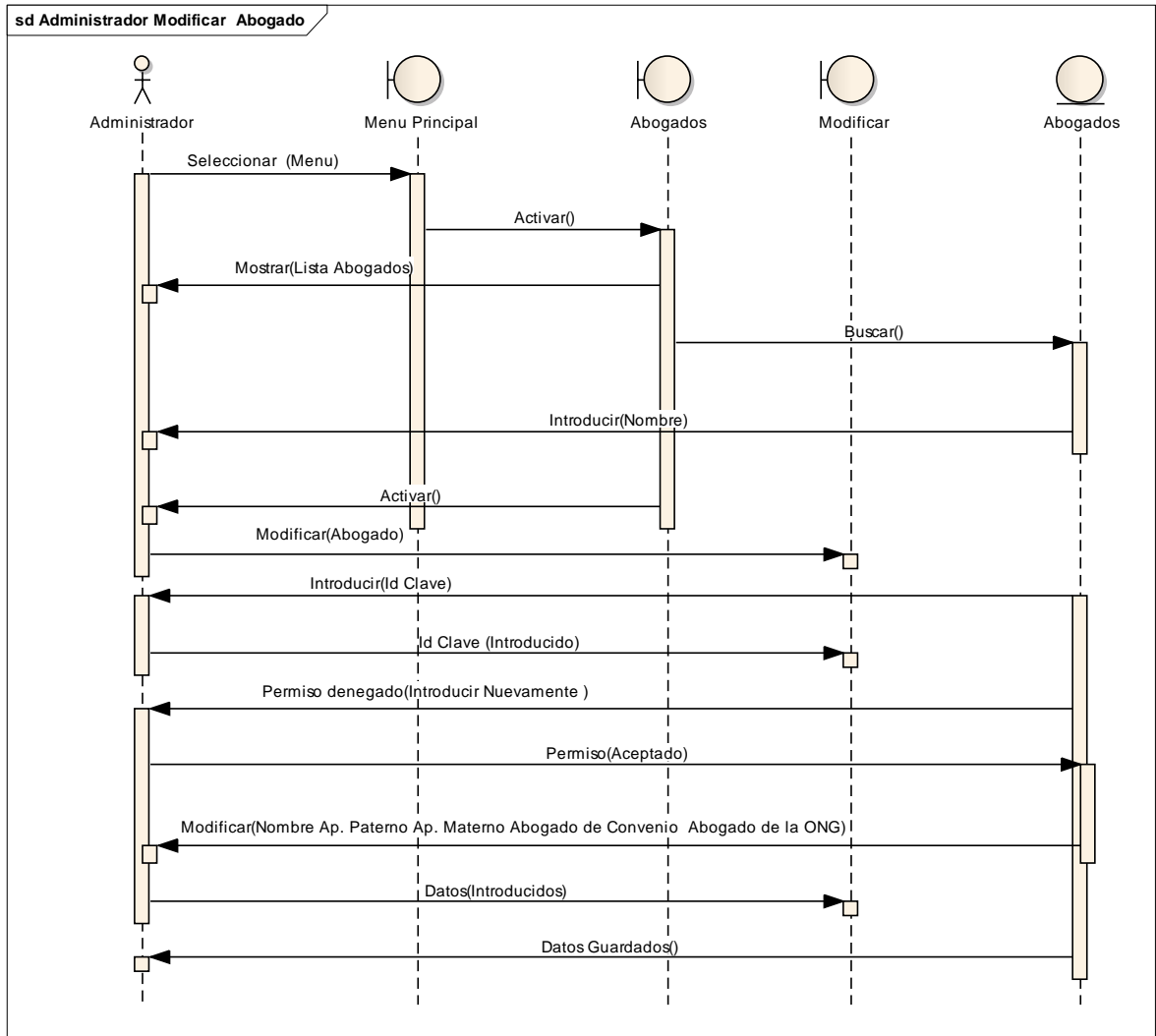


Figura 18.

II.1.5.4.19. Diagrama de Secuencia Administrador Borrar Abogado

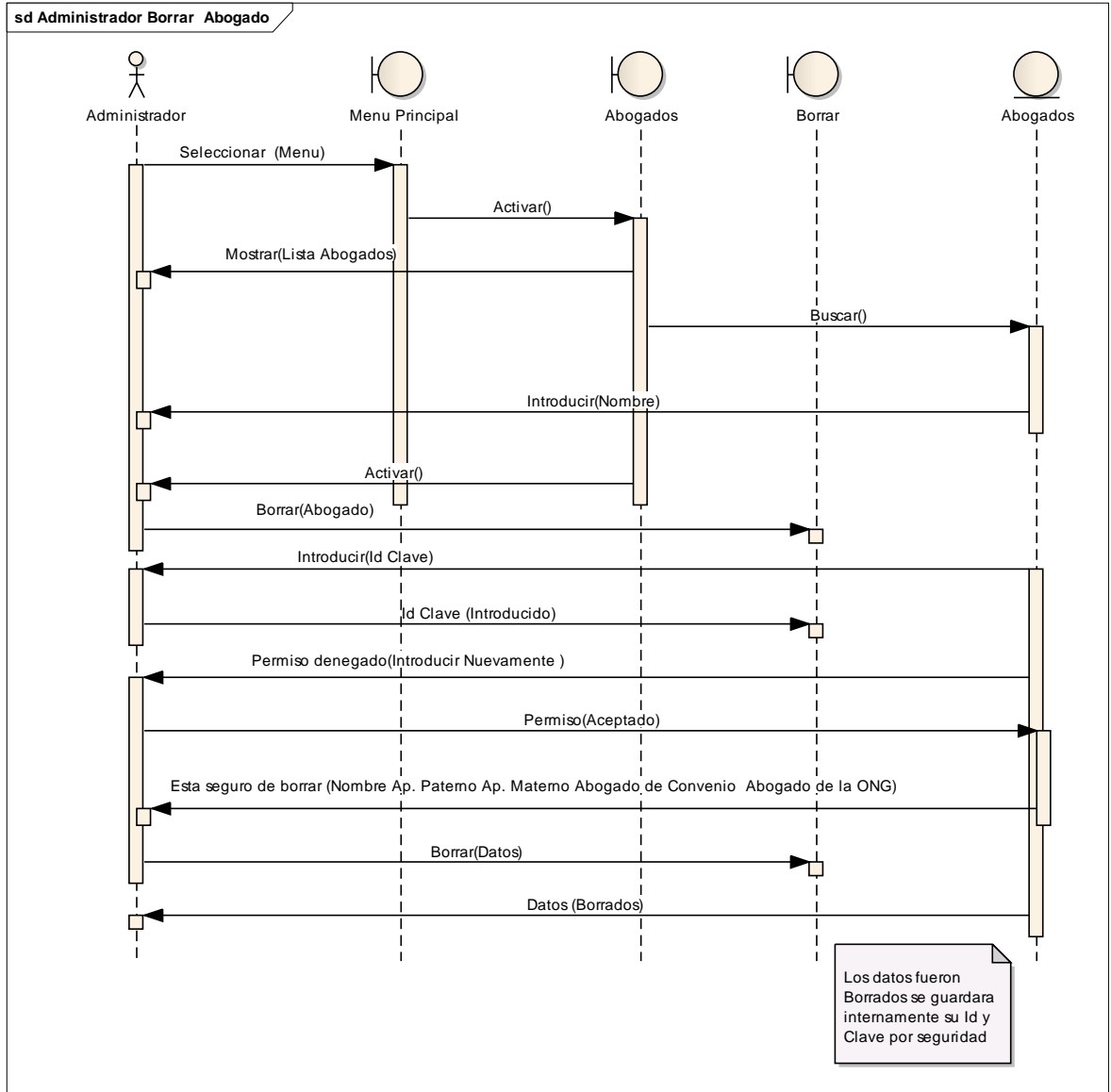


Figura 19.

II.1.5.4.20. Diagrama de Secuencia Administrador Adicionar Curriculum

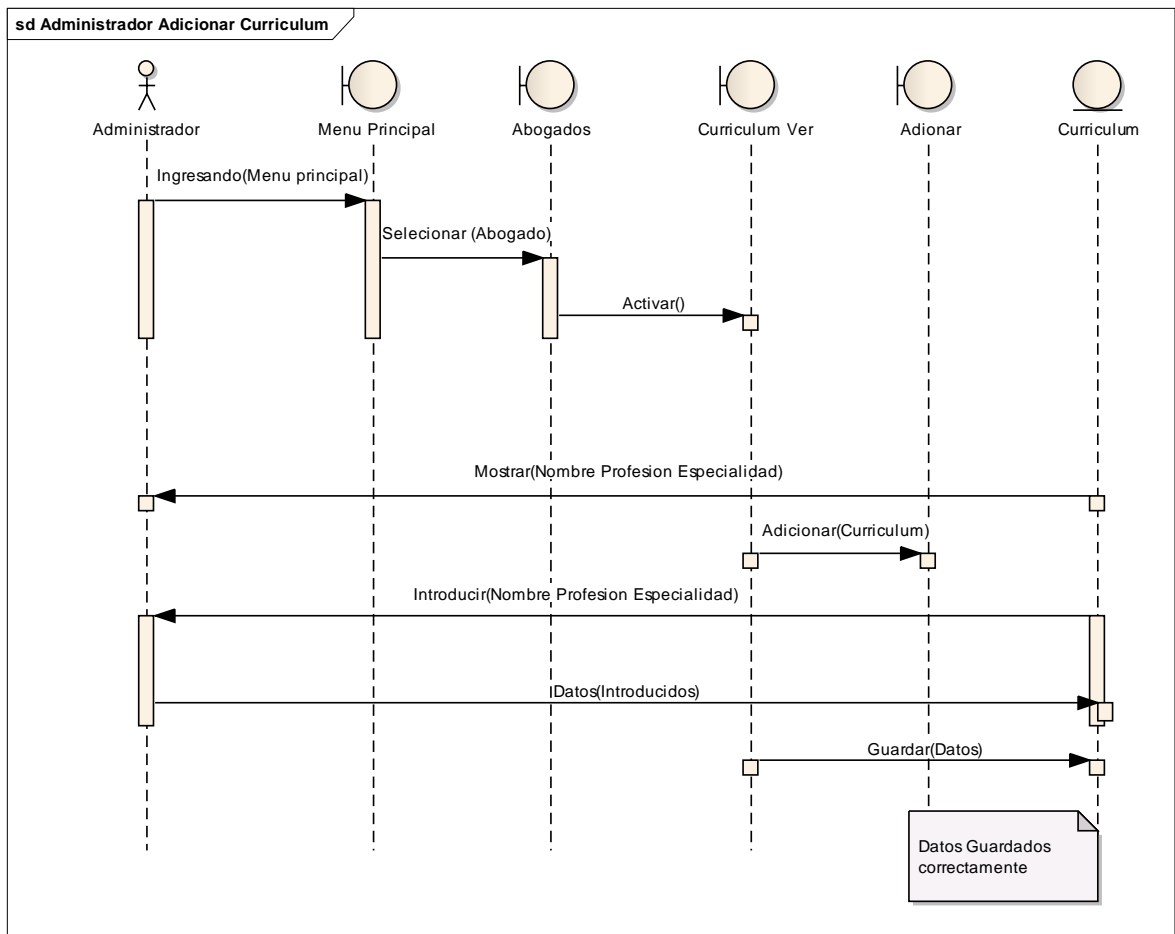


Figura 20.

II.1.5.4.21. Diagrama de Secuencia Administrador Modificar Curriculum

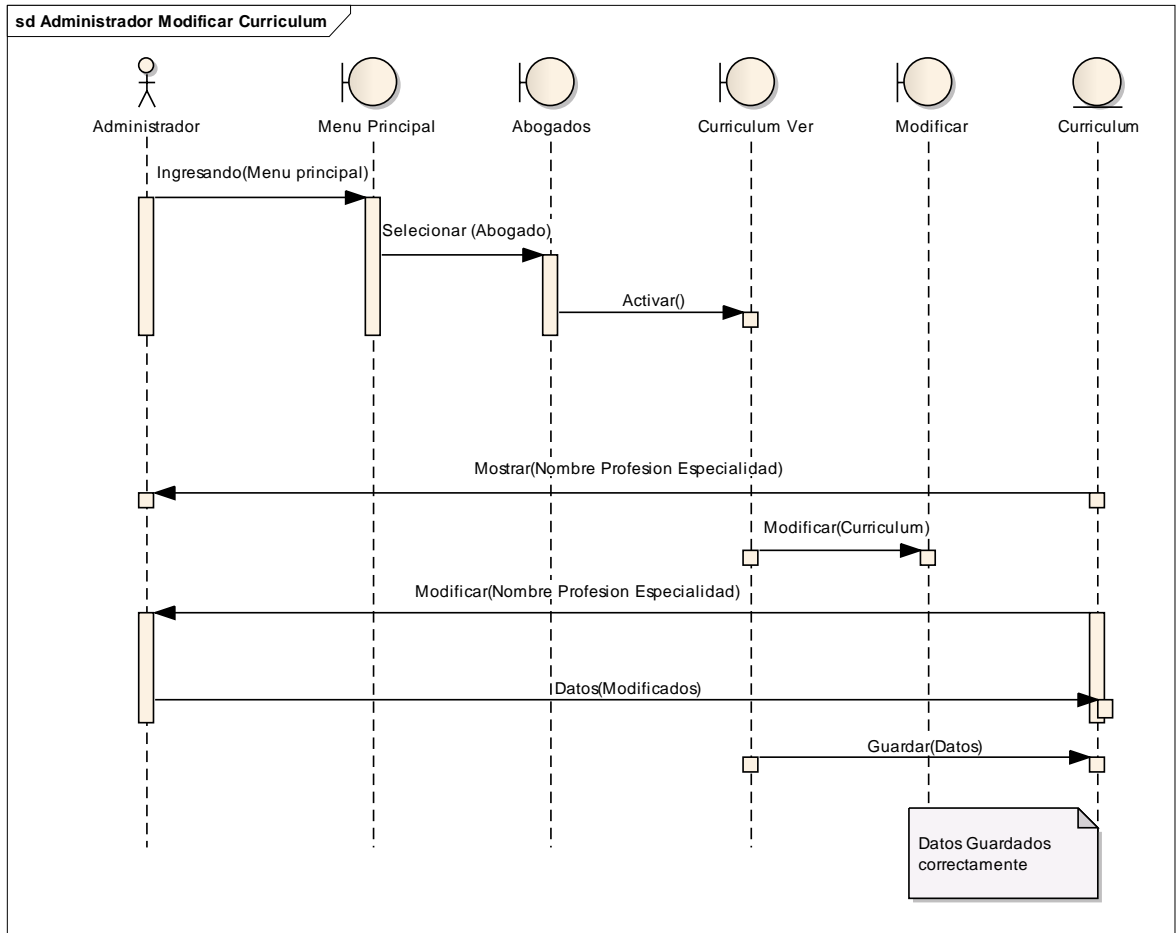


Figura 21.

II.1.5.4.22. Diagrama de Secuencia Administrador Borrar Curriculum

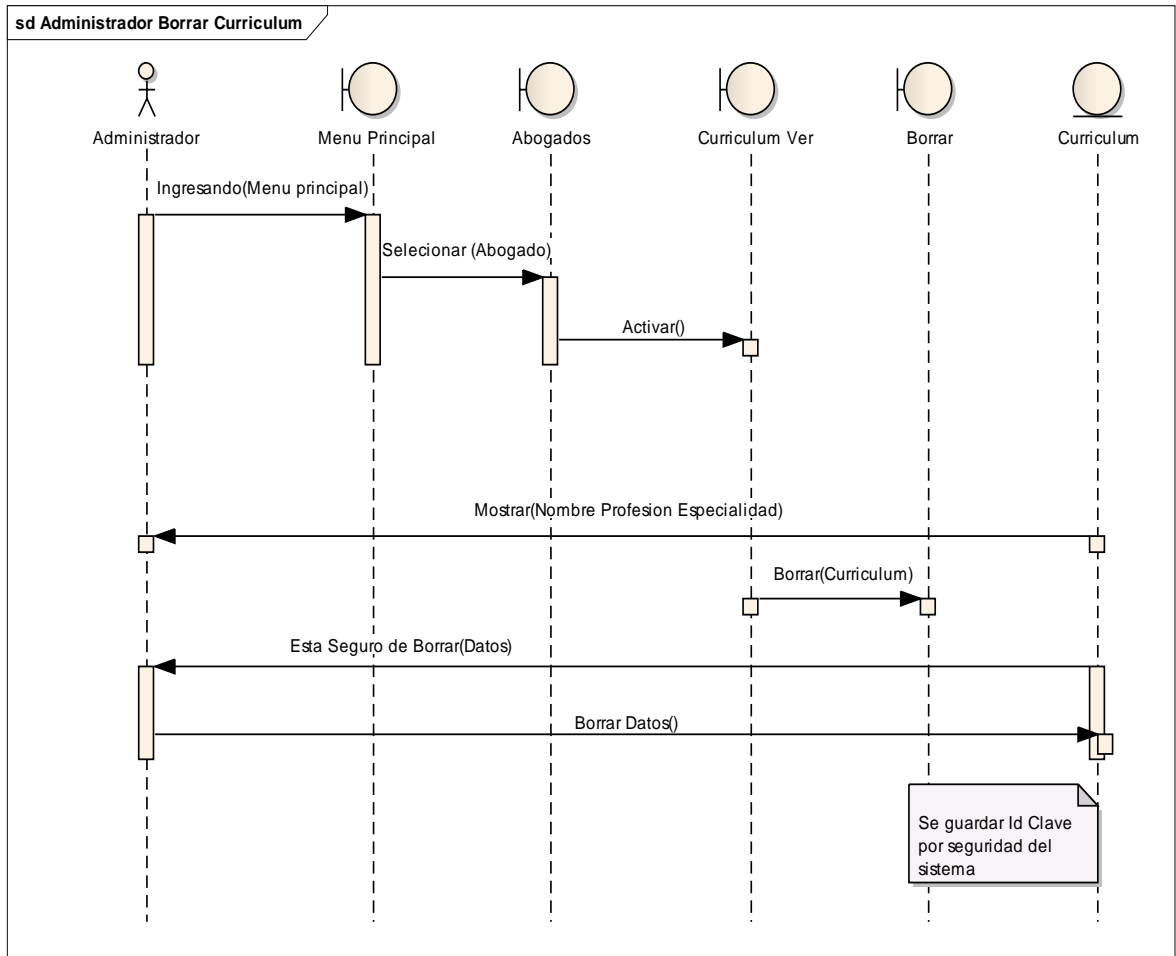


Figura 22.

II.1.5.4.23. Diagrama de Secuencia Administrador Adicionar Actividades

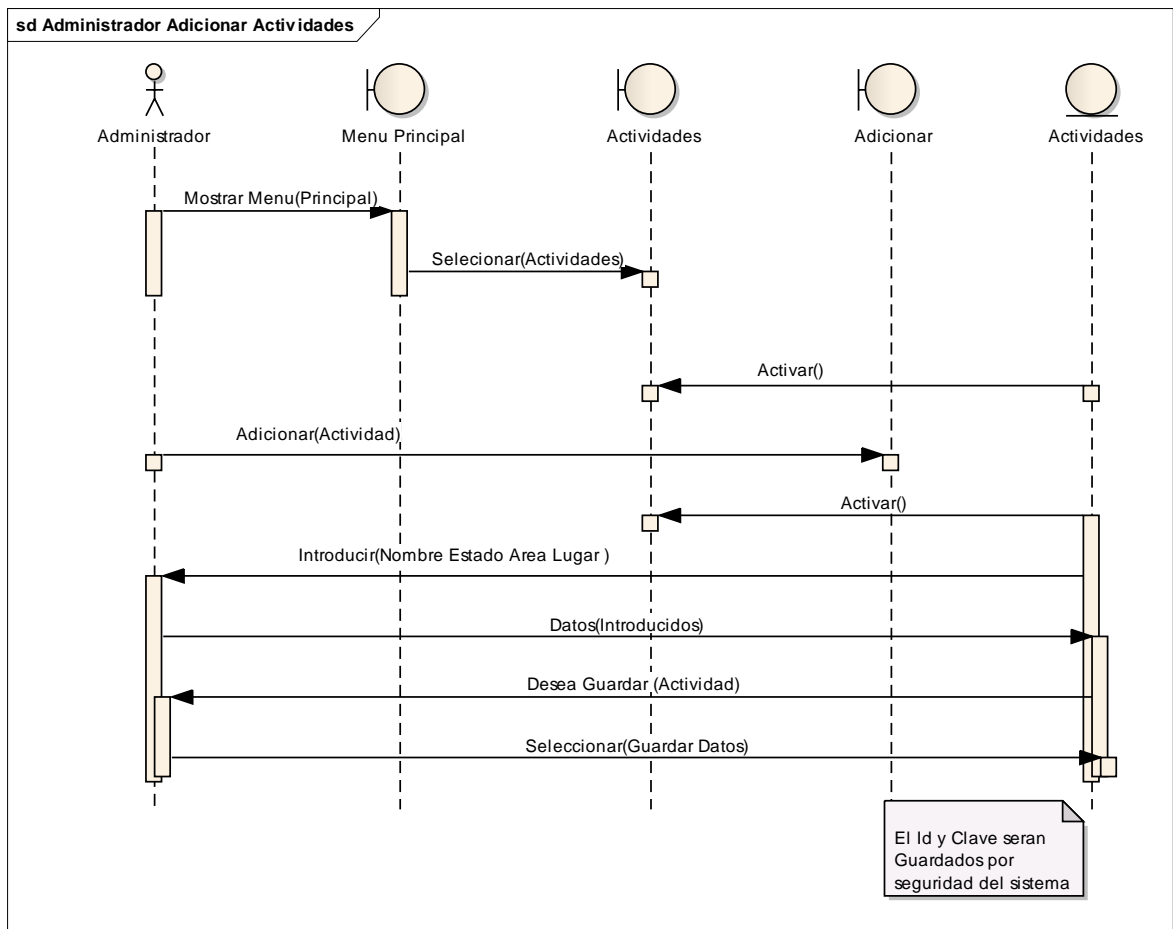


Figura 23.

II.1.5.4.24. Diagrama de Secuencia Administrador Modificar Actividades

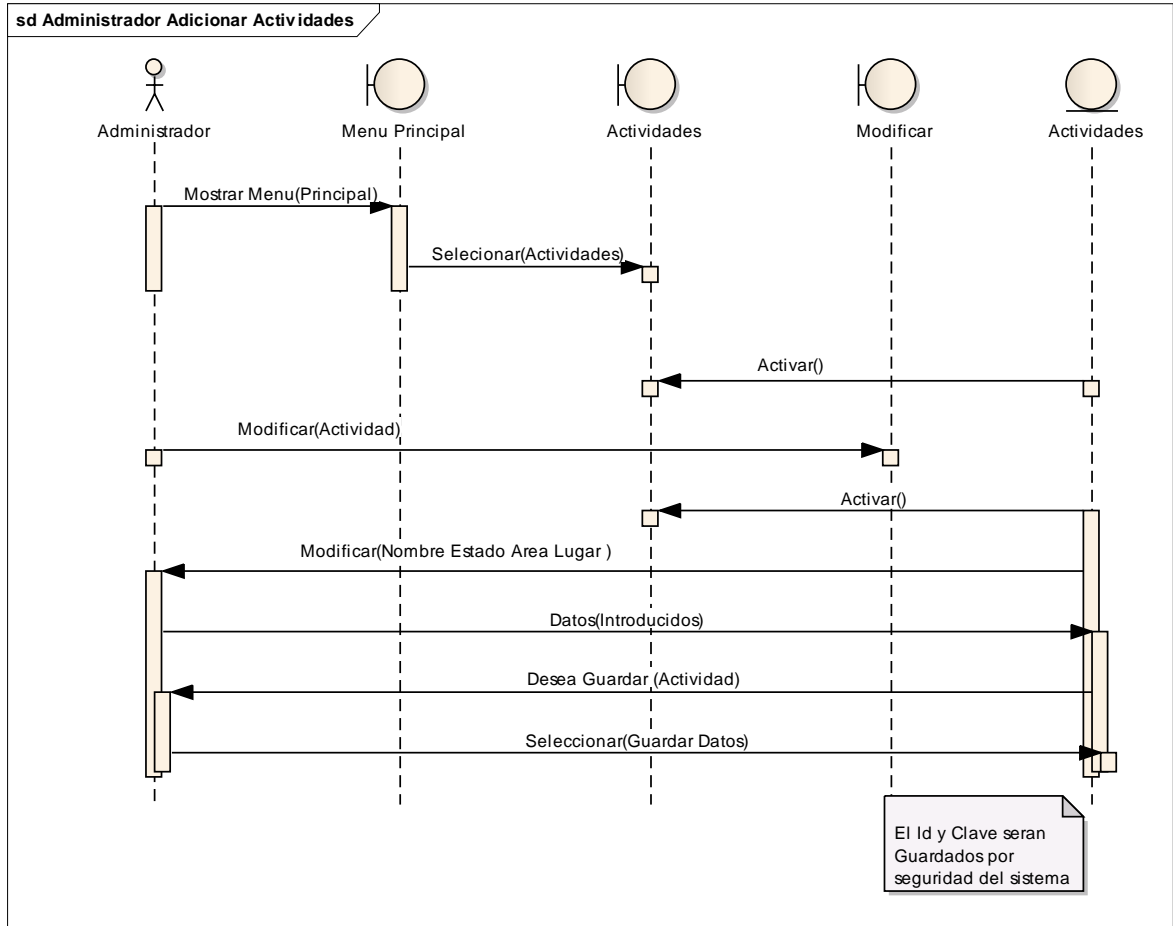


Figura 24.

II.1.5.4.25. Diagrama de Secuencia Administrador Borrar Actividades

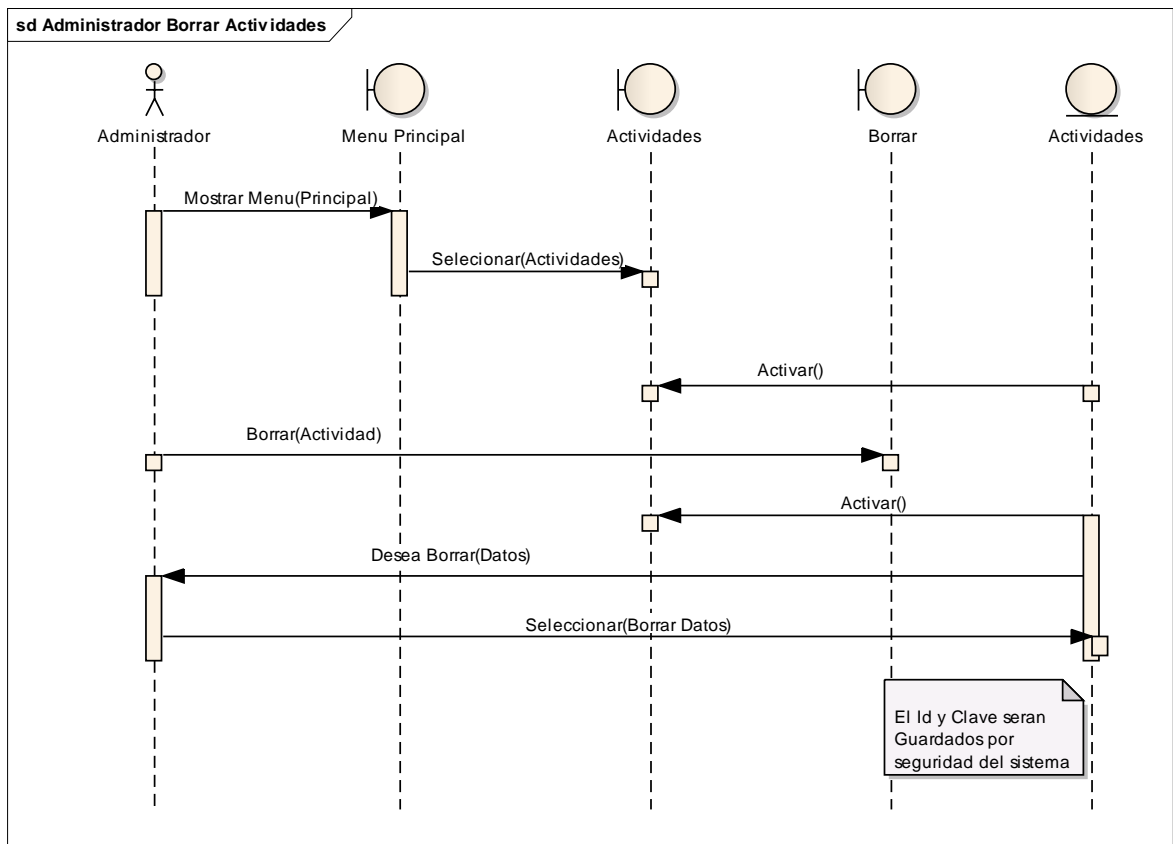


Figura 25.

II.1.5.4.26. Diagrama de Secuencia Administrador Modificar Area

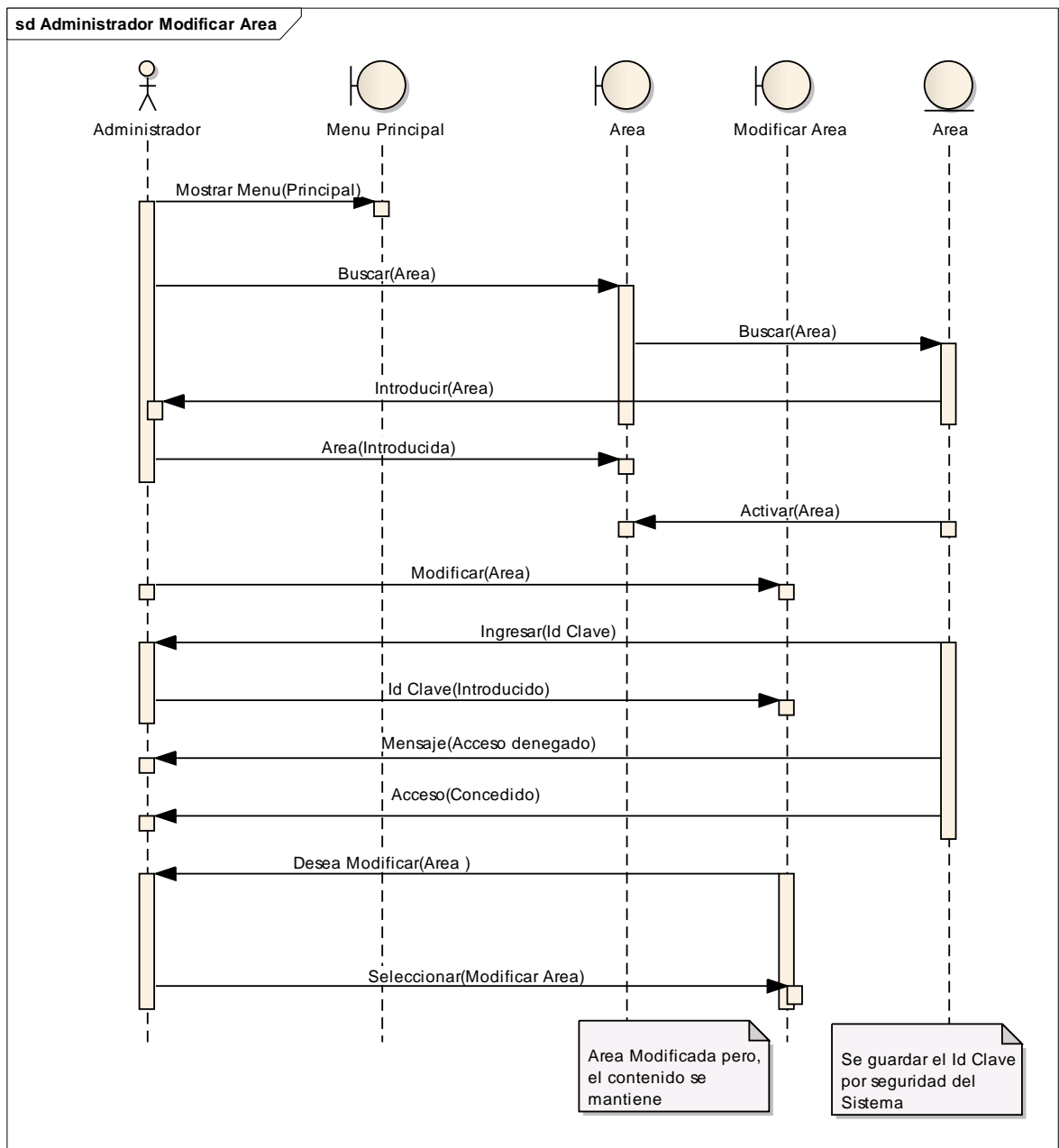


Figura 26.

II.1.5.4.27. Diagrama de Secuencia Modificar Descripción Area

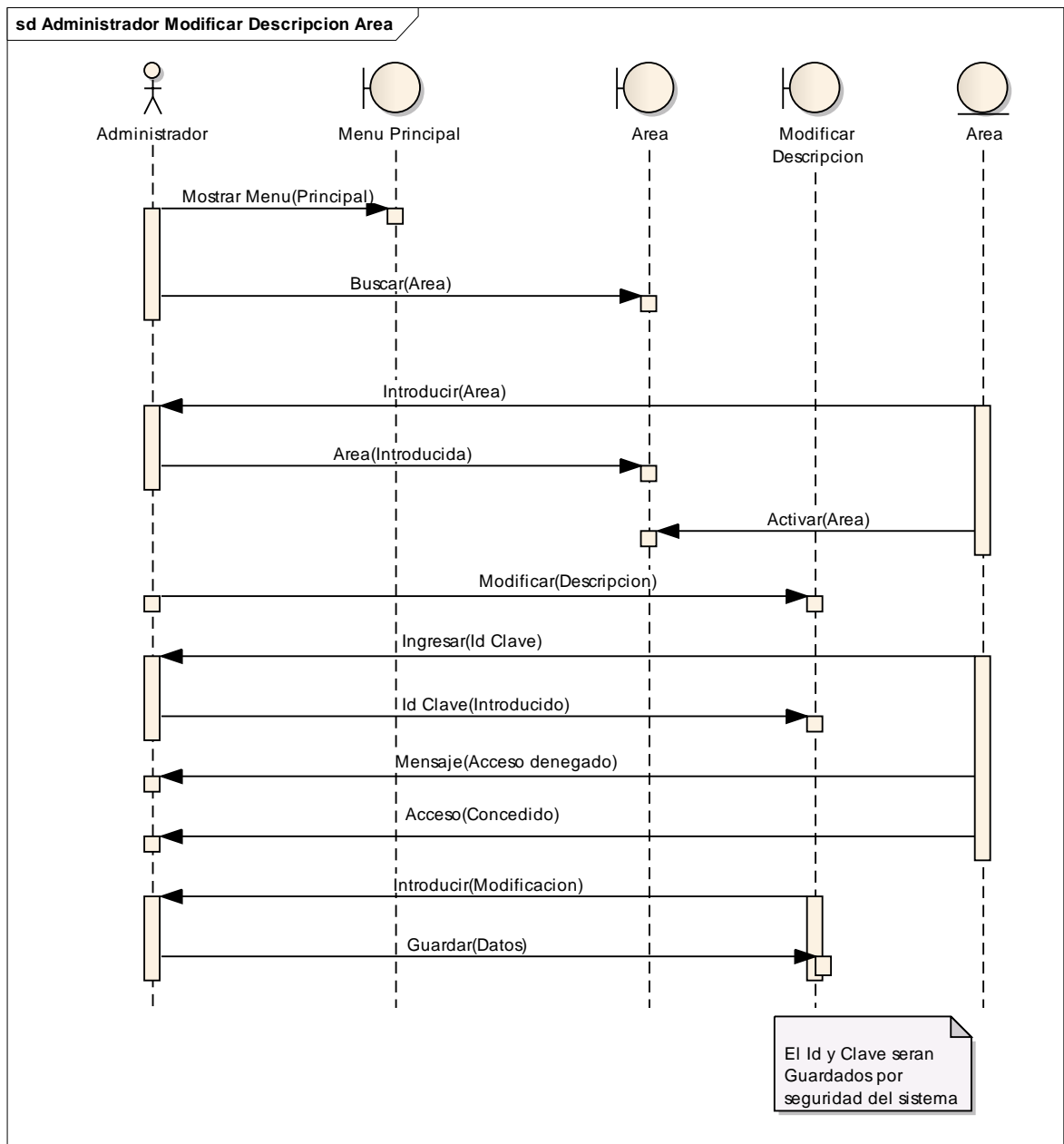


Figura 27.

II.1.5.4.28. Diagrama de Secuencia Administrador Adicionar Area

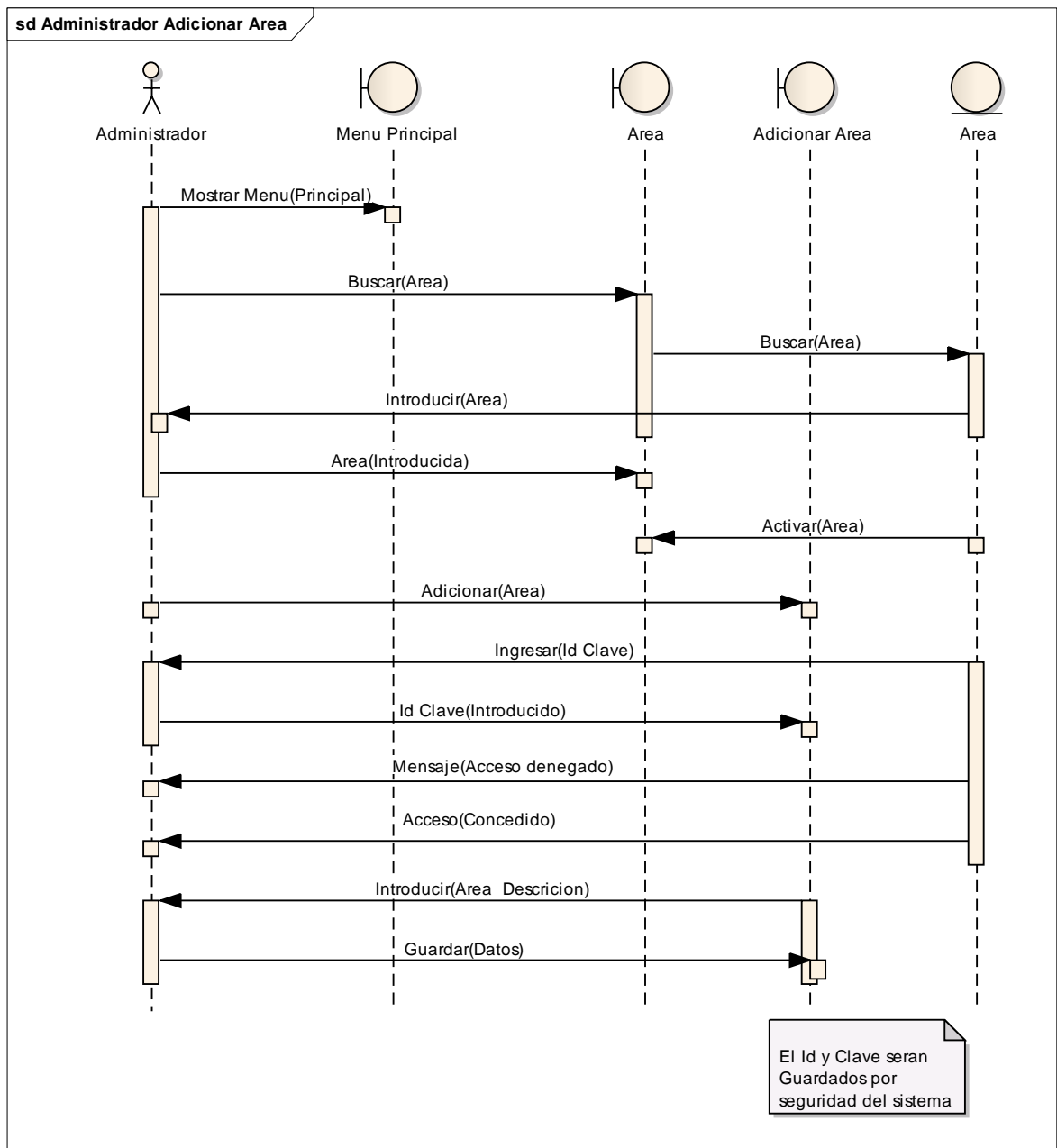


Figura 28.

II.1.5.4.29. Diagrama de Secuencia Administrador Borrar Area

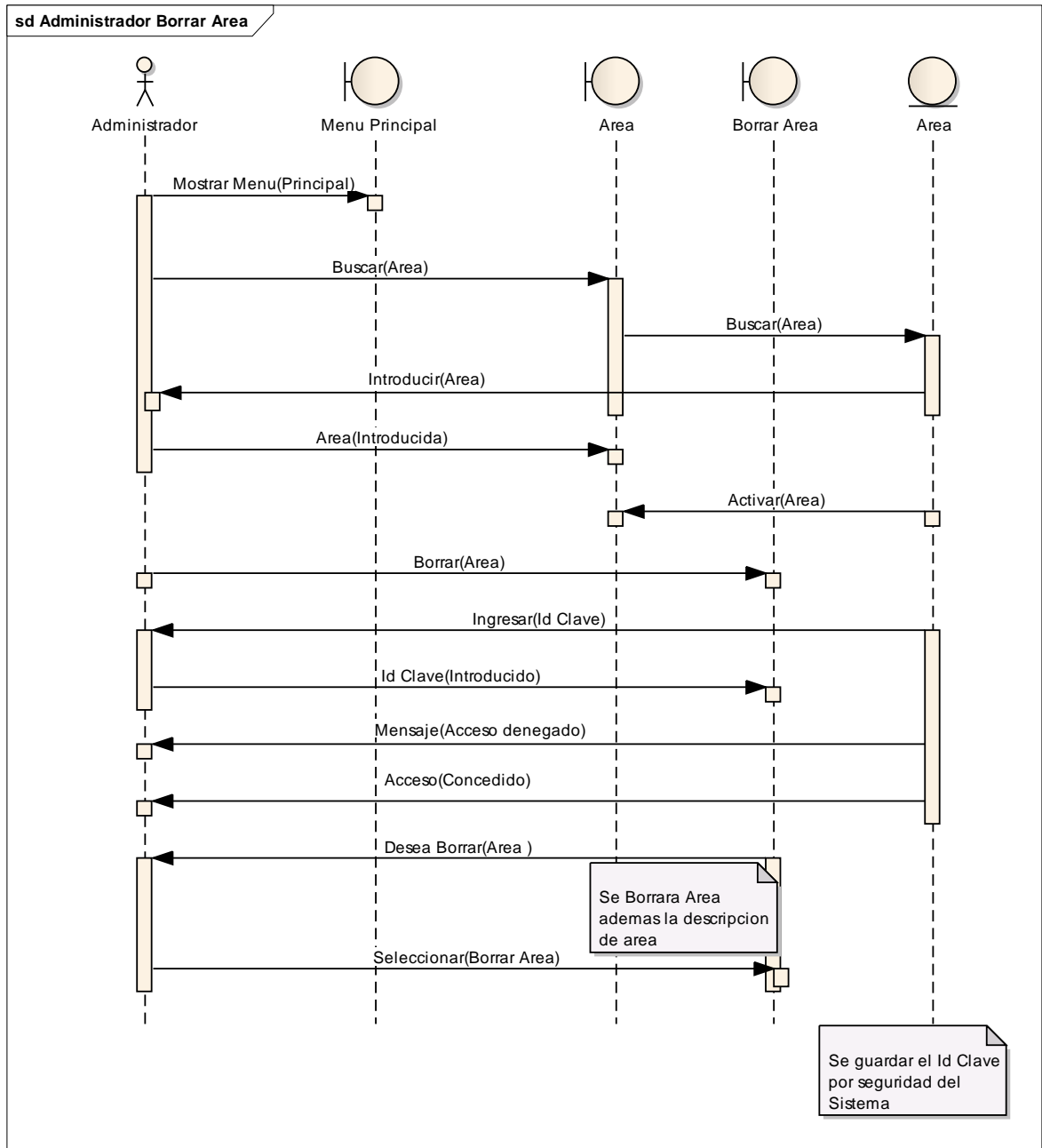


Figura 29.

II.1.5.4.30. Diagrama de Secuencia Administrador Modificar Lugar

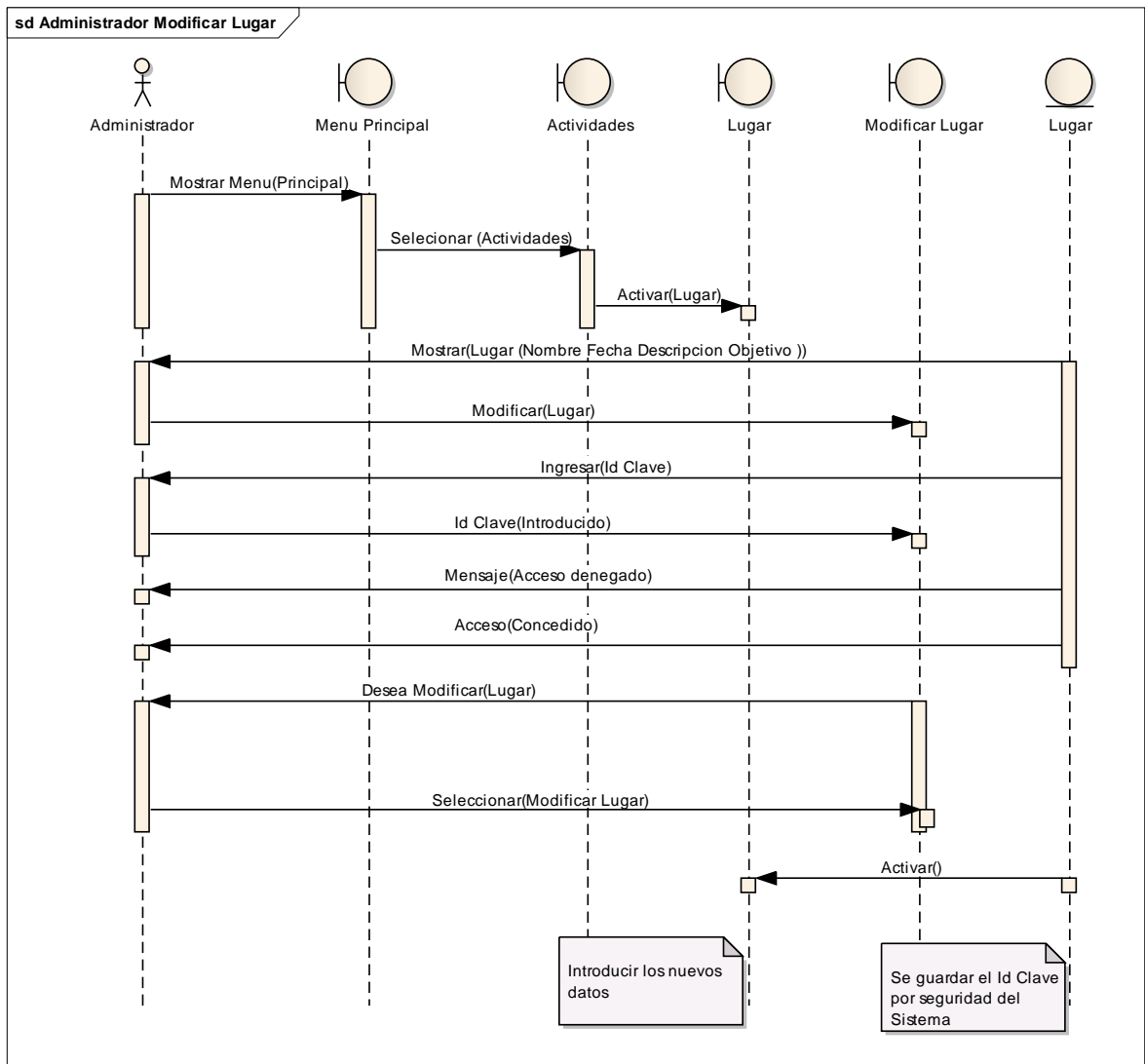


Figura 30.

II.1.5.4.31. Diagrama de Secuencia Administrador Borrar Noticia

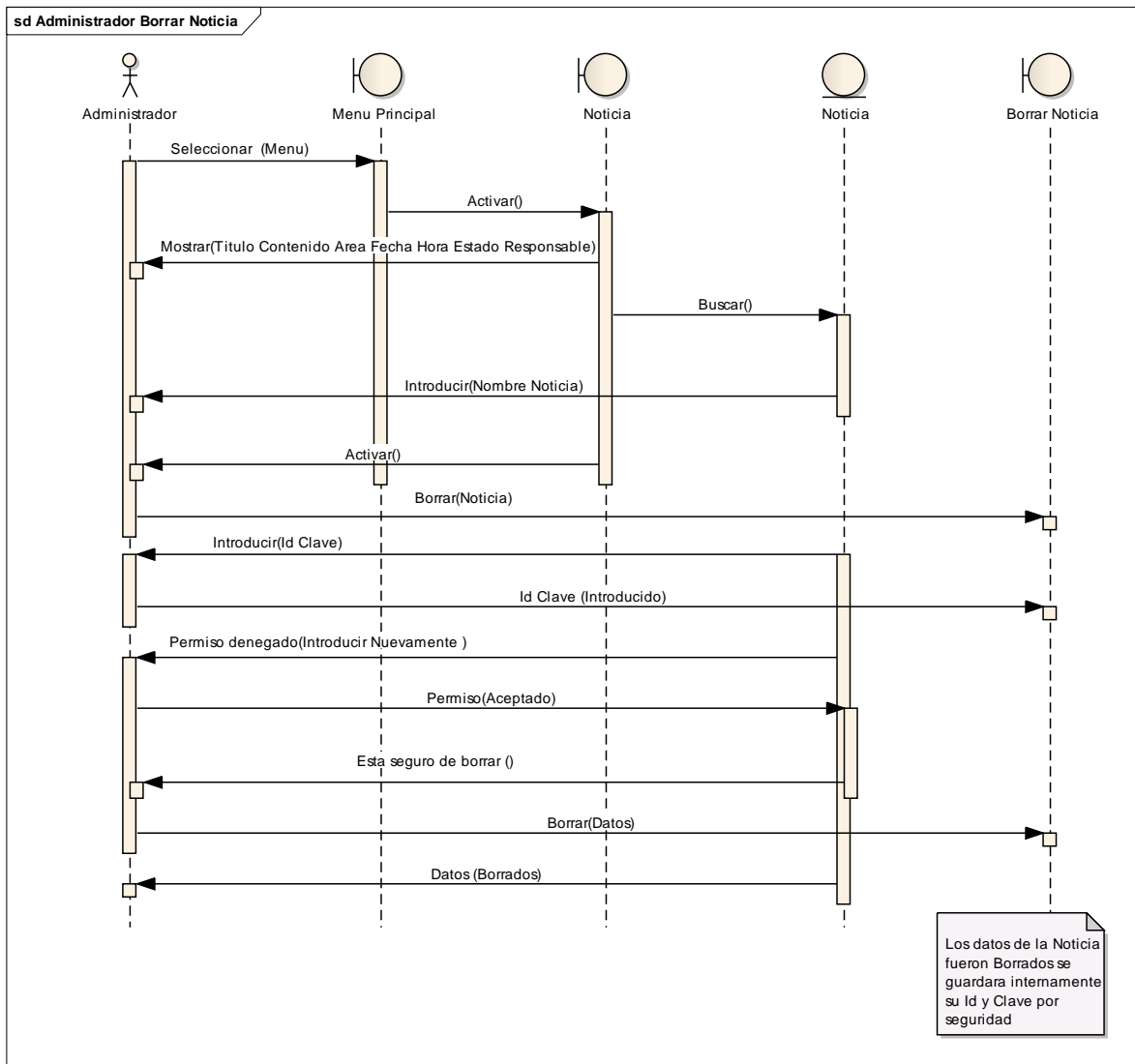


Figura 31.

II.1.5.4.32. Diagrama de Secuencia Administrador Modificar Noticia

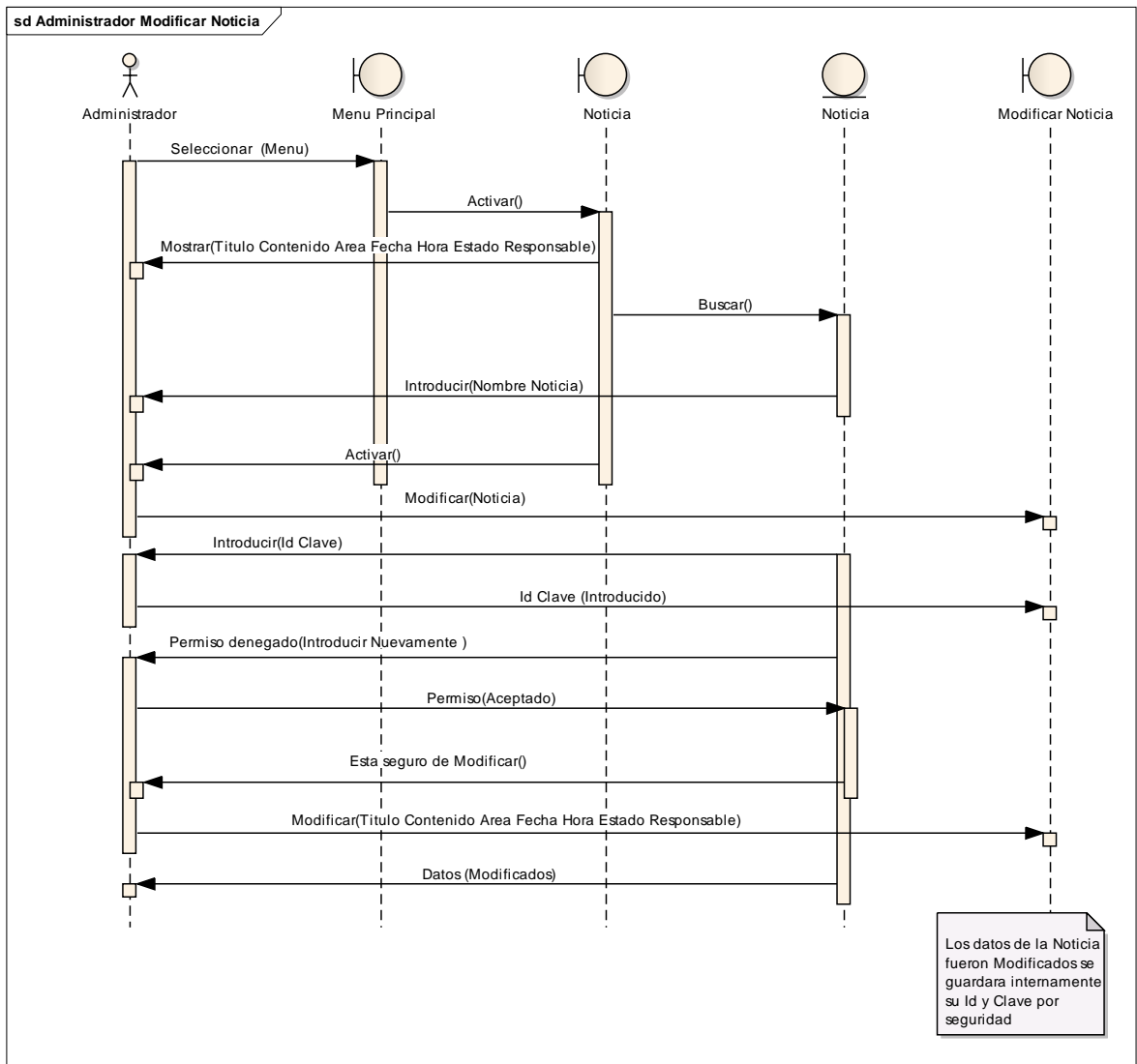


Figura 32.

II.1.5.4.33. Diagrama de Secuencia Administrador Adicionar Noticia

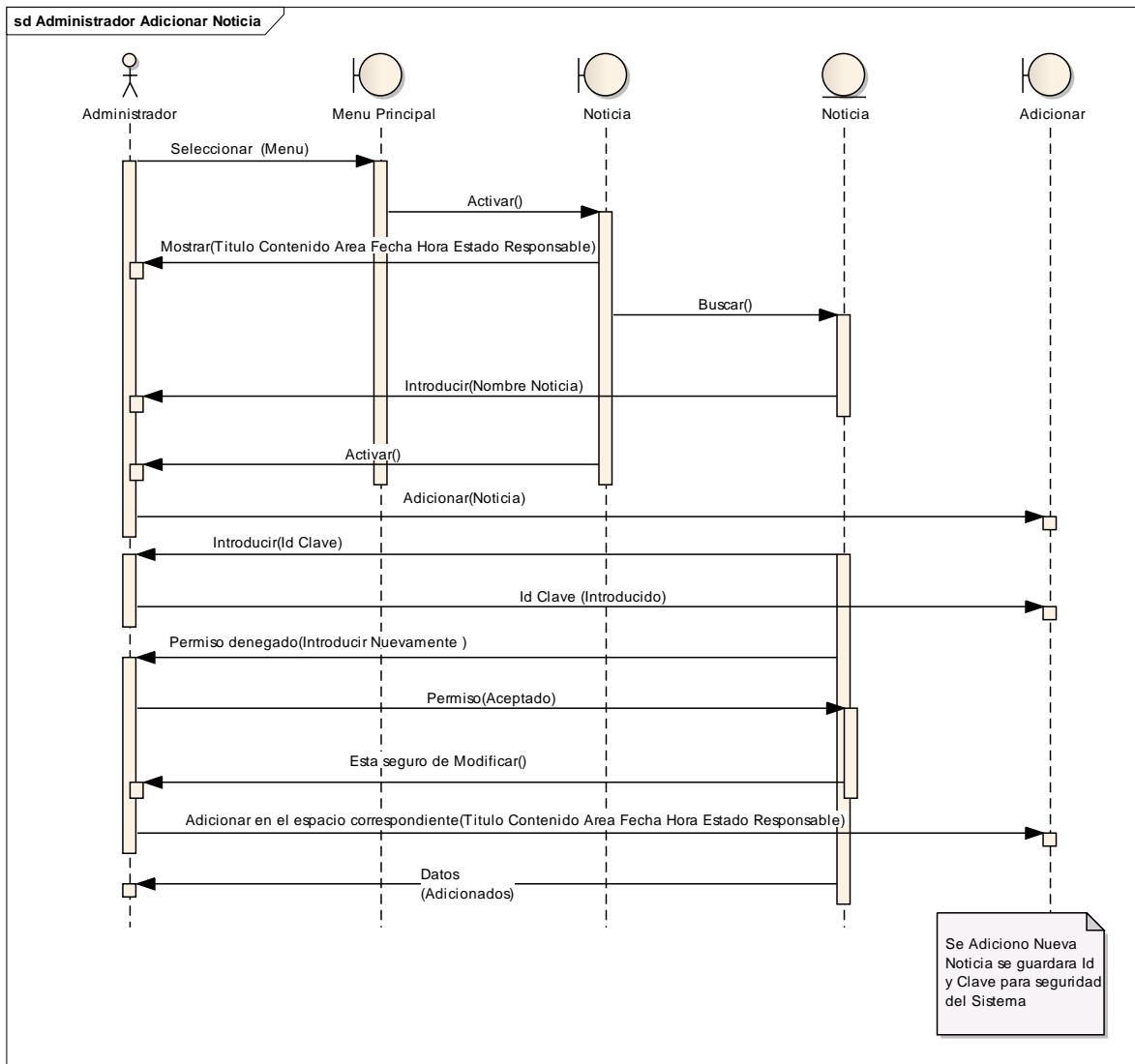


Figura 33.

II.1.5.4.34. Diagrama de Secuencia Administrador Adicionar Delito

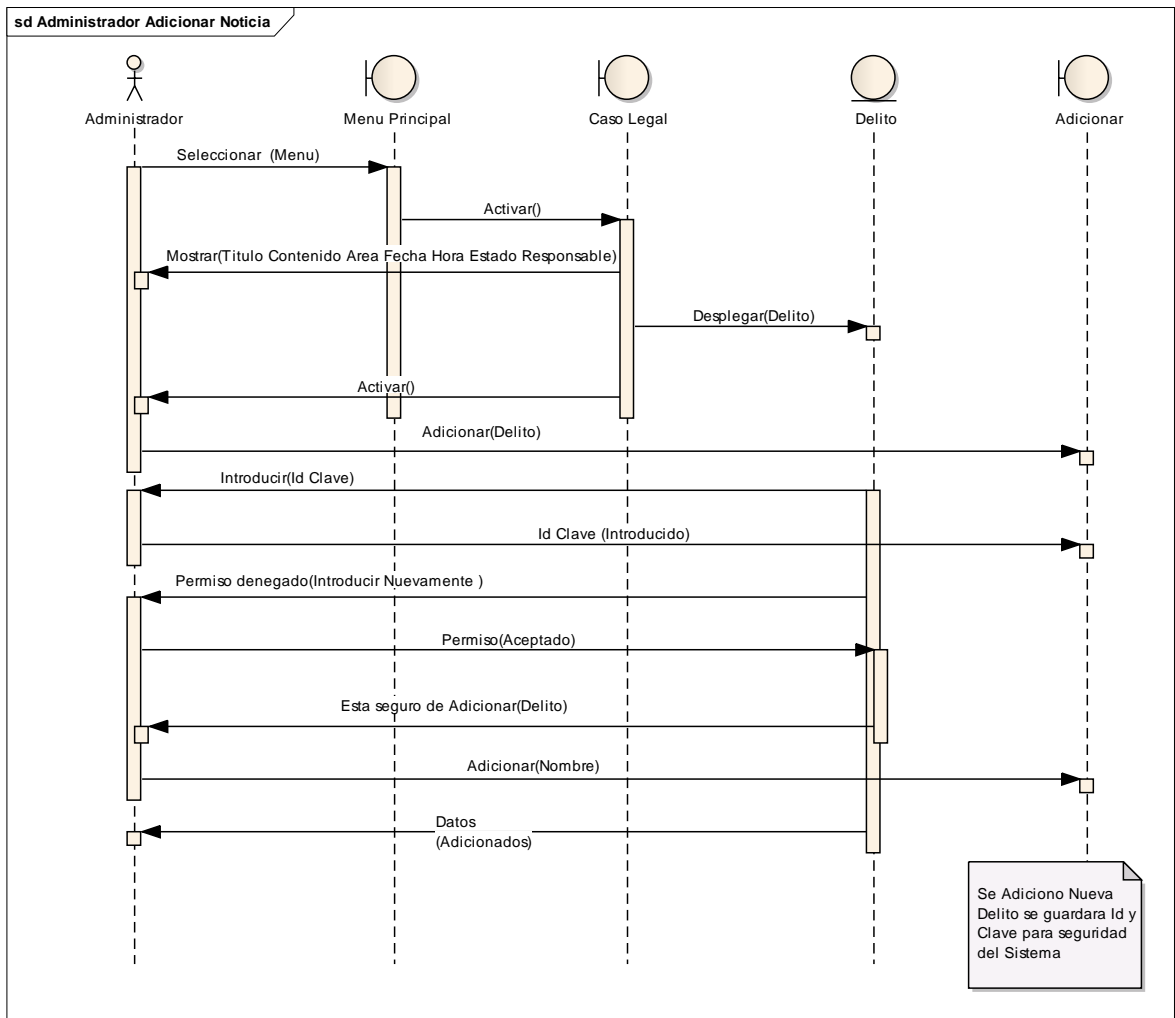


Figura 34.

II.1.5.4.35. Diagrama de Secuencia Administrador Adicionar Delito

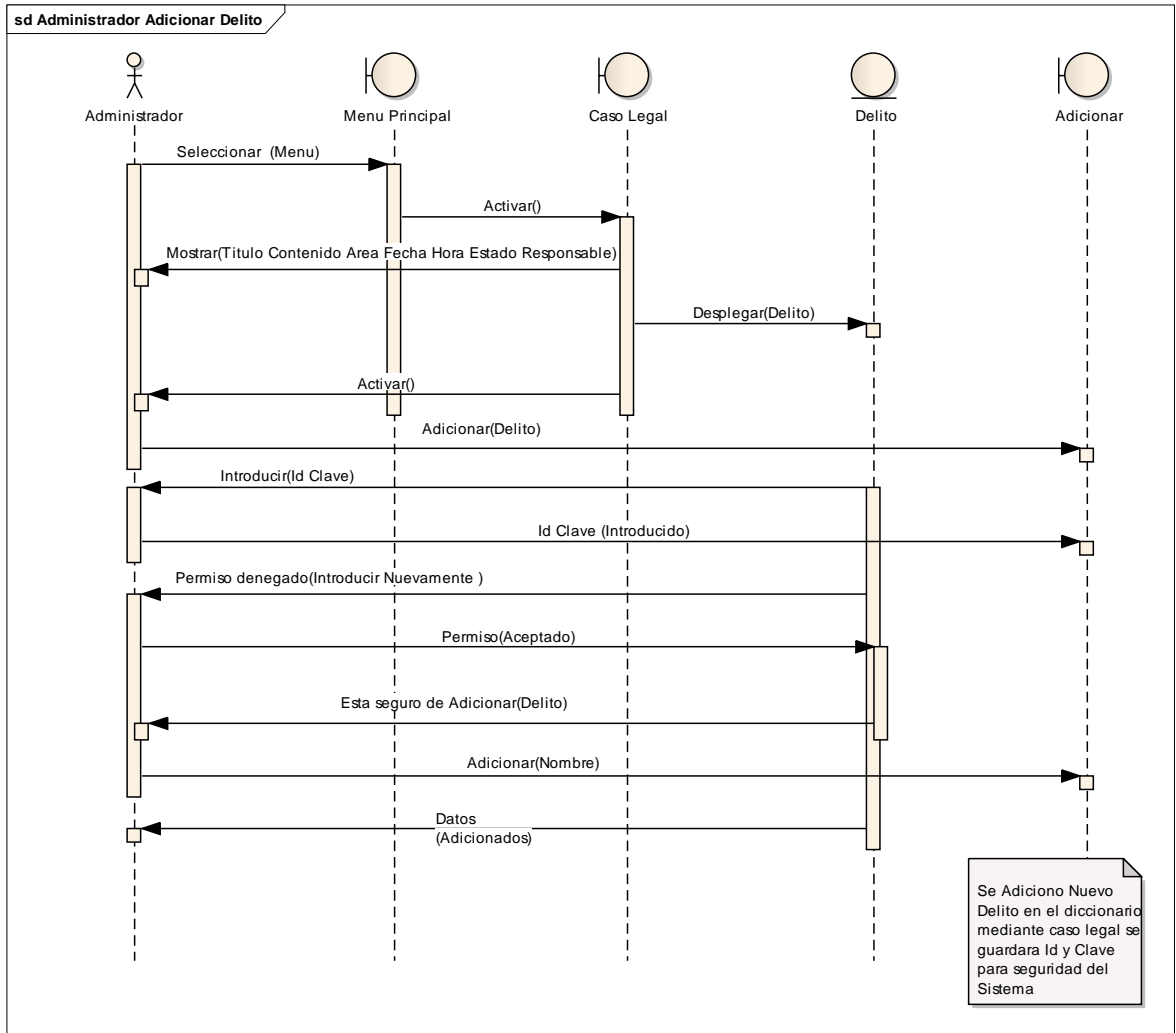


Figura 35.

II.1.5.4.36. Diagrama de Secuencia Administrador Borrar Comentario

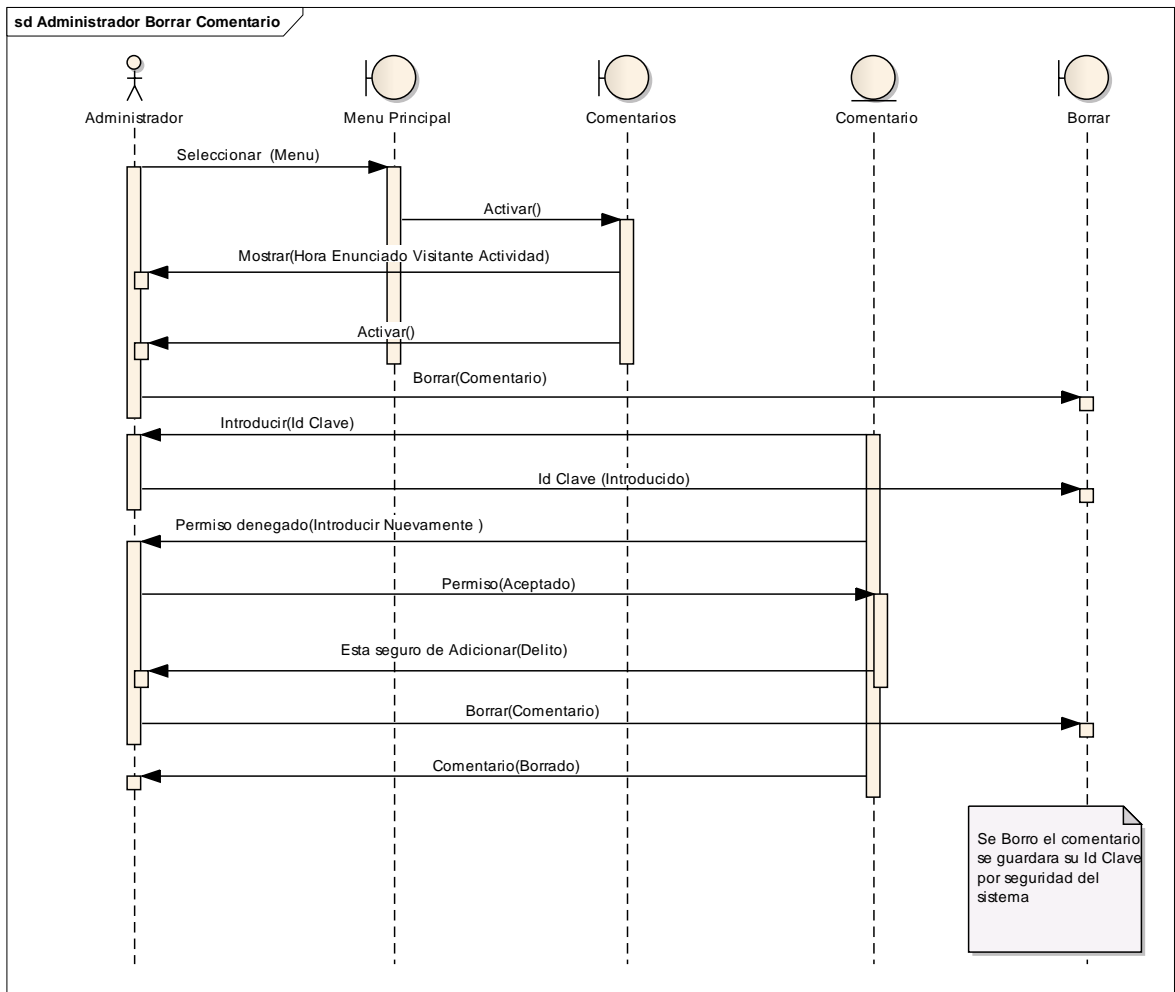
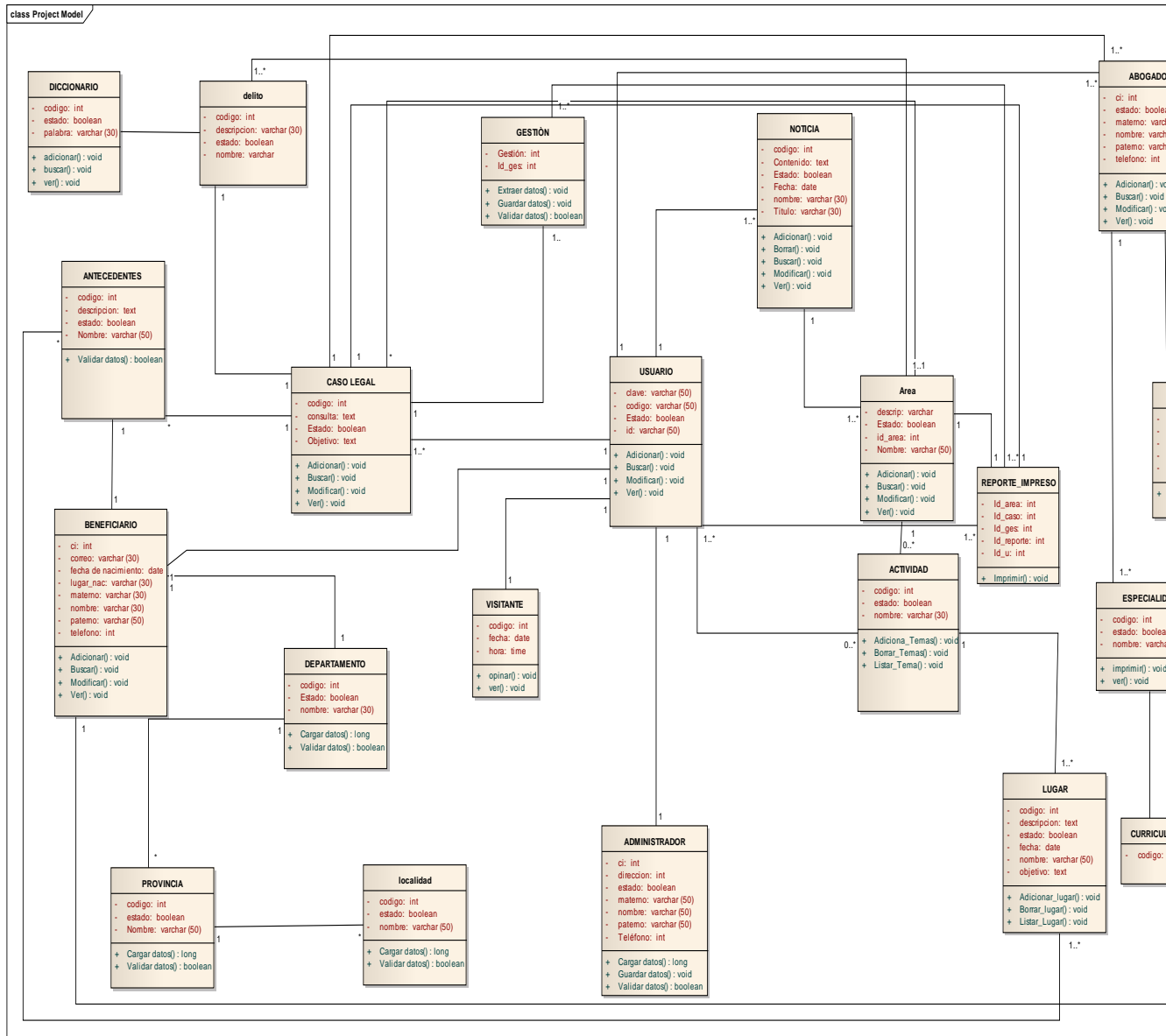


Figura 36.

II.1.5.5. Diseño de la base de datos del Sistema.

II.1.5.5.1. Diagrama de Clases.



II.1.5.5.3. Diseño de la base de datos Programacion SQL.

```
create table usuario(  
codigo serial not null primary key,  
id varchar(20) not null,  
clave varchar(20) not null,  
estado boolean default true  
);
```

```
create table departamento(  
codigo serial not null primary key,  
nombre varchar(15) not null,  
estado boolean default true  
);
```

```
create table provincia(  
codigo serial not null primary key,  
nombre varchar(15) not null,  
estado boolean default true,  
dpto integer not null,  
foreign key (dpto) references departamento(codigo)  
);
```

```
create table localidad(  
codigo serial not null primary key,  
nombre varchar(25) not null,  
estado boolean default true,  
prov integer not null,  
foreign key (prov) references provincia(codigo)  
;
```

```
create table direccion(  
codigo serial not null primary key,  
barrio varchar(30),  
calle varchar(30),  
numero integer,  
estado boolean default true,  
lugar integer not null,  
foreign key (lugar) references localidad(codigo)  
);
```

```
create table familiar(  
ci integer not null primary key,  
nombre varchar(30) not null,  
paterno varchar(30) not null,  
materno varchar(30),  
telefono integer,  
correo varchar(30),  
direc integer not null,  
foreign key (direc) references direccion(codigo)  
);
```

```
create table area(  
codigo serial not null primary key,  
nombre varchar(30) not null,  
descripcion varchar(30),  
estado boolean default true
```

);

```
create table delito(  
codigo serial not null primary key,  
nombre varchar(30) not null,  
descripcion varchar(50),  
estado boolean default true,  
area integer not null,  
foreign key (area) references area(codigo)  
);
```

```
create table diccionario(  
codigo serial not null primary key,  
palabra varchar(30) not null,  
estado boolean default true,  
delito integer not null,  
foreign key (delito) references delito(codigo)  
);
```

```
create table visitante(  
codigo serial not null primary key,  
fecha date not null,  
hora time not null  
);
```

```
create table profesion(  
codigo serial not null primary key,  
nombre varchar(30) not null,  
estado boolean default true  
);
```

```
create table especialidad(  
codigo serial not null primary key,  
nombre varchar(30) not null,  
estado boolean default true  
);
```

```
create table curriculum(  
codigo serial not null primary key,  
profesion integer not null,  
especialidad integer not null,  
foreign key (profesion) references profesion(codigo),  
foreign key (especialidad) references especialidad(codigo)  
);
```

```
create table abogado(  
ci integer not null primary key,  
nombre varchar(30) not null,  
paterno varchar(30) not null,  
materno varchar(30) not null,  
telefono integer,  
estado boolean default true,  
direc integer not null,  
curriculum integer not null,  
usuario integer not null,
```

```
foreign key (direc) references direccion(codigo),
foreign key (curriculum) references curriculum(codigo),
foreign key (usuario) references usuario(codigo)
);
```

```
create table caso_legal(
codigo serial not null primary key,
consulta varchar(50) not null,
delito integer not null,
abogado integer not null,
visitante integer not null,
foreign key (delito) references delito(codigo),
foreign key (abogado) references abogado(ci),
foreign key (visitante) references visitante(codigo)
);
```

```
create table beneficiario(
ci integer not null primary key,
nombre varchar(30) not null,
paterno varchar(30) not null,
materno varchar(30) not null,
telefono integer,
fecha_nac date not null,
lugar_nac integer not null,
correo varchar(50),
direc integer not null,
familiar integer not null,
usuario integer not null,
foreign key (direc) references direccion(codigo),
foreign key (familiar) references familiar(ci),
```

```
foreign key (usuario) references usuario(codigo)
);
```

```
create table antecedentes(
codigo serial not null primary key,
descr varchar(30) not null,
estado boolean default true,
caso integer not null,
benef integer not null,
lugar integer not null,
foreign key (caso) references caso_legal(codigo),
foreign key (benef) references beneficiario(ci),
foreign key (lugar) references localidad(codigo)
);
```

```
create table administrador(
ci integer not null primary key,
nombre varchar(25) not null,
paterno varchar(25) not null,
materno varchar(25),
telefono integer,
estado boolean default true,
direccion integer not null,
usuario integer not null,
foreign key (direccion) references direccion(codigo),
foreign key (usuario) references usuario(codigo)
);
```

```
create table lugar(  
codigo serial not null primary key,  
nombre varchar(30) not null,  
fecha date not null,  
descr varchar(50),  
objetivo varchar(50) not null,  
estado boolean default true  
);
```

```
create table actividad(  
codigo serial not null primary key,  
nombre varchar(25) not null,  
estado boolean default true,  
area integer not null,  
lugar integer not null,  
adm integer not null,  
foreign key (area) references area(codigo),  
foreign key (lugar) references lugar(codigo),  
foreign key (adm) references administrador(ci)  
);
```

```
create table contenido(  
codigo serial not null primary key,  
nombre varchar(30) not null,  
descr varchar(50),  
estado boolean default true,  
actividad integer not null,  
foreign key (actividad) references actividad(codigo)
```

```
);  
  
create table comentario(  
codigo serial not null primary key,  
hora time not null,  
enunciado varchar(50) not null,  
estado boolean default true,  
visitante integer not null,  
actividad integer not null,  
foreign key (visitante) references visitante(codigo),  
foreign key (actividad) references actividad(codigo)  
);
```

```
create table noticia(  
codigo serial not null primary key,  
titulo varchar(25) not null,  
contenido varchar(50) not null,  
area varchar(25) not null,  
fecha date not null,  
hora time not null,  
estado boolean default true,  
adm integer not null,  
foreign key (adm) references administrador(ci)  
);
```

I.6. Interfaces Externas

I.6.1. Programación del Sistema

ADMINISTRADOR

```
package administrador;
```

```
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import org.apache.velocity.Template;  
import org.apache.velocity.app.Velocity;  
import org.apache.velocity.context.Context;  
import org.apache.velocity.servlet.VelocityServlet;
```

```
import java.util.Map;  
import java.util.HashMap;  
import java.util.ArrayList;
```

```
import java.sql.ResultSet;  
import java.sql.SQLException;
```

```
import base.Postgres;
```

```
public class Abogados extends VelocityServlet {  
    private static final long serialVersionUID = 1L;  
  
    public Template handleRequest(HttpServletRequest request,  
        HttpServletResponse response, Context context) {  
        Template template = null;  
        ResultSet datos = null;  
        ArrayList lista = new ArrayList();  
        Map fila = null;  
        try {  
            String sql = "select a.ci, a.nombre, a.paterno, a.materno,  
a.convenio from abogado a " +  
                "where estado='true'";  
            Postgres db = new Postgres();  
            db.conectar();  
            datos = db.seleccionar(sql);  
            while( datos.next() ){  
                int ci = datos.getInt(1);  
                String nombre = datos.getString(2);
```

```

        String paterno = datos.getString(3);
        String materno = datos.getString(4);
        boolean convenio = datos.getBoolean(5);
        String conv = "NO";
        String ong = "SI";
        if( convenio ){
            conv = "SI";
            ong = "NO";
        }
        fila = new HashMap();
        fila.put("ci", ci);
        fila.put("nombre", nombre);
        fila.put("paterno", paterno);
        fila.put("materno", materno);
        fila.put("curriculum", ci);
        fila.put("convenio", conv);
        fila.put("ong", ong);

        lista.add(fila);
    }
    context.put("abogados",lista);
    template =
Velocity.getTemplate("plantillas/administrador/abogados.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Abogados: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Abogados: " + e.getMessage());
    }
    return template;
}

}

package administrador;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

```

```

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

public class Actividades extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            String sql = "select act.codigo, act.nombre, act.estado,
a.nombre, l.nombre, " +
                "p.nombre, d.nombre, adm.nombre,
adm.paterno, adm.materno, act.fecha " +
                "from actividad act, administrador adm, area a,
localidad l, " +
                "provincia p, departamento d " +
                "where act.adm=adm.ci and act.area=a.codigo
and " +
                "act.lugar=l.codigo and l.prov=p.codigo and
p.dpto=d.codigo";

            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            while( datos.next() ){
                int codigo = datos.getInt(1);
                String actividad = datos.getString(2);
                boolean est = datos.getBoolean(3);
                String estado = "Inactivo";
                if( est ){
                    estado = "Activo";
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```

        }
        String area = datos.getString(4);
        String loc = datos.getString(5);
        String prov = datos.getString(6);
        String dpto = datos.getString(7);
        String lugar = dpto + " - " + prov + " - " + loc;
        String adm = datos.getString(8)+"
"+datos.getString(9)+" "+datos.getString(10);
        String fecha = datos.getString(11);
        fila = new HashMap();
        fila.put("codigo", codigo);
        fila.put("nombre", actividad);
        fila.put("fecha", fecha);
        fila.put("area", area);
        fila.put("lugar", lugar);
        fila.put("adm", adm);
        //fila.put("mostrar", "");

        lista.add(fila);
    }
    context.put("actividades",lista);
    template =
Velocity.getTemplate("plantillas/administrador/actividades.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Actividades: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Actividades: " + e.getMessage());
    }
    return template;
}

}

package administrador;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;

```

```

import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

public class AddAbogado extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public int ci;
    public String nombre;
    public String paterno;
    public String materno;
    public int telef;
    public boolean convenio;

    public String barrio;
    public String calle;
    public int numero;
    public int dpto;
    public int prov;
    public int local;

    public int prof;
    public int esp;

    public String cuenta;
    public String clave;

    public AddAbogado(){
        System.out.println("creando abogado");
        nombre = "";
        paterno = "";
        materno = "";
        barrio = "";
        calle = "";
        cuenta = "";
        clave = "";
    }
}

```

```

public Template handleRequest(HttpServletRequest request,
HttpServletRequest response, Context context) {
    Template template = null;
    int modo = Integer.parseInt(request.getParameter("modo"));

    try {
        String html = "";
        if( modo==1 ){
            ci = 0;
            nombre = "";
            paterno = "";
            materno = "";
            telef = 0;
            barrio = "";
            calle = "";
            numero = 0;
            dpto = 0;
            prov = 0;
            local = 0;
            prof = 0;
            esp = 0;
            cuenta = "";
            clave = "";
            html =
"plantillas/administrador/adicionarAbogado.html";
        } else if( modo==2 ) {
            context.put("dptos", getDepartamentos());
            html =
"plantillas/administrador/adicionarDireccion.html";
        } else if( modo==3 ) {
            context.put("profs", getProfesiones());
            context.put("esps", getEspecialidades());
            html =
"plantillas/administrador/adicionarCurriculum.html";
        } else if( modo==4 ) {
            html =
"plantillas/administrador/adicionarUsuario.html";
        } else if( modo==5 ) {
            ci = Integer.parseInt(request.getParameter("ci"));
            nombre = request.getParameter("nombre");
            paterno = request.getParameter("paterno");

```

```

        materno = request.getParameter("materno");
        telef =
Integer.parseInt(request.getParameter("telef"));
        convenio =
Boolean.parseBoolean(request.getParameter("convenio"));

        if( ci==0 ){
            html =
"plantillas/administrador/adicionarAbogado.html";
        } else if( nombre.equals("") ){
            html =
"plantillas/administrador/adicionarAbogado.html";
        } else if( paterno.equals("") ){
            html =
"plantillas/administrador/adicionarAbogado.html";
        } else if( telef==0 ){
            html =
"plantillas/administrador/adicionarAbogado.html";
        } else if( barrio.equals("") ){
            context.put("dptos", getDepartamentos());
            html =
"plantillas/administrador/adicionarDireccion.html";
        } else if( calle.equals("") ){
            context.put("dptos", getDepartamentos());
            html =
"plantillas/administrador/adicionarDireccion.html";
        } else if( numero==0 ){
            context.put("dptos", getDepartamentos());
            html =
"plantillas/administrador/adicionarDireccion.html";
        } else if( dpto==0 ){
            context.put("dptos", getDepartamentos());
            html =
"plantillas/administrador/adicionarDireccion.html";
        } else if( prov==0 ){
            context.put("dptos", getDepartamentos());
            html =
"plantillas/administrador/adicionarDireccion.html";
        } else if( local==0 ){
            context.put("dptos", getDepartamentos());
            html =
"plantillas/administrador/adicionarDireccion.html";

```

```

        } else if( prof==0 ){
            context.put("profs", getProfesiones());
            context.put("esps", getEspecialidades());
            html =
"plantillas/administrador/adicionarCurriculum.html";
        } else if( esp==0 ){
            context.put("profs", getProfesiones());
            context.put("esps", getEspecialidades());
            html =
"plantillas/administrador/adicionarCurriculum.html";
        } else if( cuenta.equals("") ){
            html =
"plantillas/administrador/adicionarUsuario.html";
        } else if( clave.equals("") ){
            html =
"plantillas/administrador/adicionarUsuario.html";
        } else{
            System.out.println("adicionando abogado");
            insertar();
            response.sendRedirect("AdmAbogados");
            return null;
        }
    } else if( modo==6 ) {
        // adicionar direccion
        barrio = request.getParameter("barrio");
        calle = request.getParameter("calle");
        numero =
Integer.parseInt(request.getParameter("numero"));
        dpto =
Integer.parseInt(request.getParameter("dpto"));
        prov = Integer.parseInt(request.getParameter("prov"));
        local =
Integer.parseInt(request.getParameter("local"));
        html =
"plantillas/administrador/adicionarAbogado.html";
    } else if( modo==7 ) {
        // adicionar curriculum
        prof = Integer.parseInt(request.getParameter("prof"));
        esp = Integer.parseInt(request.getParameter("esp"));
        html =
"plantillas/administrador/adicionarAbogado.html";
    } else if( modo==8 ) {

```

```

        cuenta = request.getParameter("cuenta");
        clave = request.getParameter("clave");
        html =
"plantillas/administrador/adicionarAbogado.html";
        } else {
            response.sendRedirect("AdmAbogados");
            return null;
        }
        template = Velocity.getTemplate(html);
    }
    catch (SQLException e) {
        System.err.println("Error: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error: " + e.getMessage());
    }
    return template;
}

private void insertar(){
    String sql = "insert into direccion(barrio, calle, numero, localidad) " +
    "values('"+barrio+"', '"+calle+"', '"+numero+"', '"+local+"');";
    Postgres db = new Postgres();
    db.conectar();
    db.insertar(sql);
    sql = "insert into usuario(id, clave) values('"+cuenta+"', '"+clave+"');";
    db.insertar(sql);
    sql = "insert into curriculum(profesion, especialidad) values('"+prof+",
"+esp+"');";
    db.insertar(sql);
    int direc = getCodigo("direccion");
    int curri = getCodigo("curriculum");
    int usuario = getCodigo("usuario");
    sql = "insert into
abogado(ci,nombre,paterno,materno,telefono,direc,curriculum,usuario,convenio) "
+
    "values('"+ci+"', '"+nombre+"', '"+paterno+"', '"+materno+"', '"+telef+", "+direc+",
"+curri+", "+usuario+", '"+convenio+"');";
    db.insertar(sql);
    db.cerrar();
}

```

```

    }

    public int getCodigo( String tabla ) {
        int codigo = 0;
        try {
            String sql = "select max(codigo) from "+tabla;
            Postgres db = new Postgres();
            db.conectar();
            ResultSet datos = db.seleccionar(sql);
            if( datos.next() ){
                codigo = datos.getInt(1);
            }
        }
        catch (SQLException e) {
            System.err.println("Error en AddAbogado: " +
e.getMessage());
        }
        return codigo;
    }

    public ArrayList<Map> getDepartamentos() {
        ArrayList<Map> lista = new ArrayList<Map>();
        Map fila = null;
        try {
            String sql = "select codigo, nombre from departamento where
estado='true' order by codigo;";
            Postgres db = new Postgres();
            db.conectar();
            ResultSet datos = db.seleccionar(sql);
            while( datos.next() ){
                int codigo = datos.getInt(1);
                String nombre = datos.getString(2);
                fila = new HashMap();
                fila.put("codigo", codigo);
                fila.put("nombre", nombre);
                lista.add(fila);
            }
        }
        catch (SQLException e) {
            System.err.println("Error en AddAbogado: " +
e.getMessage());
        }
    }

```

```

        catch (Exception e) {
            System.err.println("Error en AddAbogado: " +
e.getMessage());
        }
        return lista;
    }

    public ArrayList<Map> getProfesiones() {
        ArrayList<Map> lista = new ArrayList<Map>();
        Map fila = null;
        try {
            String sql = "select codigo, nombre from profesion where
estado='true' order by codigo;";
            Postgres db = new Postgres();
            db.conectar();
            ResultSet datos = db.seleccionar(sql);
            while( datos.next() ){
                int codigo = datos.getInt(1);
                String nombre = datos.getString(2);
                fila = new HashMap();
                fila.put("codigo", codigo);
                fila.put("nombre", nombre);
                lista.add(fila);
            }
        }
        catch (SQLException e) {
            System.err.println("Error en AddAbogado: " +
e.getMessage());
        }
        catch (Exception e) {
            System.err.println("Error en AddAbogado: " +
e.getMessage());
        }
        return lista;
    }

    public ArrayList<Map> getEspecialidades() {
        ArrayList<Map> lista = new ArrayList<Map>();
        Map fila = null;
        try {
            String sql = "select codigo, nombre from especialidad where
estado='true' order by codigo;";

```

```

        Postgres db = new Postgres();
        db.conectar();
        ResultSet datos = db.seleccionar(sql);
        while( datos.next() ){
            int codigo = datos.getInt(1);
            String nombre = datos.getString(2);
            fila = new HashMap();
            fila.put("codigo", codigo);
            fila.put("nombre", nombre);
            lista.add(fila);
        }
    }
    catch (SQLException e) {
        System.err.println("Error en AddAbogado: " +
e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en AddAbogado: " +
e.getMessage());
    }
    return lista;
}

}
package administrador;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

public class AddArea extends VelocityServlet {

    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;

```

```

        int modo = Integer.parseInt(request.getParameter("modo"));
        String vm = "";
        try{
            if( modo==1 ){
                vm = "plantillas/administrador/adicionarArea.html";
            } else if( modo==2 ){
                String nombre = request.getParameter("nombre");
                String descr = request.getParameter("descr");
                insertar(nombre, descr);
                response.sendRedirect("AdmArea");
                return null;
            }
            template = Velocity.getTemplate(vm);
        } catch( Exception e){
            System.err.println("Error en AddArea: " + e.getMessage());
        }
        return template;
    }

    private void insertar(String nombre, String descr){
        String sql = "insert into area(nombre, descripcion)
values("+nombre+", "+descr+");";
        Postgres db = new Postgres();
        db.conectar();
        db.insertar(sql);
        db.cerrar();
    }

}

package administrador;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

```

```

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Administrador extends VelocityServlet {

    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        Map usuario = null;
        try {
            HttpSession sesion = request.getSession(true);
            int ci = (Integer)sesion.getAttribute("ci");
            String sql = "select nombre, paterno, materno from
administrador where ci="+ci;
            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            while( datos.next() ){
                String nombre = datos.getString(1);
                String paterno = datos.getString(2);
                String materno = datos.getString(3);
                usuario = new HashMap();
                usuario.put("nombre", nombre);
                usuario.put("paterno", paterno);
                usuario.put("materno", materno);
            }
            context.put("usuario", usuario);
            template =
Velocity.getTemplate("plantillas/administrador/index.html");

            response.sendRedirect("plantillas/administrador/index.html");
        } catch (SQLException e) {

```

```

        System.err.println("Error en Administrador: " +
e.getMessage());
    } catch (Exception e) {
        System.err.println("Error en Administrador: " +
e.getMessage());
    }
    //return template;
    return null;
}

}

package administrador;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

public class Area extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            String sql = "select codigo, nombre, descripcion from area
where estado='true'";
            Postgres db = new Postgres();

```

```

        db.conectar();
        datos = db.seleccionar(sql);
        while( datos.next() ){
            int codigo = datos.getInt(1);
            String nombre = datos.getString(2);
            String descr = datos.getString(3);
            fila = new HashMap();
            fila.put("codigo", codigo);
            fila.put("nombre", nombre);
            fila.put("desc", descr);
            lista.add(fila);
        }
        context.put("areas",lista);
        template =
Velocity.getTemplate("plantillas/administrador/areas.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Area: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Area: " + e.getMessage());
    }
    return template;
}

}

package administrador;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

```

```

public class Contenido extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            int act = Integer.parseInt(request.getParameter("actividad"));
            String sql = "select nombre, descr, estado from contenido
where actividad="+act;
            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            if( datos.next() ){
                String nombre = datos.getString(1);
                String descr = datos.getString(2);
                boolean est = datos.getBoolean(3);
                String estado = "Inactivo";
                if( est ){
                    estado = "Activo";
                }
                context.put("nombre",nombre);
                context.put("desc",descr);
                context.put("estado",estado);
            }
            template =
Velocity.getTemplate("plantillas/administrador/contenidos.html");
        }
        catch (SQLException e) {
            System.err.println("Error en Contenido: " + e.getMessage());
        }
        catch (Exception e) {
            System.err.println("Error en Contenido: " + e.getMessage());
        }
        return template;
    }
}

```

```

package administrador;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
/*
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;
*/
import java.io.PrintWriter;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

import org.apache.velocity.app.Velocity;

import base.Postgres;

public class Localidades extends HttpServlet{

    public void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/xml");
        int prov = Integer.parseInt(request.getParameter("prov"));
        PrintWriter out = response.getWriter();
        try {
            String sql = "select codigo, nombre from localidad where
estado='true' and prov=" + prov;
            Postgres db = new Postgres();
            db.conectar();
            ResultSet datos = db.seleccionar(sql);
            StringBuffer xml = new StringBuffer("\r\n<localidades>");
            while( datos.next() ){
                int codigo = datos.getInt(1);

```

```

        String nombre = datos.getString(2);
        xml.append("\r\n<localidad codigo=" + Integer.toBinaryString(codigo) + ">");
        xml.append(nombre);
        xml.append("\r\n</localidad>");
    }
    xml.append("\r\n</localidades>");
    db.cerrar();
    out.write(xml.toString());
    out.flush();
    out.close();
} catch (SQLException e) {
    System.err.println("Error en Validar: " + e.getMessage());
}
}
public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
    doPost(request, response);
}
}
package administrador;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
/*
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;
*/
import java.io.PrintWriter;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

```

```

import org.apache.velocity.app.Velocity;

import base.Postgres;

public class Provincias extends HttpServlet{

    public void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/xml");
        int dpto = Integer.parseInt(request.getParameter("dpto"));
        PrintWriter out = response.getWriter();
        try {
            String sql = "select codigo, nombre from provincia where
estado='true' and dpto=" + dpto;
            Postgres db = new Postgres();
            db.conectar();
            ResultSet datos = db.seleccionar(sql);
            StringBuffer xml = new StringBuffer("\r\n<provincias>");
            while( datos.next() ){
                int codigo = datos.getInt(1);
                String nombre = datos.getString(2);
                xml.append("\r\n<provincia codigo=" +
Integer.toBinaryString(codigo) + ">");
                xml.append(nombre);
                xml.append("\r\n</provincia>");
            }
            xml.append("\r\n</provincias>");
            db.cerrar();
            out.write(xml.toString());
            out.flush();
            out.close();
        } catch (SQLException e) {
            System.err.println("Error en Validar: " + e.getMessage());
        }
    }

    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        doPost(request, response);
    }
}

```

```

BASE
package base;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Postgres {

    private Connection conexion=null;
    private Statement consultas=null;
    private String controlador=null;
    private String base=null;
    private String usuario=null;
    private String clave=null;

    // Busca el controlador(el archivo jar de postgres)
    public Postgres(){
        this.controlador="org.postgresql.Driver";
        this.base="jdbc:postgresql://127.0.0.1:5432/mundomejor";
        this.usuario="postgres";
        this.clave="master";

        try {
            Class.forName(controlador);
        }
        catch (ClassNotFoundException cnf) {
            System.out.println("Driver(controlador) no encontrado.");
            System.exit(1);
        }
    }

    // se conecta con la base de datos
    public void conectar() {
        try {
            conexion=DriverManager.getConnection(base,usuario,clave);
            consultas=conexion.createStatement();

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```

}

public ResultSet seleccionar(String consulta) {
    ResultSet tabla=null;
    try {
        tabla=consultas.executeQuery(consulta);
    }
    catch (SQLException e) {
        e.printStackTrace();
    }
    return tabla;
}

public boolean addBatch(String sql){
    try{
        consultas.addBatch(sql);
        return true;
    }
    catch(SQLException se){
        System.err.println("Se produjo un error al agregar una consulta.");
        return false;
    }
}

public void limpiarBatch(){
    try{
        consultas.clearBatch();
    }
    catch(SQLException se){
        System.err.println("Se produjo un error al limpiar el batch.");
    }
}

public void ejecutarBatch(){
    try{
        int[] resp=consultas.executeBatch();
    }
    catch(SQLException se){
        System.err.println("Se produjo un error al ejecutar el batch.\n"+se.toString());
    }
}

```

```

public void insertar(String consulta){
    ejecutar(consulta);
}

public void modificar(String consulta){
    ejecutar(consulta);
}

public void eliminar(String consulta){
    ejecutar(consulta);
}

private void ejecutar(String consulta){
    try {
        consultas.executeUpdate(consulta);
    }
    catch (SQLException e) {
        e.printStackTrace();
    }
}

public void cerrar() {
    try {
        if (consultas != null) {
            consultas.close();
        }
        if (conexion != null) {
            conexion.close();
        }
    }
    catch (SQLException e){
        e.printStackTrace();
    }
}

}
BENEFICIARIO
package beneficiario;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

public class Abogados extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            String sql = "select a.ci, a.nombre, a.paterno, a.materno,
a.convenio from abogados a " +
                "where estado='true'";
            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            while( datos.next() ){
                int ci = datos.getInt(1);
                String nombre = datos.getString(2);
                String paterno = datos.getString(3);
                String materno = datos.getString(4);
                //int curriculum = datos.getInt(8);
                boolean convenio = datos.getBoolean(5);
                String conv = "NO";
                String ong = "SI";
                if( convenio ){
                    conv = "SI";
                    ong = "NO";
                }
            }
        }
    }
}

```

```

        }
        fila = new HashMap();
        fila.put("nombre", nombre);
        fila.put("paterno", paterno);
        fila.put("materno", materno);
        fila.put("curriculum", ci);
        fila.put("convenio", conv);
        fila.put("ong", ong);

        lista.add(fila);
    }
    context.put("abogados", lista);
    template =
Velocity.getTemplate("plantillas/beneficiario/abogados.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Abogados: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Abogados: " + e.getMessage());
    }
    return template;
}

}

package beneficiario;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

```

```

public class Actividades extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            String sql = "select a.codigo, a.nombre, a.estado, a.area,
a.lugar, adm.nombre, adm.paterno, adm.materno " +
                "from actividad a, administrador adm " +
                "where a.adm=adm.ci";
            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            while( datos.next() ){
                int codigo = datos.getInt(1);
                String actividad = datos.getString(2);
                boolean est = datos.getBoolean(3);
                String estado = "Inactivo";
                if( est ){
                    estado = "Activo";
                }
                int area = datos.getInt(4);
                int lugar = datos.getInt(5);
                String adm = datos.getString(6)+"
+datos.getString(7)+" "+datos.getString(8);
                fila = new HashMap();
                fila.put("codigo", codigo);
                fila.put("nombre", actividad);
                fila.put("estado", estado);
                fila.put("area", area);
                fila.put("lugar", lugar);
                fila.put("adm", adm);
                fila.put("mostrar", "");

                lista.add(fila);
            }
            context.put("actividades",lista);
            template =

```

```

Velocity.getTemplate("plantillas/beneficiario/actividades.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Actividades: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Actividades: " + e.getMessage());
    }
    return template;
}
}

package beneficiario;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

public class AdicionarBeneficiario extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
HttpServletResponse response, Context context) {
        Template template = null;
        int modo = Integer.parseInt(request.getParameter("modo"));

        try {
            String html = "";
            if( modo==1 ){
                html =
"plantillas/beneficiario/adicionarBeneficiario.html";
            } else if( modo==2 ) {
                html =

```

```

"plantillas/beneficiario/adicionarDireccion.html";
        } else if( modo==3 ) {
            html = "plantillas/beneficiario/adicionarFamiliar.html";
        } else if( modo==4 ) {
            html =
"plantillas/beneficiario/adicionarDireccion.html";
        }
        template = Velocity.getTemplate(html);
    }
    catch (SQLException e) {
        System.err.println("Error: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error: " + e.getMessage());
    }
    return template;
}
}

```

```

package beneficiario;
import java.sql.SQLException;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

public class AdicionarDireccion extends VelocityServlet {
    private static final long serialVersionUID = 1L;
    public int modo = 0;

    public Template handleRequest(HttpServletRequest request,
HttpServletResponse response, Context context) {
        Template template = null;

        try {
            System.out.println("modo: "+(++modo));
            template =

```

```

Velocity.getTemplate("plantillas/beneficiario/adicionarBeneficiario.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Actividades: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Actividades: " + e.getMessage());
    }
    return template;
}
}

package beneficiario;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

public class Area extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            String area51 = request.getParameter("area");
            String sql = "";
            if( area51 == null ){

```

```

        sql = "select nombre, descripcion from area where
estado='true'";
    }
    else{
        int area = Integer.parseInt(area51);
        sql = "select nombre, descripcion from area where
codigo="+area;
    }
    Postgres db = new Postgres();
    db.conectar();
    datos = db.seleccionar(sql);
    while( datos.next() ){
        String nombre = datos.getString(1);
        String descr = datos.getString(2);
        fila = new HashMap();
        fila.put("nombre", nombre);
        fila.put("desc", descr);
        lista.add(fila);
    }
    context.put("areas",lista);
    template =
Velocity.getTemplate("plantillas/beneficiario/areas.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Area: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Area: " + e.getMessage());
    }
    return template;
}

}

package beneficiario;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;

```

```

import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

public class Beneficiarios extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            String sql = "select * from beneficiario";
            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            while( datos.next() ){
                int ci = datos.getInt(1);
                String nombre = datos.getString(2);
                String paterno = datos.getString(3);
                String materno = datos.getString(4);
                int telf = datos.getInt(5);
                String fechaNac = datos.getString(6);
                int direc = datos.getInt(8);
                int familiar = datos.getInt(9);

                fila = new HashMap();
                fila.put("ci", ci);
                fila.put("nombre", nombre);
                fila.put("paterno", paterno);
                fila.put("materno", materno);
                fila.put("telefono", telf);
                fila.put("fechaNac", fechaNac);
                fila.put("lugarNac", "");
                fila.put("direccion", direc);
            }
        }
    }
}

```

```

        fila.put("familiar", familiar);

        lista.add(fila);
    }
    context.put("comentarios", lista);
    template =
Velocity.getTemplate("plantillas/beneficiario/beneficiarios.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Actividades: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Actividades: " + e.getMessage());
    }
    return template;
}

}

package beneficiario;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

public class CasoLegal extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;

```

```

ArrayList lista = new ArrayList();
Map fila = null;
try {
    int visitante = 0;
    do{
        visitante = (int)(Math.random() * 10000);
    }while( (visitante == 0) || esRepetido(visitante) );
    String sql = "select codigo, nombre from delito where
estado='true'";

    Postgres db = new Postgres();
    db.conectar();
    datos = db.seleccionar(sql);
    while( datos.next() ){
        int codigo = datos.getInt(1);
        String nombre = datos.getString(2);
        fila = new HashMap();
        fila.put("codigo", codigo);
        fila.put("nombre", nombre);

        lista.add(fila);
    }
    int ci = 5049801;
    sql = "select nombre, paterno, materno from abogado where
ci="+ci;

    db.conectar();
    datos = db.seleccionar(sql);
    String abogado = "";
    if( datos.next() ){
        String nombre = datos.getString(1);
        String paterno = datos.getString(2);
        String materno = datos.getString(3);
        abogado = nombre+" "+paterno+" "+materno;
    }

    context.put("delitos",lista);
    context.put("visitante",visitante);
    context.put("ci",ci);
    context.put("abogado",abogado);
    template =
Velocity.getTemplate("plantillas/beneficiario/caso_legal.html");
}
catch (SQLException e) {

```

```

        System.err.println("Error en Area: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Area: " + e.getMessage());
    }
    return template;
}

private boolean esRepetido(int codigo){
    String sql = "select * from visitante where codigo="+codigo;
    try{
        Postgres db = new Postgres();
        db.conectar();
        ResultSet datos = db.seleccionar(sql);
        if( datos.next() ){
            System.out.println("repetido: "+codigo);
            return true;
        }
    }
    catch (SQLException e) {
        System.err.println("Error en Caso Legal: " + e.getMessage());
    }
    return false;
}
}

```

```

package beneficiario;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

```

```

public class Comentarios extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            String sql = "select * from comentario";
            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            while( datos.next() ){
                //int codigo = datos.getInt(1);
                String hora = datos.getString(2);
                String enunciado = datos.getString(3);
                String vst = ""; //datos.getString(5);
                String act = ""; //datos.getString(6);
                fila = new HashMap();
                //fila.put("codigo", codigo);
                fila.put("hora", hora);
                fila.put("enunciado", enunciado);
                fila.put("visitante", vst);
                fila.put("actividad", act);

                lista.add(fila);
            }
            context.put("comentarios", lista);
            template =
Velocity.getTemplate("plantillas/beneficiario/comentarios.html");
        }
        catch (SQLException e) {
            System.err.println("Error en Actividades: " + e.getMessage());
        }
        catch (Exception e) {
            System.err.println("Error en Actividades: " + e.getMessage());
        }
        return template;
    }
}

```

```

}

package beneficiario;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

public class Contenido extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            int act = Integer.parseInt(request.getParameter("actividad"));
            String sql = "select nombre, descr, estado from contenido
where actividad="+act;
            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            if( datos.next() ){
                String nombre = datos.getString(1);
                String descr = datos.getString(2);
                boolean est = datos.getBoolean(3);
                String estado = "Inactivo";
                if( est ){

```

```

                estado = "Activo";
            }
            context.put("nombre",nombre);
            context.put("desc",descr);
            context.put("estado",estado);
        }
        template =
Velocity.getTemplate("plantillas/beneficiario/contenidos.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Contenido: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Contenido: " + e.getMessage());
    }
    return template;
}

}

package beneficiario;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

public class Curriculum extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;

```

```

        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        int ci = Integer.parseInt(request.getParameter("ci"));
        try {
            String sql = "select a.ci, a.nombre, a.paterno, a.materno,
p.nombre, e.nombre " +
                        "from abogado a, curriculum c, profesion p,
especialidad e " +
                        "where a.curriculum=c.codigo and
c.profesion=p.codigo and c.especialidad=e.codigo and a.ci="+ci;
            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            while( datos.next() ){
                //int ci = datos.getInt(1);
                String nombre = datos.getString(2);
                String paterno = datos.getString(3);
                String materno = datos.getString(4);
                String profesion = datos.getString(5);
                String especialidad = datos.getString(6);
                fila = new HashMap();
                fila.put("nombre", nombre+" "+paterno+" "+materno);
                fila.put("profesion", profesion);
                fila.put("especialidad", especialidad);

                lista.add(fila);
            }
            context.put("curriculum",lista);
            template =
Velocity.getTemplate("plantillas/beneficiario/curriculums.html");
        }
        catch (SQLException e) {
            System.err.println("Error en Abogados: " + e.getMessage());
        }
        catch (Exception e) {
            System.err.println("Error en Abogados: " + e.getMessage());
        }
        return template;
    }
}

```

```

package beneficiario;

public class Direccion {

    private String barrio;
    private String calle;
    private String numero;
    private int localidad;
    private boolean estado;

    public Direccion(){
        barrio = "";
        calle = "";
        numero = "";
        localidad = 0;
    }

    public void setBarrio(String barrio) {
        this.barrio = barrio;
    }

    public String getBarrio() {
        return barrio;
    }

    public void setCalle(String calle) {
        this.calle = calle;
    }

    public String getCalle() {
        return calle;
    }

    public void setNumero(String numero) {
        this.numero = numero;
    }

    public String getNumero() {
        return numero;
    }

    public void setLocalidad(int localidad) {
        this.localidad = localidad;
    }

    public int getLocalidad() {
        return localidad;
    }

    public void setEstado(boolean estado) {
        this.estado = estado;
    }

    public boolean isEstado() {

```

```

        return estado;
    }
}
package beneficiario;

public class Familiar {

    private String ci;
    private String nombre;
    private String paterno;
    private String materno;
    private String telefono;
    private String correo;

    public Familiar(){
        ci = "";
        nombre = "";
        paterno = "";
        materno = "";
        telefono = "";
        correo = "";
    }

    public void setCi(String ci) {
        this.ci = ci;
    }

    public String getCi() {
        return ci;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getNombre() {
        return nombre;
    }

    public void setPaterno(String paterno) {
        this.paterno = paterno;
    }

    public String getPaterno() {
        return paterno;
    }

    public void setMaterno(String materno) {
        this.materno = materno;
    }

    public String getMaterno() {
        return materno;
    }
}

```

```

    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setCorreo(String correo) {
        this.correo = correo;
    }

    public String getCorreo() {
        return correo;
    }
}

package beneficiario;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

import java.util.Map;
import java.util.HashMap;
import java.util.ArrayList;

import java.sql.ResultSet;
import java.sql.SQLException;

public class Lugar extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {

```

```

        int lugar = Integer.parseInt(request.getParameter("lugar"));
        String sql = "select nombre, fecha, descr, objetivo from lugar
where codigo="+lugar;
        Postgres db = new Postgres();
        db.conectar();
        datos = db.seleccionar(sql);
        if( datos.next() ){
            String nombre = datos.getString(1);
            String fecha = datos.getString(2);
            String descr = datos.getString(3);
            String obj = datos.getString(4);
            context.put("nombre",nombre);
            context.put("fecha",fecha);
            context.put("desc",descr);
            context.put("obj",obj);
        }
        template =
Velocity.getTemplate("plantillas/beneficiario/lugares.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Lugar: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Lugar: " + e.getMessage());
    }
    return template;
}

}

package beneficiario;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;

```

```

import org.apache.velocity.servlet.VelocityServlet;

import base.Postgres;

public class Noticia extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
        HttpServletResponse response, Context context) {
        Template template = null;
        ResultSet datos = null;
        ArrayList lista = new ArrayList();
        Map fila = null;
        try {
            String sql = "select n.titulo, n.contenido, n.area, n.fecha,
n.hora, n.estado, a.nombre, a.paterno, a.materno " +
                "from noticia n, administrador a where
a.ci=n.adm;";

            Postgres db = new Postgres();
            db.conectar();
            datos = db.seleccionar(sql);
            while( datos.next() ){
                String titulo = datos.getString(1);
                String contenido = datos.getString(2);
                String area = datos.getString(3);
                String fecha = datos.getString(4);
                String hora = datos.getString(5);
                boolean est = datos.getBoolean(6);
                String estado = "Inactivo";
                if( est ){
                    estado = "Activo";
                }
                String adm = datos.getString(7)+"
"+datos.getString(8)+" "+datos.getString(9);
                fila = new HashMap();
                fila.put("titulo", titulo);
                fila.put("contenido", contenido);
                fila.put("area", area);
                fila.put("fecha", fecha);
                fila.put("hora", hora);
                fila.put("estado", estado);
                fila.put("adm", adm);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        lista.add(fila);
    }
    context.put("noticias",lista);
    template =
Velocity.getTemplate("plantillas/beneficiario/noticias.html");
    }
    catch (SQLException e) {
        System.err.println("Error en Noticias: " + e.getMessage());
    }
    catch (Exception e) {
        System.err.println("Error en Noticias: " + e.getMessage());
    }
    return template;
}
}

```

```

package beneficiario;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.velocity.Template;
import org.apache.velocity.app.Velocity;
import org.apache.velocity.context.Context;
import org.apache.velocity.servlet.VelocityServlet;

import util.Tiempo;

import base.Postgres;

public class Visitante extends VelocityServlet {
    private static final long serialVersionUID = 1L;

    public Template handleRequest(HttpServletRequest request,
HttpServletResponse response, Context context) {

```

```

Template template = null;
int visitante = Integer.parseInt(request.getParameter("visitante"));
String consulta = request.getParameter("consulta");
int delito = Integer.parseInt(request.getParameter("delitos"));
//int abogado = Integer.parseInt(request.getParameter("abogado"));
int abogado = 5049801;
Tiempo tm = new Tiempo();
String fecha = tm.getFecha();
String hora = tm.getHora();
System.out.println("visitante: "+visitante);
System.out.println("consulta: "+consulta);
System.out.println("delito: "+delito);
System.out.println("fecha: "+fecha);
System.out.println("hora: "+hora);

try {
    String sql = "insert into visitante(codigo,fecha,hora)
values("+visitante+", "+fecha+", "+hora+");";
    Postgres db = new Postgres();
    db.conectar();
    db.insertar(sql);
    db.cerrar();
    sql = "insert into caso_legal(consulta,delito,abogado,visitante)
values("+consulta+", "+delito+", "+abogado+", "+visitante+");";
    db.conectar();
    db.insertar(sql);
    db.cerrar();

    //Map datos = new HashMap();
    //datos.put("visitante", visitante);

    context.put("visitante", visitante);

    template =
Velocity.getTemplate("plantillas/beneficiario/visitante.html");
}
catch (SQLException e) {
    System.err.println("Error en Actividades: " + e.getMessage());
}
catch (Exception e) {
    System.err.println("Error en Actividades: " + e.getMessage());
}
}

```

```

        return template;
    }
}

HTML
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body >
<center>
    
    <table width="100%" border="0">
        <tr>
            <th colspan="2" scope="col"><h1><strong>LISTA DE ABOGADOS
</strong></h1></th>
        </tr>
        <tr>
            <td colspan="2"><table width="100%" border="1" align="center">
                <tr>
                    <td><h4>Nombre</h4></td>
                    <td><h4>Ap. Paterno</h4></td>
                    <td><h4>Ap. Materno</h4></td>
                    <td><h4>Curriculum</h4></td>
                    <td><h4>Abogado de Convenio</h4></td>
                    <td><h4>Abogado de la ONG</h4></td>
                </tr>
                #foreach ($fila in $abogados)
                <tr>
                    <td>$fila.nombre</td>
                    <td>$fila.paterno</td>
                    <td>$fila.materno</td>
                    <td><A
href="/Mundo Mejor/Curriculum?ci=$fila.curriculum">Ver</A> </td>
                    <td>$fila.convenio</td>
                    <td>$fila.ong</td>
                </tr>
                #end
            </table></td>
        </tr>
        <tr>
            <td width="50%" align="left"><a href="#">Volver</a></td>
            <td width="50%" align="right">Buscar
            :
            <input name="act" type="text" id="act" /></td>
        </tr>

```

```

    </table>
    <p>&nbsp;</p>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<center>

    <table width="100%" border="0">
        <tr>
            <th colspan="2" scope="col"><h1><strong>LISTA DE ACTIVIDADES
</strong></h1></th>
        </tr>
        <tr>
            <td colspan="2"><table width="100%" border="1" align="center">
                <tr>
                    <td><h4>Nombre</h4></td>
                    <td><h4>Estado</h4></td>
                    <td><h4>Area</h4></td>
                    <td><h4>Lugar</h4></td>
                    <td><h4>Administrador</h4></td>
                </tr>
                #foreach ($fila in $actividades)
                <tr>
                    <td><a
href="/Mundo_Mejor/Contenido?actividad=$fila.codigo">$fila.nombre</a
> </td>
                    <td>$fila.estado</td>
                    <td align="center"><a
href="/Mundo_Mejor/Area?area=$fila.area">Ver</a></td>
                    <td align="center"><a
href="/Mundo_Mejor/Lugar?lugar=$fila.lugar">Ver</a></td>
                    <td>$fila.adm</td>
                </tr>
                #end
            </table></td>
        </tr>
        <tr>
            <td colspan="2" align="right"><a href="#">opinar</a></td>
        </tr>
        <tr>
            <td width="50%" align="left"><a href="#">Volver</a></td>
            <td width="50%" align="right">Buscar
            :
            <input name="act" type="text" id="act" /></td>

```

```

        </tr>
    </table>
    <p>&nbsp;</p>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<form id="form1" name="form1" method="post" action="">
<div align="center">
<h2>BENEFICIARIO </h2>
    <table width="22%" border="0">
        <tr>
            <td><div align="left">Visitante: </div></td>
            <td><input name="visitante" type="text" id="visitante"
value="" /></td>
        </tr>
        <tr>
            <td width="42%"><div align="left">C.I.:</div></td>
            <td width="58%">
                <input type="text" name="ci" id="ci" />
            </td>
        </tr>
        <tr>
            <td><div align="left">Nombre: </div></td>
            <td><input type="text" name="nombre" id="nombre" /></td>
        </tr>
        <tr>
            <td><div align="left">Ap. Paterno</div></td>
            <td><input type="text" name="paterno" id="paterno" /></td>
        </tr>
        <tr>
            <td><div align="left">Ap. Materno: </div></td>
            <td><input type="text" name="materno" id="materno" /></td>
        </tr>
        <tr>
            <td><div align="left">Telefono: </div></td>
            <td><input type="text" name="telefon" id="telefon" /></td>
        </tr>
        <tr>
            <td><div align="left">Fecha de Nac.: </div></td>
            <td><input type="text" name="fechaNac" id="fechaNac" /></td>
        </tr>
        <tr>
            <td><div align="left">Lugar de Nac.: </div></td>
            <td><input type="text" name="lugarNac" id="lugarNac" /></td>
        </tr>
    </table>
</div>
</form>
</body>
</html>

```

```

</tr>
<tr>
  <td><div align="left">Correo: </div></td>
  <td><input type="text" name="correo" id="correo" /></td>
</tr>
<tr>
  <td colspan="2"><div align="right"><a
href="adicionarBeneficiario?modo=2">Direccion</a></div></td>
</tr>
<tr>
  <td colspan="2"><div align="right"><a
href="adicionarBeneficiario?modo=3">Familiar</a></div></td>
</tr>
<tr>
  <td colspan="2"><div align="center">
    <input type="submit" name="button" id="button"
value="Adicionar" />
    <input type="button" name="button2" id="button2"
value="Cancelar" onclick="history.back();" />
  </div></td>
</tr>
</table>
</div>
</form>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<form id="form1" name="form1" method="post" action="">
<div align="center">
<h2>DIRECCION</h2>
  <table width="22%" border="0">
    <tr>
      <td width="42%"><div align="left">Barrio.:</div></td>
      <td width="58%">
        <input type="text" name="barrio" id="barrio" />
      </td>
    </tr>
    <tr>
      <td><div align="left">Calle: </div></td>
      <td><input type="text" name="calle" id="calle" /></td>
    </tr>
    <tr>
      <td><div align="left">Numero</div></td>
      <td><input type="text" name="numero" id="numero" /></td>
    </tr>
  </table>

```

```

<tr>
  <td><div align="left">Departamento: </div></td>
  <td><select name="dpto" id="dpto">
</select></td>
</tr>
<tr>
  <td><div align="left">Provincia: </div></td>
  <td><select name="prov" id="prov">
</select></td>
</tr>
<tr>
  <td><div align="left">Localidad.: </div></td>
  <td><select name="local" id="local">
</select></td>
</tr>
<tr>
  <td colspan="2"><div align="center">
    <input type="submit" name="button" id="button"
value="Guardar" />
    <input type="button" name="button2" id="button2"
value="Cancelar" onclick="history.back();" />
  </div></td>
</tr>
</table>
</div>
</form>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<form id="form1" name="form1" method="post" action="">
<div align="center">
<h2>FAMILIAR</h2>
  <table width="22%" border="0">
    <tr>
      <td width="42%"><div align="left">C.I.:</div></td>
      <td width="58%">
        <input type="text" name="ci" id="ci" />
      </td>
    </tr>
    <tr>
      <td><div align="left">Nombre: </div></td>
      <td><input type="text" name="nombre" id="nombre" /></td>
    </tr>
    <tr>
      <td><div align="left">Ap. Paterno</div></td>

```

```

        <td><input type="text" name="paterno" id="paterno" /></td>
    </tr>
    <tr>
        <td><div align="left">Ap. Materno: </div></td>
        <td><input type="text" name="materno" id="materno" /></td>
    </tr>
    <tr>
        <td><div align="left">Telefono: </div></td>
        <td><input type="text" name="telefon" id="telefon" /></td>
    </tr>
    <tr>
        <td><div align="left">Correo: </div></td>
        <td><input type="text" name="correo" id="correo" /></td>
    </tr>
    <tr>
        <td colspan="2"><div align="right"><a
href="adicionarDireccion.html">Direccion</a></div></td>
    </tr>
    <tr>
        <td colspan="2"><div align="center">
            <input type="submit" name="button" id="button"
value="Adicionar" />
            <input type="button" name="button2" id="button2"
value="Cancelar" onclick="history.back();" />
        </div></td>
    </tr>
</table>
</div>
</form>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title></title>
</head>

<body>
<center>

    <table width="100%" border="0">
        <tr>
            <th colspan="2" scope="col"><h1><strong>DERECHO DE
FAMILIA</strong></h1></th>
        </tr>
        <tr>
            <td colspan="2"><table width="100%" border="1" align="center">
                <tr>
                    <td><h4>Tipo de Proceso</h4></td>
                    <td><h4>Descripcion</h4></td>
                </tr>
            </table>
        </td>
    </tr>
    </table>

```

```

#foreach ($fila in $areas)
_____ <tr>
      <td>$fila.nombre</td>
      <td>$fila.desc</td>
    </tr>
#end
_____ </table></td>
_____ </tr>
_____ <tr>
      <td width="50%" align="left"><a
href="javascript:history.back();">Volver</a></td>
      <td width="50%" align="right">Buscar
      :
      <input name="act" type="text" id="act" /></td>
    </tr>
  </table>
  <p>&nbsp;</p>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<center>

<h2>BENEFICIARIOS</h2>

<table width="80%" border="0">
  <tr>
    <th scope="col"><form id="form1" name="form1" method="post"
action="">
      <div align="right">Buscar:
      <input type="text" name="buscar" id="buscar" />
      </div>
    </form></th>
  </tr>
  <tr>
    <td><div align="center">
      <table width="100%" border="0">
        <tr>
          <th scope="col">C.I.</th>
          <th scope="col">Nombre</th>
          <th scope="col">Ap. Paterno</th>
          <th scope="col">Ap. Materno</th>
          <th scope="col">Telefono</th>
          <th scope="col">Fecha de Nac.</th>
        </tr>
      </table>
    </div>
  </td>
  </tr>
</table>

```

```

        <th scope="col">Lugar de Nac.</th>
        <th scope="col">Direccion</th>
        <th scope="col">Familiar</th>
    </tr>
#foreach ($beneficiario in $beneficiarios)
    <tr>
        <td>$beneficiario.ci</td>
        <td>$beneficiario.nombre</td>
        <td>$beneficiario.paterno</td>
        <td>$beneficiario.materno</td>
        <td>$beneficiario.telefono</td>
        <td>$beneficiario.fechaNac</td>
        <td>$beneficiario.lugarNac</td>
        <td><div align="center"><a
href="Direccion?direc=$direccion">Ver</a></div></td>
        <td><div align="center"><a
href="Familiar?familiar=$familiar">Ver</a></div></td>
    </tr>
#end
</table>
</div></td>
</tr>
<tr>
    <td><div align="right">
        <form id="form2" name="form2" method="post"
action="adicionarBeneficiario">
            <input type="hidden" name="modo" id="modo" value="1" />
            <input type="submit" name="adicionar" id="adicionar"
value="Adicionar" />
        </form>
    </div></td>
</tr>
</table>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
</head>

<body bgcolor="red">

<marquee>
<h1>
ONG "Mundo Mejor"
</h1>
</marquee>

</body>

```

```

</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title></title>
</head>

<body>
<center>

<form id="form1" name="form1" method="post" action="Visitante">
  <table width="100%" border="0">
    <tr>
      <th scope="col"><h1><strong>CASO LEGAL</strong></h1></th>
    </tr>
    <tr>
      <td>
        <table width="100%" border="1" align="center">
          <tr>
            <td width="23%"><h4>Consulta</h4></td>
            <td width="27%"><h4>Delito</h4></td>
            <td width="16%"><h4>Abogado</h4></td>
            <td width="16%"><h4>Visitante</h4></td>
          </tr>
          <tr>
            <td>
              <input type="text" name="consulta" id="clave" />
            </td>
            <td>
              <label>
                <select name="delitos" id="delito">
                  <option value=0>Seleccione</option>
                  #foreach ($fila in $delitos)
                    <option
value=$fila.codigo>$fila.nombre</option>
                  #end
                </select>
              </label>
            </td>
            <td><A
href="/Mundo Mejor/Curriculum?ci=$ci">$abogado</A></td>
          <td align="center">$visitante <input type="hidden"
name="visitante" id="visitante" value="$visitante" /> </td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td align="left">
      <label>

```

```

        <input type="checkbox" name="continuar" id="continuar" />
    </label>
    Desea continuar con el caso
  </td>
</tr>
<tr>
  <td>
    <input type="submit" value="Continuar" />
  </td>
</tr>
</table>
</form>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<center>

  <h1>&nbsp;</h1>
  <table width="75%" border="0">
    <tr>
      <th colspan="2"
scope="col"><h1><strong>COMENTARIOS</strong></h1></th>
    </tr>
    <tr>
      <td colspan="2"><table width="100%" border="1" align="center">
        <tr>
          <td><h4>Hora</h4></td>
          <td><h4>Enunciado</h4></td>
          <td><h4>Visitante</h4></td>
          <td><h4>Actividad</h4></td>
        </tr>
        <tr>
          <td>&nbsp;</td>
          <td>&nbsp;</td>
          <td>&nbsp;</td>
          <td>&nbsp;</td>
        </tr>
      </table></td>
    </tr>
    <tr>
      <td width="50%" align="right"><input value="Adicionar"
type="submit" id="act" /></td></td>
      <td width="50%" align="left"><input value="Cancelar"
type="submit" id="act" /></td>

```

```

        </tr>
    </table>
    <p>&nbsp;</p>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<center>

    <table width="100%" border="0">
        <tr>
            <th colspan="2"
scope="col"><h1><strong>CONTENIDO</strong></h1></th>
        </tr>
        <tr>
            <td colspan="2"><table width="100%" border="1" align="center">
                <tr>
                    <td><h4>Nombre</h4></td>
                    <td><h4>Descripcion</h4></td>
                    <td><h4>Estado</h4></td>
                </tr>
                <tr>
                    <td>${nombre}</td>
                    <td>${desc}</td>
                    <td>${estado}</td>
                </tr>
            </table></td>
        </tr>
        <tr>
            <td width="50%" align="left"><a
href="javascript:history.back();">Volver</a></td>
            <td width="50%" align="right">Buscar
                :
                <input name="act" type="text" id="act" /></td>
        </tr>
    </table>
    <p>&nbsp;</p>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"

```

```

/>
<title>Documento sin título</title>
</head>

<body>
<center>

  <table width="100%" border="0">
    <tr>
      <th colspan="2" scope="col"><h1><strong>CURRICULUM
</strong></h1></th>
    </tr>
    <tr>
      <td colspan="2"><table width="100%" border="1" align="center">
        <tr>
          <td><h4>Nombre</h4></td>
          <td><h4>Profesion</h4></td>
          <td><h4>Especialidad</h4></td>
        </tr>
#foreach ($fila in $currículum)
        <tr>
          <td>$fila.nombre</td>
          <td>$fila.profesion</td>
          <td>$fila.especialidad</td>
        </tr>
#end
      </table></td>
    </tr>
    <tr>
      <td width="50%" align="left"><a
href="javascript:history.back();">Volver</a></td>
      <td width="50%" align="right">&nbsp;</td>
    </tr>
  </table>
  <p>&nbsp;</p>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>ONG "MUNDO MEJOR"</title>
</head>
<frameset rows="80,*" cols="*" framespacing="0" frameborder="no"
border="0">
  <frame src="cabecera.html" name="topFrame" scrolling="No"
id="topFrame" title="cabecera" />
  <frameset rows="*" cols="200,*" framespacing="0" frameborder="no"
border="0">
    <frame src="menu.html" name="leftFrame" scrolling="auto"
noresize="noresize" id="leftFrame" title="menu" />

```

```

        <frame src="vista.html" name="mainFrame"
scrolling="auto" id="mainFrame" title="mainFrame" />
    </frameset>
</frameset>
<noframes>
    <body>
    </body>
</noframes>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<center>
    <table width="100%" border="0">
        <tr>
            <th colspan="2"
scope="col"><h1><strong>LUGAR</strong></h1></th>
        </tr>
        <tr>
            <td colspan="2"><table width="100%" border="1" align="center">
                <tr>
                    <td><h4>Nombre</h4></td>
                    <td><h4>Fecha</h4></td>
                    <td><h4>Descripcion</h4></td>
                    <td><h4>Objetivo</h4></td>
                </tr>
                <tr>
                    <td>$nombre</td>
                    <td>$fecha</td>
                    <td>$desc</td>
                    <td>$obj</td>
                </tr>
            </table></td>
        </tr>
        <tr>
            <td width="50%" align="left"><a
href="javascript:history.back();">Volver</a></td>
            <td width="50%" align="right">Buscar
:
            <input name="act" type="text" id="act" /></td>
        </tr>
    </table>
    <p>&nbsp;</p>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>

HOLA AJAX

</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title></title>
<script type="text/javascript" language="javascript" >

    function validar(){
        form = document.forms['form1'];
        lg = form.login.value;
        cl = form.clave.value;
        enviar(form.action, lg, cl);
    }

    function salida(){
        if( self.xmlHttpRequest.readyState == 4 ){
            var resp = self.xmlHttpRequest.responseText;
            //alert(resp);
            form = document.forms['form1'];
            form.login.value = '';
            form.clave.value = '';
            if( resp==1 ){
                // administrador
                form.action = 'Administrador';
                //alert('continuar: '+form.action);
                //form.submit();
            } else if( resp==2 ) {
                // abogado
                form.action = 'Beneficiarios';
                //alert('continuar: '+form.action);
                form.submit();
            } else {
                alert('Login y/o Clave incorrecta. ');
                form.login.focus();
            }
        }
    }
}

```

```

</script>
</head>

<body>
<table width="150" border="1" align="center" cellpadding="0"
cellspacing="0">
  <caption>
    Menu Principal
  </caption>
  <tr>
    <td align="left" scope="col"><a href="nosotros.html"
target="mainFrame">Quienes Somos</a></td>
  </tr>
  <tr>
    <td align="left"><a href="/Mundo_Mejor/Abogados"
target="mainFrame">Abogados</a></td>
  </tr>
  <tr>
    <td align="left"><a href="/Mundo_Mejor/Actividades"
target="mainFrame">Actividades</a></td>
  </tr>
  <tr>
    <td align="left"><a href="/Mundo_Mejor/Area"
target="mainFrame">Area</a></td>
  </tr>
  <tr>
    <td align="left"><a href="/Mundo_Mejor/Noticia"
target="mainFrame">Noticias</a></td>
  </tr>
  <tr>
    <td align="left"><a href="/Mundo_Mejor/CasoLegal"
target="mainFrame">Caso Legal</a></td>
  </tr>
  <tr>
    <td align="left"><a href="/Mundo_Mejor/Comentarios"
target="mainFrame">Comentarios</a></td>
  </tr>
</table>
<p>&nbsp;</p>
<form id="form1" name="form1" method="post" target="mainFrame"
action="Validar" onsubmit="validar(); return false;">
<table width="150" border="1" align="center" cellpadding="0"
cellspacing="0">
  <tr>
    <th colspan="2" scope="col">ENTRADA</th>
  </tr>
  <tr>
    <td width="42%">Login: </td>
    <td width="58%">
      <input name="login" type="password" id="login" size="10"
maxlength="20" />
    </td>
  </tr>
</table>

```

```

        <td>Clave: </td>
        <td><input name="clave" type="password" id="clave" size="10"
maxlength="20"/></td>
    </tr>
    <tr>
        <td colspan="2"><div align="center">
            <input type="submit" name="button" id="button"
value="INGRESAR" />
        </div></td>
    </tr>
</table>
</form>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin ttulo</title>
</head>

<body>
<div align="center">

</div>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<center>

    <table width="100%" border="0">
        <tr>
            <th colspan="2" scope="col"><h1><strong>LISTA DE
NOTICIAS</strong></h1></th>
        </tr>
        <tr>
            <td colspan="2"><table width="100%" border="1" align="center">
                <tr>
                    <td><h4>Titulo</h4></td>
                    <td><h4>Contenido</h4></td>
                    <td><h4>Area</h4></td>
                    <td><h4>Fecha</h4></td>
                    <td><h4>Hora</h4></td>
                </tr>
            </table>
        </td>
    </tr>

```

```

        <td>Estado</td>
        <td><h4>Responsable</h4></td>
    </tr>
#foreach ($fila in $noticias)
    <tr>
        <td>$fila.titulo</td>
        <td align="center">$fila.contenido</td>
        <td>$fila.area</td>
        <td align="center">$fila.fecha</a></td>
        <td>$fila.hora</td>
        <td>$fila.estado</td>
        <td>$fila.adm</td>
    </tr>
#end
</table></td>
</tr>
<tr>
    <td width="50%" align="left"><a href="#">Volver</a></td>
    <td width="50%" align="right">Buscar
    :
    <input name="act" type="text" id="act" /></td>
</tr>
</table>
<p>&nbsp;</p>
</center>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title></title>
</head>
<center>
Usted debe apersonarse por las oficinas de la ONG, con este codigo
'$visitante'.
<input type="button" value="VOLVER"
onclick="window.location='CasoLegal'" />
</center>
<body>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>
<body>

```

```

<div align="center">
  <p>Pagina por defecto.
</p>
  <p></p>
</div>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
</head>

<body bgcolor="red">

<marquee>
<h1>
ONG "Mundo Mejor"
</h1>
</marquee>

</body>

</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>ONG "MUNDO MEJOR"</title>
</head>
<frameset rows="80,*" cols="*" framespacing="0" frameborder="no"
border="0">
  <frame src="plantillas/cabecera.html" name="topFrame"
scrolling="No" id="topFrame" title="cabecera" />
  <frameset rows="*" cols="200,*" framespacing="0" frameborder="no"
border="0">
    <frame src="plantillas/menu.html" name="leftFrame"
scrolling="auto" noresize="noresize" id="leftFrame" title="menu" />
    <frame src="plantillas/vista.html" name="mainFrame"
scrolling="auto" id="mainFrame" title="mainFrame" />
  </frameset>
</frameset>
<noframes>
  <body>
  </body>
</noframes>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
<script type="text/javascript" language="javascript"
src="clases/ajax.js" ></script>
<script type="text/javascript" language="javascript" >

    function validar(){
        form = document.forms['form1'];
        lg = form.login.value;
        cl = form.clave.value;
        enviar(form.action, lg, cl);
    }

    function salida(){
        if( self.xmlHttpRequest.readyState == 4 ){
            var resp = self.xmlHttpRequest.responseText;
            //alert(resp);
            form = document.forms['form1'];
            form.login.value = '';
            form.clave.value = '';
            if( resp==1 ){
                // administrador
                form.target = 'top';
                form.action = 'administrador/index.html';
                //alert('continuar: '+form.action);
                //form.onsubmit(true);
                form.submit();
            } else if( resp==2 ) {
                // abogado
                form.target = 'top';
                form.action = 'abogado/index.html';
                //alert('continuar: '+form.action);
                form.submit();
            } else {
                alert('Login y/o Clave incorrecta. ');
                form.login.focus();
            }
        }
    }

</script>
</head>

<body>
<table width="150" border="1" align="center" cellpadding="0"
cellspacing="0">
    <caption>
        Menu Principal
    </caption>
    <tr>
        <td align="left" scope="col"><a

```

```

href="beneficiario/nosotros.html" target="mainFrame">Quienes
Somos</a></td>
</tr>
<tr>
<td align="left"><a href="/Mundo_Mejor/Abogados"
target="mainFrame">Abogados</a></td>
</tr>
<tr>
<td align="left"><a href="/Mundo_Mejor/Actividades"
target="mainFrame">Actividades</a></td>
</tr>
<tr>
<td align="left"><a href="/Mundo_Mejor/Area"
target="mainFrame">Area</a></td>
</tr>
<tr>
<td align="left"><a href="/Mundo_Mejor/Noticia"
target="mainFrame">Noticias</a></td>
</tr>
<tr>
<td align="left"><a href="/Mundo_Mejor/CasoLegal"
target="mainFrame">Caso Legal</a></td>
</tr>
<tr>
<td align="left"><a href="/Mundo_Mejor/Comentarios"
target="mainFrame">Comentarios</a></td>
</tr>
</table>
<p>&nbsp;</p>
<form id="form1" name="form1" method="post" target="mainFrame"
action="../Validar" onsubmit="validar(); return false;">
<table width="150" border="1" align="center" cellpadding="0"
cellspacing="0">
<tr>
<th colspan="2" scope="col">ENTRADA</th>
</tr>
<tr>
<td width="42%">Login: </td>
<td width="58%">
<input name="login" type="password" id="login" size="10"
maxlength="20" />
</td>
</tr>
<tr>
<td>Clave: </td>
<td><input name="clave" type="password" id="clave" size="10"
maxlength="20"/></td>
</tr>
<tr>
<td colspan="2"><div align="center">
<input type="submit" name="button" id="button"
value="INGRESAR" />
</div></td>
</tr>

```

```

</table>
</form>
</body>
</html>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Documento sin título</title>
</head>

<body>
<div align="center">
  <p></p>
</div>
</body>
</html>

```

WEB XML

```

<web-app>
  <servlet>
    <servlet-name>Actividades</servlet-name>
    <servlet-class>beneficiario.Actividades</servlet-class>
    <init-param>
      <param-name>properties</param-name>
      <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
  </servlet>
  <servlet>
    <servlet-name>Abogados</servlet-name>
    <servlet-class>beneficiario.Abogados</servlet-class>
    <init-param>
      <param-name>properties</param-name>
      <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
  </servlet>
  <servlet>
    <servlet-name>Curriculum</servlet-name>
    <servlet-class>beneficiario.Curriculum</servlet-class>
    <init-param>
      <param-name>properties</param-name>
      <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
  </servlet>
  <servlet>
    <servlet-name>Area</servlet-name>
    <servlet-class>beneficiario.Area</servlet-class>
    <init-param>
      <param-name>properties</param-name>
      <param-value>/WEB-INF/velocity.properties</param-

```

```

value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>Lugar</servlet-name>
    <servlet-class>beneficiario.Lugar</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>Contenido</servlet-name>
    <servlet-class>beneficiario.Contenido</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>Noticia</servlet-name>
    <servlet-class>beneficiario.Noticia</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>CasoLegal</servlet-name>
    <servlet-class>beneficiario.CasoLegal</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>Comentarios</servlet-name>
    <servlet-class>beneficiario.Comentarios</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>Beneficiarios</servlet-name>
    <servlet-class>beneficiario.Beneficiarios</servlet-
class>
    <init-param>
        <param-name>properties</param-name>

```

```

        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>Visitante</servlet-name>
    <servlet-class>beneficiario.Visitante</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>Validar</servlet-name>
    <servlet-class>util.Validar</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>AdicionarBeneficiario</servlet-name>
    <servlet-
class>beneficiario.AdicionarBeneficiario</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>

<!--Administrador-->
<servlet>
    <servlet-name>Administrador</servlet-name>
    <servlet-class>administrador.Administrador</servlet-
class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>AdmAbogados</servlet-name>
    <servlet-class>administrador.Abogados</servlet-class>
    <init-param>
        <param-name>properties</param-name>
        <param-value>/WEB-INF/velocity.properties</param-
value>
    </init-param>
</servlet>
<servlet>

```

```

        <init-param>
            <param-name>properties</param-name>
            <param-value>/WEB-INF/velocity.properties</param-
value>
        </init-param>
    </servlet>
    <servlet>
        <servlet-name>Provincias</servlet-name>
        <servlet-class>administrador.Provincias</servlet-class>
        <init-param>
            <param-name>properties</param-name>
            <param-value>/WEB-INF/velocity.properties</param-
value>
        </init-param>
    </servlet>
    <servlet>
        <servlet-name>Localidades</servlet-name>
        <servlet-class>administrador.Localidades</servlet-class>
        <init-param>
            <param-name>properties</param-name>
            <param-value>/WEB-INF/velocity.properties</param-
value>
        </init-param>
    </servlet>
    <servlet>
        <servlet-name>AdmArea</servlet-name>
        <servlet-class>administrador.Area</servlet-class>
        <init-param>
            <param-name>properties</param-name>
            <param-value>/WEB-INF/velocity.properties</param-
value>
        </init-param>
    </servlet>
    <servlet>
        <servlet-name>AddArea</servlet-name>
        <servlet-class>administrador.AddArea</servlet-class>
        <init-param>
            <param-name>properties</param-name>
            <param-value>/WEB-INF/velocity.properties</param-
value>
        </init-param>
    </servlet>
    <servlet>
        <servlet-name>AdmActividades</servlet-name>
        <servlet-class>administrador.Actividades</servlet-class>
        <init-param>
            <param-name>properties</param-name>
            <param-value>/WEB-INF/velocity.properties</param-
value>
        </init-param>
    </servlet>
    <servlet>

```

```

        <init-param>
            <param-name>properties</param-name>
            <param-value>/WEB-INF/velocity.properties</param-
value>
        </init-param>
    </servlet>
    <!--Administrador-->

    <servlet-mapping>
        <servlet-name>Abogados</servlet-name>
        <url-pattern>/Abogados</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Curriculum</servlet-name>
        <url-pattern>/Curriculum</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Actividades</servlet-name>
        <url-pattern>/Actividades</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Area</servlet-name>
        <url-pattern>/Area</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Lugar</servlet-name>
        <url-pattern>/Lugar</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Contenido</servlet-name>
        <url-pattern>/Contenido</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Noticia</servlet-name>
        <url-pattern>/Noticia</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>CasoLegal</servlet-name>
        <url-pattern>/CasoLegal</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Comentarios</servlet-name>
        <url-pattern>/Comentarios</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Beneficiarios</servlet-name>
        <url-pattern>/Beneficiarios</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>Visitante</servlet-name>
        <url-pattern>/Visitante</url-pattern>

```

```

</servlet-mapping>
<servlet-mapping>
    <servlet-name>Validar</servlet-name>
    <url-pattern>/Validar</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AdicionarBeneficiario</servlet-name>
    <url-pattern>/adicionarBeneficiario</url-pattern>
</servlet-mapping>

<!-- Administrador -->
<servlet-mapping>
    <servlet-name>Administrador</servlet-name>
    <url-pattern>/Administrador</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AdmAbogados</servlet-name>
    <url-pattern>/AdmAbogados</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AddAbogado</servlet-name>
    <url-pattern>/AddAbogado</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>Provincias</servlet-name>
    <url-pattern>/Provincias</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>Localidades</servlet-name>
    <url-pattern>/Localidades</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AdmArea</servlet-name>
    <url-pattern>/AdmArea</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AddArea</servlet-name>
    <url-pattern>/AddArea</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AdmActividades</servlet-name>
    <url-pattern>/AdmActividades</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AdmContenido</servlet-name>
    <url-pattern>/AdmContenido</url-pattern>
</servlet-mapping>
<!-- Administrador -->
</web-app>

```

II.1.6.2. Pantallas

II.1.6.2.1 Quienes Somos

ONG "Mundo Mejor"

Menu Principal

Quienes Somos
Abogados
Actividades
Area
Noticias
Caso Legal
Comentarios



MUNDO MEJOR

EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

ENTRADA	
Login:	<input type="text"/>
Clave:	<input type="text"/>
<input type="button" value="INGRESAR"/>	

Figura 1.

II.1.6.2.2. Pantalla Abogados

ONG "Mundo Mejor"

Menu Principal

- [Quienes Somos](#)
- [Abogados](#)
- [Actividades](#)
- [Area](#)
- [Noticias](#)
- [Caso Legal](#)
- [Comentarios](#)



MUNDO MEJOR

EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

ENTRADA	
Login:	<input type="text"/>
Clave:	<input type="text"/>
<input type="button" value="INGRESAR"/>	

LISTA DE ABOGADOS

Nombre	Ap. Paterno	Ap. Materno	Curriculum	Abogado de Convenio	Abogado de la ONG
Marcelo	Plaza	Barrios	Ver	SI	NO
Juan	Fernandez		Ver	SI	NO

[Volver](#) Buscar:

Figura 2.

II.1.6.2.3. Pantalla Actividad

ONG "Mundo Mejor"

Menu Principal

- [Quienes Somos](#)
- [Abogados](#)
- [Actividades](#)
- [Area](#)
- [Noticias](#)
- [Caso Legal](#)
- [Comentarios](#)



MUNDO MEJOR
EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

ENTRADA

Login:

Clave:

LISTA DE ACTIVIDADES

Nombre	Estado	Area	Lugar	Administrador
ITS	Activo	Ver	Ver	Marcelo Plaza Barrios
Vigamia	Activo	Ver	Ver	Marcelo Plaza Barrios
Orientacion Vocacional	Activo	Ver	Ver	Marcelo Plaza Barrios

[Volver](#)
Buscar: [opinar](#)

Figura 3

II.1.6.2.4 . Pantalla Lugar

ONG "Mundo Mejor"

Menu Principal

- [Quienes Somos](#)
- [Abogados](#)
- [Actividades](#)
- [Area](#)
- [Noticias](#)
- [Caso Legal](#)
- [Comentarios](#)

LUGAR

Nombre	Fecha	Descripcion	Objetivo
San Jorge	2009-05-06	Instruccion	Orientacion

[Volver](#)
Buscar:

ENTRADA

Login:

Clave:

Figura 4.

II.1.6.2.5. Pantalla Area

ONG "Mundo Mejor"

Menú Principal

- [Quiénes Somos](#)
- [Abogados](#)
- [Actividades](#)
- [Área](#)
- [Noticias](#)
- [Caso Legal](#)
- [Comentarios](#)

ENTRADA	
Login:	<input type="text"/>
Clave:	<input type="password"/>
<input type="button" value="INGRESAR"/>	



MUNDO MEJOR

EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

DERECHO DE FAMILIA

Tipo de Proceso	Descripcion
Procesos Ordinarios	Orientacion
Procesos Sumarios	Hogar
Procesos Especiales	Embarazo
Proceso Penal	Justicia

[Volver](#)Buscar :

Figura 5

II.1.6.2.6. Pantalla Area

ONG "Mundo Mejor"

Menu Principal

Quienes Somos
Abogados
Actividades
Area
Noticias
Caso Legal
Comentarios

ENTRADA	
Login:	<input type="text"/>
Clave:	<input type="password"/>
<input type="button" value="INGRESAR"/>	



MUNDO MEJOR

EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

DERECHO DE FAMILIA

Tipo de Proceso	Descripcion
Procesos Sumarios	Hogar

[Volver](#) Buscar :

Figura 6.

II.1.6.2.7. Pantalla Noticias

ONG "Mundo Mejor"

Menu Principal

- [Quienes Somos](#)
- [Abogados](#)
- [Actividades](#)
- [Area](#)
- [Noticias](#)
- [Caso Legal](#)
- [Comentarios](#)



MUNDO MEJOR

EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

ENTRADA

Login:

Clave:

LISTA DE NOTICIAS

Titulo	Contenido	Area	Fecha	Hora	Estado	Responsable
Minero	Rechazo de minero a Evo	1	2010-10-14	13:30:00	Activo	Marcelo Plaza Barrios
Nueva Ley Mordaza	Libertad de expresion	4	2010-10-03	12:23:00	Activo	Marcelo Plaza Barrios
Ley Modelo	Prohibido usar a la mujer como objeto	4	2010-10-15	14:37:00	Activo	Marcelo Plaza Barrios

[Volver](#) Buscar:

Figura 7.

II.1.6.2.8. Pantalla

ONG "Mundo Mejor"

Menu Principal

- [Quiénes Somos](#)
- [Abogados](#)
- [Actividades](#)
- [Area](#)
- [Noticias](#)
- [Caso Legal](#)
- [Comentarios](#)

ENTRADA

Login:

Clave:



MUNDO MEJOR

EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

CASO LEGAL

Consulta	Delito	Abogado	Visitante
<input type="text" value="extracion de cosas"/>	<input type="text" value="Robo"/>	Marcelo Plaza Barrios	9126

Desea continuar con el caso

Figura 8.

II.1.6.2.9. Pantalla caso aceptado por beneficiario

ONG "Mundo Mejor"

Menu Principal

- [Quiénes Somos](#)
- [Abogados](#)
- [Actividades](#)
- [Área](#)
- [Noticias](#)
- [Caso Legal](#)
- [Comentarios](#)

ENTRADA

Login:

Clave:



MUNDO MEJOR

EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

CASO LEGAL

Consulta	Delito	Abogado	Visitante
extracion de cosas	Asistencia	Marcelo Plaza Barrios	9126

Desea continuar con el caso

Figura 9.

II.1.6.2.10. Pantalla Antecedentes

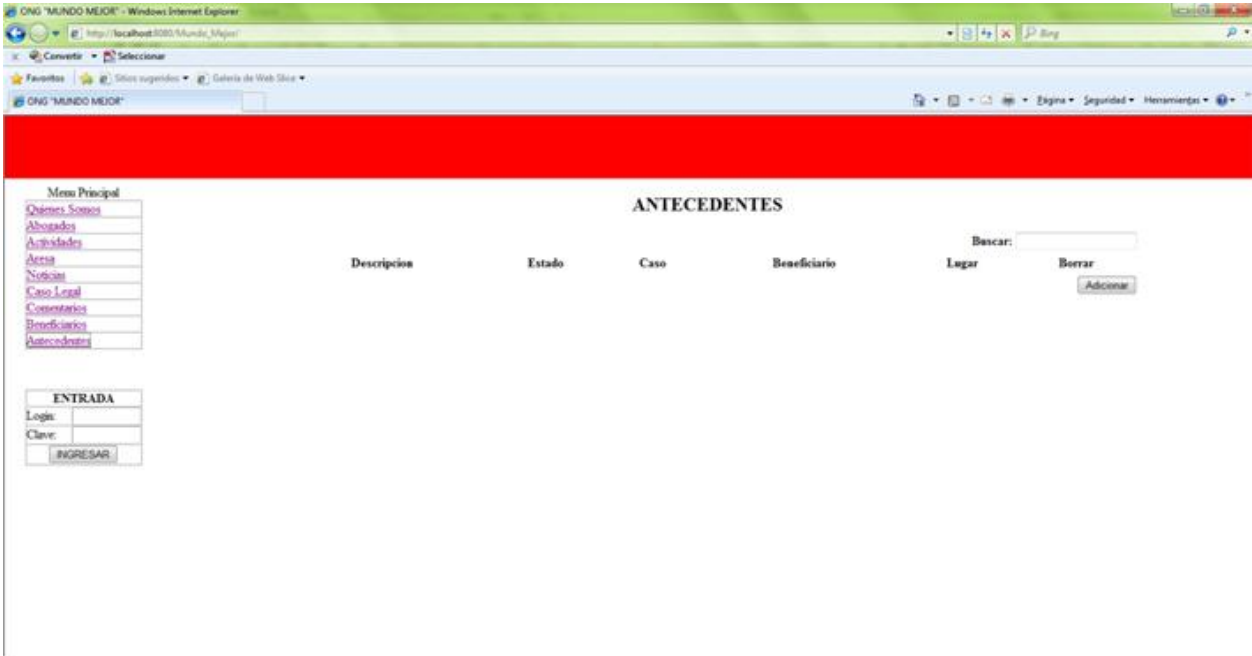


Figura 10.

II.1.6.2.11. Pantalla Comentarios



Figura 11.

II.1.6.2.12. Pantalla adicionar antecedente

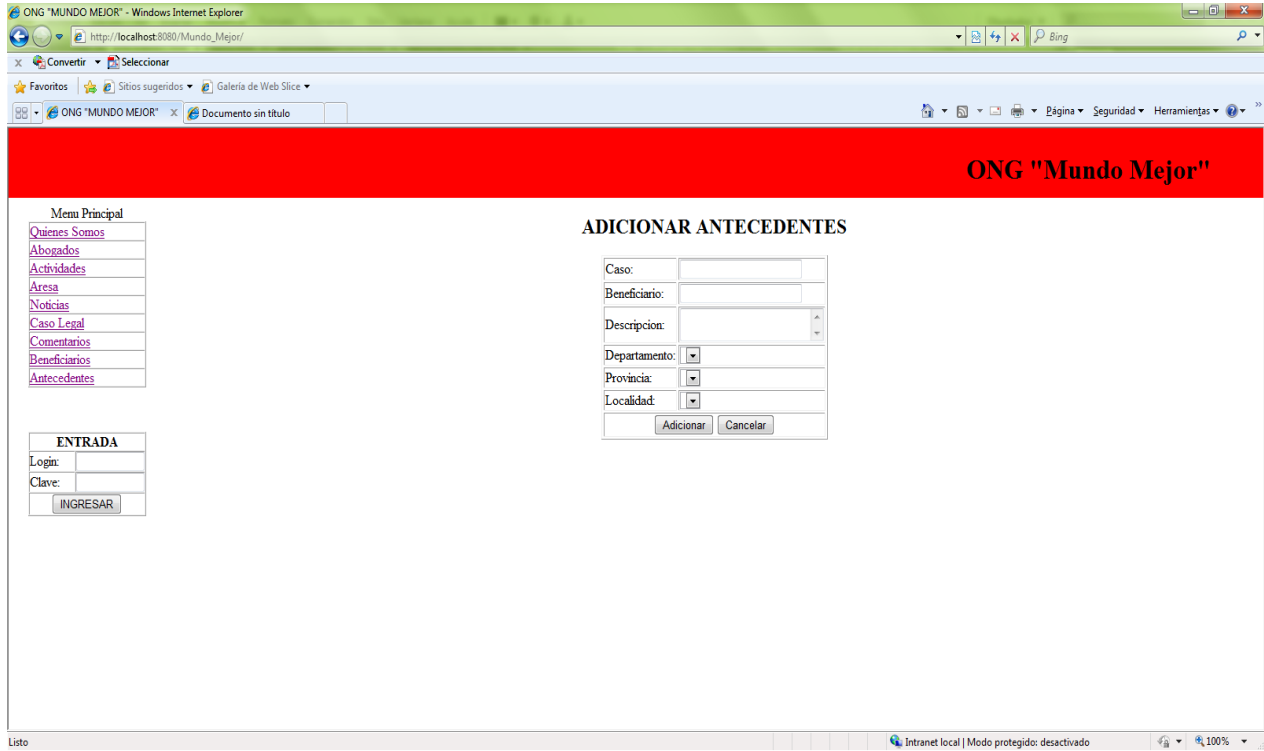


Figura 12

II.1.6.2.13. Pantalla eliminar antecedente

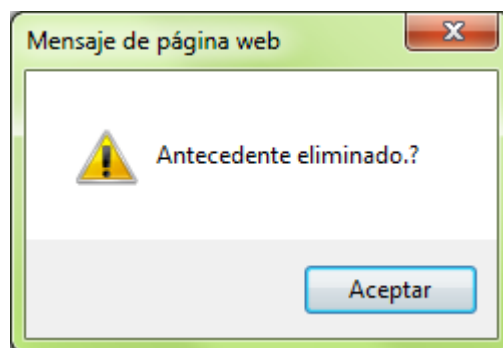
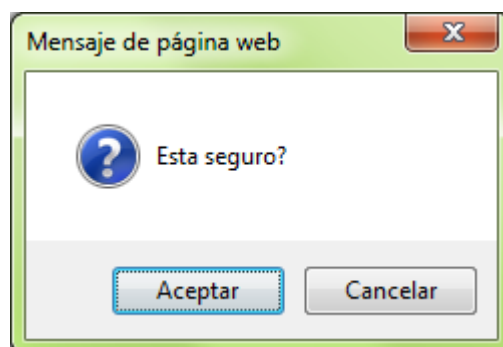


Figura 13

II.1.6.2.14. Pantalla Adicionar beneficiario

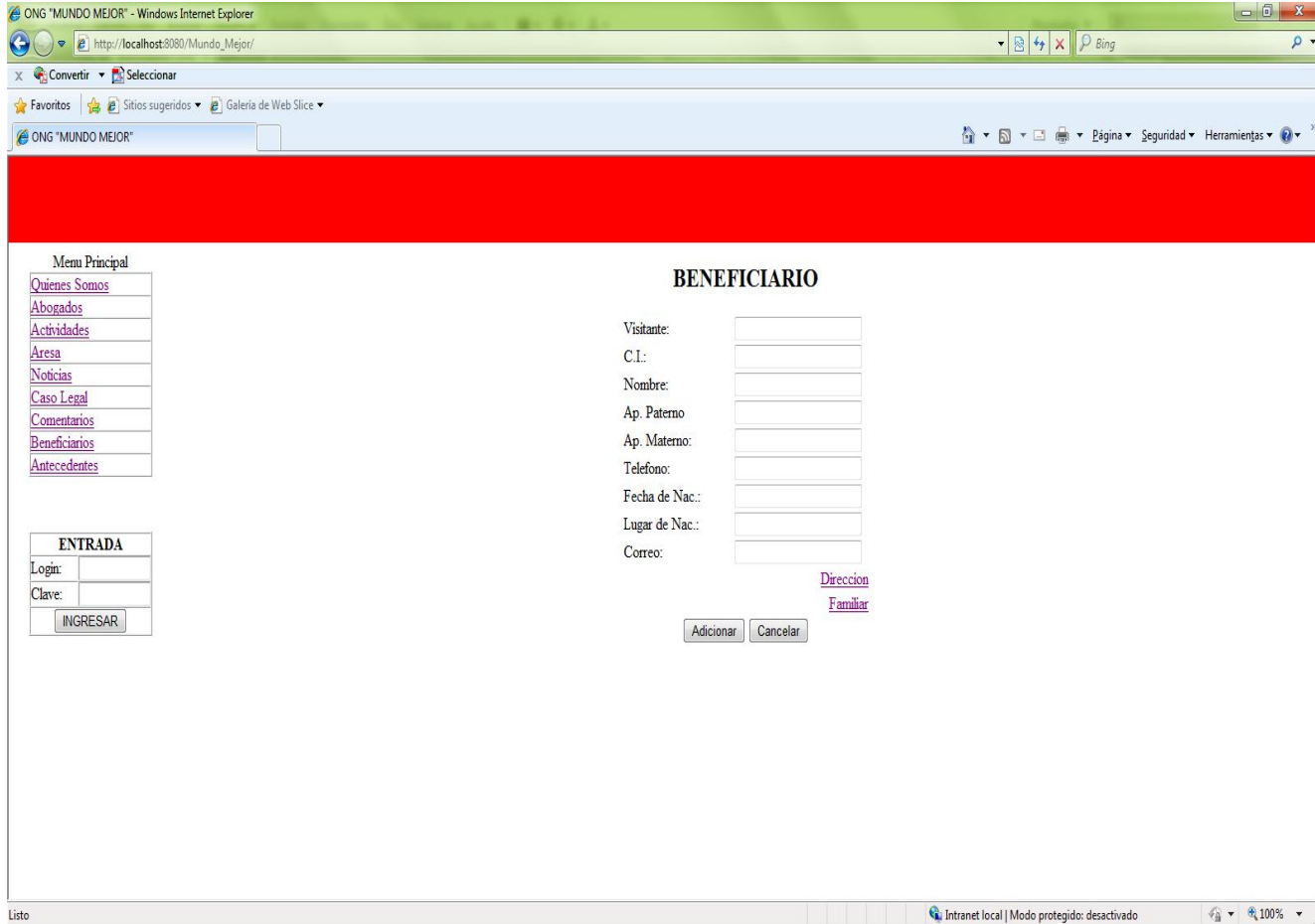


Figura 14.

II.1.6.2.15. Pantalla Direccion

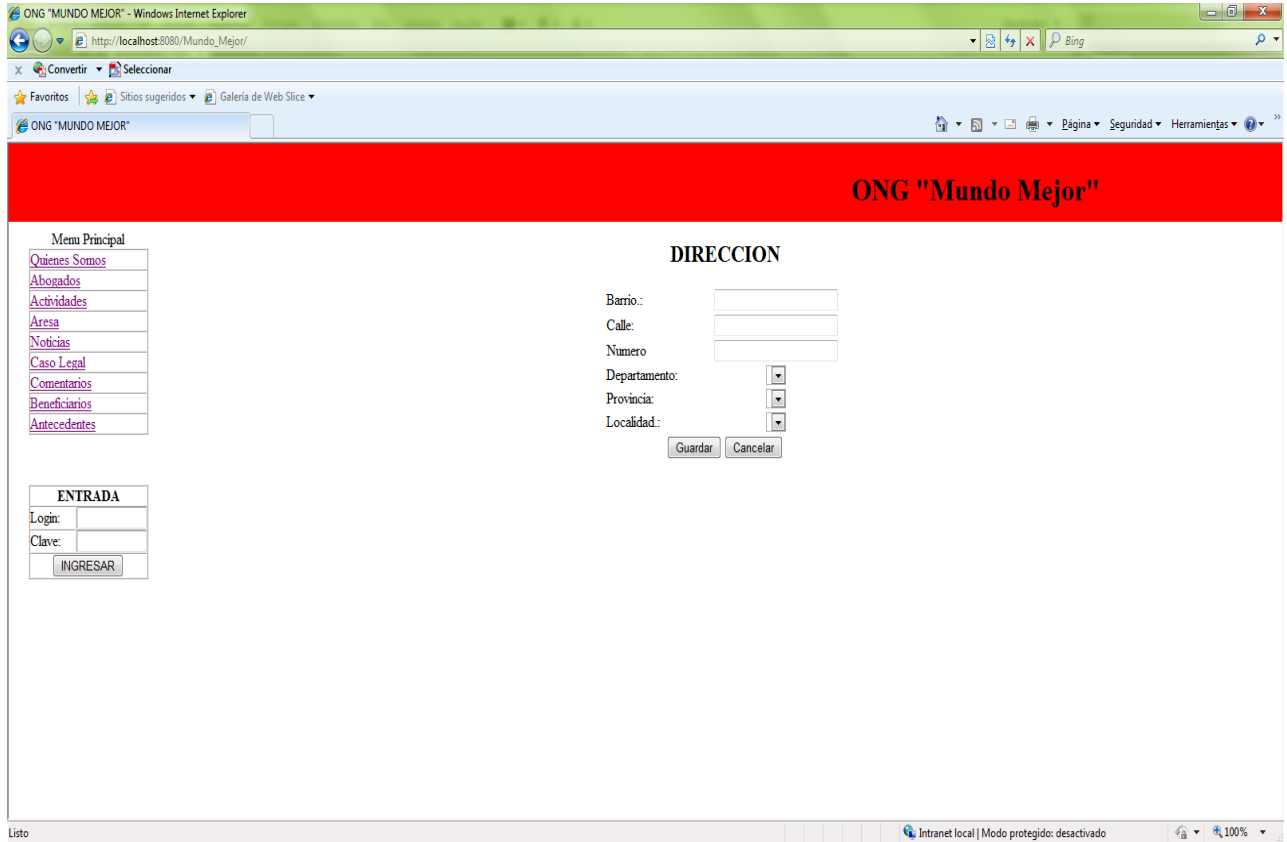


Figura 15.

II.1.6.2.16. Pantalla Curriculum

ONG "Mundo Mejor"

Menu Principal

- [Quienes Somos](#)
- [Abogados](#)
- [Actividades](#)
- [Area](#)
- [Noticias](#)
- [Caso Legal](#)
- [Comentarios](#)

ENTRADA

Logín:

Clave:



MUNDO MEJOR

EQUIPO MULTIDISCIPLINARIO DE FORMACION INTEGRAL

CURRICULUM

Nombre	Profesion	Especialidad
Marcelo Plaza Barrios	Abogado	Derecho Penal

[Volver](#)

Figura 16.

II.1.6.2.17. Pantalla Familiar

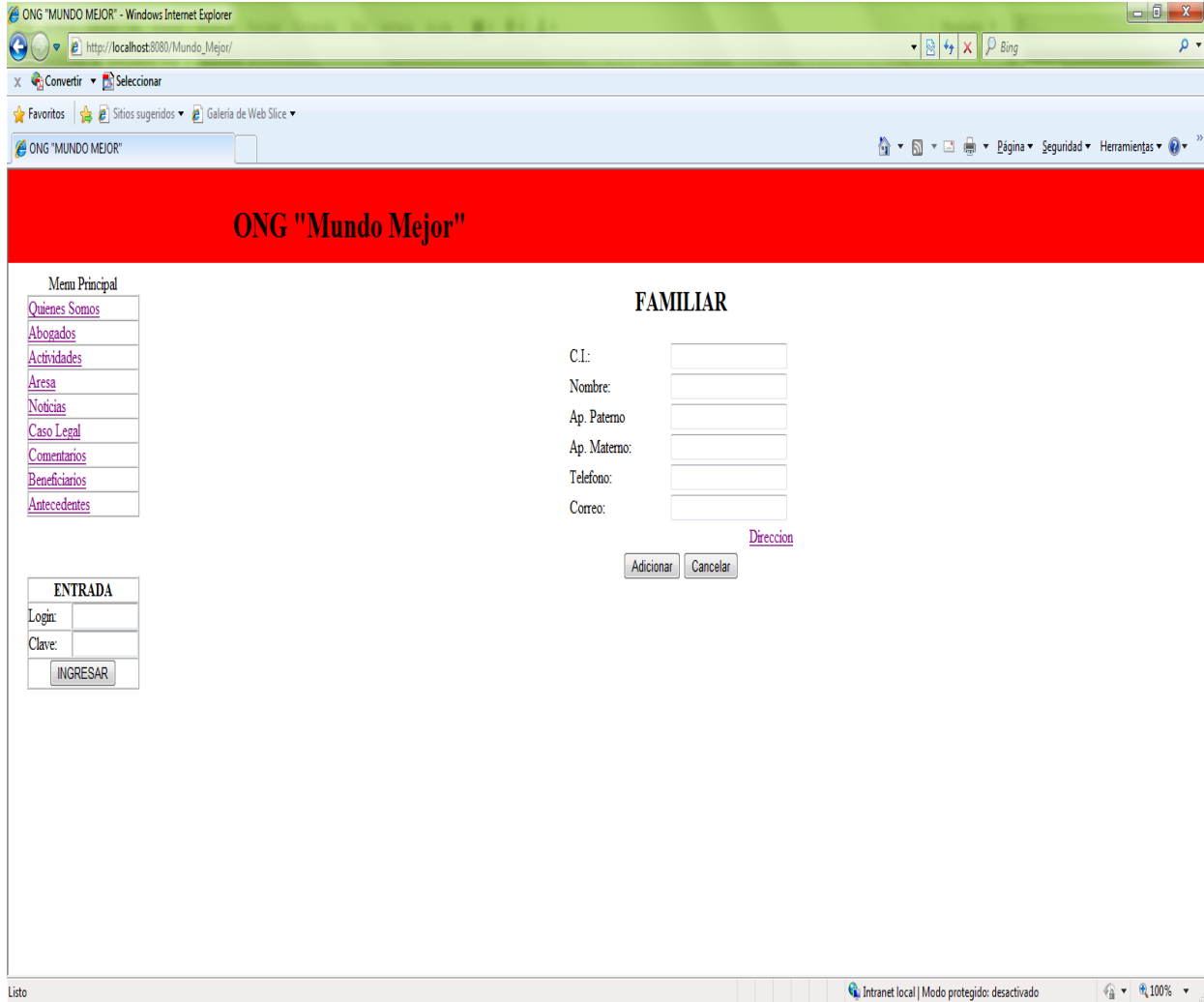


Figura 17.

II.1.6.2.18. Pantalla de Adicion

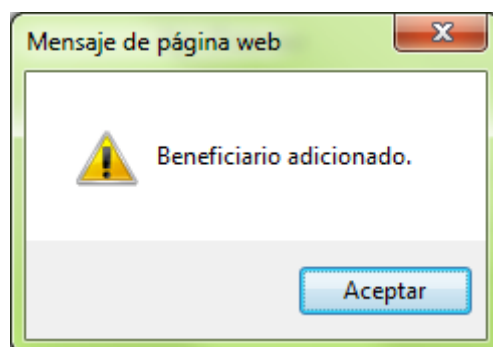


Figura 18

II.1.6.2.19. Pantalla Elenimar Abogado

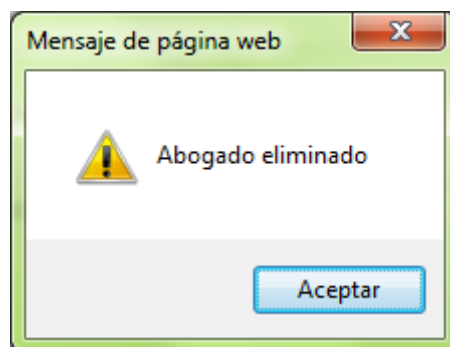


Figura 19.

I.7. Medio de Verificación

Documentación prestada por la ONG “MUNDO MEJOR” para dar inicio al Sistema.

VER ANEXO DEL COMPONENTE I

Carta firmada con la conformidad de la Directora de la ONG “Mundo Mejor”.

VER ANEXO DEL COMPONENTE I

Componente 2

II.2.1. Nombre

Implementar Estrategias de Socializacion del sistema Informático dirigido a los miembros de la ONG “MUNDO MEJOR”

II.2.2. Introducción

El colaborador que recibe capacitación siente que la empresa lo estima y, por lo tanto, le está asignando un salario espiritual y considera que están invirtiendo en su talento para mejorar su rendimiento, la calidad de su trabajo, elevar su productividad y, consecuentemente, piensa que puede estar próximo a un asenso.

Si bien es cierto que el aumento del salario económico es importante para mejorar la calidad de vida, también es cierto que, pasado cierto período, la nueva remuneración se diluye en satisfacer ciertas necesidades y, nuevamente, se requiere nuevo aumento; en cambio, el salario espiritual permite mejorar la calidad humana del hombre, coadyuva a la felicidad de su hogar.

El desarrollo de los recursos humanos es central ante el reto tan importante que la empresa enfrentan en este mundo globalizado y competitivo.

La capacitación y desarrollo del recurso humano, es una estrategia empresarial importante que deben acompañar a los demás esfuerzos de cambio que las organizaciones lleven adelante.

Mediante esta estrategia los colaboradores aprenden cosas nuevas, crecen individualmente, establecen relaciones con otros individuos, coordinan el trabajo a realizar, se ponen de acuerdo para introducir mejoras.

La capacitación ha demostrado ser un medio muy eficaz para hacer productivas a las personas, su eficacia se ha demostrado en más del 80% de todos los programas de capacitación.

El presente trabajo tiene como objeto desarrollar los métodos de asociación para la comprensión del sistema por parte de los interesados directos con el proyecto.

El componente socialización va dirigida a los integrantes de la ONG “MUNDO MEJOR”, ah adoptado una forma de llevar a cabo la realización de este componente:

- Mediante un curso de capacitación sobre el uso del Sistema Informatico ONG “MUNDO MEJOR”.

II.2.2.1. Concepto de Socializacion

la socialización es el proceso por el cual una sociedad se reproduce por sí misma en una nueva generación, es decir, mediante la socialización los valores y la tradición del pasado se continúan y perpetúan. Así, la socialización es lo que proporciona a la sociedad la continuidad en el tiempo, socializar algo pasa a ser el compartir ese algo con otros para que ellos lo interioricen y lo hagan parte de su ser (cultural).

II.2.2.2. Proceso de Socializacion

Es la manera con que los miembros de una colectividad aprenden los modelos culturales de su sociedad, los asimilan y los convierten en sus propias reglas personales de vida.

II.2.2.3. Agentes de Socializacion

Se puede decir que la totalidad de la gente de socialización y que cada persona que se entre en contacto de cirto modo es un agente sociabilizador.

II.2.2.4. Proposito

Capacitar a los integrantes de la ONG “MUNDO MEJOR”.

II.2.3. Marco Teorico

II.2.3.1. Comunicación Social

Es un campo de estudio sociológico que explora especialmente aquello relacionado con la información, como es percibida, transmitida y entendida y su impacto social.

II.2.3.1.1. Objetivos

Uno de los principales objetivos de la publicidad es crear demanda o modificar las tendencias de demanda de un producto.

Medios convencionales:

- Anuncios Televisivos
- Anuncios de Radio
- Anuncios de prensa
- Tripticos
- Afiches

II.2.3.2. Propuesta del Proyecto referente a la Socializacion

II.2.3.2.1. Planteamiento del Problema

La ONG “MUNDO MEJOR” de un tiempo ahora se ve obligada a incursionar en el ámbito informático por el crecimiento de las tecnologías además de facilitar la información para el conocimiento y su transferencia en el area de derecho de familia.

Y los medios que se dan a conocer no se difunden para el conocimiento general, por lo tanto es importante dar a conocer los beneficios que traen las aplicaciones del Sistema.

II.2.3.2.2 Objetivo General

Sociabilizar todas las aplicaciones del Sistema informático ONG “MUNDO MEJOR”.

II.2.3.2.3. Objetivos Especificos

- Brindar información actual de todas las actividades que realicen
- Brindar información sobre el area en la que se trabaja.

- Tranferencia de información.

II.2.3.2.4. Metodos de Socializacion

Se pretende adoptar este método:

- Estetica: Mostrar imágenes llamativas del Sistema Informático
- Humor: Llegar al publico mediante frases.

II.2.3.2.5. Publico Objetivo

- Los integrantes de la ONG “MUNDO MEJOR”.

II.2.3.2.6. Resultados Esperados

- El manejo del Sistema de los Integrantes de la ONG “MUNDO MEJOR”.

II.2.3.3. Medios de Verificacion

Ver anexo del componente 2.

Carta firmada por la Directora de la ONG “MUNDO MEJOR” con la conformidad del sistema.

Ver anexo del componente 2.

III.1 Conclusiones

Las conclusiones son los resultados concretos que se obtuvieron en el desarrollo de la investigación. Van acorde al número de objetivos planteados en la investigación; esto no quiere decir que no se excluya otra información importante obtenida durante el estudio.

Tras un estudio completo de las aplicaciones que realiza la ONG “MUNDO MEJOR”, se detecto la información a automatizar es muy compleja de tal manera se adecuo el sistema a una sola area y su información es de vital importancia.

Con el desarrollo del sistema se automatizo, la información requerida por la institución se origino mejoras de tal manera dando el cumplimiento acordado con la directora de la institución, con el transcurso del tiempo podrá realizar una ampliación del sistema en las diferentes area legales que faltan.

Las tecnologías utilizadas para el desarrollo del sistema, fueron las adecuadas para este tipo de aplicaciones, ya que java (programación orientada a objetos) es un código abierto y no necesita de alguna licencia para utilizarla, también el gestor de base de datos Postgres y el servidor Web Apache. Lo mismo con ajax se utilizo para optimizar el rendimiento y carga de las páginas logrando una mejor interacción con el usuario.

Tambien se demuestra aunque que el uso adecuado de las metodologías, consiguen un objetivo sustancioso, en este caso, el uso UML creemos que debemos resaltar debido a que es un método que refleja el trabajo de los métodos de la ONG “MUNDO MEJOR”

La metodología utilizada asegura que los requerimientos del sistema ha sido cubiertos en su totalidad en la propuesta de solución presentada.

Con todo lo mencionado se puede decir que se cumplieron con todos los objetivos planteados en la documentación de este trabajo de grado.

III.2 Recomendaciones

Se recomienda al Administrador tener cuidado necesario, al momento de realizar cualquier tipo de alteración en el sistema por que se guardara un respaldo de Id y Clave por seguridad.

Generar copias de seguridad, claro que es a criterio de el, hacer la copia de seguridad pero se recomienda que se realice al concluir cada gestión.

No dejar que otras terceras personas se adueñen de su Id y Clave por que pueden alterar datos que perjudiquen a la ONG y asi al Sistema.

La actualización del sistema se debe generar el administrador o usuario que tenga los privilegios establecidos por el sistema.

Lo más importante, es que si existiría una desorientación en el llenado de datos en el sistema se recomienda a ir manual de usuario del sistema.

Los beneficiarios deben realizar correctamente sus funciones dentro del sistema, para evitar inconvenientesy posibles conflictos

Capacitar a los beneficiarios dentro un determinado tiempo para el manejo del sistema.

Componente 2

II.2.1. Nombre

Estrategias de Socialización del sistema Informático dirigido a los miembros de la ONG “MUNDO MEJOR”

II.2.2. Introducción

El colaborador que recibe capacitación siente que la empresa lo estima y, por lo tanto, le está asignando un salario espiritual y considera que están invirtiendo en su talento para mejorar su rendimiento, la calidad de su trabajo, elevar su productividad y, consecuentemente, piensa que puede estar próximo a un asenso.

Si bien es cierto que el aumento del salario económico es importante para mejorar la calidad de vida, también es cierto que, pasado cierto período, la nueva remuneración

se diluye en satisfacer ciertas necesidades y, nuevamente, se requiere nuevo aumento; en cambio, el salario espiritual permite mejorar la calidad humana del hombre, coadyuva a la felicidad de su hogar.

El desarrollo de los recursos humanos es central ante el reto tan importante que la empresa enfrenta en este mundo globalizado y competitivo.

La capacitación y desarrollo del recurso humano, es una estrategia empresarial importante que deben acompañar a los demás esfuerzos de cambio que las organizaciones lleven adelante.

Mediante esta estrategia los colaboradores aprenden cosas nuevas, crecen individualmente, establecen relaciones con otros individuos, coordinan el trabajo a realizar, se ponen de acuerdo para introducir mejoras.

La capacitación ha demostrado ser un medio muy eficaz para hacer productivas a las personas, su eficacia se ha demostrado en más del 80% de todos los programas de capacitación.

El presente trabajo tiene como objeto desarrollar los métodos de asociación para la comprensión del sistema por parte de los interesados directos con el proyecto.

El componente socialización va dirigido a los integrantes de la ONG “MUNDO MEJOR”, ah adoptado una forma de llevar a cabo la realización de este componente:

- Mediante un curso de capacitación sobre el uso del Sistema Informático ONG “MUNDO MEJOR”.

II.2.2.1. Concepto de Socialización

La socialización es el proceso por el cual una sociedad se reproduce por sí misma en una nueva generación, es decir, mediante la socialización los valores y la tradición del pasado se continúan y perpetúan. Así, la socialización es lo que proporciona a la

sociedad la continuidad en el tiempo, socializar algo pasa a ser el compartir ese algo con otros para que ellos lo interioricen y lo hagan parte de su ser (cultural).

II.2.2.2. Proceso de Socialización

Es la manera con que los miembros de una colectividad aprenden los modelos culturales de su sociedad, los asimilan y los convierten en sus propias reglas personales de vida.

II.2.2.3. Agentes de Socialización

Se puede decir que la totalidad de la gente de socialización y que cada persona que se entre en contacto de cierto modo es un agente sociabilizado.

II.2.2.4. Propósito

Capacitar a los integrantes de la ONG “MUNDO MEJOR”.

II.2.3. Marco Teórico

II.2.3.1. Comunicación Social

Es un campo de estudio sociológico que explora especialmente aquello relacionado con la información, como es percibida, transmitida y entendida y su impacto social.

II.2.3.1.1. Objetivos

Uno de los principales objetivos de la publicidad es crear demanda o modificar las tendencias de demanda de un producto.

Medios convencionales:

- Anuncios Televisivos
- Anuncios de Radio
- Anuncios de prensa
- Trípticos
- Afiches

II.2.3.2. Propuesta del Proyecto referente a la Socialización

II.2.3.2.1. Planteamiento del Problema

La ONG “MUNDO MEJOR” de un tiempo ahora se ve obligada a incursionar en el ámbito informático por el crecimiento de las tecnologías además de facilitar la información para el conocimiento y su transferencia en el área de derecho de familia. Y los medios que se dan a conocer no se difunden para el conocimiento general, por lo tanto es importante dar a conocer los beneficios que traen las aplicaciones del Sistema.

II.2.3.2.2 Objetivo General

Sociabilizar todas las aplicaciones del Sistema informático ONG “MUNDO MEJOR”.

II.2.3.2.3. Objetivos Específicos

- Brindar información actual de todas las actividades que realicen
- Brindar información sobre el área en la que se trabaja.
- Transferencia de información.

II.2.3.2.4. Métodos de Socialización

Se pretende adoptar este método:

- Estética: Mostrar imágenes llamativas del Sistema Informático
- Humor: Llegar al público mediante frases.

II.2.3.2.5. Público Objetivo

- Los integrantes de la ONG “MUNDO MEJOR”.

II.2.3.2.6. Resultados Esperados

- El manejo del Sistema de los Integrantes de la ONG “MUNDO MEJOR”.

II.2.3.3. Medios de Verificación

Ver anexo del componente 2.

Carta firmada por la Directora de la ONG “MUNDO MEJOR” con la conformidad del sistema.

Ver anexo del componente 2.

III.1 Conclusiones

Con el proyecto se pretende brindar un mejor servicio con calidad y calidez al beneficiario además se mejoro la eficiencia en el desempeño de la ONG “MUNDO MEJOR”.

En la parte administrativa y organizacional, la transferencia de información es mas segura tiene una interactividad con el beneficiario más acorde a las necesidades de los usuarios, se muestra que la ONG con el proyecto tiene un acercamiento a la población mediante esta tecnología por las distintas actividades y foros que dan a conocer.

EL propósito establecido se cumplió, al mejorar la Administración los casos legales de la ONG “MUNDO MEJOR” se identifica que un buen numero atenciones intervenidas son mejor atendidas, se tiene la información de alcance rápido y seguro.

El Sistema desarrollo procesos automatizados para administración y transferencia de información de la ONG “MUNDO MEJOR”. Por medio de fichas de atención al beneficiario las cuales son confidenciales.

Al finalizar los tres meses de la aplicación se observara la interacción entre los beneficiarios y la ONG “MUNDO MEJOR”, a través de la automatización de la información requerida por la institución acordado con la directora de la misma.

La capacitación brindara una interactividad entre las personas vinculadas a la ONG “MUNDO MEJOR” y el sistema informático.

La capacitación impulsara a buscar nuevos recursos humanos en beneficio de la ONG.

A partir de la capacitación el personal de la ONG podrá socializar el proyecto por medio de spots televisivos, radio, prensa escrita, lo cual beneficiara al proyecto.

Las tecnologías utilizadas para el desarrollo del sistema, fueron las adecuadas para este tipo de aplicaciones, ya que java (programación orientada a objetos) es un código abierto y no necesita de alguna licencia para utilizarla, también el gestor de base de datos Postgres y el servidor Web Apache. Lo mismo con ajax se utilizo para optimizar el rendimiento y carga de las páginas logrando una mejor interacción con el usuario.

También se demuestra aunque que el uso adecuado de las metodologías, consiguen un objetivo sustancioso, en este caso, el uso UML creemos que debemos resaltar debido a que es un método que refleja el trabajo de los métodos de la ONG “MUNDO MEJOR”

La metodología utilizada asegura que los requerimientos del sistema ha sido cubiertos en su totalidad en la propuesta de solución presentada.

Con todo lo mencionado se puede decir que se cumplieron con todos los objetivos planteados en la documentación de este trabajo de grado.

III.2 Recomendaciones

La institución requiere la actualización, observando los avances en las mejoras del sistema informático se buscara implementar en otras áreas como se área civil, área penal, temas de acceso a la justicia, área de salud, Psicológica y trabajo social para el beneficio de la ONG “MUNDO MEJOR”, que se busque una manera de interactividad entre otras instituciones que formen parte de la red de derechos humanos para no tener una doble atención o re victimización.

Los casos legales de la ONG “MUNDO MEJOR” se administrara de una manera más eficiente, segura y con datos fidedignos.

A largo plazo se lograra que el sistema informático sea visible, interactuable y comunicable entre el visitante, beneficiario, abogado, administrador y personal de la ONG “MUNDO MEJOR” para que se tenga una conexión mediante la WEB, ya q la web es medio de comunicación actual y una tecnología global.

Se Recomienda que a partir de la capacitación al personal ONG “MUNDO MEJOR”, el sistema se lo socialice mediante spots televisivos, radio, prensa escrita con un conocimiento y manipulación del Sistema Informático, para un mejor contacto con la población en general y su uso del mismo.

Se recomienda a los estudiantes que opten esta tesis como bibliografía para sus próximos trabajos de grado, también que enfatizen la coordinación entre el árbol de objetivos, la matriz de marco lógico y su documentación dado cuenta que todo tiene que coincidir, además de tener un medio de verificación que respalde.

Se recomienda al Administrador tener cuidado necesario, al momento de realizar cualquier tipo de alteración en el sistema por que se guardara un respalda de Id y Clave por seguridad.

Generar copias de seguridad, claro que es a criterio del administrador, hacer la copia de seguridad pero se recomienda que se realice al concluir cada gestión.

No dejar que otras terceras personas se adueñen de su Id y Clave por que pueden alterar datos que perjudiquen a la ONG y por siguiente al Sistema.

La actualización del sistema se debe generar el administrador o usuario que tenga los privilegios establecidos por el sistema.

Lo más importante, es que si existiría una desorientación en el llenado de datos en el sistema se recomienda a ir manual de usuario del sistema.

III.3. Bibliografía

<http://www.rinconespecial.com>

<http://www.geocites.com/orientacionderecho/public/evaluacion>

<http://www.monografias.com/sobre-internet>

<http://www.cje.org/inice/pcinet/conecto.htm>

<http://www.java.org/servlets.htm>

<http://www.monografias/trabajos>

<http://www.informatizate.com>

[http://www.metodologias de desarrollo de software.html](http://www.metodologias_de_desarrollo_de_software.html)

<http://www.pergaminovirtual.html>

<http://www.carebolivia.org/default.asp>