

CAPITULO I

EL PROYECTO

I.EL PROYECTO

I.1 PRESENTACIÓN DEL PROYECTO

I.1.1TÍTULO

MEJORAR LA GESTIÓN DEL ÁREA DE ARCHIVOS DE LA O. M. A. DEL
GOBIERNO MUNICIPAL AUTÓNOMO DE SAN LORENZO.

I.1.2 PERSONAL VINCULADO AL PROYECTO

I.1.2.1 Director de Proyecto

Apellido Paterno	Apellido Materno	Nombre	C.I.
	Castillo	María de los Ángeles	7118838
Profesión	Carrera ó Unidad	Facultad: Ciencias y Tecnología	
Estudiante	Ing. Informática		
Teléf. Oficina	Celular	Correo electrónico	Firma
	60257877	Mari_m_123_@hotmail.com	

Tabla I.1director del proyecto

I.1.2.2 Participantes equipo de trabajo (**señale categoría: investigador, investigador junior, asesor, etc.)**

Participantes Equipo de Trabajo

Categoría	Nombres y Apellidos	Profesión	C.I.	Firma
Director	Maria de los Angeles Castillo	Estudiante	7118838	
Investigador	Maria de los Angeles Castillo	Estudiante	7118838	
Investigador Jr.	Maria de los Angeles Castillo	Estudiante	7118838	
Programador	Maria de los Angeles Castillo	Estudiante	7118838	

Tabla I.2Participantes equipo de trabajo

I.1.2.3 Equipo de trabajo de: Empresas/Instituciones/Organizaciones participantes/cooperantes.

Nombre: Gobierno municipal autónomo de san Lorenzo			
Dirección: San Lorenzo		Teléf. Oficina:	
Nombre y Apellidos	Cargo	C.I.	Firma
ING. HECTOR ARENAS	Encargado del Dpto. de sistemas de la Alcaldía Municipal de San Lorenzo	5047028 tja	
LIC. ELIZABETH CASTRO	Docente de la carrera de Ing. Informática.		

Tabla I.3Equipo de Trabajo

I.1.2.4 ACTIVIDADES PREVISTAS PARA LOS INTEGRANTES DEL EQUIPO DE INVESTIGACIÓN

Responsable *	Actividades
➤ María de los Ángeles Castillo.	Seguimiento y Control del Proyecto ➤ Planificación y control del cronograma del proyecto. ➤ Asignar y gestionar recursos y prioridades a los componentes y actividades del proyecto. ➤ Coordinar las iteraciones entre el equipo de trabajo y los usuarios del proyecto. ➤ Mantener al equipo enfocado en los objetivos del proyecto. ➤ Establecer un conjunto prácticas que aseguren la calidad e integridad del proyecto. Desarrollo de los Componentes ➤ Realizar el análisis y diseño del proyecto. ➤ Capturar la especificación y validación de requisitos interactuando con los usuarios. ➤ Elaboración del Análisis, Diseño y Desarrollo del Sistema. ➤ Elaboración del Modelo de datos. ➤ Diseño de la Base de Datos del Sistema.

	➤ Programación del Sistema. ➤ Preparación de Pruebas Funcionales del Sistema. ➤ Planificar el contenido temático de los cursos de capacitación ➤ Formar al personal en el uso de las TIC para el manejo del producto final. ➤ Construcción del producto final.
--	--

Actividades previstas para los integrantes del equipo de investigación

Tabla I.4 Actividades

I.2 Descripción del Proyecto

I.2.1 Resumen Ejecutivo del Proyecto

Las tecnologías de la información y la comunicación (TIC) son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de y que se integran a un sistema de información interconectado y complementario.

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

El manejo de la información que se realiza en la O.M.A. (Oficialia Mayor Administrativa) del Gobierno Municipal de San Lorenzo es manual, debido a esta situación el proceso de administrar la información se torna complicado y moroso. Otro aspecto a mencionar es que al ser la información manejada manualmente

resulta difícil y lento poder encontrar la información que se solicita y puede que se de el caso de que la información no se encuentre completa.

Por lo que el proyecto pretende realizar un sistema para mejorar el registro de archivos de la O.M.A. (Oficialía Mayor Administrativa), que apoyara directamente a todo el personal que trabaja en la O.M.A. y usuarios que lo necesiten que sean empleados en el gobierno municipal autónomo de san Lorenzo los cuales contarán con un servicio eficiente y podrán realizar sus actividades de una forma más segura y confiable.

También se realizará una capacitación para que los usuarios puedan conocer la manera correcta del manejo del sistema instalando el sistema para todos los usuarios en sus respectivas computadoras (se realiza la práctica de la capacitación en 5 equipos) , y así de esa forma puedan conocer todo el manejo del sistema y no se les pueda presentar problemas al momento de usarlo, en el cual también los usuarios tienen su propio usuario y contraseña y puedan saber el funcionamiento del sistema y todas las tareas que pueden realizar . Y terminada la capacitación se les otorgara un certificado para las personas que asistan a la capacitación.

I.2.2 Descripción y Fundamentación del Proyecto (qué y por qué)

El presente proyecto propone en primer lugar el desarrollo de un sistema automatizado para la administración de información que se registra en la oficialía mayor administrativa del gobierno municipal autónomo de san Lorenzo, en segundo lugar el desarrollo de una capacitación para el correcto uso del sistema.

Desarrollar un sistema de gestión para el registro de archivos de la oficialía mayor administrativa (O.M.A.)

Este sistema permitirá mejorar significativamente la administración de información que se registra como así también ayudara a controlar el orden el de los archivos.

Este sistema contribuirá a la eliminación de la mayoría de los problemas que enfrenta actualmente la oficialía mayor administrativa, entre lo que podemos mencionar es que causa de que la información es manejada manualmente el

tratamiento de la misma se torna difícil y morosa, y por tal motivo no se puede tener una perspectiva clara sobre la ubicación de cada archivo que se guarda, otro problema que podemos destacar es que a causa del tratamiento manual de información, no se puede contar con una información disponible en cualquier momento que se lo necesita.

El proyecto tiene como uno de sus objetivos mejorar la administración de Compras, Empleados, Inventarios, por la simple razón de que la mayoría de los casos presenta pérdida de datos al efectuar el registro de estas operaciones, lo que puede llevar a una información errada y no encontrar todos los datos que se necesitan.

El proyecto también pretende agilizar la información necesaria para la OMA, esto a causa de la demora que existe al momento de realizar los reportes.

El proyecto ofrece mayor rapidez y control acerca de qué materiales necesitan ser registrados, ya que se cuenta con un almacén actualizado en todo momento.

Capacitación sobre el uso del sistema al personal de la oficialía mayor administrativa.

La capacitación sobre el uso adecuado del Sistema de escritorio para la oficialía mayor administrativa, permitirá a los miembros de la oficialía mayor administrativa interactuar de manera satisfactoria con el sistema.

Se planeó esta capacitación porque es de vital importancia instruir a los usuarios en el uso adecuado del sistema desarrollado.

I.2.3 SISTEMA DE MARCO LÓGICO

TRABAJO QUE DESEMPEÑA EL EQUIPO DE LA OFICINA MAYOR ADMINISTRATIVA (O.M.A.) DE LA ALCALDIA DE SAN LORENZO

ANÁLISIS DE INVOLUCRADO

GRUPO	INTERESES	PROBLEMAS PERSISTIDOS	RECURSOS Y MANDATOS
Administración de la O.M.A.	➤ Tener un sistema informático eficiente, que brinde información confiable. ➤ Facilitar los registros de todas las compras y entradas y salidas de almacén.	➤ Mucho tiempo en la búsqueda de información referente a las compras que se realizan.	R: Disponibilidad de la información de las compras. M: Contribuir con el mejoramiento de la organización de los archivos.
Personal	➤ Tener un Sistema de Control de Personal confiable, seguro y que garantice la veracidad de sus datos.	➤ El registro de información no es confiable. ➤ Pérdida de información de registros del personal.	R: Disponibilidad de información personal para contribuir con el desarrollo del sistema.
Encargado del Proyecto	➤ Contar con toda la información referente al registro de los archivos y conocer los procedimientos para realizar una compra.	➤ El registro de información no es confiable. ➤ Pérdida de documentos. ➤ Mal manejo de documentos referente al registro de las compras.	R: Desarrollar un sistema de control de las compras que facilite el manejo de la información y ayude en la toma de decisiones en el área Administrativa.

Tabla I.5 Análisis de Involucrados

I.2.4 ANÁLISIS DE CAUSAS DEL PROBLEMAS

Árbol de Problemas

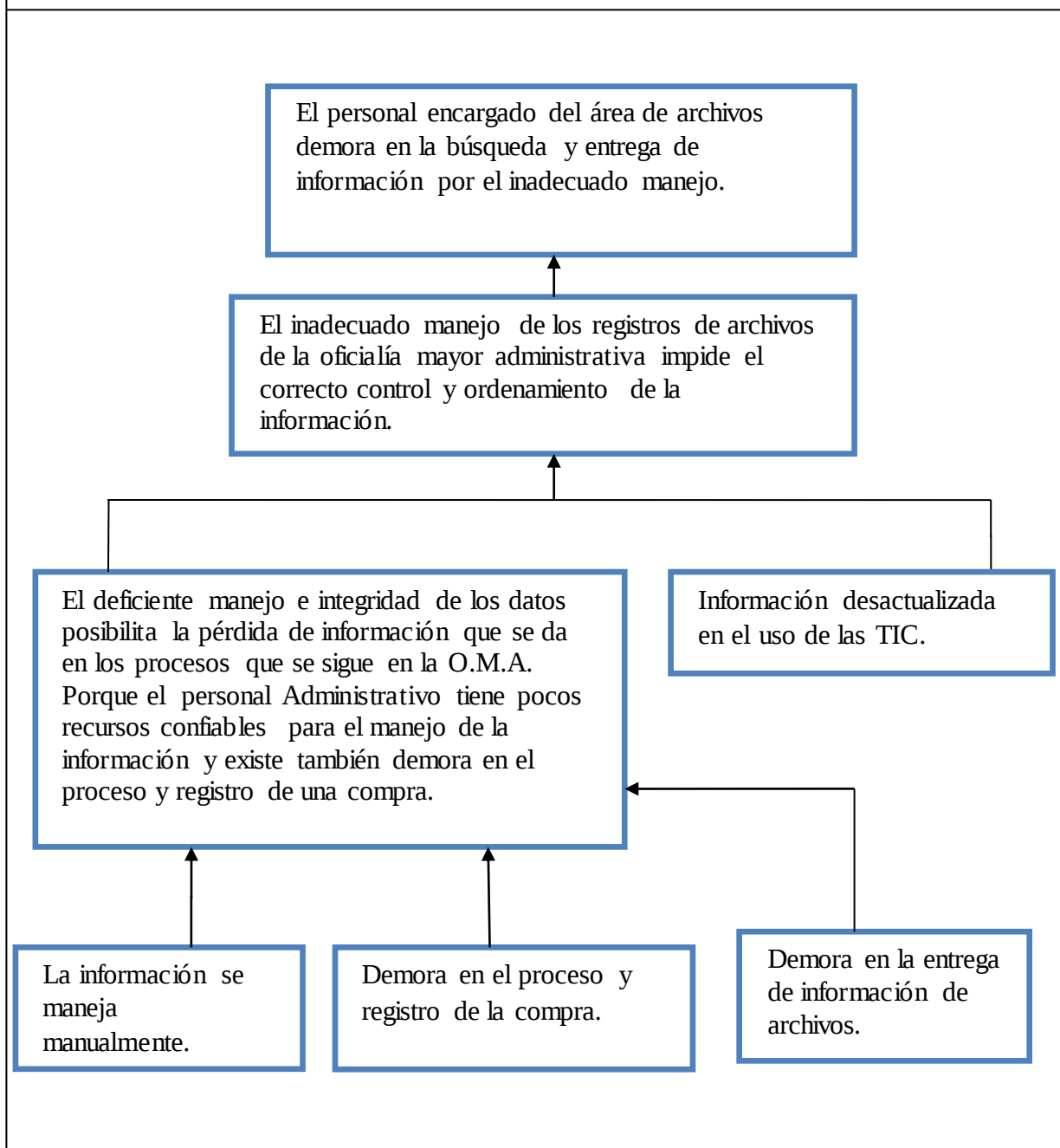


Tabla I.6 Arbol de problemas

I.2.5 ANÁLISIS DE OBJETIVOS

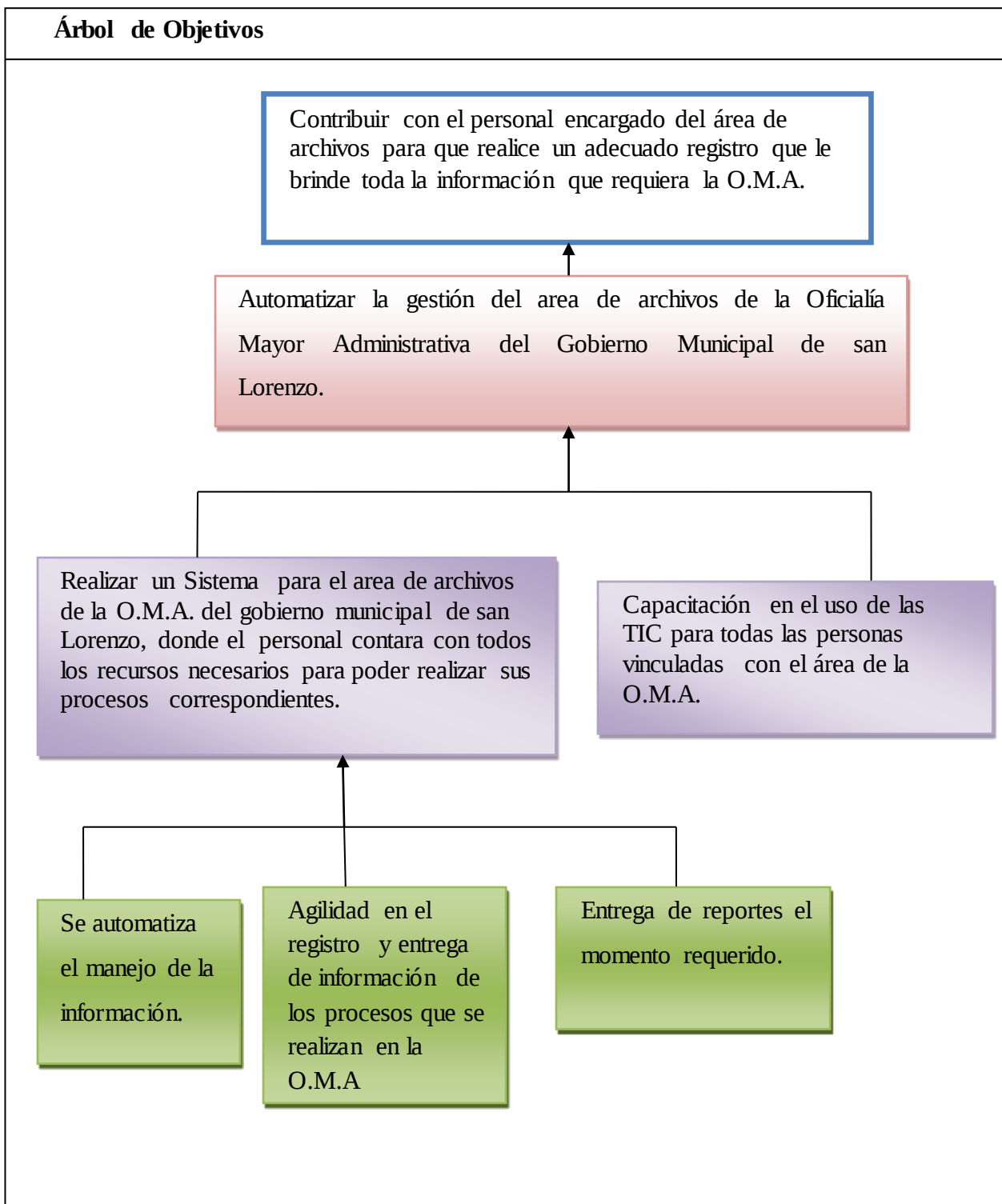


Tabla I.7 Árbol de Objetivos

I.2.6 Situación planteada Con y Sin Proyecto

Situación sin proyecto	Situación con proyecto
Inadecuada ordenación de los archivos por lo que se hace moroso poder acceder a la información que se busca. Falta de organización y entrega de información en el tiempo requerido.	La oficialía mayor administrativa de la alcaldía municipal de San Lorenzo podrá disponer de la información de los archivos el momento que lo necesite. Los usuarios cuentan con un ordenamiento y acceso a la información sobre todos los registros que se archivan en el momento que lo soliciten.

Tabla I.8 Situación sin y con proyecto

I.2.7 OBJETIVOS

I.2.7.1 Objetivo General

Automatizar la gestión del área de archivos de la Oficialía Mayor Administrativa del Gobierno Municipal de San Lorenzo.

I.2.7.2 Objetivos Específicos

1.-Modulo I

- Identificar los requerimientos del usuario de la Oficialía Mayor Administrativa
- Utilizar las herramientas de UML, en toda la fase del desarrollo del sistema
- Utilizar la metodología RUP (Proceso Unificado Racional) para el desarrollo del software.
- Modelar casos de uso para el sistema en funcionamiento.
- Diseñar un buen entorno gráfico del sistema tomando en cuenta los requerimientos de los usuarios.

2.-Modulo II

- Capacitación para todas las personas vinculadas con el área de la O.M.A.
- Se entregara certificado de capacitación a los usuarios del sistema que sean capacitados.

I.2.7.3 ALCANCES

En esta documentación se presenta un informe general del desarrollo del software, el cual contiene una descripción de requerimientos del sistema actual presentando modelos para la solución de los problemas que se presenta en el área de registro archivos de la “O.M.A.”, además de incorporar nuevas características que involucren a los actores principales que manejaran el sistema una vez terminado, Nos estamos basando en lo más principal en la captura de información que se tiene de las entrevistas.

En cada Iteración nos basamos con mucho mas criterio a lo que se desea lograr para futuro añadir nuevas características mucho más claras y objetivas según al desenvolvimiento de cada integrante “del equipo de desarrollo”.

El producto final o software será diseñado para funcionar dentro de la honorable alcaldía municipal de san Lorenzo y en especialmente en el área de archivos de la oficialía mayor administrativa, el cual presenta las siguientes características.

Presentará un entorno gráfico, que sea amigable al usuario para el buen funcionamiento del sistema.

Ayudará a tener un mejor control y almacenamiento ordenado de la información generada dentro del área de archivos de la oficialía administrativa.

Control de los registros de compras.

Se realizara un proceso para la búsqueda de las compras que se realicen.

La parte de Administración de usuarios: Se toma en cuenta las altas bajas y 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 altas, bajas y modificaciones de los roles, donde solo el administrador tiene permiso de realizar dichas acciones.

La parte de Administrar Compras: Se toma en cuenta los procesos que hace para registrar una compra, el administrador y dichas acciones incluye el id o código de compra, fecha, hora, forma de cotización, numero de factura, y estado.

La parte de Administrar Cotización: Se podrá ver todas las cotizaciones que se registren en el sistema.

La parte de Administrar Aprobaciones: se podrá consultar todas las solicitudes aprobadas y las solicitudes que fueron rechazadas.

Generar Reportes de las compras que se realiza, entrada de material, salida de material, control de material que existe en almacén.

I.2.7.4 LIMITACIONES

- El sistema no soporta un módulo contable.
- El sistema será implementado solo en la plataforma Windows.
- El sistema no emitirá facturación electrónica.
- El sistema no se actualizara.
- El sistema será diseñado según los requerimientos de la oficialía mayor administrativa del gobierno municipal de san lorenzo.

I.2.8 MATRIZ MARCO LÓGICO DEL PROYECTO

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
<u>Fin</u> Contribuir con el personal encargado del área de archivos para que realice un adecuado registro que le brinde toda la información que requiera la O.M.A.	A un año luego de finalizar el proyecto, se podrá observar que el objetivo se cumple de manera eficiente en un 70% en la O.M.A. Categoría: Calidad Expresión matemática: El 70% de los trabajadores de la O.M.A. califican de exitoso el proyecto.	■ Informes de los encargados de archivos realizados cada año.	■ El alcalde de san Lorenzo mantiene el apoyo humano y financiero a la incorporación del sistema para el municipio.
<u>Objetivo General</u> (Propósito) Automatizar la gestión del area de archivos de la Oficialía Mayor Administrativa del Gobierno Municipal de san Lorenzo.	Al finalizar el proyecto, el sistema brinda informes precisos de los procesos automatizados en nomas de 10 segundos de haberse ejecutado la consulta. Mejorando asi el proceso que sigue el area de archivos de la OMA del	Información del Área de Estadística(Oficialia Mayor Administrativa) Indicando que ha mejorado el proceso de registro de información y obtención de reportes del registro de	■ Predisposición del personal de la alcaldía para utilizar sistemas de información. - El personal de la alcaldía brindara toda la información necesaria para el desarrollo del

	gobierno municipal de san Lorenzo. El de obtención de reportes mejorado, reduciendo el tiempo en un 60% Categoría: Calidad ((N° de Funciones en el control de Archivos afectados por el proyecto en el área Administrativa/total de los reportes requeridos)* 100)	compras y almacén.	proyecto.
<u>Objetivos Específicos</u> (Componentes) 1.- Sistema para la gestión del área de archivos de la oficialía mayor administrativa	En Julio del 2013 se concluye el desarrollo del SIECAM se ha concluido el desarrollo del Sistema automatizado para el registro del área de archivos de	1.1.1 Documentación derivada del desarrollo del sistema. 1.2.1 Principales reportes producidos por el sistema 1.2.2 Manuales de usuario.	a) El área de la oficialía administrativa de la alcaldía aprueba ofrecer información para la elaboración del sistema de escritorio para la administración de todos los

desarrollado. 2.- Capacitación para los usuarios vinculados con el Sistema.	la oficialía mayor administrativa de acuerdo con las especificaciones de requerimientos identificados. Al finalizar el proyecto se ha elaborado el manual de configuración e instalación del Sistema. Al finalizar el proyecto se han desarrollado talleres de capacitación. Ver anexo capacitación	1.2.3 Carta de conformidad por parte de la alcaldía municipal de san Lorenzo. 2.2.3 Fotografías tomadas durante las capacitaciones. 2.2.5 Certificado de capacitación del sistema para la administración de todo el personal de la oficialía mayor administrativa, firmados por el director del proyecto.	registros que se realizan. b) El personal del área de la oficialía administrativa con disponibilidad e interés en capacitarse en el uso del sistema.
---	---	---	---

<u>Actividades</u> 1.1 Especificación de requerimientos 1.2. Analisis y Diseño del Sistema. 1.3. Programación 2.1. Elaboración de documentación 2.2 Manual de usuario 2.3 Manual de instalación. 2.4 Capacitación 2.5 desarrollo de Capacitación	Presupuesto	Documento del análisis y diseño del producto. Documento impreso del manual de usuario.	Actividades realizadas de acuerdo al cronograma.
--	--------------------	--	---

Tabla I.9 Matriz de Marco Logico

I.2.9 METODOLOGÍA DE TRABAJO

Se desarrolla la metodología RUP por presentar múltiples ventajas y presentar tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

Fases de RUP:

Inicio

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto.

Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20 % del modelo completo).

Los objetivos esta fase son:

- Establecer el ámbito del proyecto y sus límites.
- Encontrar los casos de uso críticos del sistema.
- Estimar el coste en recursos y tiempo de todo el proyecto.

Elaboración

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final.

Construcción

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones.

Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

Transición

La finalidad de esta fase es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas con el ajuste, configuración, instalación y facilidad de uso del producto.

I.2.10 Descripción y Relación de las Estrategias con los Objetivo

Estrategias	Objetivos Específicos
Mejora el manejo de información Usuarios satisfechos con el sistema.	*Identificar los diferentes requerimientos del usuario de la “Oficialía Mayor Administrativa”. *Diseñar un sistema automatizado para la administración de archivos de la oficialía mayor administrativa del gobierno municipal autónomo de san Lorenzo. *El sistema será interactivo y tendrá un entorno amigable y de fácil manejo para que cualquier usuario.
Entrega oportuna de reportes. Mayor información disponible Reducción del tiempo en la búsqueda de información Entrega inmediata de información que se solicite.	*Utilizar la metodología RUP (Proceso Unificado Racional) para el desarrollo del Software. *Proveer mayor seguridad en información que se maneja dando así acceso al sistema de manera rápida y segura. *El acceso y manejo se dará mediante una base de datos.
Facilitar al usuario el uso del nuevo sistema que se habilita.	*Capacitar al personal involucrado del sistema automatizado.

Tabla I.10 Estrategias y Objetivos Específicos

I.2.11 Cronograma de Actividades

N°	Actividad	N° días	M1	M2	M3	M4	M5	M6	M7	M8
1	Desarrollo del Sistema para la administración de registros para la oficialía administrativa del gobierno municipal autónomo de san Lorenzo.	200 días	X	X	X	X	X	X	X	X
2	Especificación de requerimientos	90 días	X	X	X					
3	Analisis y diseño	60 días				X	X			
4	Modelos de Casos de Uso	30 días				X				
5	Modelo de Datos	30 días					X			
6	Modelo de Implemetación	90 días						X	X	X
7	Prototipo de Interfaz de Usuario	30 días						X		
8	Programación	90 días						X	X	X

Tabla I.11 cronograma de actividades

I.2.12 Resultados esperados

Objetivos específicos	Resultados Esperados
• Identificar los requerimientos del usuario de la Oficialia Mayor Administrativa • Utilizar las herramientas de UML, en toda la fase del desarrollo del sistema • Utilizar la metodología RUP (Proceso Unificado Racional) para el desarrollo del software. • Modelar casos de uso para el sistema en funcionamiento. • Diseñar un buen entorno grafico del sistema tomando en cuenta los requerimientos de los usuarios. • Proveer mayor seguridad en información que se maneja dando así acceso al sistema de manera rápida y segura. • El acceso y manejo se dará mediante una base de datos. • Capacitar al personal involucrado del sistema automatizado.	➤ En julio del 2013 se pondrá en funcionamiento el sistema para el uso de todos los usuarios que lo requieran. ➤ Mejoramiento en el manejo de información. ➤ Buena aceptación del sistema a utilizar. ➤ Nuevo sistema se pone en marcha en julio del 2013 el funcionamiento de este es adecuado a las exigencias de los usuarios. ➤ La información que manejan los con el sistema tiene un buen acceso, esto lo hace precisa y eficaz. ➤ Usuarios satisfechos con el nuevo sistema. ➤ Usuarios manipulan el sistema de una manera eficaz. ➤ El personal maneja el sistema con facilidad sin ninguna deficiencia.

Tabla I.12 Resultados Esperados

I.2.13 TRANSFERENCIA DE RESULTADOS

a) Medios y estrategias para la transferencia de resultados.

- La transferencia de datos se realizara mediante
- Convenio con la universidad.

I.3. Presupuesto / Justificación

h) SUB GRUPO 32000. Descripción de los gastos de Productos de Papel, Cartón e Impresos

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
32400	Manual de Usuario	6	30	180
	Manual de Instalación	6	30	180
Total				360

Tabla I.13 Papel, Carton e impresos

* Se refiere a la adquisición de; papel y cartón en sus diversas formas y clases,

CAPITULO II

COMPONENTES

II. COMPONENTES

II.1 COMPONENTE 1:

MEJORAR LA GESTIÓN DEL ÁREA DE ARCHIVOS DE LA O. M. A. DEL GOBIERNO MUNICIPAL AUTÓNOMO DE SAN LORENZO.

II.1.1 PLAN DE DESARROLLO DE SOFTWARE

II.1.1.1 Introducción

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

El proyecto ha sido ofertado por la universitaria María de los Angeles Castillo basado en la metodología de Rational Unified Process (RUP) en la que únicamente se procederá a cumplir con las tres primeras fases, las cuales marcan la metodología. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio, Elaboración y Construcción.

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

Como el área de archivos de la oficialía mayor administrativa de la alcaldía de San Lorenzo, en la actualidad realiza todas sus tareas manualmente, el presente proyecto contribuye al mejoramiento de la gestión de información en el área de archivos de la oficialía mayor administrativa de la alcaldía de San Lorenzo. Se pretende desarrollar un sistema de información para controlar las compras y almacenes de la

O.M.A., el mismo podrá facilitar el acceso a la información, tener un registro completo, verídico y eficiente del personal.

El proyecto de la misma forma contribuye al propósito del mejoramiento de la difusión de la información que cuenta la O.M.A., a través de un sistema automatizado, de esta manera permitirá medir el éxito que tiene como institución social.

II.1.1.2 Propósito

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

Los usuarios del Plan de Desarrollo del Software son:

El director responsable del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.

Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

II.1.1.3 Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del Proyecto titulado “Mejorar la gestión del área de archivos de la “O.M.A.” El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma separada. Lo que se pretende alcanzar con este plan es documentar las mejoras planteadas el control de la información, manejada en la Sección de Archivo como también en las secciones relacionadas con esta, para que sus actividades cotidianas relacionadas con las solicitudes de servicios internos a la Sección de Archivo sean de la mejor manera posible.

Posteriormente el avance del proyecto y seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

II.1.1.4 Resumen

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

Vista General del Proyecto — proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto — describe la estructura organizacional del equipo de desarrollo.

Gestión del Proceso — explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

Planes y Guías de aplicación — proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

II.1.2 VISTA GENERAL DEL PROYECTO

II.1.2.1 PROPÓSITO, ALCANCE Y OBJETIVOS

Propósito

- Contribuir de manera eficiente y eficaz en el Control adecuado de la información en el área de archivos de la Oficialía Mayor Administrativa (O.M.A.), que permitirá otorgar información rápida y confiable.

Alcance

- El sistema MGAOMA utilizará el gestor de base de datos PostgreSQL para almacenar toda la información generada, esto debido a que tiene licencia libre y permitirá una inmediata implementación del sistema, sin traer problemas o inconvenientes a la institución.
- El acceso al sistema MGAOMA es de modo web.

- Los usuarios ingresaran al sistema MGAOMA al mediante códigos de usuario y clave, esto para brindar responsabilidades a los usuarios y seguridad al sistema MGAOMA.
- El sistema MGAOMA .automatizará los registros solicitudes, cotizaciones, aprobaciones, compras, almacenes y salidas y entradas de material se podrá generar reportes de toda la información que se considere necesaria.

Objetivos

Objetivo General

Analizar, Diseñar y Desarrollar un sistema automatizado que ayude a controlar los registros del área de la Oficialía Mayor Administrativa de la alcaldía de San Lorenzo.

Objetivos Específicos

1.-Modulo I

- Identificar los requerimientos del usuario de la Oficialia Mayor Administrativa
- Utilizar las herramientas de UML, en toda la fase del desarrollo del sistema
- Utilizar la metodología RUP (Proceso Unificado Racional) para el desarrollo del software.
- Modelar casos de uso para el sistema en funcionamiento.
- Diseñar un buen entorno grafico del sistema tomando en cuenta los requerimientos de los usuarios.

2.-Modulo II

- Capacitación para todas las personas vinculadas con el área de la O.M.A.
- Se entregara certificado de capacitación a los usuarios del sistema que sean capacitados.

II.1.2.2 SUPOSICIONES Y RESTRICCIONES

Suposiciones

- El personal involucrado apoya de manera constante y voluntaria, proporciona toda la información necesaria para un correcto desarrollo y evolución del sistema MGAOMA.
- El equipo de trabajo cuenta con todas las herramientas necesarias para la elaboración del sistema MGAOMA.

Restricciones

- El sistema MGAOMA funcionará en la plataforma Windows.
- La institución va trabajando con Windows XP desde hace años, debido a esto el sistema se desarrolló para trabajar en la plataforma actual que alojará al sistema MGAOMA.
- El sistema MGAOMA por lo menos debe contar con un Administrador (Oficial Mayor), un Encargado del sistema para cumplir con su objetivo.

II.2. ENTREGABLES DEL PROYECTO

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

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

II.2.1. MODELO DE CASOS DE USO DEL NEGOCIO

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

II.2.1.1. DIAGRAMA DE CASOS DEL NEGOCIO

Modelo del caso de uso del negocio: Secretaria

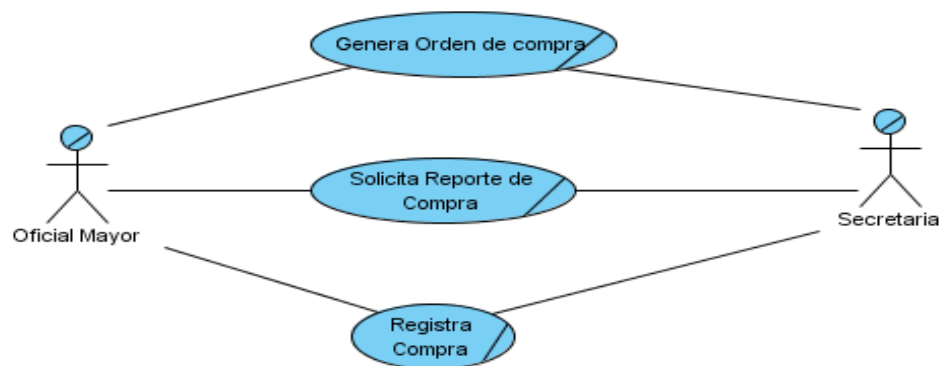


Fig.II.1 modelo de negocio-Secretaria

Modelo del caso de uso del negocio: Encargado de Solicitud

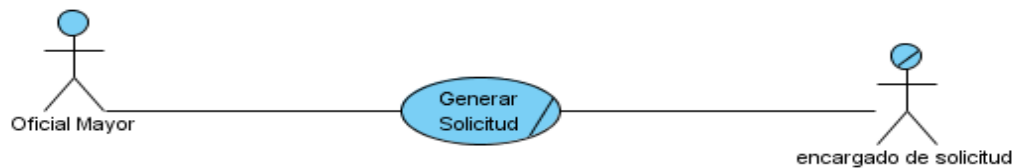
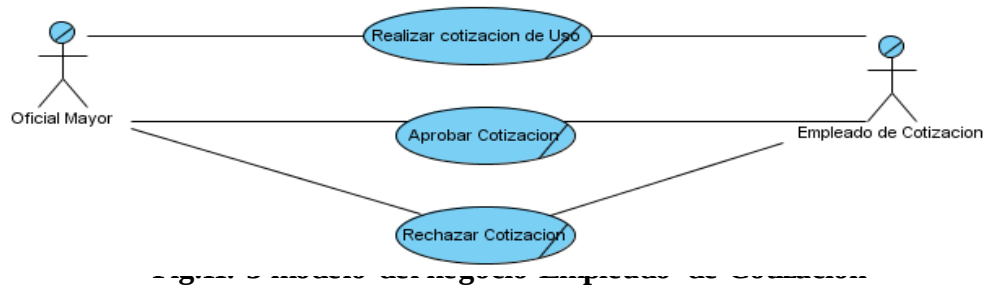


Fig.II. 2 modelo de negocio-Solicitud

Modelo del caso de uso del negocio: Empleado de Cotización



Modelo del caso de uso del negocio: Encargado de Aprobación

Fig.II. 4 modelo de negocio-



Modelo del caso de uso del negocio: Empleado de Almacén

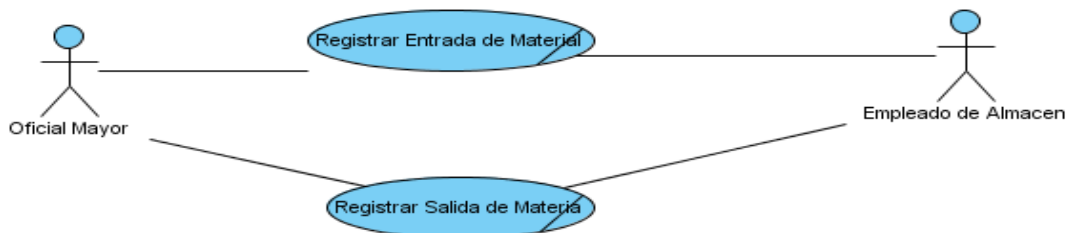


Fig.II. 5 modelo del negocio-Empleado de Almacén

Modelo del caso de uso del negocio: Auxiliar Contable

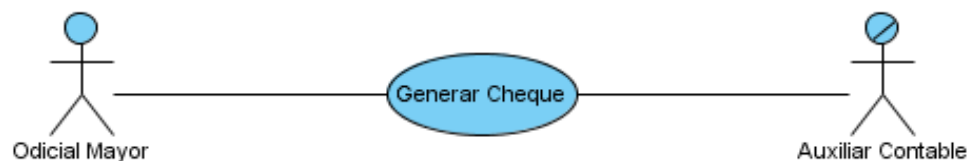


Fig.II. 6 modelo del negocio-Auxiliar Contable

II.2.2 GLOSARIO

II.2.2.1 INTRODUCCIÓN

Este documento recoge todos y cada uno de los términos manejados a lo largo de todo el proyecto de desarrollo del sistema para el control de archivos de la Oficialía Mayor Administrativa de la alcaldía municipal de san Lorenzo. Se trata de un diccionario informal de datos y definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para el proyecto.

II.2.2.2 PROPÓSITO

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto en desarrollo. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores del proyecto.

II.2.2.3 ALCANCES

El alcance del presente documento se extiende a la sección del área de archivos de la oficialía mayor administrativa de la alcaldía de san Lorenzo.

II.2.2.4 REFERENCIAS

- Plan de desarrollo de software
- Diagrama de casos de uso
- Casos de prueba

II.2.2.5. ORGANIZACIÓN DEL GLOSARIO

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según el alfabeto.

II.2.2.6. DEFINICIONES

- **Base de datos.-** Sistema gestor de base de datos en el cual se guarda toda la información del sistema.
- **Informática.-** Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento de la información por medio de ordenadores.
- **NTIC.-** Nueva tecnología de información y de la comunicación
- **Sistema web.-** se dice sistema web a un programa informático de información que trabaja el internet
- **Tecnología.-** Conjunto de teorías y técnicas que permiten el aprovechamiento practico del conocimiento científico, en nuestro caso tecnología de computación.

II.2.2.7. ESTEREOTIPOS UML

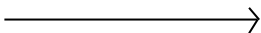
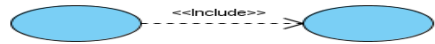
Caso de uso del negocio	
Caso de uso	
Comunicación	
Relación de extensión	
Relación e inclusión	
Actor del negocio	
Actor	

Tabla II.1 Diagramas UML

II.2.3. DIAGRAMAS DE CASOS DE USO

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

INTRODUCCIÓN

El caso de uso del documento narrativo que describe la secuencia de eventos de un actor que utiliza un sistema para completar el proceso, los casos de uso son historias o casos de utilización de un sistema; no son exactamente los requerimientos ni las especificaciones funcionales, sino que ejemplifican e incluyen tácitamente los requerimientos en las historias que narran.

PROPOSITO

- Modelar el contexto del sistema
- Modelar los requerimientos del sistema
- Identificar los requerimientos los procesos del sistema
- Es una excelente herramienta para estimular a que los usuarios potenciales hablen del sistema desde sus propios puntos de vista.
- Involucrar a los usuarios en las etapas iniciales del análisis y el diseño del sistema.
- Nos ayuda a obtener los requerimientos desde el punto vista del usuario.

ALCANCE

- Describe lo que el sistema de software realizara dentro del negocio
- Describe los alcances del sistema
- Describe los procesos del sistema y los usuarios.

II.2.2.3.1. DIAGRAMA DE CASOS DE USO GENERAL

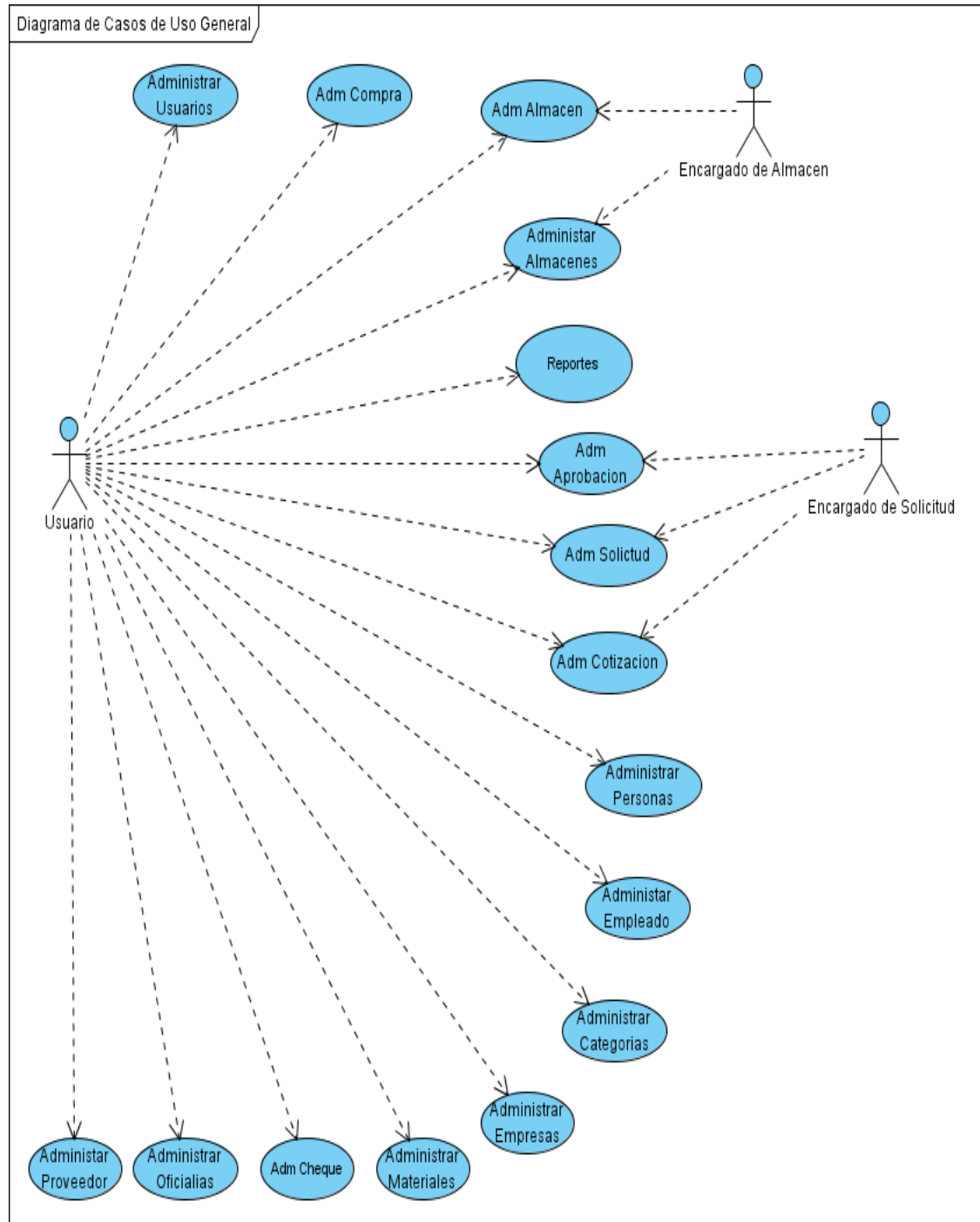


Fig.II. 7 Diagrama de Casos de Uso General

IDENTIFICACION DE ACTORES

- **Actor –administrador del sistema**

El administrador tiene la responsabilidad de administrar a todos los usuarios del sistema administrar las compras y generar los reportes y puede ver todos los módulos del sistema.

- **Actor – Encargado de solicitud**

El encargado de solicitud tiene la responsabilidad de administrar todo lo que es solicitudes, cotizaciones y aprobaciones que realiza el sistema.

- **Actor – Encargado de almacén**

El encargado de almacén tiene la responsabilidad de administrar todo lo que es la entrada y salidas de almacenes que realiza el sistema y generar e imprimir reportes.

II.2.2.3.2. MODELOS DE CASOS DE USO

Caso de uso: Ingresar al Sistema

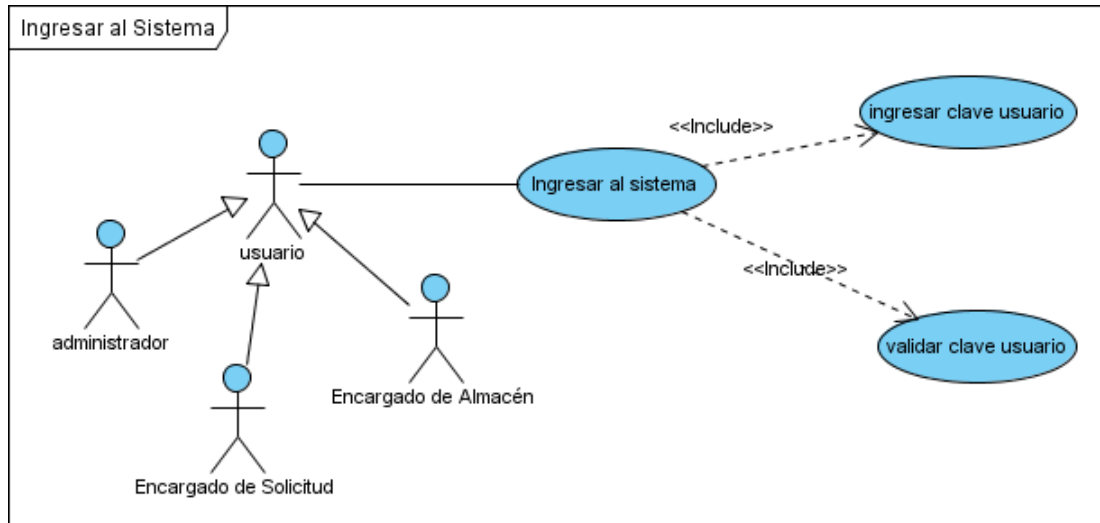


Fig. II.8 caso de uso- Ingresar al Sistema

Caso de uso: Administrar Usuarios

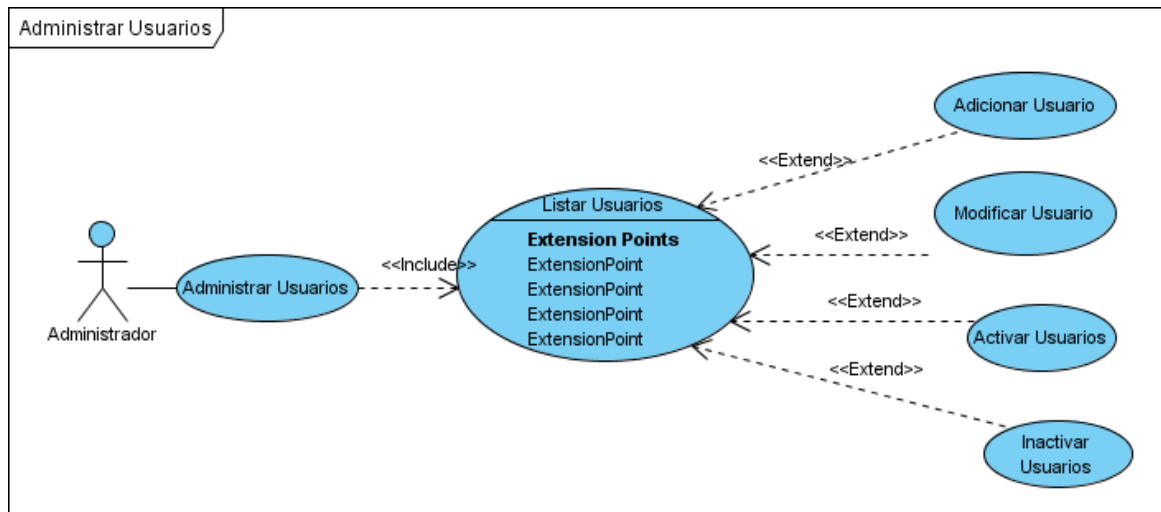


Fig. II.9 caso de uso- Administrar Usuarios

Casos de Uso: Administrar Personas

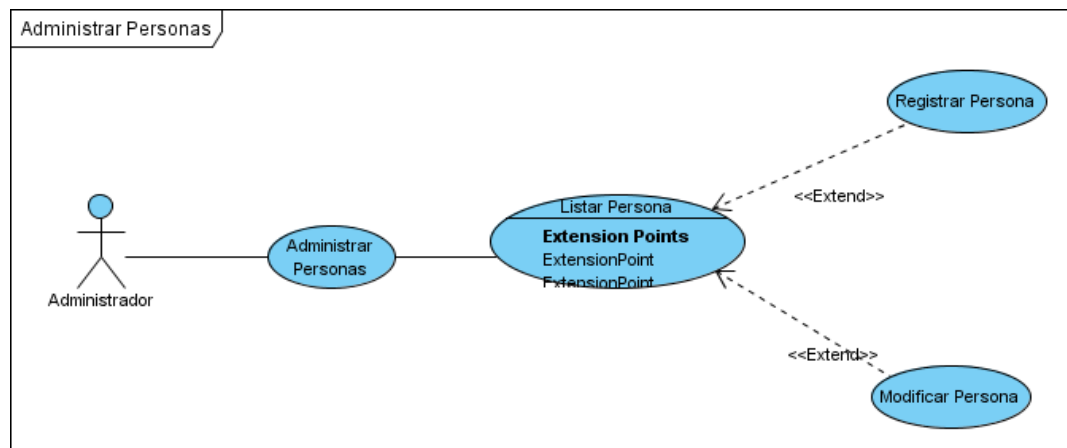


Fig. II.10 caso de uso-Administrar Persona

Casos de Uso: Administrar Empleados

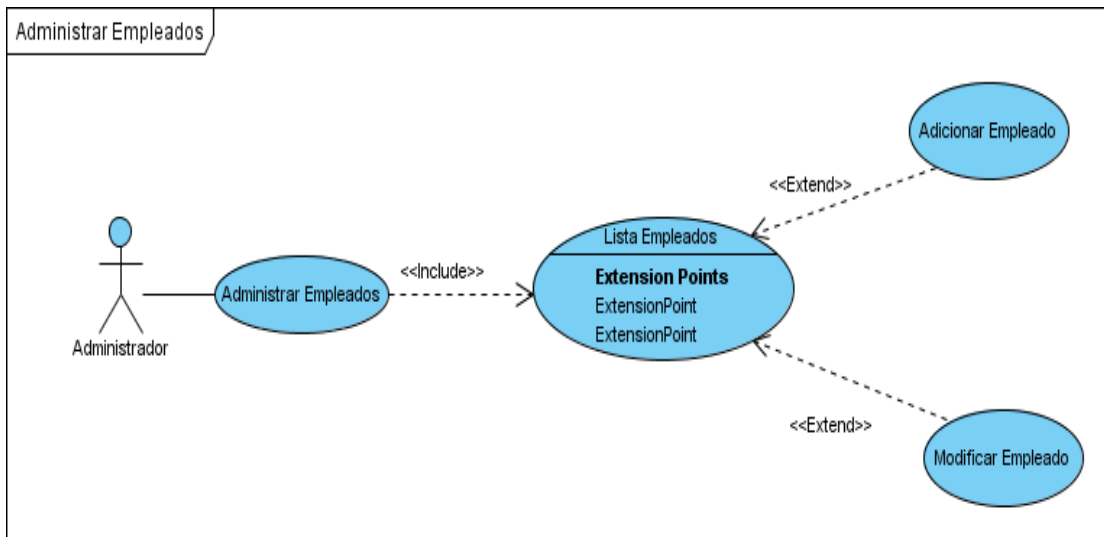


Fig. II.11 caso de uso-Administrar Empleados

Casos de Uso: Administrar Empresas

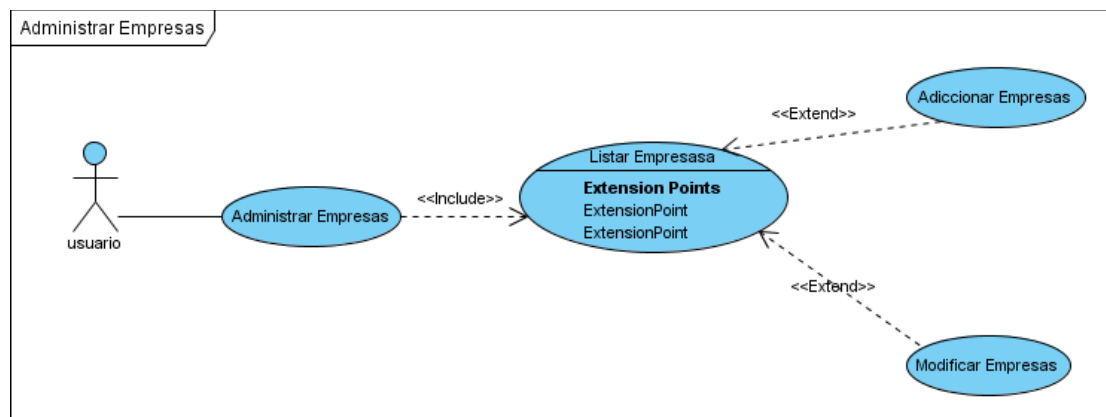


Fig.II.12 caso de uso-Administrar Empresas

Casos de Uso: Administrar Materiales

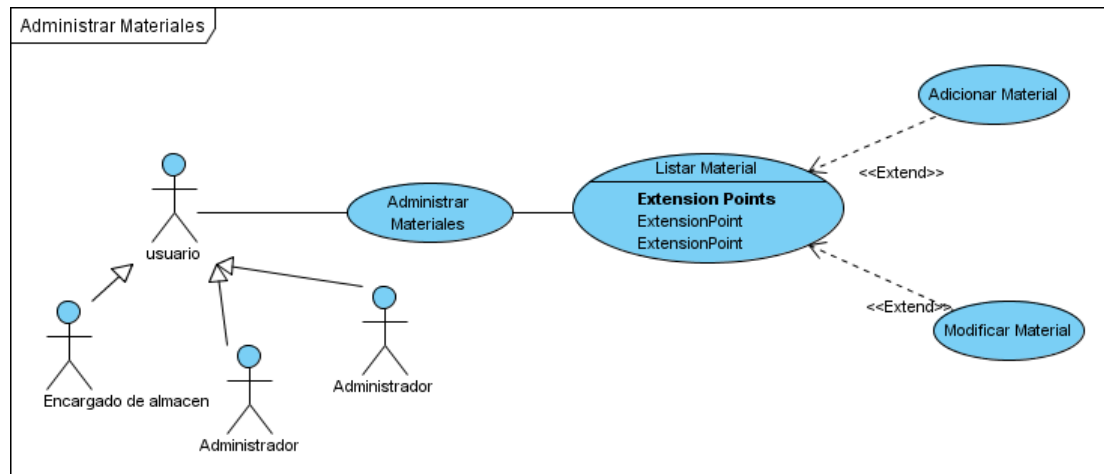


Fig.II.13 caso de uso-Administrar Materiales

Casos de Uso: Administrar Proveedor

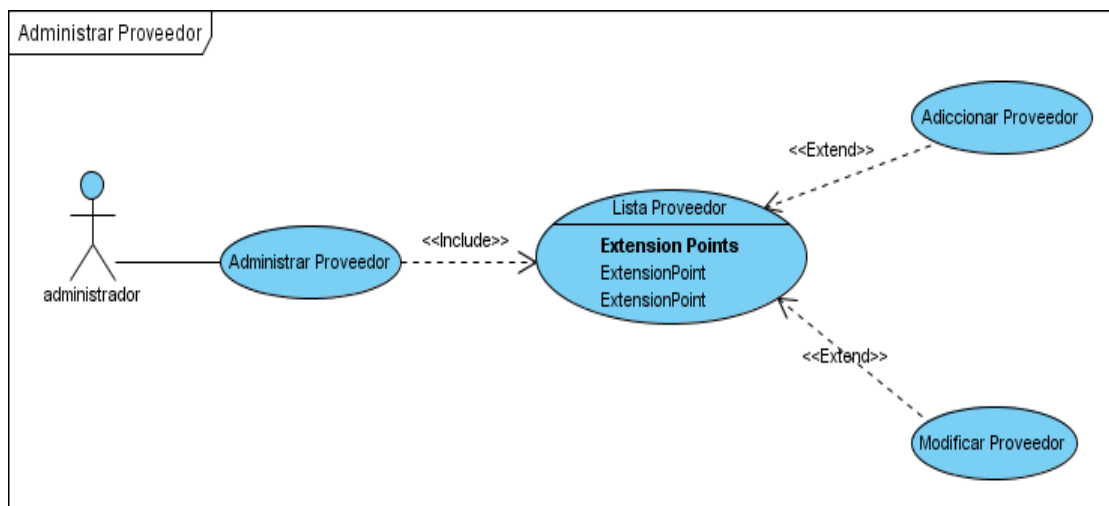


Fig.CII.14 caso de uso-Administrar Proveedor

Casos de Uso: Administrar Categorías

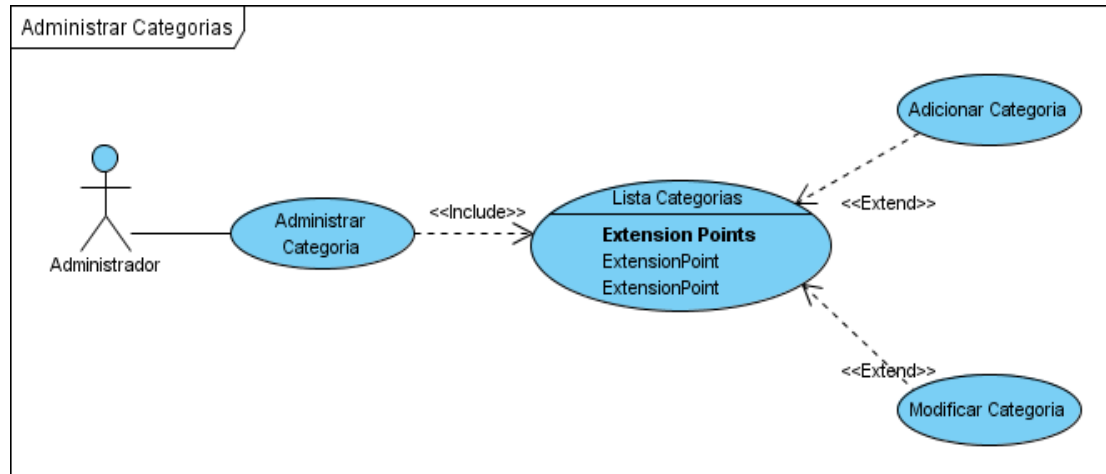


Fig.II.15 caso de uso-Administrar Categorías

Casos de Uso: Administrar Solicitud

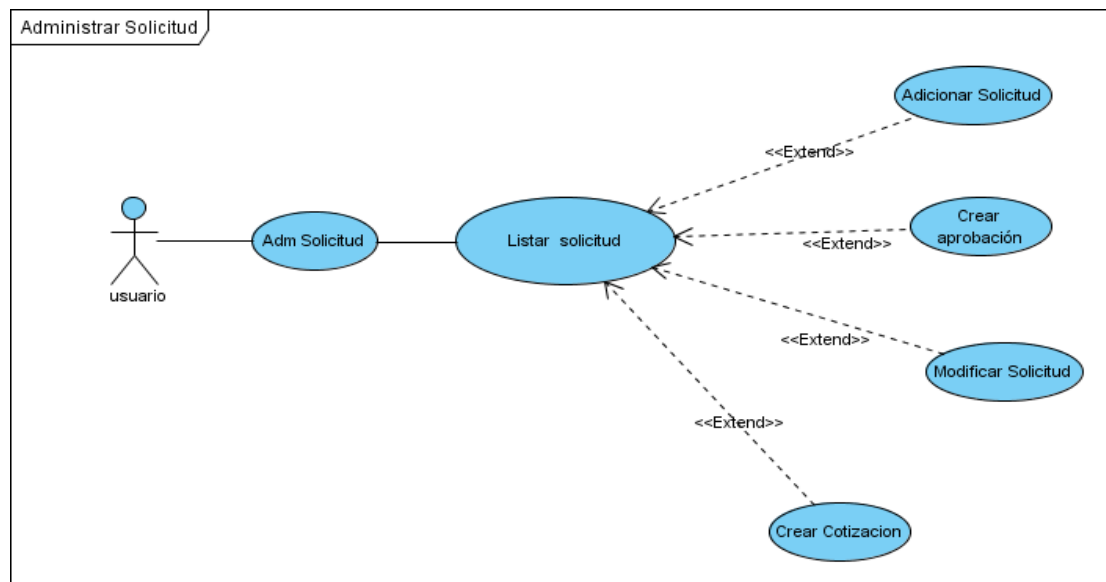


Fig.II.16 caso de uso-Administrar Solicitud

Casos de Uso: Administrar Cotización

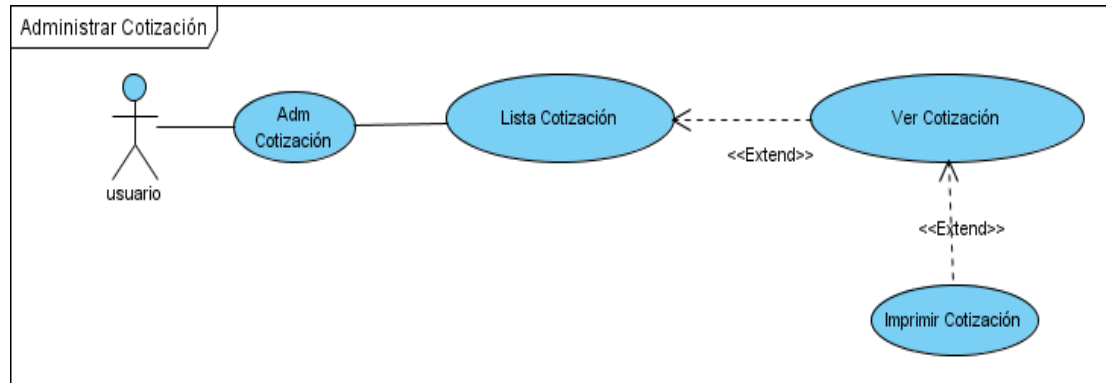


Fig.II.17 casos de uso-Administrar Cotización

Casos de Uso: Administrar Aprobación

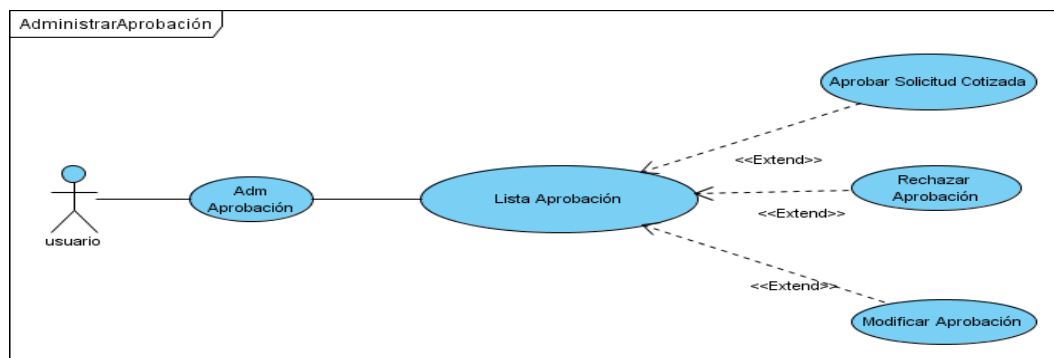


Fig.II.18 caso de uso-Administrar Aprobación

Casos de Uso: Administrar Compra

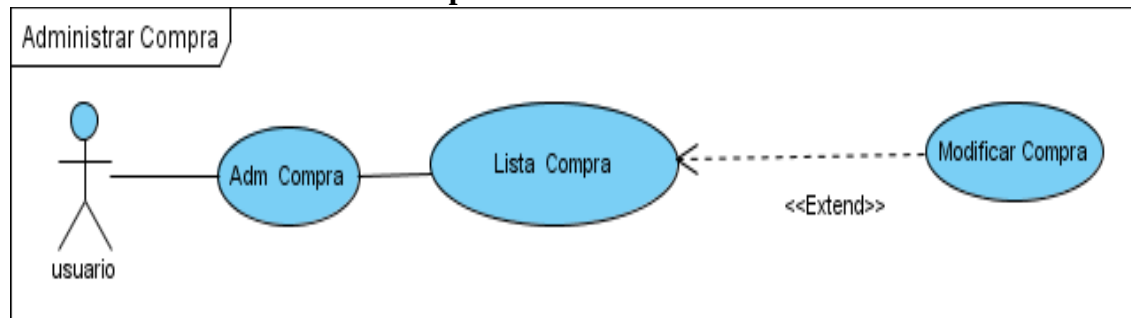


Fig.II.19 caso de uso-Administrar Compra

Casos de Uso: Administrar Almacén

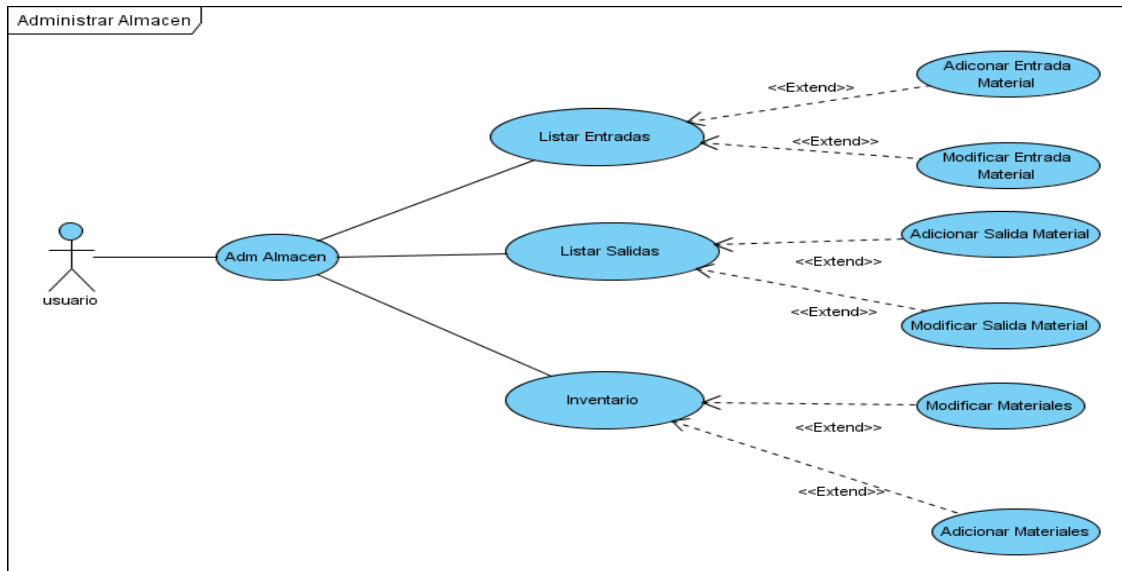


Fig.II.20 caso de uso-Administrar Almacén

Casos de Uso: Administrar Almacenes

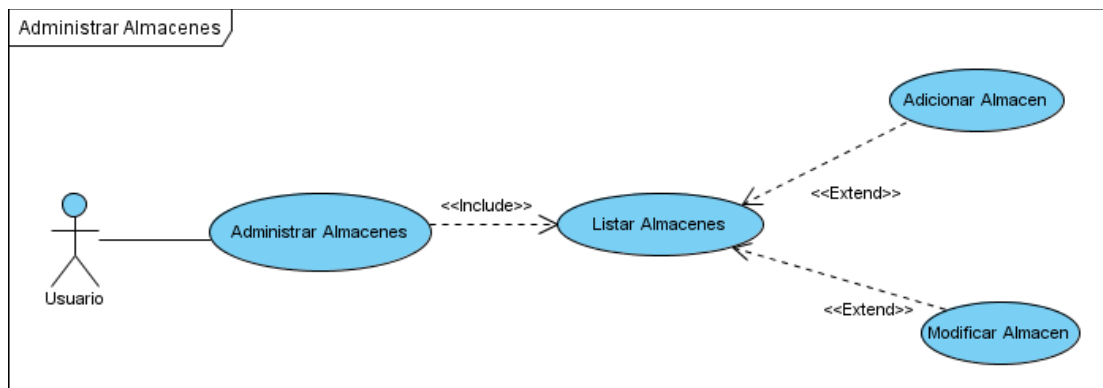


Fig.21 casos de uso-administrar almacenes

Casos de Uso: Administrar Cheque

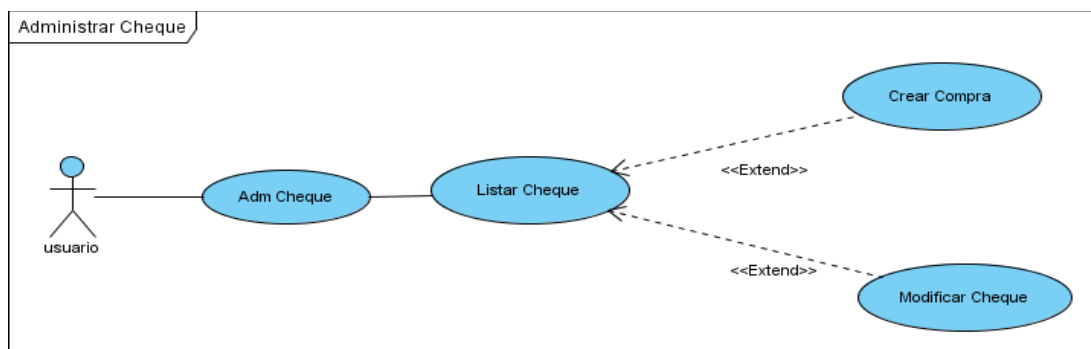


Fig.22 caso de uso-administrarcheque

Casos de Uso: Reportes

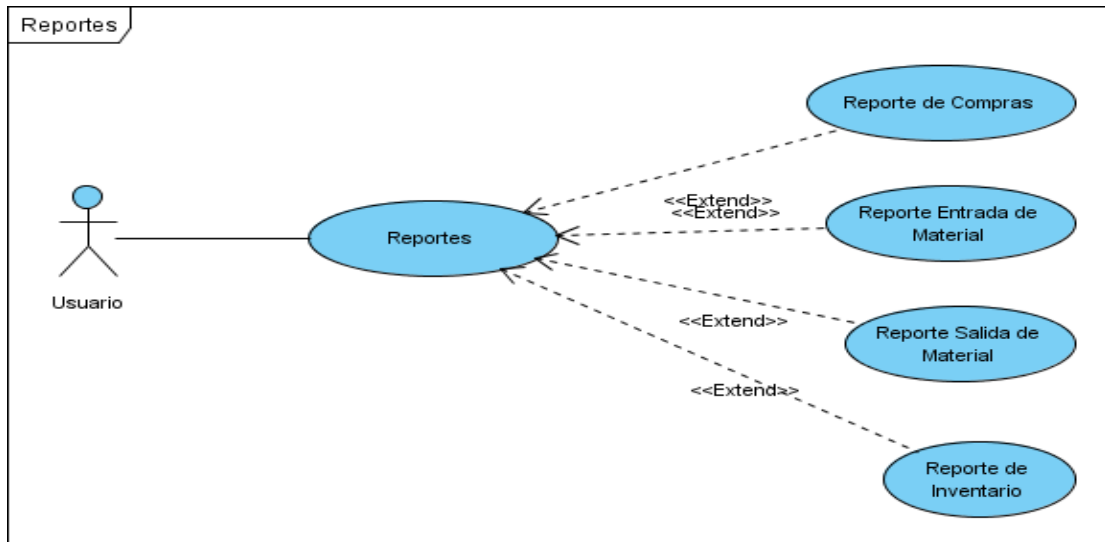


Fig.23 casos de uso- reportes

II.2.4. VISIÓN

II.2.4.1. INTRODUCCIÓN

Propósito

El propósito es definir la visión del producto desde la perspectiva del usuario, especificando las necesidades y características del sistema. Constituye una base de acuerdo a los requisitos del sistema.

Alcance

El documento visión se ocupa, del Sistema del área de archivos de la oficialía mayor administrativa de la alcaldía de San Lorenzo. Dicho sistema será desarrollado por la universitaria María de los Ángeles Castillo. El sistema pretende mejorar los registros de las compras y los registros de entradas y salidas de material que se realizan en la oficialía mayor administrativa de la alcaldía de San Lorenzo.

Limitación

Entre las limitantes del producto, señalamos que el sistema no estará en red ya que el área involucrada solicito que el producto tenga funcionamiento solo en el área de administración.

II.4.2. POSICIONAMIENTO

Oportunidad del Negocio

Con este sistema se mejorará el control del personal y procesara la información de manera rápida y confiable.

Sentencias que define el Proyecto

El problema de	No existe un adecuado registro de las compras y de las entradas y salidas de material.
Afecta a	Administrador(a) del sistema MGAOMA Encargado de solicitudes Encargado de Almacén
El impacto asociado es	Mejorar el control de las compras y entradas y salidas de material del sistema MGAOMA. Mejorar los mecanismos para que los reportes sean obtenidos rápidamente y con información confiable. Tener toda la información oportuna, adecuada y disponible. Es un proceso que sin automatizarlo es bastante tardío y molesto.
Una solución adecuada sería	Automatizar las compras y la entrada y salida de material, usando un sistema de información con una base de datos accesible y eficaz, con un entorno amigable y sencillo.

Tabla.II.2 Sentencias que definen el proyecto

Sentencia que define la Posición del Proyecto

Para	Administrador(a) del Sistema MGAOMA Encargado de Solicitudes Encargado de Almacen
Quienes	Interactuaran de manera directa e indirecta con el sistema.
El nombre del producto	Mejorar la Gestión del Área de la Oficialía Mayor Administrativa “MGAOMA”
Que	Mejorar el control de las compras y los registros realizados en el área de la O.M.A. Mejorar los mecanismos para que los reportes sean generados de manera eficiente y rápida. Tener información oportuna, adecuada y disponible. Brindar seguridad a la información.

No como	El procedimiento que se realiza actualmente, es un proceso manual que requiere mucho tiempo.
Nuestro producto	Registra las compras, entradas/salidas del almacén de forma óptima y segura. Mantiene y permite el acceso a toda la información actualizada. Brinda seguridad ante cualquier suceso inesperado. Permite generar reportes al instante. No depende de ningún otro sistema.

Tabla.II.3 Sentencia que define la posición del proyecto

II.2.4.3. DESCRIPCIÓN DE STAKEHOLDERS(PARTICIPANTES EN (PARTICIPANTES EN EL PROYECTO) Y USUARIOS.

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes del proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sesión muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que estos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por qué estos requisitos son necesarios.

Resumen de Stakeholders

Nombre	Descripción	Responsabilidades
Maria de los Angeles Castillo	Jefe de proyecto Analista en Sistemas Programador Ingeniero de Software	El Stakeholders realiza: La supervisión el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto. La interacción con el cliente y los usuarios mediante entrevistas. Construcción de prototipos. Elaborar la base de datos y gestionar cambios
	Empleado de la Oficialía Mayor Administrativa de la alcaldía de san Lorenzo.	El Stakeholders realiza: Representa a todos los usuarios posibles del sistema. Seguimiento del desarrollo del proyecto.

		Aprueba los requisitos y funcionalidades.
Lic. Deysi Arancibia	Docente de Taller III	Seguimiento y control del desarrollo del proyecto.

Tabla II.4 Resumen de Stakeholders

Perfil de los Stakeholders

Representante del área técnica y sistema de información

Representante	Maria de los Angeles Castillo
Descripción	Universitaria de decimo semestre de la carrera de ingeniería informática.
Tipo	Jefe del Proyecto, Ingeniera de Software, Analista de Sistemas del Proyecto y Programador del sistema.
Responsabilidades	Encargada de mostrar las necesidades de cada usuario del sistema, además lleva a cabo un seguimiento del desarrollo del proyecto y aprobación de los requisitos y funcionalidades del sistema. Elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación, elaboración de modelos de implementación, encargado de elaborar el modelo de análisis y diseño. Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.
Criterio de Éxito	El proyecto tendrá éxito.
Grado de participantes	Revisión del proyecto, estructura del sistema Revisión de requerimientos, diseño del sistema Diseño de la base de datos del sistema. Construcción del programa de forma lógica y física del sistema.

Tabla II.5 Representantes del area técnica e información

II.2.4.4.DESCRIPCIÓN GLOBAL DEL PRODUCTO

Perspectiva del producto

El producto solo será aplicable para administrar la información que maneje la Oficialía Mayor Administrativa de la alcaldía de san Lorenzo y el mismo permitirá que el trabajo de usuario sea más liviano, rápido y seguro, dando de esta manera la seguridad necesaria al momento de manejar la información.

A continuación se mostrara en una tabla los beneficios que obtendrá el usuario a partir del producto.

Beneficio del Usuario	Características que lo apoyan
Seguridad de la información	Se ingresara al sistema con un usuario y contraseña que solo el usuario conozca y además se cuenta una base de datos encriptado.
Encontrara la información de una forma más rápida	El sistema contara con un buscador para lo cual solo hará falta meter una característica de la información a buscar.
Ordenamiento de la información	El sistema contara con una base de datos donde la información será ordenada de acuerdo a la fecha que se la registró.
Reportes	El usuario podrá ver los reportes que genera el sistema de manera confiable y segura

Tabla.II.6 Perspectiva del producto

Supuestos y Dependencias

Las suposiciones y restricciones están mencionadas en la Matriz de Marco Lógico del proyecto.

II.2.4.5. REQUISITOS DE DOCUMENTACION

Manual de Usuario

Si el usuario contara con este documento para aprender a manejar el sistema, el manejo será más rápido.

Ayuda en línea

El usuario tendrá ayuda a cualquier duda mediante correo electrónico, y vía telefónica

Guías de Instalación y Configuración

Todo esto será entregado al usuario conjuntamente con el sistema que será de gran ayuda al momento de tener dudas en el manejo del sistema.

II.2.5. ESPECIFICACIÓN DE CASOS DE USO

Introducción

La Especificación de Casos de Uso es una descripción detallada de los casos de uso del sistema.

Propósito

- Comprender los casos de Uso del Sistema.
- Describir específicamente cada caso de uso
- Entender la relación que tendrá el sistema con el caso de uso y el usuario.

Alcance

- Describir los procesos internos de los casos de uso
- Describir los flujos de cada **caso de uso** según lo establecido por la institución involucrada.

II.2.5.1. ESPECIFICACIONES DE CASOS DE USO

Especificación del Caso de Uso: Ingresar al Sistema

Caso de uso	Ingresar al sistema
Explicación: Este caso de uso nos muestra el ingreso al sistema mediante un usuario y contraseña, cuando el usuario entra al sistema puede administrar distintos módulos de acuerdo al rol que cumple en la institución.	
Curso Normal	Alternativas
El usuario ingresa su nombre de usuario	
El usuario ingresa su contraseña de acceso	
El sistema valida sus datos introducidos.	3.1 Si los datos no son válidos el sistema muestra un mensaje “usuario o clave incorrecto” y se retorna a la pantalla de ingreso

Tabla II.7 Especificación del caso de uso- Ingresar al sistema

Especificación del Caso de Uso: Administrar Personas

Caso de uso	Administrar Personas
Explicación: Este caso de uso permite administrar a las personas que intervienen en el sistema.	
Curso Normal	Alternativas
1. Listar Personas: se muestra un formulario con la lista de todas	

las personas del sistema	
2. Modificar Persona: se muestra un formulario con los datos de la persona a modificar y el sistema actualiza los datos en la tabla.	2.1 Si no se selecciona un empleado de la lista de empleados el sistema no permite modificar los datos.

TablaII.8 Especificacion del caso uso-Administrar Personas

Especificación del Caso de Uso: Administrar usuarios

Caso de uso	Administrar Usuarios
Explicación: Este caso de uso permite administrar los usuarios que intervienen en el sistema	
Curso Normal	Alternativas
1.Listar usuarios: se muestra un formulario con la lista de todos los usuarios del sistema	
2. Adicionar Usuarios: El sistema muestra el formulario adicionar usuarios y se validan los datos si el formulario es llenado correctamente..	
3. Modificar usuarios: se muestra un formulario con los datos del usuario a modificar.	3.1 Si no se selecciona un usuario de la lista de usuarios el sistema no permite modificar los datos.
4.Usuarios Activos: El sistema	

muestra lo usuarios que están activos	
5.Usuarios Inactivos: El sistema muestra los usuarios que están inactivos	

TablaII.9 Especificacion del caso uso-Administrar Usuarios

Especificación del Caso de Uso: Administrar Empleados

Caso de uso	Administrar Empleados
Explicación: Este caso de uso permite administrar los empleados que intervienen en el sistema	
Curso Normal	Alternativas
1.Listar Empleados: se muestra un formulario con la lista de todos los empleados del sistema	
2. Adicionar Empleados: El sistema muestra el formulario adicionar empleados , se validan los datos si el formulario es llenado correctamente..	
3. Modificar Empleados: se muestra un formulario con los datos del empleado a modificar.	3.1 Si no se selecciona un empleado de la lista de empleados el sistema no permite modificar los datos.

TablaII.10 Especificacion del caso uso-Administrar Empleados

Especificación del Caso de Uso: Administrar Proveedor

Caso de uso	Administrar Proveedor
Explicación: Este caso de uso permite administrar los proveedores de las empresas que intervienen en el sistema	
Curso Normal	Alternativas
1. Listar Proveedor: se muestra un formulario con la lista de todos los proveedores del sistema	
2. Adicionar Proveedor: El sistema muestra el formulario adicionar proveedor, y se validan los datos si el formulario es llenado correctamente. .	
3. Modificar Proveedor: se muestra un formulario con los datos del proveedor a modificar y el sistema actualiza los datos si fueron llenados correctamente.	3.1 Si no se selecciona un proveedor de la lista de proveedores el sistema no permite modificar los datos.

Tabla II.11 Especificación del caso uso-Administrar Proveedor

Especificación del Caso de Uso: Administrar Empresas

Caso de uso	Administrar Empresas
Explicación: Este caso de uso permite administrar a las empresas que intervienen en el sistema	
Curso Normal	Alternativas
1.Lista Empresas: se muestra un formulario con la lista de todas las empresas adicionadas en el sistema	
2. Adicionar Empresa: El sistema muestra el formulario adicionar empresa, y se validan los datos si el formulario es llenado correctamente. .	
3. Modificar Empresa: se muestra un formulario con los datos de las empresas a modificar y el sistema actualiza los datos si fueron llenados correctamente.	3.1 Si no se selecciona una empresa de la lista, el sistema no permite modificar los datos.

TablaII.12 Especificacion del caso uso-Administrar Empresa

Especificación del Caso de Uso: Administrar Materiales

Caso de uso	Administrar Materiales
Explicación: Este caso de uso permite administrar los materiales que intervienen en el sistema	

Curso Normal	Alternativas
1.Lista Materiales: se muestra un formulario con la lista de todas los materiales adicionadas en el sistema.	
2. Adiciona material: El sistema muestra el formulario adicionar material, y se validan los datos si el formulario es llenado correctamente. .	
3. Modificar Material: se muestra un formulario con los datos de materiales a modificar y el sistema actualiza los datos si fueron llenados correctamente.	3.1 Si no se selecciona un material de la lista, el sistema no permite modificar los datos.

TablaII.13 Especificacion del caso uso-Administrar Materiales

Especificación del Caso de Uso: Administrar Categorías

Caso de uso	Administrar Materiales
Explicación: Este caso de uso permite administrar las categorías de los materiales que intervienen en el sistema.	
Curso Normal	Alternativas
1. Lista Categorías: se muestra un formulario con la lista de todas las categorías de materiales adicionadas en el sistema.	

2. Adicionar Categorías: El sistema muestra el formulario adicionar categoría, y se validan los datos si el formulario es llenado correctamente. .	
3. Modificar categoría: se muestra un formulario con los datos de materiales a modificar y el sistema actualiza los datos si fueron llenados correctamente.	3.1 Si no se selecciona una categoría de la lista, el sistema no permite modificar los datos.

TablaII.14 Especificacion del caso uso-Administrar Categorías

Especificación del Caso de Uso: Administrar Solicitud

Caso de uso	Solicitud
Explicación: Este caso de uso permite administrar las solicitudes en el sistema.	
Curso Normal	Alternativas
1. Listar Solicitud: se muestra un formulario con la lista de todas las solicitudes de materiales adicionadas en el sistema.	
2. Adicionar Solicitudes: El sistema muestra el formulario adicionar solicitud, y se validan los datos si el formulario es llenado correctamente. .	

3. Modificar solicitud: se muestra un formulario con los datos de las solicitudes a modificar y el sistema actualiza los datos si fueron llenados correctamente.	3.1 Si no se selecciona una solicitud de la lista, el sistema no permite modificar los datos.
--	---

TablaII.15 Especificacion del caso uso-Administrar Solicitud

Especificación del Caso de Uso: Administrar Cotización

Caso de uso	Cotización
Explicación: Este caso de uso permite administrar las Cotizaciones en el sistema	
Curso Normal	Alternativas
1. Listar Cotización: se muestra un formulario con la lista de todas las cotizaciones de solicitudes adicionadas en el sistema.	
2. Ver Cotizaciones: El sistema muestra el formulario ver cotización donde el sistema sugiere que empresa es la más optima para la compra de los materiales.	

TablaII.16 Especificación del caso uso-Administrar Cotización

Especificación del Caso de Uso: Administrar Aprobación

Caso de uso	Adm Aprobación
Explicación: Este caso de uso permite administrar las Aprobaciones de las compras de material en el sistema.	
Curso Normal	Alternativas
1. Lista Aprobación: se muestra un formulario con la lista de todas las solicitudes de materiales aprobadas en el sistema.	1.1 Por Aprobar: el sistema muestra en una lista las solicitudes por aprobar, se puede aprobar o rechazar. 1.2 Rechazadas: el sistema muestra en una lista las solicitudes rechazadas.
2. Modificar Aprobación: El sistema muestra un formulario con los datos de las solicitudes aprobadas a modificar y el sistema actualiza los datos si fueron llenados correctamente.	2.1 Si no se selecciona una solicitud aprobada de la lista, el sistema no permite modificar los datos.

TablaII.17 Especificacion del caso uso-Administrar Aprobación

Especificación del Caso de Uso: Administrar Compras

Caso de uso	Adm Compras
Explicación: Este caso de uso permite administrar las Compras en el sistema.	
Curso Normal	Alternativas
1. Lista Compras: se muestra un formulario con la lista de todas	

las Compras de los materiales adicionadas en el sistema.	
2. Registrar Compra: El sistema muestra el formulario registrar compra, y se validan los datos el formulario sí es llenado correctamente.	
3. Modificar Compra: se muestra un formulario con los datos de las compras a modificar y el sistema actualiza los datos si fueron llenados correctamente.	3.1 Si no se selecciona una compra de la lista, el sistema no permite modificar los datos.

TablaII.18 Especificacion del caso uso-Administrar Compra

Especificación del Caso de Uso: Administrar Almacén

Caso de uso	Adm Almacén
Explicación: Este caso de uso permite las entradas, salidas de material y el inventario de material en el sistema.	
Curso Normal	Alternativas
1. Listar Entradas: se muestra un formulario con la lista de todas las Entradas de los materiales adicionados en el sistema.	
2. Adicionar entrada material: El sistema muestra el formulario adicionar materiales almacenes, y se validan los	

datos el formulario sí es llenado correctamente.	
3. Modificar entrada material: se muestra un formulario con los datos de las entradas de material a modificar y el sistema actualiza los datos si fueron llenados correctamente.	3.1 Si no se selecciona un dato de la lista, el sistema no permite modificar los datos.
4. Listar Salida de Material: se muestra un formulario con la lista de todas las salidas de los materiales adicionados en el sistema.	
Adicionar la Salida de Material: El sistema muestra el formulario adicionar los materiales que salen del almacén, y se validan los datos el formulario sí es llenado correctamente.	
6. Modificar Salida de Material: se muestra un formulario con los datos de las salidas de material a modificar y el sistema actualiza los datos si fueron llenados correctamente.	6.1 Si no se selecciona un dato de la lista, el sistema no permite modificar los datos.
7. Adicionar Inventario	
8.Modificar Inventario	8.1 Si no se selecciona un dato de la lista, el sistema no permite modificar los datos.

TablaII.19 Especificacion del caso uso-Administrar Almacén

Especificación del Caso de Uso: Administrar Almacenes

Caso de uso	Administrar Almacenes
Explicación: Este caso de uso permite administrar los Almacenes en el sistema.	
Curso Normal	Alternativas
1. Listar Almacenes: se muestra un formulario con la lista de todos los almacenes.	
2. Adicionar Almacenes: muestra el formulario adicionar almacen, y se validan los datos el formulario sí es llenado correctamente.	
3. Modificar Almacen: se muestra un formulario con los datos de los datos del almacen a modificar y el sistema actualiza los datos si fueron llenados correctamente.	3.1 Si no se selecciona un almacen de la lista, el sistema no permite modificar los datos.

TablaII.20 Especificacion del caso uso-Administrar Almacenes

Especificacion de Casos de Uso: Reportes

Caso de uso	Reportes
Explicación: Este caso de uso permite administrar los Reportes en el sistema.	
Curso Normal	Alternativas
1. Reporte de Compra: se muestra la opción del reporte compra.	
2. Reporte Entrada de Material: muestra el los reportes de los materiales que se compran y entran a almacen.	
3. Reporte Salida de Almacen: Muestra los reportes de los materiales que se entrega a salen de almacen	
4. Reporte Inventario: Muestra los reportes del inventario del sistema	

TablaII.21 Especificacion del caso uso-Roportes

II.2.6. MODELADO DE DIAGRAMA DE ACTIVIDADES

Introducción

Un diagrama de actividades representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un diagrama de Actividades muestra el flujo de control general.

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

Propósito

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

Alcance

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

II.II.6.1. DIAGRAMAS DE ACTIVIDAD

Diagrama de Actividad: Ingresar al Sistema

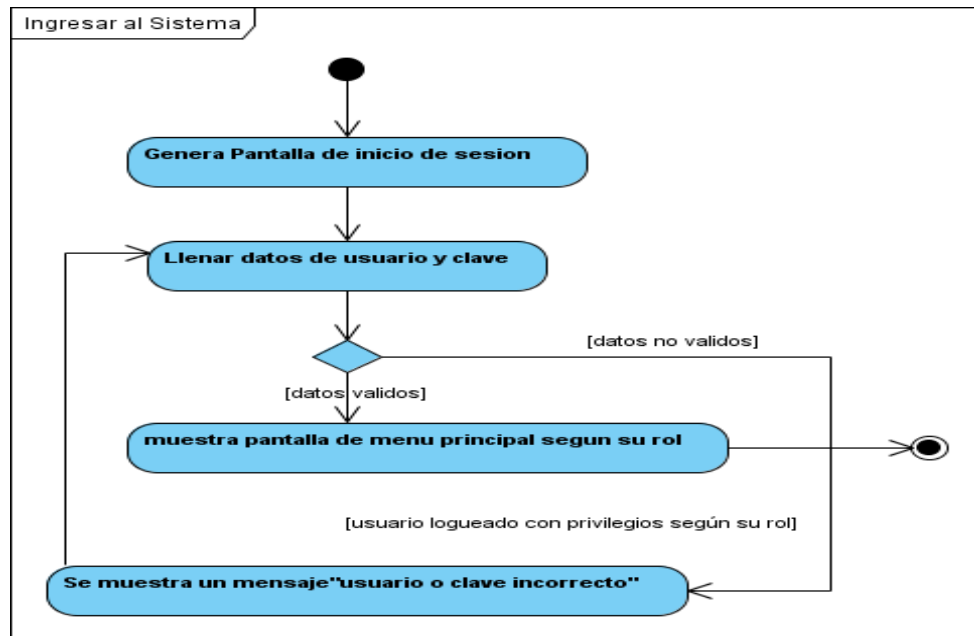


Fig.II.24 diagrama de actividad-Ingresar al Sistema

ADMINISTRAR USUARIOS

Diagrama de Actividad: Listar Usuarios

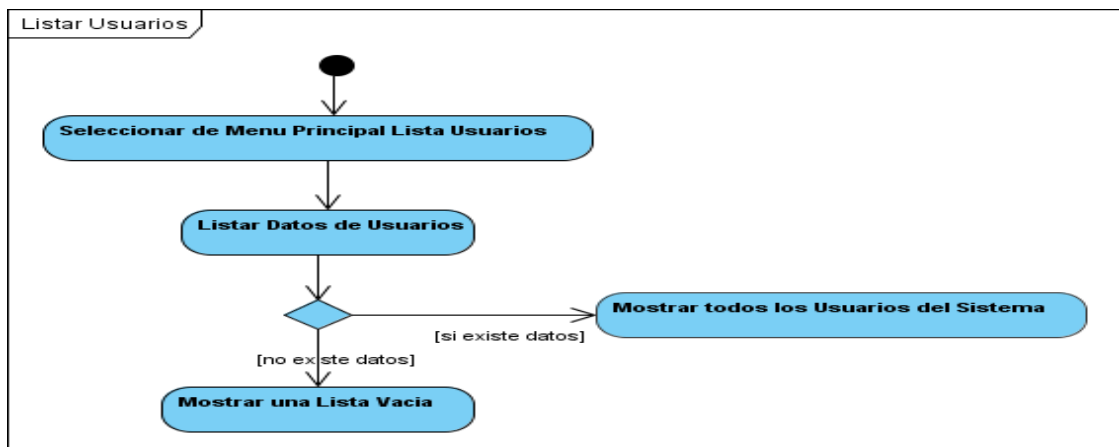


Fig.II.25 diagrama de actividad-Listar Usuario

Diagrama de Actividad: Adicionar Usuarios

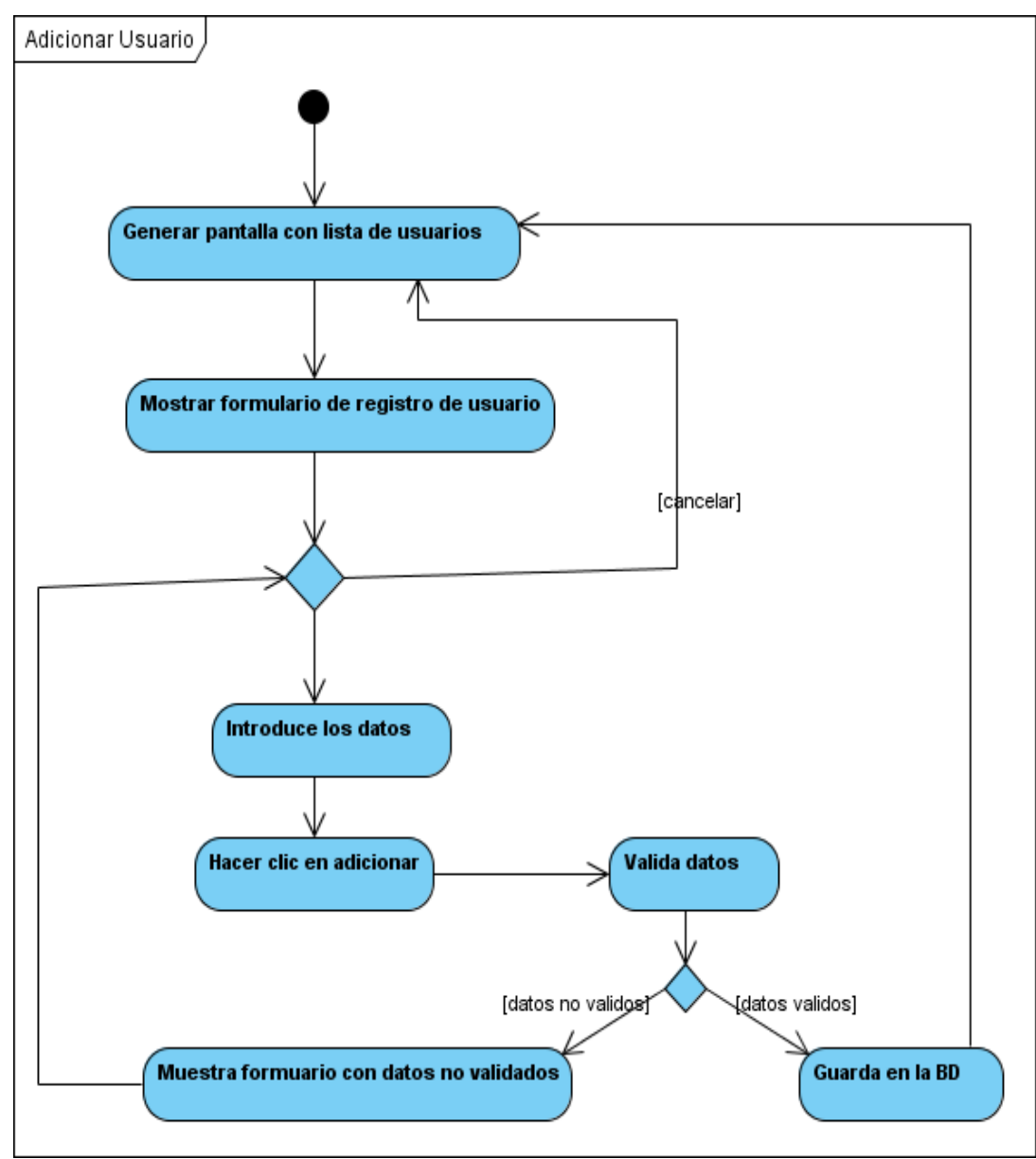


Fig.II.26 diagrama de actividad-Adicionar Usuarios

Diagrama de Actividad: Modificar Usuarios

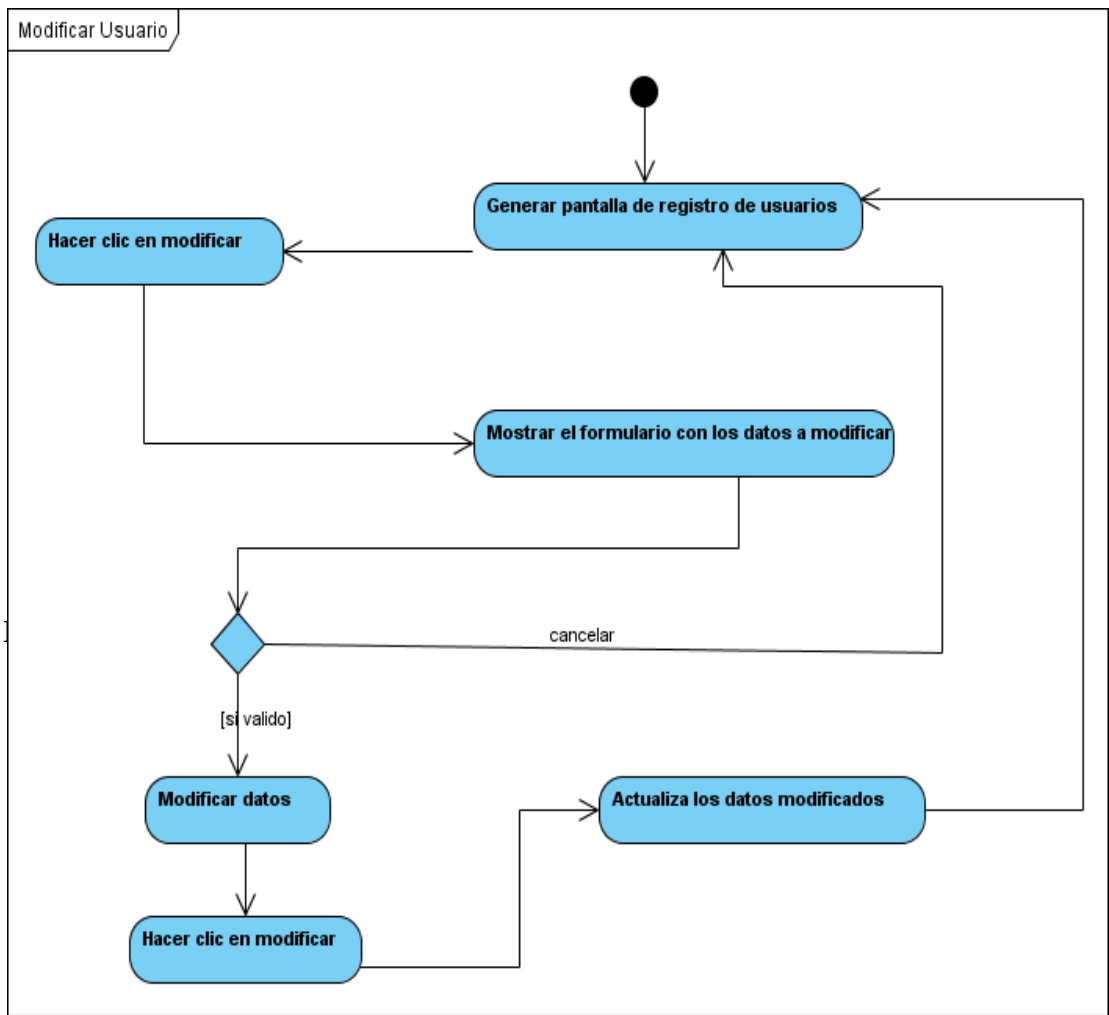


Fig.II.27 diagrama de actividad-Modificar Usuarios

ADMINISTRAR PERSONA

Diagrama de Secuencia: Registrar Persona

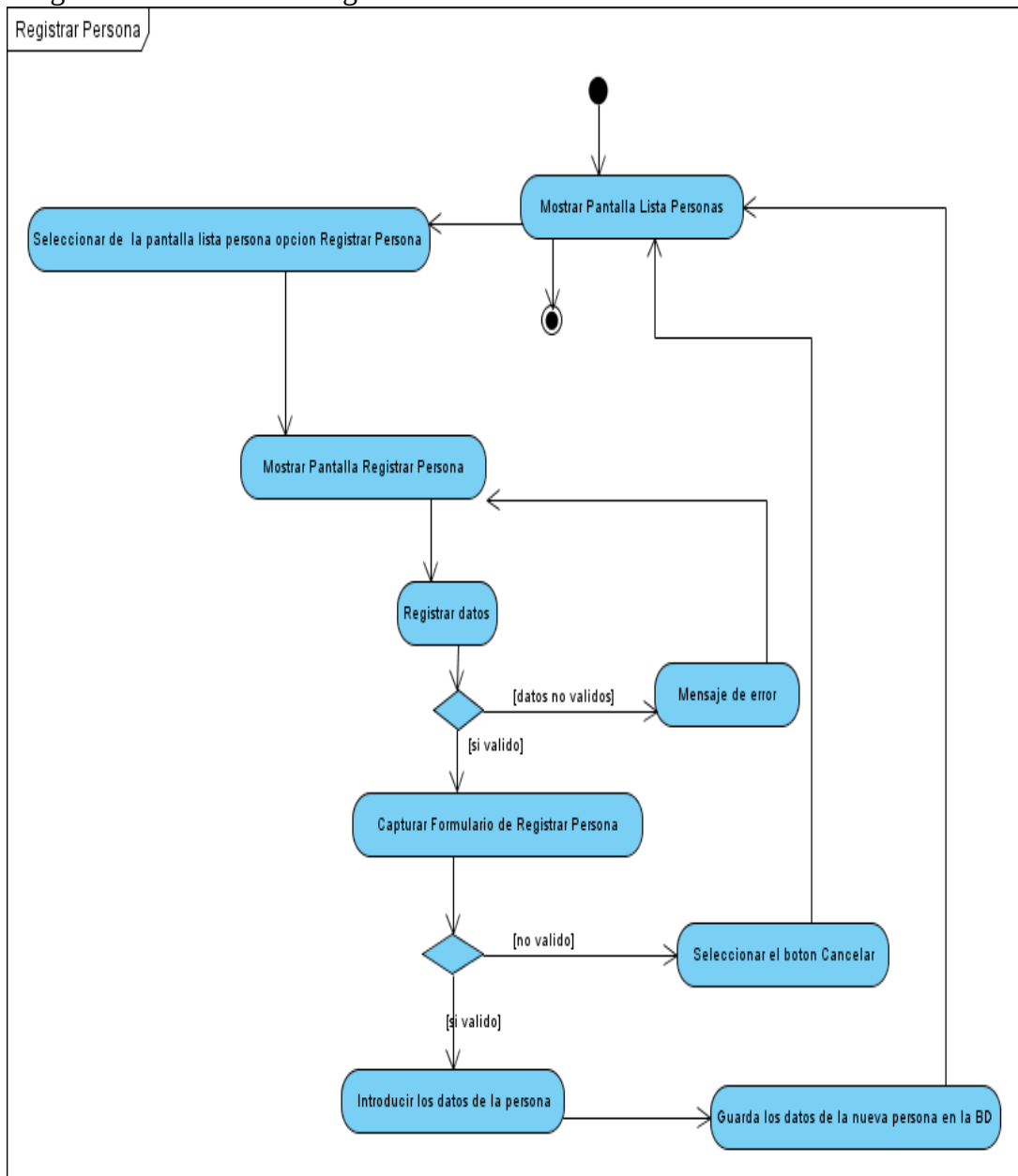


Fig.II.28 diagrama de actividad-Registrar Persona

Diagrama de Actividad: Modificar Persona

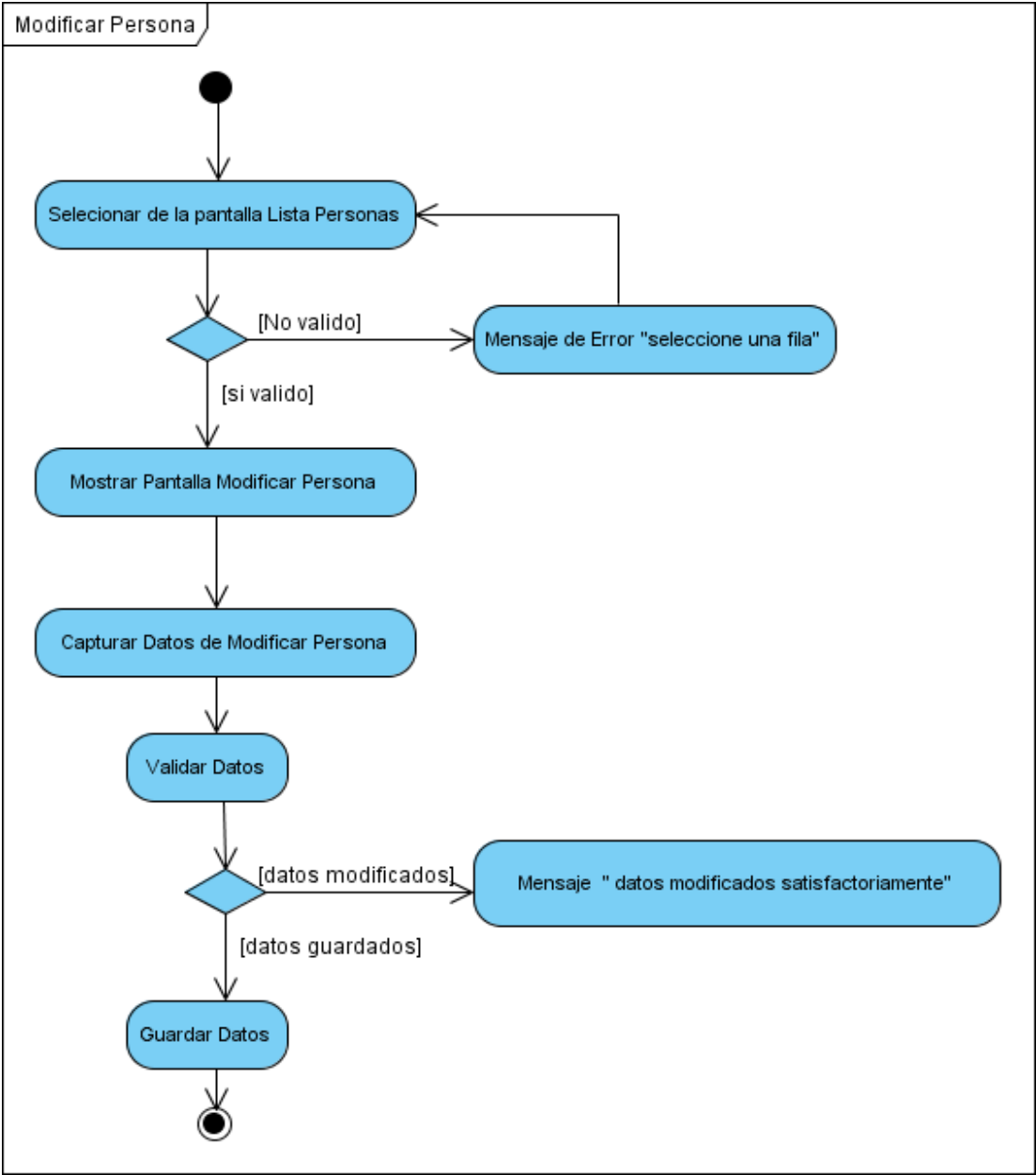


Fig.II.29 diagrama de actividad-Modificar Persona

ADMINISTRAR EMPLEADO

Diagrama de Actividad: Listar Empleado

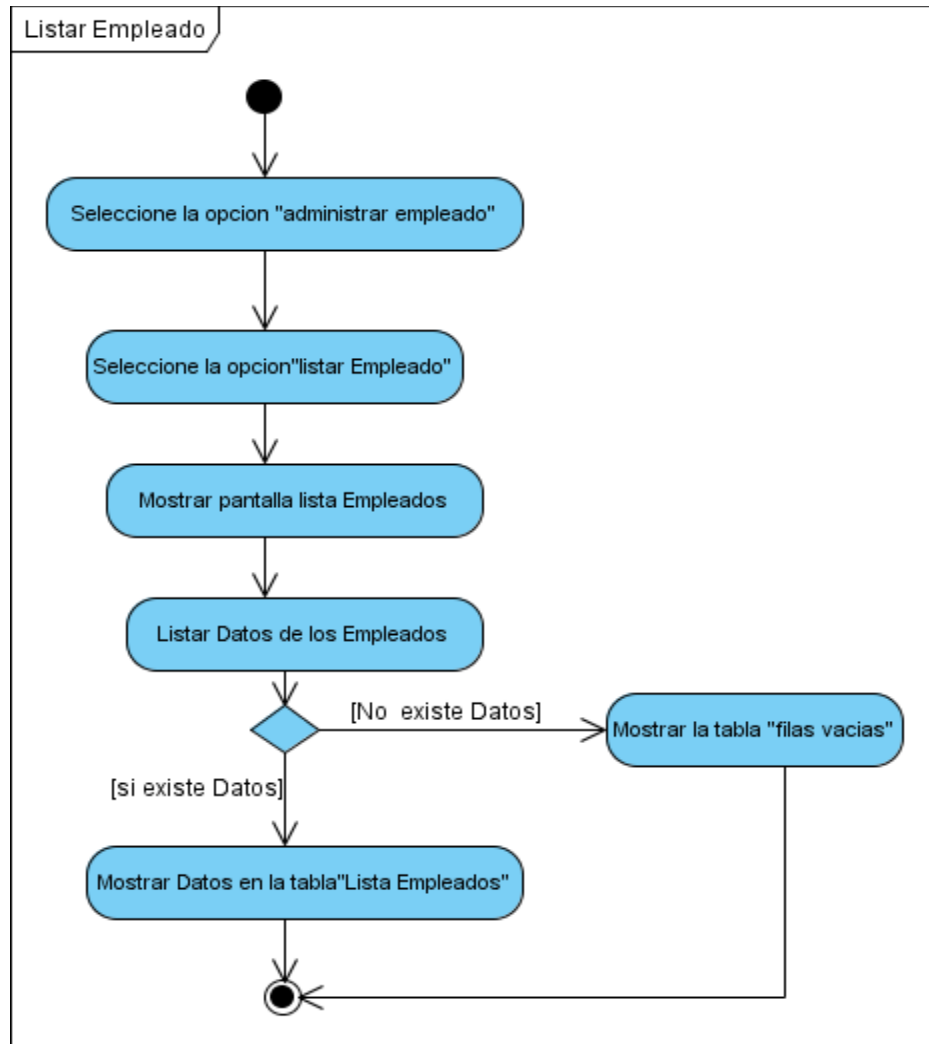


Fig.II.30 diagrama de actividad-Listar Empleado

Diagrama de Actividad: Adicionar Empleado

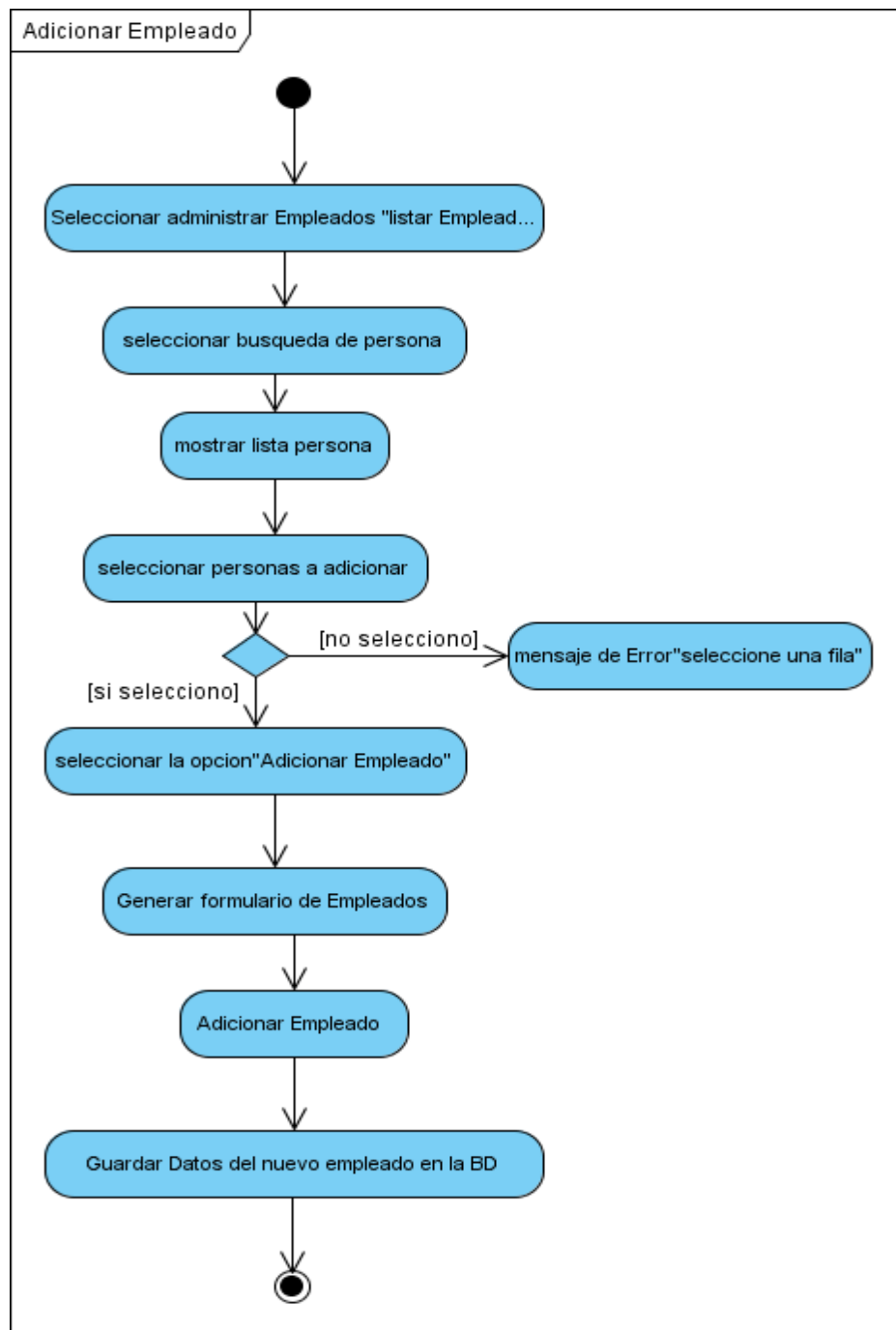


Fig.II.31 diagrama de actividad-Adicionar Empleado

Diagrama de Actividad: Modificar Empleado

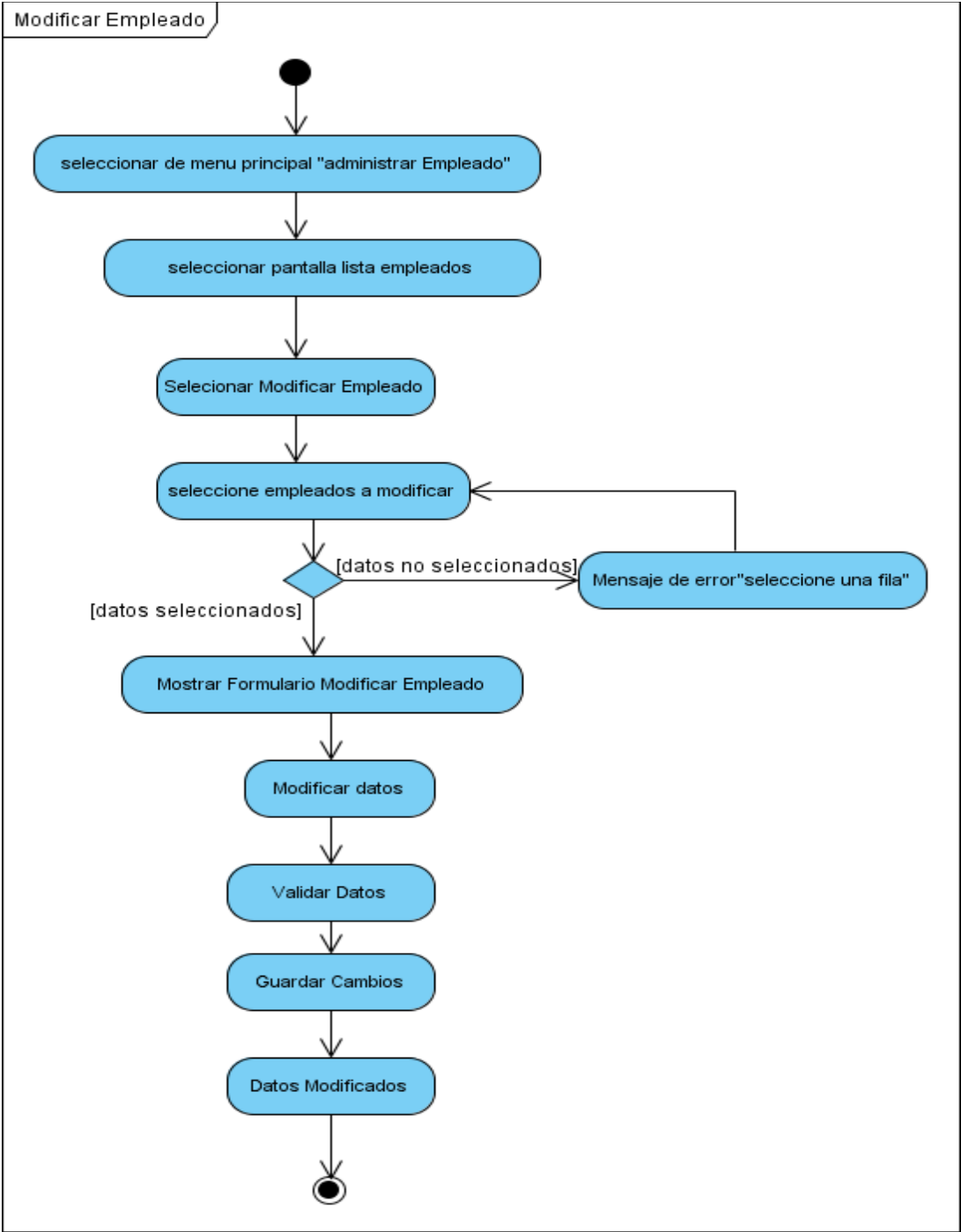


Fig.II.32 diagrama de actividad-Modificar Empleado

ADMINISTRAR SOLICITUD

Diagrama de Actividad: Listar Solicitud

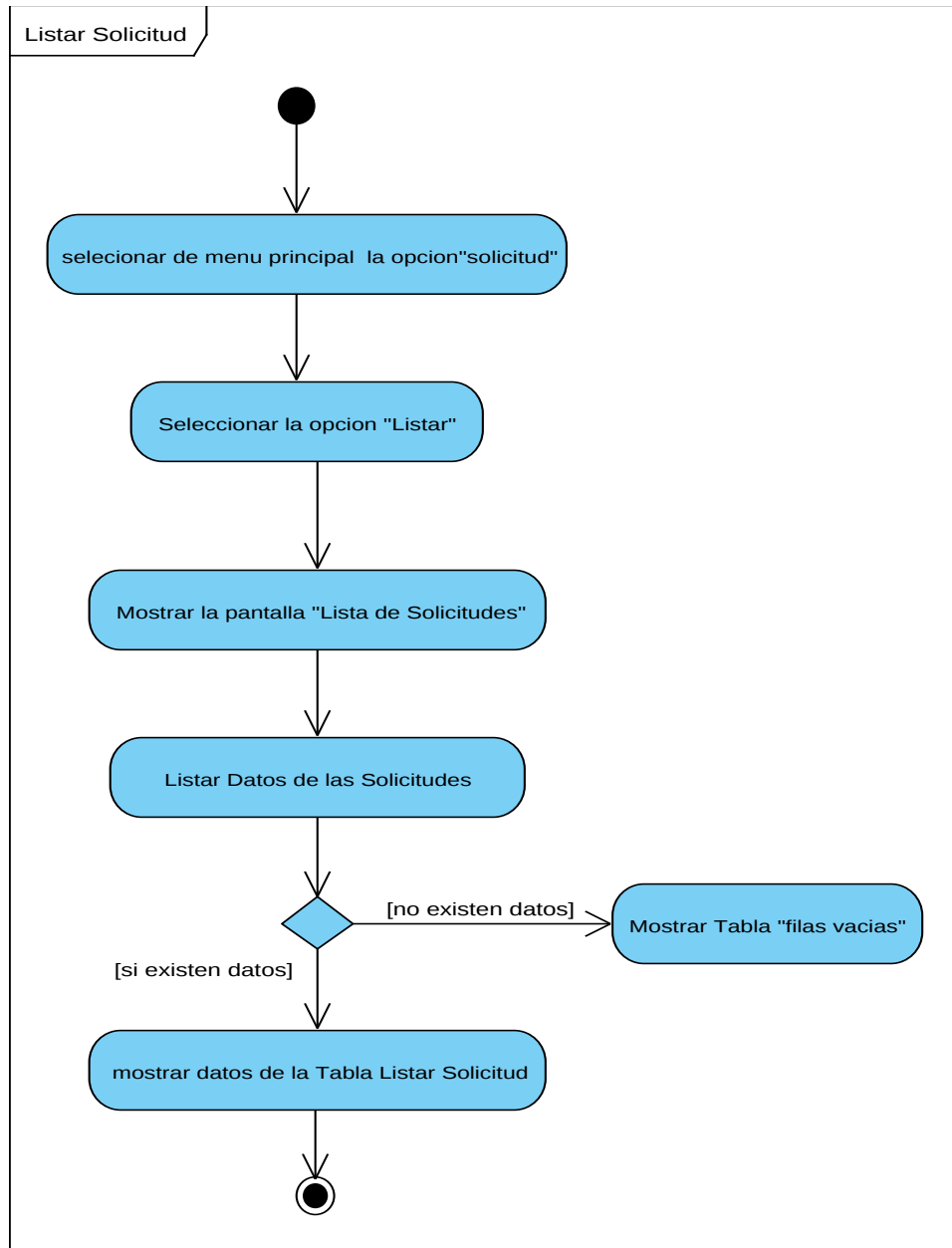


Fig.II.33 diagrama de actividad-Listar solicitud

Diagrama de Actividad: Adicionar Solicitud

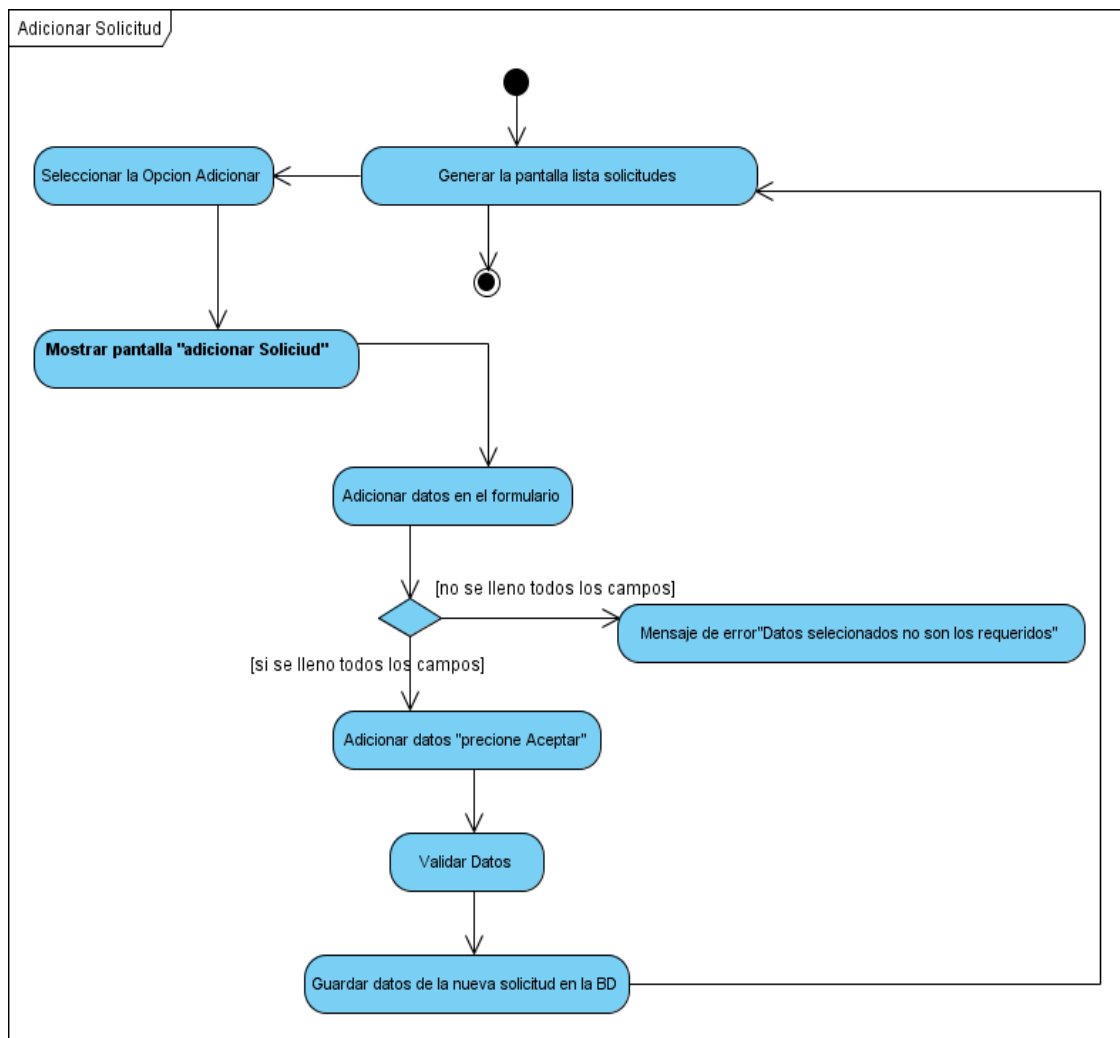


Fig.II.34 diagrama de actividad-Adicionar solicitud

Diagrama de Actividad: Modificar Solicitud

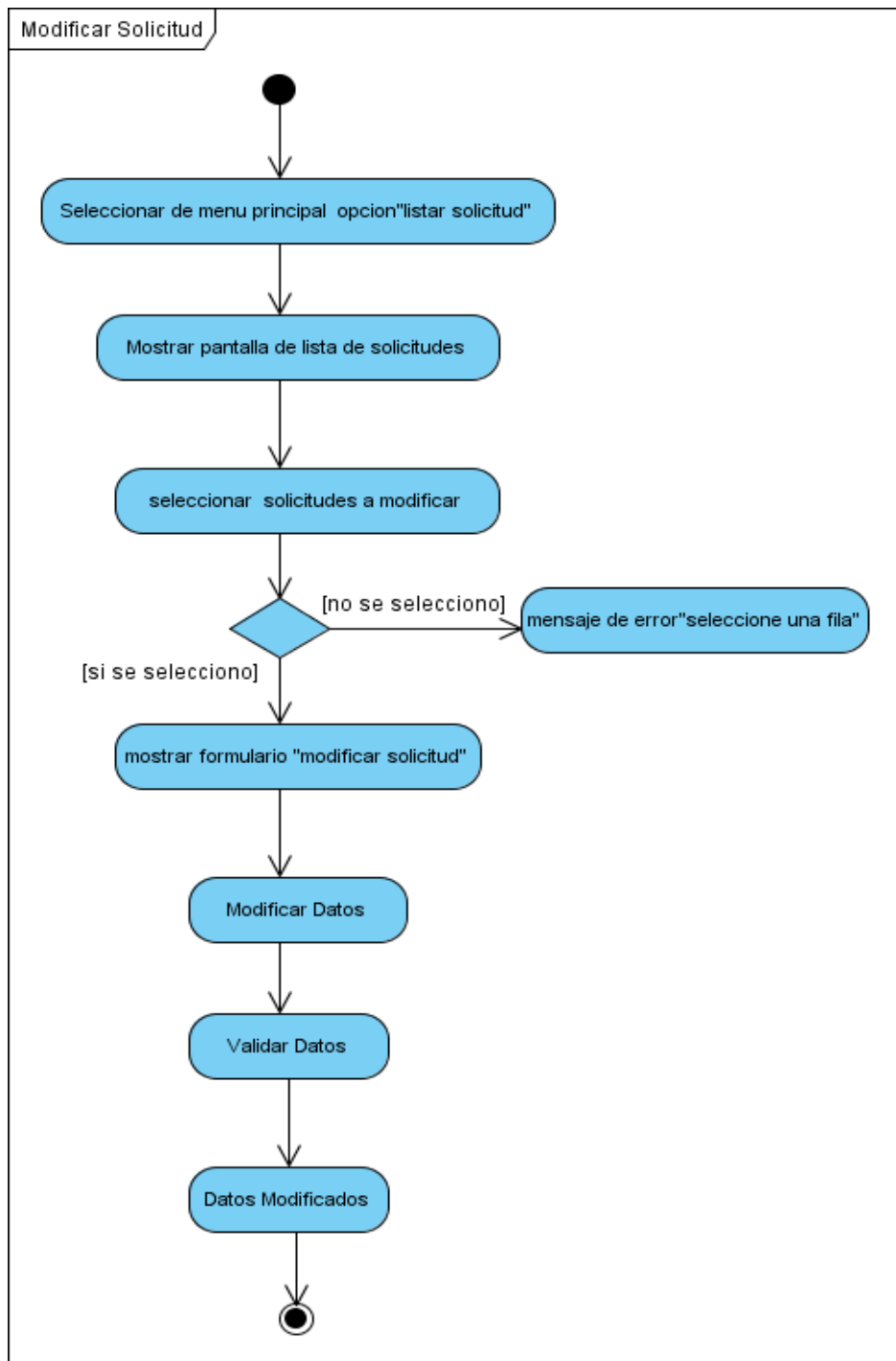


Fig.II.35 diagrama de actividad-Modificar Solicitud

ADMINISTRAR COTIZACIÓN

Diagrama de Actividad: Listar Cotizaciones

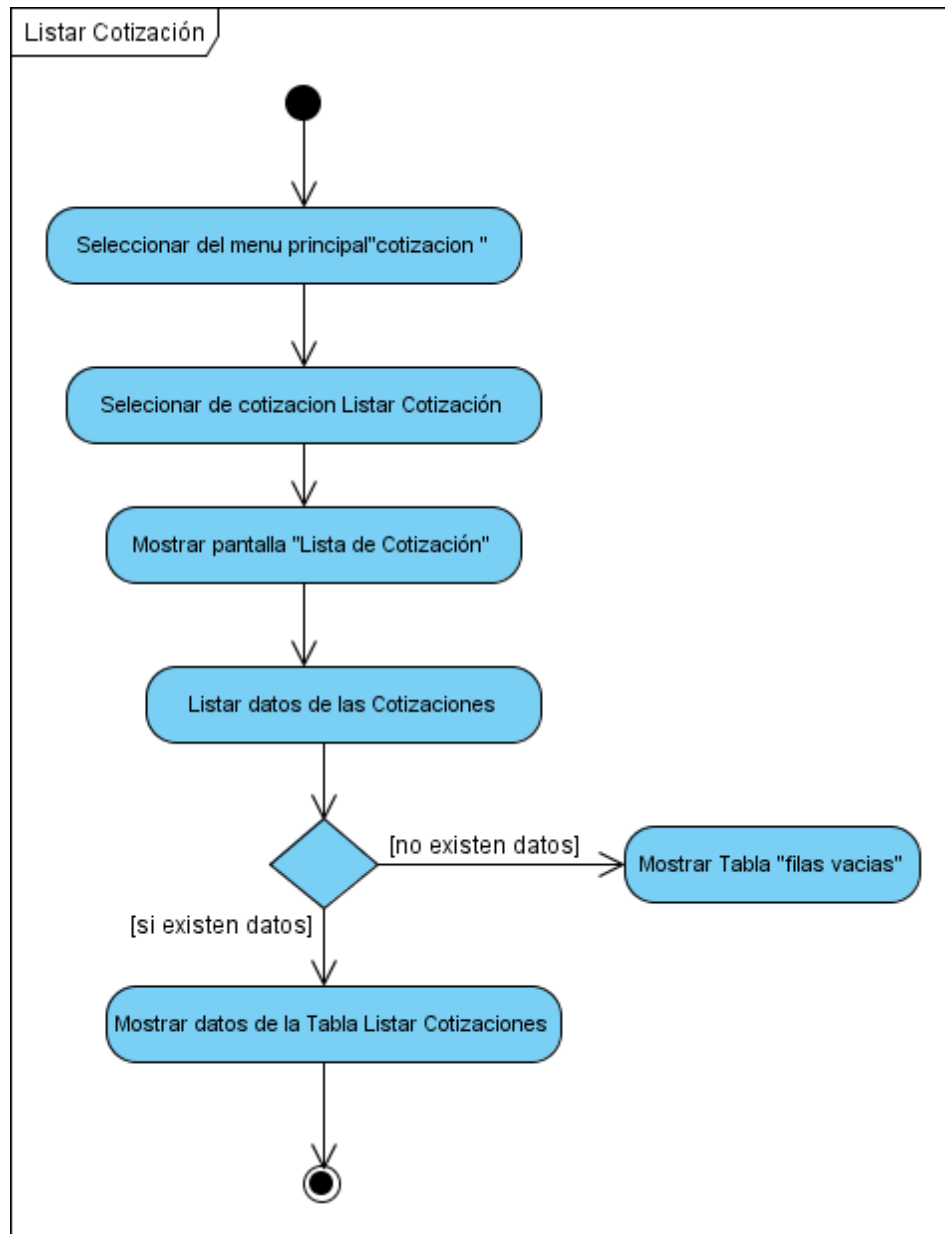


Fig.II.36 diagrama de actividad-Listar Cotizaciones

Diagrama de Actividad: Ver Cotización

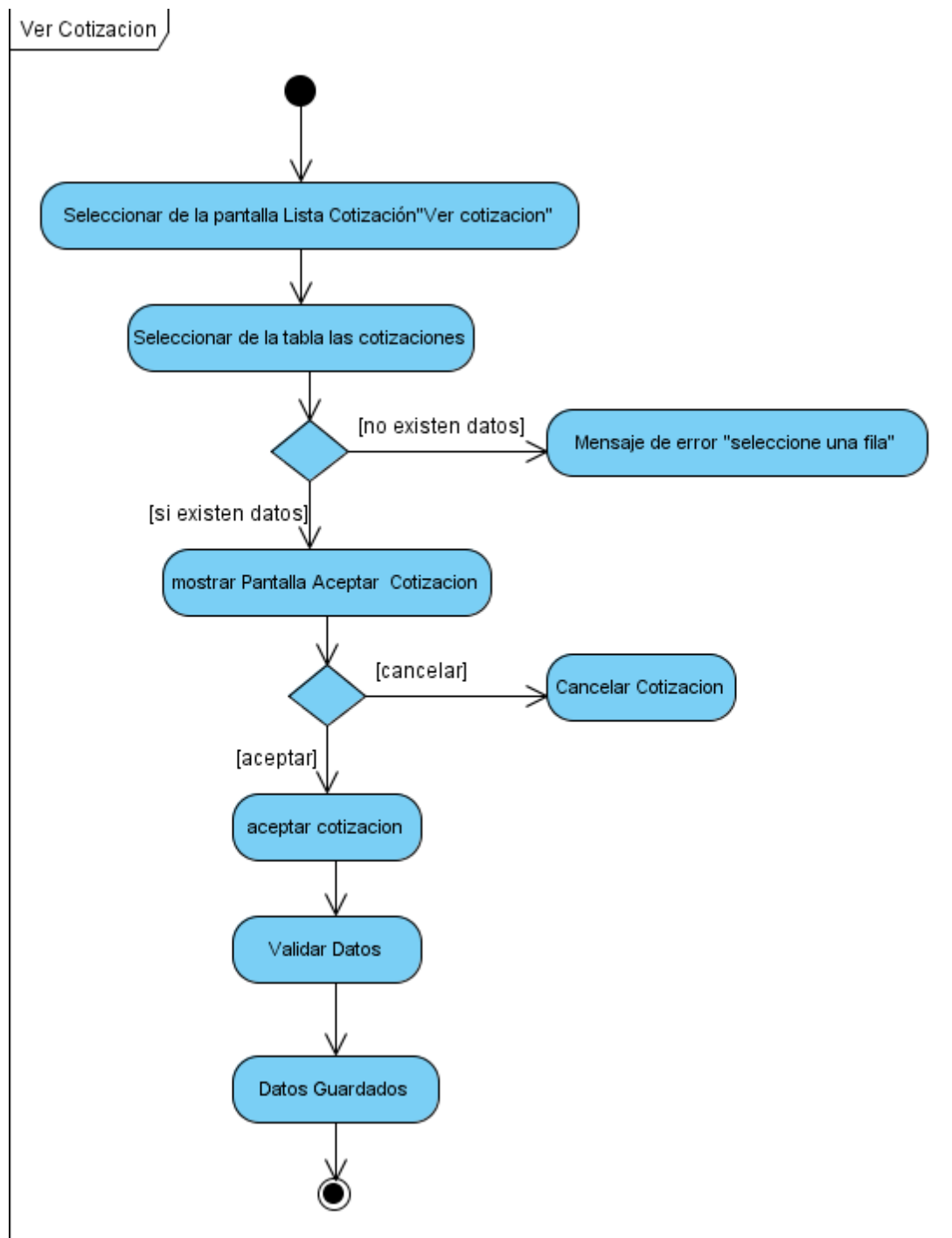


Fig.II.37 diagrama de actividad-Ver Cotización

ADMINISTRAR APROBACIÓN

Diagrama de Actividad: Listar Aprobación

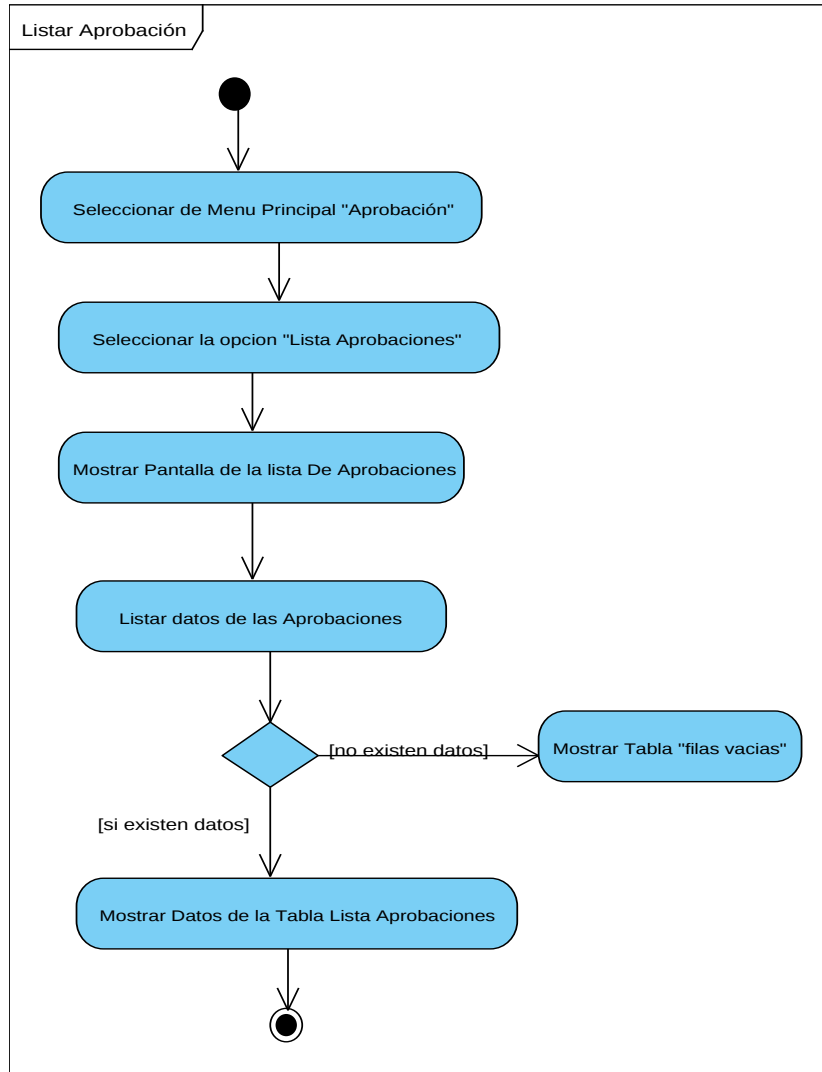


Fig.II.38 diagrama de actividad-Listar Aprobación

Diagrama de Actividad: Crear Aprobación

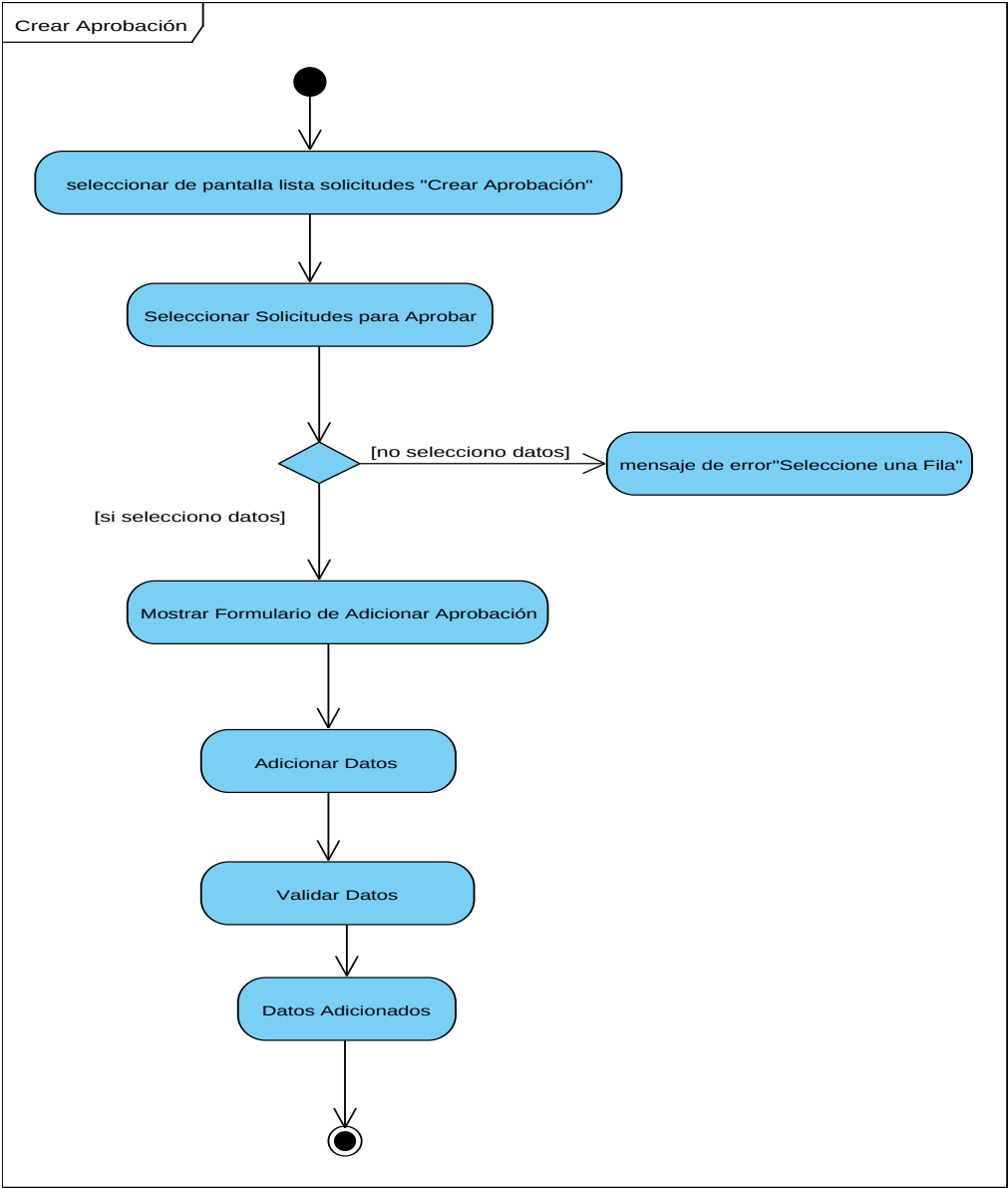


Fig.II.39 diagrama de actividad-Crear Aprobación

Diagrama de Actividad: Modificar Aprobación

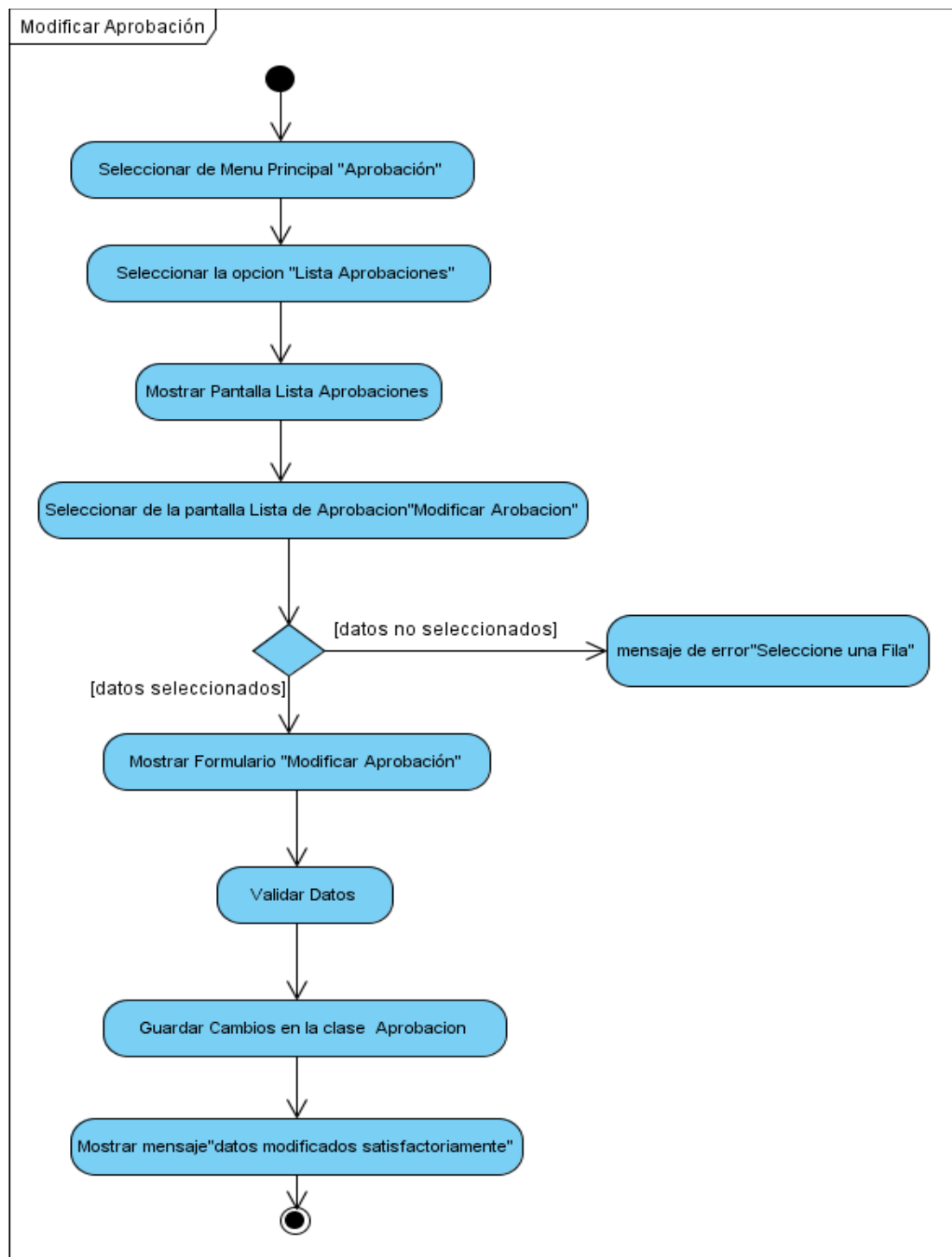


Fig.II.40 diagrama de actividad-Modificar Aprobación

ADMINISTRAR COMPRA

Diagrama de Actividad: Listar Compras

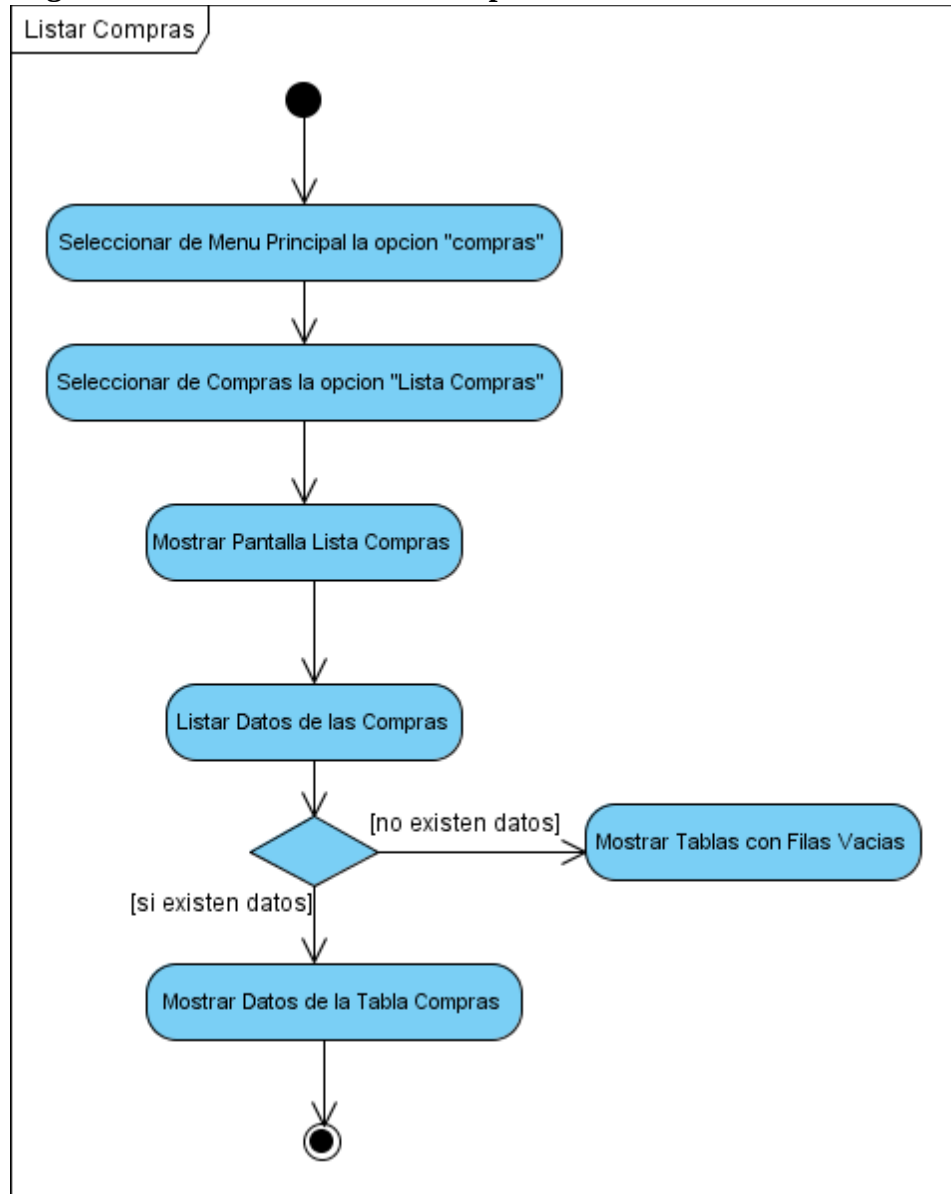


Fig.II.41 diagrama de actividad-Listar Compras

Diagrama de Actividad: Modificar Compra

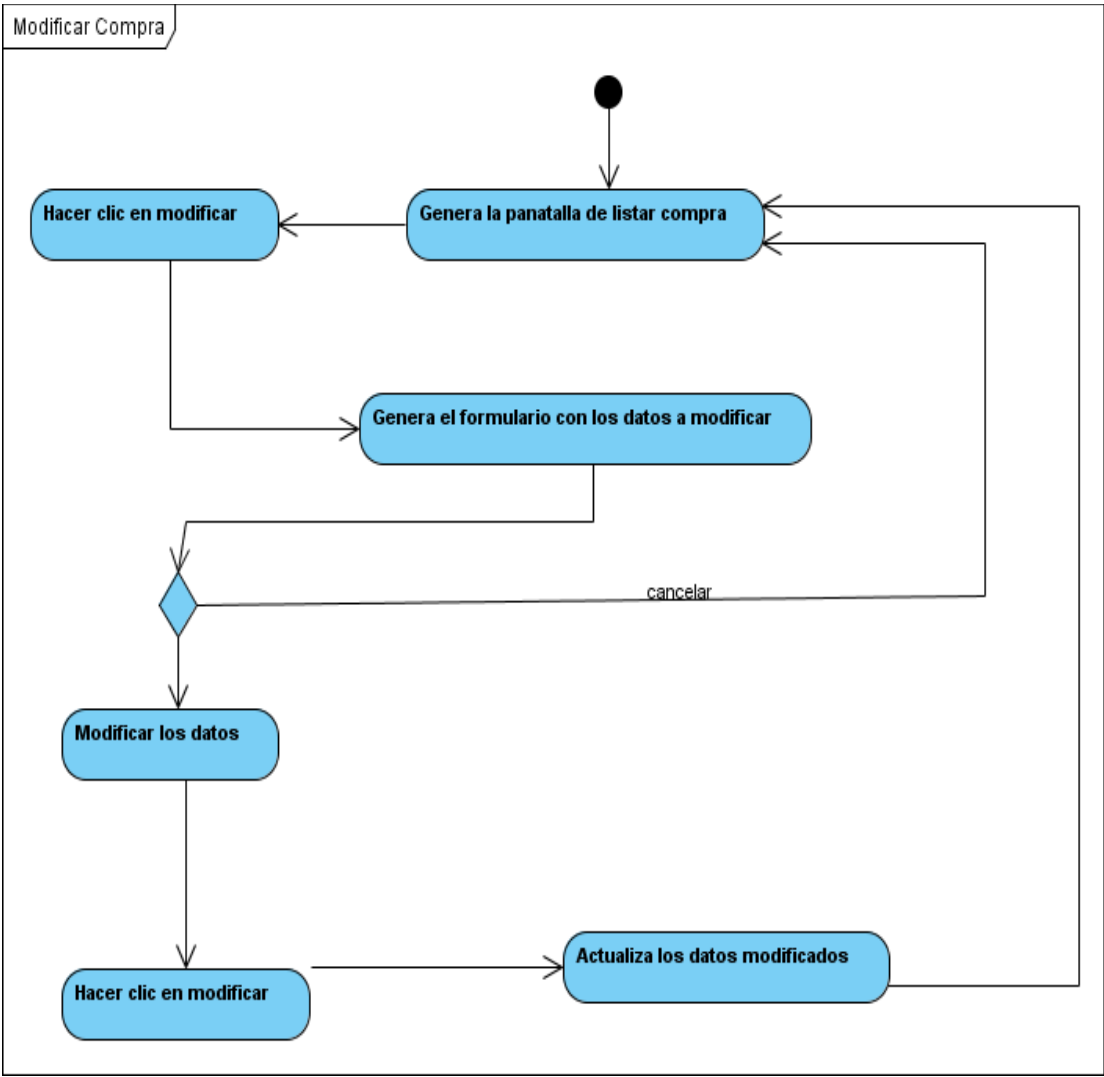


Fig.II.42 diagrama de actividad-Modificar Compra

Diagrama de Actividad: Listar Proveedores

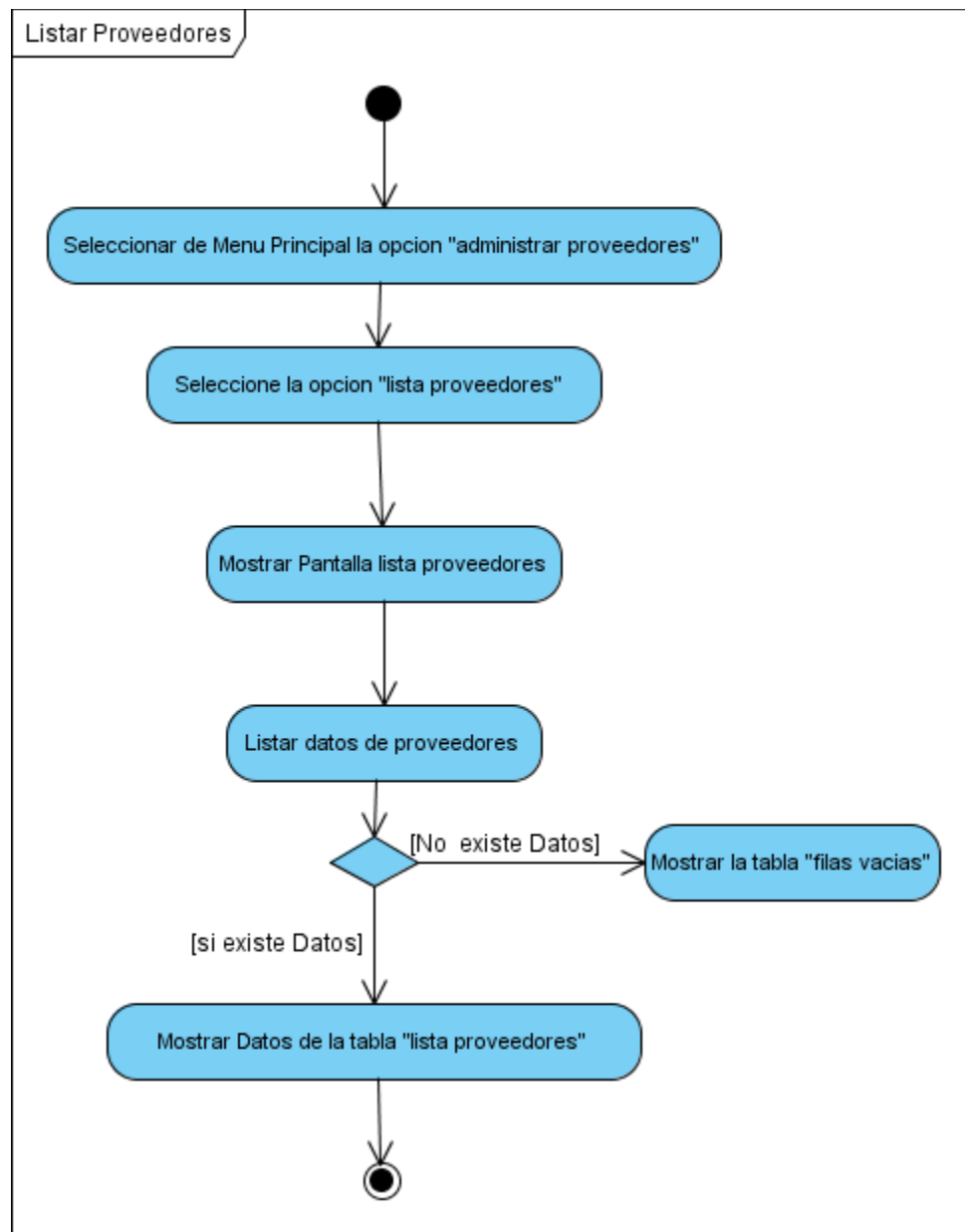


Fig.II.43 diagrama de actividad-Listar Proveedores

Diagrama de Actividad: Adicionar Proveedor

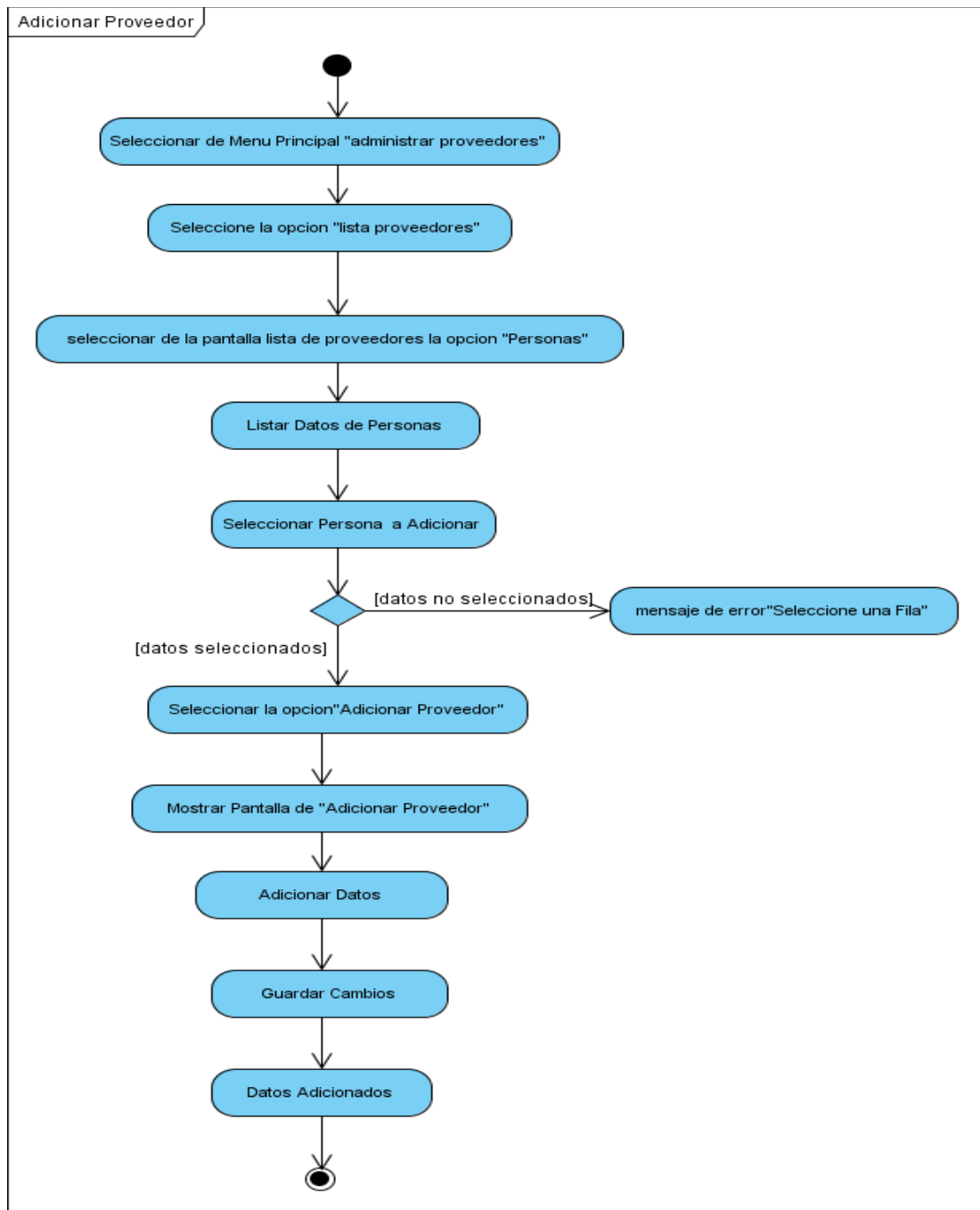


Fig.II.44 diagrama de actividad-Adicionar Proveedor

Diagrama de Actividad: Modificar Proveedores

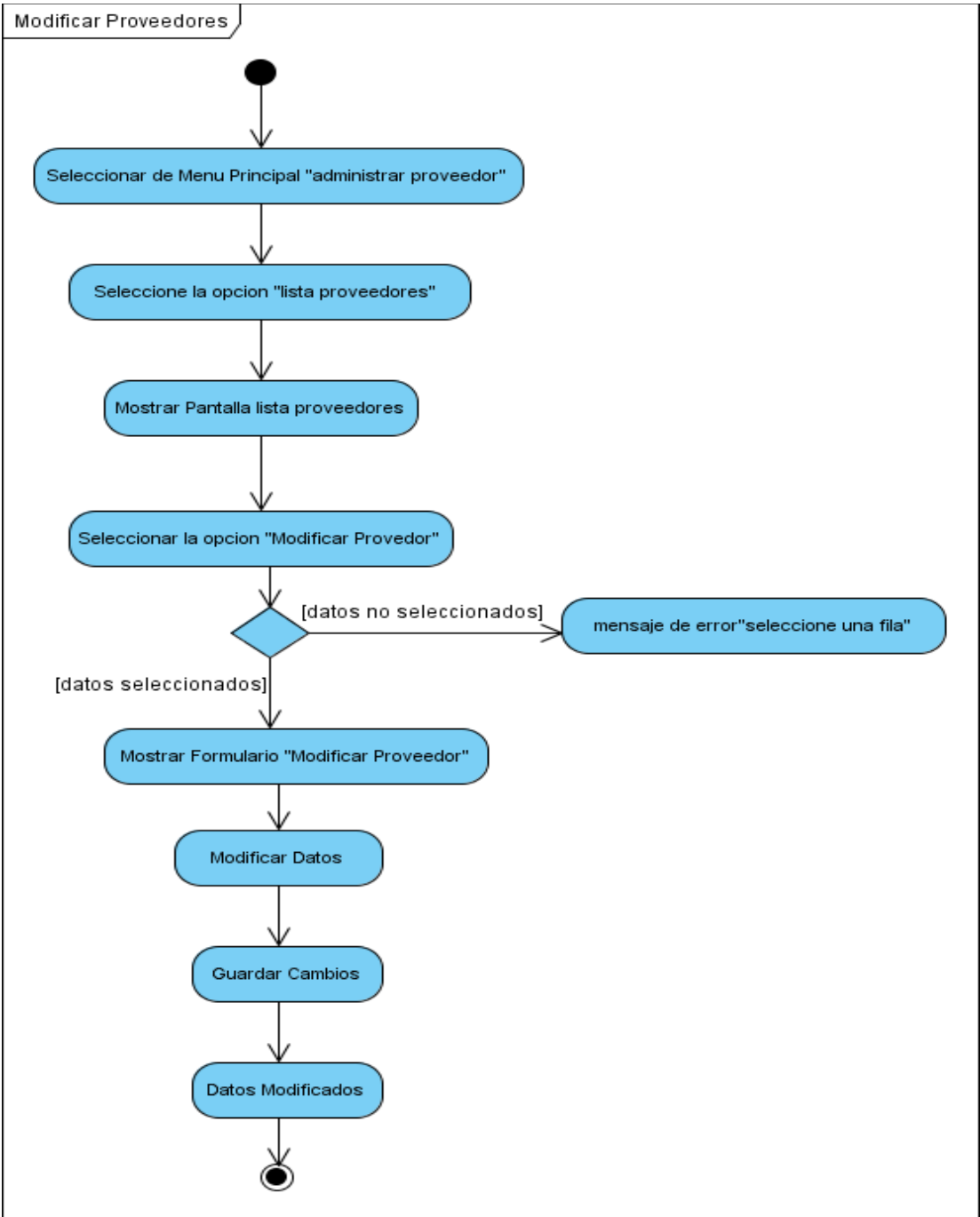


Fig.II.45 diagrama de actividad-Modificar Proveedores

Diagrama de Actividad: Listar Material

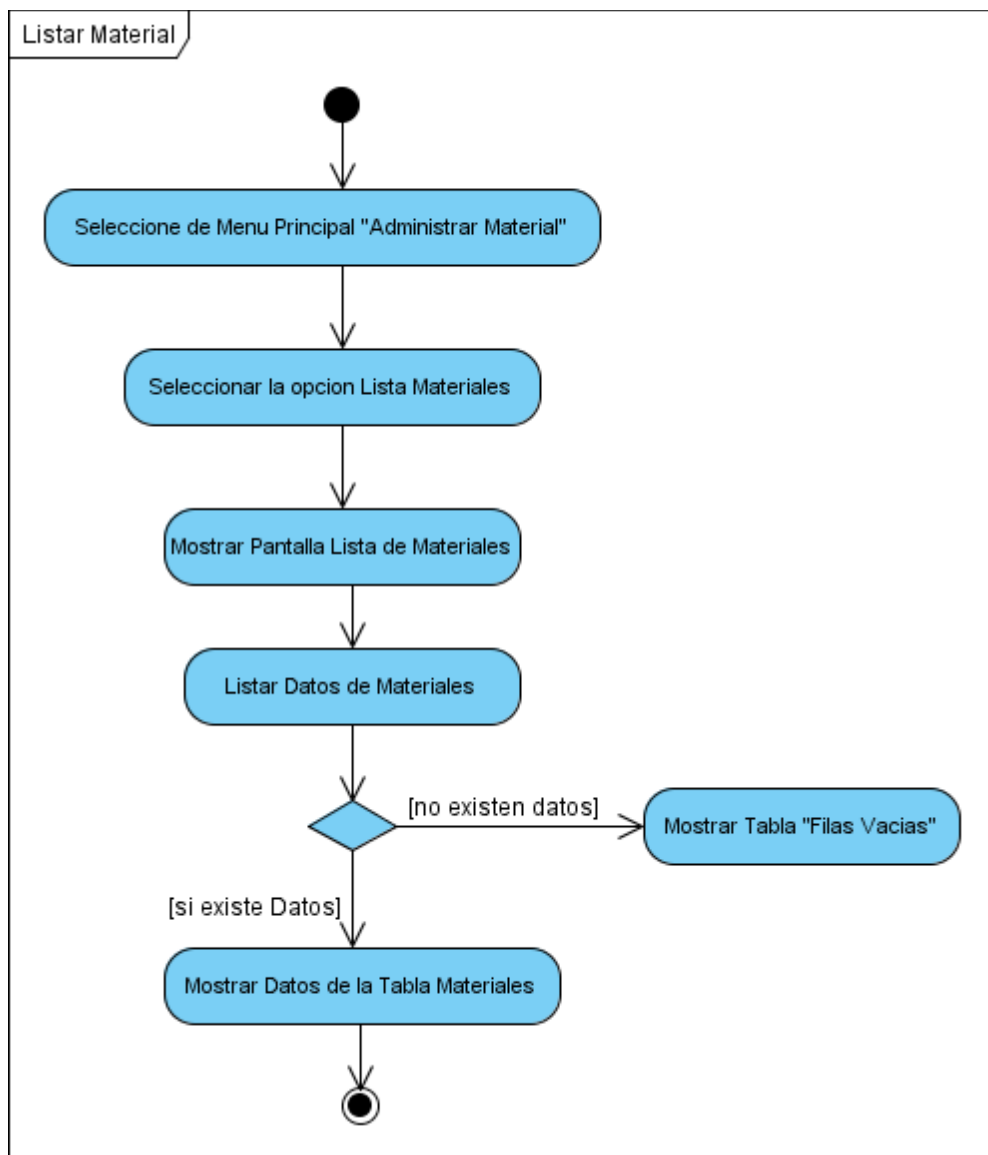


Fig.II.46 diagrama de actividad-Listar Material

Diagrama de Actividad: Adicionar Material

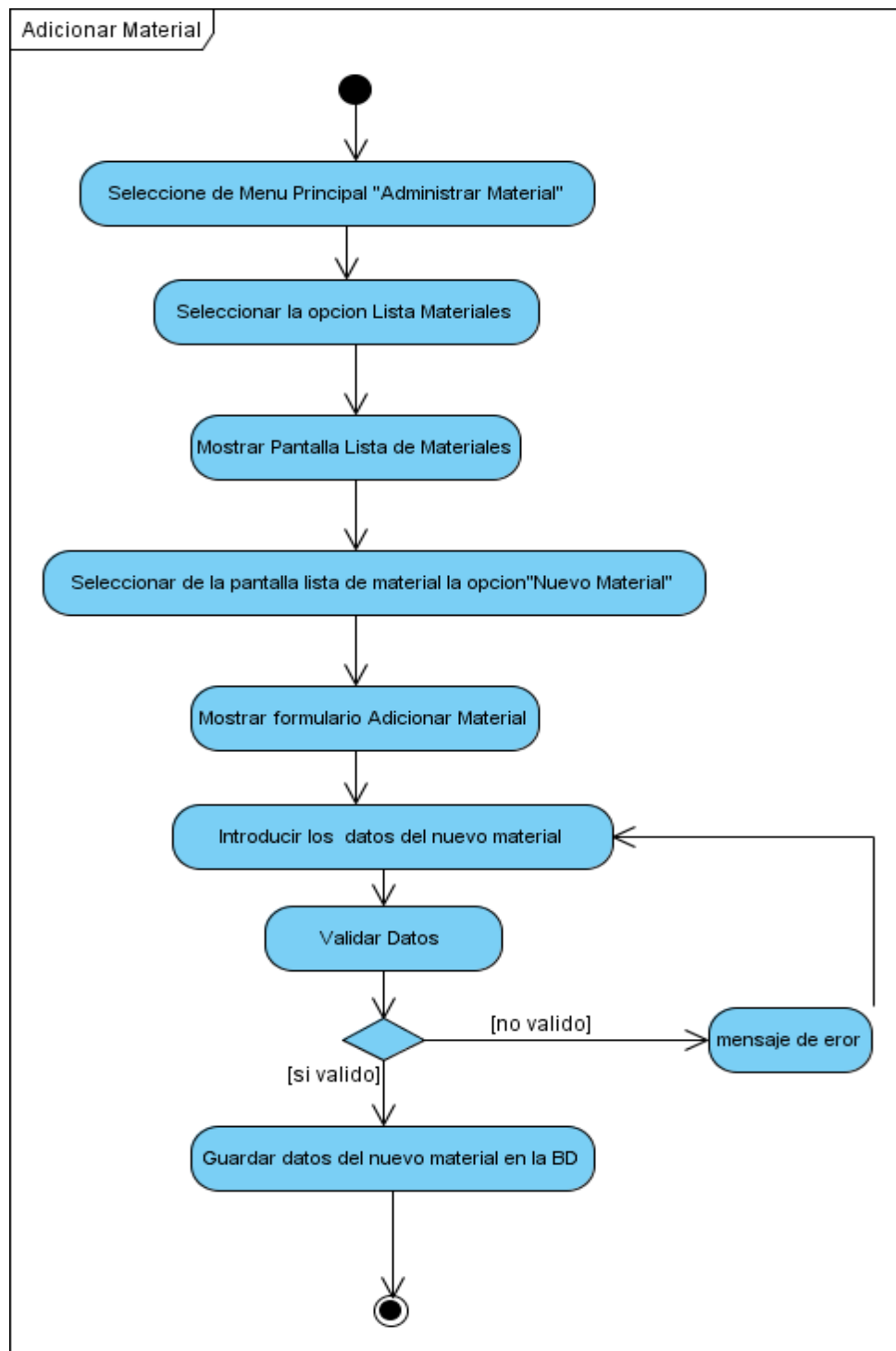


Fig.II.47 diagrama de actividad-Adicionar Material

Diagrama de Actividad: Modificar Material

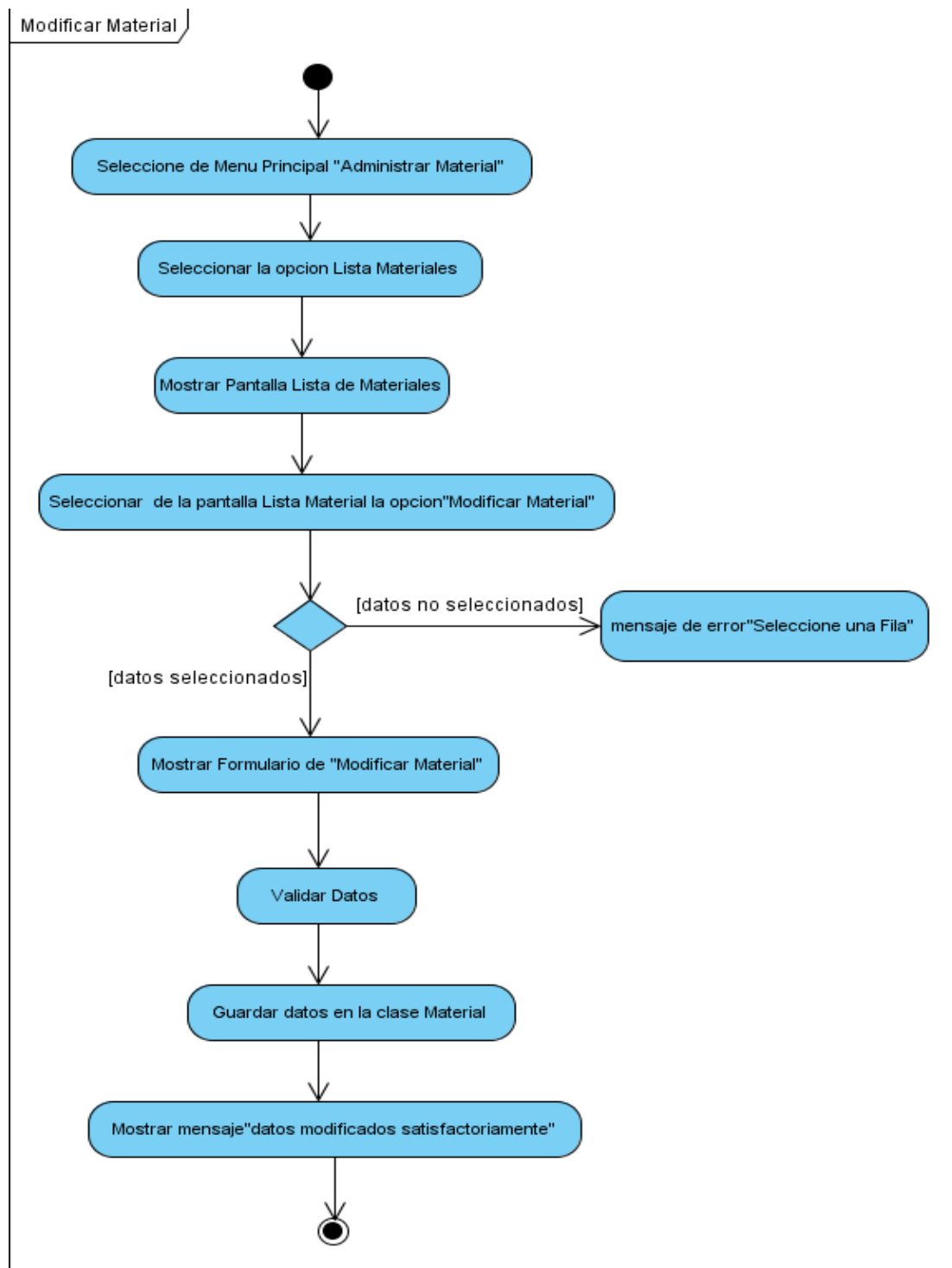


Fig.II.48 diagrama de actividad-Modificar Material

ADMINISTRAR CATEGORIAS

Diagrama de Actividad: Listar Categorías

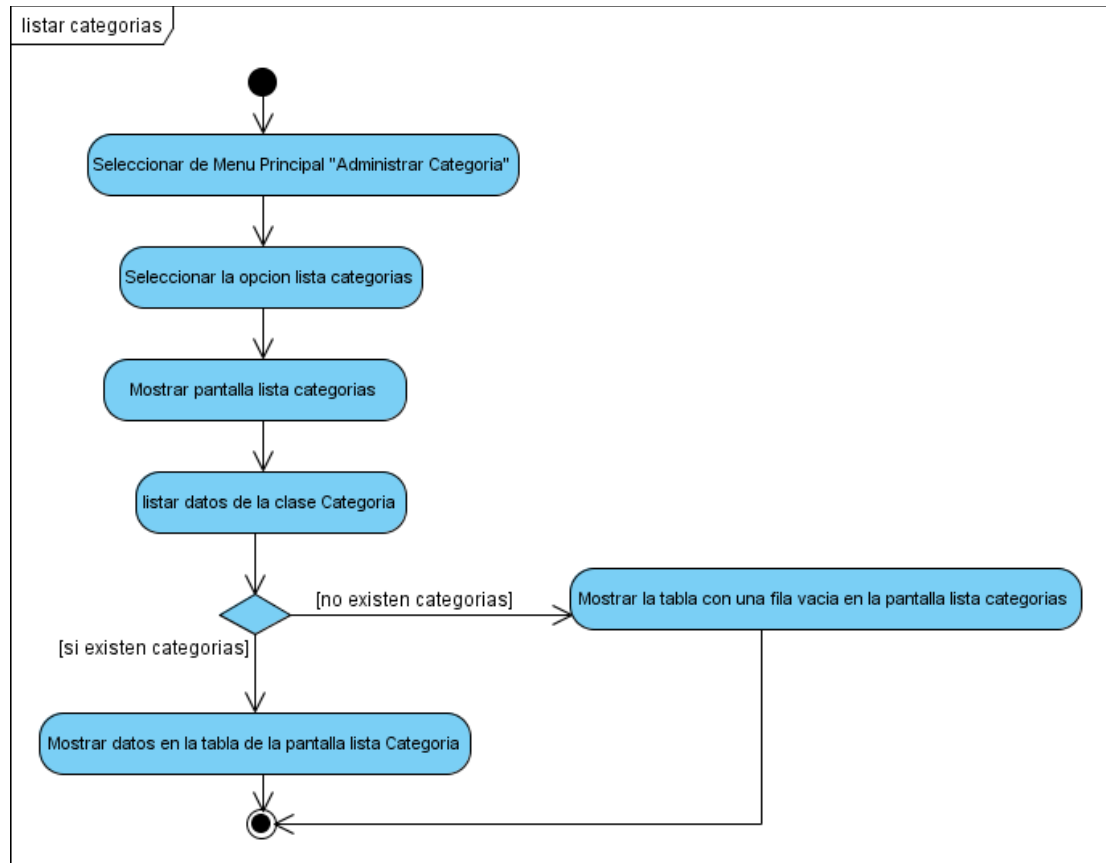


Fig.II.49 diagrama de actividad-Listar Categorías

Diagrama de Actividad: Adicionar Categorías

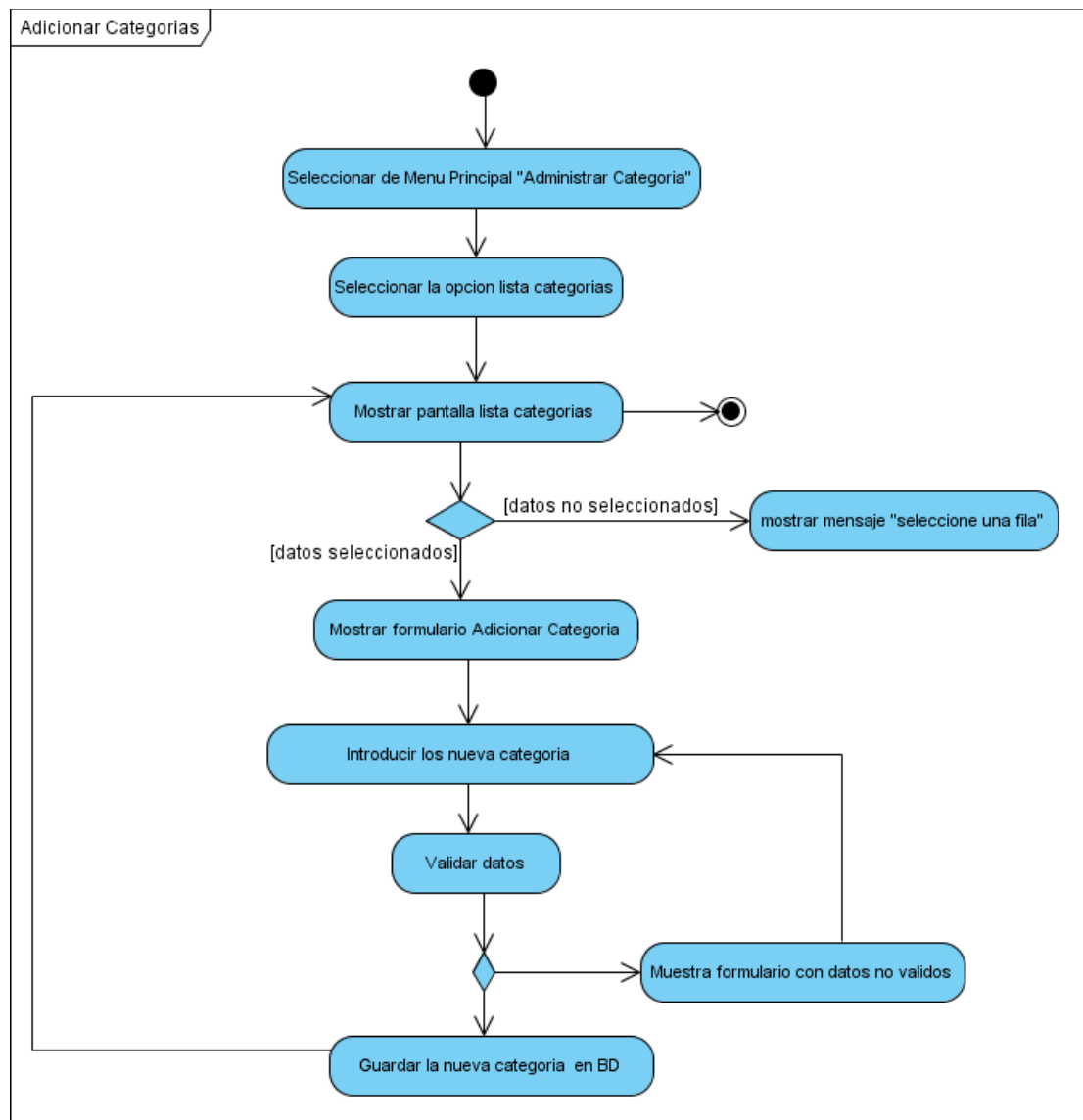


Fig.II.50 diagrama de actividad-Adicionar Categorías

Diagrama de Actividad: Modificar Categorías

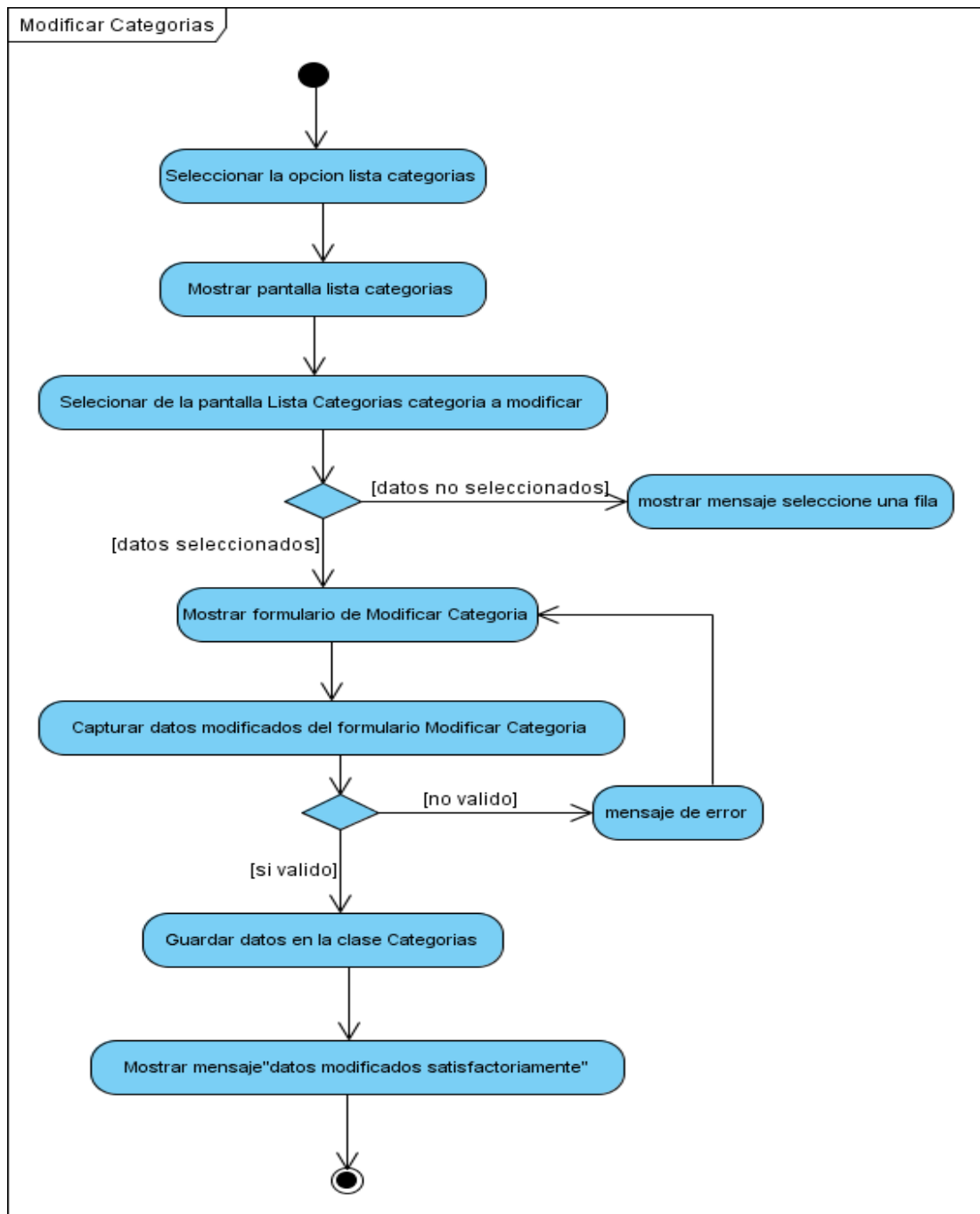


Fig.II.51 diagrama de actividad-Modificar Categorías

ADMINISTRAR ALMACEN

Diagrama de Actividad: Listar Entradas de Material

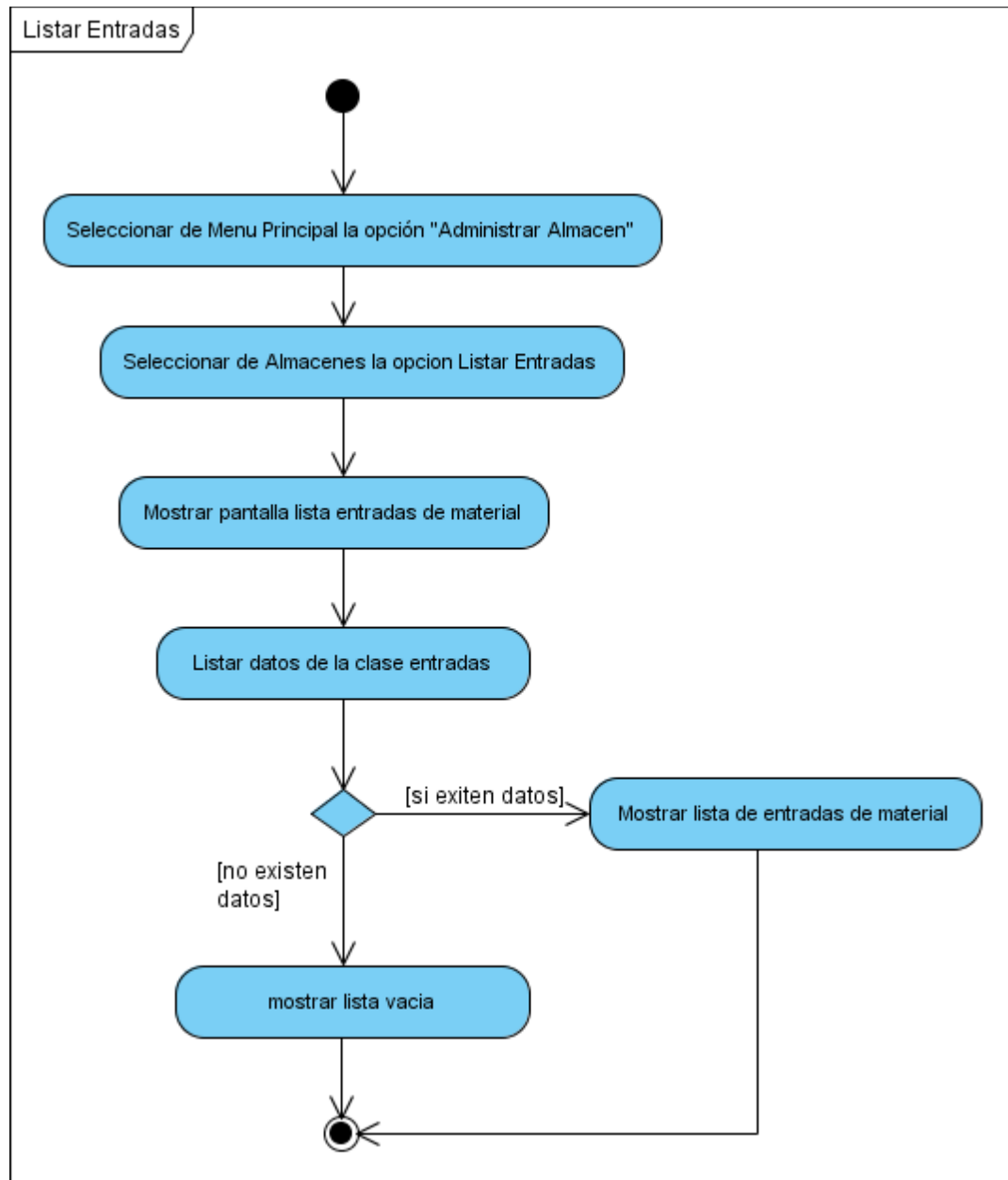


Fig.II.52 diagrama de actividad-Listar Entradas

Diagrama de Actividad: Modificar Entradas

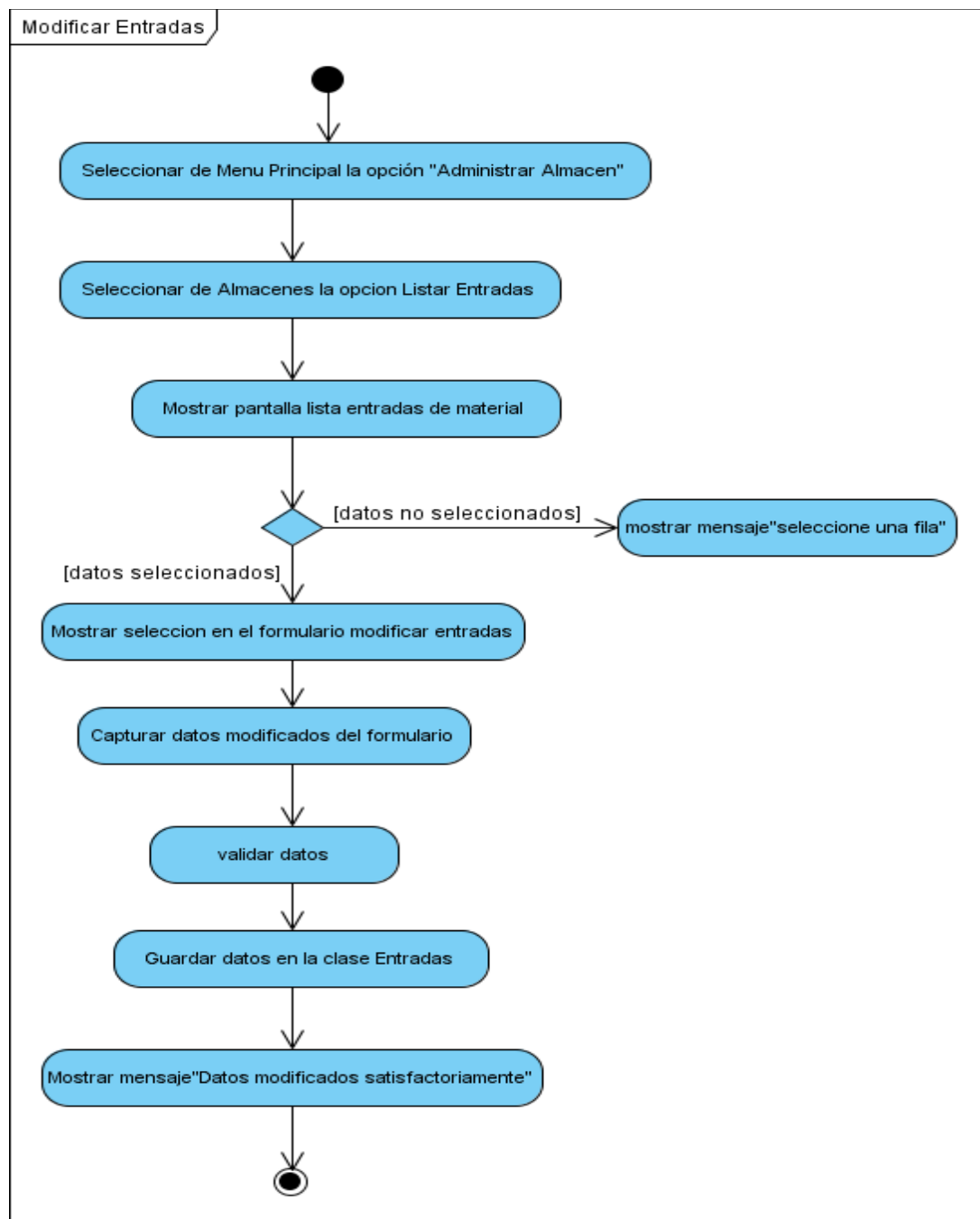


Fig.II.53 diagrama de actividad-Modificar Entrada

Diagrama de Actividad: Listar Salidas de Material

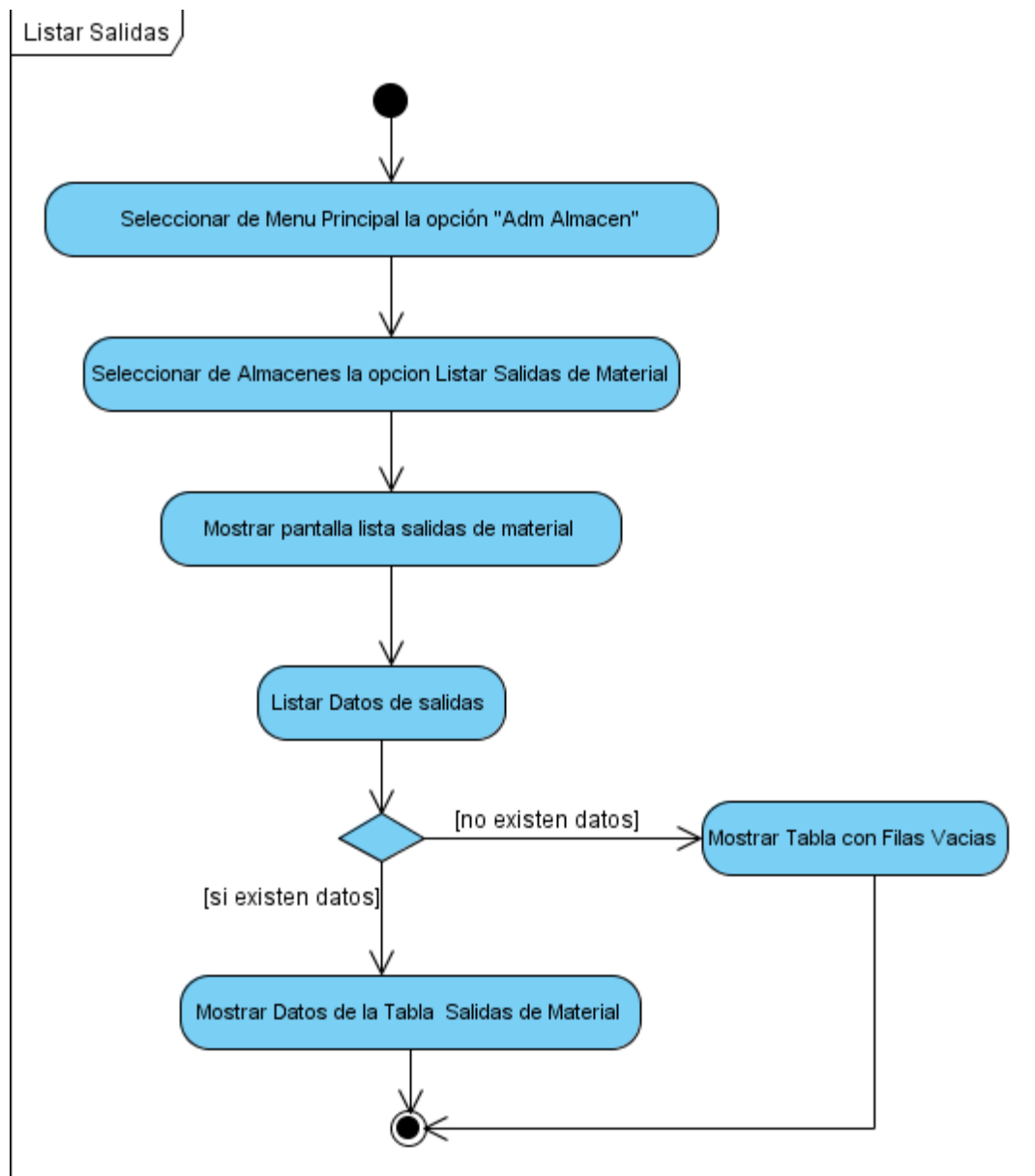


Fig.II.54 diagrama de actividad-Listar Salidas

Diagrama de Actividad: Modificar Salidas de Material

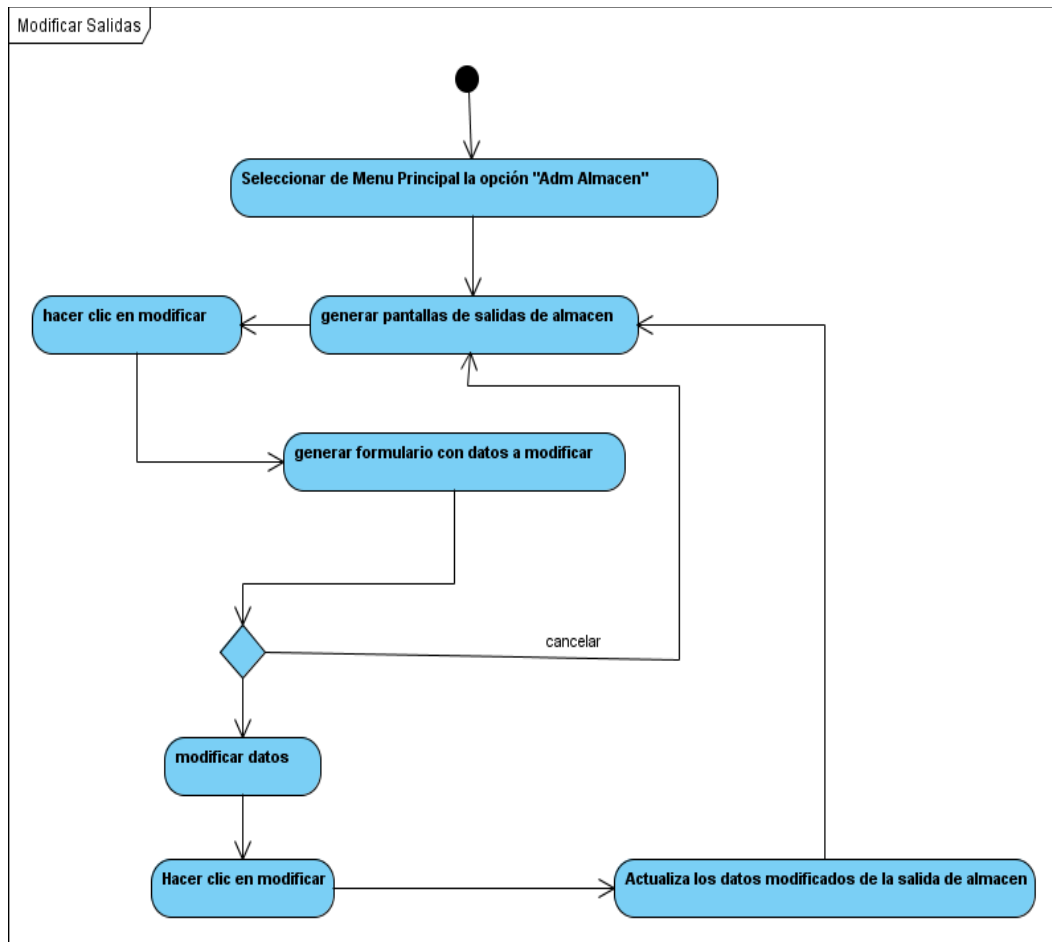


Fig.II. 55 diagrama de actividad-Modificar Salidas de Material

Diagrama de Actividad: Administrar Reportes

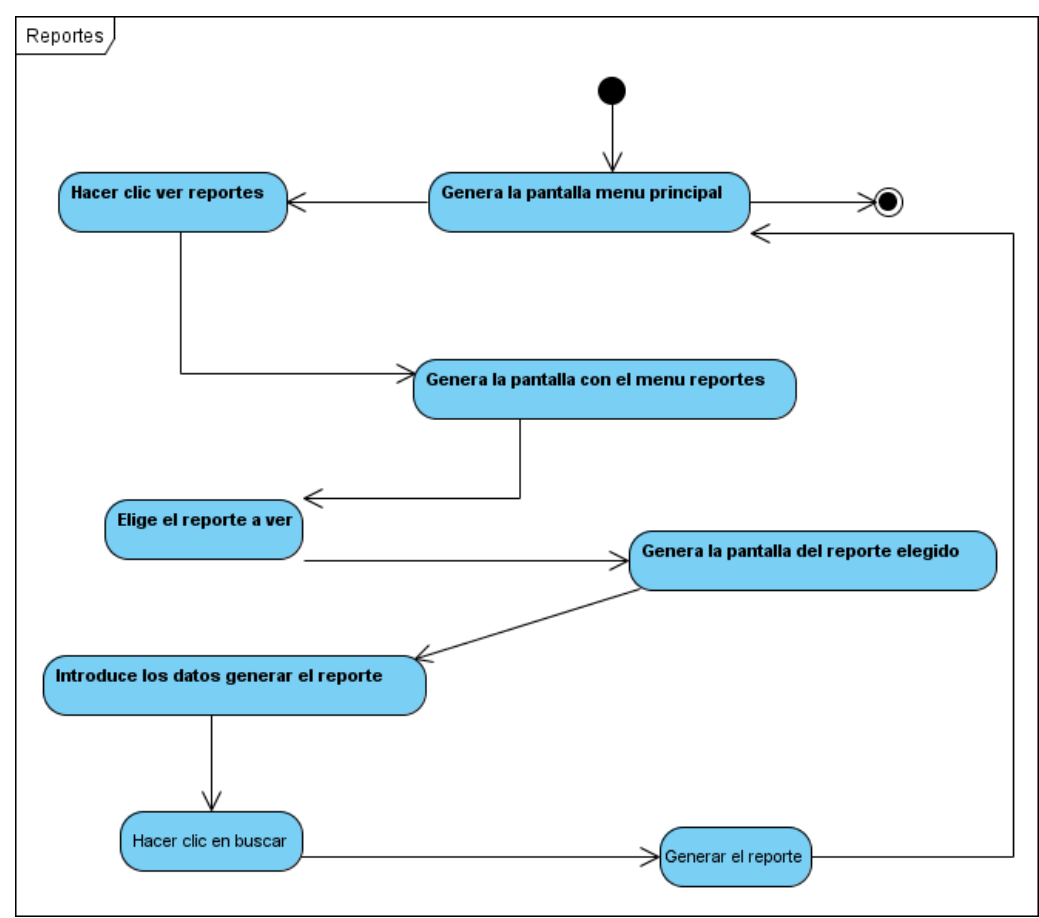


Fig.II.56 diagrama de actividad-Administrar Reportes

II.2.7.MODELO DE DIAGRAMAS DE SECUENCIA

Introducción

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

UML incluye la notación de los diagramas de Secuencia.

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

Propósito

- Comprender la Estructura y la dinámica del sistema diseñado para la Organización.
- Comprender la interacciona de los actores del sistema.

Alcance

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

DIAGRAMAS DE SECUENCIAS

Diagrama de Secuencia: Ingresar al Sistema

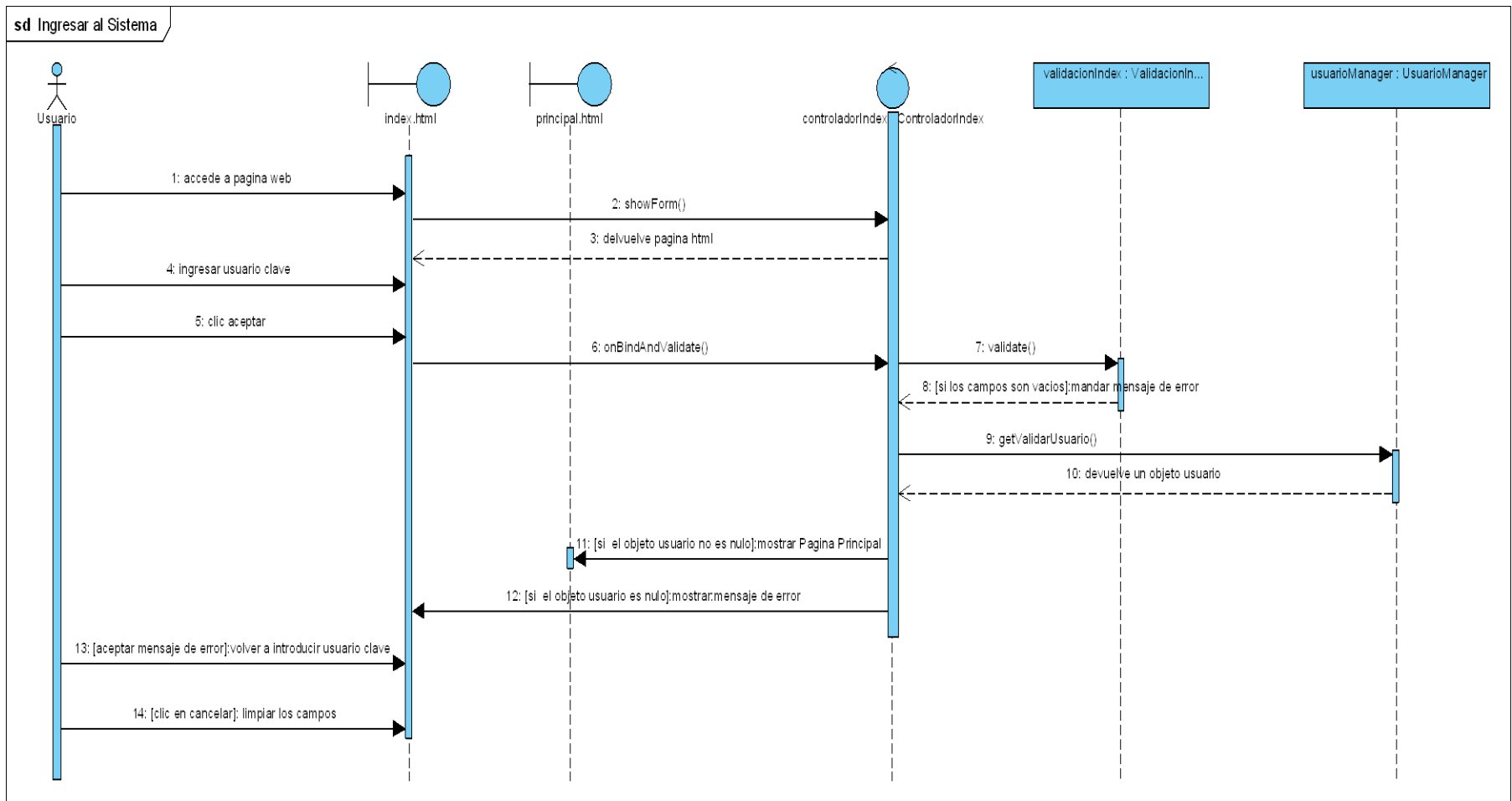


Fig.II.57 diagrama de secuencia-Ingresar al Sistema

Diagrama de Secuencia: Listar Usuario

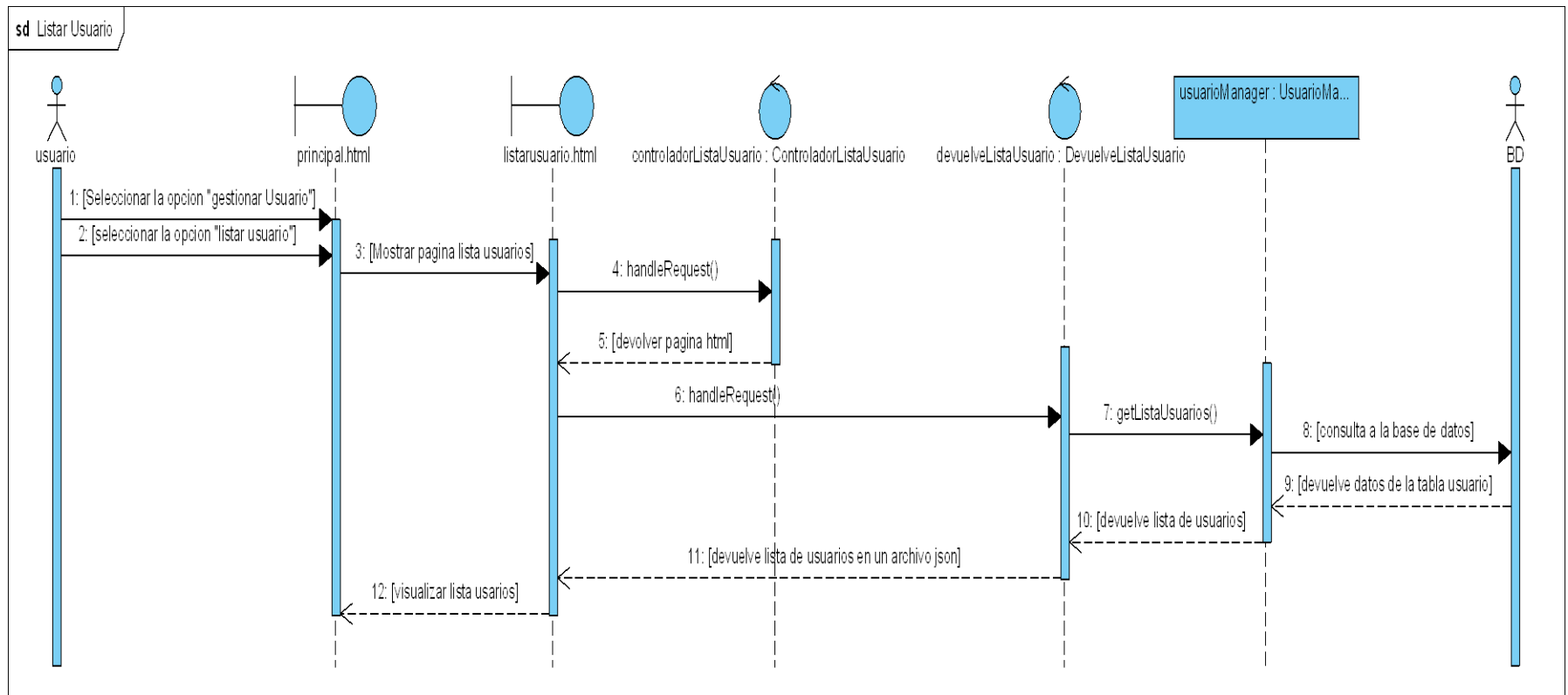


Fig.II.58 diagrama de secuencia-Listar Usuario

Diagrama de Secuencia: Adicionar Usuario

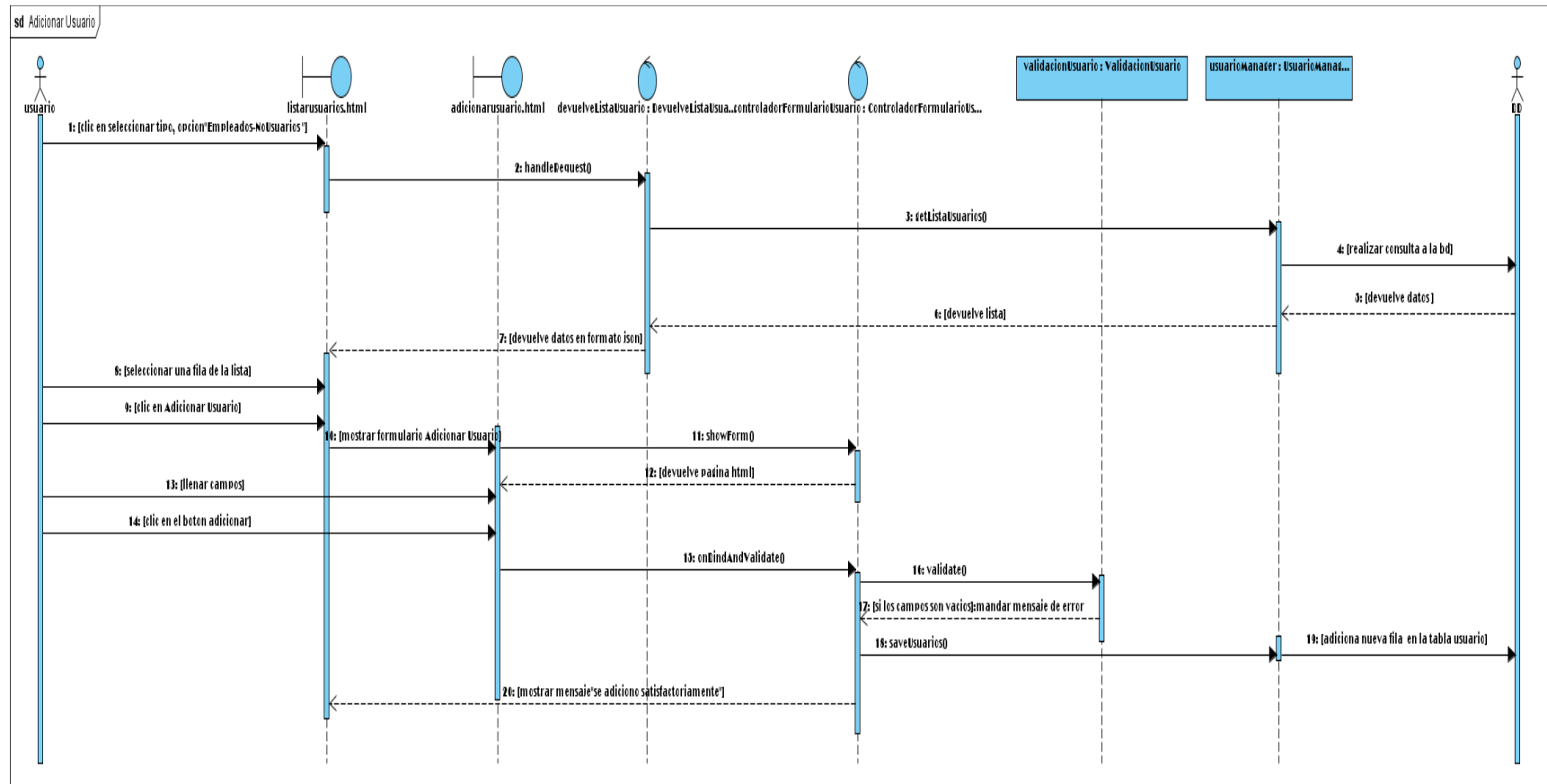


Fig.II.59 diagrama de secuencia-Adicionar Usuarios

Diagrama de Secuencia: Modificar Usuarios

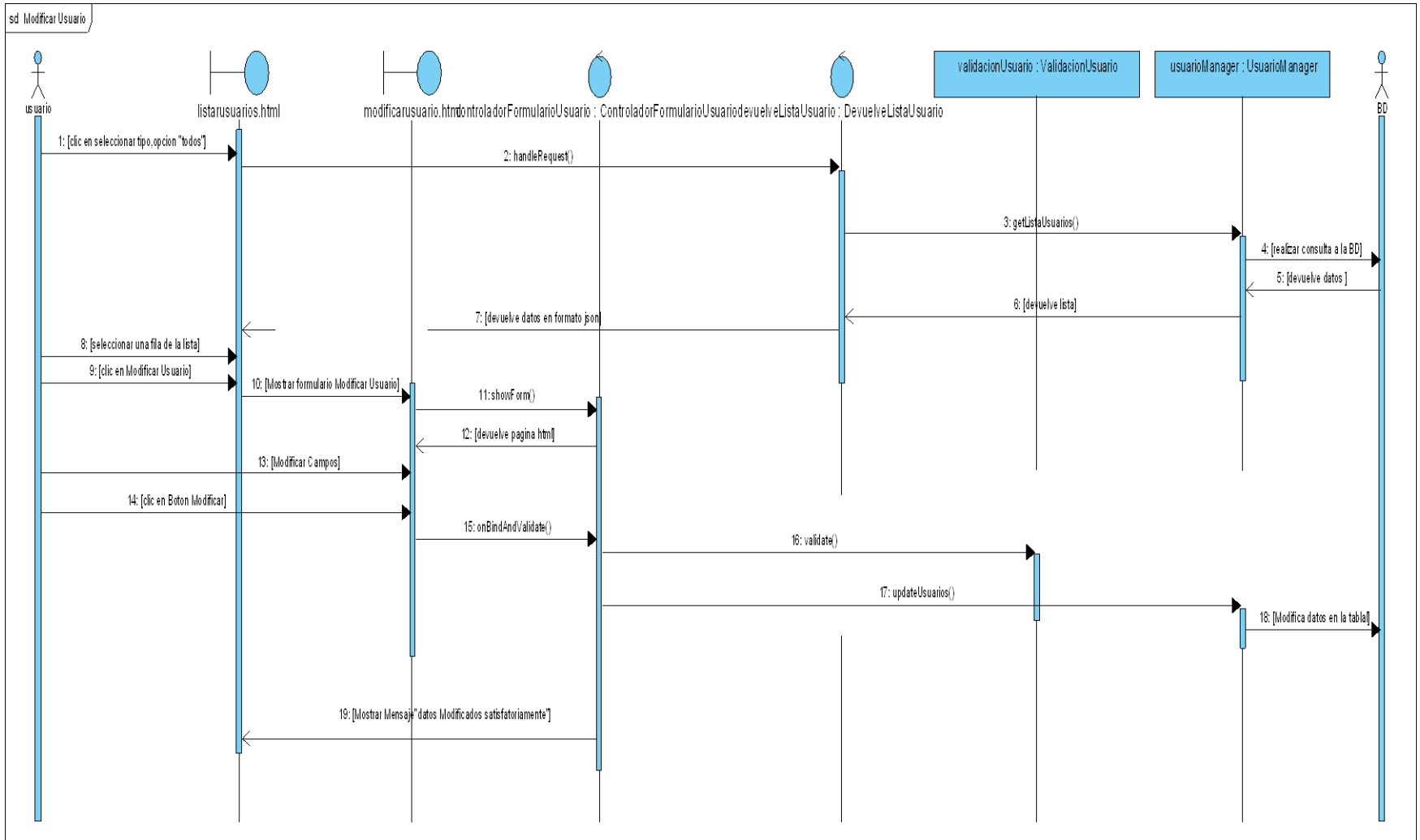


Fig.II.60 diagrama de secuencia-Modificar Usuarios

Diagrama de Secuencia: Listar Personas

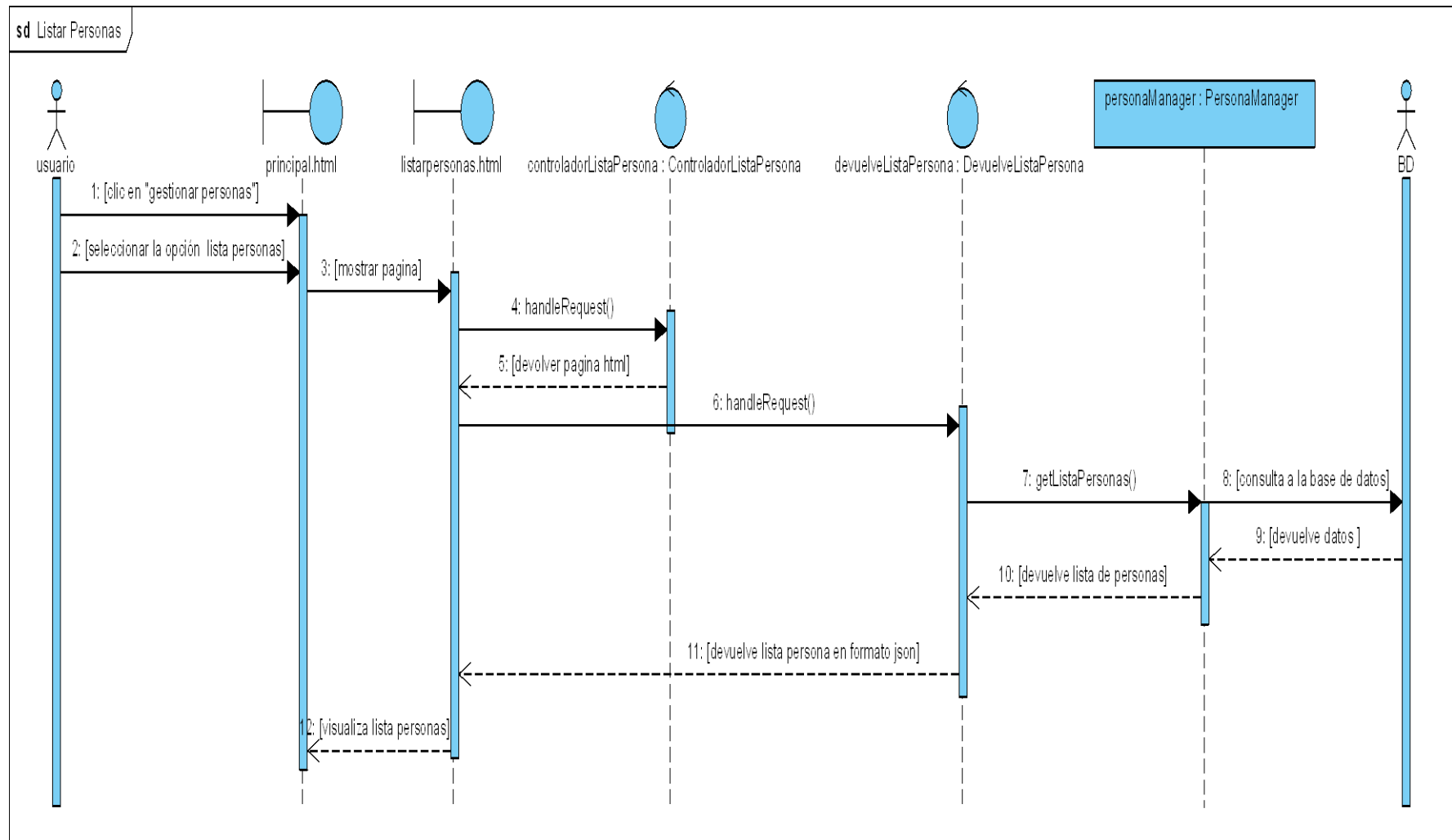


Fig.II.61 diagrama de secuencia-Listar Personas

Diagrama de Secuencia: Lista Empleados

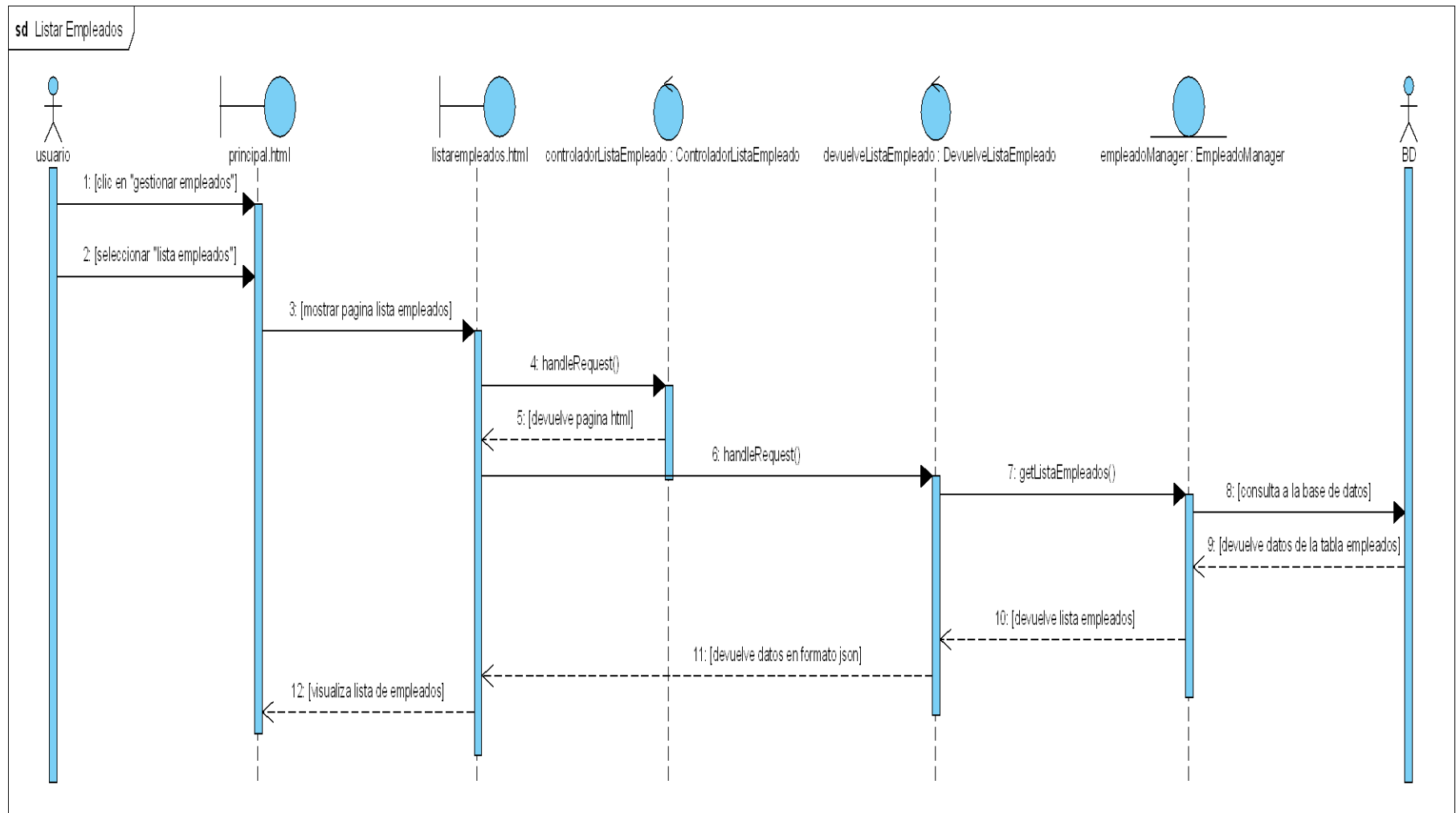


Fig.II.62 diagrama de secuencia-Lista Empleados

Diagrama de Secuencia: Adicionar Empleados

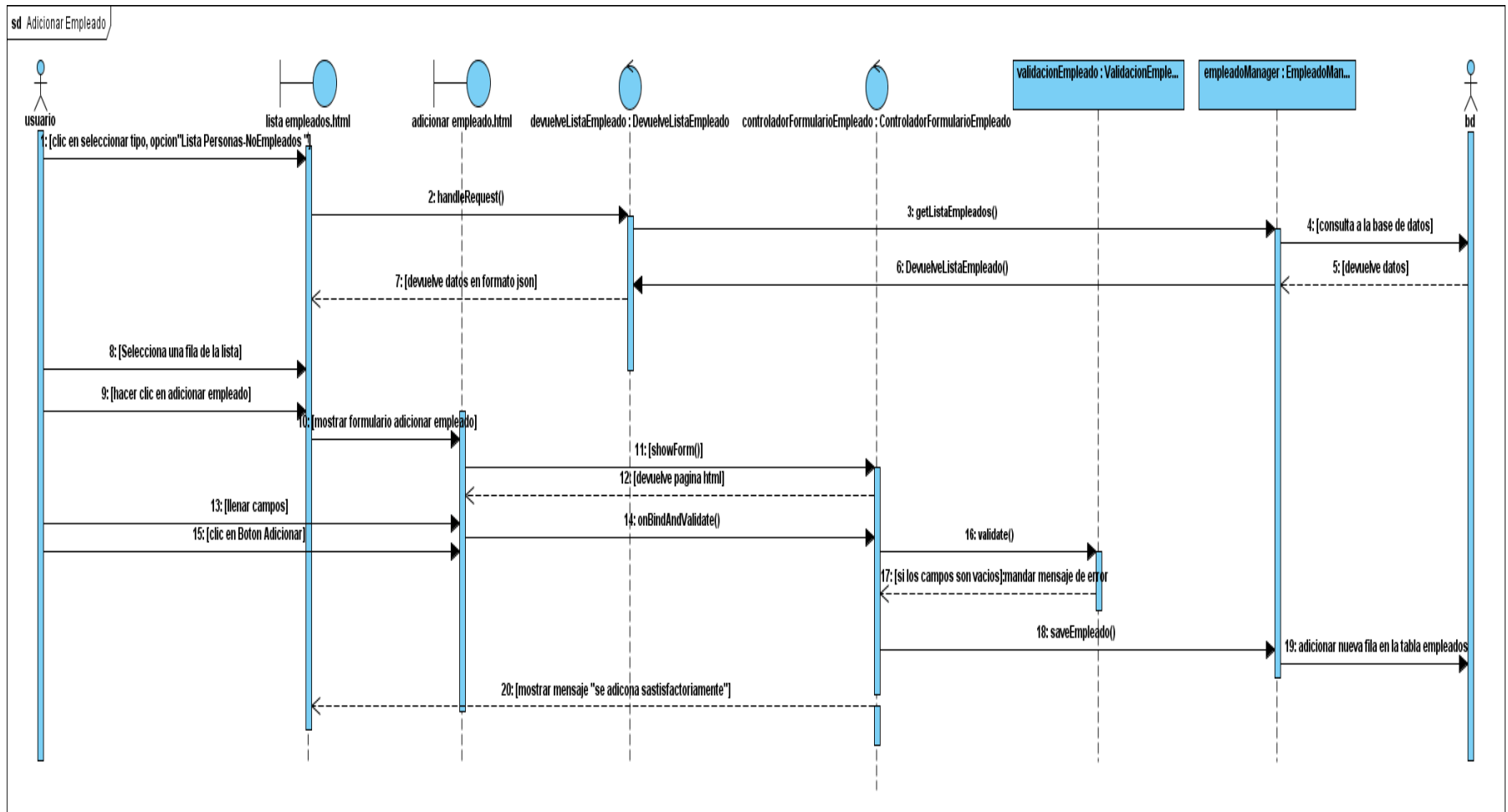


Fig.II.63 diagrama de secuencia-Adicionar Empleados

Diagrama de Secuencia: Modificar Empleado

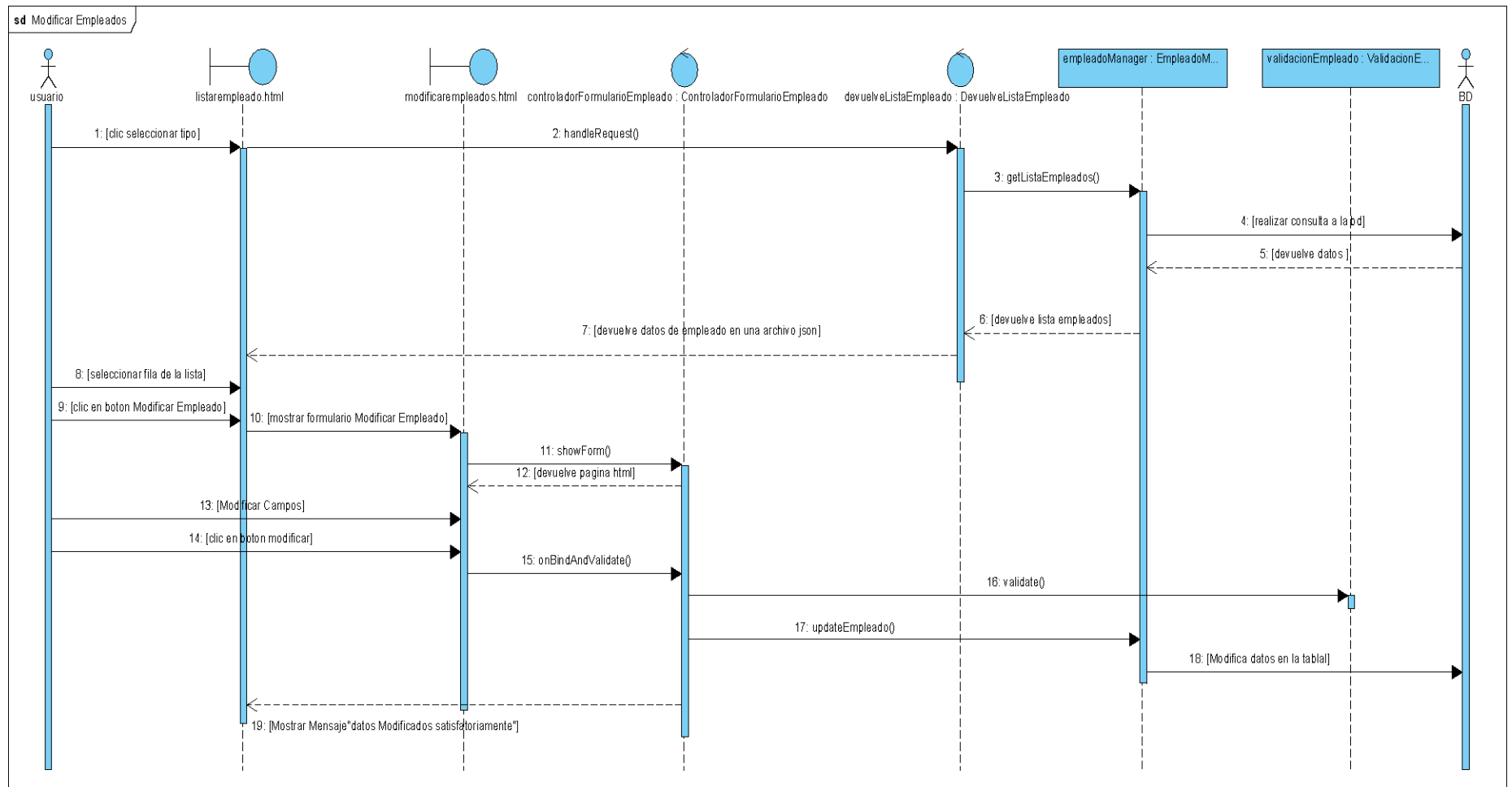


Fig.II.64 diagrama de secuencia-Modificar Empleados

Diagrama de Secuencia: Lista Proveedores

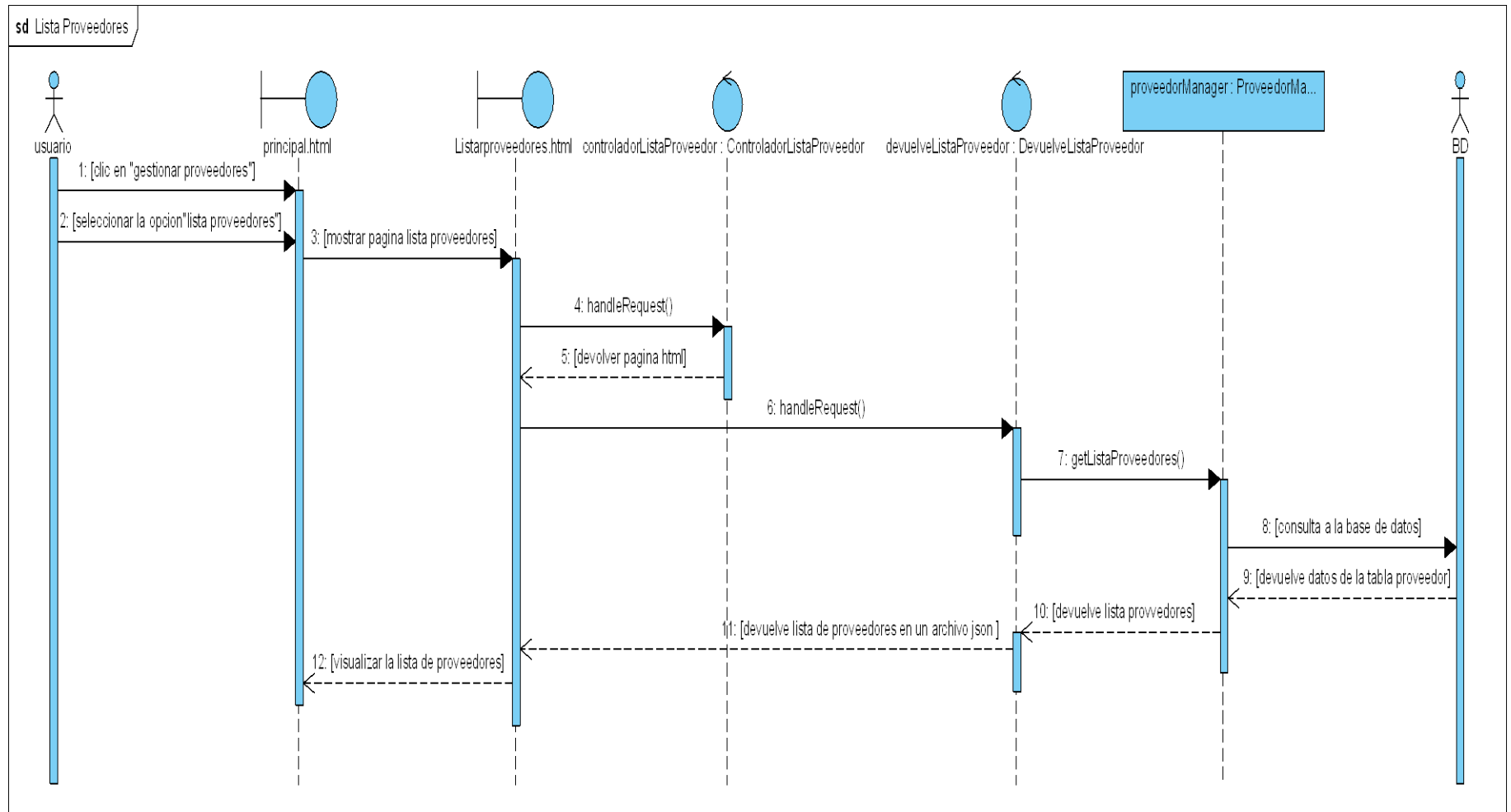


Fig.II.65 diagrama de secuencia-Lista Proveedores

Diagrama de Secuencia: Listar Materiales

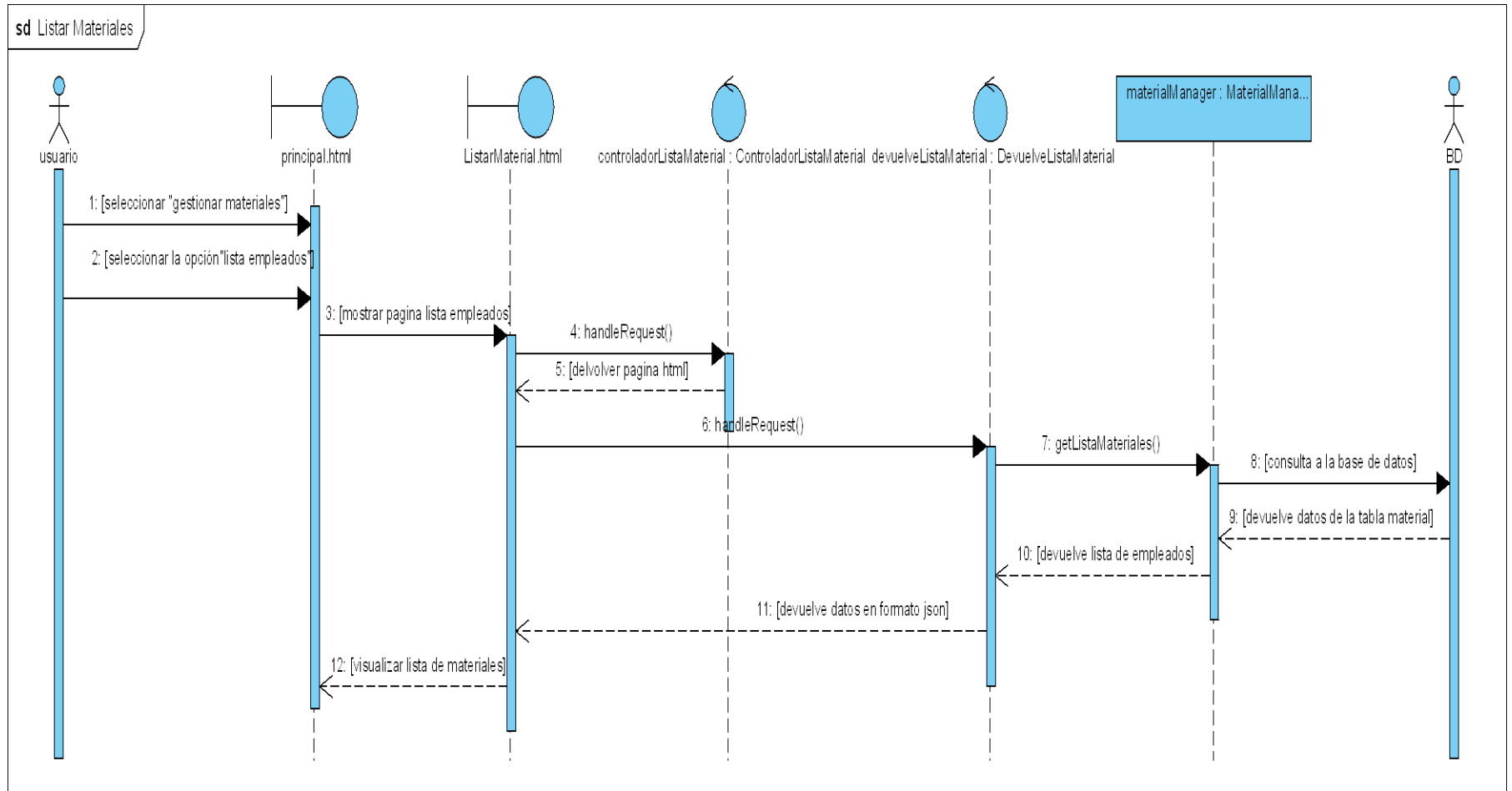


Fig.II.66 diagrama de secuencia-Listar Materiales

Diagrama de Secuencia: Adicionar Materiales

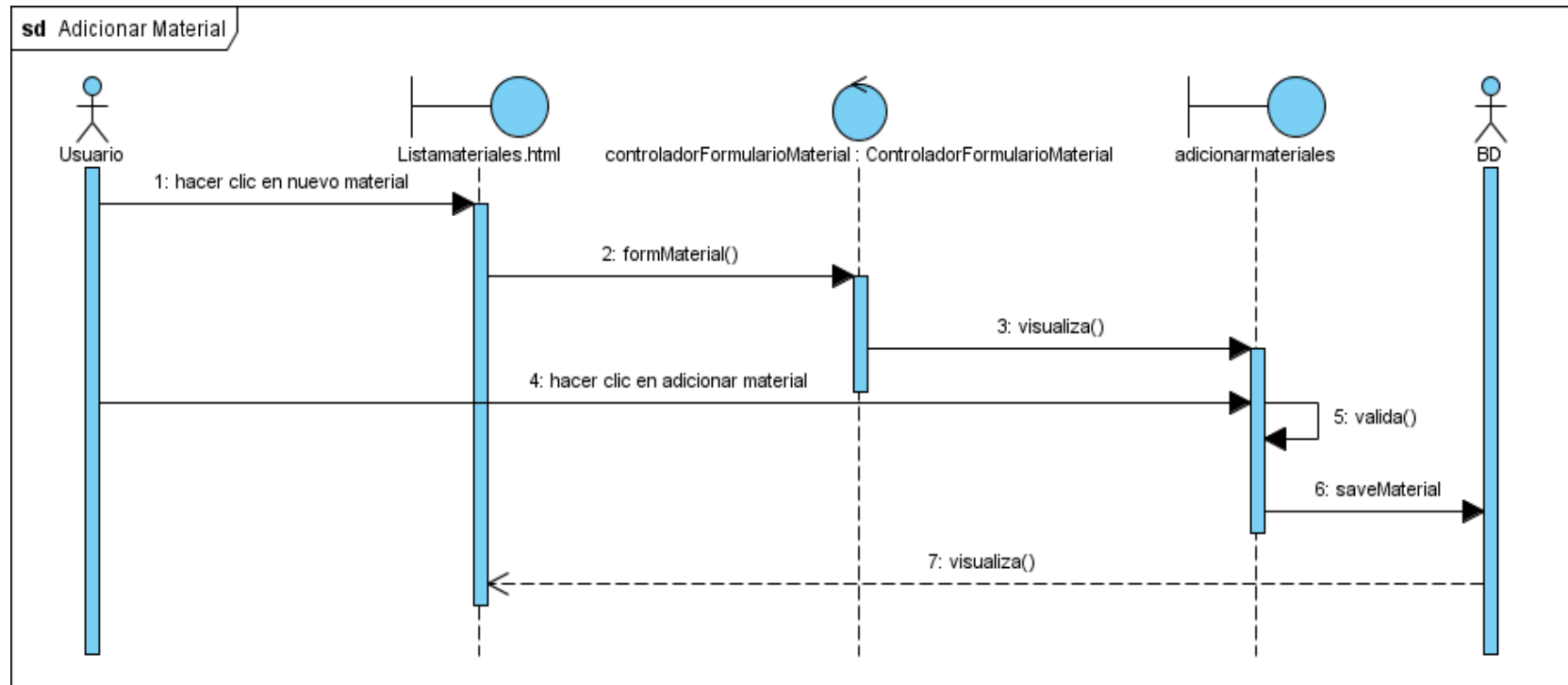


Fig.II.67 diagrama de secuencia-Adicionar Materiales

Diagrama de Secuencia: Modificar Materiales

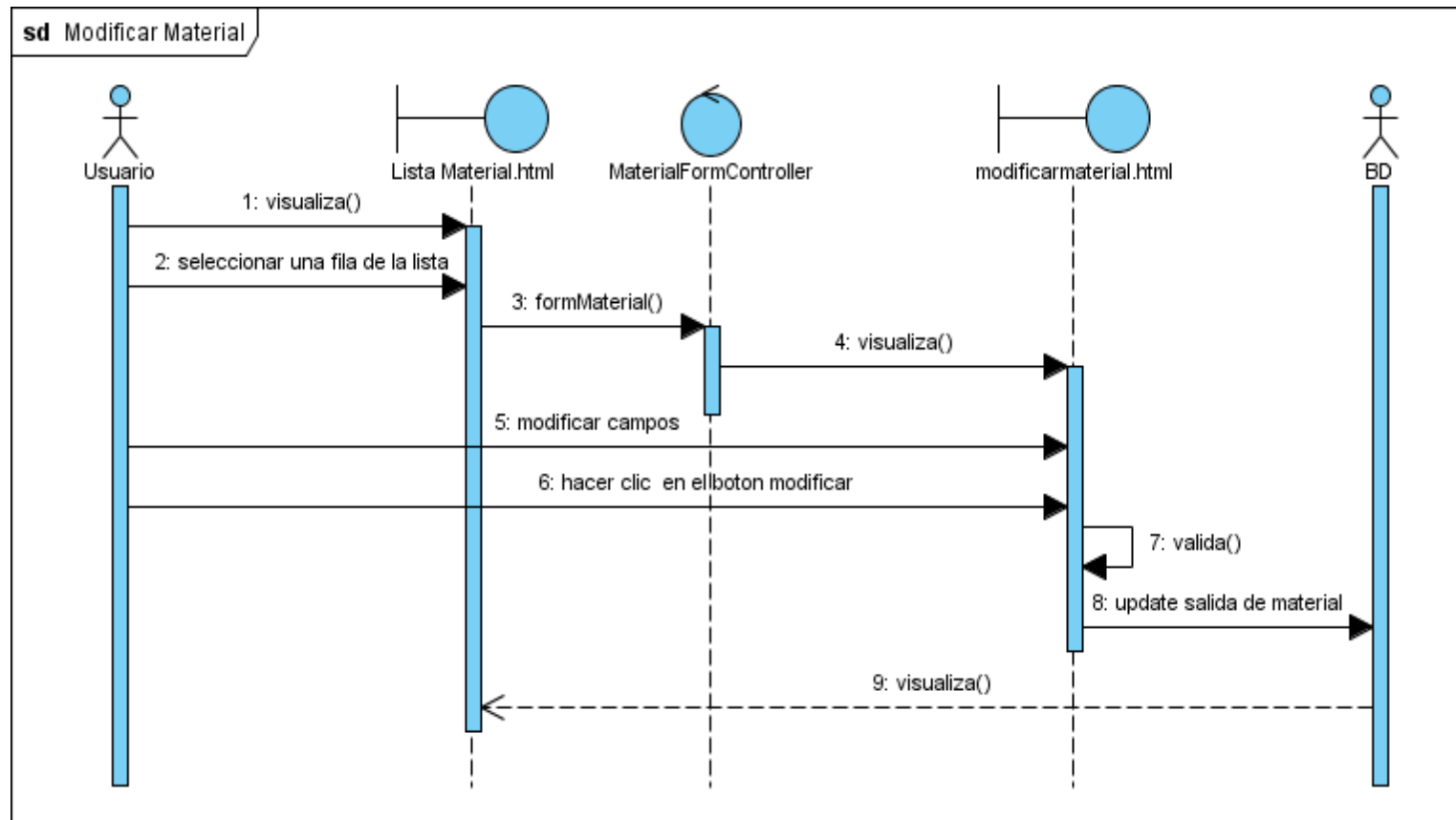


Fig.II.68 diagrama de secuencia-Modificar Materiales

Diagrama de Secuencia: Listar Categorías

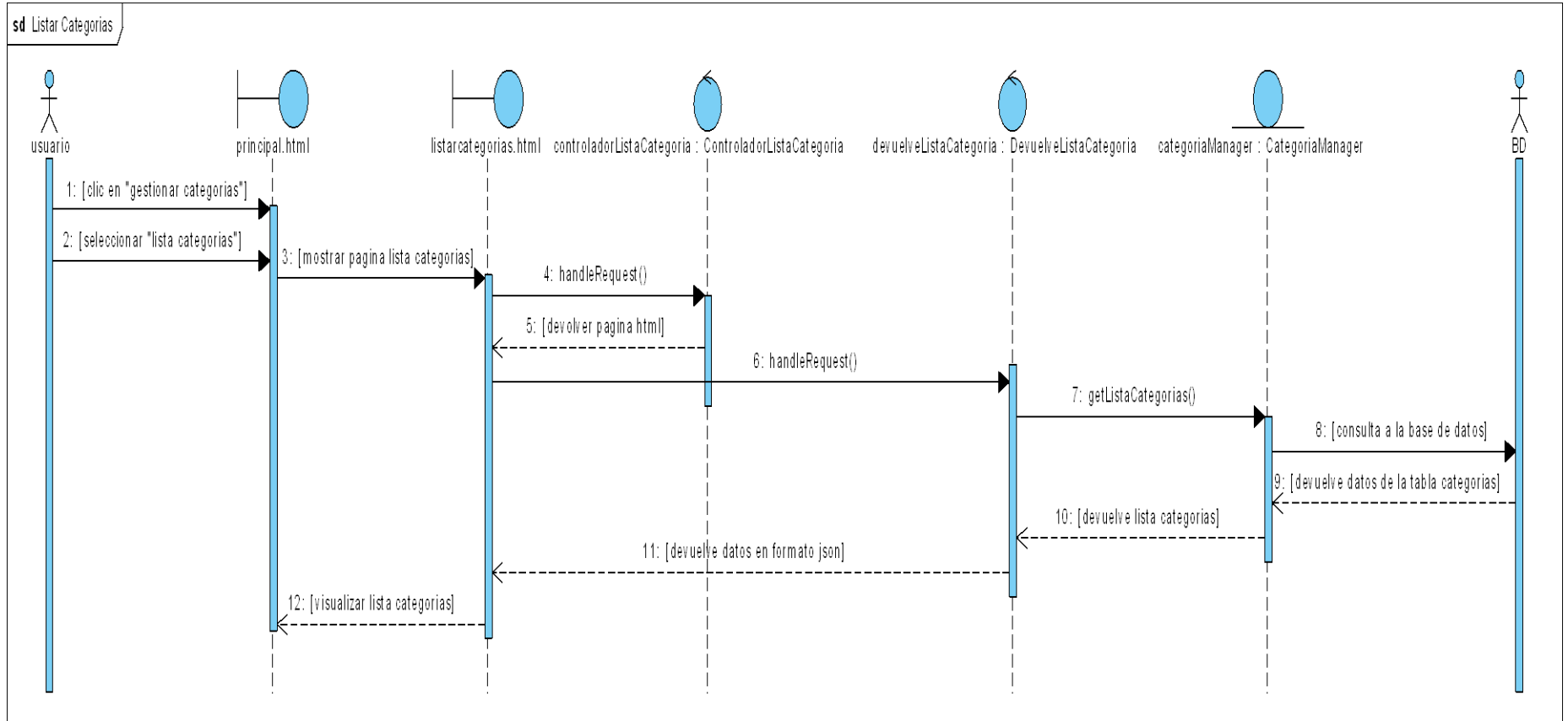


Fig.II.69 diagrama de secuencia-Listar Categoría

Diagrama de Secuencia: Adicionar Categoría

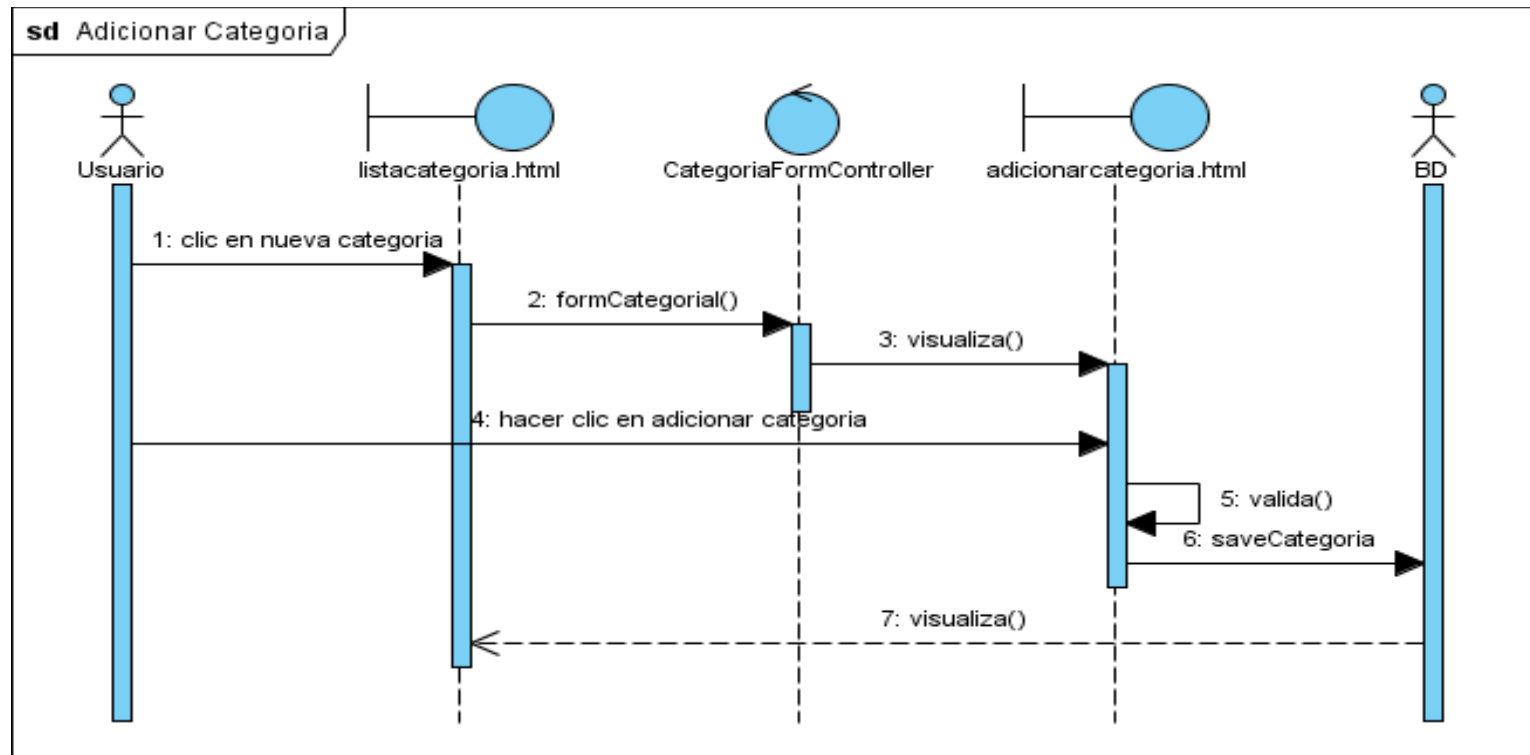


Fig.II.70 diagrama de secuencia-Adicionar Categoría

Diagrama de Secuencia: Modificar Categoría

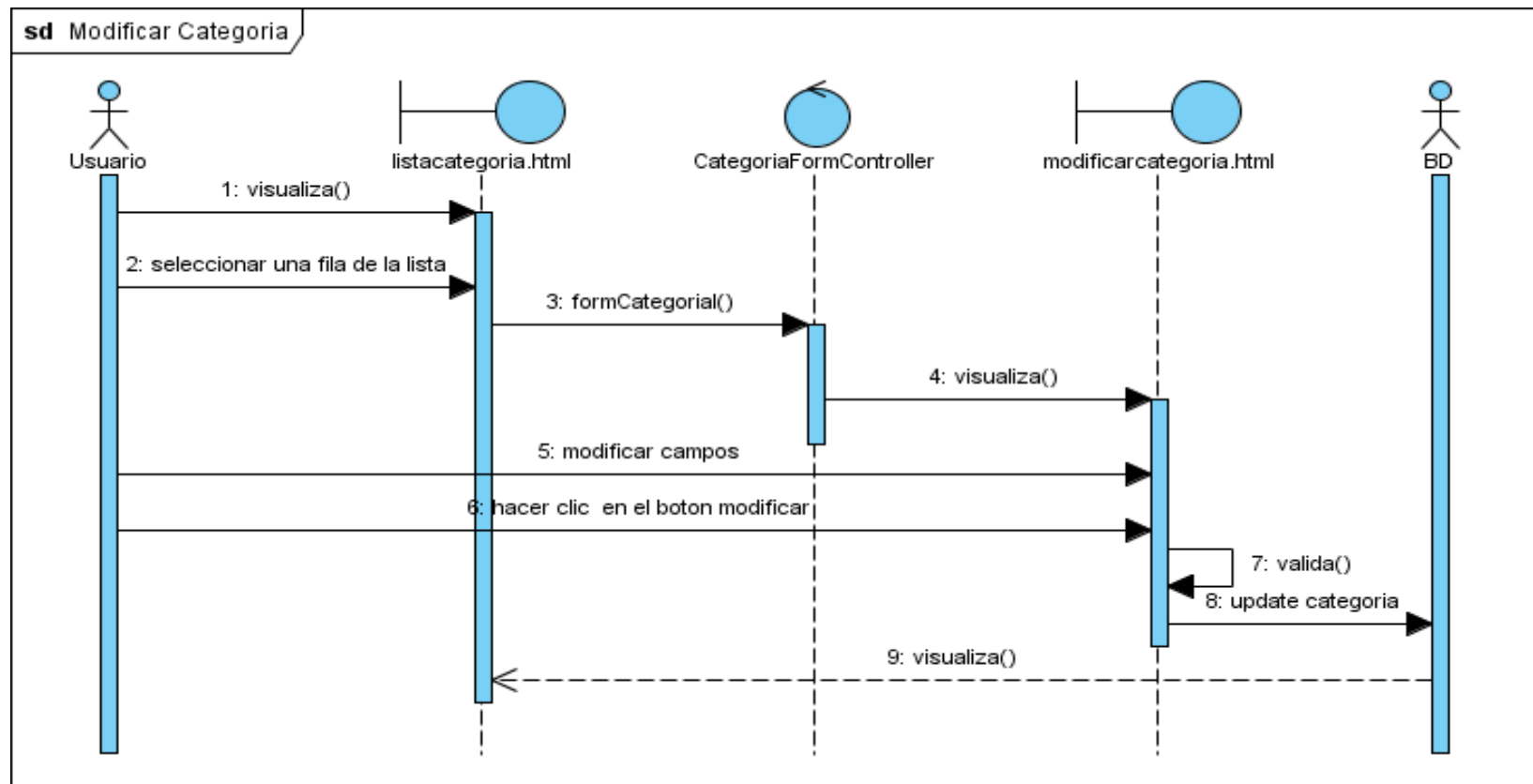


Fig.II.71 diagrama de secuencia-Modificar Categoría

Diagrama de Secuencia: Listar Solicitud

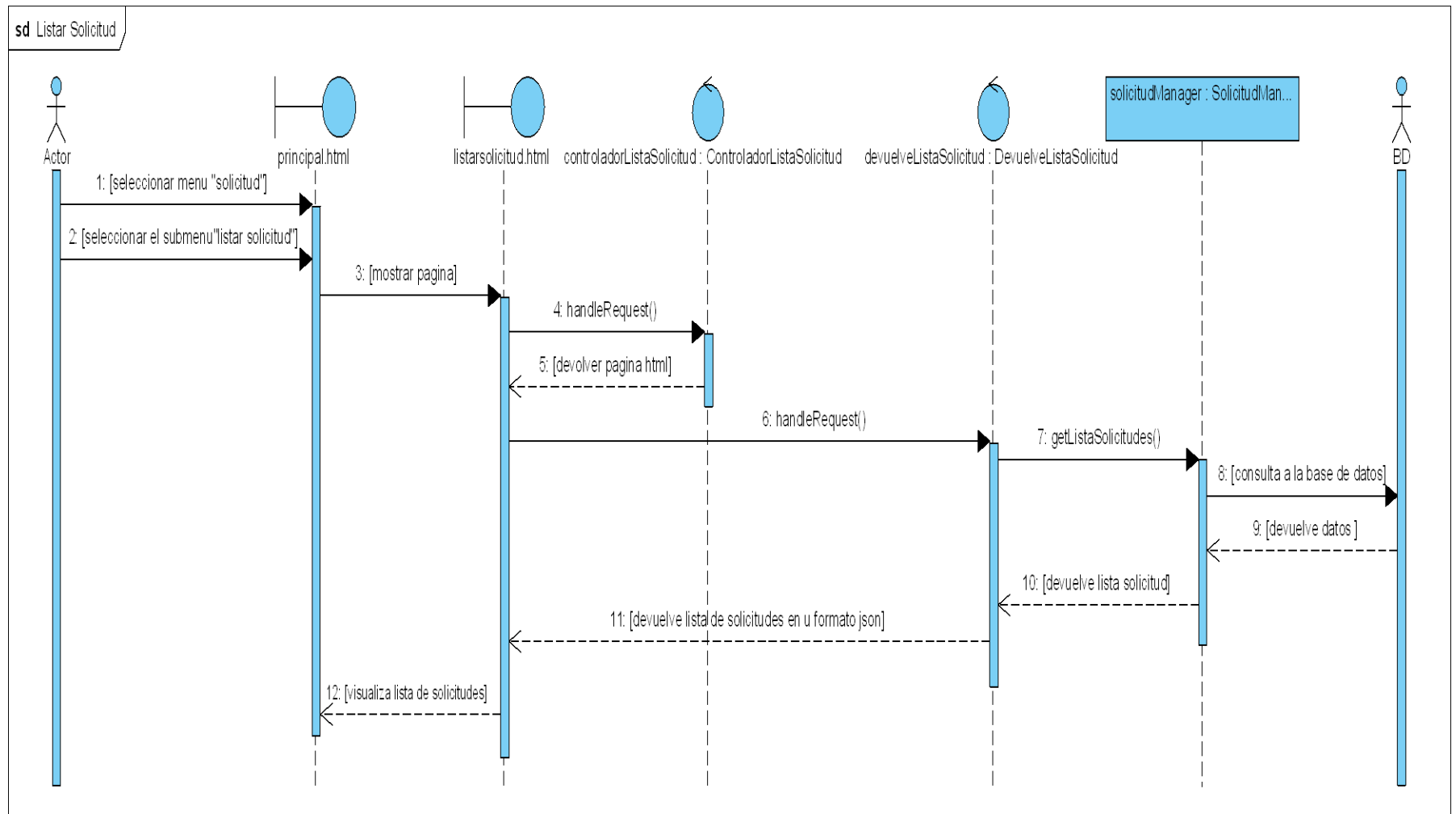


Fig.II.72 diagrama de secuencia-Listar Solicitud

Diagrama de Secuencia: Adicionar Solicitud

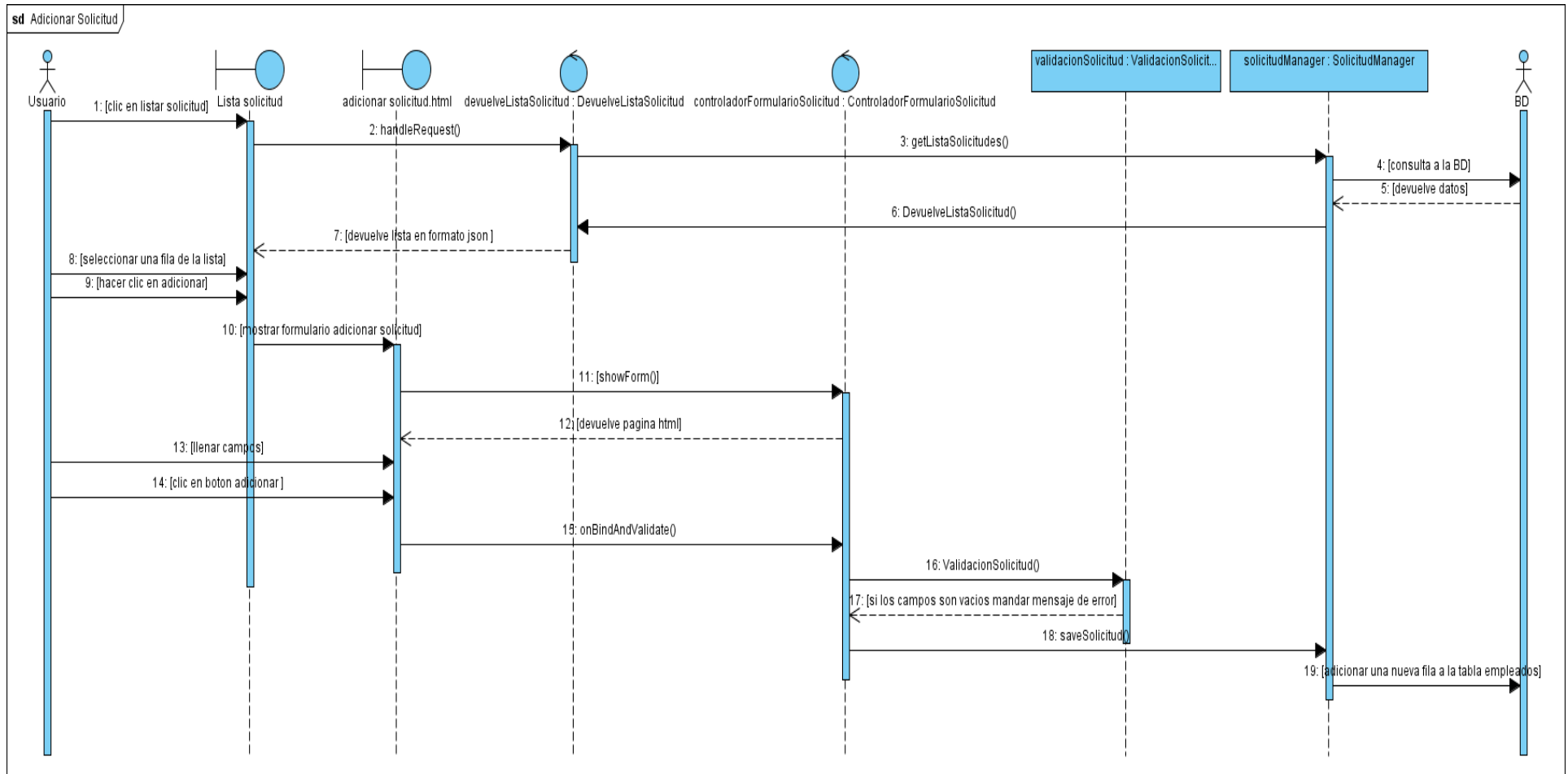


Fig. 73 diagrama de secuencia-Adicionar Solicitud

Diagrama de Secuencia: Modificar Solicitud

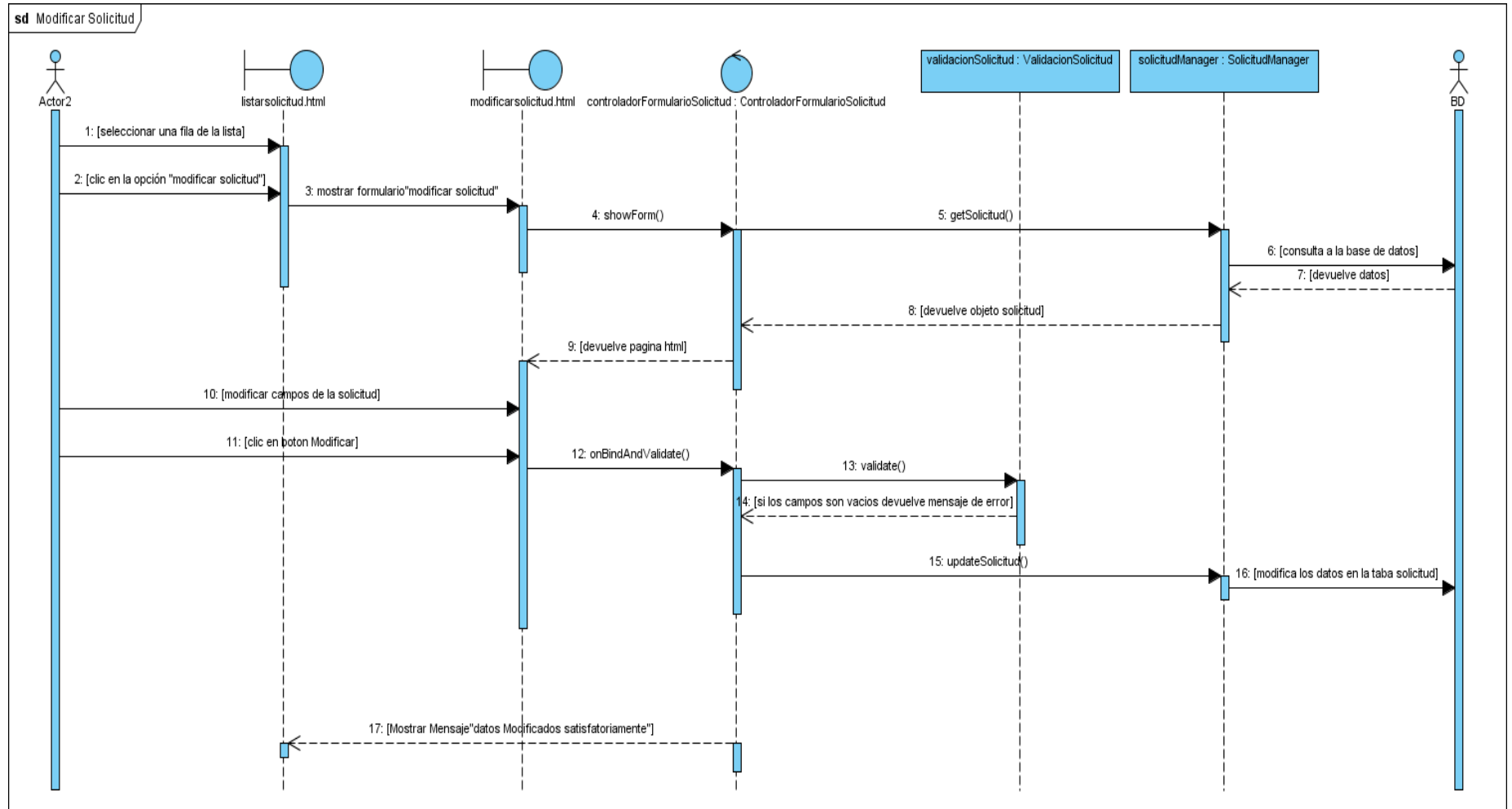


Fig.II.74 diagrama de secuencia-Modificar solicitud

Diagrama de Secuencia: Listar Cotización

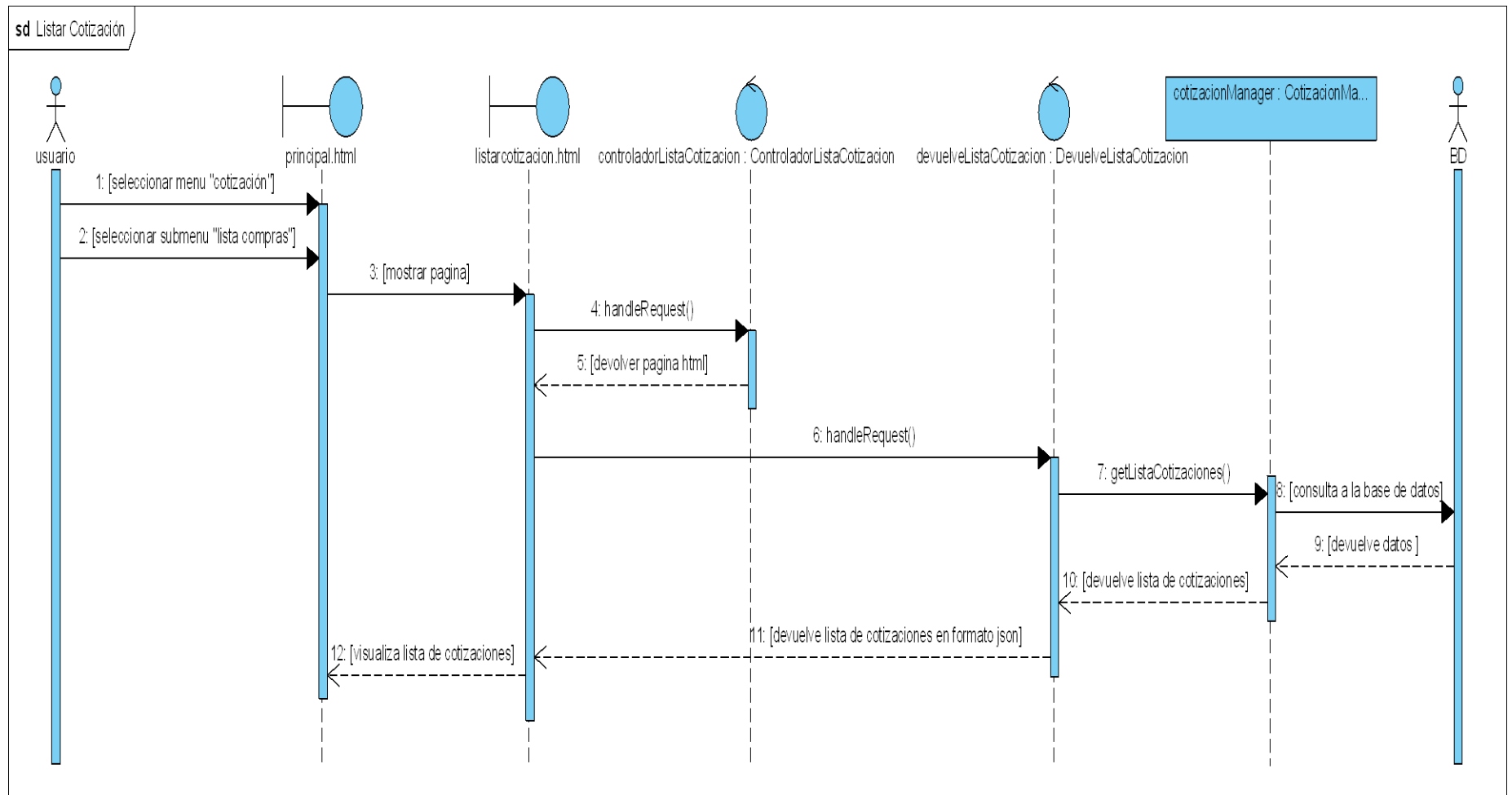


Fig.II.75 diagrama de secuencia-Listar Cotización

Diagrama de Secuencia: Lista Aprobación

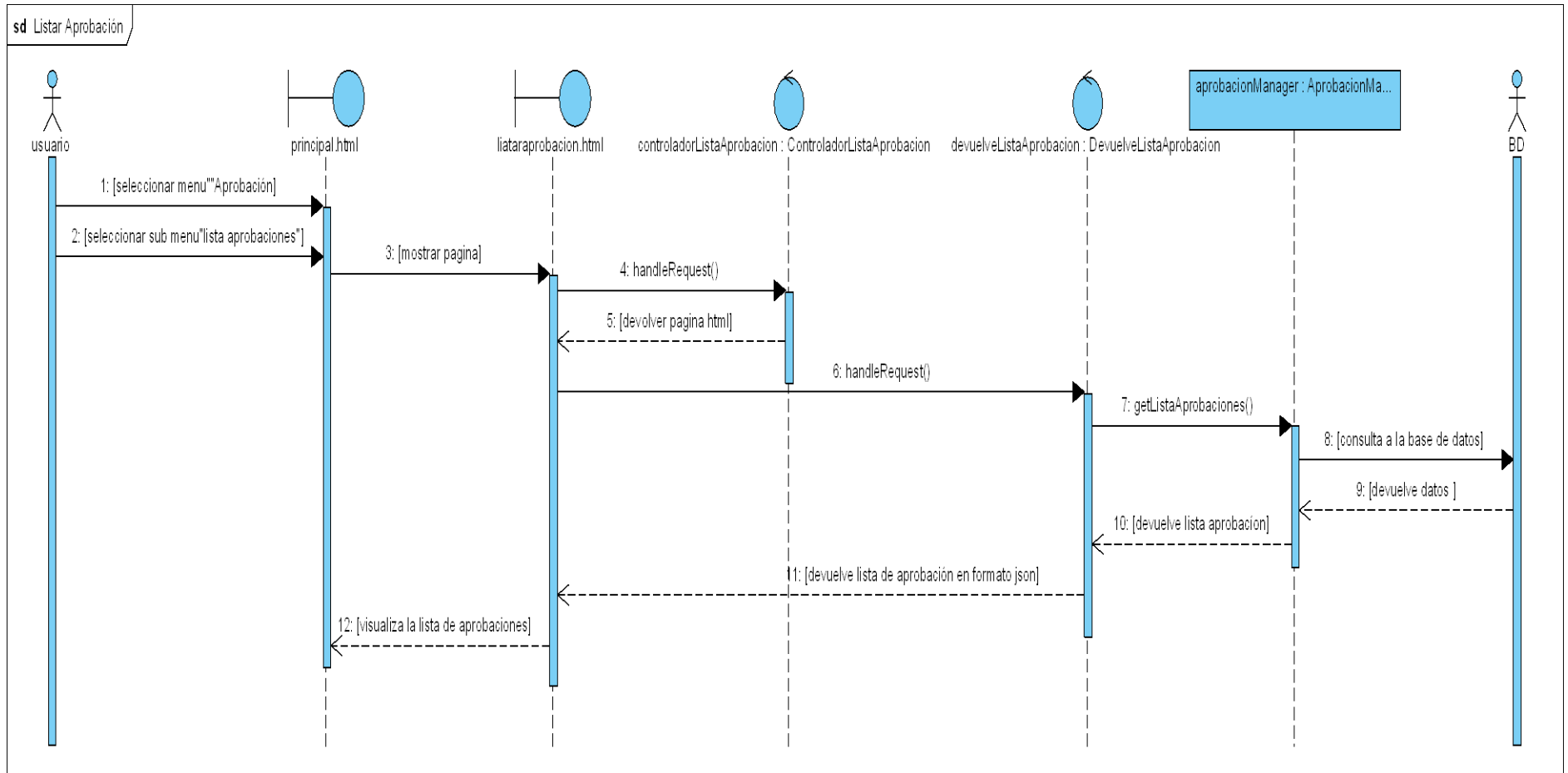


Fig.II.76 diagrama de secuencia-Lista Aprobación

Diagrama de Secuencia: Modificar Aprobación

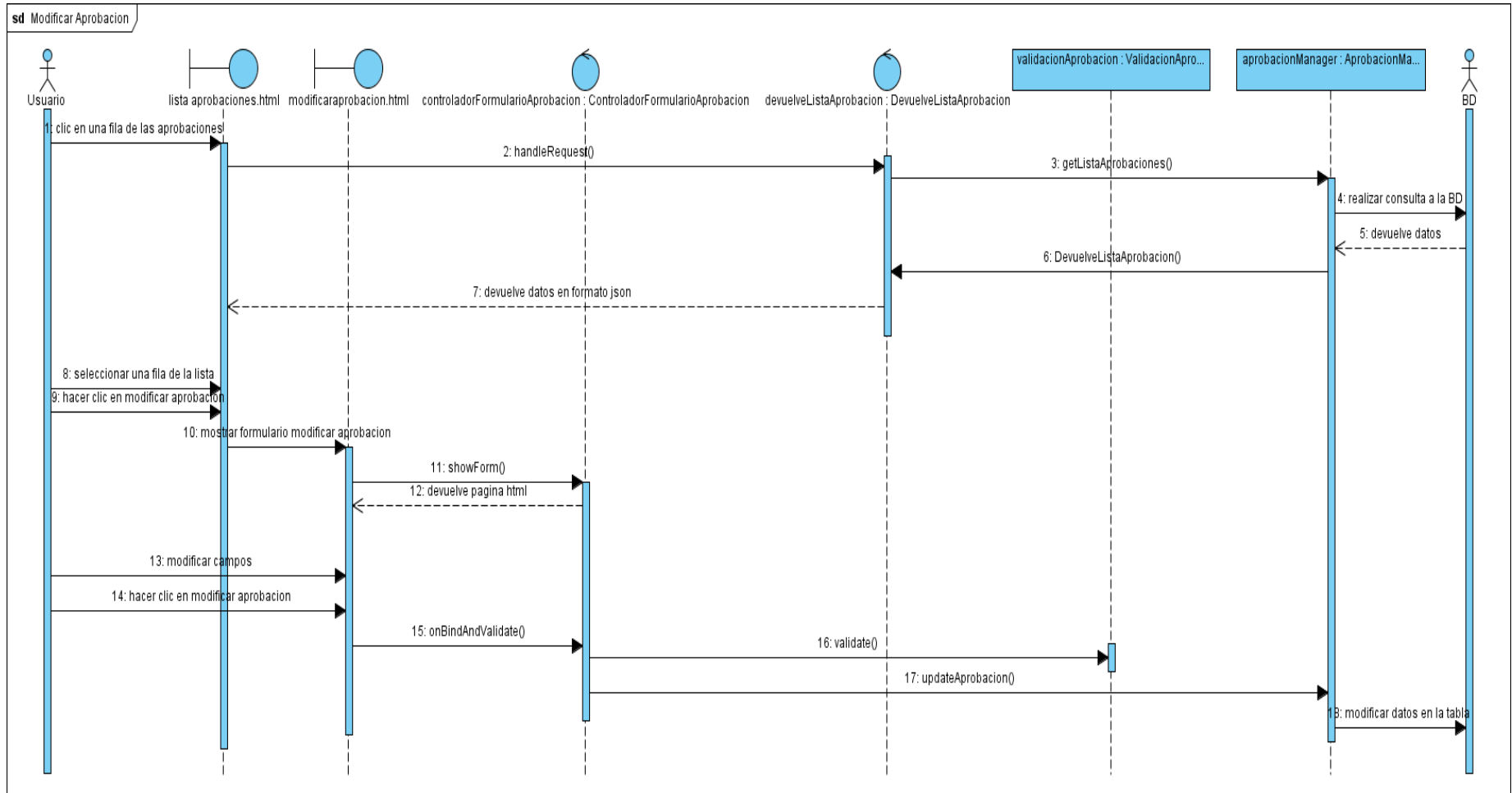


Fig.II.77 diagrama de secuencia-Modificar Aprobación

Diagrama Secuencia: Listar Compra

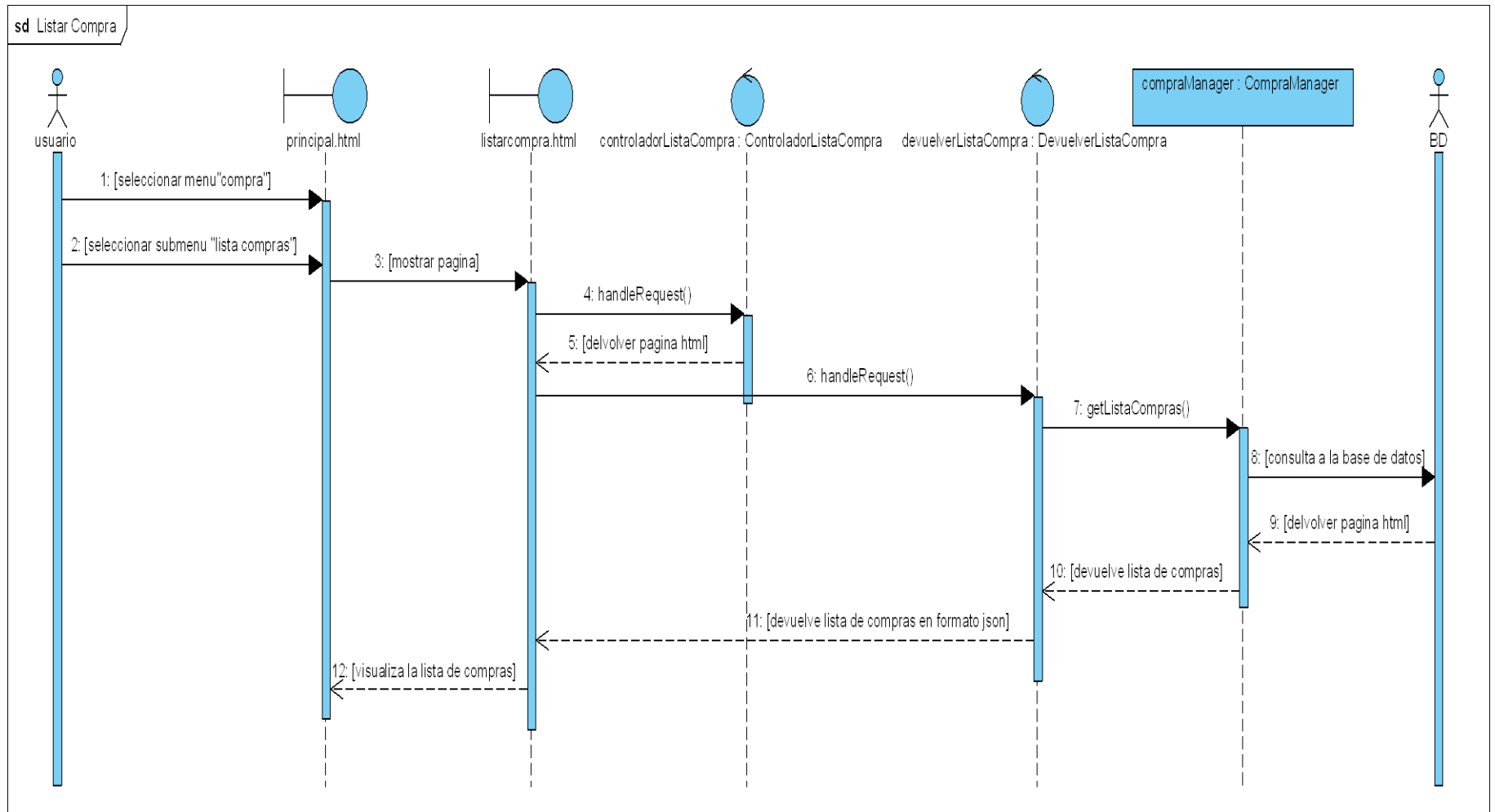


Fig.II.78 diagrama de secuencia-Lista Compra

Diagrama Secuencia: Modificar Compra

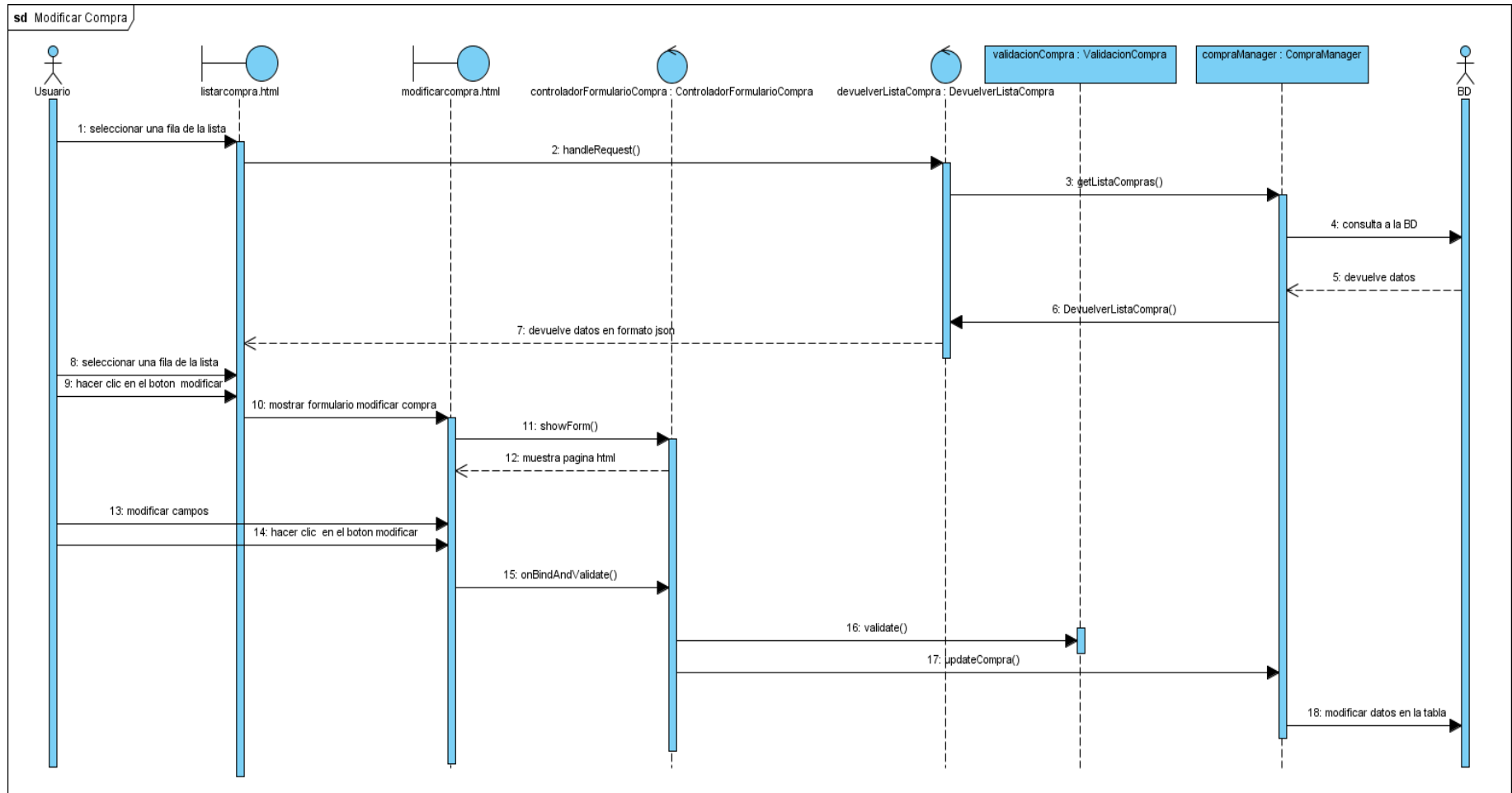


Fig.II.79 diagrama de secuencia-Modificar Compra

Diagrama Secuencia: Listar entradas de Material

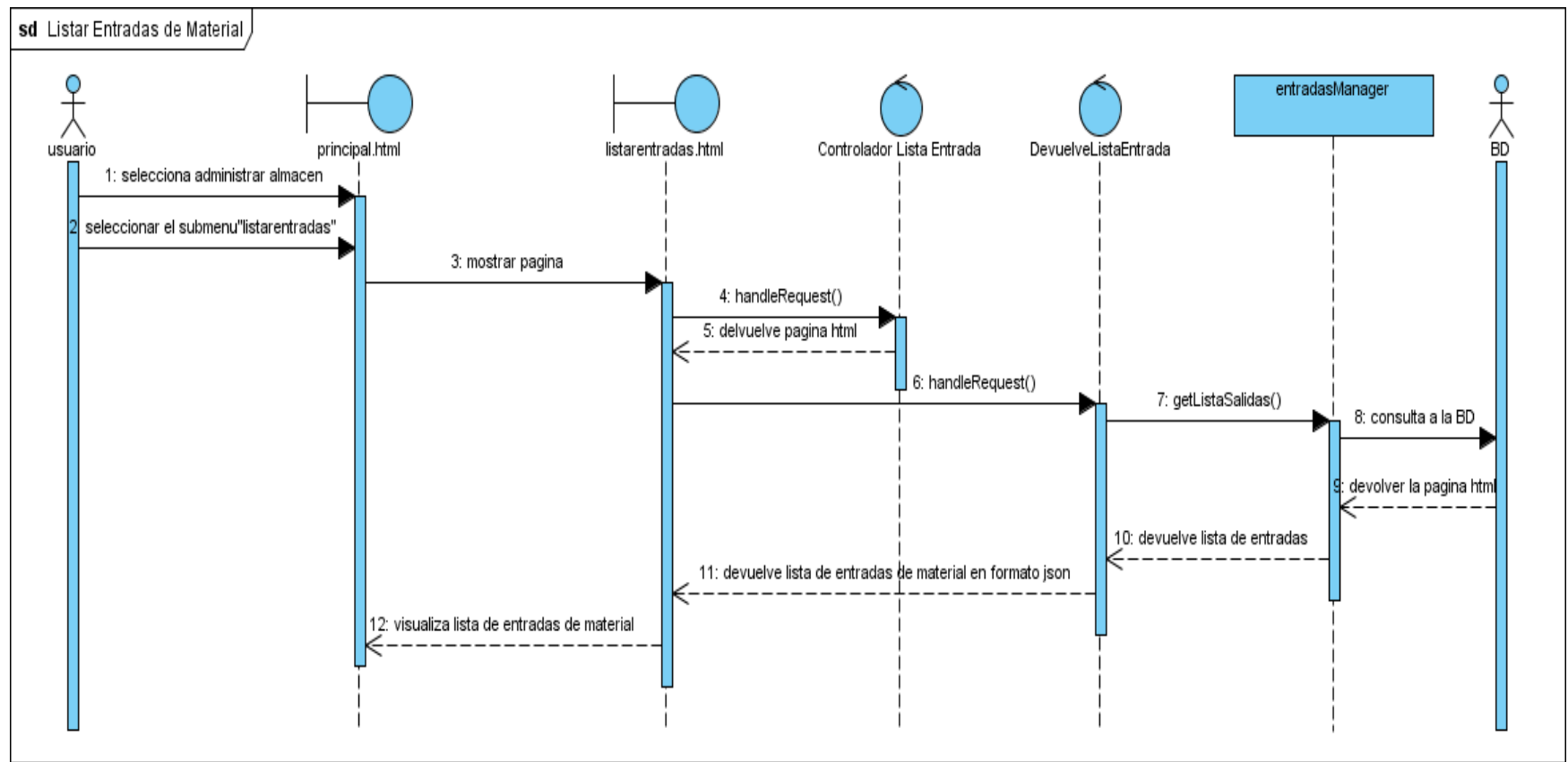


Fig.II.80 diagrama de secuencia-Listar Entradas de Material

Diagrama de Secuencia: Adicionar Entradas de Material

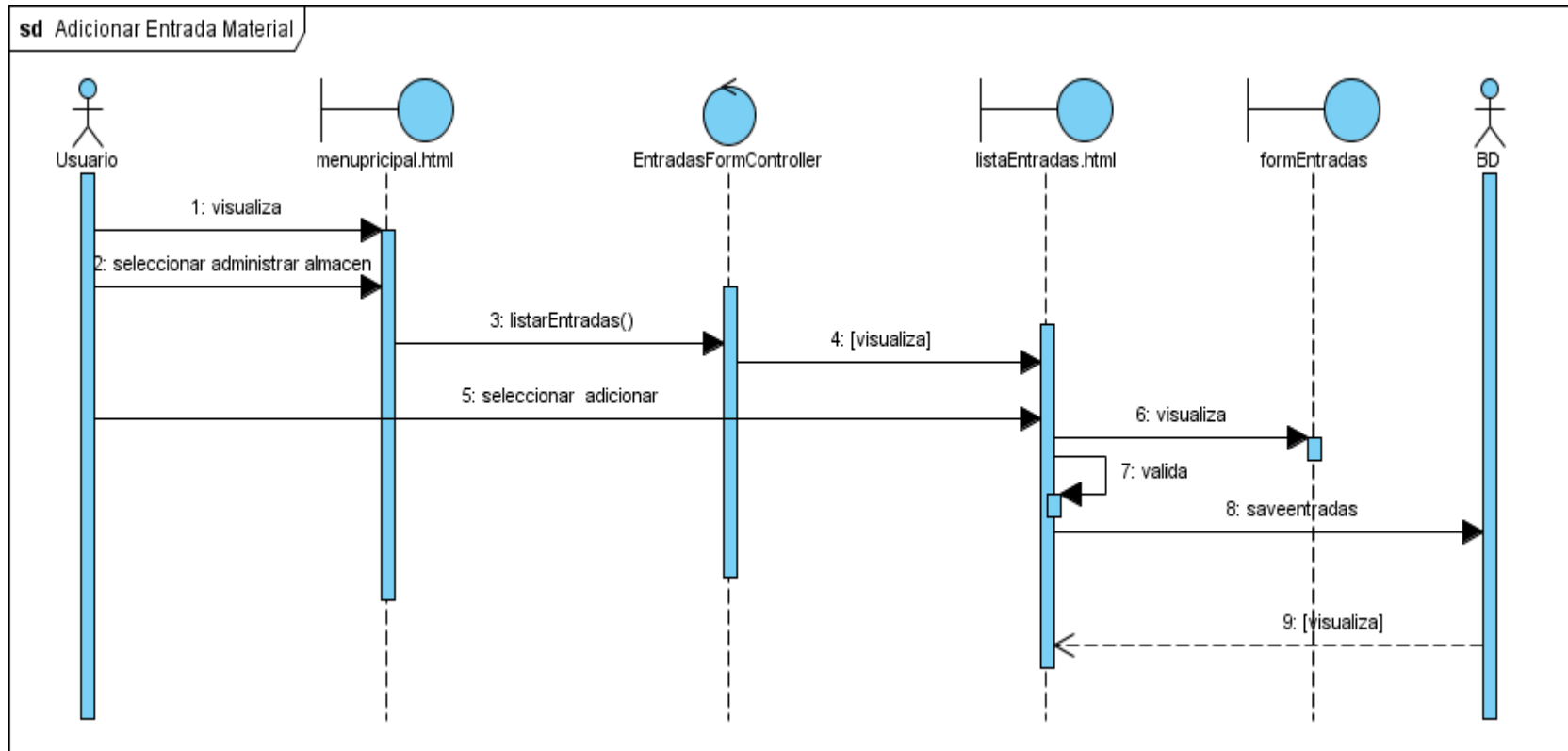


Fig.II.81 diagrama de secuencia-Adicionar Entradas de Material

Diagrama de Secuencia: Modificar Entradas de Material

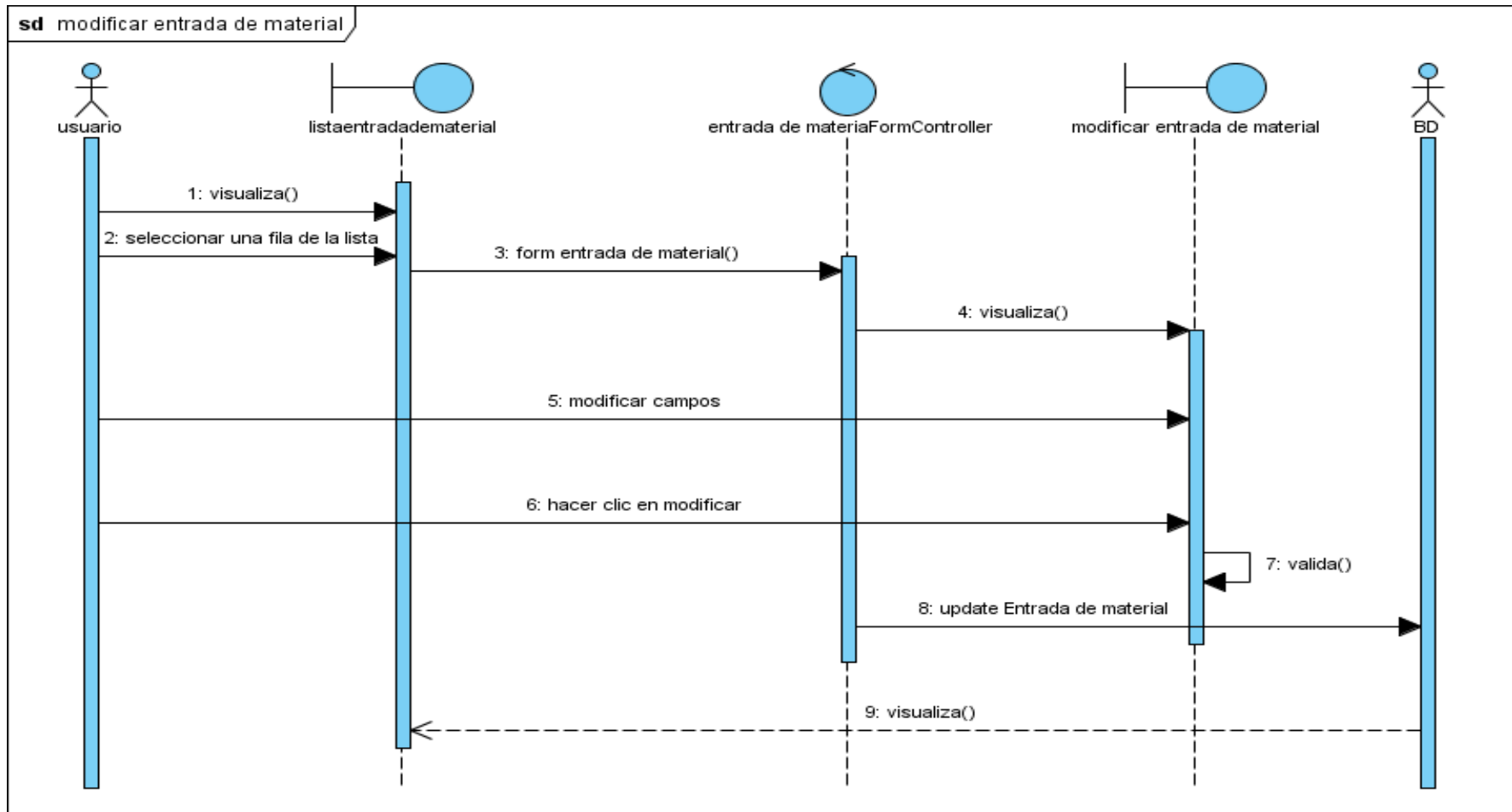


Fig.II.82 diagrama de secuencia-Modificar Entradas de Material

Diagrama de secuencia: Listar Salida de Material

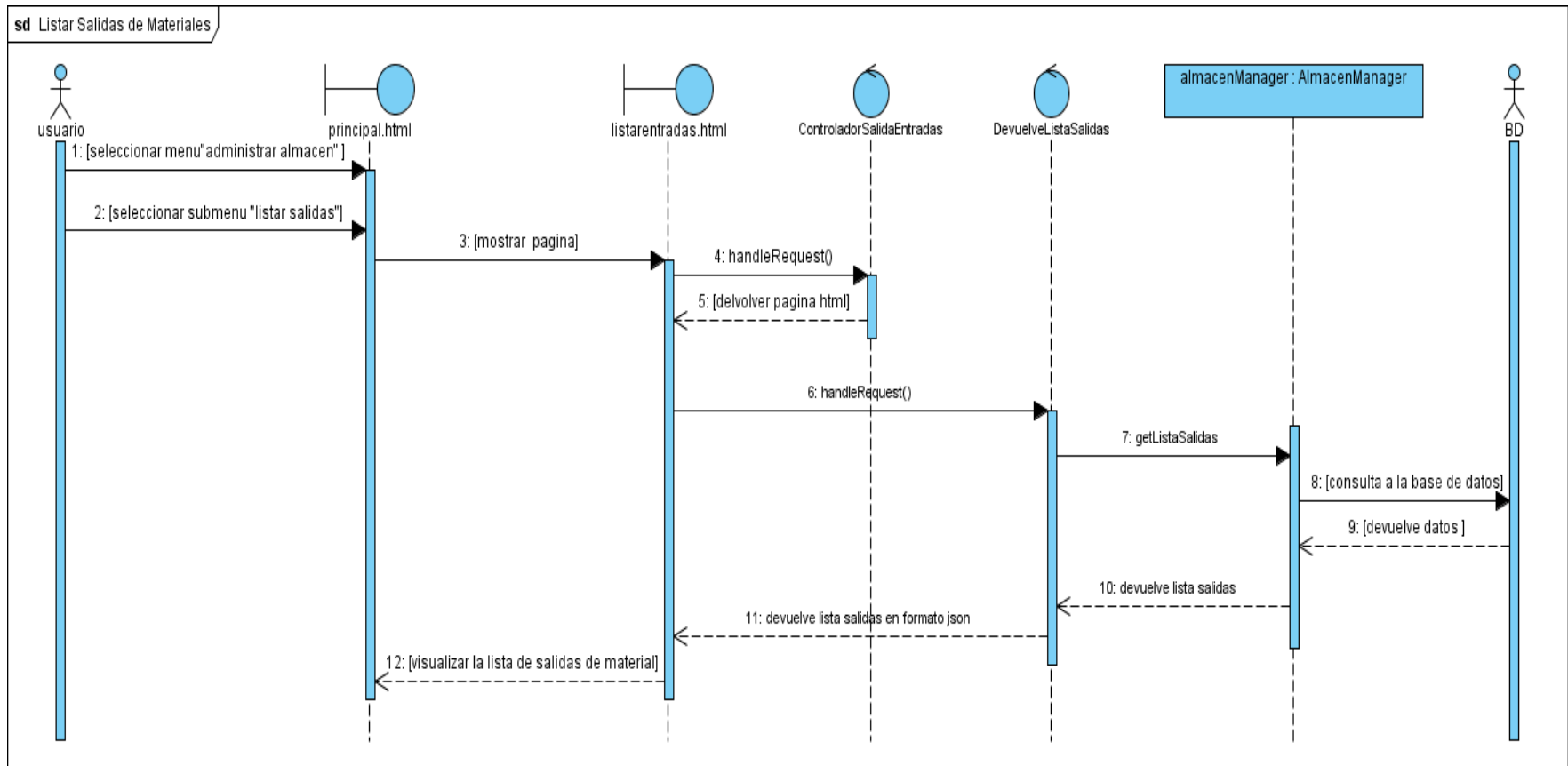


Fig.II.83 diagrama de secuencia-Listar Salida de Material

Diagrama de Secuencia: Adicionar Salidas de Material

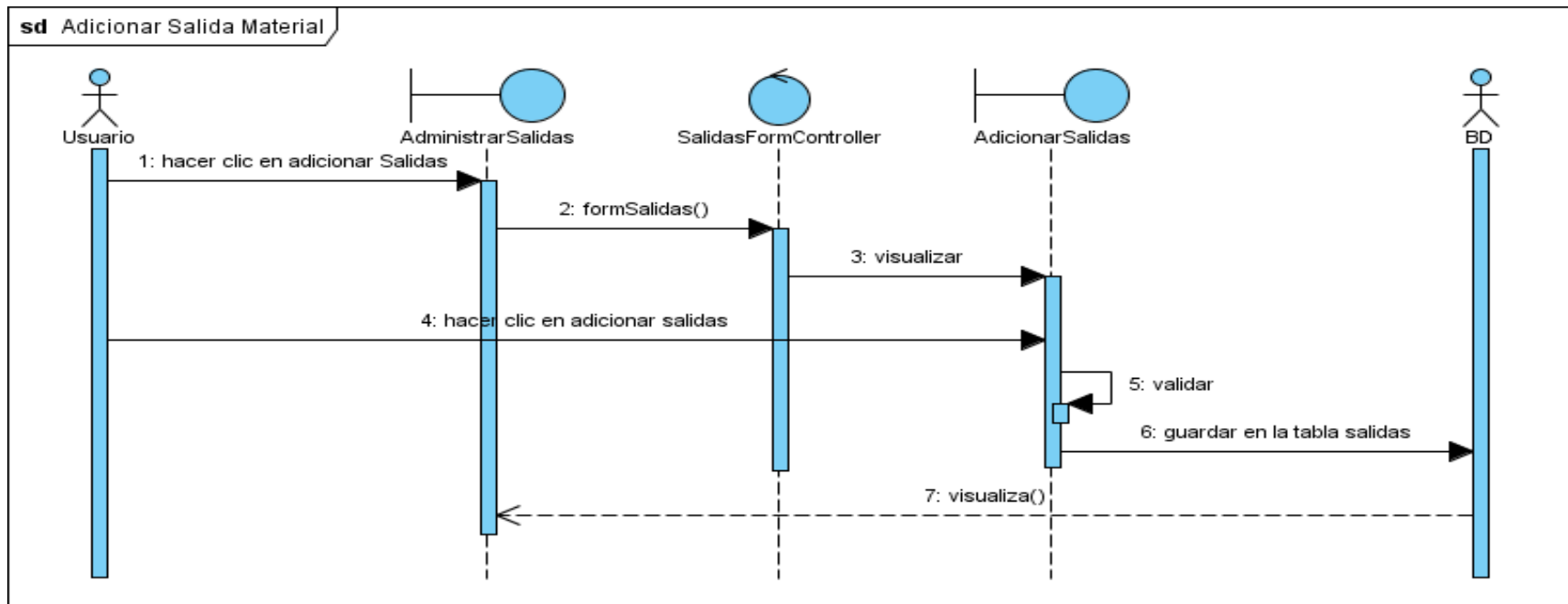


Fig.II.84 diagrama de secuencia-Adicionar Salidas de material

Diagrama de Secuencia: Modificar Salidas de Material

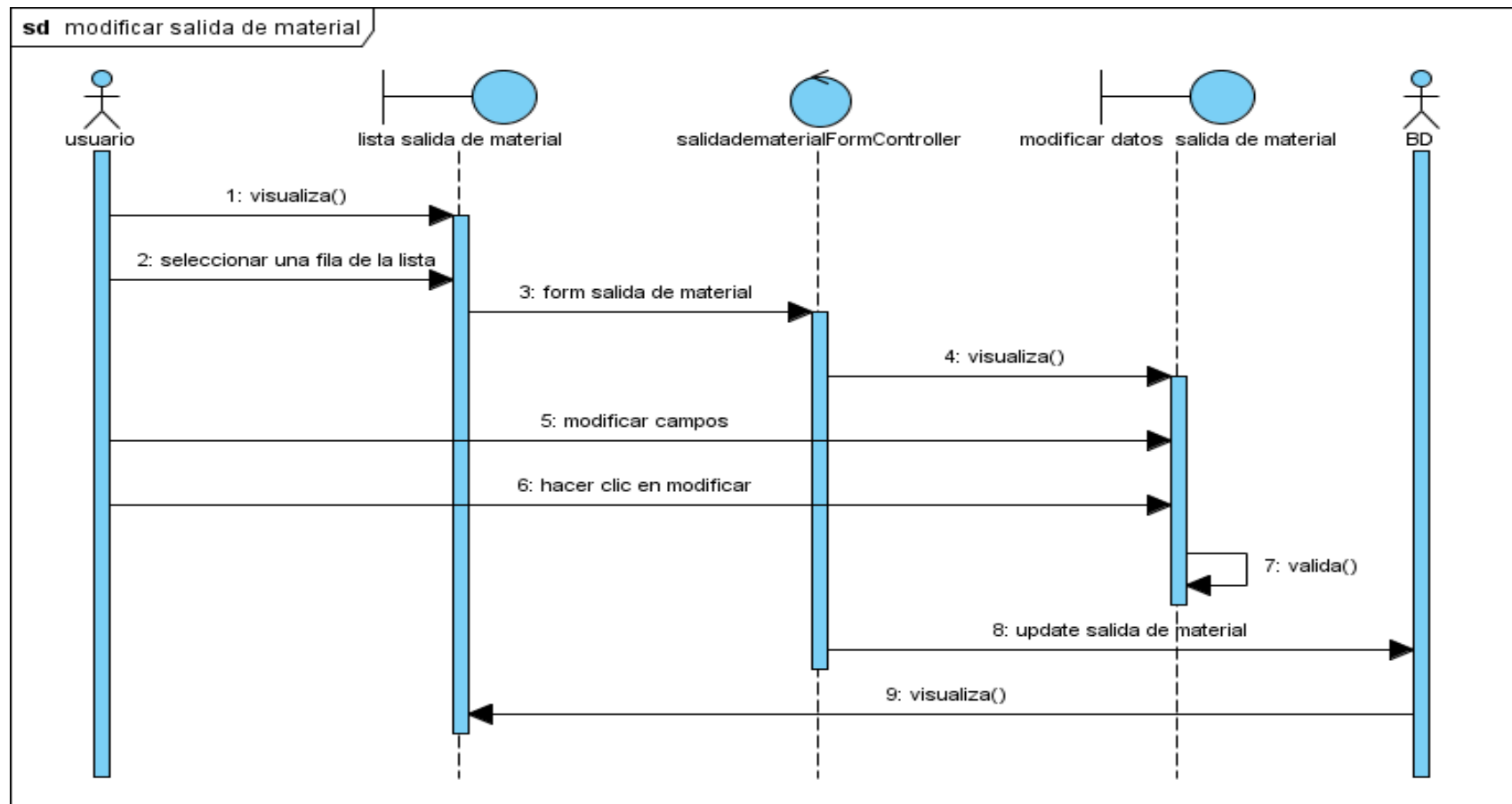


Fig.II.85 diagrama de secuencia-Modificar Salidas de material

Diagrama de Secuencia: Reportes

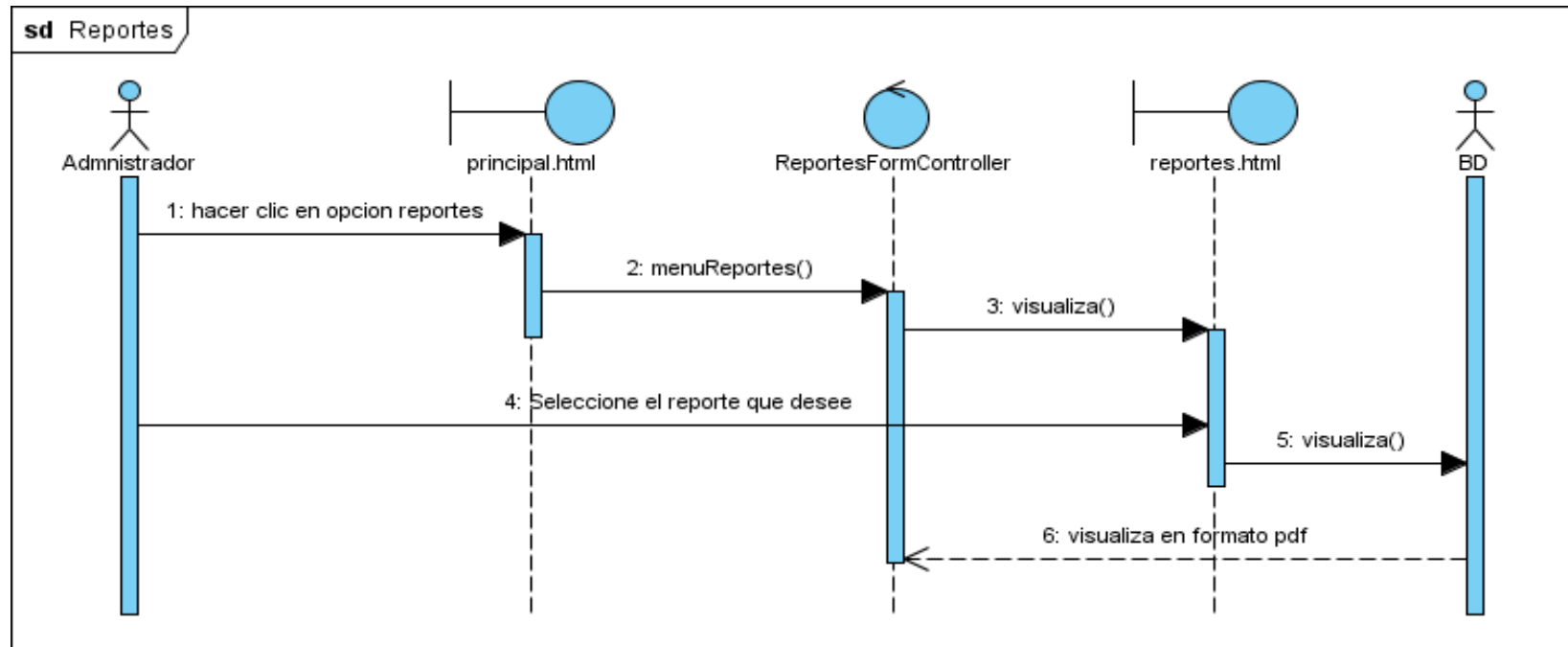


Fig.II.86 diagrama de secuencia-Reportes

I.2.8. MODELOS DE DIAGRAMAS DE COLABORACION

Introducción

Mediante el uso de los diagramas de colaboración podemos mejorar el flujo de control entre actividades del sistema. La idea es generar una especie de diagrama Pert, en el que puede ver el flujo de actividades que tiene lugar a lo largo del tiempo, así como las tareas concurrentes que pueden realizarse a la vez. Gráficamente es un conjunto de arcos y nodos. Desde un punto de vista conceptual, el diagrama de colaboración muestra como fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecuencia de un proceso más complejo.

Propósito

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

Alcance

- Describe los procesos del sistema.
- Identificar y definir los **procesos de los casos de uso** según los **objetivos** de la Organización.

II.2.9. DIAGRAMAS DE COLABORACION

Diagrama de colaboración: Ingresar al Sistema

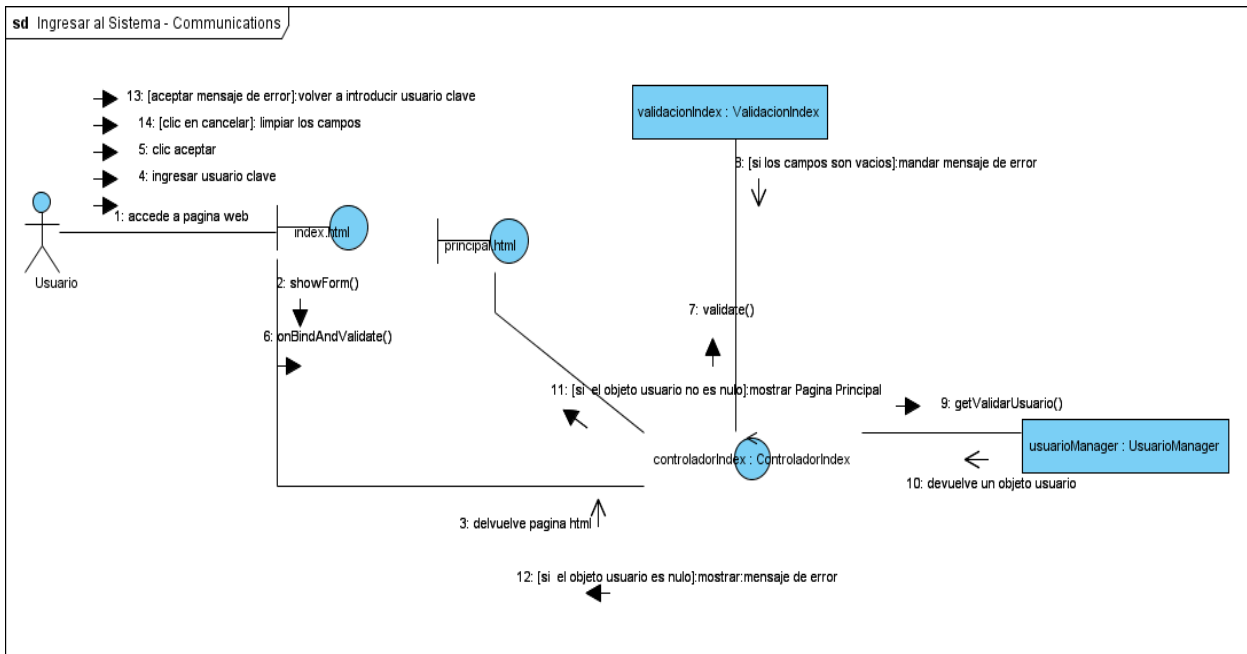


Fig.II.87 diagrama de Colaboración –Ingresar al Sistema

Diagrama de colaboración: Listar Usuario

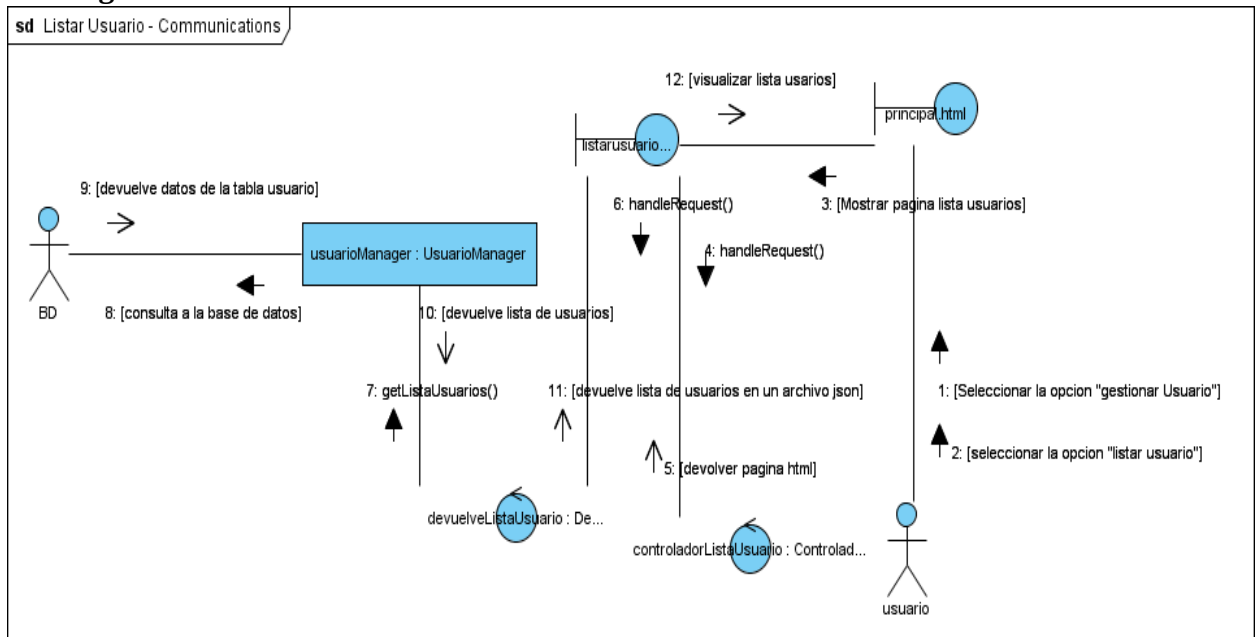


Fig.II.88 diagrama de Colaboración –Ingresar al Sistema

Diagrama de colaboración: Adicionar Usuario

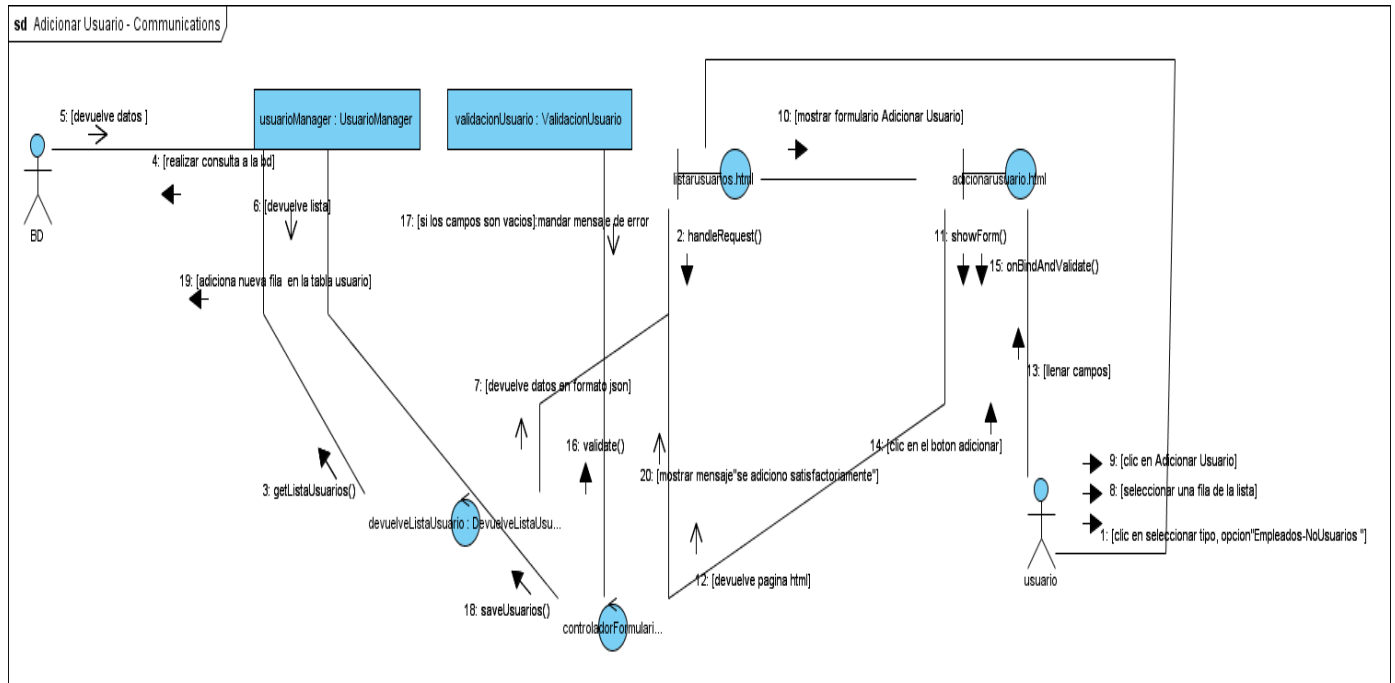


Fig.II.89 diagrama de Colaboración –Adicionar Usuario

Diagrama de colaboración: Modificar Usuario

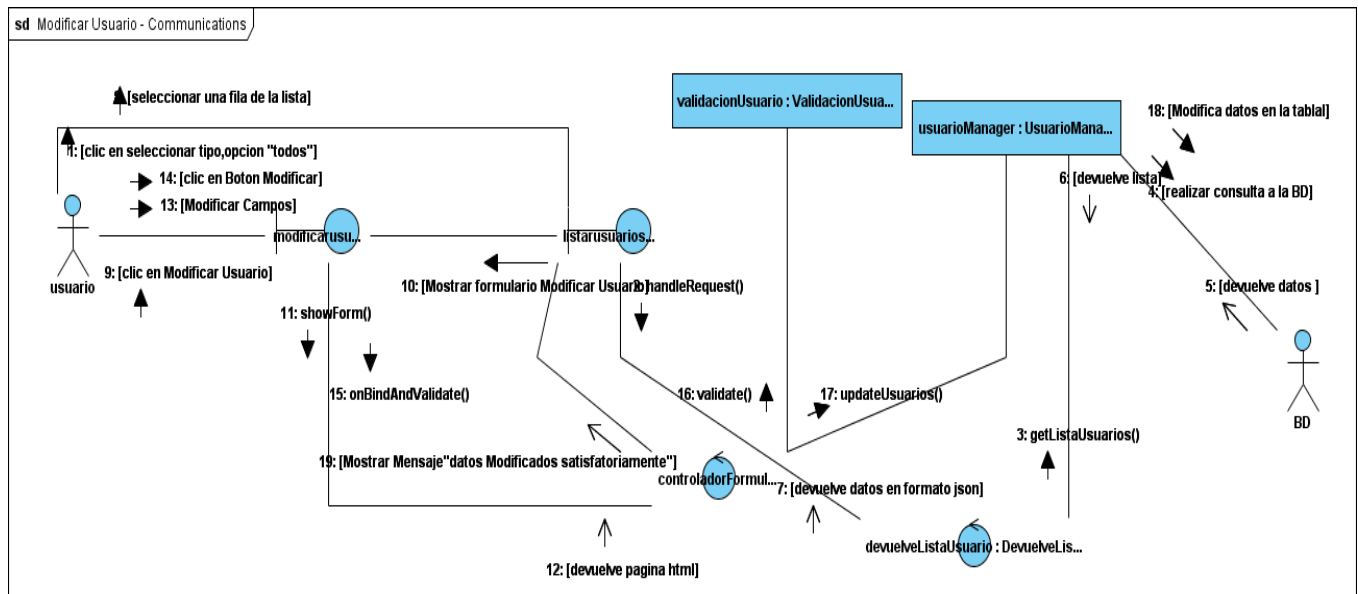


Fig.II.90 diagrama de Colaboración –Modificar Usuario

Diagrama de colaboración: Listar Persona

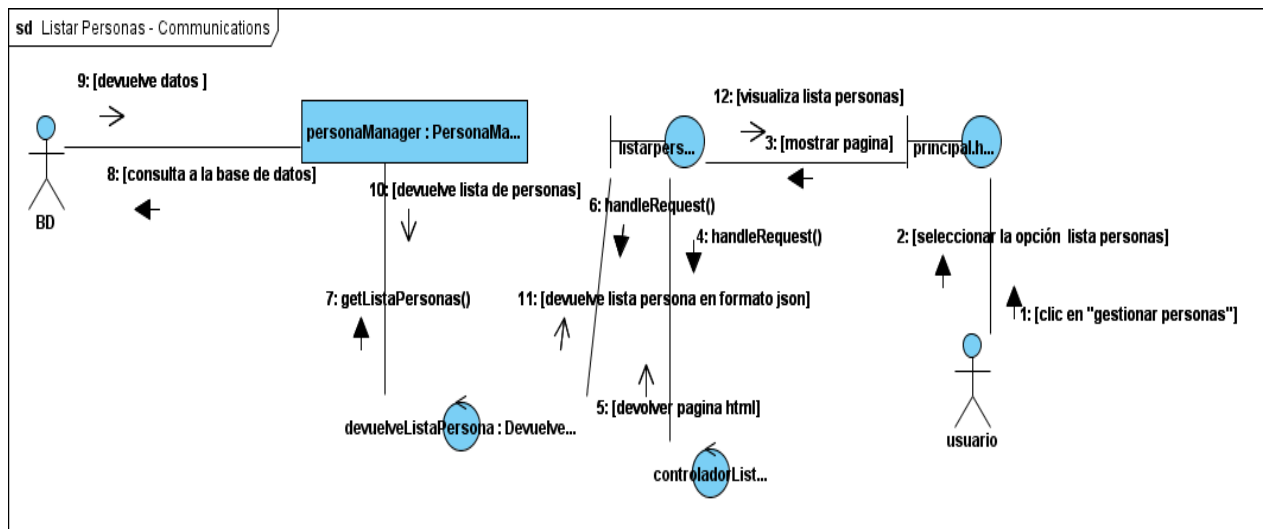


Fig.II.91 diagrama de Colaboración –Listar Persona

Diagrama de Colaboración: Listar Empleados

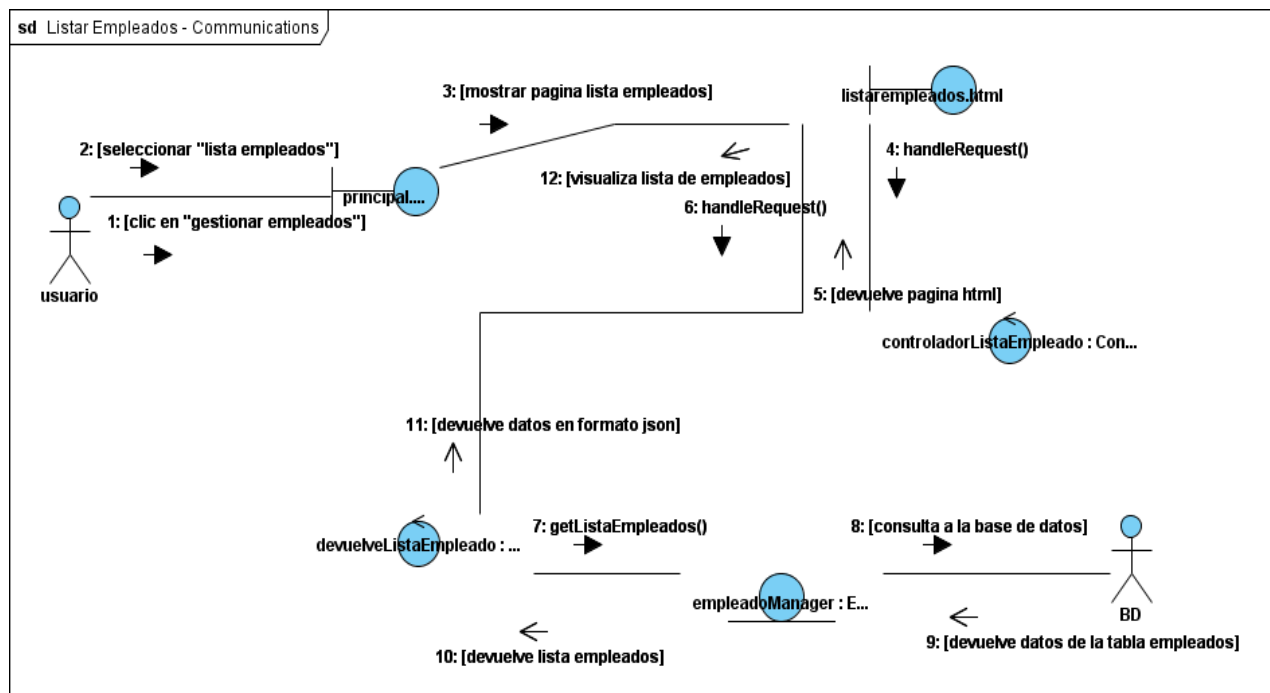


Fig.II.92 diagrama de Colaboración –Listar Empleados

Diagrama de Colaboración: Adicionar Empleado

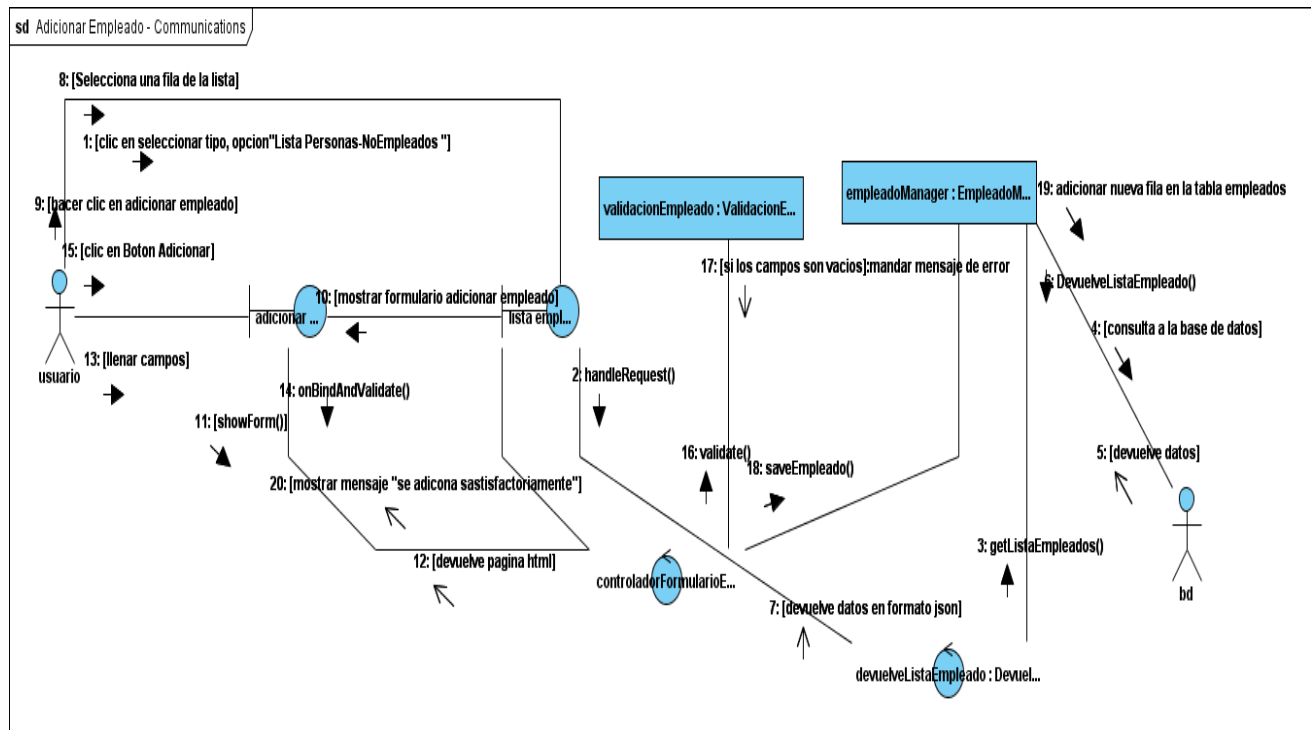


Fig.II.93 diagrama de Colaboración –Adicionar empleado

Diagrama de Colaboración: Modificar Empleado

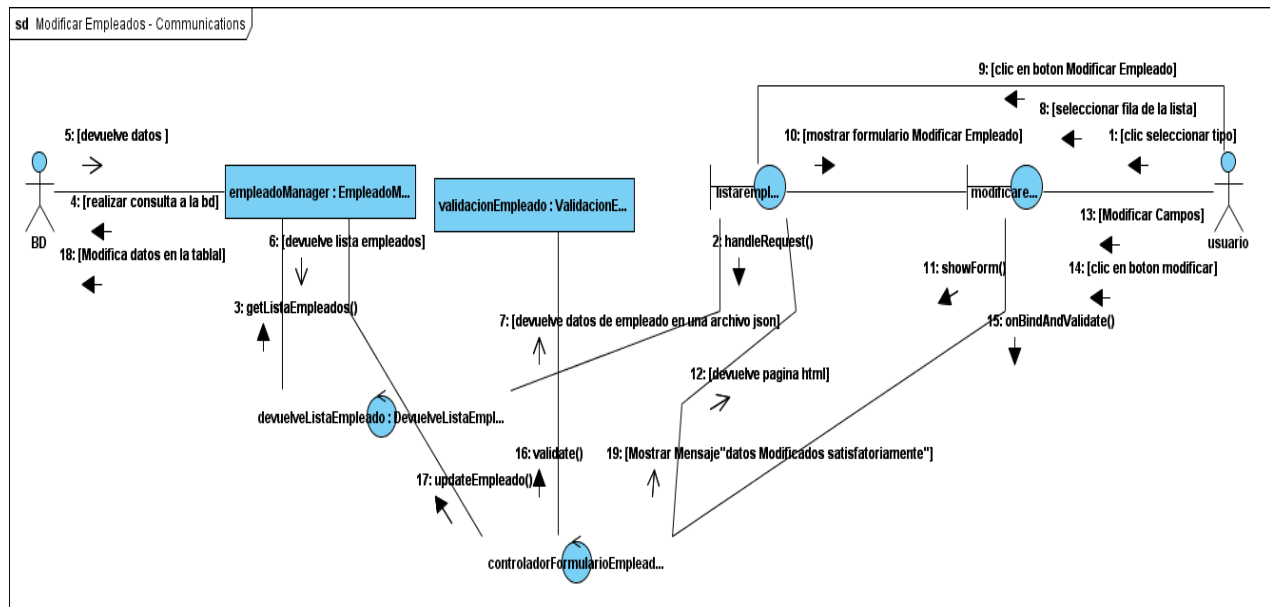


Fig.II.94 diagrama de Colaboración –Modificar Empleado

Diagrama de Colaboración: Lista Proveedores

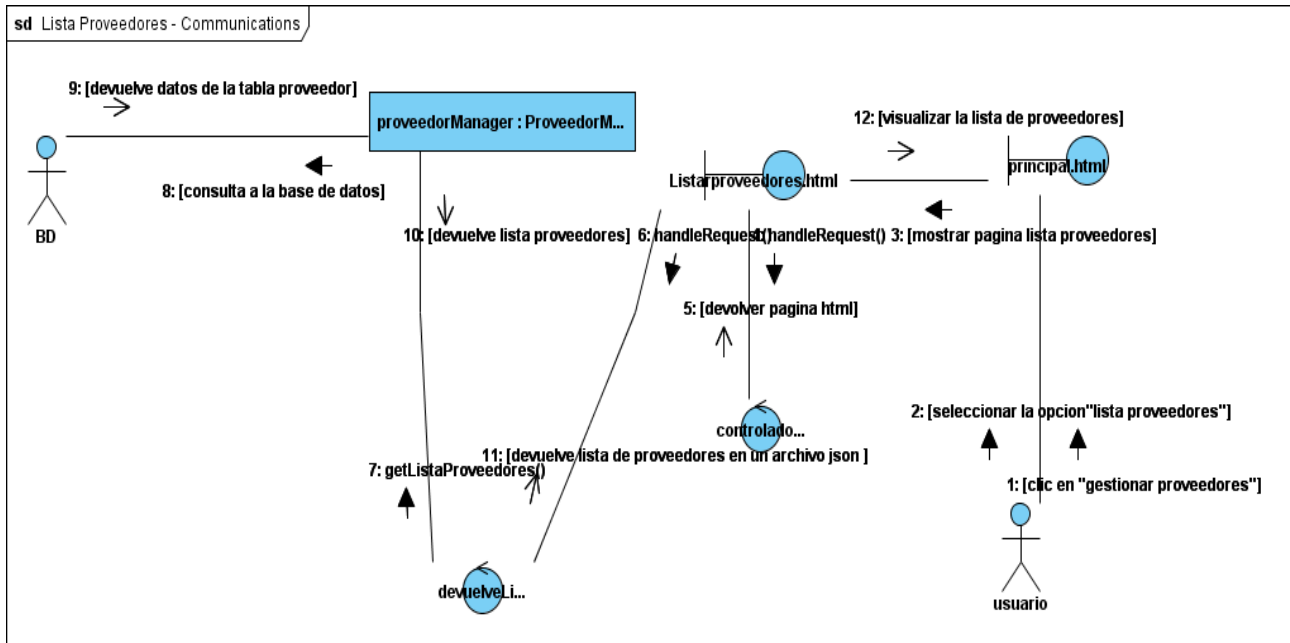


Fig.II.95 diagrama de Colaboración –Lista Proveedores

Diagrama de Colaboración: Listar Materiales

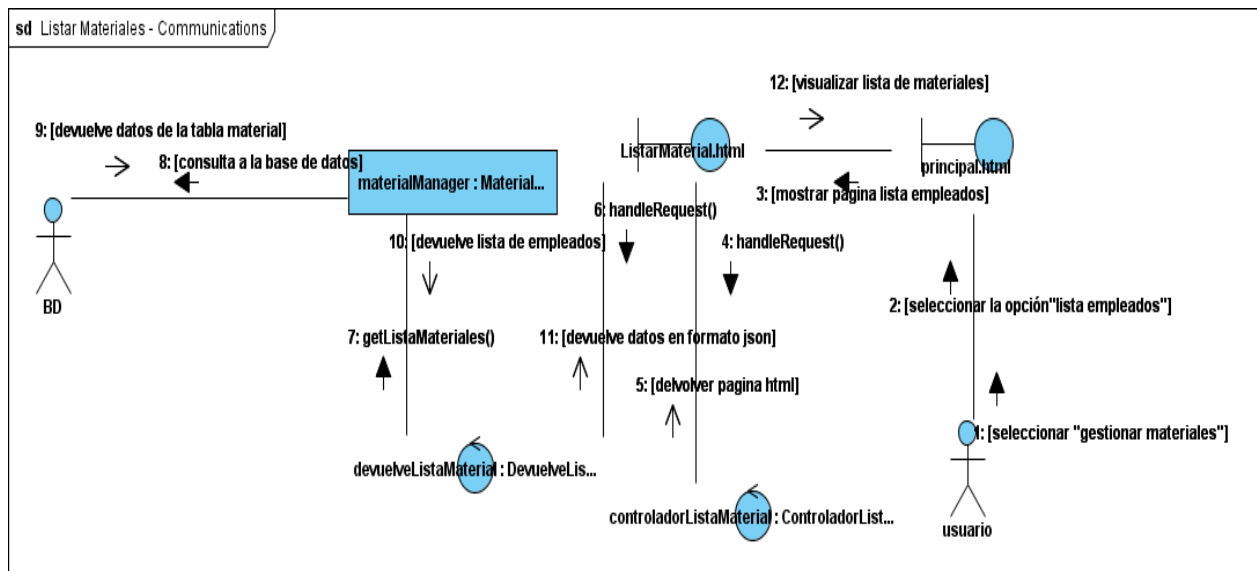


Fig.II.96 diagrama de Colaboración –Listar Materiles

Diagrama de Colaboración: Listar Categorías

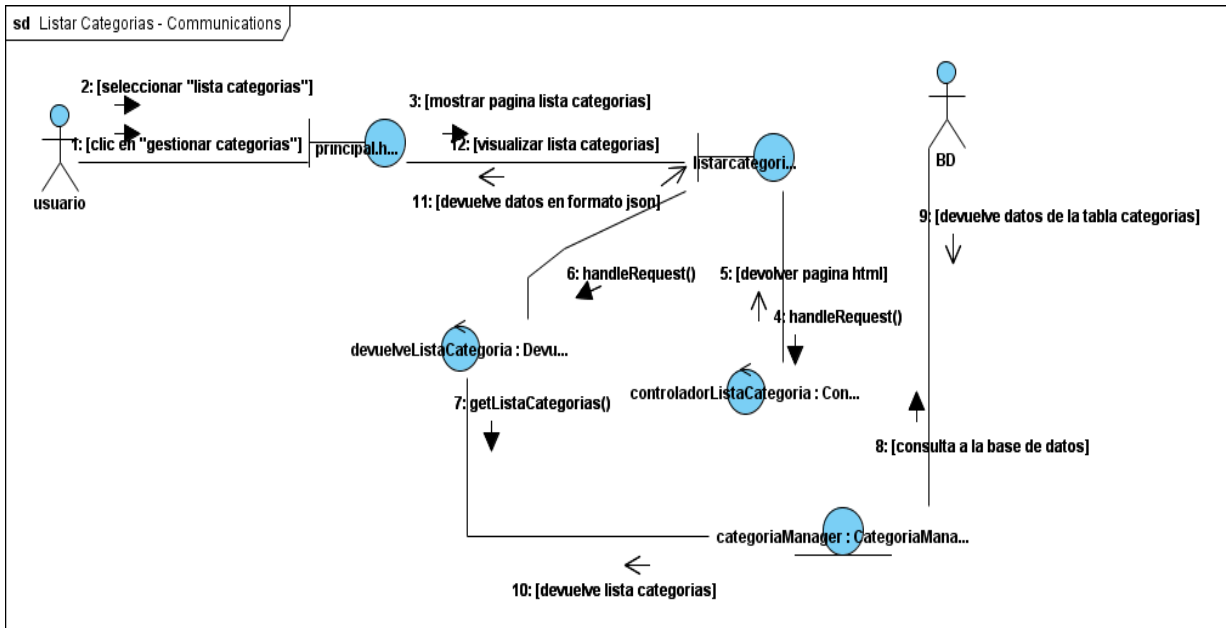


Fig.II.97 diagrama de Colaboración –Listar Categorías

Diagramas de Colaboración: Listar Solicitud

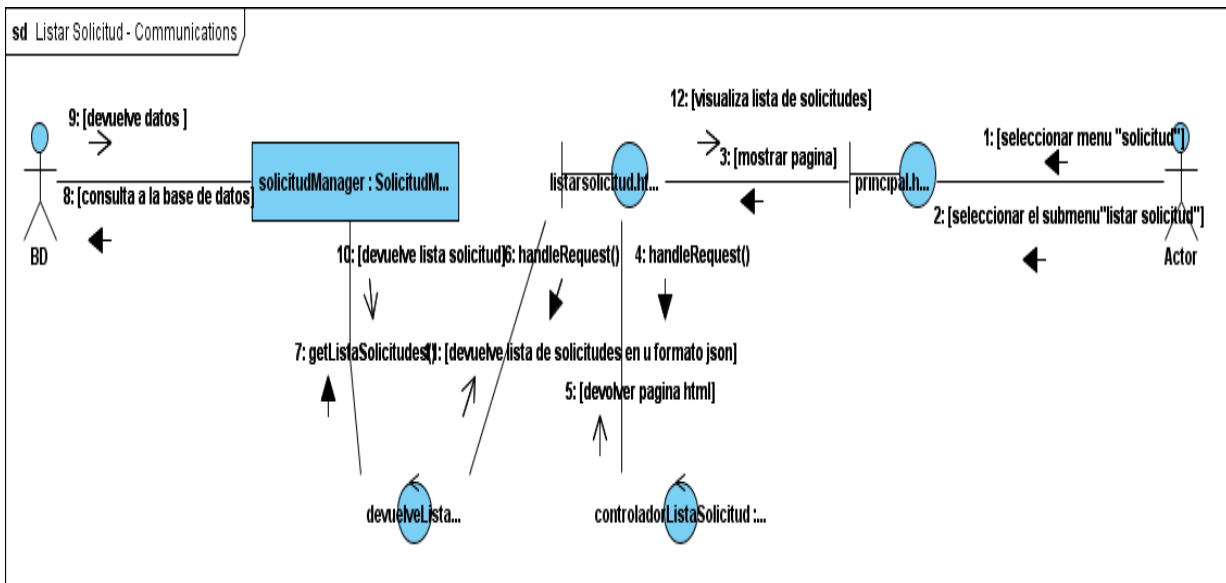


Fig.II.98 diagrama de Colaboración –Listar Solicitud

Diagrama de Colaboración: Modificar solicitud

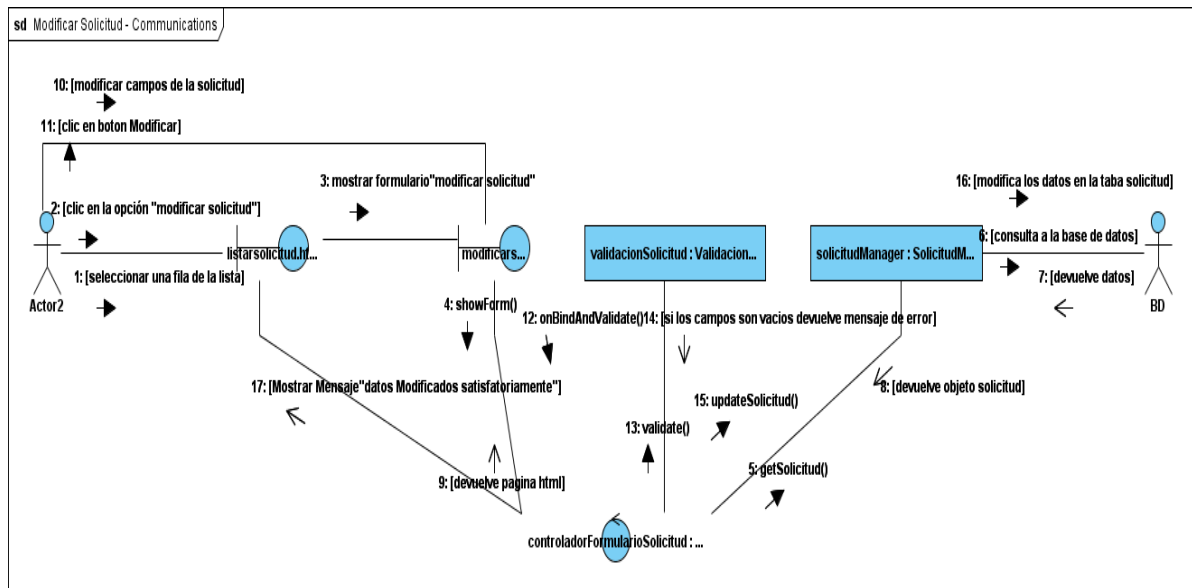


Fig.II.99 diagrama de Colaboración –Modificar Solicitud

Diagrama de Colaboración: Listar Cotización

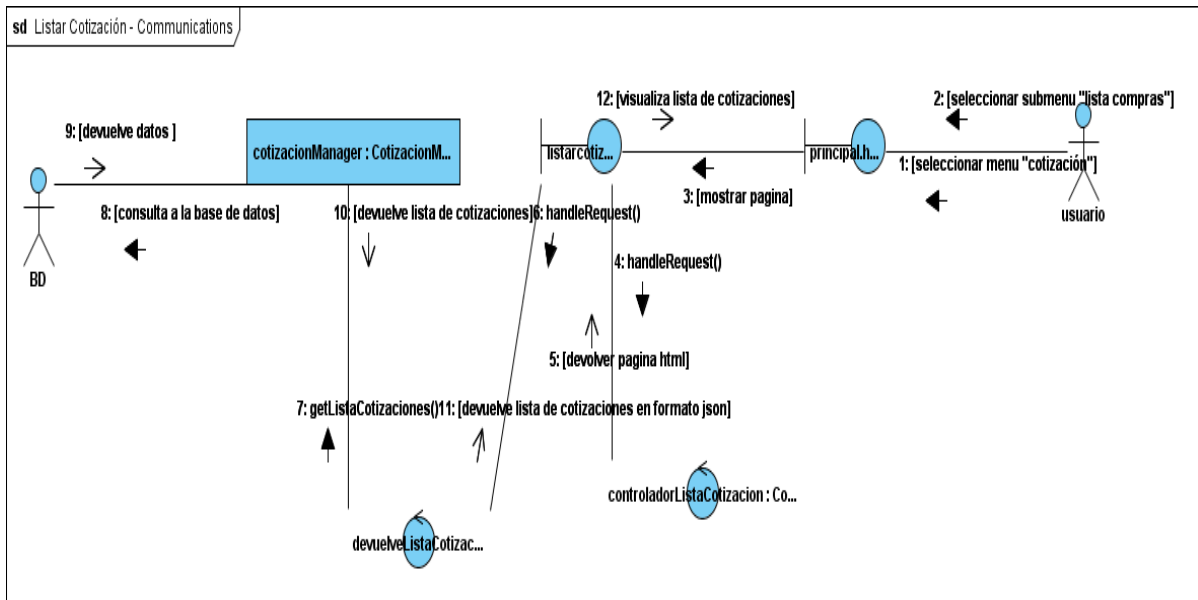


Fig.II.100 diagrama de Colaboración –Listar Cotización

Diagrama de Colaboración: Listar Aprobación

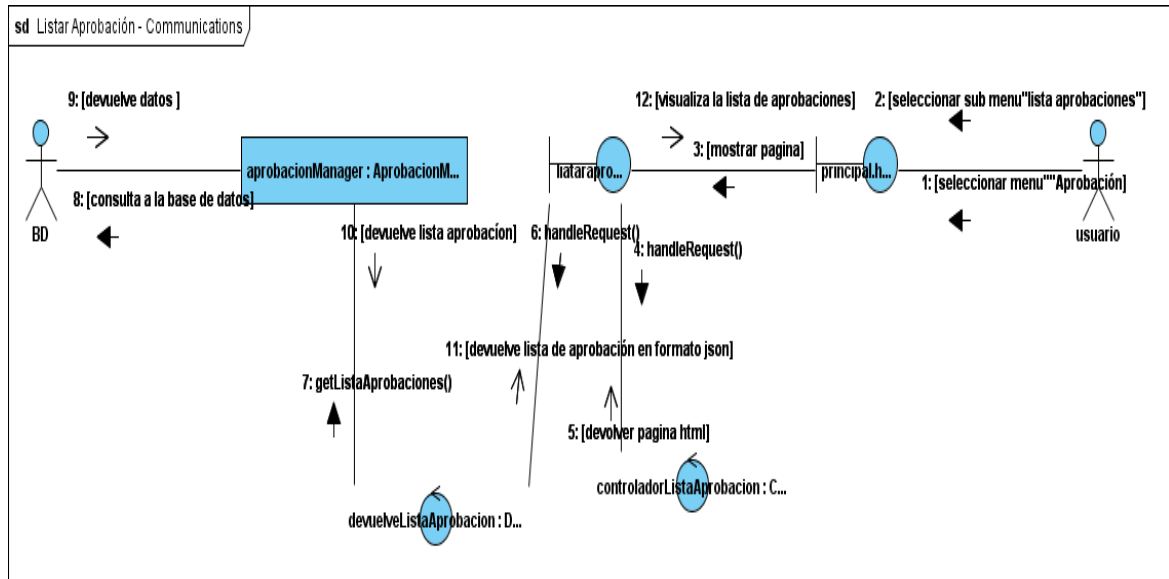


Fig.II.101 diagrama de Colaboración –Listar aprobación

Diagrama de Colaboración: Listar Compra

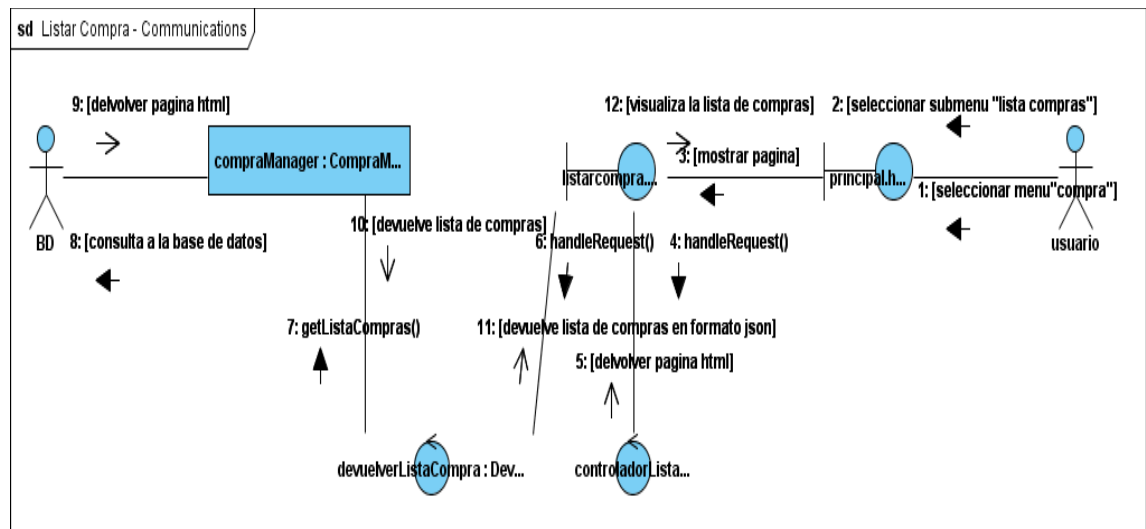


Fig.II.102 diagrama de Colaboración –Listar Compras

Diagrama de Colaboración: Adicionar Entradas

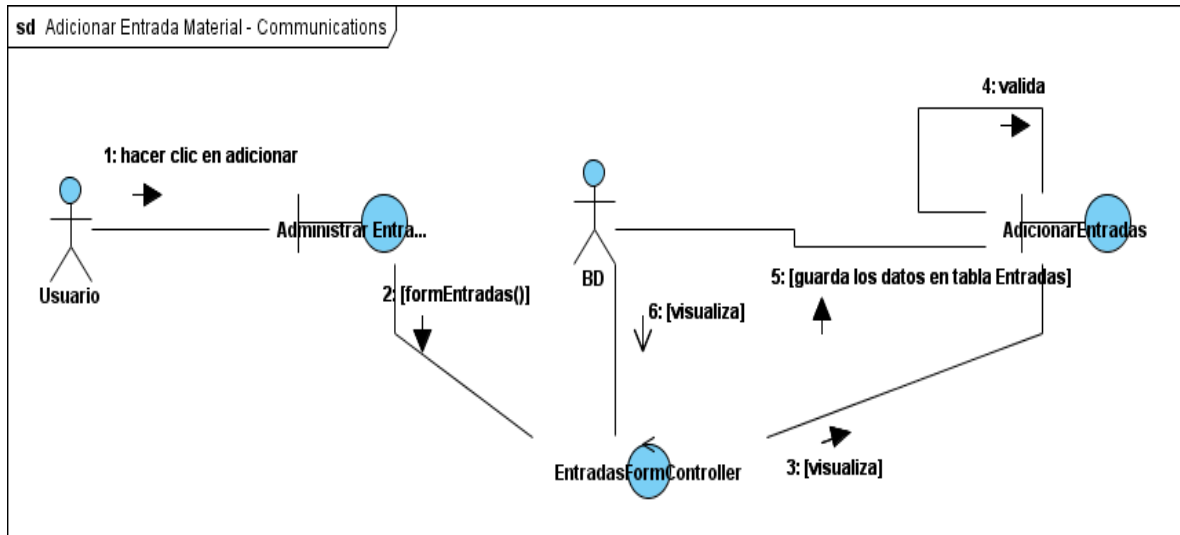


Fig.II.103 diagrama de Colaboración –Listar Entradas

Diagramas de Colaboración: Adicionar Salidas

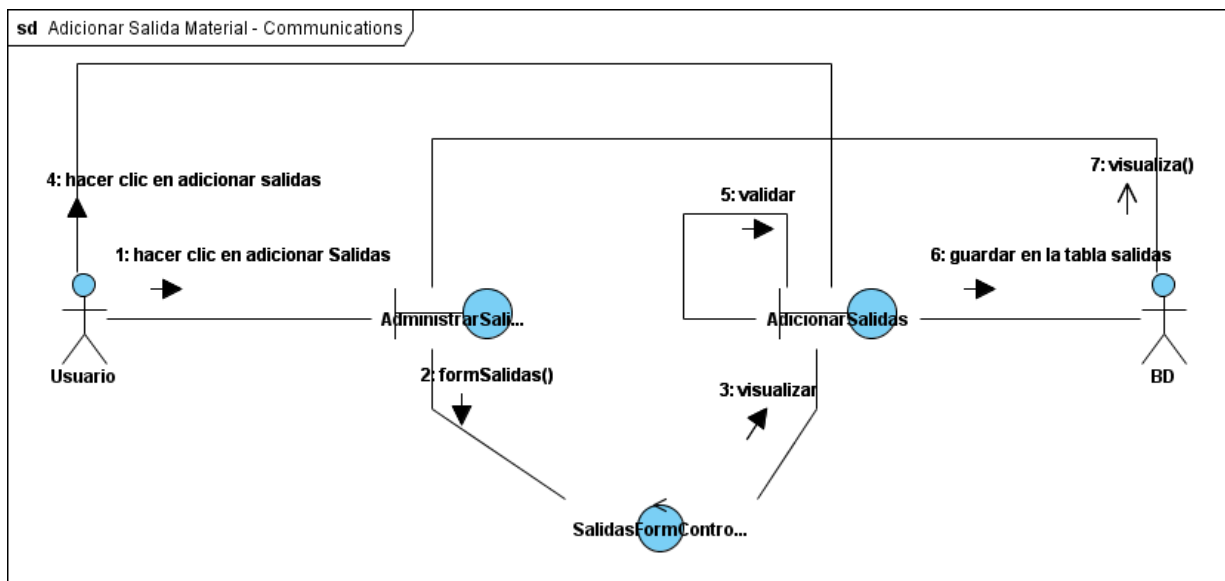


Fig.II.104 diagrama de Colaboración –Adicionar Salida

II.2.10. MODELO DE DATOS

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.

II.2.11. IMPLEMENTACION DEL DIAGRAMA DE OBJETOS DE CLASES EN JAVA.

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código de fuente y todo tipo de ficheros necesarios para la implementación y despliegue del sistema.

CLASE: SOLICITUD

```
public class Solicitud implements java.io.Serializable {
    private int codSolicitud;
    private Empleado empleado;
    private Date fechaSolicitud;
    private String descripcion;
    private Set cheques = new HashSet(0);
    private Set oficialias = new HashSet(0);
    private Set cotizacions = new HashSet(0);
    private Set aprobacions = new HashSet(0);
    private Set compras = new HashSet(0);
    private Set detalleSolicitudes = new HashSet(0);

    public Solicitud() {
    }

    public Solicitud(int codSolicitud) {
        this.codSolicitud = codSolicitud;
    }

    public Solicitud(int codSolicitud, Empleado empleado, Date fechaSolicitud,
        String descripcion, Set cheques, Set oficialias, Set cotizacions,
        Set aprobacions, Set compras, Set detalleSolicitudes) {
        this.codSolicitud = codSolicitud;
        this.empleado = empleado;
        this.fechaSolicitud = fechaSolicitud;
```

```
        this.descripcion = descripcion;
        this.chèques = chèques;
        this.oficialias = oficialias;
        this.cotizacions = cotizacions;
        this.aprobacions = aprobacions;
        this.compras = compras;
        this.detalleSolicituds = detalleSolicituds;
    }

    public int getCodSolicitud() {
        return this.codSolicitud;
    }

    public void setCodSolicitud(int codSolicitud) {
        this.codSolicitud = codSolicitud;
    }

    public Empleado getEmpleado() {
        return this.empleado;
    }

    public void setEmpleado(Empleado empleado) {
        this.empleado = empleado;
    }

    public Date getFechaSolicitud() {
        return this.fechaSolicitud;
    }

    public void setFechaSolicitud(Date fechaSolicitud) {
        this.fechaSolicitud = fechaSolicitud;
    }

    public String getDescripcion() {
```

```

        return this.descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public Set getCheques() {
        return this.cheques;
    }

    public void setCheques(Set cheques) {
        this.cheques = cheques;
    }

    public Set getOficialias() {
        return this.oficialias;
    }

    public void setOficialias(Set oficialias) {
        this.oficialias = oficialias;
    }

    public Set getCotizaciones() {
        return this.cotizaciones;
    }

    public void setCotizaciones(Set cotizaciones) {
        this.cotizaciones = cotizaciones;
    }

    public Set getAprobaciones()
    {
        return this.aprobaciones;
    }

    public void setAprobaciones(Set aprobaciones) {
        this.aprobaciones = aprobaciones;
    }

```

```

    }
    public Set getCompras() {
        return this.compras;
    }
    public void setCompras(Set compras) {
        this.compras = compras;
    }
    public Set getDetalleSolicitudes() {
        return this.detalleSolicitudes;
    }
    public void setDetalleSolicitudes(Set detalleSolicitudes) {
        this.detalleSolicitudes = detalleSolicitudes;
    }
}

```

CLASE: COTIZACIÓN

```

Public class Cotizacion implements java.io.Serializable {

    private int codCotizacion;
    private TipoCot tipoCot;
    private Empleado empleado;
    private Empresa empresa;
    private Solicitud solicitud;
    private Double montoTotal;
    private Date fechaCotizacion;
    private Set detalleCotizaciones = new HashSet(0);

    public Cotizacion() {
    }
}

```

```

    public Cotizacion(int codCotizacion, TipoCot tipoCot, Empleado empleado,
        Empresa empresa, Date fechaCotizacion) {
        this.codCotizacion = codCotizacion;
        this.tipoCot = tipoCot;
        this.empleado = empleado;
        this.empresa = empresa;
        this.fechaCotizacion = fechaCotizacion;
    }

    public Cotizacion(int codCotizacion, TipoCot tipoCot, Empleado empleado,
        Empresa empresa, Solicitud solicitud, Double montoTotal,
        Date fechaCotizacion, Set detalleCotizacions) {
        this.codCotizacion = codCotizacion;
        this.tipoCot = tipoCot;
        this.empleado = empleado;
        this.empresa = empresa;
        this.solicitud = solicitud;
        this.montoTotal = montoTotal;
        this.fechaCotizacion = fechaCotizacion;
        this.detalleCotizacions = detalleCotizacions;
    }

    public int getCodCotizacion() {
        return this.codCotizacion;
    }

    public void setCodCotizacion(int codCotizacion) {
        this.codCotizacion = codCotizacion;
    }

    public TipoCot getTipoCot() {
        return this.tipoCot;
    }

```

```
public void setTipoCot(TipoCot tipoCot) {
    this.tipoCot = tipoCot;
}

public Empleado getEmpleado() {
    return this.empleado;
}

public void setEmpleado(Empleado empleado) {
    this.empleado = empleado;
}

public Empresa getEmpresa() {
    return this.empresa;
}

public void setEmpresa(Empresa empresa) {
    this.empresa = empresa;
}

public Solicitud getSolicitud() {
    return this.solicitud;
}

public void setSolicitud(Solicitud solicitud) {
    this.solicitud = solicitud;
}

public Double getMontoTotal() {
    return this.montoTotal;
}

public void setMontoTotal(Double montoTotal) {
```

```

        this.montoTotal = montoTotal;
    }

    public Date getFechaCotizacion() {
        return this.fechaCotizacion;
    }

    public void setFechaCotizacion(Date fechaCotizacion) {
        this.fechaCotizacion = fechaCotizacion;
    }

    public Set getDetalleCotizacions() {
        return this.detalleCotizacions;
    }

    public void setDetalleCotizacions(Set detalleCotizacions) {
        this.detalleCotizacions = detalleCotizacions;
    }
}

```

CLASE: APROBACIÓN

```

public class Aprobacion implements java.io.Serializable {
    private int codAprobacion;
    private Empleado empleado;
    private Solicitud solicitud;
    private Date fechaLlegada;
    private Date fechaAprobacion;
    private Integer resultadoAprob;
    private Set ordenSalidas = new HashSet(0);
    private Set materialAprobacions = new HashSet(0);
}

```

```

public Aprobacion() {
}

public Aprobacion(int codAprobacion) {
    this.codAprobacion = codAprobacion;
}

public Aprobacion(int codAprobacion, Empleado empleado,
                  Solicitud solicitud, Date fechaLlegada, Date fechaAprobacion,
                  Integer resultadoAprob, Set ordenSalidas, Set
materialAprobacions) {
    this.codAprobacion = codAprobacion;
    this.empleado = empleado;
    this.solicitud = solicitud;
    this.fechaLlegada = fechaLlegada;
    this.fechaAprobacion = fechaAprobacion;
    this.resultadoAprob = resultadoAprob;
    this.ordenSalidas = ordenSalidas;
    this.materialAprobacions = materialAprobacions;
}

public int getCodAprobacion() {
    return this.codAprobacion;
}

public void setCodAprobacion(int codAprobacion) {
    this.codAprobacion = codAprobacion;
}

```

```
    public Empleado getEmpleado() {
        return this.empleado;
    }

    public void setEmpleado(Empleado empleado) {
        this.empleado = empleado;
    }

    public Solicitud getSolicitud() {
        return this.solicitud;
    }

    public void setSolicitud(Solicitud solicitud) {
        this.solicitud = solicitud;
    }

    public Date getFechaLlegada() {
        return this.fechaLlegada;
    }

    public void setFechaLlegada(Date fechaLlegada) {
        this.fechaLlegada = fechaLlegada;
    }

    public Date getFechaAprobacion() {
        return this.fechaAprobacion;
    }

    public void setFechaAprobacion(Date fechaAprobacion) {
        this.fechaAprobacion = fechaAprobacion;
    }

    public Integer getResultadoAprob() {
        return this.resultadoAprob;
    }

    public void setResultadoAprob(Integer resultadoAprob) {
        this.resultadoAprob = resultadoAprob;
    }
}
```

```

    }

    public Set getOrdenSalidas() {
        return this.ordenSalidas;
    }

    public void setOrdenSalidas(Set ordenSalidas) {
        this.ordenSalidas = ordenSalidas;
    }

    public Set getMaterialAprobaciones() {
        return this.materialAprobaciones;
    }

    public void setMaterialAprobaciones(Set materialAprobaciones) {
        this.materialAprobaciones = materialAprobaciones;
    }
}

```

CLASE: COMPRA

```

public class Compra implements java.io.Serializable {
    private int codCompra;

    private Solicitud solicitud;
    private Date fechaCompra;
    private String formaPago;
    private Double descuento;
    private Double total;
    private String observacion;
    private Integer nroCheque;
    private String nombreEmpresa;
    private Set presupuestos = new HashSet(0);
    private Set detalleCompras = new HashSet(0);

    public Compra() {
    }
}

```

```

public Compra(int codCompra) {
    this.codCompra = codCompra;
}

public Compra(int codCompra, Solicitud solicitud, Date fechaCompra,
    String formaPago, Double descuento, Double total,
    String observacion, Integer nroCheque, String
nombreEmpresa,
    Set presupuestos, Set detalleCompras) {
    this.codCompra = codCompra;
    this.solicitud = solicitud;
    this.fechaCompra = fechaCompra;
    this.formaPago = formaPago;
    this.descuento = descuento;
    this.total = total;
    this.observacion = observacion;
    this.nroCheque = nroCheque;
    this.nombreEmpresa = nombreEmpresa;
    this.presupuestos = presupuestos;
    this.detalleCompras = detalleCompras;
}

public int getCodCompra() {
    return this.codCompra;
}

public void setCodCompra(int codCompra) {
    this.codCompra = codCompra;
}

public Solicitud getSolicitud() {
    return this.solicitud;
}

```

```
    }  
    public void setSolicitud(Solicitud solicitud) {  
        this.solicitud = solicitud;  
    }  
    public Date getFechaCompra() {  
        return this.fechaCompra;  
    }  
    public void setFechaCompra(Date fechaCompra) {  
        this.fechaCompra = fechaCompra;  
    }  
    public String getFormaPago() {  
        return this.formaPago;  
    }  
    public void setFormaPago(String formaPago) {  
        this.formaPago = formaPago;  
    }  
    public Double getDescuento() {  
        return this.descuento;  
    }  
    public void setDescuento(Double descuento) {  
        this.descuento = descuento;  
    }  
    public Double getTotal() {  
        return this.total;  
    }  
    public void setTotal(Double total) {  
        this.total = total;  
    }  
    public String getObservacion() {  
        return this.observacion;  
    }
```

```

    }

    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }

    public Integer getNroCheque() {
        return this.nroCheque;
    }

    public void setNroCheque(Integer nroCheque) {
        this.nroCheque = nroCheque;
    }

    public String getNombreEmpresa() {
        return this.nombreEmpresa;
    }

    public void setNombreEmpresa(String nombreEmpresa) {
        this.nombreEmpresa = nombreEmpresa;
    }

    public Set getPresupuestos() {
        return this.presupuestos;
    }

    public void setPresupuestos(Set presupuestos) {
        this.presupuestos = presupuestos;
    }

    public Set getDetalleCompras() {
        return this.detalleCompras;
    }

    public void setDetalleCompras(Set detalleCompras) {
        this.detalleCompras = detalleCompras;
    }
}

```

CLASE: DETALLE COMPRA

```
public class DetalleCompra implements java.io.Serializable {

    private DetalleCompraId id;
    private Material material;
    private Compra compra;
    private int cantidadDetalle;
    private float precioDetalle;
    private float montoTotalDetalle;
    private String observacionDetalle;

    public DetalleCompra() {
    }

    public DetalleCompra(DetalleCompraId id, Material material, Compra
compra,
                        int cantidadDetalle, float precioDetalle, float
montoTotalDetalle) {
        this.id = id;
        this.material = material;
        this.compra = compra;
        this.cantidadDetalle = cantidadDetalle;
        this.precioDetalle = precioDetalle;
        this.montoTotalDetalle = montoTotalDetalle;
    }

    public DetalleCompra(DetalleCompraId id, Material material, Compra
compra,
```

```
        int cantidadDetalle, float precioDetalle, float  
montoTotalDetalle,
```

```
        String observacionDetalle) {  
    this.id = id;  
    this.material = material;  
    this.compra = compra;  
    this.cantidadDetalle = cantidadDetalle;  
    this.precioDetalle = precioDetalle;  
    this.montoTotalDetalle = montoTotalDetalle;  
    this.observacionDetalle = observacionDetalle;  
}
```

```
public DetalleCompraId getId() {  
    return this.id;  
}
```

```
public void setId(DetalleCompraId id) {  
    this.id = id;  
}
```

```
public Material getMaterial() {  
    return this.material;  
}
```

```
public void setMaterial(Material material) {  
    this.material = material;  
}
```

```
public Compra getCompra() {  
    return this.compra;  
}
```

```
}
```

```
public void setCompra(Compra compra) {  
    this.compra = compra;  
}
```

```
public int getCantidadDetalle() {  
    return this.cantidadDetalle;  
}
```

```
public void setCantidadDetalle(int cantidadDetalle) {  
    this.cantidadDetalle = cantidadDetalle;  
}
```

```
public float getPrecioDetalle() {  
    return this.precioDetalle;  
}
```

```
public void setPrecioDetalle(float precioDetalle) {  
    this.precioDetalle = precioDetalle;  
}
```

```
public float getMontoTotalDetalle() {  
    return this.montoTotalDetalle;  
}
```

```
public void setMontoTotalDetalle(float montoTotalDetalle) {  
    this.montoTotalDetalle = montoTotalDetalle;  
}
```

```

    public String getObservacionDetalle() {
        return this.observacionDetalle;
    }

    public void setObservacionDetalle(String observacionDetalle) {
        this.observacionDetalle = observacionDetalle;
    }

}

```

CLASE: MATERIAL

```

package model.domain;

public class Material implements java.io.Serializable {
    private int codMaterial;

    private Categoria categoria;
    private String nombre;
    private String unidadMedida;
    private String descripcion;
    private Set empresaMaterials = new HashSet(0);
    private Set detalleSolicitudes = new HashSet(0);
    private Set detalleSalidaMaterials = new HashSet(0);
    private Set almacenMaterials = new HashSet(0);
    private Set detalleCotizaciones = new HashSet(0);
    private Set materialAprobaciones = new HashSet(0);
    private Set detalleCompras = new HashSet(0);
    private Set detalleEntradas = new HashSet(0);

    public Material() {
    }
}

```

```

public Material(int codMaterial, Categoria categoria) {
    this.codMaterial = codMaterial;
    this.categoria = categoria;
}

public Material(int codMaterial, Categoria categoria, String nombre,
                String unidadMedida, String descripcion, Set
empresaMaterials,
                Set detalleSolicitudes, Set detalleSalidaMaterials,
                Set almacenMaterials, Set detalleCotizaciones,
                Set materialAprobaciones, Set detalleCompras, Set
detalleEntradas) {
    this.codMaterial = codMaterial;
    this.categoria = categoria;
    this.nombre = nombre;
    this.unidadMedida = unidadMedida;
    this.descripcion = descripcion;
    this.empresaMaterials = empresaMaterials;
    this.detalleSolicitudes = detalleSolicitudes;
    this.detalleSalidaMaterials = detalleSalidaMaterials;
    this.almacenMaterials = almacenMaterials;
    this.detalleCotizaciones = detalleCotizaciones;
    this.materialAprobaciones = materialAprobaciones;
    this.detalleCompras = detalleCompras;
    this.detalleEntradas = detalleEntradas;
}

public int getCodMaterial() {
    return this.codMaterial;
}

```

```
public void setCodMaterial(int codMaterial) {  
    this.codMaterial = codMaterial;  
}
```

```
public Categoria getCategoria() {  
    return this.categoria;  
}
```

```
public void setCategoria(Categoria categoria) {  
    this.categoria = categoria;  
}
```

```
public String getNombre() {  
    return this.nombre;  
}
```

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

```
public String getUnidadMedida() {  
    return this.unidadMedida;  
}
```

```
public void setUnidadMedida(String unidadMedida) {  
    this.unidadMedida = unidadMedida;  
}
```

```
public String getDescripcion() {
```

```
        return this.descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public Set getEmpresaMaterials() {
        return this.empresaMaterials;
    }

    public void setEmpresaMaterials(Set empresaMaterials) {
        this.empresaMaterials = empresaMaterials;
    }

    public Set getDetalleSolicitudes() {
        return this.detalleSolicitudes;
    }

    public void setDetalleSolicitudes(Set detalleSolicitudes) {
        this.detalleSolicitudes = detalleSolicitudes;
    }

    public Set getDetalleSalidaMaterials() {
        return this.detalleSalidaMaterials;
    }

    public void setDetalleSalidaMaterials(Set detalleSalidaMaterials) {
        this.detalleSalidaMaterials = detalleSalidaMaterials;
    }
}
```

```
public Set getAlmacenMaterials() {  
    return this.almacenMaterials;  
}
```

```
public void setAlmacenMaterials(Set almacenMaterials) {  
    this.almacenMaterials = almacenMaterials;  
}
```

```
public Set getDetalleCotizacions() {  
    return this.detalleCotizacions;  
}
```

```
public void setDetalleCotizacions(Set detalleCotizacions) {  
    this.detalleCotizacions = detalleCotizacions;  
}
```

```
public Set getMaterialAprobacions() {  
    return this.materialAprobacions;  
}
```

```
public void setMaterialAprobacions(Set materialAprobacions) {  
    this.materialAprobacions = materialAprobacions;  
}
```

```
public Set getDetalleCompras() {  
    return this.detalleCompras;  
}
```

```
public void setDetalleCompras(Set detalleCompras) {
```

```

        this.detalleCompras = detalleCompras;
    }

    public Set getDetalleEntradas() {
        return this.detalleEntradas;
    }

    public void setDetalleEntradas(Set detalleEntradas) {
        this.detalleEntradas = detalleEntradas;
    }
}

```

CLASE: EMPRESA

```

public class Empresa implements java.io.Serializable {
    private int codEmpresa;
    private String nombre;
    private Integer nit;
    private String direccion;
    private String telefono;
    private String correoElectronico;
    private Integer estado;
    private Set cotizacions = new HashSet(0);
    private Set detalleCotizacions = new HashSet(0);
    private Set empresaMaterials = new HashSet(0);
    private Set proveedors = new HashSet(0);

    public Empresa() {
    }

    public Empresa(int codEmpresa) {
        this.codEmpresa = codEmpresa;
    }
}

```

```

public Empresa(int codEmpresa, String nombre, Integer nit,
               String direccion, String telefono, String correoElectronico,
               Integer estado, Set cotizacions, Set detalleCotizacions,
               Set empresaMaterials, Set proveedores) {
    this.codEmpresa = codEmpresa;
    this.nombre = nombre;
    this.nit = nit;
    this.direccion = direccion;
    this.telefono = telefono;
    this.correoElectronico = correoElectronico;
    this.estado = estado;
    this.cotizacions = cotizacions;
    this.detalleCotizacions = detalleCotizacions;
    this.empresaMaterials = empresaMaterials;
    this.proveedores = proveedores;
}

public int getCodEmpresa() {
    return this.codEmpresa;
}

public void setCodEmpresa(int codEmpresa) {
    this.codEmpresa = codEmpresa;
}

public String getNombre() {
    return this.nombre;
}

```

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

```
public Integer getNit() {  
    return this.nit;  
}
```

```
public void setNit(Integer nit) {  
    this.nit = nit;  
}
```

```
public String getDireccion() {  
    return this.direccion;  
}
```

```
public void setDireccion(String direccion) {  
    this.direccion = direccion;  
}
```

```
public String getTelefono() {  
    return this.telefono;  
}
```

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

```
public String getCorreoElectronico() {  
    return this.correoElectronico;  
}
```

```
}
```

```
public void setCorreoElectronico(String correoElectronico) {  
    this.correoElectronico = correoElectronico;  
}
```

```
public Integer getEstado() {  
    return this.estado;  
}
```

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

```
public Set getCotizacions() {  
    return this.cotizacions;  
}
```

```
public void setCotizacions(Set cotizacions) {  
    this.cotizacions = cotizacions;  
}
```

```
public Set getDetalleCotizacions() {  
    return this.detalleCotizacions;  
}
```

```
public void setDetalleCotizacions(Set detalleCotizacions) {  
    this.detalleCotizacions = detalleCotizacions;  
}
```

```

    public Set getEmpresaMaterials() {
        return this.empresaMaterials;
    }

    public void setEmpresaMaterials(Set empresaMaterials) {
        this.empresaMaterials = empresaMaterials;
    }

    public Set getProveedores() {
        return this.proveedores;
    }

    public void setProveedores(Set proveedores) {
        this.proveedores = proveedores;
    }
}

```

CLASE: EMPRESA MATERIAL

```

public class EmpresaMaterial implements java.io.Serializable {

    private EmpresaMaterialId id;
    private Material material;
    private Empresa empresa;
    private double precioRef;

    public EmpresaMaterial() {
    }
}

```

```
public EmpresaMaterial(EmpresaMaterialId id, Material material,  
    Empresa empresa, double precioRef) {  
    this.id = id;  
    this.material = material;  
    this.empresa = empresa;  
    this.precioRef = precioRef;  
}
```

```
public EmpresaMaterialId getId() {  
    return this.id;  
}
```

```
public void setId(EmpresaMaterialId id) {  
    this.id = id;  
}
```

```
public Material getMaterial() {  
    return this.material;  
}
```

```
public void setMaterial(Material material) {  
    this.material = material;  
}
```

```
public Empresa getEmpresa() {  
    return this.empresa;  
}
```

```
public void setEmpresa(Empresa empresa) {  
    this.empresa = empresa;  
}
```

```

    }

    public double getPrecioRef() {
        return this.precioRef;
    }

    public void setPrecioRef(double precioRef) {
        this.precioRef = precioRef;
    }
}

```

CLASE: ENTRADA MATERIAL

```

public class EntradaMaterial implements java.io.Serializable {
    private int codEntradamat;
    private Empleado empleado;
    private Almacen almacen;
    private Date fechaEntrada;
    private String observaciones;
    private Integer idEmpleadoEntrega;
    private Set detalleEntradas = new HashSet(0);

    public EntradaMaterial() {
    }

    public EntradaMaterial(int codEntradamat, Empleado empleado,
        Almacen almacen, Date fechaEntrada) {
        this.codEntradamat = codEntradamat;
        this.empleado = empleado;
    }
}

```

```

        this.almacen = almacen;
        this.fechaEntrada = fechaEntrada;
    }

    public EntradaMaterial(int codEntradamat, Empleado empleado,
                           Almacen almacen, Date fechaEntrada, String observaciones,
                           Integer idEmpleadoEntrega, Set detalleEntradas) {
        this.codEntradamat = codEntradamat;
        this.empleado = empleado;
        this.almacen = almacen;
        this.fechaEntrada = fechaEntrada;
        this.observaciones = observaciones;
        this.idEmpleadoEntrega = idEmpleadoEntrega;
        this.detalleEntradas = detalleEntradas;
    }

    public int getCodEntradamat() {
        return this.codEntradamat;
    }

    public void setCodEntradamat(int codEntradamat) {
        this.codEntradamat = codEntradamat;
    }

    public Empleado getEmpleado() {
        return this.empleado;
    }

    public void setEmpleado(Empleado empleado) {
        this.empleado = empleado;
    }

```

```
public Almacen getAlmacen() {
    return this.almacen;
}

public void setAlmacen(Almacen almacen) {
    this.almacen = almacen;
}

public Date getFechaEntrada() {
    return this.fechaEntrada;
}

public void setFechaEntrada(Date fechaEntrada) {
    this.fechaEntrada = fechaEntrada;
}

public String getObservaciones() {
    return this.observaciones;
}

public void setObservaciones(String observaciones) {
    this.observaciones = observaciones;
}

public Integer getIdEmpleadoEntrega() {
    return this.idEmpleadoEntrega;
}

public void setIdEmpleadoEntrega(Integer idEmpleadoEntrega) {
    this.idEmpleadoEntrega = idEmpleadoEntrega;
}

public Set getDetalleEntradas() {
    return this.detalleEntradas;
}

public void setDetalleEntradas(Set detalleEntradas) {
    this.detalleEntradas = detalleEntradas;
}
```

```
    }  
}
```

CLASE: SALIDA MATERIAL

```
public class SalidaMaterial implements java.io.Serializable {  
    private int codSalidaMaterial;  
        private Empleado empleado;  
        private Almacen almacen;  
        private Date fechaSalida;  
        private String observaciones;  
        private String unidadSolicitante;  
        private int numeroSolicitud;  
        private int estado;  
        private int idEmpleadoSolicitante;  
        private Date fechaSolicitado;  
        private int idEmpleadoAutoriza;  
        private Date fechaAutoriza;  
        private Date fechaEntrega;  
        private int idEmpleadoRecibe;  
        private Date fechaRecibe;  
        private Set detalleSalidaMaterials = new HashSet(0);  
  
    public SalidaMaterial() {  
    }  
  
    public SalidaMaterial(int codSalidaMaterial, Empleado empleado,  
        Almacen almacen, Date fechaSalida, String unidadSolicitante,  
        int numeroSolicitud, int estado, int idEmpleadoSolicitante,
```

```

        Date fechaSolicitado, int idEmpleadoAutoriza, Date
fechaAutoriza,
        Date fechaEntrega, int idEmpleadoRecibe, Date fechaRecibe)
{
    this.codSalidaMaterial = codSalidaMaterial;
    this.empleado = empleado;
    this.almacen = almacen;
    this.fechaSalida = fechaSalida;
    this.unidadSolicitante = unidadSolicitante;
    this.numeroSolicitud = numeroSolicitud;
    this.estado = estado;
    this.idEmpleadoSolicitante = idEmpleadoSolicitante;
    this.fechaSolicitado = fechaSolicitado;
    this.idEmpleadoAutoriza = idEmpleadoAutoriza;
    this.fechaAutoriza = fechaAutoriza;
    this.fechaEntrega = fechaEntrega;
    this.idEmpleadoRecibe = idEmpleadoRecibe;
    this.fechaRecibe = fechaRecibe;
}

```

```

public SalidaMaterial(int codSalidaMaterial, Empleado empleado,
        Almacen almacen, Date fechaSalida, String observaciones,
        String unidadSolicitante, int numeroSolicitud, int estado,
        int idEmpleadoSolicitante, Date fechaSolicitado,
        int idEmpleadoAutoriza, Date fechaAutoriza, Date
fechaEntrega,
        int idEmpleadoRecibe, Date fechaRecibe, Set
detalleSalidaMaterials) {
    this.codSalidaMaterial = codSalidaMaterial;
    this.empleado = empleado;

```

```
        this.almacen = almacen;
        this.fechaSalida = fechaSalida;
        this.observaciones = observaciones;
        this.unidadSolicitante = unidadSolicitante;
        this.numeroSolicitud = numeroSolicitud;
        this.estado = estado;
        this.idEmpleadoSolicitante = idEmpleadoSolicitante;
        this.fechaSolicitado = fechaSolicitado;
        this.idEmpleadoAutoriza = idEmpleadoAutoriza;
        this.fechaAutoriza = fechaAutoriza;
        this.fechaEntrega = fechaEntrega;
        this.idEmpleadoRecibe = idEmpleadoRecibe;
        this.fechaRecibe = fechaRecibe;
        this.detalleSalidaMaterials = detalleSalidaMaterials;
    }

    public int getCodSalidaMaterial() {
        return this.codSalidaMaterial;
    }

    public void setCodSalidaMaterial(int codSalidaMaterial) {
        this.codSalidaMaterial = codSalidaMaterial;
    }

    public Empleado getEmpleado() {
        return this.empleado;
    }

    public void setEmpleado(Empleado empleado) {
        this.empleado = empleado;
    }
```

```
}
```

```
public Almacen getAlmacen() {  
    return this.almacen;  
}
```

```
public void setAlmacen(Almacen almacen) {  
    this.almacen = almacen;  
}
```

```
public Date getFechaSalida() {  
    return this.fechaSalida;  
}
```

```
public void setFechaSalida(Date fechaSalida) {  
    this.fechaSalida = fechaSalida;  
}
```

```
public String getObservaciones() {  
    return this.observaciones;  
}
```

```
public void setObservaciones(String observaciones) {  
    this.observaciones = observaciones;  
}
```

```
public String getUnidadSolicitante() {  
    return this.unidadSolicitante;  
}
```

```
public void setUnidadSolicitante(String unidadSolicitante) {  
    this.unidadSolicitante = unidadSolicitante;  
}
```

```
public int getNumeroSolicitud() {  
    return this.numeroSolicitud;  
}
```

```
public void setNumeroSolicitud(int numeroSolicitud) {  
    this.numeroSolicitud = numeroSolicitud;  
}
```

```
public int getEstado() {  
    return this.estado;  
}
```

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

```
public int getIdEmpleadoSolicitante() {  
    return this.idEmpleadoSolicitante;  
}
```

```
public void setIdEmpleadoSolicitante(int idEmpleadoSolicitante) {  
    this.idEmpleadoSolicitante = idEmpleadoSolicitante;  
}
```

```
public Date getFechaSolicitado() {  
    return this.fechaSolicitado;  
}
```

```
}
```

```
public void setFechaSolicitado(Date fechaSolicitado) {  
    this.fechaSolicitado = fechaSolicitado;  
}
```

```
public int getIdEmpleadoAutoriza() {  
    return this.idEmpleadoAutoriza;  
}
```

```
public void setIdEmpleadoAutoriza(int idEmpleadoAutoriza) {  
    this.idEmpleadoAutoriza = idEmpleadoAutoriza;  
}
```

```
public Date getFechaAutoriza() {  
    return this.fechaAutoriza;  
}
```

```
public void setFechaAutoriza(Date fechaAutoriza) {  
    this.fechaAutoriza = fechaAutoriza;  
}
```

```
public Date getFechaEntrega() {  
    return this.fechaEntrega;  
}
```

```
public void setFechaEntrega(Date fechaEntrega) {  
    this.fechaEntrega = fechaEntrega;  
}
```

```

    public int getIdEmpleadoRecibe() {
        return this.idEmpleadoRecibe;
    }

    public void setIdEmpleadoRecibe(int idEmpleadoRecibe) {
        this.idEmpleadoRecibe = idEmpleadoRecibe;
    }

    public Date getFechaRecibe() {
        return this.fechaRecibe;
    }

    public void setFechaRecibe(Date fechaRecibe) {
        this.fechaRecibe = fechaRecibe;
    }

    public Set getDetalleSalidaMaterials() {
        return this.detalleSalidaMaterials;
    }

    public void setDetalleSalidaMaterials(Set detalleSalidaMaterials) {
        this.detalleSalidaMaterials = detalleSalidaMaterials;
    }

}

```

CLASE: ORDEN SALIDA

```

public class OrdenSalida implements java.io.Serializable {
    private int codOrden;
    private Aprobacion aprobacion;
    private Date fechaOrden;
}

```

```

public OrdenSalida() {
}

public OrdenSalida(int codOrden, Aprobacion aprobacion) {
    this.codOrden = codOrden;
    this.aprobacion = aprobacion;
}

public OrdenSalida(int codOrden, Aprobacion aprobacion, Date fechaOrden)
{
    this.codOrden = codOrden;
    this.aprobacion = aprobacion;
    this.fechaOrden = fechaOrden;
}

public int getCodOrden() {
    return this.codOrden;
}

public void setCodOrden(int codOrden) {
    this.codOrden = codOrden;
}

public Aprobacion getAprobacion() {
    return this.aprobacion;
}

public void setAprobacion(Aprobacion aprobacion) {
    this.aprobacion = aprobacion;
}

public Date getFechaOrden() {
    return this.fechaOrden;
}

public void setFechaOrden(Date fechaOrden) {
    this.fechaOrden = fechaOrden;
}

```

}

II.2.12. DIAGRAMA GENERAL DE COMPONENTES

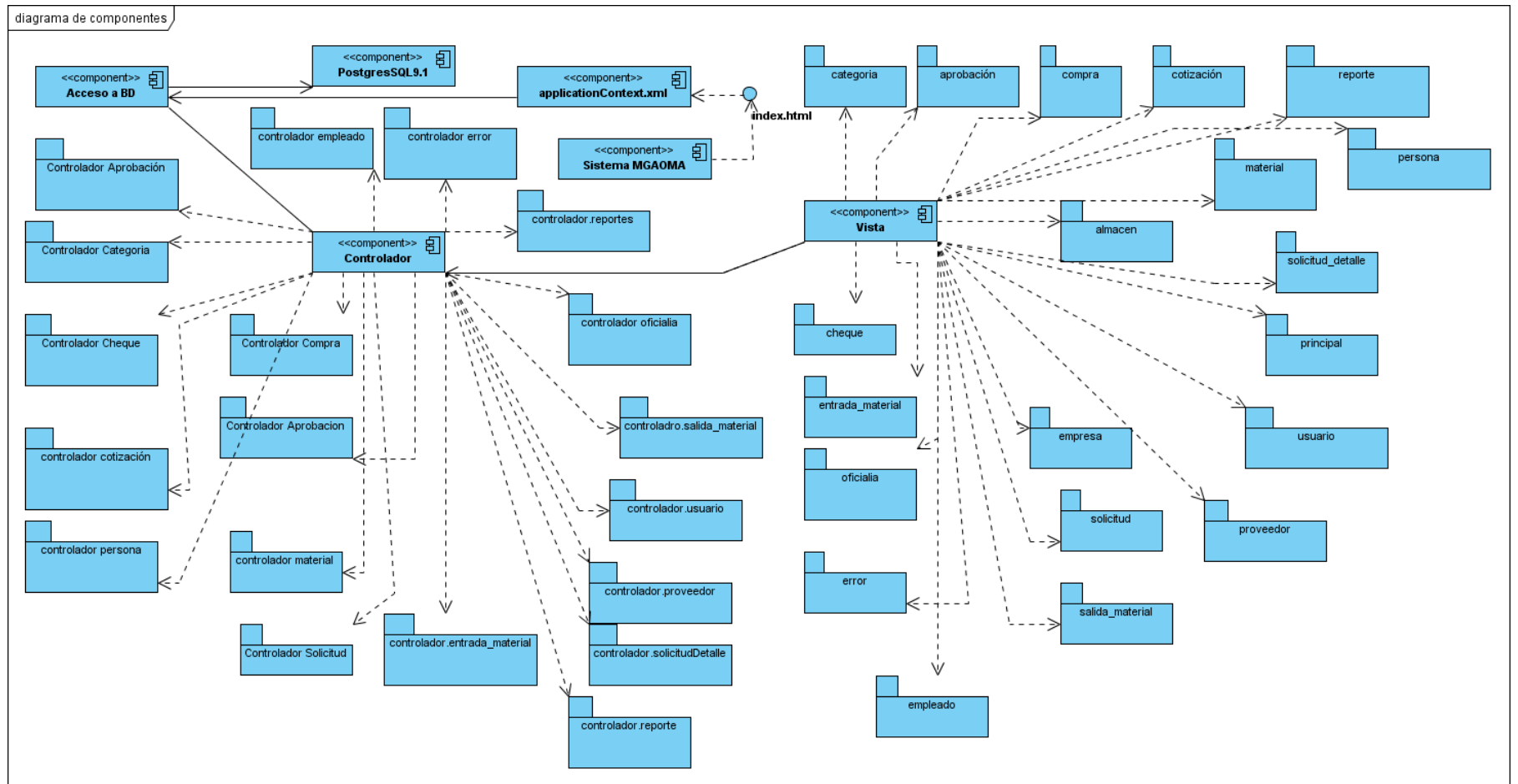


Fig.II .105 Diagrama de Componentes

II.2.13. PROTOTIPOS DE LA INTERFAZ DEL USUARIO

INTRODUCCION

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así conseguir retroalimentación de su parte respecto a los requisitos del sistema.

ALCANCE

Los prototipos de interfaz grafica no representan a las interfaces graficas definitivas que tendrá el sistema. Pueden sufrir algunas modificaciones leves.

II.2.13.1. DISEÑO DE LAS PANTALLAS DEL SISTEMA

A continuación se presentan los prototipos:

Pantalla1: Acceso al Sistema

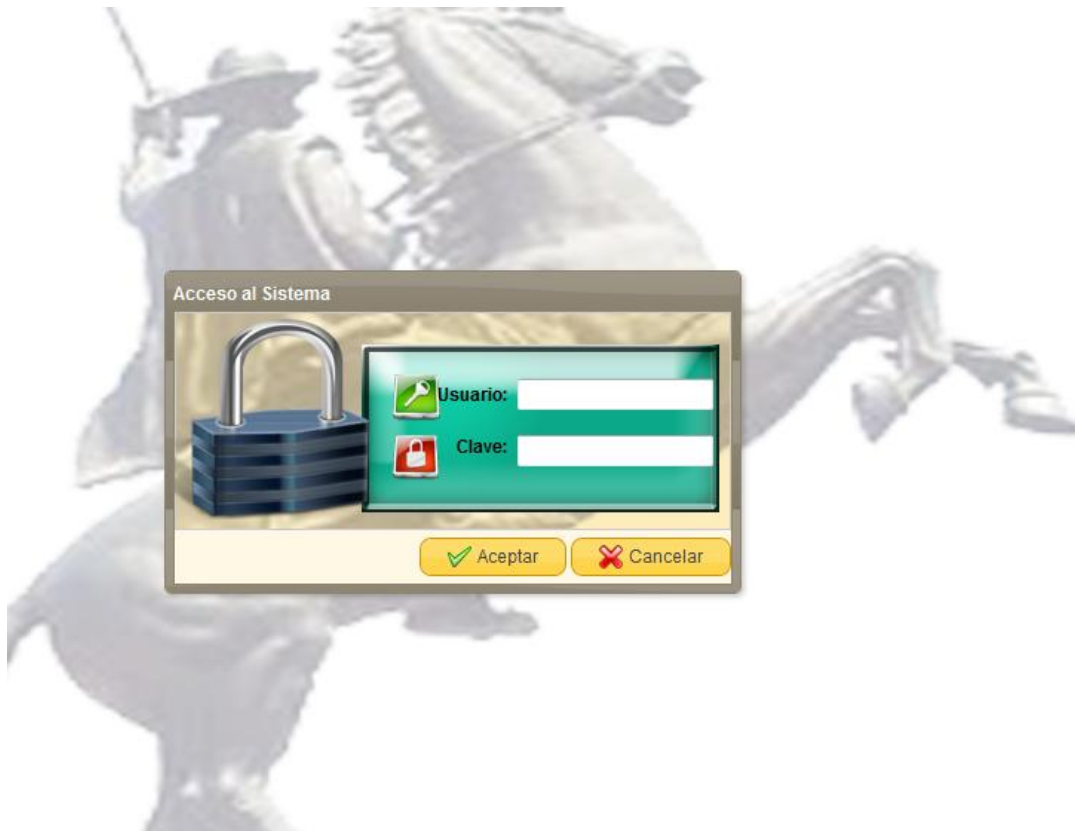


Fig.II.106 Pantalla-Ingresar Sistema

Pantalla 2:Menú Principal



Fig.II.107 Pantalla-Menu Principal

ADMINISTRAR SOLICITUD

Pantalla3:Listar solicitud

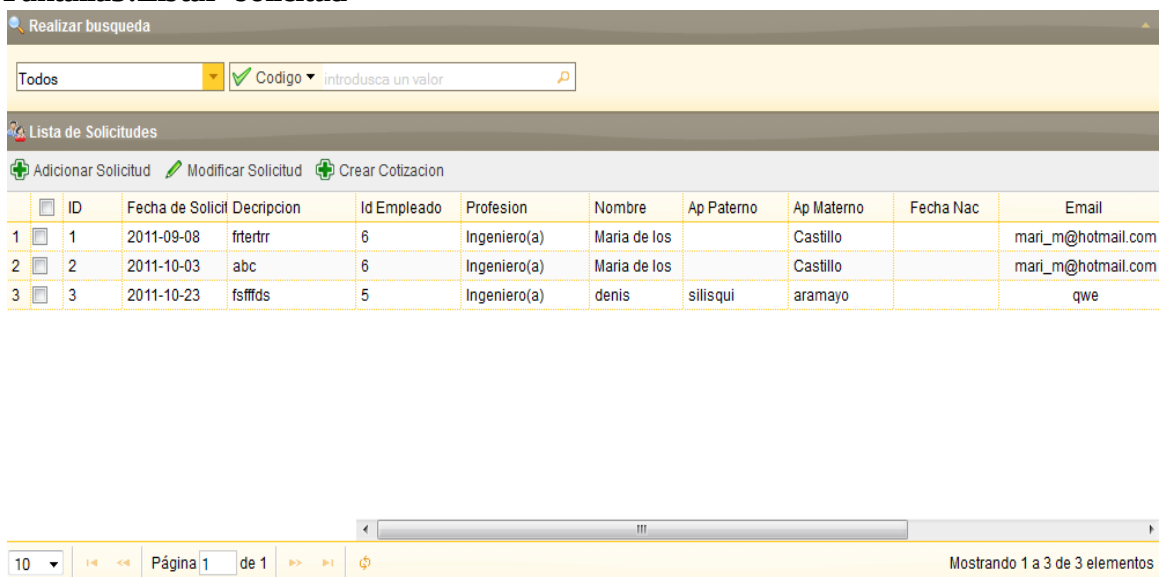


Fig.II.108 Pantalla-Listar Solicitud

Pantalla 4: Adicionar Solicitud

The screenshot shows a web application window titled 'Formulario de Solicitud'. The main heading is 'ADICIONAR SOLICITUD'. Below this, there are three input fields: 'Datos del Empleado' (a dropdown menu), 'Fecha Solicitud:' (a date picker showing '23-08-2013'), and 'Descripcion:' (a text area). Below these fields are three buttons: 'Fin Editar', 'Quitar Material', and 'Adicionar Detalle de Materiales a la Solicitud'. At the bottom, there is a section titled 'Detalle De Los Materiales' which contains a table with the following columns: ID, COD Material, Material, Cantidad, and Observacion. The table is currently empty.

Fig.II.109 Pantalla-Adicionar Solicitud

Pantalla 5: Modificar Solicitud

The screenshot shows a web application window titled 'Formulario de Solicitud'. The main heading is 'MODIFICAR SOLICITUD'. Below this, there are three input fields: 'Datos del Empleado' (a dropdown menu showing 'Rueda Castro Paola'), 'Fecha Solicitud:' (a date picker showing '16-12-2013'), and 'Descripcion:' (a text area containing 'Compra de material de plastico'). Below these fields are three buttons: 'Fin Editar', 'Quitar Material', and 'Adicionar Detalle de Materiales a la Solicitud'. At the bottom, there is a section titled 'Detalle De Los Materiales' which contains a table with the following columns: ID, COD Mat, Material, Cantidad, and Observacion. The table contains three rows of data:

ID	COD Mat	Material	Cantidad	Observacion
13	4	baneras	1	Ninguna
11	5	Silla	3	esta rota
12	9	Bañeras Grandes	4	Ninguna

At the bottom right of the window, there are two buttons: 'Modificar' and 'Cancelar'.

Fig.II.110 Pantalla-Modificar Solicitud

ADMINISTRAR COTIZACIÓN

Pantalla6: Listar Cotización

Realizar búsqueda

Cotizaciones ☒ Código de Cotización introduzca un valor

Lista de Cotizaciones

Ver Cotización

	ID	Monto Total	Código Solicitud	Fecha de Solicitud	Código Empresa	Empresa	Tipo de Cotización
1	1	34.0	1	2011-09-08	1	digital&computer	xxx
2	2	70.0	2	2011-10-03	5	Muebleria	yyy
3	3	64.0	3	2011-10-23	3	digitalpc	rrrr
4	4	250.5	14	2012-11-09	5	Muebleria	xxx
5	5	588.0	4	2011-11-16	5	Muebleria	xxx
6	6	491.8	15	2012-11-25	5	Muebleria	xxx

10

Fig.II.111 Pantalla-Listar Cotización

Pantalla7: Ver Cotización

Formulario de Cotización

COTIZACION

Monto_total: 93.6

Empresa: empresa 2

Cuadro Comparativo:
(Expresado en Bolivianos)

ITEM	DESCRIPCION	CANT.	UNIDAD DE MEDIDA	EMPRESAS				OBSERVACIONES	
				Muebleria "Lapacho"	empresa 2				
				P.UNIT	P.TOTAL	P.UNIT	P.TOTAL		
1	1	resmas	1	unidad	30.0	30.0	30.0	30.0	empresa 2
2	4	baneras	1	piezas	100.0	100.0	22.0	22.0	empresa 2
3	7	Mesa Redonda	1	pieza	50.0	50.0	41.6	41.6	empresa 2

Se recomienda la Empresa " empresa 2 " para realizar la Compra de estos Materiales por su bajo costo.

Fig.II.112 Pantalla-Ver Cotización

ADMINISTRAR APROBACIÓN

Pantalla 8: Lista Aprobación

Realizar búsqueda

Todos ☐ Código de Aprobación introduzca un valor

Lista de Aprobaciones

Modificar Aprobación

	ID	Fecha de Llegada	Fecha de Aprobación	Estado de Aprobación	Código de Solicitud	Fecha de Solicitud	Número de Documento	Nombre	Apellido Pater	Apellido Mater
1	1	2011-02-11	2011-04-11	Aprobados	1	2011-09-08	1234565 tja	Maria	Perez	Lopez
2	2	2011-07-11	2011-09-11	Aprobados	2	2011-10-03	345675 beni	Gabriela	Perez	Lopez
3	3	2011-03-04	2011-09-07	Aprobados	3	2011-10-23	467964 tja	Jose	Rojas	Tapia

10 Página 1 de 1 Mostrando 1 a 3 de 3 elementos

Fig.II.113 Pantalla-Lista Aprobación

Pantalla 9: Modificar Aprobación

Formulario de Aprobacion

MODIFICAR APROBACION

Cod_Aprobacion: 4

Fecha Llegada: 15-12-2013

✓ Materiales Aprobados

ID	COD Mate	Material	Cantidad	Observacion
7	Mesa Red	1	41.6	
1	resmas	1	30.0	
4	baneras	1	22.0	

3

Fig.II.114 Pantalla-Modificar Aprobación

ADMINISTRAR COMPRA

Pantalla 10 : Listar Compra

Realizar búsqueda

Registrados ☐ ☒ Código de Compra

Lista de Compras

☒ Registrar Compra ☒ Modificar Compra

	☐	ID	Código de Solic	Fecha de Solicitud	Fecha de Compra	Forma de Pago	Total	Descuento	Observaciones
1	☐	1	1	2011-09-08	2000-05-02	cheque	124.0	4.0	erer
2	☐	2	2	2011-10-03	2000-06-02	cheque	245.0	3.0	fewjre
3	☐	3	3	2011-10-23	2003-07-03	cheque	134.0	5.0	jghie
4	☐	4	15	2012-11-25	2012-11-25	cheque	170.0	0.0	asas asdfa

10 de 1 Mostrando 1 a 4 de 4 elementos

Fig.II.115 Pantalla-Listar Compra

Pantalla 11: Crear Compra

Formulario de Compra

COMPRA

Fecha de Compra:	24-08-2013	Forma Pago:	
Nombre Empresa:			
Numero de Cheque:	1		
Descuento:	0.0	Monto Total:	0.0
Observacion:			

☒ Fin Editar ☒ Quitar Material ☒ Adicionar Detalle de Materiales a la Solicitud

Detalle De Los Materiales

Cod Material	Material	Cantidad	Precio	Total	Observacion
--------------	----------	----------	--------	-------	-------------

☒ ☐ Cancelar

Fig.II.116 Pantalla-Crear Compra

Pantalla 12: Modificar Compra

Formulario de Compra

COMPRA

Fecha de Compra: 11-09-2013 Forma Pago: cheque

Nombre Empresa: compute po

Numero de Cheque: 123456

Descuento: 0.0 Monto Total: 0.0

Impresora a tinta

Seleccione un Material de La lista

Lista de Materiales

ID	Nombre del Material	Categoria	Unidad de Medida	Descripción
1	resmas	material_escritorio	unidades	es de
2	Impresora canon	Impresoras	unidades	Impre
3	Impresora hp 3600	Impresoras	unidades	Impre
4	baneras	material_escritorio	piezas	es de
5	Silla	muebles	pieza	de m
6	Mesa	muebles	pieza	de m
7	Mesa Redonda	muebles	pieza	gran
8	Silla replegable	muebles	pieza	de m
9	Bañeras Grandes	Limpieza e higiene	pieza	klash
10	Bañeras Medianas	Limpieza e higiene	pieza	sjont

Página 1 de 2 Mostrando 1 a 10 de 16 elementos

Aceptar Cancelar

Fig.II.117 Pantalla-Modificar Compra

ADMINISTRAR PERSONA

Pantalla 13: Listar Persona

Realizar búsqueda

Todos Id Persona introduzca un valor

Lista de Personas

Registrar Persona Registrar Empleado Adicionar Usuario

- ✓ Id Persona
- ✓ Numero de Documento
- ✓ Nombre
- ✓ Apellido Paterno
- ✓ Apellido Materno

ID	Docu	Id Persona	Nombre	Ap Paterno	Ap Materno	Fecha Nac	El
1	ci	1234565 tja.	Maria	Perez	Lopez	1987-10-19	j_213_m@
2	run	345675 beni	Gabriela	Perez	Lopez	1987-10-19	j_213_m@
3	pasaporte	467964 tja	Jose	Rojas	Tapia	1987-10-19	j_213_m@
4	ci	6359824	raul	Ruiz	Meji	1982-02-20	luar_123@
5	ci	453786	Paola	Rueda	Castro	1987-10-19	j_213_m@
6	ci	6638075	denis	silisqui	aramayo	1987-07-19	c
7	ci	123123	eqweq	eqweqw	qweqwe	2011-10-18	sdf
8	run	31231	moises	hismac	karton	2011-10-20	ssdfd@h
9	pasaporte	122312	dasdas	asfasf	sfasfa	2011-10-11	sdfsdfsdf

Página 1 de 2 Mostrando 1 a 10 de 14 elementos

Fig.II.118 Pantalla-Listar Persona

Pantalla 14: Adicionar Persona

ADICIONAR PERSONA	
Tipo Documento:	ci Nuevo
Numero Documento:	
Nombre:	
Apellido Paterno:	
Apellido Materno:	
Fecha Nacimiento:	Calendario
Telefono:	
Email:	
Direccion:	
Fotografia:	Examinar...

Fig.II.119 Pantalla-Adicionar persona

Pantalla 15: Modificar Persona

MODIFICAR PERSONA	
Tipo Documento:	ci Nuevo
Numero Documento:	1234565 tja.
Nombre:	Maria
Apellido Paterno:	Perez
Apellido Materno:	Cruz
Fecha Nacimiento:	19-10-1987 Calendario
Telefono:	66-54952
Email:	j_213_m@hotmail.com
Direccion:	calle/beni
Fotografia:	Examinar...

Fig.II.120 Pantalla-Modificar Persona

ADMINISTRAR EMPLEADOS

Pantalla 16: Listar Empleados

Realizar búsqueda

Personas

Identificador

introduzca un valor

Lista de Empleados

Adicionar Empleado

Modificar Empleado

Adicionar Usuario

	ID	Profesion	Oficialia	Cargo de Trabajo	Documento	Nro Documento	Nombre	Ap Paterno	Ap Materno
1	8	-----	-----	-----	run	31231	moises	hismac	karton
2	9	-----	-----	-----	pasaporte	122312	dasdas	asfasf	sfasfa
3	10	-----	-----	-----	ci	2321314	eqwdd	sfasfas	asfasfasf
4	14	-----	-----	-----	run	qwweeeee	fdgss	sdfdg	fdggfgrf
5	15	-----	-----	-----	ci	7118838	juan	sanchez	cruz
6	17	-----	-----	-----	ci	66325648	gdsdf	sdgsdf	gdfsgdfg
7	18	-----	-----	-----	ci	23423523	asfasf	sfasdfsdf	sdfsdfsdf
8	19	-----	-----	-----	ci	53456346	asfasf	ytregerg	jhtyhrth
9	20	-----	-----	-----	ci	123124333	sdf	hfgf	kytygh
10	21	-----	-----	-----	ci	sadasda	asfasf	sdfsdf	sdfsdf

10

<< < > >>

Página 1 de 2

>> >

Mostrando 1 a 10 de 11 elementos

Fig.II.121 Pantalla-Listar Empleado

Pantalla 17: Adicionar Empleado

ADICIONAR EMPLEADO

Datos de Persona

eqwdd sfasfas asfasfasf

Profesion:

Abogado(a)

Nueva_Profesion

Oficialia:

Primera Oficialia

Cargo:

JEFE

Nuevo Cargo

Especialidad:

Derecho Penal

agregar

Nuevo

IdEspecialidad

Nombre_Especialidad

Fecha_Titulacion

asdas

quitar

Adicionar

Cancelar

Fig.II.122 Pantalla-Adicionar Empleado

Pantalla 18: Modificar Empleado

MODIFICAR EMPLEADO

Datos de Persona	Gabriela Perez Lopez		
Profesion:	Auditor(ora) ▼	Nueva_Profesion	
Oficialia:	Primera Oficialia ▼		
Cargo:	JEFE ▼	Nuevo Cargo	
Especialidad:	Derecho agrario ▼ agregar Nuevo		
	1 Derecho Penal 13-09-2011		
	asdas		
	quitar		
Modificar Cancelar			

Fig.II.123 Pantalla-Modificar Empleado

ADMINISTRAR USUARIOS

Pantalla 19: Listar Usuarios

Realizar búsqueda

Todos ▼ ✓ Identificador ▼

Lista de Usuarios

+ Adicionar Usuario ✎ Modificar Usuario

	ID	Usuario	Rol	Documento	Nro Documento	Nombre	Ap Paterno	Ap Materno	Fecha Nac	Email
1	1	drsa	administrador	ci	6638075	denis	sillisqui	aramayo	1987-07-19	denis.sillisqui@c
2	2	maria	administrador	ci	1234567	Maria de los		Castillo	1987-01-06	mari_m@hotm
3	3	maria23	Jefe	ci	1234565 tja	Maria	Perez	Lopez	1987-10-19	j_213_m@hotn
4	4	gaby	Secretaria(o)	run	345675 beni	Gabriela	Perez	Lopez	1987-10-19	j_213_m@hotn
5	5	user1	Jefe	ci	123123	eqweq	eqweqw	qweqwe	2011-10-18	sdfsdfs
6	6	simon	Encargado	ci	66359824	Simon	Droy	Fhreithr	1987-05-26	simoy@hotm
7	7	marcos	administrador	pasaporte	467964 tja	Jose	Rojas	Tapia	1987-10-19	j_213_m@hotn

10 ▼

 << < > >>

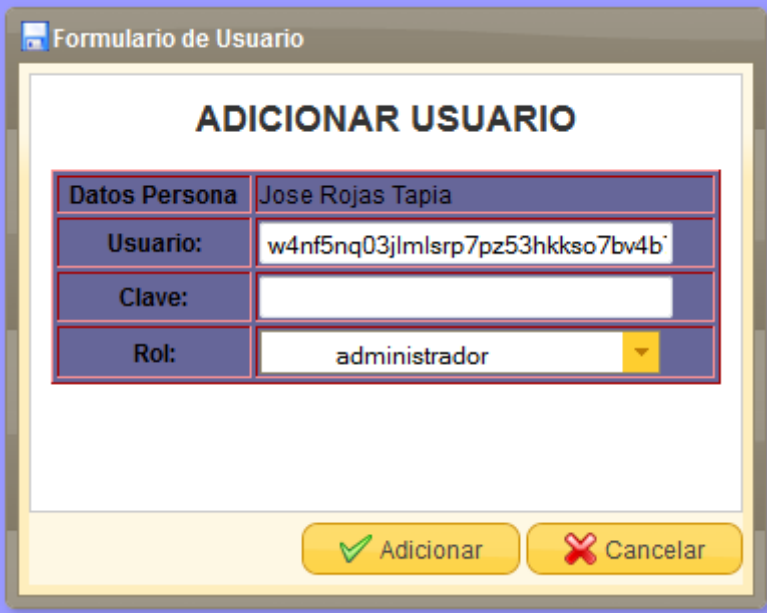
Página 1 de 1

 🔍

Mostrando 1 a 7 de 7 elementos

Fig.II.124 Pantalla-Listar Usuarios

Pantalla 20: Adicionar Usuario



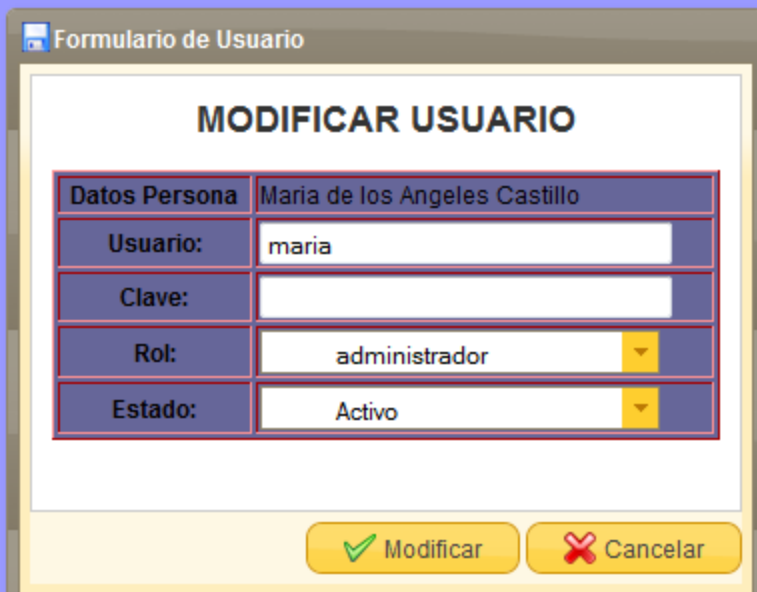
The screenshot shows a window titled "Formulario de Usuario" with a sub-header "ADICIONAR USUARIO". It contains a form with the following fields:

Datos Persona	Jose Rojas Tapia
Usuario:	w4nf5nq03jlmisrp7pz53hkso7bv4b
Clave:	
Rol:	administrador

At the bottom of the form are two buttons: "Adicionar" (with a green checkmark icon) and "Cancelar" (with a red X icon).

Fig.II.125 Pantalla-Adicionar Usuario

Pantalla 22: Modificar Usuario



The screenshot shows a window titled "Formulario de Usuario" with a sub-header "MODIFICAR USUARIO". It contains a form with the following fields:

Datos Persona	Maria de los Angeles Castillo
Usuario:	maria
Clave:	
Rol:	administrador
Estado:	Activo

At the bottom of the form are two buttons: "Modificar" (with a green checkmark icon) and "Cancelar" (with a red X icon).

Fig.II.126 Pantalla-Modificar Usuario

ADMINISTRAR PROVEEDOR

Pantalla23: Listar Proveedores

Realizar búsqueda

Todos ☐ Identificador

Lista de Proveedores

[+ Adicionar Proveedor](#) [✎ Modificar Proveedor](#)

	ID	Usuario	Documento	Nro Documento	Nombre	Ap Paterno	Ap Materno	Fecha Nac	Email
1	1	Muebleria	run	31231	moises	hismac	karton	2011-10-20	ssdfd@hotmail.com
2	2	empresa 2	ci	453786	Paola	Rueda	Castro	1987-10-19	j_213_m@hotmail.com
3	3	empresa 3	ci	4234	sdfs	ewefs	trdgeg	2011-04-12	sdfweq
4	4	digital&compute	ci	2321314	eqwdd	sfasfas	asfasfasf	1990-10-23	gsdgsdg@hotmail.com
5	6	Formas	ci	453786	Paola	Rueda	Castro	1987-10-19	j_213_m@hotmail.com

10 Página 1 de 1 Mostrando 1 a 5 de 5

Fig.II.127 Pantalla-Listar Proveedores

Pantalla 24: Adicionar Proveedor

Formulario de Proveedor

ADICIONAR PROVEEDOR

Nombre:	Ada
ApPaterno:	Gutierres
ApMaterno:	Gallardo
Empresa:	digital&computer
Adicionar Cancelar	

Fig.II.128 Pantalla-Adicionar Proveedor

Pantalla 25: Modificar Proveedor

Formulario de Proveedor

MODIFICAR PROVEEDOR

Nombre:	moises
ApPaterno:	hismac
ApMaterno:	karton
Empresa:	Muebleria "Lapacho" ▼

Modificar Cancelar

Fig.II.129 Pantalla-Modificar Proveedor

ADMINISTRAR EMPRESAS

Pantalla 26: Listar Empresa

Realizar búsqueda

Todos Cod Empresa introduzca un valor

Lista de Empresas

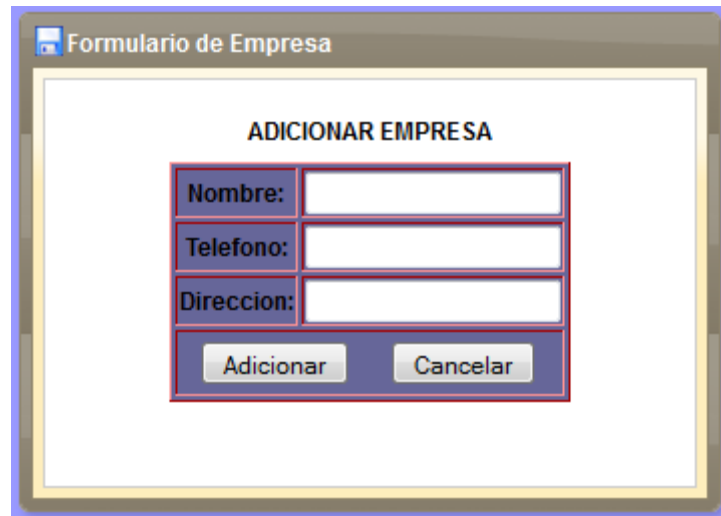
+ Adicionar Empresa ✎ Modificar Empresa

	ID	Nit	Nombre	Email	Telefono	Direccion	Estado
1	1	4353543	digital&compute	www.digital_compu	5-67833	av.la paz	Activo
2	2	5463574	muebleria_forta	www.digital_compu	5-67833	av. Domingo Paz	Activo
3	3	4534543	digitalpc	jdjf@hotmail.com	4543543	c/mendez	Activo
4	4	23535555	Formas	empresa_56@gma	66-52468	carretera el chaco	Activo
5	5	35415484	Muebleria	empresa_1@gmail	66-12481	avenida lapaz numero	Activo
6	6	65735454	empresa 2	empresa_2@gmail	66-45785	en un lugar	Activo
7	7	65754571	empresa 3	emresa_3@hotmail	66-89524	ksjhfsbjbjkb	Activo
8	8	233543	empresa4	empresa_89@gma	66-75123	avenida las americas	Activo
9	9	234234	Corbin	empresa_125@gm	34414	calle/colon	Activo

10 < > << >> Página 1 de 1 Mostrando 1 a 9 de 9 elementos

Fig.II.130 Pantalla-Listar Empresa

Pantalla 27: Adicionar Empresa



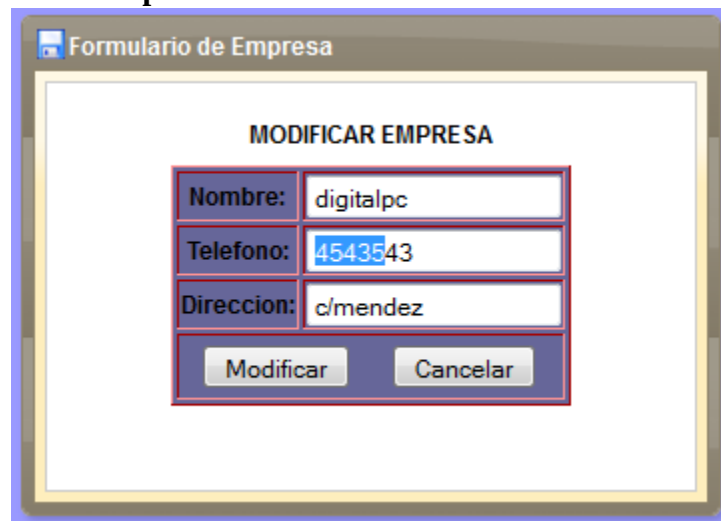
The screenshot shows a window titled "Formulario de Empresa" with a subtitle "ADICIONAR EMPRESA". It contains three input fields labeled "Nombre:", "Telefono:", and "Direccion:". Below these fields are two buttons: "Adicionar" and "Cancelar".

Field	Value
Nombre:	
Telefono:	
Direccion:	

Buttons: Adicionar, Cancelar

Fig.II.131 Pantalla-Adicionar Empresa

Pantalla 28: Modificar Empresa



The screenshot shows a window titled "Formulario de Empresa" with a subtitle "MODIFICAR EMPRESA". It contains three input fields labeled "Nombre:", "Telefono:", and "Direccion:". The fields are pre-filled with the values "digitalpc", "4543543", and "c/mendez" respectively. Below these fields are two buttons: "Modificar" and "Cancelar".

Field	Value
Nombre:	digitalpc
Telefono:	4543543
Direccion:	c/mendez

Buttons: Modificar, Cancelar

Fig.II.132 Pantalla-Modificar Empresa

ADMINISTRAR MATERIALES

Pantalla 29: Listar Materiales

Realizar búsqueda

Todos Código Material

Lista de Materiales

Nuevo Material Modificar Material

	☐	ID	Nombre del Material	Categoría	Unidad de Medida	Descripción del Material
1	☐	1	resmas	material_escritorio	unidades	es de 75 gramos
2	☐	2	impresora canon	impresoras	unidades	impresora a chorro
3	☐	3	impresora hp 3600	impresoras	unidades	impresora a chorro
4	☐	4	baneras	material_escritorio	piezas	es de 70 gramos
5	☐	5	Silla	muebles	pieza	de madera
6	☐	6	Mesa	muebles	pieza	grande
7	☐	7	Mesa Redonda	muebles	pieza	grande
8	☐	8	Silla replegable	muebles	pieza	de metal
9	☐	9	Bañeras Grandes	Limpieza e higiene	pieza	kjasfkajsfkbksc
10	☐	10	Bañeras Medianas	Limpieza e higiene	pieza	sjdhfsjkh

10 Página 1 de 2 Mostrando 1 a 10 de 16 elementos

Fig.II.133 Pantalla-Listar Material

Pantalla 30: Adicionar Materiales

Formulario de Material

ADICIONAR MATERIAL

Nombre:	
Unidad de Medida:	
Descripción:	
Categoría:	material_escritorio Nuevo
Adicionar Cancelar	

Fig.II.134 Pantalla-Adicionar Material

Pantalla 31: Modificar Materiales

Formulario de Material

MODIFICAR MATERIAL

Nombre: baneras

Unidad de Medida: piezas

Descripción: es de 70 gramos

Categoría: material_escritorio

Nuevo

Modificar

Cancelar

Fig.II.135 Pantalla-Modificar Material

ADMINISTRAR CATEGORÍAS

Pantalla 32: Listar Categorías

Realizar búsqueda

Todos

Código Categoría: introduzca un valor

Lista de Categorías

Nueva Categoría

Modificar Categoría

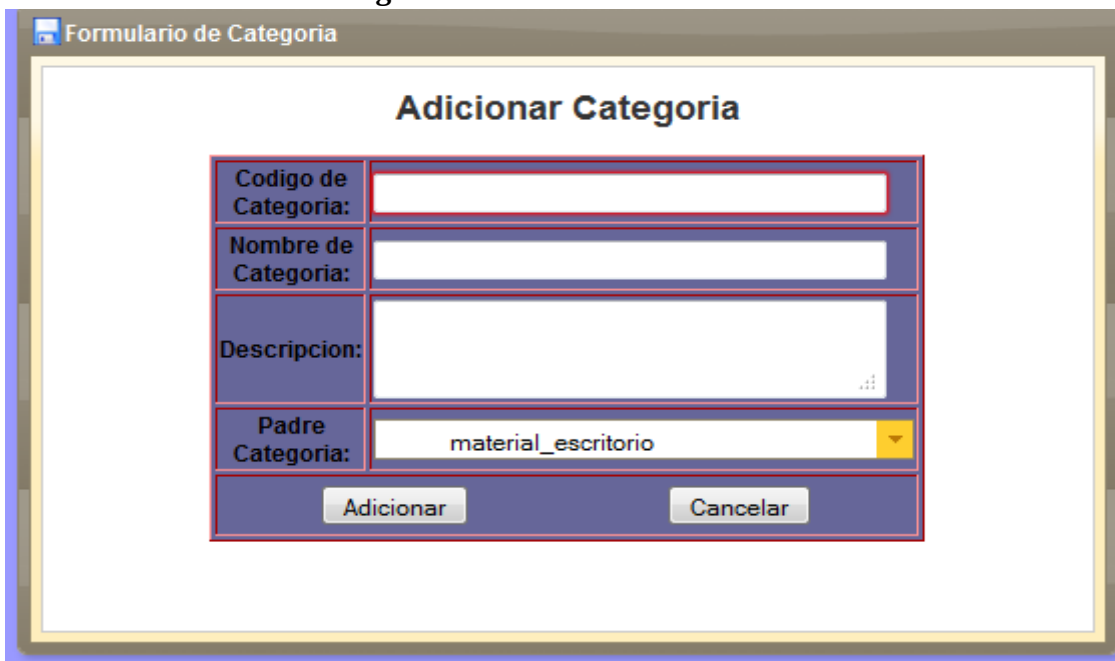
ID	Nombre de Categoría	Padre Categoría	Descripción de la Categoría
1	material_escritorio		
2	impresoras		
3	muebles		
4	Limpieza e higiene		
5	otros		
6	construccion		
7	categoriay		
8	plasticos		
9	Alimentos y productos Agrícolas		akjsdha aksjdhakjsf aksjfhassjf
10	Alimentos y Bebidas para Personas,Desay.	Alimentos y productos Agrícolas	

Página 1 de 5

Mostrando 1 a 10 de 40 elementos

Fig.II.136 Pantalla-Listar Categorías

Pantalla 33: Adicionar Categoría

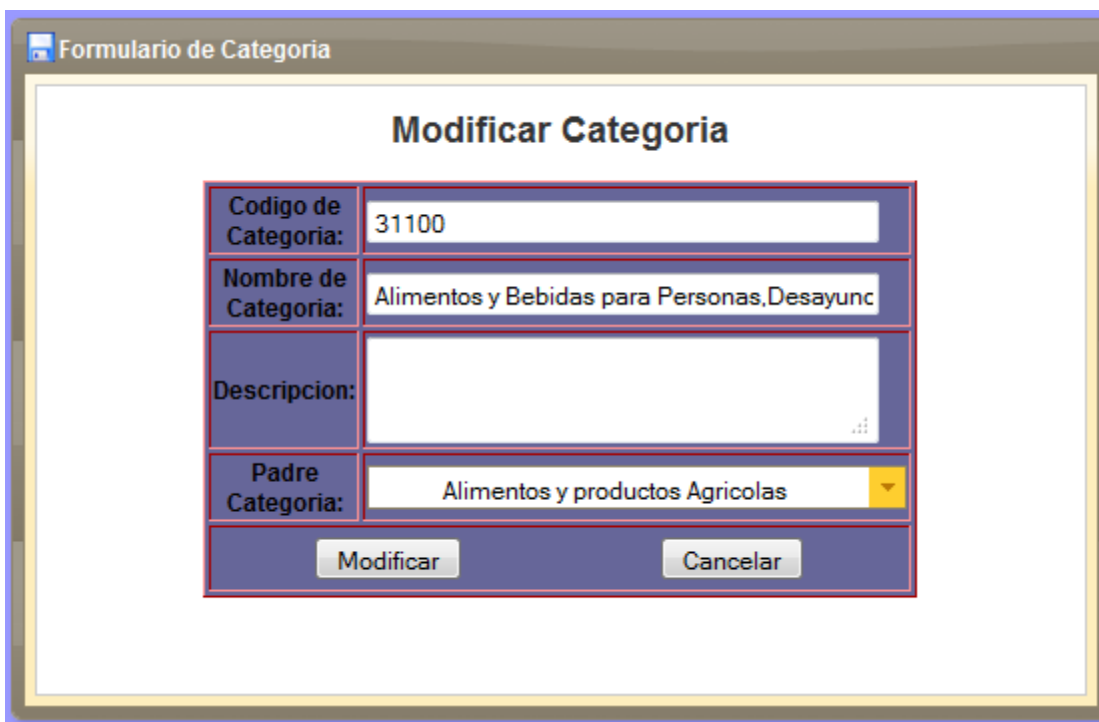


The screenshot shows a software window titled "Formulario de Categoría". Inside, there is a section titled "Adicionar Categoría". This section contains a form with four input fields: "Codigo de Categoría:" (empty), "Nombre de Categoría:" (empty), "Descripción:" (empty), and "Padre Categoría:" (containing "material_escritorio" with a dropdown arrow). At the bottom of the form are two buttons: "Adicionar" and "Cancelar".

Codigo de Categoría:	
Nombre de Categoría:	
Descripción:	
Padre Categoría:	material_escritorio
Adicionar Cancelar	

Fig.II.137 Pantalla-Adicionar Categoría

Pantalla34: Modificar Categoría



The screenshot shows a software window titled "Formulario de Categoría". Inside, there is a section titled "Modificar Categoría". This section contains a form with four input fields: "Codigo de Categoría:" (containing "31100"), "Nombre de Categoría:" (containing "Alimentos y Bebidas para Personas,Desayunc"), "Descripción:" (empty), and "Padre Categoría:" (containing "Alimentos y productos Agrícolas" with a dropdown arrow). At the bottom of the form are two buttons: "Modificar" and "Cancelar".

Codigo de Categoría:	31100
Nombre de Categoría:	Alimentos y Bebidas para Personas,Desayunc
Descripción:	
Padre Categoría:	Alimentos y productos Agrícolas
Modificar Cancelar	

Fig.II.138 Pantalla-Modificar Categoría

ADMINISTRAR OFICIALÍAS

Pantalla35: Listar Oficialías

Realizar búsqueda

Todos Código de Oficialia

Lista de Oficialías

	ID	Nombre de Oficialia	Decripcion
1	1	Primera Oficialia	Primer
2	2	Segunda Oficialia	Segundo
3	3	Tercera Oficialia	Tercer
4	4	Oficialia 3	adm
5	5	Sistemas	Encargados de la Administracion
6	6	Oficialia de	Encargados de los almacenes y

10 de 1

Fig.II.139 Pantalla-Listar Oficialias

Pantalla 36: Adicionar Oficialías

Formulario de Almacen

ADICIONAR OFICIALIA

Nombre:

Descripcion:

Fig.II.140 Pantalla-Adicionar Oficialia

Pantalla37: Modificar Oficialías



Formulario de Almacen

MODIFICAR OFICIALIA

Nombre: Tercera Oficialia

Descripcion: Tercer

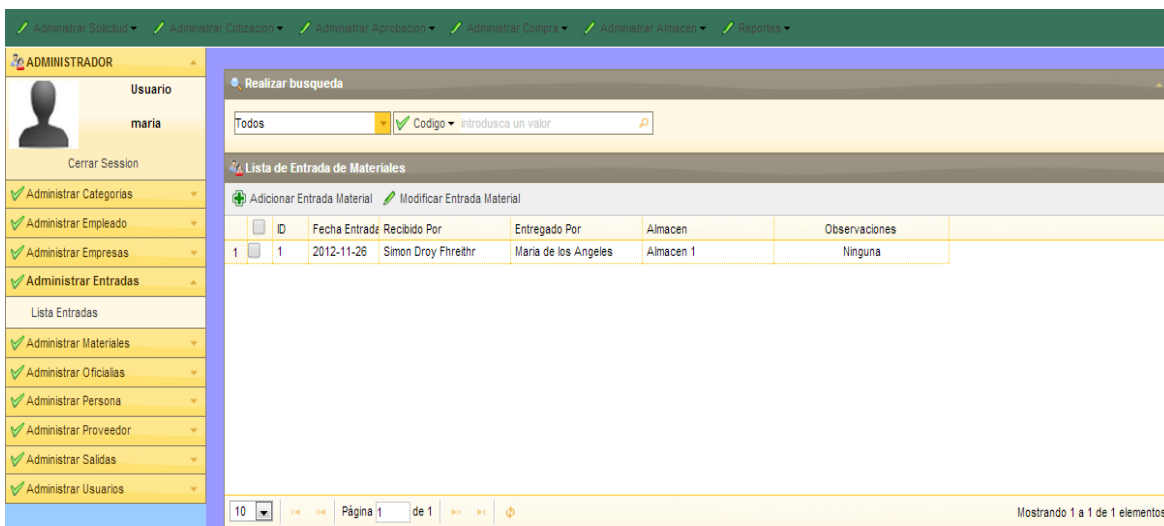
Modificar Cancelar

✓ Modificar ✗ Cancelar

Fig.II.141 Pantalla-Modificar Oficialia

ADMINISTRAR ALMACEN

Pantalla38: Administrar Entradas



Administrador

Usuario: maria

Cerrar Session

Realizar búsqueda

Todos Codigo introduzca un valor

Lista de Entrada de Materiales

Adicionar Entrada Material Modificar Entrada Material

ID	Fecha Entrada	Recibido Por	Entregado Por	Almacen	Observaciones
1	2012-11-28	Simon Droy Fhreithr	Maria de los Angeles	Almacen 1	Ninguna

Mostrando 1 a 1 de 1 elementos

Fig.II.142 Pantalla-Administrar Entradas

Pantalla39: Administrar Salidas

Realizar búsqueda

Todos introduzca un valor

Seleccione un Valor

- Todos
- Activos
- Cancelados

Modificar Salida Material

Unidad Solicitante	Estado	Entregado por	Fecha Salida	Solicitado por	Fecha Solicita	Autorizado por	Fecha Autoriz
nadie	Activo	Simon Droy Fhreithr	2012-11-26	Maria Perez Cruz	2012-11-26	Gabriela Perez Lopez	2012-11-26
centro infro	Activo	denis silisqui aramayo	2012-11-26	Maria de los Angeles	2012-11-26	Maria Perez Cruz	2012-11-26

10 Mostrando 1 a 2 de 2 elementos

Fig.II.143 Pantalla-Administrar Salidas

Pantalla 40: Lista Salida de Material

Realizar búsqueda

Todos introduzca un valor

Lista de Salida de Materiales

+ Adicionar Salida Material ✎ Modificar Salida Material

	ID	Nro Solicitud	Unidad Solicitante	Estado	Entregado por	Fecha Salida	Solicitado por	Fecha Solicita	Autorizado por
1	1	25	nadie	Activo	Simon Droy Fhreithr	2012-11-26	Maria Perez Cruz	2012-11-26	Gabriela Perez Lopez
2	2	34	centro infro	Activo	denis silisqui aramayo	2012-11-26	Maria de los Angeles	2012-11-26	Maria Perez Cruz

10 Mostrando 1 a 2 de 2 elementos

Fig.II.144 Pantalla-Lissta salida de material

Pantalla 41: Adicionar Salida de Material

Formulario de Salida de Materiales

ADICIONAR SALIDA DE MATERIA

Almacen:	Almacen 1
Fecha Salida:	25-05-2012
Entregado Por:	0
Fecha Entrega:	25-05-2012
Solicitado Por:	
Fecha Solicitud:	25-05-2012
Autorizado Por:	
Fecha Autorizado:	25-05-2012
Recibido Por:	
Fecha Recibir:	25-05-2012
Unidad Solicitante:	
Numero de Solicitud:	0
Estado:	Activo
Observaciones:	

Detalle De Los Materiales

COD Material	Material	Cantidad	Observacion

Fig.II.145 Pantalla-Adicionar Salida de Material

Pantalla 42: Modificar Salida de Material

Formulario de Salida de Materiales

MODIFICAR SALIDA DE MATERIA

Almacén:	Almacén 1
Fecha Salida:	25-11-2012
Entregado Por:	Alagüa Amayo Denis
Fecha Entrega:	25-11-2012
Solicitado Por:	Castillo Maria de los Angeles
Fecha Solicita:	25-11-2012
Autorizado Por:	Perez Cruz Maria
Fecha Autorizado:	25-11-2012
Recibido Por:	Castillo Maria de los Angeles
Fecha Recibe:	25-11-2012
Unidad Solicitante:	centro info
Numero de Solicitud:	34
Estado:	Activo
Observaciones:	adfa fed fed fed fedf

☒ Editar
☒ Quitar Material
☒ Agregar Detalle de Materiales a la Solicitud

Detalle De Los Materiales

COD Material	Material	Cantidad	Observacion
7	Mesa Redonda	2	Ninguna

Fig.II.146 Pantalla-Modificar Salida Material

Pantalla 43: Adicionar Salida de Material

Formulario de Salida de Materiales

ADICIONAR SALIDA DE MATERIA

Almacén:	Almacén 1
Fecha Salida:	25-05-2013
Entregado Por:	5
Fecha Entrega:	25-05-2013
Solicitado Por:	
Fecha Solicita:	25-05-2013
Autorizado Por:	
Fecha Autorizado:	25-05-2013
Recibido Por:	
Fecha Recibe:	25-05-2013
Unidad Solicitante:	
Número de Solicitud:	0
Estado:	Activo
Observaciones:	

Detalle De Los Materiales

CCC Material	Material	Cantidad	Observacion

Fig.II.147 Pantalla-Adicionar Salida Material

Pantalla 44: Reportes-Ver Reportes



Fig.II.148 Pantalla -Ver Reportes

REPORTES



Gobierno Municipal de San Lorenzo

PRIMERA SECCION - PROVINCIA MENDEZ
c./ Rodolfo Avila s/n - Edificio COSSET
TARIJA - BOLIVIA

Fecha : 11/09/2013

Hora : 1:42:09 PM

Pagina: 1 de 1

MEJORAMIENTO DE LA GESTION DEL AREA DE ARCHIVOS DE LA O.M.A.

REPORTE DE COMPRAS

Desde: 05/05/0031 Hasta: 05/05/0031

COMPRA					
COD COMPRA: 4		COD SOLICITUD: 15		FECHA SOLICITUD: 25/11/2012	
FECHA COMPRA: 25/11/2012		FORMA DE PAGO: cheque		DESCUENTO: 0.0	
NUMERO DE CHEQUE: 32654125		EMPRESA: no me			
DETALLE DE COMPRA					
CODIGO	MATERIAL	CANTIDAD	PRECIO	TOTAL	OBSERVACION
8	Silla replegable	6	20.0	120.0	Ninguna
6	Mesa	1	50.0	50.0	Ninguna
MONTO TOTAL				170.0	
OBSERVACION					
asas asdfa sfas fasfasf					

Fig.II.149 Pantalla-Reporte Compra



Gobierno Municipal de San Lorenzo

PRIMERA SECCION - PROVINCIA MENDEZ
c./ Rodolfo Avila s/n - Edificio COSSET
TARUA - BOLIVIA

Fecha : 11/09/2013
Hora : 12:23:42 PM
Pagina: 1 de 1

MEJORAMIENTO DE LA GESTION DEL AREA DE ARCHIVOS DE LA O.M.A.

SALIDA DE MATERIALES

Desde: 04/05/0032 Hasta: 04/05/0032

SALIDA DE ALMACENES				
CODIGO: 1	FECHA SALIDA: 26/11/2012	ALMACEN: Almacen 1		
UNIDAD SOLICITANTE: nadie		NRO. SOLICITUD: 25		
DETALLE DE MATERIALES				
CODIGO	MATERIAL	CANTIDAD	UNIDAD	OBSERVACION
6	Mesa	1	pieza	Ninguna
8	Silla replegable	8	pieza	Ninguna

	SOLICITADO POR	AUTORIZADO POR	ENTREGADO POR	RECIBIDO POR
RESPONSABLE	Maria Perez Cruz	Gabriela Perez Lopez	Droy Fhrelthr Simon	Jose Rojas Tapia
FECHA	26/11/2012	26/11/2012	26/11/2012	26/11/2012
OBSERVACION ninguna				

SALIDA DE ALMACENES				
CODIGO: 2	FECHA SALIDA: 26/11/2012	ALMACEN: Almacen 1		
UNIDAD SOLICITANTE: centro infro		NRO. SOLICITUD: 34		
DETALLE DE MATERIALES				
CODIGO	MATERIAL	CANTIDAD	UNIDAD	OBSERVACION
7	Mesa Redonda	3	pieza	Ninguna

	SOLICITADO POR	AUTORIZADO POR	ENTREGADO POR	RECIBIDO POR
RESPONSABLE	Maria de los Angeles	Maria Perez Cruz	silisqui aramayo denis	Maria de los Angeles
FECHA	26/11/2012	26/11/2012	26/11/2012	26/11/2012
OBSERVACION sdfs fsd fsd fsd fsdf				

Fig.II.150 Pantalla-Salidas de Almacén

II.2.14. MODELO DE DATOS

Introducción

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

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

Propósito

Comprende la estructura del sistema deseado para la Organización.

Identificar posibles mejoras.

Alcance

Describir las tablas de diseño del sistema en su segunda iteración.

Identificar y definir las **relaciones entre tablas** según los **objetivos** del sistema deseado aprobado por la Organización.

II.2.14. 1. ESPECIFICACIÓN DE TABLAS DE LA BASE DE DATOS

Tabla persona

Create table persona (

id_persona serial not null primary key ,

nombre varchar (50),

ap_paternovarchar (50),

ap_materno varchar (50),

teléfono varchar (30),

direccion varchar (60),

email varchar (50));

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_persona</u>	serial	no	Si	Auto Incremental	Identificador de la Tabla.
Nombre	<u>varchar(50)</u>	si	No		Nombre de la Persona.
<u>ap_paterno</u>	<u>varchar(50)</u>	si	No		Apellido Paterno de la Persona.
<u>Am materno</u>	<u>varchar(50)</u>	si	No		Apellido Materno de la Persona.
<u>telefono</u>	<u>varchar(30)</u>	si	No		Teléfono de la persona
<u>direccion</u>	<u>varchar(60)</u>	si	No		Dirección de la persona
email	<u>varchar(50)</u>	si	No		Email de la persona

Tabla Datos1. Datos Simples de la Tabla Persona

Tabla Empleado

```
create table empleado (
id_empleado serial not null primary key ,
profesion varchar (60),
estado integer default 1,
id_persona integer,
foreignkey (id_persona) references persona(id_persona)
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id empleado</u>	serial	no	si	Auto Incremental	Identificador de la Tabla.
<u>profesion</u>	<u>varchar(60)</u>	si	no		Profesión del empleado
<u>estado</u>	<u>int</u>	si	no		por defecto 1

Datos Simples de la Tabla Empleado

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>id_persona</u>	<u>int</u>	si	no	Persona	<u>id_persona</u>	Identificador de la tabla persona

Tabla Datos2.Claves Foráneas de la Tabla Empleado

Tabla especialidad

```
Create table especialidad(  
id_especialidad serial not null primary key ,  
nombre_especialidad varchar (60),  
descripcion varchar (60)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_especialidad</u>	serial	no	Si	Auto Incremental	Identificador de la Tabla.
<u>nombre_especialidad</u>	<u>varchar(60)</u>	si	No		Nombre de la especialidad
<u>descripcion</u>	<u>varchar(60)</u>	si	No		Descripción de la especialidad

Tabla datos 3.Datos Simples de la Tabla Especialidad

Tabla especialidad_empleado

```
create table especialidad_emp(  
id_empleado integer not null,  
id_especialidad integernotnull,  
fecha_titulo date,  
foreignkey (id_especialidad) references especialidad(id_especialidad),  
foreign key (id_empleado) references empleado(id_empleado),  
primary key(id_empleado,id_especialidad)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_empleado</u>	<u>Int</u>	no	si	Introducir datos	Identificador de la tabla
<u>id_especialidad</u>	<u>Int</u>	no	si	Introducir datos	Identificador de la tabla
<u>fecha_titulo</u>	Date	si	no		Fecha de la titulación del empleado

Tabla datos 4. Datos Simples de la Tabla Especialidad_empleado

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>id_empleado</u>	Int	no	si	Empleado	<u>id_persona</u>	Identificador de la tabla empleado
<u>id_especialidad</u>	int	no	si	especialidad		Identificador de la tabla especialidad

Tabla datos 5. Claves Foráneas de la Tabla Especialidad_empleado

Tabla rol

```
create table rol(
id_rol serial not null primary key ,
nombre varchar (60),
estado integer default 1
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_rol</u>	Serial	no	si	Auto Incremental	Identificador de la Tabla.
nombre	<u>varchar(60)</u>	si	si		Nombre del rol
estado	<u>Int</u>	no	no		Por defecto 1

Tabla datos 6. Datos Simples de la Tabla Rol

Tabla usuario

```
Create table usuario(
id_usuario serial not null primary key ,
nombre_usu varchar (60),
clave varchar (60),
estado integer default 1,
id_persona integer,
id_rol integer,
foreign key (id_persona) references persona(id_persona),
foreign key (id_rol) references rol(id_rol));
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_usuario</u>	Serial	no	si	Auto Incremental	Identificador de la Tabla.
<u>nombre_usu</u>	<u>varchar(60)</u>	si	no		Nombre de la Persona.
<u>clave</u>	<u>varchar(60)</u>	si	no		Apellido Paterno de la Persona.
<u>estado</u>	<u>Int</u>	si	no		Por defecto 1

Tabla de datos 7. Datos Simples de la Tabla Usuario

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>id_persona</u>	<u>int</u>	si	no	Persona	<u>id_persona</u>	Identificador de la tabla persona
<u>id_rol</u>	<u>int</u>	si	no	Rol	<u>id_rol</u>	Identificador de la tabla rol

Tabla de datos 8. Claves Foráneas de la Tabla Usuario

Tabla proceso

```

createtable proceso(
id_proceso serial not null primary key ,
nombre varchar (60),
estado integer default 1
);

```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_proceso</u>	serial	no	si	Auto Incremental	Identificador de la tabla
nombre	<u>varchar(60)</u>	si	no		Nombre del proceso
estado	<u>int</u>	si	no		Por defecto 1

Tabla de datos 9. Datos Simples de la Tabla Proceso

Tabla rol_pro

```
create table rol_pro(
id_rol integer not null,
id_proceso integer not null,
estado integer default 1,
foreign key (id_rol) references rol(id_rol),
foreign key (id_proceso) references proceso(id_proceso),
primary key(id_rol,id_proceso)
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_rol</u>	<u>Int</u>	no	si	Introducir datos	Identificador de la Tabla.
<u>id_proceso</u>	<u>Int</u>	no	si	Introducir datos	Identificador de la Tabla.
estado	<u>Int</u>	si	no		Por defecto 1

Tabla de datos 10. Datos Simples de la Tabla Rol_pro

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>id_rol</u>	<u>int</u>	no	si	Persona	<u>id_rol</u>	Identificador de la tabla rol
<u>id_proceso</u>	<u>int</u>	no	si	Rol	<u>id_proceso</u>	Identificador de la tabla proceso

Tabla de datos 11. Claves Foráneas de la Tabla Rol_pro

Tabla categoria

```
create table categoria(  
cod_categoria serial primary key ,  
nombrevarchar (60)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_categoria</u>	serial	No	Si	Auto Incremental	Identificador de la tabla
nombre	<u>varchar(60)</u>	Si	No		Nombre de la <u>categoria</u>

Tabla de datos 12. Datos Simples de la Tabla Categoria

Tabla material

```
create table material(  
cod_material serial not null primary key ,  
nombre varchar (50),  
unidad_medida varchar (50),  
precio_unitario integer,  
descripcion varchar (60),  
cod_categoria integer,  
foreign key (cod_categoria) references categoria(cod_categoria)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_material</u>	serial	no	si	Auto Incremental	Identificador de la Tabla.
Nombre	<u>varchar(50)</u>	si	no		Nombre del material
<u>unidad_medida</u>	<u>varchar(50)</u>	si	no		Catidad del material
<u>precio_unitario</u>	<u>int</u>	si	no		Precio unitario del <u>materia</u>
<u>descripcion</u>	<u>varchar(60)</u>	si	no		<u>Descripcion</u> del material

Tabla de datos 13. Datos Simples de la Tabla Material

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_categoria</u>	<u>int</u>	si	no	<u>categoria</u>	<u>cod_categoria</u>	Identificador de la tabla <u>categoria</u>

Tabla datos 14. Claves Foráneas de la Tabla Material

Tabla solicitud

```
create table solicitud(
cod_solicitud serial not null primary key,
fecha_solicitud date,
descripcion varchar (60),
id_empleado integer, --es el empleado q realiza la solicitud
foreign key (id_empleado) references empleado(id_empleado)
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_solicitud</u>	Serial	no	si	Auto Incremental	Identificador de la Tabla.
<u>fecha_solicitud</u>	Date	si	no		Nombre del material
<u>descripcion</u>	<u>varchar(60)</u>	si	no		<u>Descripcion</u> del material

Tabla datos 15. Datos Simples de la Tabla Solicitud

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>id_empleado</u>	<u>int</u>	si	no	empleado	<u>id_empleado</u>	Identificador de la tabla <u>empleado</u>

Tabla datos 16. Claves Foráneas de la Tabla Solicitud

Tabla detalle_solicitud

```
createtabledetalle_solicitud(  
cod_detalle serial not null primary key,  
cantidadfloat,  
observacionvarchar (60),  
cod_materialinteger,  
cod_solicitud integer,  
foreign key (cod_material) references material(cod_material),  
foreign key (cod_solicitud) references solicitud(cod_solicitud)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_detalle</u>	Serial	no	si	Auto Incremental	Identificador de la Tabla.
cantidad	<u>Float</u>	si	no		Cantidad detallada del material en la solicitud
<u>observacion</u>	<u>varchar(60)</u>	si	no		Observación de algún material que se detalla en la solicitud

Tabla datos 17. Datos Simples de la Tabla Detalle_solicitud

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_material</u>	<u>int</u>	No	no	material	<u>cod_material</u>	Identificador de la tabla material
<u>cod_solicitud</u>	<u>int</u>	no	no	solicitud	<u>cod_solicitud</u>	Identificador de la tabla solicitud

Tabla datos 18. Claves Foráneas de la Tabla Detalle_solicitud

Tabla aprobación

```
create table aprobacion(  
cod_aprobacion serial not null primary key,  
fecha_llegada date,  
fecha_aprobacion date,  
resultado_aprob integer, -- 1 significa aprobado , cero sig sin leer y -1 reprobado  
cod_solicitud integer,  
id_empleado integer,  
foreign key (cod_solicitud) references solicitud(cod_solicitud),  
foreign key (id_empleado) references empleado(id_empleado)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_aprobacion</u>	Serial	no	si	Auto Incremental	Identificador de la Tabla.
<u>fecha_llegada</u>	Date	si	no		Fecha en la que llega la solicitud
<u>fecha_aprobacion</u>	Date	si	no		Fecha en la que se aprueba la solicitud
<u>resultado_aprob</u>	Int	si	no		1 significa aprobado. 0 sin leer. -1 reprobado.

Datos datos 19. Datos Simples de la Tabla Aprobacion

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_solicitud</u>	<u>int</u>	no	no	solicitud	<u>cod_material</u>	Identificador de la tabla solicitud
<u>id_empleado</u>	<u>int</u>	no	no	empleado	<u>cod_solicitud</u>	Identificador de la tabla empleado

Tabla Datos 20. Claves Foráneas de la Tabla Aprobacion

Tabla material_aprobacion

```
create table material_aprobacion(  
cantidad float,  
observacion varchar (60),  
cod_material integer,      -- es el empleado podra modificar la cantidad y  
observaciones del material  
cod_aprobacion integer,  
foreignkey (cod_material) references material(cod_material),  
foreignkey (cod_aprobacion) referencesaprobacion(cod_aprobacion),  
primarykey(cod_material,cod_aprobacion)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
cantidad	<u>float</u>	si	no		Cantidad del material aprobado
<u>observacion</u>	<u>varchar(60)</u>	si	no		Observación que se detalla en la <u>aprobacion</u>
<u>cod_material</u>	<u>int</u>	no	si	Ingresar datos	Código del material
<u>cod_aprobacion</u>	<u>int</u>	no	si	Ingresar datos	Código de la <u>aprobacion</u>

Tabla datos 21. Datos Simples de la Tabla Material_aprobacion

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_material</u>	<u>int</u>	no	si	material	<u>cod_material</u>	Identificador de la tabla material
<u>cod_aprobacion</u>	<u>int</u>	no	si	<u>aprobacion</u>	<u>cod_aprobacion</u>	Identificador de la tabla <u>aprobacion</u>

Tabla datos 22. Claves Foráneas de la Tabla Material_aprobacion

Tabla empresa

```
createtable empresa(  
cod_empresa serial not null primary key,  
nombrevarchar (50),  
direccionvarchar (50),  
telefonovarchar (50),  
emailvarchar (60)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_empresa</u>	serial	no	si	Auto Incremental	Identificador de la Tabla.
Nombre	<u>varchar(50)</u>	si	no		Nombre de la empresa
Dirección	<u>varchar(50)</u>	si	no		Dirección de la empresa
Teléfono	<u>varchar(50)</u>	si	no		Teléfono de la empresa
Email	<u>varchar(60)</u>	si	no		Email de la <u>emprea</u>

Tabla datos 23. Datos Simples de la Tabla Empresa

Tabla compra

```
Create table compra (  
cod_compra serial not null primary key,  
cod_cotizacioninteger,  
fecha_compra date,  
forma_pagovarchar (50),  
descuento float,  
totalfloat,  
observacionvarchar (60),  
foreignkey (cod_cotizacion) referencescotizacion(cod_cotizacion)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_compra</u>	Serial	no	si	Auto Incremental	Identificador de la Tabla.
<u>fecha_compra</u>	Date	si	no		Fecha de la compra
<u>forma_pago</u>	<u>varchar(50)</u>	si	no		Forma de pago
<u>Descuento</u>	<u>Float</u>	si	no		Descuento en la compra
<u>Total</u>	<u>Float</u>	si	no		Monto total de la compra
<u>Observación</u>	<u>varchar(60)</u>	si	no		Alguna observación de la compra

Tabla datos 23. Datos Simples de la Tabla Compra

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_cotizacion</u>	<u>int</u>	No	no	<u>cotizacion</u>	<u>cod_cotizacion</u>	Identificador de la tabla cotización

Tabla datos 24. Claves Foráneas de la Tabla Compra

Tabla tipo_cot

```

createtabletipo_cot(
cod_tipo_cot serial not null primary key,
nombre_cotvarchar (50),
descripcionvarchar(60),
estadointeger default 1
);

```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_tipo_cot</u>	Serial	No	si	Auto Incremental	Identificador de la Tabla.
<u>nombre</u>	<u>varchar(50)</u>	Si	no		Nombre del tipo de cotización
<u>descripcion</u>	<u>varchar(60)</u>	Si	no		Descripción del tipo de cotización

Tabla datos 25. Datos Simples de la Tabla Tipo_cot

Tabla cotización

```
Create table cotizacion(  
cod_cotizacion serial not null primary key,  
monto_total float,  
cod_solicitud integer,  
cod_empresa integer not null,  
cod_compra integer not null,  
cod_tipo_cot integer not null,  
foreign key (cod_solicitud) references solicitud(cod_solicitud),  
foreign key (cod_empresa) references empresa(cod_empresa),  
foreign key (cod_compra) references compra(cod_compra),  
foreign key (cod_tipo_cot) references tipo_cot(cod_tipo_cot)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_cotizacion</u>	serial	no	si	Auto Incremental	Identificador de la Tabla.
<u>monto_total</u>	float	no	no		Monto de la cotización

Tabla datos 26. Datos Simples de la Tabla Cotización

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_solicitud</u>	int	si	no	solicitud	<u>cod_solicitud</u>	Identificador de la tabla solicitud
<u>cod_empresa</u>	int	no	no	empresa	<u>cod_empresa</u>	Identificador de la tabla empresa
<u>cod_compra</u>	int	no	no	compra	<u>cod_compra</u>	Identificador de la tabla compra
<u>cod_tipo_cot</u>	int	no	no	tipo_cot	<u>cod_tipo_cot</u>	Identificador de la tabla tipo_cot

Tabla datos 27. Claves Foráneas de la Tabla Cotización

Tabla detalle_cotizacion

```

create table detalle_cotizacion(
cod_detalle_cotizacion integer NOT NULL primary key,
cod_material integer NOT NULL,
precio_material double precision NOT NULL,
cod_empresa integer NOT NULL,
cantidad integer NOT NULL,
cod_cotizacion integer NOT NULL,
foreign key (cod_empresa) references empresa(cod_empresa),
foreign key (cod_material) references material (cod_material),
foreign key (cod_cotizacion) references cotizacion(cod_cotizacion),
);

```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_detalle_cotizacion</u>	<u>int</u>	no	no	Introducir datos	Identificador de la Tabla.
<u>precio_material</u>	<u>double</u>	no	no		Precio del material
<u>cantidad</u>	<u>int</u>	no	no		Cantidad del detalle de la cotización

Tabla datos 28. Datos Simples de la Tabla Detalle_cotizacion

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_material</u>	<u>int</u>	no	no	material	<u>cod_material</u>	Identificador de la tabla material
<u>cod_empresa</u>	<u>int</u>	no	no	empresa	<u>cod_empresa</u>	Identificador de la tabla empresa
<u>cod_cotizacion</u>	<u>int</u>	no	no	<u>cotizacion</u>	<u>cod_cotizacion</u>	Identificador de la tabla cotización

Tabla datos 29. Claves Foráneas de la Tabla Detalle_cotizacion

Tabla presupuesto

```
Create table presupuesto(  
id_presupuesto serial not null primary key,  
cantidad_presupuesto integer,  
estado integer default 1,  
cod_compra integer not null,  
foreign key (cod_compra) references compra(cod_compra)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_presupuesto</u>	serial	no	si	Auto Incremental	Identificador de la Tabla.
<u>cantidad_presupuesto</u>	<u>int</u>	si	no		Cantidad del presupuesto
<u>estado</u>	<u>int</u>	si	no		Por defecto 1

Tabla datos 30. Datos Simples de la Tabla Presupuesto

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_compra</u>	<u>int</u>	no	No	compra	<u>cod_compra</u>	Identificador de la tabla compra

Tabla datos 31. Claves Foráneas de la Tabla Presupuesto

Tabla cheque

```
create table cheque(  
num_cheque serial not null primary key,  
cod_compra integer not null,  
foreign key (cod_compra) references compra(cod_compra)  
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>Num_cheque</u>	Serial	no	si	Auto Incremental	Identificador de la Tabla.

Tabla datos 32. Datos Simples de la Tabla Cheque

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_compra</u>	int	No	no	compra	<u>cod_compra</u>	Identificador de la tabla compra

Tabla datos 33. Claves Foráneas de la Tabla Cheque

Tabla banco

```
create table banco(
cod_banco serial not null primary key,
nombre varchar (50),
direccion varchar (50),
telefono varchar (30),
num_cheque integer,
foreign key (num_cheque) references cheque(num_cheque)
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_banco</u>	serial	no	si	Auto Incremental	Identificador de la Tabla.
nombre	<u>varchar(50)</u>	si	no		Nombre del banco
<u>direccion</u>	<u>varchar(50)</u>	si	no		Dirección del banco
<u>telefono</u>	<u>varchar(30)</u>	si	no		Teléfono del banco

Tabla datos 34. Datos Simples de la Tabla Banco

Tabla oficialia

```
createtableoficialia(
cod_oficialia serial not null primary key,
nombre varchar(50),
descripcion varchar (60),
cod_tipo_oficialia integer,
```

```
foreignkey (cod_tipo_oficialia) references tipo_oficialia(cod_tipo_oficialia)
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_oficialia</u>	serial	no	si	Auto Incremental	Identificador de la Tabla.
nombre	<u>varchar(50)</u>	si	no		Nombre de la oficialia
<u>descripcion</u>	<u>varchar(60)</u>	si	no		Descripción del tipo de oficialia

Tabla datos 35. Datos Simples de la Tabla Oficialia

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_tipo_oficialia</u>	<u>int</u>	si	no	<u>tipo_oficialia</u>	<u>cod_tipo_oficialia</u>	Identificador de la tabla <u>tipo_oficialia</u>

Tabla datos 36. Claves Foráneas de la Tabla Oficialía

Tabla tipo_oficialia

```
createtabletipo_oficialia(
cod_tipo_oficialia serial notnullprimarykey,
nombrevarchar(50),
estadointeger default 1,
descripcionvarchar (60)
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_tipo_oficialia</u>	Serial	no	si	Auto Incremental	Identificador de la Tabla.
nombre	<u>varchar(50)</u>	si	no		Nombre del tipo de oficialia
estado	<u>int</u>	no	no		Por defecto 1
<u>descripcion</u>	<u>varchar(60)</u>	si	no		Descripción del tipo de oficialia

Tabla datos 37. Datos Simples de la Tabla Tipo_oficialia

Tabla oficialía_empleado

```

createtableoficialia_empleado(
    fecha_ingreso date,
    fecha_salida date,
    id_empleado integer,
    cod_oficialia integer,
    cod_cargo_trabajo integer,
    foreignkey (id_empleado) references empleado (id_empleado),
    foreignkey (cod_cargo_trabajo) references cargo_trabajo (cod_cargo_trabajo),
    foreignkey (cod_oficialia) references oficialia (cod_oficialia),
    primarykey(id_empleado,cod_oficialia)
);

```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>fecha_ingreso</u>	date	si	no	Auto Incremental	Fecha ingreso del empleado
<u>fecha_salida</u>	date	si	no		Fecha de salida del empleado
<u>id_empleado</u>	int	no	no		Identificador de la Tabla.
<u>cod_oficialia</u>	int	no	no		Identificador de la Tabla.

Tabla datos 38. Datos Simples de la Tabla Oficialia_empleado

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>Id_empleado</u>	int	no	no	empleado	<u>cod_material</u>	Identificador de la tabla material
<u>cod_cargo_trabajo</u>	int	no	no	<u>cargo_trabajo</u>	<u>cod_cargo_trabajo</u>	Identificador de la tabla empresa
<u>cod_oficialia</u>	int	no	no	<u>oficialia</u>	<u>cod_oficialia</u>	Identificador de la tabla cotizacion

Tabla datos 39. Claves Foráneas de la Tabla Oficialia_empleado

Tabla cargo_trabajo

```

create table cargo_trabajo(
cod_cargo_trabajo serial NOT NULL primarykey,
nombrecharactervarying(50),
descripcioncharactervarying(100),
estadointeger DEFAULT 1
);

```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_cargo_trabajo</u>	Serial	no	si	Auto Incremental	Identificador de la tabla
nombre	<u>varchar(50)</u>	si	no		Nombre del cargo trabajo
<u>descripcion</u>	<u>varchar(1000)</u>	no	no		Descripcion del cargo trabajo
estado	<u>Int</u>	no	no		Por defecto 1

Tabla datos 40. Datos Simples de la Tabla Cargo_trabajo

Tabla proveedor

```

create table proveedor(
id_proveedor serial NOT NULL primary key,
id_persona integer NOT NULL,
cod_empresa integer NOT NULL,
foreign key (id_persona) references persona (id_persona),
foreign key (cod_empresa) references empresa (cod_empresa)
);

```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>id_proveedor</u>	Serial	no	si	Auto Incremental	Identificador de la tabla

Tabla datos 41. Datos Simples de la Tabla Proveedor

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>id_persona</u>	int	No	no	persona	<u>id_persona</u>	Identificador de la tabla persona
<u>cod_empresa</u>	int	No	no	empresa	<u>cod_empresa</u>	Identificador de la tabla empresa

Tabla datos 42. Claves Foráneas de la Tabla Proveedor

Tabla Almacen

```
create table almacen(
cod_almacen integer NOT NULL primary key,
nombre_almacen varchar (200) NOT NULL,
direccion text,
telefono character varying(30),
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_almacen</u>	Integer	no	si	Auto Incremental	Identificador de la tabla
<u>nombre_almacen</u>	varchar(200)	no	no		Nombre del <u>almacen</u>
<u>direccion</u>	Text	si	no		Dirección del <u>almacen</u>
<u>telefono</u>	varchar(30)	si	no		Teléfono del <u>almacen</u>

Tabla datos 43. Datos Simples de la Tabla Almacén

Tabla almacen_material

```
create table almacen_material (
cod_almacen integer NOT NULL
cod_material integer NOT NULL,
cantidad integer NOT NULL,
```

```

estado boolean DEFAULT true NOT NULL,
material_bueno integer DEFAULT 0 NOT NULL,
material_regular integer DEFAULT 0 NOT NULL,
material_deshuso integer DEFAULT 0 NOT NULL
foreign key (cod_material) references material (cod_material),
foreign key (cod_almacen) references almacen (cod_almacen)
primarykey(cod_material,cod_almacen)
);

```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_almacen</u>	<u>Int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>cod_material</u>	<u>Int</u>	no	si	Ingresar datos	Identificador de la tabla
cantidad	<u>Int</u>	si	no		Cantidad del material
estado	<u>Boolean</u>	no	no	true	<u>true</u> : Habilitado. <u>false</u> :Deshabilitado.
<u>material_bueno</u>	<u>Int</u>	no	no		Por defecto 0
<u>material_regular</u>	<u>Int</u>	no	no		Por defecto 0
<u>material_deshuso</u>	<u>Int</u>	no	no		Por defecto 0

Tabla datos 44. Datos Simples de la Tabla Almacen_material

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>Cod_material</u>	<u>int</u>	No	no	material	<u>Cod_material</u>	Identificador de la tabla material
<u>Cod_almacen</u>	<u>int</u>	No	no	<u>almacen</u>	<u>Cod_almacen</u>	Identificador de la tabla almacén

Tabla datos 45. Claves Foráneas de la Almacen_material

Tabla empresa_material

```

create table empresa_material (
cod_empresa integer NOT NULL,

```

```

cod_material integer NOT NULL,
precio_ref double precision NOT NULL
foreign key (cod_material) references material (cod_material),
foreign key (cod_empresa) references empresa (cod_empresa)
primarykey(cod_material,cod_empresa)
);

```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_empresa</u>	<u>Int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>cod_material</u>	<u>Int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>precio_ref</u>	Doblé	no	no		Precio referencial

Tabla datos 46. Datos Simples de la Tabla Empresa_material

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>Cod_material</u>	<u>int</u>	No	no	material	<u>Cod_material</u>	Identificador de la tabla material
<u>Cod_empresa</u>	<u>int</u>	No	no	<u>almacen</u>	<u>Cod_empresa</u>	Identificador de la tabla empresa

Tabla datos 47. Claves Foráneas de la Empresa_material

Tabla Encargado _alamcen

```

CREATE TABLE encargado_almacen (
id_empleado integer NOT NULL,
cod_almacen integer NOT NULL,
fecha_entrada date NOT NULL,
fecha_salida date,
estado integer NOT NULL
foreign key (id_empleado) references empleado (id_empleado),
foreign key (cod_almacen) references almacen (cod_almacen)
primarykey(cod_empleado,cod_almacen)

```

);

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>Id_ empleado</u>	<u>Int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>cod_ almacen</u>	<u>Int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>Fecha_ entrada</u>	Date	no	no		Fecha de entrada
<u>Fecha_ salida</u>	Date	si	no		Fecha de salida
<u>Estado</u>	<u>Int</u>	no	no		

Tabla datos 48. Datos Simples de la Tabla Encargado_almacen

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>Id_ empleado</u>	<u>int</u>	No	si	<u>empleado</u>	<u>Id_ empleado</u>	Identificador de la tabla empleado
<u>cod_ almacen</u>	<u>int</u>	No	si	<u>almacen</u>	<u>cod_ almacen</u>	Identificador de la tabla <u>almacen</u>

Tabla datos 49. Claves Foráneas de la Encargado_almacen

Tabla Entrada_material

```
CREATE TABLE entrada_material (  
cod_entradamat integer NOT NULL primary key,  
fecha_entrada timestamp without time zone NOT NULL,  
id_empleado_entrega integer NOT NULL,  
observaciones text,  
cod_almacen integer NOT NULL  
foreign key (id_empleado) references empleado (id_empleado),  
foreign key (cod_almacen) references almacen (cod_almacen));
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>Cod entradamat</u>	<u>Int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>Fecha entrada</u>	Times	no	no		Fecha de entrada
<u>observaciones</u>	<u>Text</u>	si	no		Observaciones de la entrada material

Tabla datos 50. Datos Simples de la Tabla Entrada_material

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>Id empleado</u>	<u>int</u>	No	si	<u>empleado</u>	<u>Id empleado</u>	Identificador de la tabla empleado
<u>cod almacen</u>	<u>int</u>	No	si	<u>almacen</u>	<u>cod almacen</u>	Identificador de la tabla almacen

Tabla datos 51. Claves Foráneas de la Entrada_material

Tabla Salida_material

```
CREATE TABLE salida_material (
cod_salida_mater integer NOT NULL primary key,
fecha_salida timestamp without time zone NOT NULL,
cod_almacen integer NOT NULL,
id_encargado_almacen integer NOT NULL,
observaciones text
foreign key (id_empleado_entrega) references encargado_almacen (id_empleado_entrega),
foreign key (cod_almacen) references almacen (cod_almacen)
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>Cod Salida mater</u>	<u>Int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>Fecha salida</u>	Times	no	no		Fecha de salida
<u>observaciones</u>	<u>Text</u>	si	no		Observaciones de la salida del material

Tabla datos 52. Datos Simples de la Tabla Salida_material

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>Id_ empleado</u>	<u>int</u>	No	si	<u>empleado</u>	<u>id_ empleado</u>	Identificador de la tabla empleado
<u>cod_ almacen</u>	<u>int</u>	No	si	<u>almacen</u>	<u>cod_ almacen</u>	Identificador de la tabla <u>almacen</u>

Tabla datos 53. Claves Foráneas de la Salida_material

Tabla detalle_entrada

```
CREATE TABLE detalle_entrada(
cod_entrada integer NOT NULL,
cod_material integer NOT NULL,
cantidad_material integer NOT NULL,
observacion text,
foreign key (cod_entrada) references entrada_material (cod_entrada),
foreign key (cod_material) references material (cod_material),
primarykey(cod_entrada,cod_material)
);
```

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_entrada</u>	<u>int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>cod_material</u>	<u>int</u>	no	si	Ingresar datos	Identificador de la tabla
<u>cantidad_material</u>	<u>int</u>	no	no		Cantidad del material detallada
<u>observacion</u>	<u>text</u>	si	no		

Tabla datos 54. Datos Simples de la Tabla Detalle_entrada

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_entrada</u>	<u>int</u>	No	si	<u>entrada_material</u>	<u>cod_entrada</u>	Identificador de la tabla <u>entrada_material</u>
<u>cod_material</u>	<u>int</u>	No	si	material	<u>cod_material</u>	Identificador de la tabla material

Tabla datos 55. Claves Foráneas de la Detalle_entrada

Tabla detalle_salida_material

```
CREATE TABLE detalle_salida_material(
```

cod_material integer NOT NULL,
 cod_salida_material integer NOT NULL,
 cantidad_material_entregado integer,
 observacion text,
 foreign key (cod_material) references material (cod_material),
 foreign key (cod_salida_material) references salida_material (cod_salida_material),
 primarykey(cod_material,cod_salida_material)
);

Campo	Dato	Nulo	Unique	Predeterminado	Descripción
<u>cod_material</u>	<u>Int</u>	no	no	Ingresar datos	Identificador de la tabla
<u>cod_salida_material</u>	<u>Int</u>	no	no	Ingresar datos	Identificador de la tabla
<u>cantidad_material_entregado</u>	<u>Int</u>	no	no		Cantidad del material entregado
Observaciones	<u>text</u>	si	no		Observaciones de las salidas del material

Tabla datos 56. Datos Simples de la Tabla detalle_salida_material

Campo	Dato	Nulo	Unique	Referencia	Atributo	Descripción
<u>cod_material</u>	<u>int</u>	No	no	material	<u>cod_material</u>	Identificador de la tabla material
<u>cod_salida_material</u>	<u>int</u>	No	no	<u>salida_material</u>	<u>cod_salida_material</u>	Identificador de la tabla <u>salida_material</u>

Tabla datos 57. Claves Foráneas de la detalle_salida_material

II.2.15. Prueba de Caja Negra

Su objetivo es probar la funcionalidad del código y encontrar casos en los que el módulo no atiende a su especificación, especialmente indicadas en los módulos que van a ser interfaz con el usuario.

Los métodos de prueba de Caja Negra se llevan a cabo sobre la interfaz del software y se basan en los requisitos funcionales del software.

La prueba de Caja Negra trata de un enfoque complementario que intenta descubrir diferentes tipos de errores. Los casos de prueba de la Caja Negra pretenden demostrar que:

- ❖ Las funciones del Software son operativas
- ❖ La entrada de datos se acepta de forma adecuada
- ❖ Se produce una salida correcta
- ❖ La integridad de la información externa se mantiene

Se derivan conjunto de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa.

La Prueba de Caja Negra intenta encontrar errores de las siguientes categorías:

- ❖ Funciones incorrectas o ausentes.
- ❖ Errores de Interface
- ❖ Errores de estructuras de datos o en accesos a base de datos externas
- ❖ Errores de rendimiento
- ❖ Errores de inicialización y de terminación

Los casos de prueba deben satisfacer los siguientes criterios:

- ❖ Reducir en un coeficiente que es mayor que uno, el de Casos de Prueba adicionales que se deben diseñar para alcanzar una prueba razonable.
- ❖ Que digan algo sobre la presencia o ausencia de clases de errores en lugar de un error asociado solamente con la prueba, en particular, que se encuentra disponible.

Modelos de Casos de Prueba

FOFMULARIO: Ingresar al Sistema como usuario



Fig. II. 151. Ingresar al sistema al Sistema Como Usuario

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

Tabla II.27 Formulario Acceso al Sistema Como Usuario

FORMULARIO: Adicionar Compras

ADICIONAR COMPRA

Fecha de Compra:	22-08-2013	Forma Pago:	Contado
Nombre Empresa:	SETAR		
Numero de Cheque:			
Descuento:	0.0	Monto Total:	100157
Observacion:	Se Pago al Contado		

Detalle De Los Materiales					
Cod Material	Material	Cantidad	Precio	Total	Observacion
3	impresora hp	5	5.0	25	Ninguna
4	baneras	6	6.0	36	Ninguna
5	Silla	12	8.0	96	Ninguna
2	impresora car	200	500.0	100000	Ninguna

Fig.II. 152 Formulario Adicionar Compras

Condiciones de Entrada	Condiciones Válidas	Condiciones Inválidas
Fecha de Compra	1. Selección	2. Otro Valor
Valor de entrada de Nombre Empresa	3. Alfanumérico	4. Otro Valor
Tamaño de Nombre	5. $5 \leq \text{Valor} \leq 50$	6. Valor < 5 7. Valor > 50
Valor de entrada de Forma de pago	8. Alfanumérico	9. Otro Valor
Tamaño de Forma de pago	10. $0 \leq \text{Valor} \leq 50$	11. Valor < 1 12. Valor > 50
Valor de entrada de Numero de cheque	13. Numérico	14. Otro Valor
Tamaño Numero de cheque	15. $1 \leq \text{Valor} \leq 50$	16. Valor < 1 17. Valor > 60
Valor de entrada	18. Numérico	19. Otro Valor

Descuento		
Tamaño de Descuento	20. $1 \leq \text{Valor} \leq 4$	21. Valor < 1 22. Valor >6
Valor de entrada monto total	23. Numérico	24. Otro Valor
Tamaño de monto total	25. $1 \leq \text{Valor} \leq 4$	26. Valor < 1 27. Valor >6
Valor de entrada Observación	28. Alfanumérico	29. Otro Valor
Tamaño de Observación	30. $0 \leq \text{Valor} \leq 200$	31. Valor < 1 32. Valor >200
Valor de entrada Detalle de Materiales	33. Lista de Selección	34. Otro Valor
Tamaño de Detalle de materiales	35. Selecciona los materiales deseados	36. Seleccionar en el botón

Tabla II.Formulario Adicionar Compra

FORMULARIO: Formulario Adicionar Persona

The screenshot shows a web form titled "Formulario de Persona" with a sub-header "ADICIONAR PERSONA". The form fields are as follows:

- Tipo Documento: A dropdown menu with "ci" selected.
- Numero Documento: A text input field containing "7104911".
- Nombre: A text input field containing "Juan".
- Apellido Paterno: A text input field containing "Perez".
- Apellido Materno: A text input field containing "Perez".
- Fecha Nacimiento: A date picker showing "8-8-1989".
- Telefono: A text input field containing "46545645".
- Email: A text input field containing "kjlj@hotmail.com".
- Direccion: A text input field containing "B/ la pampa".
- Fotografia: A file upload area with a button "Seleccionar archivo" and a file named "Koala.jpg" selected.

At the bottom of the form are two buttons: "Adicionar" (with a green checkmark icon) and "Cancelar" (with a red X icon).

Fig. II.153. Formulario Adicionar Persona

Condiciones de Entrada	Condiciones Válidas	Condiciones Inválidas
Valor de entrada Tipo de Documento	1. Selección	2. Otro Valor
Tamaño de tipo de documento	3. 1 Seleccionar	4.
Valor de entrada de Número de Documento	6. Numérico	7. Otro Valor
Tamaño de número de documento	8. $5 \leq \text{Valor} \leq 50$	9. Valor<5 10. Valor >50
Valor de entrada Nombre	11. Alfanumérico	12. Otro
Tamaño de Nombre	12. $0 \leq \text{Valor} \leq 50$	13. Valor <1 14. Valor >50
Valor de entrada Apellido	15. Alfanumérico	16. Otro

Paterno		
Tamaño de Apellido Paterno	17. $1 \leq \text{Valor} \leq 50$	18. Valor < 1 19. Valor > 60
Valor de entrada Apellido Materno	20. Alfanumérico	21. Otro
Tamaño de Apellido Paterno	22. $1 \leq \text{Valor} \leq 50$	23. Valor < 1 24. Valor > 60
Valor de entrada Teléfono	25. Alfanumérico	26. otro
Tamaño de Teléfono	27. Selecciona un tipo de Centro Salud válido	28. Selección por defecto ---seleccione---
Valor de entrada E-mail	29. Alfanumérico	30. Otro
Tamaño de Teléfono	31. $0 \leq \text{Valor} \leq 10$	32. Valor < 0 33. Valor > 10
Valor de entrada Dirección	34. Alfanumérico	35. Otro valor
Tamaño de dirección	36. $0 \leq \text{valor} \leq 200$	37. Valor < 1 38. Valor >= 201

Tabla II. Formulario Adicionar Usuario

Formulario Adicionar Usuario

The image shows a software window titled 'Formulario de Usuario'. Inside, there's a section titled 'ADICIONAR USUARIO'. This section contains a table-like form with four rows, each with a label on the left and an input field on the right. The first row is 'Datos Persona' with the value 'raul Ruiz Meji'. The second row is 'Usuario:' with the value 'qq4x8bzd'. The third row is 'Clave:' with an empty input field. The fourth row is 'Rol:' with a dropdown menu showing 'administrador'. Below the form are two buttons: 'Agregar' (with a green plus icon) and 'Cancelar' (with a red heart icon). The window has a standard OS-style title bar and scrollbars.

Fig. II.154. Formulario Adicionar Usuario

CONDICIONES DE ENTRADA	CONDICIONES DE ENTRADAS VALIDAS	CONDICIONES DE ENTRADAS INVALIDAS
Datos Persona	1. $1 < \text{valor} < 50$	2. $\text{Valor} < 1$ 3. $\text{Valor} > 50$
Usuario	4. Alfanumerico	5. Otro
Datos de persona.	6. Letras	7. $\text{Valor} < 1$ 8. $\text{Valor} > 50$
Valor de entrada de Clave	9. Alfanumerico	10. Otrp
Tamaño de Rol	11. $< \text{valor} < 50$	12. $\text{Valor} < 1$ 13. $\text{Valor} > 50$

FORMULARIO: Adicionar Empresa



Fig. II.155. Formulario Registro de Nueva Empresa

Condiciones de Entrada	Condiciones Válidas	Condiciones Inválidas
Valor de entrada de Nombre	1. Alfanumérico	2. Otro Valor
Tamaño de nombre	3. 3 <= Valor <= 20	4. Valor < 3 5. Valor >20
Valor de entrada Teléfono	6. Alfanumérico	7. Otro Valor
Tamaño de Telefono	8. 4 < Valor <= 40	9. Valor <4 10. Valor > 40
Valor de entrada dirección	11. Alfanumérico	12. otro valor
Tamaño de dirección	13. 4 < Valor <= 40	14. Valor < 4 15. Valor > 40

Tabla II. Formulario Registro de Nueva Empresa

1. Caso de Prueba “Ingresar al Sistema”

1.1.Descripción del Caso de Prueba

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

Las pruebas realizadas a este caso de uso son:

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

1.2.Condición de ejecución

A. Introducir Datos Correctos.

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

Precondiciones.

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

Valor de entrada.

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

- ❖ Introducimos: “maria” en el campo Usuario.
- ❖ Introducimos: “123h” en el campo Clave.
- ❖ Pulsamos el botón “Aceptar”.
- ❖ El sistema nos muestra la pantalla Principal del Administrar con los privilegios de acuerdo al rol que lo pertenece.

Observaciones

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

Resultados esperados

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

B. Introducir Datos Incorrectos de Ingreso.

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

Precondiciones.

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

Valor de entrada.

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

- ❖ Introducimos: “hola1” en el campo Usuario.
- ❖ Introducimos: “holitas” en el campo Clave.
- ❖ Pulsamos el botón “Aceptar”.
- ❖ El Sistema nos muestra un mensaje de error.

Observaciones

Los errores se muestran en el centro de los datos de entrada del formulario en un ventana de alerta.

Resultados esperados

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

2. Caso de Prueba “Adicionar Persona”

2.1.Descripción del Caso de Prueba

En esta parte del documento solo se cubrirá la lista de pruebas realizadas sobre el Caso de Uso “Adicionar Persona”.

Las pruebas realizadas a este Caso de Uso son:

- A. Adicionar Nueva Persona.
- B. Datos incorrectos: nombre con caracteres no permitidos.
- C. Datos incorrectos: nombre nulo.
- D. Datos incorrectos: Apellido Paterno con caracteres no permitidos.
- E. Datos incorrectos: Apellido Materno con caracteres no permitidos.
- F. Datos incorrectos: Fecha Nulo.

2.2.Condición de ejecución

A. Adicionar un Persona.

Ingresamos al Sistema como usuario Administrador y hacemos clic en Administrar Persona, elegimos la opción Lista Personas y hacemos clic en Registrar Persona el Sistema nos mostrará el formulario de registro del nueva persona.

Precondiciones.

La condición para este Caso de Prueba es que la Persona no exista en la base de datos del Sistema.

Valor de entrada.

Se lista las entradas con los datos Adicionar un Nueva Persona.

- ❖ Introducimos: Seleccionamos Ci en el Tipo de documento.
- ❖ Introducimos: “1234565 Tja.” en el campo Número de Documento.
- ❖ Introducimos: “María” en el campo Nombre.
- ❖ Introducimos: “Perez” en el campo Apellido Paterno.
- ❖ Introducimos: “Lopez” en el campo Apellido Materno.
- ❖ Introducimos: “19-10-1987” Seleccionamos en el campo Fecha Nacimiento.

- ❖ Introducimos: “66-54952” en el campo Teléfono.
- ❖ Introducimos: “j_213_m@hotmail.com” en el campo E-mail.
- ❖ Seleccionamos: “Barrio la Pampa” en el campo Dirección.

Observaciones

Para llenar el campo Fecha se muestra un calendario en el cual podemos seleccionar la fecha.

Resultados esperados

El Sistema almacena los datos del cliente en la base de datos.

B. Datos incorrectos: Nombre con caracteres no permitido.

Ingresamos al Sistema como usuario Administrador y hacemos clic en Administrar Persona, elegimos la opción Lista Personas y hacemos clic en Registrar Persona el Sistema nos mostrará el formulario de registro del nueva persona.

Precondiciones.

La condición para este Caso de Prueba es que el Paciente no exista en la base de datos del Sistema.

Valor de entrada.

Se lista las entradas con los datos Adicionar un Nueva Persona.

- ❖ Introducimos: Seleccionamos Ci en el Tipo de documento.
- ❖ Introducimos: “asd123.” en el campo Número de Documento.
- ❖ Introducimos: “Juan” en el campo Nombre.
- ❖ Introducimos: “Lopez” en el campo Apellido Paterno.
- ❖ Introducimos: “Martinez” en el campo Apellido Materno.
- ❖ Introducimos: “3-9-2013” Seleccionamos en el campo Fecha Nacimiento.
- ❖ Introducimos: “66-9875” en el campo Teléfono.
- ❖ Introducimos: “hola123hotmail.com” en el campo E-mail.
- ❖ Seleccionamos: “calle/beni” en el campo Dirección

Observaciones

Para llenar el campo Fecha de Nacimiento se cuenta con un calendario.**Resultados esperados**

El proceso termina no se almacena la persona en la base de datos y el sistema detecta el error cometido en el campo Nombre, el cual muestra un mensaje el campo sólo acepta alfanuméricos.

El mensaje no es claro porque solo dice que los campos seleccionados son requeridos.

Postcondiciones

El Sistema no realiza ninguna operación y permanece en el mismo estado.

C. Datos incorrectos: Nombre nulo.

Ingresamos al Sistema como usuario Administrador y hacemos clic en Administrar Persona, elegimos la opción Lista Personas y hacemos clic en Registrar Persona el Sistema nos mostrará el formulario de registro del nueva persona.

Precondiciones.

La condición para este caso de prueba es que la persona no exista en la base de datos del sistema.

Valor de entrada.

Se lista las entradas con los datos Adicionar Persona.

- ❖ Introducimos: Seleccionamos Ci en el Tipo de documento.
- ❖ Introducimos: “6012537.” en el campo Número de Documento.
- ❖ Introducimos: “nulo” en el campo Nombre.
- ❖ Introducimos: “Gonzales” en el campo Apellido Paterno.
- ❖ Introducimos: “Villegas” en el campo Apellido Materno.
- ❖ Introducimos: “3-9-2013” Seleccionamos en el campo Fecha Nacimiento.
- ❖ Introducimos: “66-9875” en el campo Teléfono.
- ❖ Introducimos: “hola123hotmail.com” en el campo E-mail.
- ❖ Seleccionamos: “calle/beni” en el campo Dirección.

Observaciones

Para llenar el campo Fecha de Nacimiento se cuenta con un calendario.

Resultados esperados

El proceso termina y no se almacena el cliente en la base de datos del sistema, se detecta que el nombre es nulo este campo está con una cadena nula.

Postcondiciones

El sistema no realiza ninguna operación y Muestra el campo con un mensaje que dice que esta nulo.

D. Datos incorrectos: Apellido Paterno con caracteres no permitido.

Ingresamos al Sistema como usuario Administrador y hacemos clic en Administrar Persona, elegimos la opción Lista Personas y hacemos clic en Registrar Persona el Sistema nos mostrará el formulario de registro del nueva persona.

Precondiciones.

La condición para este Caso de Prueba es que el Paciente no exista en la base de datos del sistema

Valor de entrada.

Se lista las entradas con los datos Adicionar Persona.

- ❖ Introducimos: Seleccionamos Ci en el Tipo de documento.
- ❖ Introducimos: "6012537." en el campo Número de Documento.
- ❖ Introducimos: "Gustavo" en el campo Nombre.
- ❖ Introducimos: "Gonzales123@" en el campo Apellido Paterno.
- ❖ Introducimos: "Villegas" en el campo Apellido Materno.
- ❖ Introducimos: "3-9-2013" Seleccionamos en el campo Fecha Nacimiento.
- ❖ Introducimos: "66-9875" en el campo Teléfono.
- ❖ Introducimos: "hola123hotmail.com" en el campo E-mail.
- ❖ Seleccionamos: "calle/beni" en el campo Dirección.

Observaciones

Para poner la fecha de nacimiento se selecciona del calendario

Resultados esperados

El proceso termina y no se almacena el Paciente en la base de datos del Sistema, se detecta que el Apellido Paterno es incorrecto.

Postcondiciones

Se muestra un mensaje de error a lado del campo Apellido Paterno.

E. Datos incorrectos: Apellido Materno con caracteres no permitido.

Ingresamos al Sistema como usuario Administrador y hacemos clic en Administrar Persona, elegimos la opción Lista Personas y hacemos clic en Registrar Persona el Sistema nos mostrará el formulario de registro de la nueva persona.

Precondiciones.

La condición para este Caso de Prueba es que el Paciente no exista en la base de datos del sistema

Valor de entrada.

- ❖ Introducimos: Seleccionamos Ci en el Tipo de documento.
- ❖ Introducimos: "6012537." en el campo Número de Documento.
- ❖ Introducimos: "Gustavo" en el campo Nombre.
- ❖ Introducimos: "Gonzales" en el campo Apellido Paterno.
- ❖ Introducimos: "Villegas123@" en el campo Apellido Materno.
- ❖ Introducimos: "4-9-1986" Seleccionamos en el campo Fecha Nacimiento.
- ❖ Introducimos: "66-9875" en el campo Teléfono.
- ❖ Introducimos: "hola123@hotmail.com" en el campo E-mail.
- ❖ Seleccionamos: "calle/beni" en el campo Dirección.

Observaciones

Para llenar el campo Fecha Nacimiento se muestra un calendario en el cual podemos seleccionar la fecha.

Resultados esperados

El proceso termina y no se almacena el paciente en la base de datos del Sistema, se detecta que el Apellido Materno es incorrecto.

Postcondiciones

Se muestra un mensaje de error a lado del campo apellido materno diciendo que sólo acepta caracteres alfanuméricos.

F. Datos incorrectos: Fecha nulo.

Ingresamos al Sistema como usuario Administrador y hacemos clic en Administrar Persona, elegimos la opción Lista Personas y hacemos clic en Registrar Persona el Sistema nos mostrará el formulario de registro del nueva persona.

Precondiciones.

La condición para este caso de prueba es que el Paciente no exista en la base de datos del Sistema.

Valor de entrada.

Se lista las entradas con los datos Adicionar un Nuevo Paciente.

- ❖ Introducimos: Seleccionamos Ci en el Tipo de documento.
- ❖ Introducimos: “6012537.” en el campo Número de Documento.
- ❖ Introducimos: “Gustavo” en el campo Nombre.
- ❖ Introducimos: “Gonzales” en el campo Apellido Paterno.
- ❖ Introducimos: “Villegas123@” en el campo Apellido Materno.
- ❖ Introducimos: “nulo” Seleccionamos en el campo Fecha Nacimiento.
- ❖ Introducimos: “66-9875” en el campo Teléfono.
- ❖ Introducimos: “hola123@hotmail.com” en el campo E-mail.
- ❖ Seleccionamos: “calle/beni” en el campo Dirección.

Observaciones

Para llenar el campo Fecha Nacimiento se muestra un calendario en el cual podemos seleccionar la fecha.

Resultados esperados

El proceso termina y no se almacena el paciente en la base de datos del Sistema, se detecta que el Campo fecha es requerido y no puede ser nulo.

Postcondiciones

Se muestra un mensaje de error a lado del campo Fecha Registro diciendo que es un campo requerido.

II.1.14.6 PRUEBA DE CONTROL DE CÓDIGO

Modelos de Casos de PruebaCrap4j es una implementación de la métrica CRAP que significa "Análisis y Predicciones sobre los Riesgos de Realizar Cambios".

Su función es analizar código fuente y buscar aquellas clases y métodos que alguien hizo en el pasado y que, a estas alturas del desarrollo, son muy difíciles de modificar, bien sea porque el código está programado de un modo que sea difícil de leer y de entender, o bien porque es prácticamente imposible determinar cuál puede ser el alcance de nuestros cambios. Si funciona bien, se suele decir que *"si funciona no lo toques"*, pero si funciona mal o queremos cambiar su modo de funcionamiento podemos tener un problema muy grande.

Pero, ¿en qué se basa la métrica CRAP para evaluar un método?. Pues en la combinación de dos factores:

- ❖ Su complejidad ciclomática, o número de bifurcaciones únicas presentes en un método.
- ❖ Su cobertura por pruebas unitarias, o % del código que es ejecutado cuando se lanzan las pruebas unitarias.
- ❖ CRAP(m) es un índice que, a mayor valor, implicará mayor dificultad de mantenimiento en el método .
- ❖ comp(m) es el valor de la complejidad ciclomática del método *m*.
- ❖ cov(m) es el % de cobertura de código del método *m* por pruebas unitarias.

Valores pequeños de CRAP(m) indican que los métodos son fácilmente mantenibles, mientras que a medida que el valor aumenta, también aumenta la dificultad de mantenimiento. Para disminuir el valor de CRAP(m) se debe hacer dos cosas: escribir pruebas unitarias que aseguren que cualquier cambio en el código no va a afectar al resto de la aplicación y reducir su complejidad ciclomática partiendo ese método en otros más pequeños. Un método se considera difícilmente mantenible cuando su índice CRAP es mayor que 30.

Caso de prueba para el proyecto Sistema Web MGAOMA

porcentaje de métodos de CRAPpy



Fig. II.61 Reporte Barra

Resumen	
Porcentaje de métodos de CRAPpy	3.13 %
CRAP Load	475
Total Method Count	1532
CRAPpy Method Count (CRAP > 30)	48

Tabla II.32 Resumen de Prueba Crap

El 3.13% de los métodos de la clase son CRAPpy (difícilmente mantenibles)

La complejidad ciclomática de todo el proyecto es 475

Cantidad de métodos analizados es de 1532

El proyecto es considerado difícilmente mantenible por tener un índice mayor a 30

Distribución de método crap

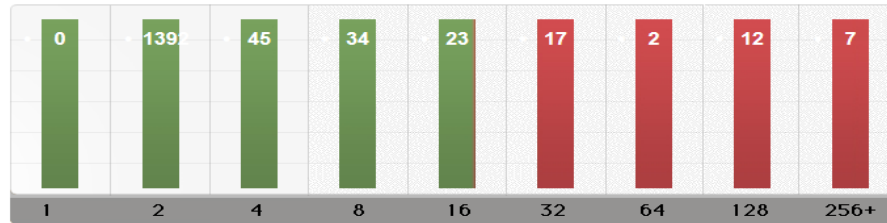


Fig. II. Distribución de métodos crap

Valores entre 1-10

Programa Simple, sin mucho riesgo

Valores entre 11-20

Más complejo, riesgo moderado

Valores entre 21-50

Complejo, Programa de alto riesgo

Valores entre 50

Programa no testeable, muy alto riesgo.

II.3. ORGANIZACIÓN DEL PROYECTO

II.3.1. Participantes en el Proyecto

La institución donde se realiza el proyecto, es decir la Oficialía Mayor Administrativa de la alcaldía de San Lorenzo.

Director y Desarrollador del Proyecto: María de los Ángeles Castillo, alumna del Decimo semestre de la Carrera de Ingeniería Informática en la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho. Con experiencia en metodologías de desarrollo, notaciones UML, herramientas CASE, desarrollo de sistemas en entorno Escritorio y Web, diseño de Datos, conocimiento del lenguaje de programación Java y varias metodologías de Prueba.

Cabe recalcar que la universitaria María de los Angeles Castillo tendrá todos los roles que se necesita para el desarrollo de un proyecto que son: Jefe de Proyecto, Analista de Sistemas, Programador e Ingeniero de Software.

II.3.2. Interfaces Externas

El sistema MGAOMA permitirá el ingreso los siguientes usuarios que son:

Administrador(a) del sistema MGAOMA – El cual podrá adicionar, modificar, dar de baja, dar alta a usuarios, y manejar todo el sistema crear y descargar backups de la base de datos del sistema MGAOMA y acceder a toda la información transformada en reportes.

Encargado de Almacen-del sistema MGAOMA – El cual podrá realizar consultas, generar reportes del modulo Almacen.

Encargado de solicitudes- del sistema M GAOMA– El cual podrá realizar Solicitudes, cotizaciones y aprobaciones.

II.3.3. Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Roles	Responsabilidades
-------	-------------------

Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con los usuarios. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.

Tabla II.3.3 Roles y responsabilidades

II.4. GESTIÓN DEL PROYECTO

II.4.1. Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

II.4.2. Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para la fase de Construcción sólo una aproximación muy preliminar).

Fase	Nro. de Interacciones	Duración
Fase de Inicio	2	15 semanas
Fase de Elaboración	2	18 semanas
Fase de Construcción	3	18 semanas
Fase de Transición	-	-

Tabla II.4.2 Plan de las fases

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Roles	Hito
Fase de Inicio	En esta fase desarrollarán los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente /usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera reléase de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación

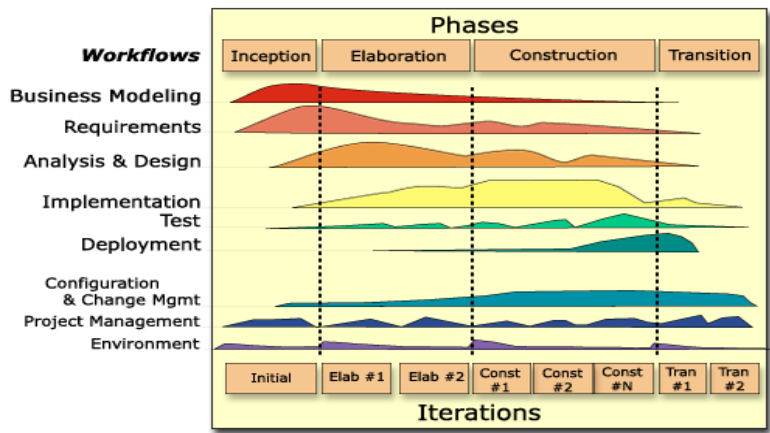
	y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de una semana.
Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una reléase a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la reléase 2.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta.
Fase de Transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

Tabla II.4.2Hitos

II.4.3. Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o

menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.



Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas / Artefactos generados o modificados durante las Fases	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio	1/06/2013	2/08/2013
Requisitos		
Visión	12/07/2013	2/08/2013
Modelo de Casos de Uso	18/06/2013	2/08/2013
Especificación de Casos de Uso	18/06/2013	2/08/2013
Análisis/Diseño		
Modelo de Datos	2/08/2013	6/11/2013
Especificación de Datos	2/08/2013	6/11/2013
Modelo de Actividades	18/06/2013	2/08/2013
Modelo de Secuencia	18/06/2013	2/08/2013

Modelos de Colaboración	18/06/2013	2/08/2013
Implementación		
Diagrama de Clases	2/08/2013	6/11/2013
Especificación de Clases	2/08/2013	6/11/2013

Tabla II.4.3 Diciplinas y Artefactos

II.4.4. Seguimiento y Control del Proyecto

II.4.4.1. Gestión de Requisitos

Los requisitos del sistema son expresados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambios, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

II.4.4.2. Control de Plazos

El calendario tendrá un seguimiento y evaluación semanal por el director de proyecto y los involucrados como ser el Director y los técnicos del sistema MGAOMA. Además de las presentaciones de acuerdo a un calendario efectuado por el Comité de Seguimiento y Control (Docentes de Taller III).

II.4.4.3. Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambios tendrán un seguimiento para asegurar la conformidad respecto a la solución de dicha deficiencia, en este proyecto se realizaron diferentes solicitudes de cambio, las cuales fueron efectuadas con éxito.

II.4.4.4. Gestión de Riesgos

A partir de la fase de inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategias para mitigarlos o acciones de configuración.

Para realizar la administración de los riesgos del proyecto se tomó en cuenta el plan de administración de riesgos.

Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones, basado en la metodología RUP el proyecto constará de cuatro versiones que fueron evolucionando con el pasar del tiempo hasta llegar a ser un documento completo.

II.COMPONENTE2: CAPACITACIÓN AL USUARIO DEL SISTEMA

II.2.1. Introducción.

Para la implementación de un sistema dentro de una empresa o institución es imprescindible capacitar a los involucrados con el manejo del sistema. Lo cual lograra que se pueda sacar los máximos beneficios al sistema y reducirá enormemente los riesgos, errores que pueda surgir al momento de interactuar con el sistema.

En el caso del sistema MGAOMA (Mejorar la Gestión del Área de Archivos de la O. M. A. del Gobierno Municipal Autónomo de San Lorenzo.) Se realizo la capacitación a los usuarios que están involucrados directamente con el sistema, estos son:

- Administrador
- Encargado de Solicitud
- Encargado de Almacén

II.2.2. PLANTEAMIENTO DEL PROBLEMA

Al no contar con conocimientos adecuados del sistema informático que se quiere implantar se presente instruir al usuario que interactuaran con el sistema. El sistema cumple con requerimientos del usuario final.

II.2.3. OBJETIVO

El principal objetivo de la capacitación del personal de la institución es lograr que los usuarios que interactuaran a diario con el sistema cuenten con una preparación adecuada

que les ayudara a manejar correctamente el sistema lo cual se traducirá directamente en obtener beneficios del sistema que es el principal objetivo en sí del presente proyecto.

II.2.4. DESCRIPCION DE LOS METODOS DE CAPACITACION

El proyecto plantea llevar a cabo para la capacitación una metodología de enseñanza socializada el cual pretende ejecutar una metodología de enseñanza socializada dirigida a la Oficialía Mayor Administrativa de honorable alcaldía municipal de San Lorenzo.

Para llevar a cabo se desarrollará usando las siguientes técnicas:

Técnica Expositiva

Consiste en la exposición oral, por parte del capacitador el cual se estimulara la participación del personal capacitado, este requiera una buena motivación para atraer la atención de los participantes.

Técnica de la Experiencia

La experiencia es un procedimiento eminentemente activo y que procura:

Exponer de forma presencial los pasos a seguir para realizar alguna acción

Explicar el funcionamiento de algo mientras es observado

Comprobar, con razones lo que va suceder, partiendo de la experiencia.

Conferir confianza para actuar en el terreno de la realidad de manera lógica.

Convencer a cerca de la veracidad de la ley de causa y efecto

Fortalecer la confianza en sí mismo

Formar la mentalidad científica

Orientar para solucionar problemas.

Esta técnica se realiza permitiendo a los participantes interactuar con el sistema web, para enseñarle el manejo y uso del sistema.

II.2.5. ALCANCES Y LIMITACIONES

Se llego a capacitar a todos los que intervienen en el sistema, y también a algunos empleados que trabajan dentro de la institución por el caso de ser necesario en algún momento de emergencia.

II.2.6. ESPECIFICACIÓN DE LA CAPACITACIÓN

Para la ejecución de la capacitación se elaboro un manual de usuario con el objetivo que dicho manual sea usado en caso de surgir alguna duda sobre el manejo del sistema. El cual cuenta con imágenes impresas de las pantallas del sistema los posibles mensajes que podrían salir al usuario.

II.2.7.MEDIOS A UTILIZAR.

Materiales Impresos.

- Manual Usuario
- Guías de usuario

Otros materiales

Materiales Audiovisuales

- Equipos de computación
- Otros Manuales

II.2.8. CONTENIDO

CRONOGRAMA		
Días	Lecciones	Fechas
Lección 1	Introducción y manejo del sistema	10/07/13
Lección2	Manejo del modulo Administrador	11/07/13
Lección 3	Manejo de modulo Encargado de Solicitud	12/07/13
Lección 3	Manejo del modulo Encargado de Aprobación	15/07/13
Lección 4	Manejo del modulo Encargado de Cotización	16/07/13
Lección 5	Manejo del modulo Encargado de Compras	17/07/13
Lección 6	Manejo del modulo Encargado de Almacén	18/07/13

II.2.9. CONCLUSIONES

Durante la capacitación se hizo evidente el interés, la entrega y la atención del personal en recibir la capacitación poniendo de esta manera de relieve que la incorporación del nuevo sistema les facilitara el trabajo relacionado con la gestión de compra control del almacén de la O.M.A., etc.

Otro rasgo que sobresalió durante la capacitación es que el uso de tecnologías de información, dentro de la institución son verdadera importancia con ellas se puede aumentar la productividad y el desempeño del personal de la institución.

II.2.10. RECOMENDACIONES.

Las experiencias vividas durante la capacitación enseñaron algunas lecciones que vale la pena tomar en cuenta a la hora de realizar la capacitación del personal de la institución, las mismas son las siguientes:

Organizar los manuales de usuario de manera ordenada, clara y sencilla en lo referente al funcionamiento del sistema.

Si los involucrados en el sistema son pocos es mejor realizar la capacitación en forma individual pues es más fácil conocer de cerca las necesidades de cada usuario.

Mostrar al usuario que el sistema reducirá altamente los porcentajes de error que se tenían en el sistema manual.

Inculcar en el usuario el uso del manual en caso de surgir alguna duda.

II.2.11. MEDIOS DE VERIFICACIÓN

Fotos que Certifican la capacitación

Ver en Anexos.

CAPITULO III

CONCLUSIONES Y RECOMENDACIONES

III. CONCLUSIONES Y RECOMENDACIONES

III.1. CONCLUSIONES

La elaboración del proyecto ha sido entretenido y muy moroso, se ha tenido que dedicar mucho tiempo, sobre todo para la curva de aprendizaje de la tecnología. En cuanto a experiencia adquirida, estamos muy satisfechos porque el proyecto nos ha dado la oportunidad de aplicar conocimientos que se han dado durante la carrera.

Siguiendo este concepto de responsabilidad informática, tres aspectos han sido claves en este proyecto, el primero está relacionado con la intención de informatizar para mejorar, por ello se plantea como propósito el mejoramiento del control del área archivos de la oficialía mayor administrativa de la alcaldía de san Lorenzo, este propósito se logra a través de un sistema informático y un programa de capacitación efectivo. El segundo aspecto, tiene relación con la responsabilidad de enlazar las necesidades de los usuarios con la funcionalidad del sistema, y para formalizar este hecho, se han expresado estas necesidades bajo la norma IEEE-830. El tercer aspecto relevante en el proyecto, es el uso de las metodologías de desarrollo acompañadas con una herramienta CASE, puesto que permite tener un mejor control en todas las etapas del sistema, en este caso, se ha hecho uso de la metodología RUP, la herramienta Visual Paradigm Enterprise 6.4.

En este proyecto se ha recopilado información acerca de la realidad actual de la oficialía mayor administrativa. Se ha planteado una solución a una parte de los problemas que enfrentan los cuales se han planteado en este documento.

La seguridad del sistema fue implementada a través de la estrategia empleado, usuarios, cargos donde los usuarios-empleados de los sistemas tienen cargos y estos le permiten acceder a determinadas funciones dependiendo el rol asignado.

III.2. RECOMENDACIONES

El fin del proyecto es coadyuvar con una adecuada gestión de la información, el sistema y el programa de capacitación, son parte sustancial para lograrlo. Sin embargo,

Existen muchos puntos que vale la pena tocar; pero el más importante tiene que ver con los conocimientos previos que es necesario poseer antes de construir un sistema informático. Una de las principales desventajas que se vio a construir este proyecto fue la carencia de información que no quisieron brindar los interesados.

Antes de crear un Sistema se debe estar seguro que los clientes o usuarios, interesados en el proyecto se comprometan a brindar información necesaria para su creación.

Para hacer proyectos informáticos debemos usar patrones estándares de diseño y desarrollo de sistema para asegurar una mayor calidad de software en cuanto a usabilidad, escalabilidad, fiabilidad y mantenibilidad del Sistema.

Antes de crear un Sistema se debe estar seguro que los clientes o usuarios, interesados en el proyecto se comprometan a brindar información necesaria para su creación.

Utilizar patrones de diseño en el desarrollo de sistemas que aseguren una mayor calidad del software en cuanto a usabilidad, escalabilidad, fiabilidad y mantenibilidad del sistema. Evaluar y seleccionar el Sistema Operativo adecuado para la implementación de herramientas de Seguridad Informáticas en Servidores Web.