

CAPITULO I: EL PROYECTO

I.1. INTRODUCCION

COSALLT se encuentra en la necesidad de automatizar el proceso de planeación estratégica-operativa anual que realizan los departamentos y áreas de la empresa, debido al proceso manual de información de adquisición y contratación, recolectados de cada uno de las áreas, existe mucha retardación en la elaboración de informes y reportes, ocasionando que la asignación, control y supervisión por parte de Almacenes no sea adecuada.

La empresa es la institución o agente económico que toma las decisiones sobre utilización de factores de la producción para obtener los bienes y servicios que se ofrecen en el mercado. En toda compañía se realizan procedimientos que ameritan de un conjunto de herramientas, que puestas en la práctica conducen a la fluidez de los objetivos macro o micro que se trace cualquier departamento con fines corporativos, por otro lado la dinámica dentro de ésta, debe consistir siempre en maximizar el nivel de funcionamiento efectivo y coordinado, ejecutando actividades en conjunto y fortaleciendo constantemente la capacidad del personal inmerso en ella, consolidando así el proceso productivo en general de forma directa o indirecta.

Se hace necesario que las gerencias de las empresas que se someten a los procesos licitatorios permitan visualizar de forma unánime los requerimientos a corto y a largo plazo de los participantes de las unidades funcionales a todo nivel y tratando de satisfacer al máximo las mismas. Todo ello da lugar a un mejor rendimiento cohesionado con los cambios internos y externos que se produzcan en el entorno de la empresa (COSALLT.).

Por lo general, esta problemática es asumida de manera normal, tratando en lo posible de evitar la desestabilización del personal y por ende la labor que se realiza, contribuyendo a su vez a que el proceso de adaptación sea rápido y confiable, para ello se valdrá de muchas técnicas, tan simples como la implementación de charlas instructivas al personal.

Por lo tanto, un proceso de licitación en todas sus fases requiere el cumplimiento de ciertos parámetros que avalados por distintas normas o procedimientos dan la consecución de planteamientos realizables de cualquier órgano empresarial dentro de un proceso la cotidianidad o costumbre se hace presente cuando una actividad se realiza de una manera específica, donde muchas veces el trabajo se

vuelve hasta monótono, sin embargo, cuando surgen circunstancias de cambio, sea a gran o pequeña escala, se evidencia la duda e incertidumbre de cómo el grupo acogerá tales cambios la incidencia que pudiera causar dentro del plano de la productividad empresarial.

La falta de aplicación de reglamento específico origina deficiencias graves, que incumplen con la normativa básica de administración de Bienes y Servicios teniendo como resultado, improvisaciones, registros desactualizados, sistemas inventarios obsoletos de control de activos fijos e inmuebles e insumos todo estas deficiencias ocasiona un daño económico a la entidad, específicamente en la cuenta de activos fijos de balance general.

I.1.1. Planteamiento del Problema

La Cooperativa de servicios de agua potable y alcantarillado de Tarija (COSAALT-LTDA), por la morosidad de realizar los formularios se automatiza el proceso de planeación operativa anual que realizan las áreas para la planificación en colaboración con todas las áreas de la empresa.

I.1.2. Situación Actual

El proceso de planeación hoy en día se realiza de forma manual, es decir se inicia con la planificación anual propiamente dicha que consiste en realizar la formulación de forma manual con ayuda de hojas de cálculo (Microsoft Excel). Estos resultados se van a registrar en un documento oficial el cual servirá como base para plantear la planificación operativa anual de la empresa.

Para la elaboración de la planificación operativa se requiere de que cada área, departamento, unidad y gerencia establezca según sus actividades y responsabilidades, el conjunto de actividad que se van a cumplir en el tiempo máximo de un año, las cuales deben ser realistas, estimulantes, coherentes, prioritarias, medibles y cuantitativas por tanto deben ser representadas con un indicador operativo. Una vez definidas las actividades a cumplir se envía un reporte al departamento de Planificación de esta actividad, en donde recopilan información, revisan y aprueban la formulación de las actividades y elaboran el Plan Estratégico Institucional.

Cada área, departamento, unidad y gerencia de la empresa debe emitir trimestralmente los resultados de gestión obtenidos en las operaciones planteadas. Estos informes deben ser entregados oportunamente al responsable

de Control Presupuestario, quien revisa la consistencia de la información, aprueba los datos, los procesa de acuerdo a distintos criterios para aplicar métodos de cálculo apropiados y elabora el informe trimestral consolidado y acumulado de la Valoración del Plan Estratégico.

Este control de gestión o monitoreo tiene por objeto presentar informes trimestrales del grado de cumplimiento de las metas, para que en base a la revisión de estos informes en juntas de directiva y reuniones, se puedan formular y aplicar a tiempo procedimientos para el cumplimiento de los objetivos estratégicos y a su vez mantener a la organización en un proceso de mejora continua.

I.1.3. Justificación

El sistema para la automatización de la valoración y control sobre la ejecución del Plan Operativo en COSAALT tiene como propósito disminuir significativamente el tiempo actualmente empleado en la valoración de la información y eliminar el error humano en la misma.

El sistema brindará la facilidad de visualizar los resultados obtenidos y la evolución de sus actividades, mediante la formulación del POA según los rangos de tolerancia establecidos en la empresa

Así también este sistema permitirá la rápida elaboración y generación de reportes, ya que cada usuario al interactuar con el sistema, trabaja sobre una única versión de la información, a la que pueden acceder fácilmente los distintos usuarios que forman parte de este proceso.

I.1.4. Objetivos

I.1.4.1. Objetivo General

Mejoramiento para la planificación de la elaboración de los Plan Operativo Anual de la Cooperativa de servicios de agua potable y alcantarillado de Tarija (COSAALT-LTDA).

I.1.4.2. Objetivos Específicos

- Desarrollo de sistema ERP dirigido a la administración de las transacciones del personal.
- Capacitación al personal para el uso del Sistema.

I.1.5.Alcance

El Sistema para la Gestión del Plan Operativo pretende automatizar el proceso actual de valoración y control sobre la ejecución del plan operativo en la empresa COSAALT, para mostrar un panorama actualizado de la situación real de la empresa referente a lo realizado contra lo planificado durante cada año.

Por tanto este contempla el registro de datos, por parte de los responsables de las unidades operativas, tanto de su planificación como de los avances ejecutados en el año, para que tras su procesamiento se disponga inmediatamente de reportes de resumen, que sirvan como una guía para la toma de decisiones tanto para los responsables de las actividades.

I.2. La Organización

I.2.1 Datos Generales COSAALT Ltda.

Nombre: Cooperativa de Servicios de Agua Potable y Alcantarillado de Tarija.

Siglas: COSAALT.

Dirección: Calle Ingavi entre Calle O'connor y Calle Mendez

Nº Casilla: 675

Sitio Web: www.cosaalt.org.bo

Correo Electrónico: cosaalt-ltda@cosaalt.org.bo

I.2.1.1. Planificación Estratégica – Operativa empresarial

Para una mejor gestión gerencial es fundamental realizar el proceso de planeación estratégica al interior de las organizaciones. Este proceso consiste en determinar la realidad del entorno y establecer una secuencia de acciones en cierta forma u orden para que con un esfuerzo sistemático, se consiga mantener estabilidad en el mercado y establecer una directriz de crecimiento, prosperidad y bienestar pese a los cambios situacionales que puedan existir.

I.2.1.2. Planificación Estratégica

La planificación estratégica se inicia con la definición de la visión, misión y objetivos estratégicos de la organización.

I.2.1.2.1. Misión

Es la formulación de un propósito duradero, que revela el concepto de una organización, su principal producto o servicio y las necesidades principales de

los clientes que la organización pretende satisfacer³. Es decir define la razón de ser de la organización

I.2.1.2.2. Visión

Es un futuro atractivo, creíble y posible que toma la forma de una declaración de intenciones cuidadosamente formulada. Está relacionada con los valores considerados por la organización. Describe una imagen del futuro deseado para la organización en donde se describe la situación ideal a la que podría llegar la empresa en largo plazo.

Una vez determinada la filosofía empresarial se prosigue a realizar un diagnóstico situacional de la organización, su productividad actual y la tendencia actual del mercado mediante el monitoreo del entorno con identificación y análisis de factores internos y externos a la organización.

I.2.1.3. Filosofía Empresarial COSAALT

I.2.1.3.1. Visión

“COSAALT entidad capacitada en gestión de agua potable y alcantarillado sanitario para mejorar la calidad de vida de la población que se basa en principios, ideales y valores compartidos”.

I.2.1.3.2. Misión

Es garantizar los servicios de agua potable y alcantarillado sanitario con calidad y sostenibilidad para encarar una estrategia de mejoramiento continuo de la organización.

I.2.1.3.3. Objetivos Estratégicos

EPSA: Entidad Prestadora de Servicios de Agua Potable y/o Alcantarillado Sanitario Responsable de la Administración, operación y mantenimiento de los servicios de agua potable y alcantarillado sanitario, a un determinado número de usuarios, bajo diferentes modalidades de gestión.

I.2.1.4. Planeamiento Operativo

El plan operativo constituye el establecimiento de metas con indicadores operativos, actividades y áreas responsables de su cumplimiento, las cuales deben siempre enmarcarse en los lineamientos de la planeación estratégica empresarial.

I.2.1.5. Metas

Son aspiraciones o puntos de referencia que se deben lograr en un periodo de un año o menos, con el propósito de alcanzar, en el futuro, objetivos a un plazo más largo. Deben ser medibles, cuantitativos, realistas, estimulantes, coherentes y prioritarios. Las metas son la base para la asignación de recursos.

Las metas planteadas en el plan operativo deben ser monitoreadas y valoradas constantemente para alertar a los responsables del cumplimiento de la misma y sugerir en caso de ser necesario, replantear estrategias en mira de cumplir con las metas propuestas.

I.2.1.6. Directivos

Presidente:	Hno. Dr. Anibal Fernandez
1er. Vicepresidente	Hno. Sr. Ciro Ortega
2do. Vicepresidente	Hno. Dr. Daniel Alfaro Cadena
Secretario	Hno. Sr. Victor Hugo Rojas
Tesorero	Hno. Lic. Enrique Ramirez Rivera
Vocal	Hno. Prof. David Pumarino Lora
Vocal	Hno. Dr. Telmo Bejarrano

I.2.1.7. Analisis de Fortalezas, Oportunidades, Debilidades y Amenazas (FODA)

I.2.1.7.1. ASPECTOS INTERNOS

I.2.1.7.1.1. FORTALEZAS

Procesos de formulación de políticas.

Capacidad de interacción con los actores sociales.

Equipo profesional multidisciplinario, calificado e institucionalizado.

Estabilidad funcionaria de los equipos técnicos.

Relación e interacción Pública.

I.2.1.7.1.2. DEBILIDADES

Deficiente uso y aprovechamiento de sistema informático instalados.

Procedimientos y procesos internos muy burocráticos.

Mecanismos de correspondencia y comunicación deficientes.

Prioridad y preeminencia de lo coyuntural versus lo estratégico.

Falta de articulación y coordinación entre el nivel gerencial y técnico.

Excesiva concentración de responsabilidad en el área administrativa

Personal técnico insuficiente en unidades de bienes y servicios
 Alta rotación de personal en Análisis del Sistema de Administración de Bienes y Servicios para Buscar eficacia en la Administración Pública.

I.2.1.7.2. ASPECTOS EXTERNOS

I.2.1.7.2.1. OPORTUNIDADES

Enfoque de desarrollo económico del Dpto con base en el sector público. Probabilidad alta de acceder a recursos financieros para lograr la reactivación del sector público mediante sus instituciones.

I.2.1.7.2.2. AMENAZAS

Apoyo de la actual gestión de gobierno
 Limitada capacidad operativa técnica.
 Definición de objetivos y acciones de la cooperación en el sector público.

I.2.2. Metodología de Desarrollo

I.2.2.1. Metodología RUP¹ (Rational Unified Process)

a) Definición.- RUP es un marco del proyecto que describe una clase de los procesos que son iterativos e incrementales. Es un proceso de Ingeniería de Software que captura las mejores prácticas del conocimiento de líderes en Ingeniería de Software y que provee un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo.

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

Nos permite realizar un levantamiento exhaustivo de requerimientos.

Las actividades de RUP se centran en crear y mantener modelos, utilizando UML, en forma efectiva.

- Busca detectar defectos en las fases iniciales.
- Intenta reducir al número de cambios tanto como sea posible.
- Realiza el Análisis y Diseño, tan completo como sea posible.

¹<http://rolandojaldin.blogspot.com/2010/10/introduccion-la-metodologia-rup-proceso.html>
<http://ingsoftware2ps.blogspot.com/2012/05/metodologia-rup-introduccion-y.html>
<http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/>

- Diseño genérico, intenta anticiparse a futuras necesidades.
- Las necesidades de clientes no son fáciles de discernir.
- Existe un contrato prefijado con los clientes.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.

b) Características esenciales.- Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

- **Está dirigido por los Casos de Uso:** Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba.

- **Está Centrado en su Arquitectura:** La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

- **Es Iterativo e Incremental:** Según el equilibrio correcto entre los Casos de Uso y la Arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y Arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

c) Fases en el ciclo de Desarrollo.- Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases que se describirán a continuación, la figura muestra las Fases de desarrollo y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica en cuál de los flujos se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.

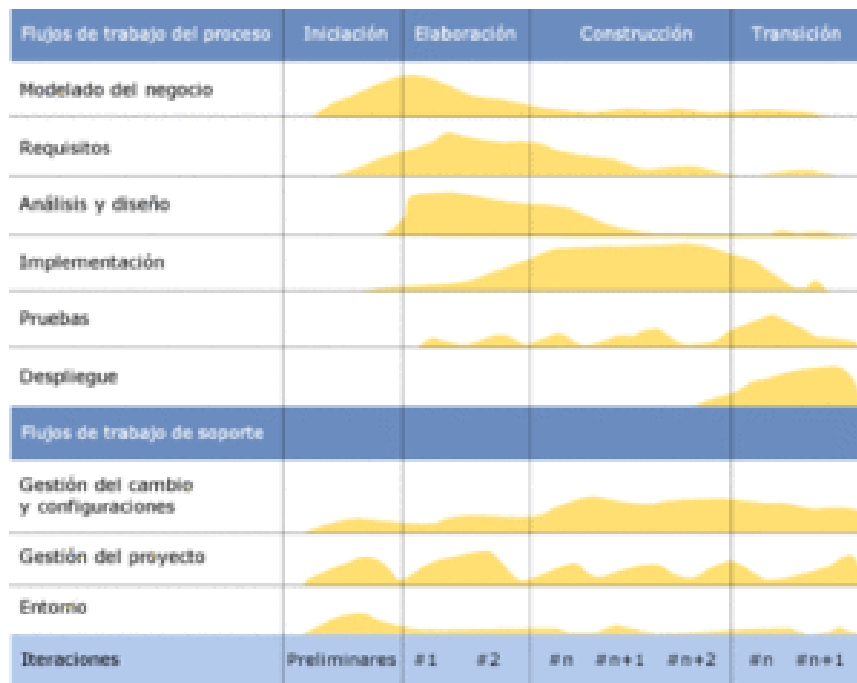


Figura 1 Fases en el Ciclo de Desarrollo de RUP

- **Fase 1: Preparación Inicial (“Incepción”)**

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

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

- **Fase 2: Preparación Detallada (“Elaboración”)**

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

El resultado de la fase de elaboración es:

Un modelo de caso de uso (por lo menos 80% completo) - todos los casos de uso y actores deben haber sido identificados-, y se han desarrollado la mayoría de las descripciones de casos de uso.

Requerimientos suplementarios que capturan los requerimientos no funcionales o cualquier requerimiento que no se asocie a un caso de uso específico.

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

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta.

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

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos.

Éste incluye:

- Operación en paralelo con un sistema anterior que el nuevo sistema esté sustituyendo.
- La conversión de las bases de datos operacionales.

Entrenamientos y capacitación de los usuarios y la gente de mantenimiento.

I.2.2.2. UML² (Lenguaje de Modelamiento Unificado - Unified Modelling Language)

El Lenguaje Unificado de Modelado (UML) es tal como su nombre lo indica, un lenguaje de modelado y no un método o un proceso. El UML está compuesto por una notación muy especificada y por las reglas semánticas relacionadas para la construcción de sistema de software. El UML en sí mismo

²<https://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html>
users.dcc.uchile.cl/~psalinas/uml

no prescribe ni aconseja como usar esta notación en el proceso de desarrollo o como parte de diseño orientado a objeto.

El UML soporta un conjunto rico en elementos de notación gráficos. Describe la notación para clases, componentes, nodos, actividades, flujo de trabajo, casos de uso, objetos, estado y como modelar la relación entre esos elementos. UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia un usuario externo.

UML introduce nuevos diagramas que representa una visión dinámica del sistema. Es decir, gracias al diseño de la parte dinámica del sistema podemos darnos cuenta en la fase de diseño de problemas de la estructura al propagar errores o de las partes que necesitan ser sincronizadas, así como del estado de cada una de las instancias de cada momento. El diagrama de clases continua siendo muy importante, pero se debe tener en cuenta que su representación es limitada, y que ayuda a diseñar un sistema robusto con partes reutilizables, pero no a solucionar problemas de propagación de mensajes ni de sincronización o recuperación ante estado de error. En resumen, un sistema debe estar bien diseñado, pero también debe funcionar bien.

Este modelamiento es usado para:

- Analizar y diseñar aplicaciones
- Manejo de complejidades
- Definir arquitectura de software

Y combina en el mismo:

- Conceptos de modelamiento de datos
- Modelado del negocio
- Modelado de objetos

Este lenguaje para modelamiento es utilizado como estándar gráfico para la visualización, especificación, construcción y documentación de sistemas⁴, se puede usar en las distintas fases de la metodología y para distintas tecnologías, por ser independiente o específica de una plataforma según se requiera.

Modelo de Objetos

Captura la estructura estática de los objetos así como sus relaciones, tras la identificación de los objetos procedentes del dominio de la aplicación.

I.2.2.1.1. Diagrama de Casos de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.

Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

I.2.2.1.1.1. Símbolos de actores

La figura de palillos, mencionada como *actor*, representa participantes en los casos de uso. Los actores pueden ser personas o cosas. Si un actor es una persona, entonces, en realidad, nunca se puede representar por medio de un código. Si un actor es otro subsistema, entonces se le puede observar como una clase o subprograma, pero todavía representarse usando el símbolo de actor en los diagramas de casos de uso.

Los actores se descubren como resultado del análisis. Conforme vaya identificando los macro usos del sistema, identificará quiénes son los participantes para esos casos de uso. En principio, registre cada actor a medida que se descubre, agregando un símbolo de actor a su modelo y describiendo cuál es su papel. Nos preocuparemos acerca de la organización y el refinamiento más adelante, en la sección titulada “Creación de los diagramas de casos de uso”.

I.2.2.1.1.2. Casos de uso

El símbolo del caso de uso se utiliza para representar capacidades. Al caso de uso se le da un nombre y una descripción mediante un texto. Este último debe describir cómo inicia y finaliza el caso de uso, e incluye una descripción de la capacidad descrita por el nombre de la misma, así como escenarios de apoyo y requisitos no funcionales. En la sección titulada “Creación de los diagramas de casos de uso”, examinaremos ejemplos de nombres de casos de uso, y en la sección titulada “Documentación de un caso de uso utilizando un borrador”, proporcionaré un borrador modelo que pueda usar para ayudarse a escribir las descripciones de los casos de uso.

I.2.2.1.1.3. Conectores

Dado que los diagramas de casos de uso tienen múltiples actores y en virtud de que los casos de uso pueden estar asociados con los actores y con otros casos de uso, se utilizan los conectores para indicar la manera en que ambos

están asociados. Además, los estilos de conectores pueden cambiar para transmitir más información acerca de la relación entre los actores y los casos de uso. Por último, los conectores pueden tener adornos y anotaciones que suministran incluso más información.

I.2.2.1.1.4. Estilos de líneas para los conectores

Existen tres estilos básicos de líneas para los conectores. Un conector de línea simple se llama *asociación* y se usa para mostrar cuáles actores están relacionados con cuáles casos de uso. Por ejemplo, en la Figura1 se mostró que un patrón está asociado con el caso de uso “Crear lista de trabajos”.

Un segundo estilo de conector es una línea punteada con una flecha direccional (Figura 1). Este estilo de conector se conoce como *dependencia*.



Figura 2 Diagrama de casos de uso - conector asociación

La flecha apunta hacia el caso de uso del que depende. Por ejemplo, suponga que a los patrones de *www.motown-jobs.com* se les debe dar acceso para crear una lista de trabajos. Entonces podemos decir que el caso de uso “Crear lista de trabajo” depende de un caso de uso “Entrar”. Ésta es la relación que se ilustra en la Figura 1.

Un tercer estilo de conector es una línea dirigida con un triángulo hueco, al cual se le conoce como *generalización*. La palabra *generalización* en el uml significa “herencia”. Cuando mostramos una relación de generalización entre dos actores o dos casos de uso, estamos indicando que el actor o el caso de uso “hijos” son un caso del actor o uso básico y algo más. En la figura 2, se muestra una relación de generalización entre dos casos de uso.

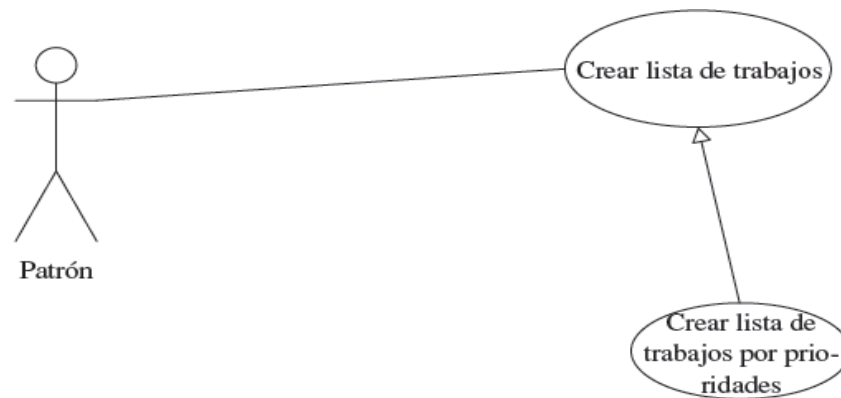


Figura 3 Diagrama de Caso de uso - conector generalización

En las relaciones de generalización, la flecha apunta hacia la cosa sobre la cual nos estamos expandiendo. Existen varias maneras en las que usted puede describir esta relación en forma verbal —acerca de la cual usted debe saber—, pero desafortunadamente, todos estos sinónimos pueden conducir a confusión verbal. Los siguientes enunciados describen las relaciones de generalización que se muestran en la Figura 2:

- El usuario es el objetivo y el patrón es la fuente.
- El patrón es un usuario.
- El usuario es el subtipo y el patrón es el supertipo.
- El patrón se hereda del usuario.
- El usuario es el tipo padre y el patrón es el tipo hijo.
- El patrón generaliza al usuario.

(En esta lista, puede sustituir la frase Crear lista de trabajos en todas partes en donde vea la palabra Usuario, y sustituir la frase Crear lista de trabajos por prioridades en todas las partes en donde vea la palabra Patrón, para transmitir la relación entre los dos casos de uso.) El último enunciado, en el cual se usa la palabra generaliza, es el más exacto en el contexto del uml, pero vale la pena reconocer que todos los enunciados son equivalentes.

I.2.2.1.1.5. Caso de uso de inclusión y de extensión

Una relación de dependencia entre dos casos de uso significa que, de alguna manera, el caso dependiente necesita al caso del que depende. Dos

estereotipos de uso común y predefinidos que refinan las dependencias en los casos de uso son el incluir y el extender.

Tomemos un minuto para ampliar nuestros comentarios de introducción sobre incluir, de la sección anterior, e introduzcamos extender.

El estereotipo extender se usa para agregar más detalle a una dependencia, lo cual significa que estamos agregando más capacidades (como ejemplo, vea la figura 3). Como se muestra en la figura, decimos que “Registrar las listas vistas” extiende (y depende de) “Ver lista”

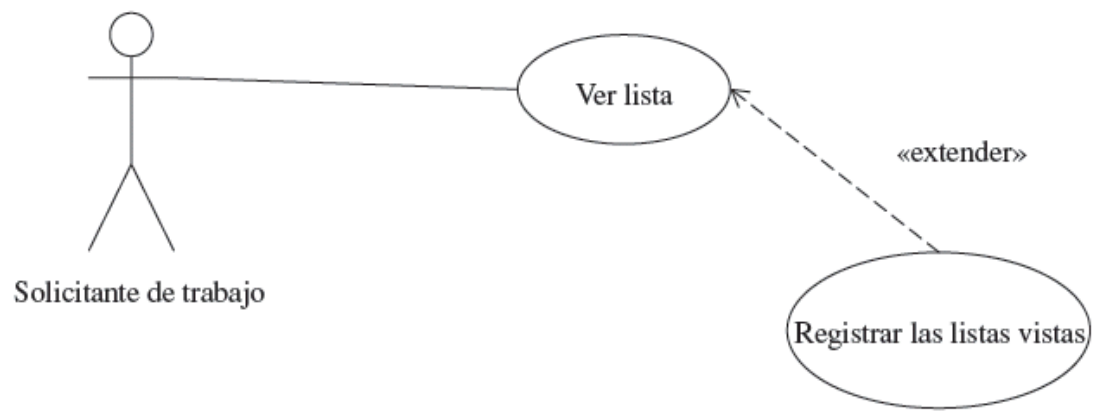


Figura 4 Diagrama de casos de uso – Extender

El estereotipo incluir significa que, finalmente, el caso de uso dependiente es para volver a usar el caso del que depende. El equipaje que va con el estereotipo incluir es que el caso de uso dependiente necesitará los servicios del caso del que depende y saber algo acerca de la realización de ésta, pero lo opuesto no es cierto.

El caso de uso del que se depende es una entidad completa y distinta que no debe depender del caso dependiente. La concesión de acceso es un buen ejemplo. Resulta claro que requerimos que un patrón tenga acceso para crear una lista de trabajos, pero también pudimos obtener acceso por otras razones.



Figura 5 Diagrama de caso de uso - Incluir

I.2.2.1.2. Diagrama de Clases

Un diagrama de clases es una descripción visual de los posibles sistemas. Un diagrama de clases y un diagrama de objetos son las alternativas de representación de modelos de objetos, aunque los diagramas de clases prevalecen más de los de objetos. Normalmente se puede construir un diagrama de clases y ocasionalmente uno de objetos para ilustrar las estructuras de datos más complejas.

Una clase captura la estructura y comportamiento común de un conjunto de objetos. Una clase es una abstracción de ítems del mundo real.

Una clase se divide en tres partes, con el nombre de la clase en la parte superior, la lista de sus atributos en la segunda y la lista de sus operaciones o métodos en la última.

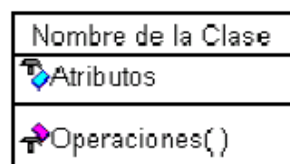


Figura 6 Diagrama de clase

Se puede definir las siguientes relaciones para una clase:

I.2.2.1.2.1. Generalización

Relación entre clase y muestra que la subclase compare la estructura o comportamiento definida en una o más superclases.

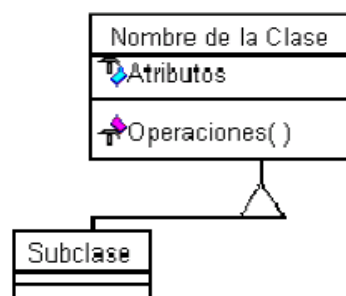


Figura 7 Generalización

I.2.2.1.2.2. Agregación

Representa una relación parte todo entre dos clases. Muestra que el objeto agregado esta físicamente construido a partir de otro objeto, o que e lógicamente lo contiene.

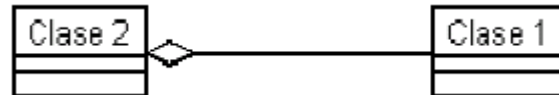


Figura 8 Agregación

Asociación: representa una conexión semántica entre dos clases. La asociación es bidireccional, es la relación más general y la más débil semánticamente.

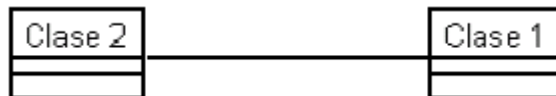


Figura 9 Asociación

El diagrama de clases es un grafo bidimensional que muestra clases, tipos, contenido, las relaciones estructurales y herencia entre los objetos modelados colocando restricciones necesarias oportunamente.

- Elementos
- Clases
- Atributos
- Operaciones
- Relaciones entre clases
- Utilidad

El diagrama de clases es utilizado para explicar pequeñas partes que contienen relaciones complicadas o recursivas.

I.2.2.1.3. Diagrama de Estados

Un diagrama de estados relaciona estados y sucesos, es decir especifica la secuencia de estados que un objeto o una interacción entre objetos pasa durante su tiempo de vida en respuesta a eventos o estímulos recibidos.

- Elementos
- Estado
- Transición
- Utilidad

Los Diagramas de Estados son utilizados para expresar concurrencia, sincronización y jerarquías de objetos.

I.2.2.1.3.1. Estado

El *estado* representa situaciones durante la vida de un *objeto*. Se representa con un rectángulo que tiene sus esquinas redondeadas.

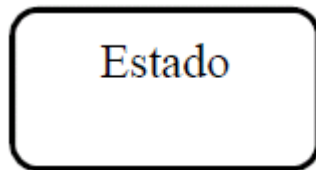


Figura 10 Estado

I.2.2.1.3.2. Transición

Una flecha representa el pasaje entre diferentes *estados* de un *objeto*. Se etiqueta con el evento que lo provoca y con la acción resultante.

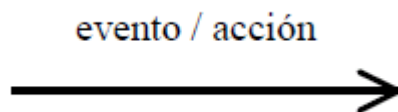


Figura 11 Transición

I.2.2.1.3.3. Estado Inicial



Figura 12 Estado Inicial

I.2.2.1.3.4. Estado Final



Figura 13 Estado Final

Modelo Funcional

Describe las transformaciones de valores de datos que ocurren dentro del sistema, representa relaciones funcionales entre valores, los cálculos existentes dentro del sistema, los valores producidos en un cálculo a partir de los valores introducidos, sin tener el orden de cálculo.

I.2.2.1.4. Diagramas de actividades

Un diagrama de actividades es la versión uml de un diagrama de flujo. Los diagramas de actividades se usan para analizar los procesos y, si es necesario, volver a realizar la ingeniería de los procesos.

Un diagrama de actividades es una herramienta excelente para analizar problemas que, al final, el sistema deberá resolver. Como una herramienta de análisis, no queremos empezar resolviendo el problema en un nivel técnico mediante la asignación de clases, pero podemos usar los diagramas de actividades para entender el problema e incluso refinar los procesos que comprenden el problema.

A cada Actividad se le presenta por un rectángulo con las esquinas redondeadas (más angosto y ovalado que la representación del estado). El procesamiento dentro de una actividad se lleva a cabo al realizarse con la siguiente actividad. Al igual que el diagrama de estado, el de actividad cuenta con un punto inicial (representando un círculo relleno) y uno final (representando por una diana).

Una secuencia de Actividad llegara a un punto donde se realizara alguna decisión. Ciertas condiciones le llevaran por un camino y otras por otro (pero ambas son mutuamente exclusivas).

I.2.2.1.4.1. Estados de Acción

Los estados de acción representan las acciones no interrumpidas de los objetos.



Figura
14 Estados
de Acción

I.2.2.1.4.2. Flujo de la Acción

Los *flujos de acción*, representados con flechas, ilustran las relaciones entre los *estados de acción*.

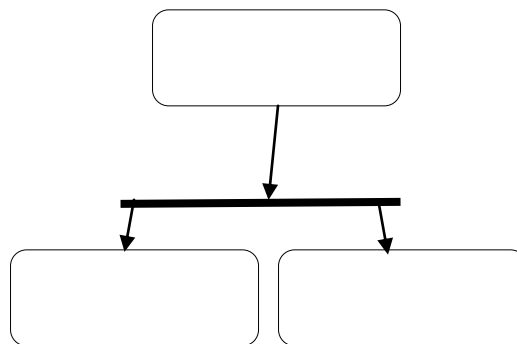


Figura 15 Flujo de la Acción

I.2.2.1.4.3. Estado Inicial

Estado inicial de un estado de *acción*.

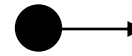


Figura 16 Estado Inicial

I.2.2.1.4.4. Estado Final

Estado final de un estado de *acción*.

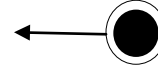


Figura 17 Estado Final

I.2.2.1.4.5. Sincronización

Una barra de sincronización ayuda a ilustrar la ocurrencia de transiciones paralelas, así quedan representadas las acciones concurrentes.

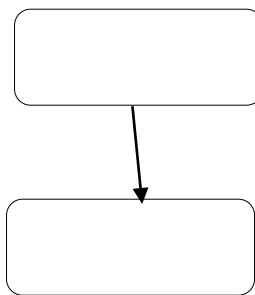


Figura 18 Sincronización

I.2.2.1.5. Diagrama de Secuencia

En estos diagramas se describe la ordenación temporal de las acciones, se relaciona con los casos de uso mostrando las acciones antes y después del mismo en orden cronológico. También se puede definir la representación del periodo durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación.

Un dialogo de secuencia posee dos dimensiones: la vertical representa el tiempo, la horizontal representa los objetos que participan en la interacción. En general, el tiempo avanza hacia abajo dentro de la página (se pueden invertir los ejes si se desea). Con frecuencia solo son importantes las secuencias de mensajes pero en aplicaciones de tiempo real el eje temporal puede ser una métrica. La ordenación horizontal de los objetos no tiene ningún significado.

Cada objeto representa una columna distinta, se pone un símbolo de objeto al final de la flecha que representa el mensaje que ha creado el objeto, está situada en el punto vertical que denota el instante en que se crea el objeto.

Este se conoce como línea de vida del objeto. Se pone una X grande en el punto en que deja de existir el objeto o en el punto en que el objeto se destruye a sí mismo. Para el periodo durante el cual este activo el objeto, la línea de vida se amplía para ser una línea doble continua. Si el objeto se llama a sí mismo, entonces se superpone otra copia de la doble línea para mostrar la doble activación. El orden relativo de los objetos no tiene significado, aun cuando resulta útil organizarlos de modo que se minimice la distancia de las flechas.

Cada mensaje representa mediante una flecha horizontal que va desde la línea de vida del objeto que envió el mensaje hasta la línea de vida del objeto que ha recibido el mensaje. Si un mensaje requiere un cierto tiempo para llegar a su destino, entonces la flecha del mensaje se dibuja diagonalmente hacia abajo.

- Elementos
- Objetos
- Operación
- Mensajes
- Utilidad

Son sumamente importantes en el modelado de la aplicación por mostrar transición entre objetos y los mensajes generados de acuerdo al tiempo de ejecución de las operaciones.

I.2.2.1.5.1. Mensaje

Un mensaje que va de un objeto a otro pasa de línea de vida de un objeto a la de otro. Un objeto puede enviarse un mensaje a sí mismo (es decir, desde su línea de vida hacia su propia línea de vida).

Un mensaje puede ser simple, sincrónico o asincrónico. Un mensaje simple es la transparencia del control de un objeto a otro. Si un objeto envía un mensaje sincrónico, espera la respuesta a tal mensaje antes de continuar con su trabajo. Si un objeto envía un mensaje asincrónico, no espera una respuesta antes de continuar. En el diagrama de secuencias, los símbolos del mensaje varían, la punta de la flecha de un mensaje simple está formada por dos líneas

la punta de la flecha de un mensaje sincronico esta rellena y la de un asincrónico tiene una sola línea.

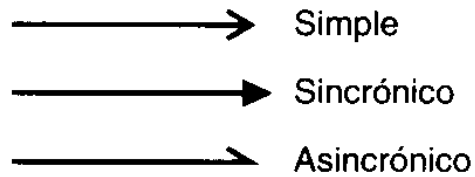


Figura 19 Mensaje

I.2.2.1.5.2. Tiempo (Lineas de Vidas)

El diagrama representa al tiempo en dirección vertical. El tiempo se inicia en la parte superior y avanza hacia a parte inferior. Un mensaje que este mas cerca de la parte superior ocurrirá antes que uno que este cerca de la parte inferior.

El diagrama de secuencia dos dimensiones. La dimensión horizontal es la disposición de los objetos y la dimensión vertical muestra el paso del tiempo.

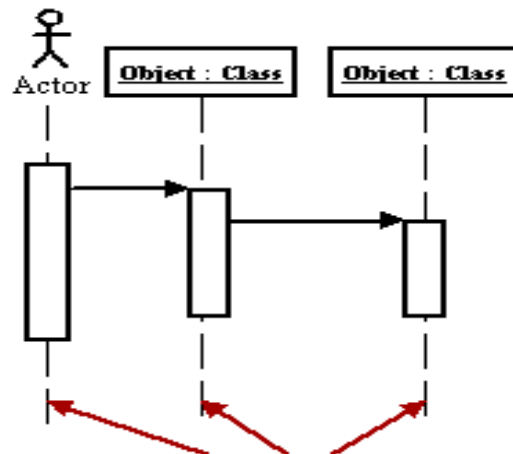


Figura 20 Tiempo o Lineas de vidas

Tiempo o Lineas de Vida

I.2.2.1.5.3. Rol de la Clase

El *rol* de la *clase* describe la manera en queun *objeto* se va a comportar en el contexto.

No se listan los atributos del *objeto*.

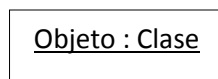


Figura 21Rol de la Clase

I.2.2.1.5.4. Activación

Los cuadros de *activación* representan el tiempo que un *objeto* necesita para completar una tarea.

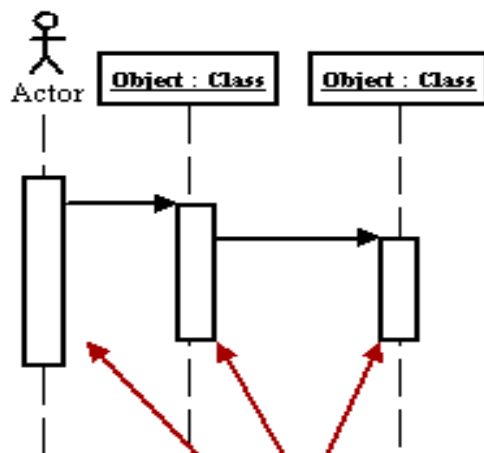


Figura 22 Activacion

Activación

I.2.3. Herramientas de Diseño y Desarrollo de Software

I.2.3.1. Herramientas CASE

Computer Aided/Assisted Software/System Engineering

Las herramientas CASE son un conjunto de programas (métodos, utilidades y técnicas) que brindan asistencia durante la automatización del proceso de desarrollo de un sistema por medio del uso de metodologías formales, descritas con el lenguaje de modelado unificado.

Entre sus objetivos están:

- Dar soporte a los analistas, diseñadores, desarrolladores e ingenieros de software en la representación clara y precisa de los diagramas del ciclo de vida o proceso de desarrollo de software
- Mostrar de acuerdo a la metodología seleccionada barras de herramientas con los posibles diagramas y sus elementos.
- Emplear más tiempo en análisis y diseño, y disminuirlo en codificación y pruebas.
- Ayudar en la realización, mantenimiento, actualización y estandarización de la documentación del software. Ej. Diccionario de Datos
- Mejorar la calidad, utilidad, fiabilidad, rendimiento y tiempo en el desarrollo de los sistemas.

- Incrementar la productividad de los proyectos con la aplicación de conceptos como: generación y reutilización de código.

I.2.3.1.1. Bases de Datos

Un Sistema de Gestión de Base de Datos (SGBC), también conocido como Data Management System (DBMS), es una estructura en donde se almacenan grandes cantidades de datos, si esta se basa en las relaciones entre tablas de un modelo se trata de una RDBMS: Relational Data Base Management System.

I.2.3.1.1.1. SQL SERVER 2008³

SQL Server es una base de datos relacional para aplicaciones cliente/servidor, distribuidas y Web que posee altas cualidades empresariales. De fácil uso en la construcción, administración e implementación de aplicaciones de negocios por su interfaz gráfica que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.

Características

- Agilidad en las operaciones de análisis y administración de datos
- Fácil acceso a los datos a través del Web, se puede utilizar http para enviar consultas a la base de datos.
- Disponibilidad por contar con el gestor de errores mejorados y estrategias de copia de seguridad.
- Tiene la capacidad para optimizar y depurar consultas interactivamente, mover, transformar rápidamente datos de cualquier origen y utilizar funciones como si estuvieran integradas en Transact-SQL
- Administración multiservidor y con una sola consola, ejecución y alerta de trabajos basadas en eventos, seguridad integrada y scripting administrativo.
- Automatizar tareas de rutina, mediante programación de tareas.
- Características de configuración y sincronización automática.
- SQL Server es un motor de base de datos que puede ser instalado tanto en un equipo portátil como en un servidor con múltiples procesadores.
- Mantienen niveles altos de seguridad y confiabilidad.
- Capacidades de replicación de multi-sitios.
- Manejo de consultas complejas y bases de datos muy grandes, procesamiento analítico en línea e integrado.

³<http://technet.microsoft.com/es-es/library/bb934498%28v=sql.110%29.aspx>

- Obtención de copias de seguridad online garantizando la consistencia de datos para la más alta disponibilidad.

Administración de base de datos

La administración de las bases de datos se ha simplificado por la existencia de los asistentes en el Administrador Corporativo (Enterprise Manager). Adicionalmente automatiza las tareas de backups, chequeos de consistencia y mantenimiento de índices.

Tipos de Datos SQL Server

Identificador	Tipo de Dato
Int	Números enteros
Decimal	Datos de precisión y escala numérica fijas
Flota	Numero con precisión de con flotante
Char	Datos de caracteres no Unicode de longitud fija

Cuadro 1 Tipos de Datos SQL Server

Ventajas

- Rendimiento
- Velocidad
- Escalabilidad
- Confiabilidad
- Rápida y fácil adaptación convirtiéndose en una ventaja competitiva.
- Fácil administración de base de datos de forma centralizada y permanencia en línea mientras se mueve y copia fácilmente bases de datos entre equipos o instancias⁵.

Desventajas:

- Posee plataforma Microsoft por tanto su instalación se la realiza sobre la plataforma Windows.

I.2.3.1.2. Navicat⁴

Navicat es un excelente programa informático capaz de gestionar los componentes gráficos, la administración y el desarrollo de sistemas de bases de datos en SQL server u otros servidores que estén instalados en tu ordenador u otro ordenador.

Este programa incluye las siguientes características: realiza conexiones seguras a través de las redes WLAN, LAN e Internet, importa y exporta con múltiples formatos compatibles con el propio programa, eficaz asistente para consultas, poderoso diseñador de reportes, genera informes, crea tareas como si fueran eventos, realiza copia de seguridad de los datos, soporta cualquier versión de SQL server, activadores (triggers), procedimientos almacenados y funciones, gestión de las distintas vistas, administración para diferentes usuarios y control de eventos, soporte para caracteres, cambio de muestras de fuentes, ayuda en cada versión de Windows que sea compatible con el propio programa, entre otras.

¿Para qué sirve Navicat?

Navicat sirve, fundamentalmente, para modificar las propiedades y las características de las tablas existentes, crear tablas nuevas, determinar los permisos específicos para cada usuario que utilice tu ordenador o servidor, realizar diversas copias de seguridad, generar informes y ejecutar tus propias consultas, entre otras.

Integración con Windows

Se integra perfectamente con algunas versiones de Windows (Windows XP y Windows Vista). No se ofrece información sobre su compatibilidad con Windows 7, Windows 8, MAC y Linux.

En constante evolución

Navicat presenta una interfaz simple, intuitiva y muy homogénea. De hecho, es capaz de trabajar con múltiples DBMSs, con ediciones para MySQL, SQL Server, SQLite, Oracle y PostgreSQL y su versión Premium permite conectarse con todos los DBMSs con la misma interfaz.

Conclusión

Navicat es, por tanto, el software ideal y perfecto para la administración y el desarrollo de bases de datos SQL server y puede ser utilizada tanto por usuarios principiantes como por expertos.

Navicat tiene los siguientes requisitos:

⁴<http://jumipe.wordpress.com/2013/10/28/navicat-mysql-tutorial-parte-1/>
<http://blog.unijimpe.net/tutorial-de-navicat/>

Requiere de una conexión a Internet correctamente configurada para su instalación.

Características:

- Gestiona los componentes gráficos, la administración y el desarrollo de sistemas de bases de datos en SQL server ya instalados en tu ordenador o servidor a utilizar.
- Realiza conexiones seguras a través de las redes WLAN, LAN e Internet.
- Importa y exporta con múltiples formatos compatibles con el propio programa
- Genera informes y crea tareas como si fueran eventos.
- Realiza copia de seguridad de los datos y soporta cualquier versión de SQL server.
- Gestión de las distintas vistas y administración para diferentes usuarios y control de eventos.
- Incluye menú de ayuda para cada versión de Windows que sea compatible con el propio programa.
- Interfaz simple, intuitiva y muy homogénea.
- Requiere de una conexión a Internet correctamente configurada para su instalación.
- Compatible con algunas versiones existentes de Windows.

I.2.3.2.3. Enterprise Architect⁵:

Enterprise Analyst es una herramienta para la compilación, análisis estático, ejecución y documentación de los modelos UML.

- Compilación para asegurar la notación precisamente establecida de un modelo de clases
- Análisis estático para establecer control sobre la estructura interna y dependencias en el modelo
- Ejecución (o análisis dinámico) para probar y validar el comportamiento del modelo de clases bajo distintos escenarios
- Documentación para generar las especificaciones necesarias para las tareas de diseño e implementación de un sistema de software

Enterprise Analyst permite capturar, analizar, validar y documentar el modelo de análisis y los requerimientos funcionales de un sistema de software

⁵<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=EnterpriseArchitectUML2.x>
<http://www.sparxsystems.com.ar/resources/tutorial/uml-tutorial.html>
<http://www.notodocodigo.com/blog/primeros-pasos-con-enterprise-architect/>

Los usuarios principales del Enterprise Analyst son los **analistas del sistema**. Esta herramienta permite a los analistas concentrarse en los aspectos funcionales del sistema y asegurar su corrección en la etapa temprana de un proyecto de desarrollo.

Enterprise Analyst permite generar rápidamente los prototipos funcionales de un sistema de software.

Enterprise Analyst es un plug-in de la herramienta Enterprise Architect.

I.2.3.2.3.1. Modelamiento con UML y Enterprise Analyst

Enterprise Analyst está basado en dos paradigmas:

- Lenguaje UML, Desarrollo orientado a Modelos y Arquitectura Dominio-Frontera

El Lenguaje Unificado de Modelamiento o UML (Unified Modeling Language) es el lenguaje estándar para el modelamiento de los sistemas de software.

I.2.3.2.3.2. Diagramas Estructurales:

- Clases
- Objetos
- Estructura Compuesta
- Componentes
- Despliegue
- Paquetes

I.2.3.2.3.3. Diagramas de Comportamiento:

- Casos de Uso
- Estados
- Actividades
- Secuencias

I.2.3.2.4. Wireframe⁶

Un wireframe para un sitio web, también conocido como un esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un sitio web. El wireframe esquematiza el diseño de página u ordenamiento del contenido del sitio web, incluyendo elementos de la interfaz y sistemas de navegación, y cómo funcionan en conjunto. Usualmente, este esquema carece de estilo tipográfico, color o aplicaciones gráficas, ya que su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de contenidos. En otras palabras, se enfoca en “qué hace la pantalla, no cómo se ve.” Los esquemas pueden ser dibujados con lápiz y papel o esquemas en una pizarra, o pueden ser producidos con medios de diseño de aplicaciones de software libre o comercial.

Los wireframes se enfocan en:

Los tipos de información que será mostrada

La cantidad de las funciones disponibles

Las prioridades relativas de la información y las funciones

Las reglas para mostrar ciertos tipos de información

El efecto de los distintos escenarios en la pantalla

El wireframe del sitio web conecta la estructura conceptual, o arquitectura de la información, con la superficie, o diseño visual del sitio web. Los wireframes ayudan a establecer funcionalidad, y las relaciones entre las diferentes plantillas de pantallas de un sitio web. Un proceso iterativo de creación de wireframes es una forma efectiva de hacer prototipos de páginas rápidos, mientras se mide la practicidad de un concepto de diseño. Típicamente, la esquematización comienza entre “diagramas de flujo de estructuras de trabajo de alto nivel o mapas de sitio y diseños de pantallas.” Dentro del proceso de construcción de un sitio web, el dibujo de un wireframe es donde el concepto se vuelve tangible.

A parte de los sitios web, los wireframes son usados para hacer prototipos de sitios móviles, aplicaciones para ordenador, u otros productos basados en pantalla que impliquen interacción hombre-máquina. Las tecnologías futuras y los medios forzarán a los wireframes a adaptarse y evolucionar.

⁶<http://wireframesketcher.com/>
<http://www.taringa.net/posts/linux/15967798/WireframeSketcher-3-8-1.html>

I.2.3.1.5. Spring Framework⁷

Spring es un framework de aplicaciones Java/J2EE desarrollado usando licencia deOpenSource.

Se basa en una configuración a base de javabeans bastante simple. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable. Es interesante el uso de programación orientada a aspectos (IoC). Tiene plantillas que permiten un más fácil uso de Hibernate, iBatis, JDBC..., se integra "de fábrica" con Quartz, Velocity, Freemarker, Struts, Webwork2 y tienen un plugin para eclipse.

Ofrece un ligero contenedor de bean para los objetos de la capa de negocio, DAOs y repositorio de Datasources JDBC y sesiones Hibernate. Mediante un xml definimos el contexto de la aplicación siendo una potente herramienta para manejar objetos Singleton o "factorias" que necesitan su propia configuración.

El objetivo de Spring es no ser intrusito, aquellas aplicaciones configuradas para usar beans mediante Spring no necesitan depender de interfaces o clases de Spring, pero obtienen su configuración a través de las propiedades de sus beans.

Este concepto puede ser aplicado a cualquier entorno, desde una aplicación J2EE a un applet.

Como ejemplo podemos pensar en conexiones a base de datos o de persistencia de datos, como Hibernate, la gestión de transacciones genérica de Spring para DAOs es muy interesante.

La meta a conseguir es separar los accesos a datos y los aspectos relacionados con las transacciones, para permitir objetos de la capa de negocio reutilizables que no dependan de ninguna estrategia de acceso a datos o transacciones.

Spring ofrece una manera simple de implementar DAOs basados en Hibernate sin necesidad de manejar instancias de sesion de Hibernate o participar en transacciones. No necesita bloques "try-catch", innecesario para el chequeo de transacciones. Podríamos conseguir un método de acceso simple a Hibernate con una sola línea.

Spring es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. La primera versión fue escrita por Rod Jonhson. Es una plataforma Java que otorga una infraestructura de apoyo global al desarrollo de

⁷<http://www.uv.es/grimo/teaching/SpringMVCv4PasoAPaso/index.html>
<http://www.davidmarco.es/spring-mvc>

aplicaciones Java. De este modo, Spring se encarga de la infraestructura para que nosotros nos centremos en la aplicación.

Unos ejemplos pueden ser:

- Inyección de dependencia e inversión de control.
- Integración del acceso a datos.
- Facilitar el desarrollo de aplicaciones web separando claramente las partes del modelo, vista y controlador.
- Poder ejecutar métodos transaccionales en una base de datos sin necesidad de lidiar con API de transacción, métodos remotos sin tener que lidiar con API de procedimientos remotos, métodos de gestión sin JMX, control de mensajes sin JMS.

¿Que proporciona?

Spring proporciona:

- Una potente gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control (IoC). Esto hace que la configuración de aplicaciones sea rápida y sencilla. Ya no es necesario tener singletons ni ficheros de configuración, una aproximación consistente y elegante. Estas definiciones de beans se realizan en lo que se llama el contexto de aplicación.
- Una capa genérica de abstracción para la gestión de transacciones, permitiendo gestores de transacción añadibles (pluggables), y haciendo sencilla la demarcación de transacciones sin tratarlas a bajo nivel. Se incluyen estrategias genéricas para JTA y un único JDBC DataSource. En contraste con el JTA simple o EJB CMT, el soporte de transacciones de Spring no está atado a entornos J2EE.
- Una capa de abstracción JDBC que ofrece una significativa jerarquía de excepciones (evitando la necesidad de obtener de SQLException los códigos que cada gestor de base de datos asigna a los errores), simplifica el manejo de errores, y reduce considerablemente la cantidad de código necesario.
- Integración con Hibernate, JDO e iBatis SQL Maps en términos de soporte a implementaciones DAO y estrategias con transacciones. Especial soporte a Hibernate añadiendo convenientes características de IoC, y solucionando muchos de los comunes problemas de integración de Hibernate. Todo ello cumpliendo con las transacciones genéricas de Spring y la jerarquía de excepciones DAO. Funcionalidad AOP, totalmente integrada en la gestión de configuración de Spring. Se puede aplicar AOP a cualquier objeto gestionado por Spring, añadiendo aspectos como gestión de transacciones declarativa.

Con Spring se puede tener gestión de transacciones declarativa sin EJB, incluso sin JTA, si se utiliza una única base de datos en un contenedor Web sin soporte JTA.

Un framework MVC (Model-View-Controller), construido sobre el núcleo de Spring. Este framework es altamente configurable vía interfaces y permite el uso de múltiples tecnologías para la capa vista como pueden ser JSP, Velocity, Tiles, iText o POI. De cualquier manera una capa modelo realizada con Spring puede ser fácilmente utilizada con una capa web basada en cualquier otro framework MVC, como Struts, WebWork o Tapestry.

Toda esta funcionalidad puede usarse en cualquier servidor J2EE, y la mayoría de ella ni siquiera requiere su uso. El objetivo central de Spring es permitir que objetos de negocio y de acceso a datos sean reutilizables, no atados a servicios J2EE específicos. Estos objetos pueden ser reutilizados tanto en entornos J2EE (Web o EJB), aplicaciones “standalone”, entornos de pruebas, etc.... sin ningún problema.

La arquitectura en capas de Spring ofrece mucha de flexibilidad. Toda la funcionalidad está construida sobre los niveles inferiores. Por ejemplo se puede utilizar la gestión de configuración basada en JavaBeans sin utilizar el frameworkMVC o el soporte AOP.

I.2.3.1.5.1. Inyección de dependencia e inversión de control

El término “Aplicación Java” es un término tan amplio que podría ir desde un applet hasta aplicaciones empresariales en servidores de nivel n.

Java proporciona una gran cantidad de herramientas para desarrollar aplicaciones, pero carece de medios para organizar los elementos. Normalmente es el arquitecto Java el que se encarga de esta tarea pudiendo utilizar patrones.

La inversión de control de Spring lo que hace es preocuparse de proporcionar una manera formal de creación de componentes dispares de una manera homogénea y de una manera funcional. Codifica componentes que se integran en las aplicaciones.

Diversas instituciones y empresas eligen Spring como una manera de diseñar aplicaciones robustas fáciles de mantener.

I.2.3.1.5.2. Módulos⁸

El framework de Spring consiste en elementos organizados en veinte módulos. Estos módulos se agrupan en el Contenedor (core container), Acceso a datos e integración, modelo vista controlador (módulo web MVC), aspectos (AOP), instrumentación y test.

El siguiente diagrama muestra cómo se divide:

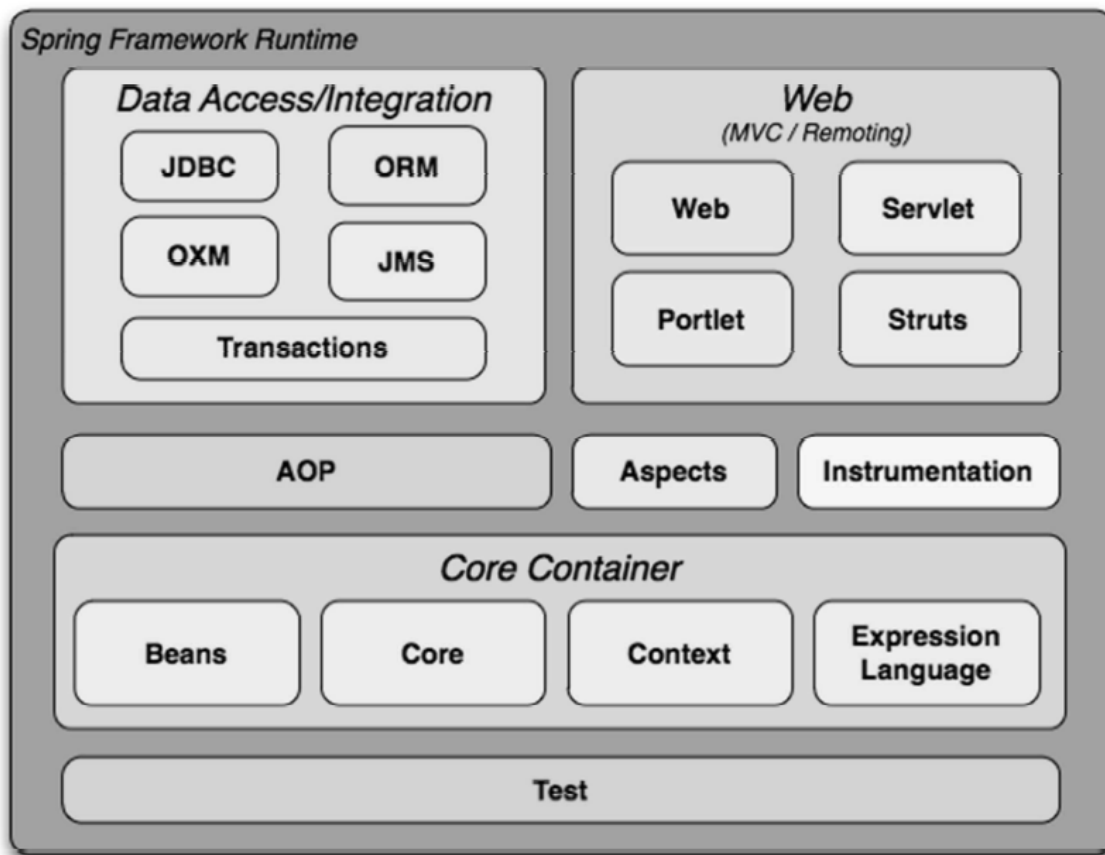


Figura 23 Módulos de Spring Framework

El contenedor consiste en un núcleo, objetos bean, un contexto y un lenguaje de expresiones.

⁸http://centrodeartigo.com/articulos-informativos/article_69801.html
<http://docs.spring.io/spring/docs/current/spring-framework-reference/html/overview.html>

El núcleo y los beans son la parte fundamental de Spring, incluyendo la inversión de control y la inyección de dependencia. Este contenedor es una versión más compleja del patrón Factory.

Elimina la necesidad de programar singletons y permite desacoplar la configuración y especificación de dependencias de la lógica de programación.

El contexto se construye sobre la sólida base del núcleo. Así permite determinadas configuraciones. Así la internacionalización, propagación de eventos, lectura de recursos o la creación de contextos (como el web) formarán parte de este módulo.

Los Lenguajes de expresión permiten una potente herramienta de consulta y manipulación de un objeto en tiempo de ejecución. Es una extensión del “Unified EL”, especificado en la especificación JSP 2.1. El lenguaje permite asignar y obtener valores de las propiedades, asignar propiedades, invocar métodos, acceder al contexto de matrices, listas, índices, operadores aritméticos y lógicos, variables, y obtención de objetos por nombre del contenedor de Spring.

I.2.3.1.5.4. Integración y acceso de Datos

La capa de Integración y acceso a datos consiste en la integración de los módulos JDBC, ORM, OXM, JMS y de transacción.

El módulo JDBC otorga una capa de abstracción que elimina la necesidad de crear código tedioso y transforma las excepciones generadas por el proveedor de base de datos.

El módulo ORM otorga una integración con los APIs más populares de ORM como puedan ser JPA, JDO, Hibernate o iBatis.

El módulo OXM otorga una capa de abstracción para el mapeo Objeto/XML en distintas implementaciones como JAXB, Castor, XMLBeans, JiBX o XStream.

El módulo JMS contiene características para la producción y consumo de mensajes.

El módulo de Transacción permite transacciones programáticas y declarativas para las clases que implementan interfaces especiales y para todos los POJO.

I.2.3.1.5.5. Web

La capa web consiste en los módulos Web, Web-Servlet, Web-Struts y Web-Portlet.

El módulo Web permite integración básica de características como la subida multiparte de unfichero, la inicialización de la inversión de control del contenedor usando listeners Servlet y uncontexto de lógica web.

El módulo Servlet contiene la implementación modelo vista controlador. El framework SpringMVC permite una separación entre el modelo, el código y los formularios web y se integra con todas las otras características de Spring.

El módulo Web-Struts permite la integración de clases integrando Struts, actualmente este soporte está obsoleto en Spring 3.0.

El módulo Web-Portlet permite las características web en sistemas empujados.

I.2.3.1.5.6.AOP

El módulo AOP de Spring permite una implementación de programación orientada a aspectos permitiendo definir métodos e interceptores, puntos de corte, etc. Para desacoplar el código.

Permite la integración con AspectJ.

El módulo de instrumentación otorga instrumentación de clases así como un cargador de clases a ser usadas en determinadas aplicaciones de servidor.

I.2.3.1.5.7.Test

El módulo de test permite probar las aplicaciones de Spring y los componentes con JUnit o TestNG. Permite la carga consistente de contextos de Spring. Así se permiten objetos mock que prueban tu código de manera aislada.

I.2.3.1.6. Jasperreports

JasperReports es una herramienta de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML.

Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico.

Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible.

JasperReports se usa comúnmente con iReport, un front-end gráfico de código abierto para la edición de informes.

Se encuentra bajo licencia libre GNU, por lo que es Software libre. Forma parte de la iniciativa apilada open source Lisog.

I.2.3.1.7 Apache Tomcat⁹

Apache Tomcat, o simplemente Tomcat, es un contenedor de servlets o programas que ofrecen funciones suplementarias a un servidor. Incluye el compilador Jasper, que compila las JSPs convirtiéndolas en servlets. A menudo se presenta como motor de servlets en combinación con un servidor Web (generalmente Apache), aunque también puede funcionar como servidor Web por sí mismo, eso sí, con peor rendimiento. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual.

Las JavaServer Pages son los elementos básicos de una tecnología desarrollada por

Sun Microsystems, basada en el procesamiento de scripts Java y que permite a los desarrolladores generar dinámicamente páginas web (HTML, XML u de otro tipo).

Para posibilitar el acceso a una base de datos desde una página JSP haciendo uso de

JDBC se necesita de un driver o controlador, que de no estar ya incluido en las librerías de la máquina virtual, pudiera hacer necesaria su adquisición.

I.2.4. El Proyecto

I.2.4.1. Descripción y Fundamentación del Proyecto

La empresa Cosaalt de prestación de servicio público a la sociedad tarijeña nace en septiembre de 1986, con el transcurrir el tiempo la empresa creció tanto económicamente.

I.2.4.1.1. Plan Operativo Anual

El proceso de formulación del POA , constituye su instrumento anual de planificación que permite concretizar los objetivos establecidos en el proyectos y actividades que se ejecutarán en el período de un año, por lo que tiene una relación directa con el Presupuesto ,COSAALT realiza su formulación de forma manual con la ayuda de hojas de cálculo (Excel) y un paquete (software) que permite organizar las actividades y sub actividades que tienen una fecha de inicio y un tiempo de vida también cuentan con un monto económico el cual es llamado presupuesto.

⁹<http://programacion.net/articulo/tomcat - introduccion 134>

Un POA de un área pasa por 3 etapas:

- ✓ **Formulación:** es proceso de Elaboración o Presupuestación, COSAALT. no cuenta con una pre-planeación que proporciona las directrices para la preparación de los presupuestos, cuantificar los programas operativos definidos en el POA, asignando unidades físicas expresadas en valores monetarios según: La formulación implica la cuantificación financiera de los programas operativos teniendo los servicios los niveles de inversión requeridos, para el cálculo de financiamiento requerido hacen una comparación de la gestión anterior, en el reajuste del presupuesto de un POA que es una operación delicada, lo que hacen es reducir o quitar algunas actividades que estaba solicitado y previsto, esto hace que haya una falta de coordinación entre la área administrativa y las demás áreas que conforma COSAALT. No existe una distribución de los correspondientes montos presupuestarios y actividades.
- ✓ **Seguimiento y evaluación:** Son herramientas importantes para fortalecer los procesos de planeación, a través de la retroalimentación y posibilitar su alcance en procura de una mayor efectividad, pero no cuenta con el proceso de retroalimentación en la formulación de POA de las diferentes áreas. Para ello las áreas deberían reunirse para discutir, análisis y mejorar las propuestas sobre los diferentes momentos del proceso de planificación y evaluación de cada área, falta de coordinación Institucional de las áreas. A la falta de coordinación y retroalimentación se podrá modificar la formulación presupuestaria de los POA de cada área.

Retardos en la elaboración en base al histórico anual de las compras realizadas durante la gestión actual, considerando además todas aquellas adquisiciones que son recurrentes año con año, en cada una de las unidades administrativas.

- ✓ **Ejecución y control:** es un proceso que da su visto bueno a las solicitudes de desembolso y la imputación de ingresos y gastos ejecutados a las partidas presupuestarias correspondientes, en este proceso hace la revisión y autorización: primero del departamento presupuestal para verificar que el recurso haya sido programado y segundo por los responsables de la administración financiera para comprobar la disponibilidad de los recursos. Cada área no cuenta con un tope de presupuesto asignado a las áreas de las planificaciones del presupuesto.

I.2.4.1.2. Adquisición de Bienes y servicios

La función compras, requiere una amplia variedad de procedimientos de operación para realizar las actividades diarias normales. Los procedimientos se encuentran reflejados en un Manual de Procedimientos que cada área de COSAALT elaborará en función a su estructura administrativa. Existen muchas razones que justifican el desarrollo de los procedimientos a ser expresados en un manual, entre las cuales tenemos:

- El elevado número de transacciones que habitualmente se efectúan, solicitando distintos artículos y servicios en una are. A
- La gran cantidad de dinero involucrado.
- La necesidad de una auditoria que permita verificar y evaluar las operaciones de compras efectuadas.
- Las consecuencias severas de un desempeño insatisfactorio.
- La contribución potencial a las operaciones efectivas de la organización, relacionadas con el funcionamiento del departamento de compras.

Con el propósito de dar un panorama general sobre las etapas sobresalientes en el proceso para la realización de compras, mencionaremos las siguientes:

- a) Reconocimiento de una necesidad.- representa el inicio del proceso de la compra, el cual surge ante una necesidad de abastecer el departamento de almacenes o, para cubrir un requerimiento de una unidad solicitante. Aquí se deberá analizar con precisión lo que se debe adquirir, la cantidad y, si es la época adecuada para hacerlo.
- b) Descripción de la necesidad.- implica una definición exacta de las características del artículo o del servicio que se requiere, expresadas en la solicitud de compra, la misma que podrá ser remitida por el departamento de almacenes u otro departamento. La solicitud de compra deberá contar con la verificación de los responsables del presupuesto de cada área, los que deben dar la conformidad acerca de la existencia de los recursos suficientes, en la partida que corresponda. El departamento de presupuestos también determinará la fuente de financiamiento de la compra solicitada, antes de ser remitida al departamento de compras.
- c) Análisis de posibles fuentes de abastecimiento.- una vez recibida la solicitud de compra, se procederá al análisis de la información disponible, para determinar el procedimiento de su abastecimiento. Para este objeto, se

recurrirá a lista de proveedores, registro de cotizaciones, catálogos, consultas de internet, etc.

d) Determinación del precio.- para este efecto se debe proceder a la obtención de cotizaciones. Esto consiste en efectuar consultas a distintos proveedores para obtener precios adecuados a través de pedidos de cotización.

e) Seleccionar el proveedor.- en función a la evaluación de los oferentes, se tomará la decisión para elegir al que presenta la mejor alternativa.

f) Asignar la orden de compra.- en función a la selección del proveedor se cerrará el contrato mediante la confección de la orden de compra o, la elaboración de un contrato si el caso amerita.

g) Seguimiento y expedición de la orden.- consiste en efectuar un seguimiento de la operación de compras, con el propósito de tener la certeza de que el proveedor entregará los materiales en la fecha acordada.

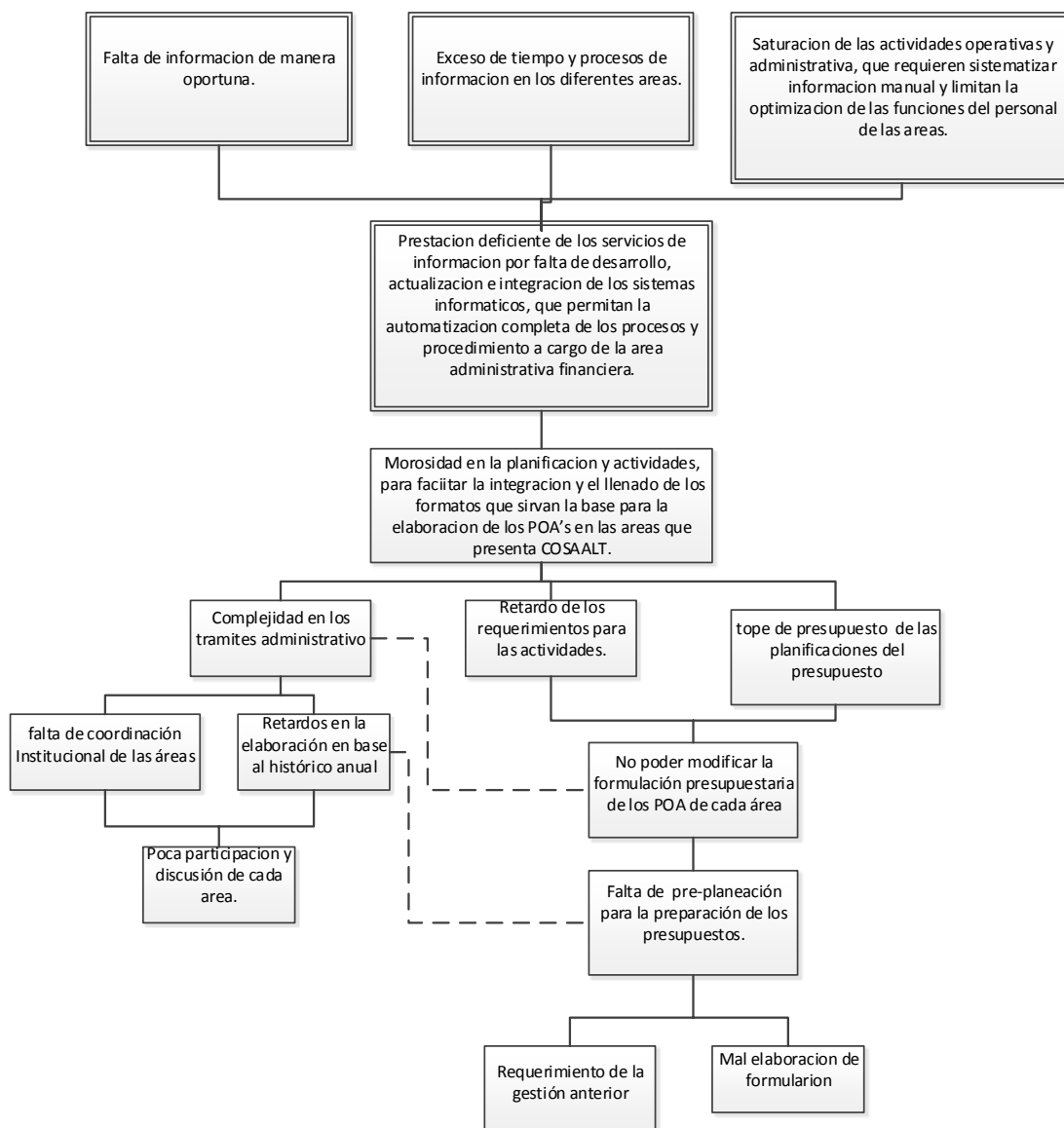
h) Recepción e inspección.- en el ingreso de los bienes se originará un control de los mismos, en base a la documentación de cada área solicitada, con el remito del proveedor, se procederá a confeccionar el informe de recepción.

i) Entrega de los materiales.- una vez que se cuentan con la conformidad de la recepción de los bienes, éstos son entregados al departamento de almacén o directamente a la unidad solicitante.

j) Liquidación y pago al proveedor.- efectuado el control de la documentación existente (remito del proveedor, informe de recepción, etc.) con los cargos formulados por el proveedor (facturas, notas de débito, etc.), se procederá a remitir el conjunto de antecedentes del área responsable de procesar la liquidación y pago al proveedor.

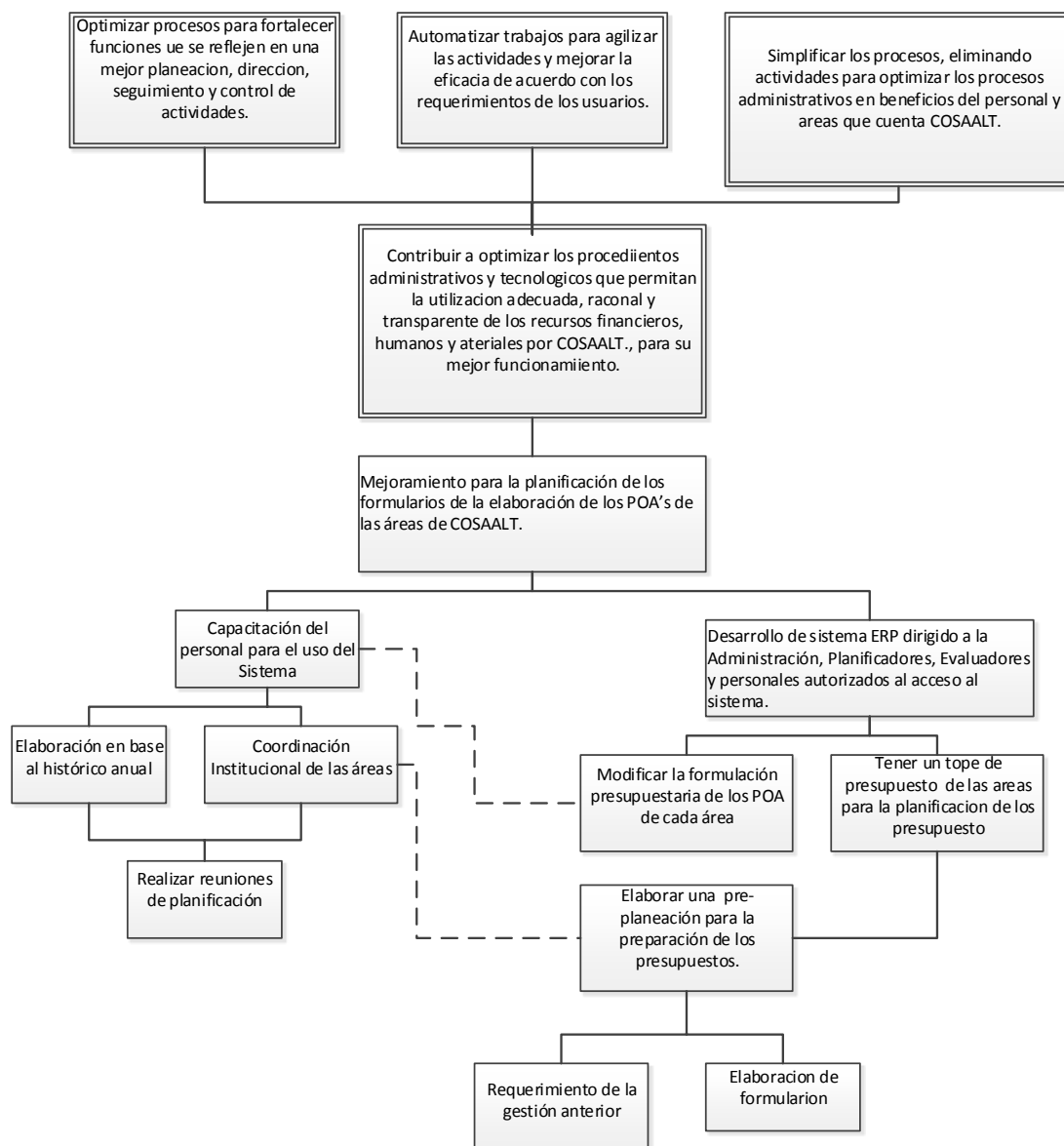
Esta descripción breve de las extensas características que contienen los procedimientos de compras, sintetizan las distintas etapas indispensables en la adquisición de bienes y servicios de COSAALT.

I.2.4.2. Análisis de Causas del Problemas



Cuadro 2 Análisis de Causas del Problemas

I.2.4.3. Análisis de objetivos



Cuadro 3 Análisis de objetivos

I.2.4.4. Matriz Lógico del Proyecto

RESUMEN NARRATIVO	INDICADORES VERIFICABLES OBJETIVAMENTE	MEDIOS DE VERIFICACIÓN	SUPUESTOS
Fin Contribuir con la inserción de las TICla optimizaciónde los procedimientos administrativos y tecnológicos que permitan la utilización adecuada, racional y transparente de los recursos financieros, humanos y materiales por Cosaalt, para su mejor funcionamiento.	Para el final del año 2015 después de la ejecución del proyecto, el 70% ¹⁰ de los trabajadores, verán las mejoras de la aplicación del nuevo sistema desde el punto de vista personal e institucional.	Informe estadístico de encuestas realizadas a una muestra del personal, para medir la satisfacción, eficiencia y eficacia del sistema.	Se dispondrán de los recursos financieros y humanos necesarios para alcanzar los objetivos y las metas propuestas.
Propósito Mejoramiento para la planificación de los formularios de la elaboración de los POA's de las áreas de COSAALT.	Culminado el Sistema Informático y la capacitacion para la Planificación de Formularios de COSAALT, la susceptibilidad de errores disminuye en un 70% en la elaboración de formularios, de los empleados de la empresa, basado en comparaciones de la gestión o mes anterior.	Informes que comparen el rendimiento de eficacia de la gestión que se implementa el sistema con una gestión anterior.	Presupuesto suficiente para la implementación y uso del Sistema Web.
Componentes 1. Desarrollo de sistema ERP dirigido a la Administración, Planificadores, Evaluadores y personales autorizados al acceso al sistema.	1.1 A finales de diciembre del 2014 el 60% ¹¹ del personal de COSAALT participaron de una conferencia de capacitación en la que se explicó el uso y las ventajas del Sistema.	Certificación de aprobación por parte Asesor del proyecto. Documentación los	El Sistema Informático terminado y aceptado por el docente de Taller III.

¹⁰ Los funcionarios responsable que utilizaran el sistema / total de funcionarios de todas las áreas = 70%

¹¹ Los encargados de cada area que participaron en la conferencia / total de encargados de cada área = 60%

Cuadro 4Matriz Lógica del Proyecto

2. Capacitación del personal para el uso del Sistema	2.1 A finales de diciembre del 2014 los Encargados de Planificacion del POA, personal de COSAALT participaran una capacitación en la que se explicara el uso y las ventajas del Sistema con una duración de 1 hora.	Requerimientos. Certificado de capacitación del personal sobre la explotación del sistema, avalado por el Director del Proyecto	El personal tendrá interés en la capacitación del Sistema El personal involucrado de Planificación deberá tener interés en realizar pruebas en el sistema.																																																									
Actividades Backup Capacitación sobre el software que soporta la base de datos. Importación automática de datos establecidos. Sistema Informático 1.1 Recopilación de requerimientos 1.2 Análisis y Diseño del Sistema. 1.3 Programación del Sistema. 1.4 Implementación del sistema Capacitación al Personal 2.1 Definición de metodologías de enseñanza para la socialización 2.2 Planificación del contenido temático de los cursos de capacitación.	Actividades <table><tr><td>Área:</td><td>Administrativa Financiera</td><td>Código:</td><td>4</td><td>Programa:</td><td>POA Y PRESUPUESTOS</td><td>Código:</td><td>4.1</td></tr><tr><td>Objetivo del Programa:</td><td colspan="7">Los(as) participantes se relacionan con la elaboración de presupuestos de la empresa, en función a su planificación y recursos factibles de ser obtenidos, de esa forma constituir un sistema de información para niveles ejecutivo, de apoyo y operativo a objeto de planificar, controlar y tomar decisiones.</td></tr><tr><td>Módulo:</td><td colspan="3">FORMULACIÓN, EJECUCIÓN Y CONTROL DEL PRESUPUESTO.</td><td>Código:</td><td>4.1.2</td><td>Requisitos:</td><td>Contar con conocimientos básicos de administración y contabilidad, o en su defecto tener experiencia de trabajo en el área financiera de una EPSA.</td></tr><tr><td>Objetivo terminal del Mod.:</td><td colspan="6">Los/as participantes Elaboran presupuestos de sus empresas, aplicando técnicas y metodologías adecuadas, considerando su realidad económica.</td><td>Tiempo total [hr:min:reloj]:</td><td>25:00</td></tr></table> <table><tr><th colspan="6">UNIDADES TEMÁTICAS</th></tr><tr><th>Cód.</th><th>Objetivo parcial</th><th>Tema y contenidos</th><th>Prerrequisitos</th><th>Bibliografía recomendada</th><th>Tiempo [hr:min:reloj]</th></tr><tr><td>4.1.2.1</td><td>Se familiarizan con los conceptos y condiciones requeridas en su clasificación de presupuestos.</td><td>Fundamentos de los presupuestos 1. Definiciones y elementos constitutivos 2. Enfoque sistémico. 3. Beneficios y debilidades de los presupuestos 4. Bases legales 5. Clasificación de presupuestos 6. Categorías Programáticas</td><td></td><td> - Rodriguez José Vicente. Elementos de Economía Presupuestaria. - Richard A. Musgrave, Peggy B. Musgrave. Hacienda Pública Teoría Aplicada. - Glenn A. Welsch, Ronald W. Hilton, Paul N. Gordon. Presupuestos, Planificación y Control de Utilidades. - Sam Goodman. Proceso de Control Directivo, Planeación, Presupuesto y Análisis de Actuación. </td><td>04:00</td></tr><tr><td>4.1.2.2</td><td>Comprenden el marco genérico referencial de los presupuestos en una EPSA del sector público</td><td>El presupuesto de las EPSAs del sector público 1. El sistema de presupuestos y sus subsistemas. 2. Particularidades de la elaboración del presupuesto. 3. Integración de los flujos de datos financieros.</td><td>UT 4.1.2.1</td><td> - Ministerio de Hacienda. Normas Básicas del Sistema de Presupuestos - Richard A. Musgrave, Peggy B. Musgrave. Hacienda Pública Teoría Aplicada. - Jorge Burbano Ruiz. Presupuesto Empresarial. Planeación y Control de Recursos. - Sam Goodman. Proceso de Control Directivo, Planeación, Presupuesto y Análisis de Actuación. </td><td>01:00</td></tr></table>	Área:	Administrativa Financiera	Código:	4	Programa:	POA Y PRESUPUESTOS	Código:	4.1	Objetivo del Programa:	Los(as) participantes se relacionan con la elaboración de presupuestos de la empresa, en función a su planificación y recursos factibles de ser obtenidos, de esa forma constituir un sistema de información para niveles ejecutivo, de apoyo y operativo a objeto de planificar, controlar y tomar decisiones.							Módulo:	FORMULACIÓN, EJECUCIÓN Y CONTROL DEL PRESUPUESTO.			Código:	4.1.2	Requisitos:	Contar con conocimientos básicos de administración y contabilidad, o en su defecto tener experiencia de trabajo en el área financiera de una EPSA.	Objetivo terminal del Mod.:	Los/as participantes Elaboran presupuestos de sus empresas, aplicando técnicas y metodologías adecuadas, considerando su realidad económica.						Tiempo total [hr:min:reloj]:	25:00	UNIDADES TEMÁTICAS						Cód.	Objetivo parcial	Tema y contenidos	Prerrequisitos	Bibliografía recomendada	Tiempo [hr:min:reloj]	4.1.2.1	Se familiarizan con los conceptos y condiciones requeridas en su clasificación de presupuestos.	Fundamentos de los presupuestos 1. Definiciones y elementos constitutivos 2. Enfoque sistémico. 3. Beneficios y debilidades de los presupuestos 4. Bases legales 5. Clasificación de presupuestos 6. Categorías Programáticas		- Rodriguez José Vicente. Elementos de Economía Presupuestaria. - Richard A. Musgrave, Peggy B. Musgrave. Hacienda Pública Teoría Aplicada. - Glenn A. Welsch, Ronald W. Hilton, Paul N. Gordon. Presupuestos, Planificación y Control de Utilidades. - Sam Goodman. Proceso de Control Directivo, Planeación, Presupuesto y Análisis de Actuación.	04:00	4.1.2.2	Comprenden el marco genérico referencial de los presupuestos en una EPSA del sector público	El presupuesto de las EPSAs del sector público 1. El sistema de presupuestos y sus subsistemas. 2. Particularidades de la elaboración del presupuesto. 3. Integración de los flujos de datos financieros.	UT 4.1.2.1	- Ministerio de Hacienda. Normas Básicas del Sistema de Presupuestos - Richard A. Musgrave, Peggy B. Musgrave. Hacienda Pública Teoría Aplicada. - Jorge Burbano Ruiz. Presupuesto Empresarial. Planeación y Control de Recursos. - Sam Goodman. Proceso de Control Directivo, Planeación, Presupuesto y Análisis de Actuación.	01:00	Documentación del análisis y diseño del sistema, utilizando las normas de modelado a los diagramas de análisis y diseño del sistema, mediante la metodología de desarrollo de sistemas RUP. Verificación de integridad referencial de la base de datos. Manual de usuario. Manual de instalación.	Disponibilidad de acceso al Sistema. Disponibilidad de comunicación y notificación con el personal involucrado (Planificador, Administración).
Área:	Administrativa Financiera	Código:	4	Programa:	POA Y PRESUPUESTOS	Código:	4.1																																																					
Objetivo del Programa:	Los(as) participantes se relacionan con la elaboración de presupuestos de la empresa, en función a su planificación y recursos factibles de ser obtenidos, de esa forma constituir un sistema de información para niveles ejecutivo, de apoyo y operativo a objeto de planificar, controlar y tomar decisiones.																																																											
Módulo:	FORMULACIÓN, EJECUCIÓN Y CONTROL DEL PRESUPUESTO.			Código:	4.1.2	Requisitos:	Contar con conocimientos básicos de administración y contabilidad, o en su defecto tener experiencia de trabajo en el área financiera de una EPSA.																																																					
Objetivo terminal del Mod.:	Los/as participantes Elaboran presupuestos de sus empresas, aplicando técnicas y metodologías adecuadas, considerando su realidad económica.						Tiempo total [hr:min:reloj]:	25:00																																																				
UNIDADES TEMÁTICAS																																																												
Cód.	Objetivo parcial	Tema y contenidos	Prerrequisitos	Bibliografía recomendada	Tiempo [hr:min:reloj]																																																							
4.1.2.1	Se familiarizan con los conceptos y condiciones requeridas en su clasificación de presupuestos.	Fundamentos de los presupuestos 1. Definiciones y elementos constitutivos 2. Enfoque sistémico. 3. Beneficios y debilidades de los presupuestos 4. Bases legales 5. Clasificación de presupuestos 6. Categorías Programáticas		- Rodriguez José Vicente. Elementos de Economía Presupuestaria. - Richard A. Musgrave, Peggy B. Musgrave. Hacienda Pública Teoría Aplicada. - Glenn A. Welsch, Ronald W. Hilton, Paul N. Gordon. Presupuestos, Planificación y Control de Utilidades. - Sam Goodman. Proceso de Control Directivo, Planeación, Presupuesto y Análisis de Actuación.	04:00																																																							
4.1.2.2	Comprenden el marco genérico referencial de los presupuestos en una EPSA del sector público	El presupuesto de las EPSAs del sector público 1. El sistema de presupuestos y sus subsistemas. 2. Particularidades de la elaboración del presupuesto. 3. Integración de los flujos de datos financieros.	UT 4.1.2.1	- Ministerio de Hacienda. Normas Básicas del Sistema de Presupuestos - Richard A. Musgrave, Peggy B. Musgrave. Hacienda Pública Teoría Aplicada. - Jorge Burbano Ruiz. Presupuesto Empresarial. Planeación y Control de Recursos. - Sam Goodman. Proceso de Control Directivo, Planeación, Presupuesto y Análisis de Actuación.	01:00																																																							

I.3. Análisis y Diseño del Sistema

I.3.1. Estándar IEEE-830

El presente documento es una Especificación de Requisitos de Software (ERS) para el Sistema de Mejoramiento para la planificación de llenado de los formularios de la elaboración de los POA's de las áreas de COSAALT. Todo su contenido ha sido elaborado en colaboración de algunos usuarios y responsables del ente Oficial que controla y Regula las actividades de la formulación, elaboración y aprobación de POA (Plan Operativo Anual).

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

También la verificación del proceso de adquisición o contratación de lo programado en el plan operativo anual. Para las asignaciones presupuestarias para la adquisición de bienes y/o servicios, contando con los recursos financieros con lo que cuenta COSAALT.

I.3.1.1. Propósito

Definir de una manera clara y precisa los requerimientos del sistema en base al proceso de POA que se realiza actualmente en COSAALT, tomando en cuenta los usuarios que forman parte de este proceso, los subprocesos, los documentos resultantes del mismo. El objetivo del Sistema es proporcionar la planificación y actividades, para facilitar la integración y el llenado de los formularios para la elaboración de los POA's en distintas áreas que presenta COSAALT.

También una realización de adquisiciones de bienes y/o servicios de las operaciones que sean indicados en el proceso presupuestario.

I.3.1.2. Ámbito del Sistema

El sistema nos permitirá proporcionar el proceso de formulación del POA, constituye su instrumento anual de planificación que permite concretizar los objetivos establecidos en el proyectos y actividades que se ejecutarán en el período de un año, por lo que tiene una relación directa con el Presupuesto, organizara las actividades y sub actividades que tienen una fecha de inicio y un tiempo de vida también cuentan con un monto económico el cual es llamado presupuesto.

Por otra parte permitirá el seguimiento y evaluación o aprobación de la formulación de los POA's de las áreas correspondientes, con este sistema será el manejo de formulación más agilizada, sistematización de los datos de cada área, proceso de retroalimentación en la formulación de POA de las diferentes áreas, etc.

Este sistema debe cumplir las distintas etapas de gestión del proceso actualmente realizado en la empresa, permitir registrar toda la información relacionada con este proceso, hacer un seguimiento de la información existente y requerida para reportarlas mediante notificaciones vía WEB, además realizar la valoración respectiva de los resultados en los tiempos previstos, mostrando resultados mediante reportes.

I.3.1.3. Definiciones, Acrónimos y Abreviaturas

FPM Formato de Planificación de Módulos

POA Programa de Operaciones Anual, coloquialmente Plan Operativo Anual

SCSA Sistema de Control y Seguimiento de Actividades.

RR.HH. recursos humanos

FPM Formato de Planificación de Módulos

Fig. Figura

Ej. Ejemplo

Bs. Boliviano(s) (moneda)

I.3.1.4. Referencias

Secretaria de planeación y finanzas. (2013) guía de programación y presupuesto 2013.

Guía para la formulación del plan operativo anual (poa).

Manual para la elaboración de los programas operativos anuales (poa).

Manual de planificación, monitoreo y evaluación.

Automatización de la valoración y control sobre la ejecución del plan operativo anual.

Manual de adquisiciones y contrataciones.

Manual de adquisiciones y contrataciones para OPP.

I.3.1.5. Visión general del documento

Este documento consta de tres secciones. Esta sección es el POA, ordena los proyectos y actividades que serán ejecutados por las diferentes áreas que tiene COSAALT., por lo que su seguimiento y control asegura una correcta utilización de los recursos disponibles. Facilita la obtención de un balance real de los efectos e impactos de los proyectos desarrollados, es decir, la medición de los alcances de la acción de COSAALT. La otra sección es el presupuesto es un instrumento de planificación económico y financiero de corto plazo, el que se expresan objetivos después del POA de cada área de COSAALT., que son traducidos en los programas de operaciones anuales, destinados a prestar bienes y servicios públicos, mediante la combinación adecuada y eficiente de los recursos.

Y la última sección es la de adquisición de bienes y servicios se orienta a obtener los bienes y servicios adecuados, de manera que satisfagan los requerimientos de calidad, cantidad y sean recibidos en el momento preciso en el lugar correcto. Tomar las decisiones en materia de compras requiere por parte del responsable, mucha habilidad y responsabilidad, debe tener en mente un conjunto de factores de calidad, cantidad, precio y oportunidad a considerar al momento de adjudicar la compra.

I.3.2. Descripción General

I.3.2.1. Perspectiva del Producto

Este sistema será realizado para el proceso de Planificación y Formulación del POA de las respectivas áreas que presenta COSAALT contando con otros sistemas para la realización del proceso mismo (sistema de contabilidad, almacenes, activos fijos, R.R.H.H., etc.).

Por tratarse de la Planificación y Formulación de un proceso vigente, se conseguirá los mismos resultados en menor tiempo y con un grado de confiabilidad, consistencia e integridad.

La confidencialidad de la información estará dada por un acceso restringido a la información por usuario, mantendrá el control sobre las posibles acciones a realizar en el sistema de acuerdo al perfil de usuario asignado.

Como valor agregado a la oportuna disponibilidad de reportes tanto de definición como de resultados se con él envió de notificaciones vía WEB para facilitar el seguimiento del proceso.

Este sistema será utilizado en toda la empresa a nivel de gerencias, departamentos y áreas departamentales por personal capacitado para esta actividad.

I.3.2.2. Funciones del Producto

El sistema eliminará el error humano en la etapa de la valoración (Formulación) del proceso, reflejando los resultados mediante notificaciones, reducirá considerablemente el tiempo empleado en la generación de formulación de los POA's y facilitará el seguimiento del proceso mediante la notificación a los responsables de los procesos, además de brindar una confiabilidad, disponibilidad y confidencialidad de la información.

Las funciones que realizará se clasifican por su orientación en el sistema.

Aplicación:

- Generar un tiempo (calendario) previsto de ejecución(es).
- Definir estructura de actividades u operaciones.
- Definir estructura de objetivos o sub objetivos.
- Asignar áreas responsables en actividades u operaciones.

Valorar las actividades u operaciones:

- Enviar notificaciones vía WEB.

Seguridades:

El sistema no controlara esta parte pero son requisitos para el sistema.

Registrar las opciones del menú.

- Crear perfiles de usuario.
- Registrar usuarios del sistema.
- Validar usuarios al ingresar al sistema.
- Registrar opciones de menú accesibles por perfil de usuario
- Registrar información de procesos y metas permitidos a nivel de usuario.
- Registrar auditoria por registro ingresado o modificado.

Reportes:

- Presentar reporte con la información base de la planificación.

- Presentar informe estructurado con la información valorada resultante.

I.3.2.3. Características de los usuarios

Para el manejo y administración del sistema se facilitará la creación de perfiles de usuario según se requiera, en base a los cuales se restringirá el acceso a las distintas opciones del menú. Inicialmente tras un análisis y por referencia de quienes se encargan del proceso de Planificación y Formulación de POA's, se han considerado 4 perfiles de usuario:

- Administrador

El administrador es el perfil inicial del sistema, este contará con la autorización sobre todas las opciones de administración del sistema, por tanto deben contar con este perfil los encargados de parametrizar el sistema, conceder autorizaciones y realizar acciones relacionadas.

- Planificador

El planificador es el perfil de usuario que realiza la definición de la información del proceso (formulación del POA).

- Evaluador

El evaluador es un tipo de perfil de usuario que tras la definición de la información base se encargará de realizar seguimiento y la valoración del proceso.

- Responsable

El responsable es el perfil que se concederá al encargado de registrar la estructura y la información que le corresponda en coordinación con el planificador además de la evolución de sus actividades u operaciones designadas.

I.3.2.4. Restricciones

El sistema debe contar con el logotipo de la empresa en una zona visible e importante, para el diseño de su interfaz de usuario se debe considerar los colores azul, amarillo y celeste (o azul) de ellos para mantener un ambiente en común con los otros sistemas.

El sistema será multiusuario, es decir permitirá el acceso de varios usuarios a la vez pero cuando se esté manipulando la misma información la primera modificación almacenada será la que permanezca en el sistema.

El sistema requiere:

Tener una estructura abierto del motor de bases de datos establecidos como estándares en la empresa:

- SQL SERVER 2008.

Para la aplicación de sistema WEB a ser ejecutado estará en un sistema operativo Windows Server 2008.

I.3.2.5. Suposiciones y Dependencias

Se requiere que la aplicación esté disponible en vía WEB ya que la plataforma utilizada en la empresa COSAALT es Windows.

I.3.3. Requisitos Específicos

I.3.3.1. Interfaces Externas

La interfaz de usuario del sistema va regida directamente con la interfaz utilizada en el SCSA (Sistema de Control y Seguimiento de Actividades), sistema que se encuentra implementado en la misma herramienta de desarrollo, por tanto se requiere las plantillas bases de dicho sistema, que contiene:

- Logo de la empresa en la ventana principal
- Nombre del sistema
- Barra de título en cada ventana
- Botones de proceso (Adicionar, Modificar, Eliminar, Cancelar, Imprimir y Enviar) en la ventana.

Barra de detalle en la parte inferior de la ventana:

- Nombre de la ventana
- Usuario conectado al sistema
- Nombre del área desde el que se conecta
- Fecha Actual

I.3.3.2. Funciones

- Determinación de metas de gestión por área funcional

Entrada: Datos generales para indicar la planificación y operaciones.

Proceso: Ingresar, modificar, eliminar, enviar e imprimir.

- Estimación de costo referencial de operaciones

Entrada: Detallar todos los requerimientos para las operaciones para ser ejecutado

Proceso: Ingresar, modificar, eliminar cancelar y aceptar.

- Determinación de operaciones

Entrada: Vigencia del plan operativo y el tiempo de periodo de valoración.

Procesos: Ingresar, modificar, eliminar cancelar, aceptar, enviar e imprimir.

- Administrar la estructura de proceso y subprocesos

Entrada: Datos del proceso.

Proceso: Ingresar, modificar, eliminar cancelar y aceptar.

- Presentar reportes con la información base del proceso de planificación

Entrada: especificar el plan estratégico-operativo.

Proceso: Recopilar la información solicitada y estructurar el informe.

I.3.3.3. Requisitos de Rendimiento

El sistema sustituirá el actual proceso de control y valoración sobre la ejecución del plan operativo de COSAALT, que se lo realiza de forma manual.

Al sistema se lo configurará en más de 30 estaciones de trabajo, de las cuales se estima que el 50% de ellas puede tratarse de usuarios simultáneos.

Se estima que mensualmente se almacenará 200 registros aproximadamente.

El ingreso de información al sistema será en un 80% de forma secuencial, para cumplir con cada una de las fases establecidas en este proceso, por tanto el 20% restante no tendrá mayor importancia el orden de ingreso de los datos.

I.3.3.4. Restricciones de Diseño

En el diseño de la base de datos y la definición de los campos se deberá seguir los estándares establecidos en la empresa para los sistemas que utilizan como motor de base de datos SQL SERVER 2008 y para la aplicación estará en UBUNTU SERVER.

I.3.3.5 Atributos del Sistema

- Las opciones de menú disponibles se registrarán, para que puedan ser accedidas de acuerdo a los permisos otorgados según el perfil asignado al usuario (Opciones de Menú por Perfil de Usuario).

- El acceso a la información del sistema se facilitará en base a los permisos otorgados de la información de procesos y metas para cada usuario (Acceso a información por usuario)
- Para iniciar a interactuar con el software se debe ingresar el nombre de usuario y clave correcta, para que el sistema cargue las políticas de seguridad definidas (Acceso de acuerdo a las opciones del menú por perfil y por usuario).

Presentar mensajes de error explicativos para que los usuarios puedan basarse en ellos para continuar con su interacción con el software.

I.3.3.6 Mecanismos de seguridad

- Contiene información del software desarrollado, configuraciones, perfiles de usuario, restricciones, etc.
- Las opciones de menú disponibles se registrarán, para que puedan ser accedidas de acuerdo a los permisos otorgados según el perfil asignado al usuario.
- El acceso a la información del sistema se facilitará en base a los permisos otorgados de la información de procesos y metas para cada usuario.
- La clave del usuario se debe almacenar de una forma encriptado para guardar su confidencialidad, para evitar el plagio y por ende la mala utilización de las mismas.
- Para iniciar a interactuar con el software se debe ingresar el nombre de usuario y clave correcta, para que el sistema cargue las políticas de seguridad definidas.

Presentar mensajes de error explicativos para que los usuarios puedan basarse en ellos para continuar con su interacción con el software.

Capítulo III: Conclusiones y Recomendaciones

III.1. Conclusiones

Culminado el presente proyecto, se detallan las siguientes conclusiones:

- Se logró cumplir el propósito del proyecto, a través de su indicador reducido el número de errores en un 70% en los procesos de Formulacion, Evaluacion y Ejecucion.
- El sistema fué desarrollado por el Frame Spring Source, el que garantizó la integridad y consistencia de datos, utilizando así el Modelo Vista Controlador (MVC), lo que permitió agilizar la conclusión del sistema.
- Se aplicó la metodología RUP para el desarrollo de software, lo que implicó apoyarse también en la metodología de prototipos, para que la etapa de desarrollo pueda abarcar con los requerimientos que no se pudieron identificar en la etapa de análisis.
- Todas las herramientas de software descritas en el documento han sido de gran utilidad habiendo facilitado el desarrollo del sistema en sus diferentes etapas.
- Por último, se puede afirmar que la aplicación de las Tecnologías de Información y Comunicación (TIC) han sido definitivamente útiles y determinantes en elMejoramiento para la planificación de la elaboración de los Plan Operativo Anual (POA) de la Cooperativa de servicios de agua potable y alcantarillado de Tarija (COSAALT-LTDA).
- Se realizó la capacitación la que asegura que tanto el administrador del sistema, como los funcionarios de la empresa COSAALT, operarán correctamente el sistema, reduciendo los errores.

III.2 Recomendaciones

- Se aplicó la herramienta Spring Source la que permitió optimizar código fuente, evitar mapeos e instancias en clases externas, facilitó la configuración con el servidor Apache Tomcat, así también su estructura de Modelo Vista Controlador

proporcionó una visualización más descriptiva para el entorno de la programación, por estos motivos se recomienda aplicar Spring Source.

- Para el desarrollo de un sistema, se recomienda estudiar las ventajas de la metodología RUP, como de otras para así elegir de acuerdo a los alcances de lo propuesto la aplicación de la metodología correcta.
- Para la implementación y explotación del proyecto se recomienda realizar periódicamente copias de respaldo de información (backups), por el motivo de evitar pérdidas de información.
- Se recomienda ampliar el sistema desarrollado para contemplar nuevos módulos que abarquen marcaciones por medios biométricos, control de descuentos adicionales, considerando que el diseño es flexible y sujeto de escalamiento. Por último se recomienda realizar la capacitación a cada nuevo funcionario para que este pueda utilizar el sistema sin mayores complicaciones y de manera correcta.

CAPITULO II: COMPONENTES

II.1. Componente 1: Sistema web

II.1.1. Plan Operativo Anual

II.1.1.1. Introducción

La Cooperativa de Servicios de Agua Potable y Alcantarillado de Tarija se encuentra en la necesidad de automatizar el proceso de planeación estratégica-operativa empresarial que realizan los departamentos de Planificación

II.1.1.2. Situación Actual

El proceso de planeación hoy en día se realiza de forma manual, es decir se inicia con la planificación estratégica propiamente dicha que consiste en utilizar hojas de cálculo (Microsoft Excel) de acuerdo a la filosofía empresarial, misión, visión, objetivos estratégicos. Estos resultados se van a registrar en un documento oficial el cual servirá como base para plantear la planificación operativa anual de la empresa.

Para la elaboración de la planificación operativa se requiere de que cada área, departamento, unidad y gerencia establezca según sus actividades y responsabilidades, el conjunto de metas que se van a cumplir en el tiempo máximo de un año, las cuales deben ser realistas, estimulantes, coherentes, prioritarias, medibles y cuantitativas, por tanto deben ser representadas con un indicador operativo. Una vez definidas las metas a cumplir se envía un reporte al departamento coordinador de esta actividad, en donde recopilan información, revisan y aprueban la formulación de las metas y elaboran el Plan Operativo Anual.

Cada área, departamento, unidad y gerencia de la empresa debe emitir trimestralmente los resultados de gestión obtenidos en las metas planteadas. Estos informes deben ser entregados oportunamente al responsable de Control de Gestión del Plan Operativo, quien revisa la consistencia de la información, aprueba los datos, los procesa de acuerdo a distintos criterios para aplicar métodos de cálculo apropiados y elabora el informe trimestral consolidado y acumulado de la Valoración del Plan Operativo.

Este control de gestión o monitoreo tiene por objeto presentar informes trimestrales del grado de cumplimiento de las operaciones, para que en base a

la revisión de estos informes en juntas de directiva y reuniones, se puedan formular y aplicar a tiempo procedimientos para el cumplimiento de los objetivos estratégicos y a su vez mantener a la organización en un proceso de mejora continua.

II.1.1.3. Análisis y diseño del componente

II.1.1.3.1 Lenguaje Unificado de Modelado (UML)

El UML es una notación que brinda elementos gráficos para realizar los diagramas, representar partes de un modelo, un modelo propiamente dicho, subsistemas, procesos de negocio, funciones de sistema, esquemas de bases de datos y componentes de software reusables, entre otros.

II.1.1.3.2. Diagrama de casos de uso

II.1.1.3.2.1. Diagrama de casos de uso General

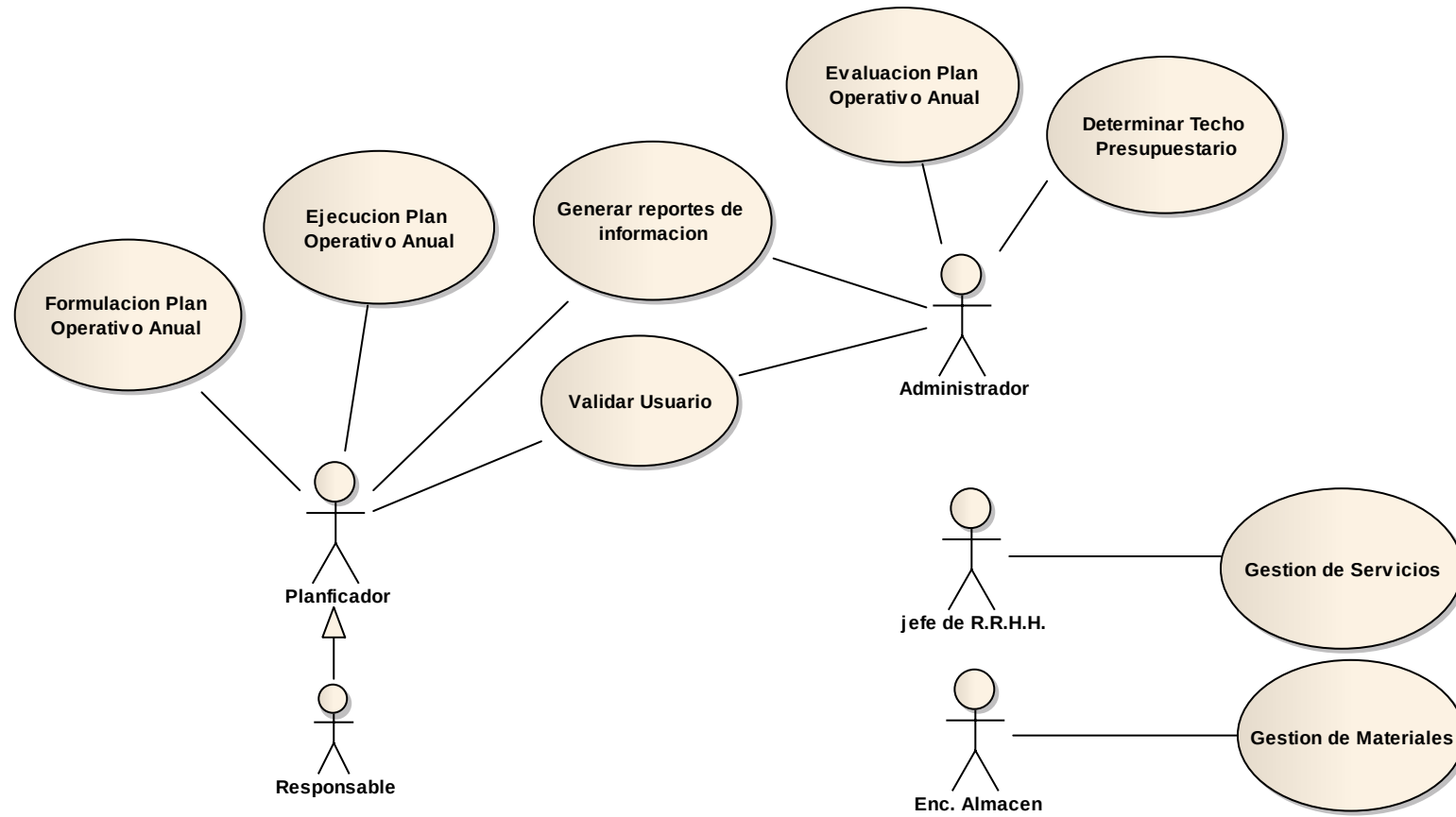


Figura 1 Diagrama de casos de uso General

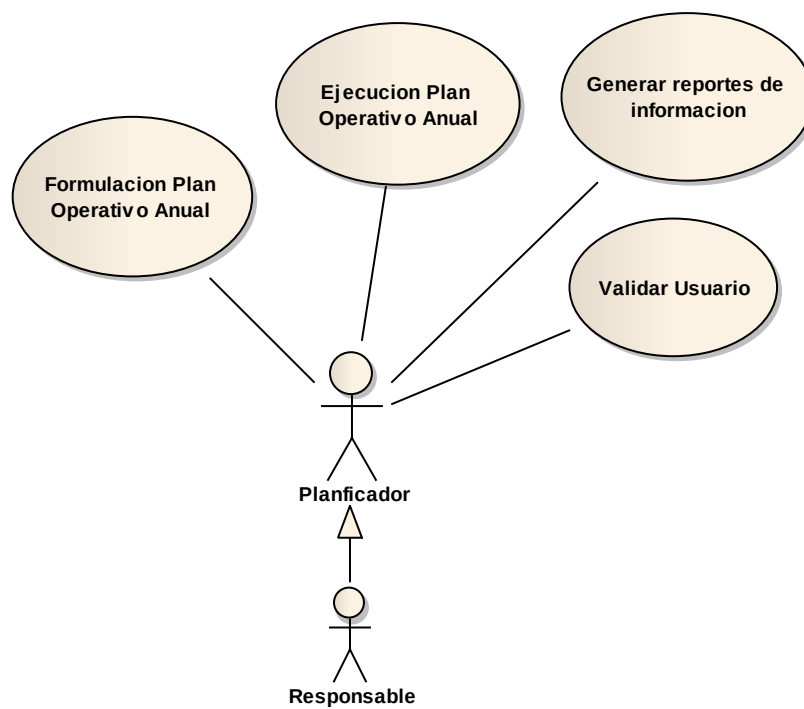


Figura 2 Diagrama de casos de uso Planificador

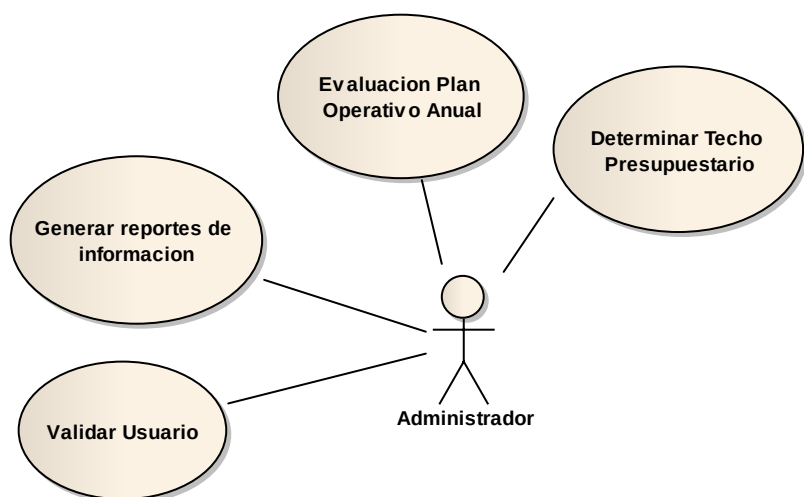


Figura 3 Diagrama de casos de uso Administrador

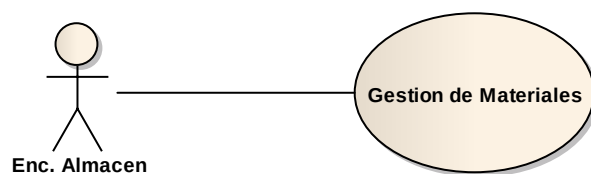


Figura 4 Diagrama de casos de uso Enc Almacen

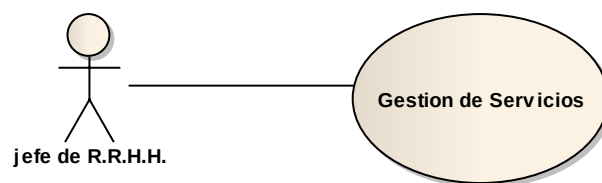


Figura 5 Diagrama de caso de usos Jefe de R.R.H.H.

II.1.1.3.2.2. Diagrama de casos de uso Específico

II.1.1.3.2.2.1. Diagrama de casos de uso Ingresar al Sistema

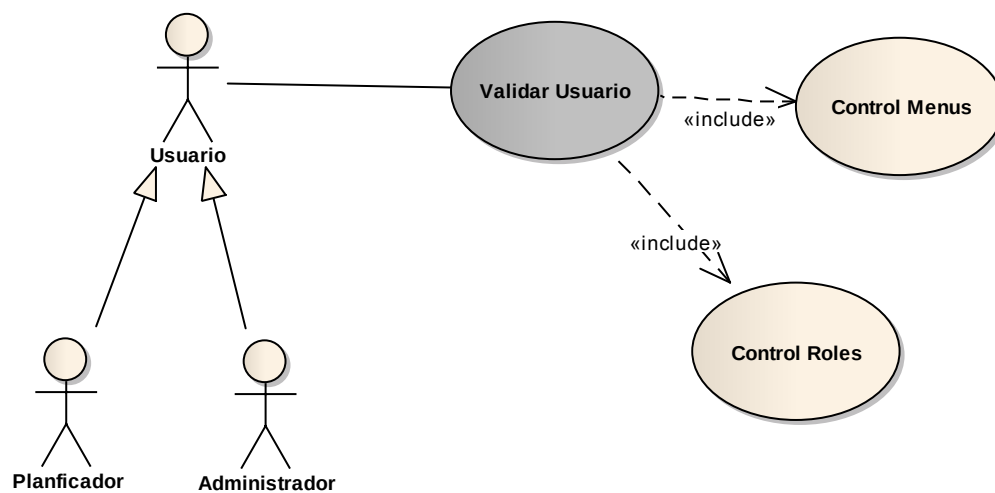


Figura 6 Diagrama de casos de uso Validar Usuario

II.1.1.3.2.2.2. Diagrama de casos de uso Formulación del Plan Operativo Anual (POA)

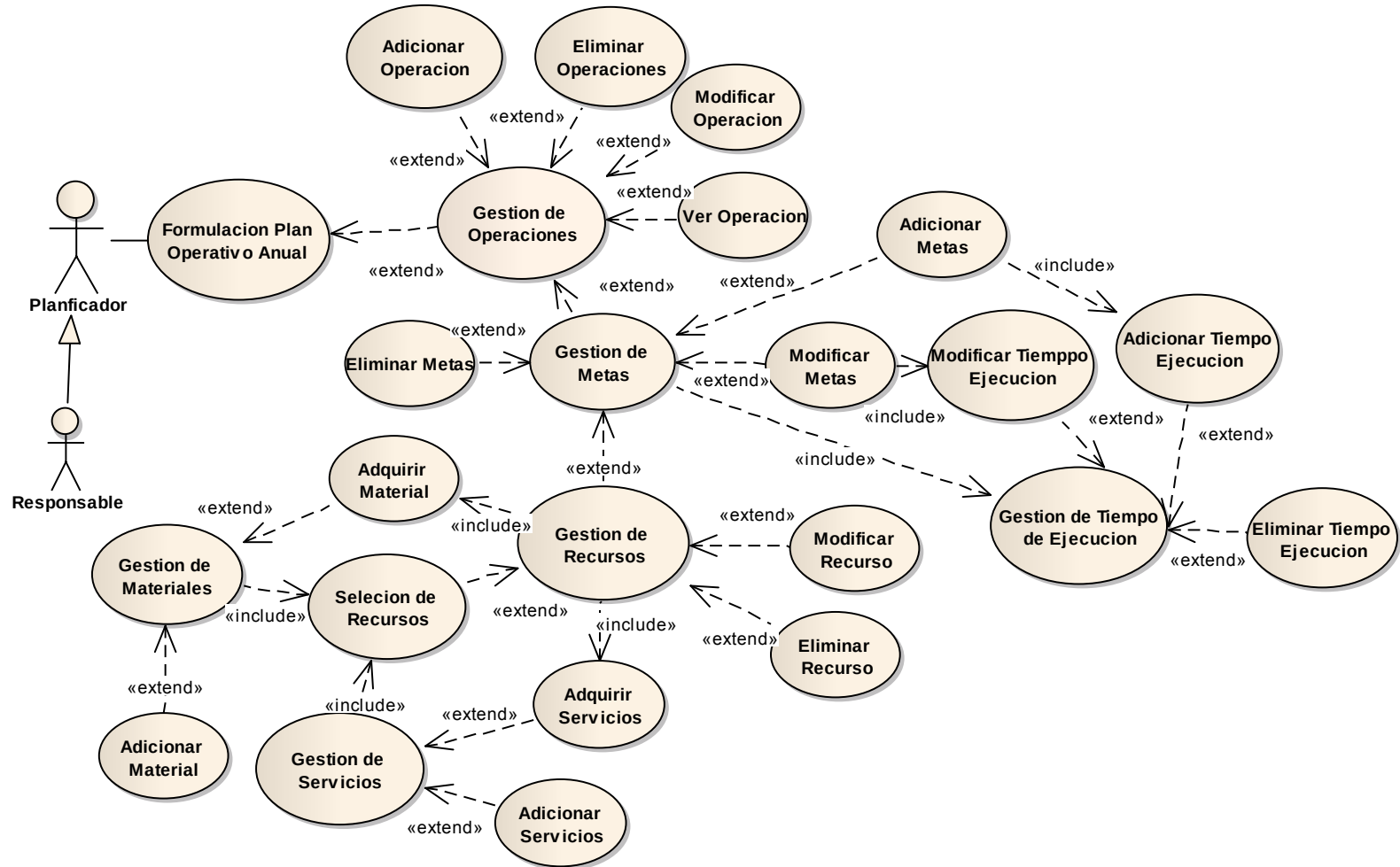


Figura 7 Diagrama de caso de usos Formulación POA

II.1.1.3.2.2.3. Diagrama de casos de uso Evaluación del Plan Operativo Anual (POA)

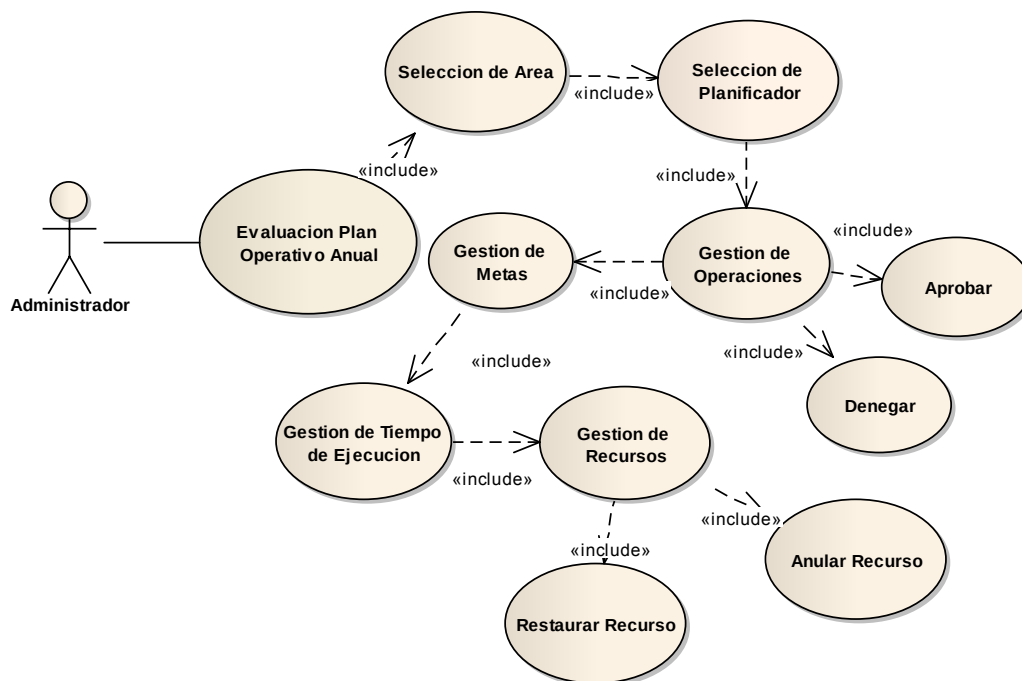


Figura 8 Diagrama de caso de usos Evaluación POA

II.1.1.3.2.2.4. Diagrama de casos de uso Ejecución del Plan Operativo Anual

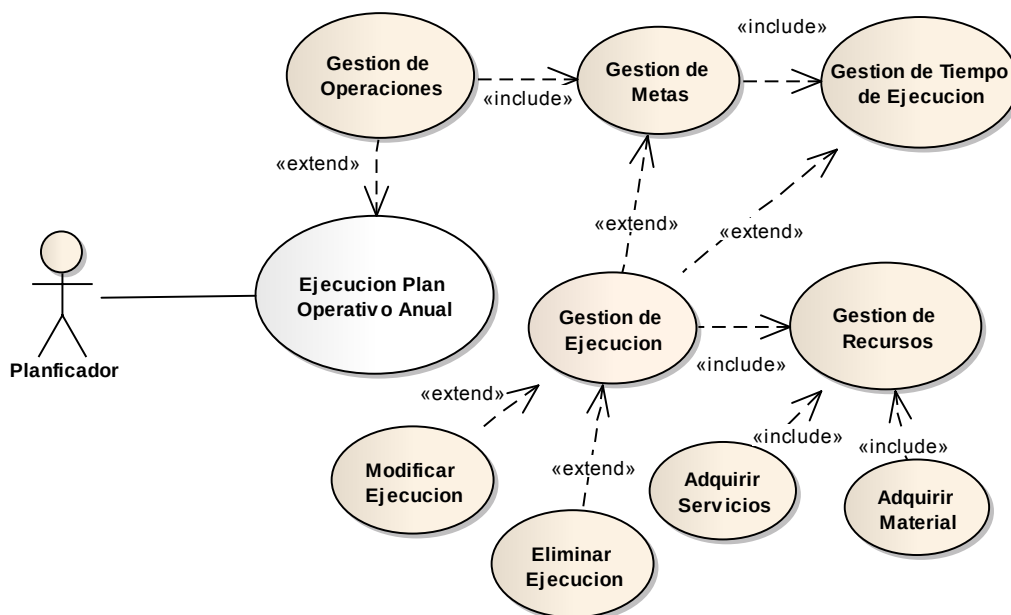


Figura 9 Diagrama de caso de usos Ejecución POA

II.1.1.3.2.2.5. Diagrama de casos de uso Determinación de Techo Presupuestario

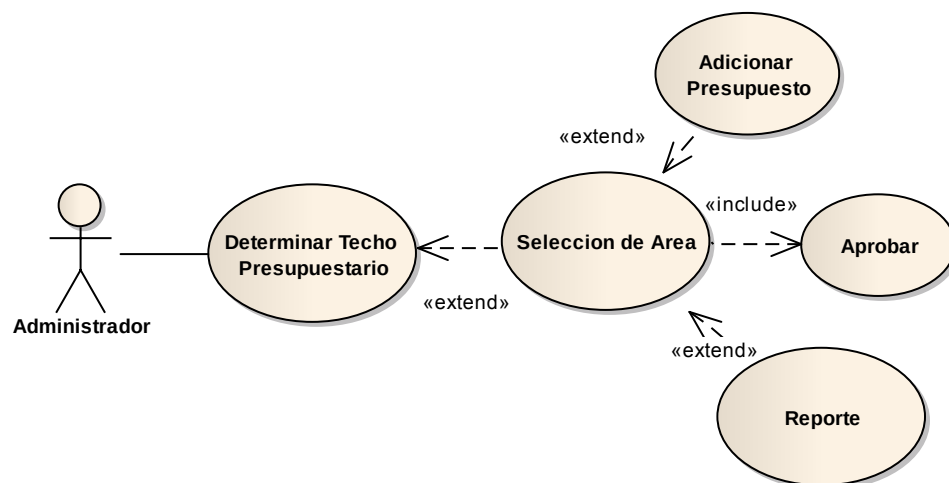


Figura 10 Diagrama de casos de uso Determinación de Techo Presupuestario

II.1.1.3.2.2.6. Diagrama de casos Gestión de Materiales

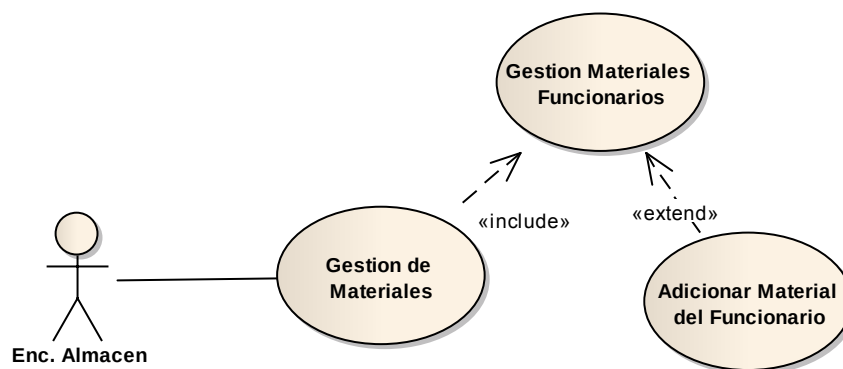


Figura 11 Diagrama de casos Gestión de Materiales

II.1.1.3.2.2.7. Diagrama de casos de uso Gestión de Servicios

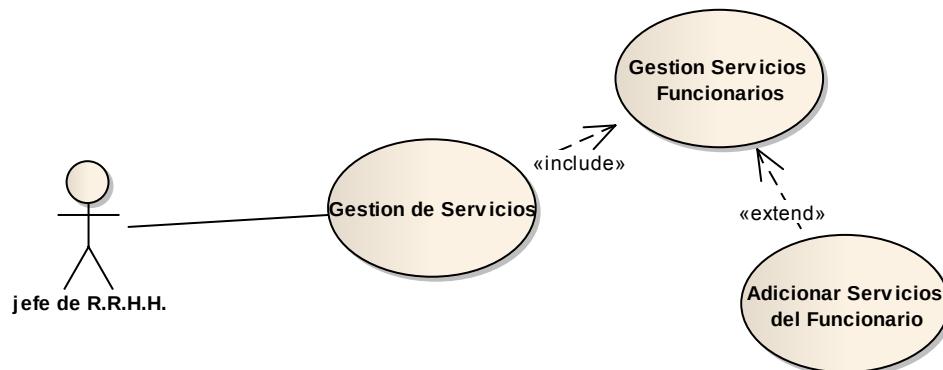


Figura 12 Diagrama de casos de uso Gestión de Servicios

II.1.1.3.2.8. Diagrama de casos de uso Generar Reportes

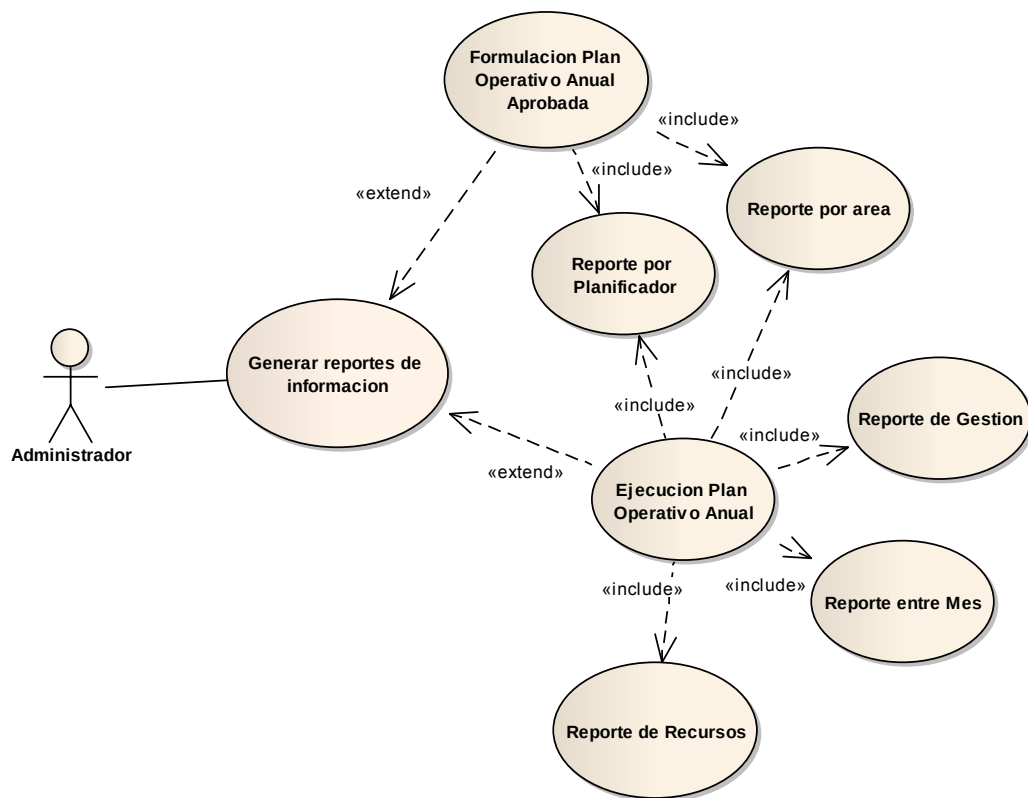


Figura 13 Diagrama de casos de uso Generar Reportes

II.1.1.3.3. Especificación de los casos de uso

II.1.1.3.3.1. Especificación de casos de uso Ingresar al Sistema

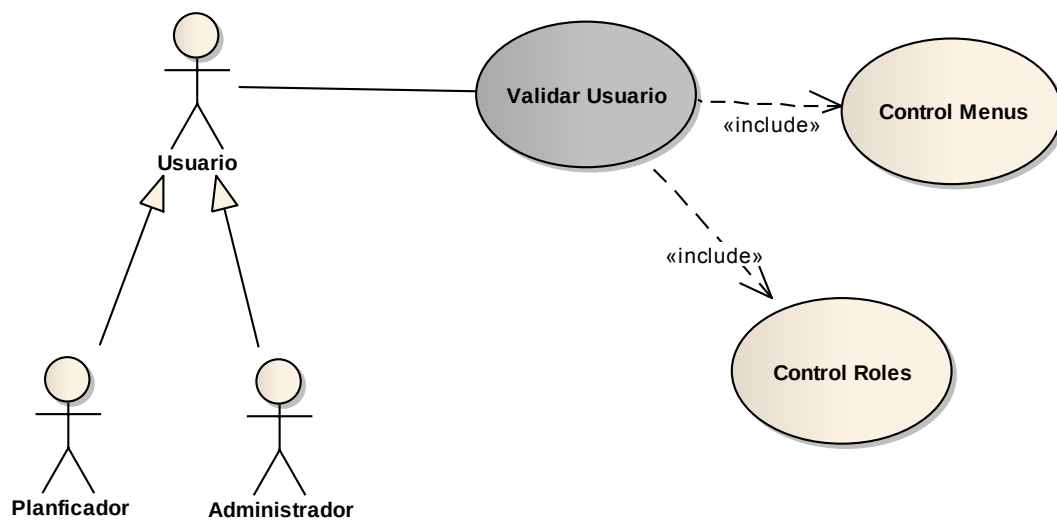


Figura 14 Especificación de casos de uso Ingresar al Sistema

Caso de uso:	Ingresar al sistema.
Descripción: Permite ingresar al sistema, este caso tiene como función controlar el acceso y al mismo tiempo recuperar los permisos correspondientes al momento que el usuario introduzca su Usuario y Clave en el sistema.	
Actores: Administrador y Planificador.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El actor introduce datos de Usuario y Clave en el sistema. 2. El actor presiona botón Ingresar. 3. El sistema valida sus datos. 4. El actor espera su validación. 5. Si los datos son correctos muestra la Pantalla Principal del Sistema "SIAF".	Flujo Alternativo: 1. Si el actor no llena el formulario de acceso, se muestra un mensaje donde se informa que los campos son requeridos. 2. Si el usuario no existe se muestra mensaje de error y vuelve a mostrar la pantalla principal.
Pos condiciones: Ninguno	

Cuadro 1 Especificación de casos de uso Ingresar al Sistema

II.1.1.3.3.2. Especificación de casos de uso del Planificador

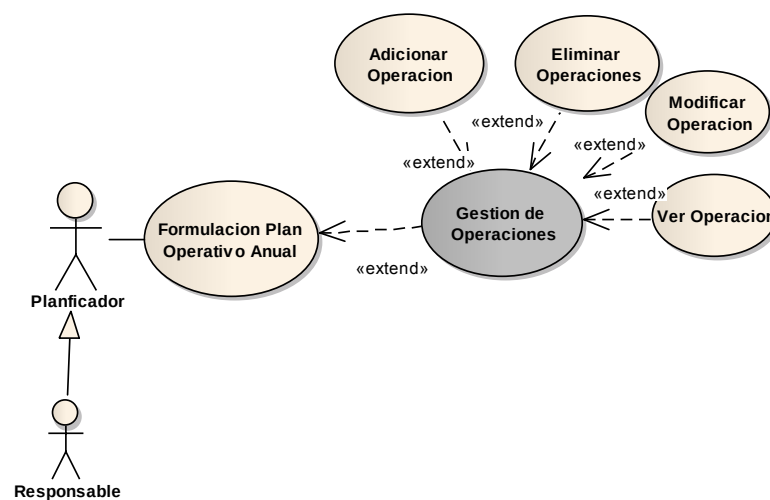
II.1.1.3.3.2.1. Especificación de casos de uso Formulación Plan Operativo Anual
Gestión de Operaciones

Figura 15 Especificación de casos de uso Formulación Plan Operativo Anual – Gestión Objetivos

Caso de uso:	Formulación Plan Operativo Anual - Gestión de Operaciones
Descripción:	Permite Listar, Adicionar, Modificar, Eliminar y Ver las Operaciones.
Actores:	Planificador.
Precondiciones:	Actor debe haber iniciado sesión.
Flujo Normal: 1. El sistema muestra Listado de las operaciones en el sistema. 2. El actor realiza la Búsqueda de las operaciones en el Listado. 3. El actor realiza Activo/Inactivo de las operaciones. 4. El sistema muestra las operaciones que cumple(n) con los Criterios de Búsqueda, Activo e Inactivo.	Flujo Alternativo:
Pos condiciones:	Ninguno

Cuadro 2 Especificación de casos de uso Formulación Plan Operativo Anual

II.1.1.3.3.2.2. Especificación de casos de uso Agregar Operación

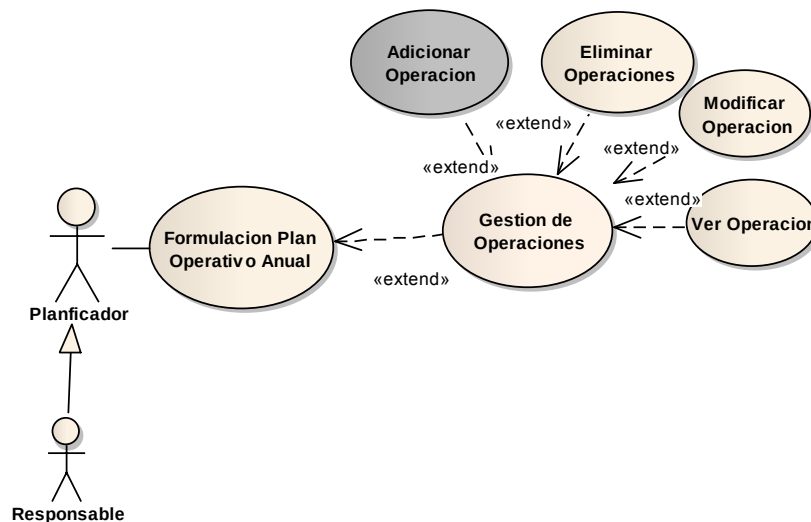


Figura 16 Especificación de casos de uso Agregar Operación

Caso de uso:	Adicionar Operación
Descripción: Permite registrar nueva Operación en el sistema.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y presionar Adicionar.	
Flujo Normal: 1. El actor presiona botón Adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos de Operación en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra la Pantalla de Gestión de Operaciones.	Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la Pantalla de Gestión de Operaciones. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: ninguno.	

Cuadro 3 Especificación de casos de uso Agregar Operación

II.1.1.3.3.2.3. Especificación de casos de uso Eliminar Operación

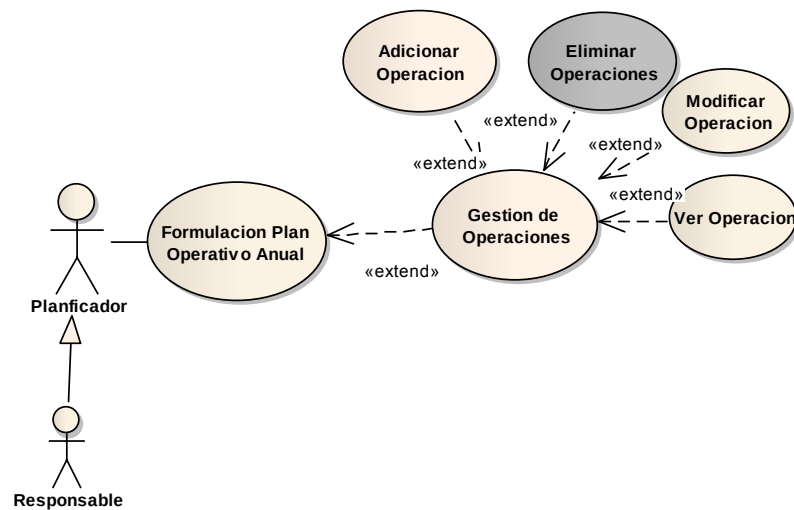


Figura 17 Especificación de casos de uso Eliminar Operación

Caso de uso:	Eliminar Operación
Descripción: Permite eliminar el registro de una operación en el sistema.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y seleccionar una operación para realizar dado de inactivo.	
Flujo Normal: 1. El actor presiona Eliminar. 2. El sistema le muestra un mensaje donde avisa si desea eliminar. 3. El actor selecciona aceptar. 4. Se valida datos para ser dado de inactivo la operación. 5. El sistema mostrar un mensaje donde avisara que fue dado de baja la operación. 6. El sistema muestra la Pantalla de Gestión de Operaciones.	Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la Pantalla de Gestión de Operaciones.
Pos condiciones: ninguno.	

Cuadro 4 Especificación de casos de uso Eliminar Operación

II.1.1.3.3.2.4. Especificación de casos de uso Modificar Operación

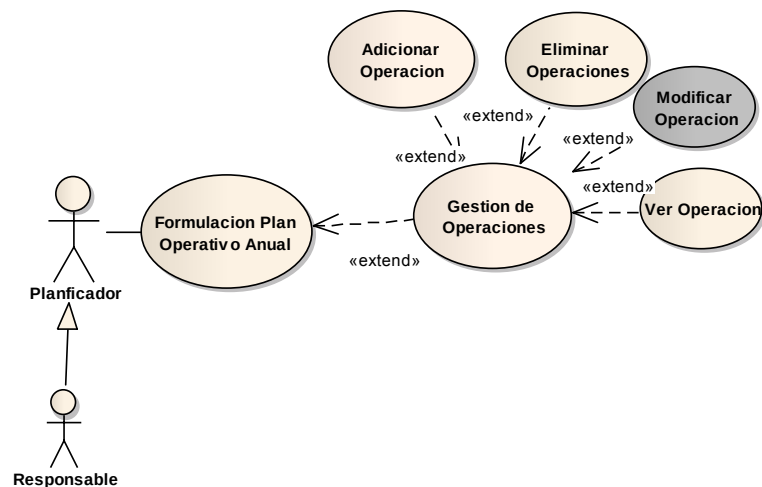


Figura 18 Especificación de casos de uso Modificar Operación

Caso de uso:	modificar Operación
Descripción: Permite modificar una Operación registrado en el sistema.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y presionar Modificar.	
Flujo Normal: 1. El actor presiona Modificar. 2. El sistema le muestra formulario con los datos de operación elegida. 3. El actor introduce datos de Operación en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra la Pantalla de Gestión de Operaciones.	Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la Pantalla de Gestión de Operaciones. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: ninguno.	

Cuadro 5Especificación de casos de uso Modificar Operación

II.1.1.3.3.2.5. Especificación de casos de uso Ver Operación

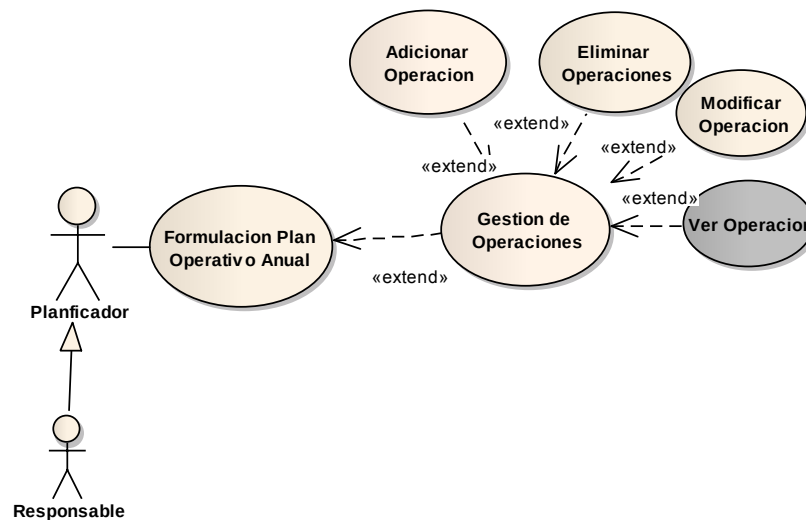


Figura 19Especificación de casos de uso Ver Operación

Caso de uso:	Ver Operación
Descripción: Permite Ver datos de una Operación.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y haber sido seleccionado del Listado Gestión Operación.	
Flujo Normal: 1. El actor presiona Ver. 2. El sistema le muestra en PDF en sistema. 3. El actor puede realizar a imprimir la operación.	Flujo Alternativo:
Pos condiciones: ninguno	

Cuadro 6Especificación de casos de uso Ver Operación

II.1.1.3.3.2.6. Especificación de casos de uso Modificar Tiempo de Ejecución

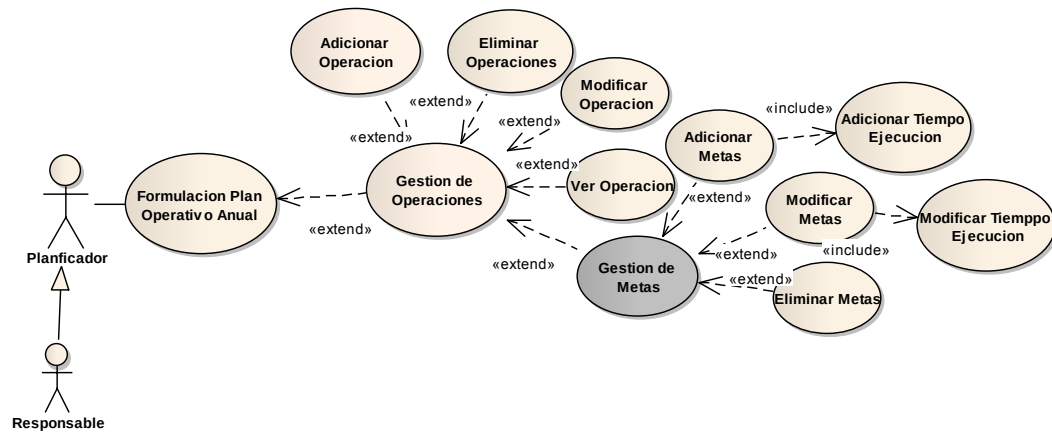


Figura 20 Especificación de casos de uso Modificar Tiempo de Ejecución

Caso de uso:	Gestión de Metas.
Descripción:	Permite Listar, Adicionar, Modificar y Eliminar Metas.
Actores:	Planificador.
Precondiciones:	Actor debe haber iniciado sesión e ingresado a Gestión de operaciones.
Flujo Normal: 1. El sistema muestra Listado de las metas en el sistema. 2. El actor realiza la Búsqueda de las metas en el Listado. 3. El actor realiza Activo/Inactivo de las metas. 4. El sistema muestra las metas que cumple(n) con los Criterios de Búsqueda, Activo e Inactivo.	Flujo Alternativo:
Pos condiciones:	ninguno

Cuadro 7 Especificación de casos de uso Modificar Tiempo de Ejecución

II.1.1.3.3.2.7. Especificación de casos de uso Adicionar Meta y Tiempo de Ejecución

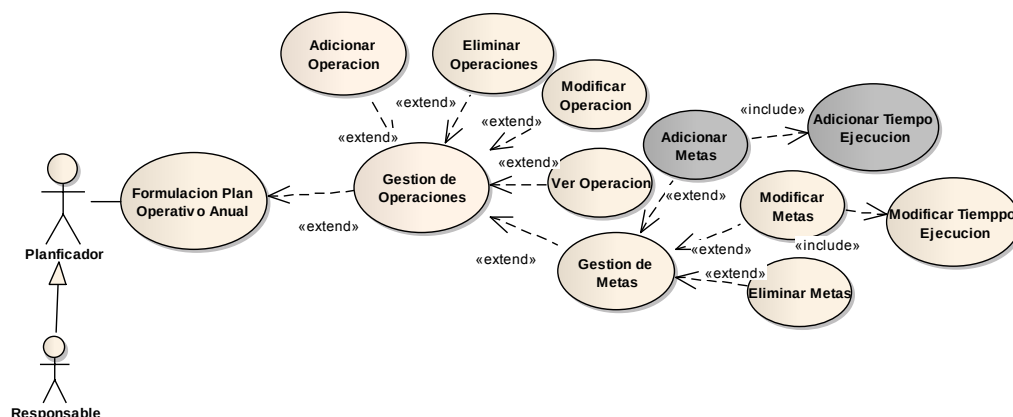


Figura 21 Especificación de casos de uso Adicionar Meta y Tiempo de Ejecución

Caso de uso:	AdicionarMetas
Descripción: Permite registrar nueva Meta en el sistema.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y presionar Adicionar.	
Flujo Normal: 1. El actor presiona Adicionar. 2. El sistema le muestra formulario. 3. El actor introduce datos de Operación en el formulario. 4. Selecciona botón Guardar. 5. Se mostrar otro formulario (Adicionar Tiempo de Ejecución), 6. El actor selecciona los meses a ejecutarse la meta en el formulario. 7. Selecciona botón Guardar. 8. Se valida datos del formulario que son obligatorios estén introducidos. 9. El sistema guarda datos de los formularios y muestra la Pantalla de Gestión de Metas.	Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la Pantalla de Gestión de Metas. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: ninguno.	

Cuadro 8 Especificación de casos de uso Adicionar Meta y Tiempo de Ejecución

II.1.1.3.3.2.9. Especificación de casos de uso Modificar Meta – Tiempo de Ejecución

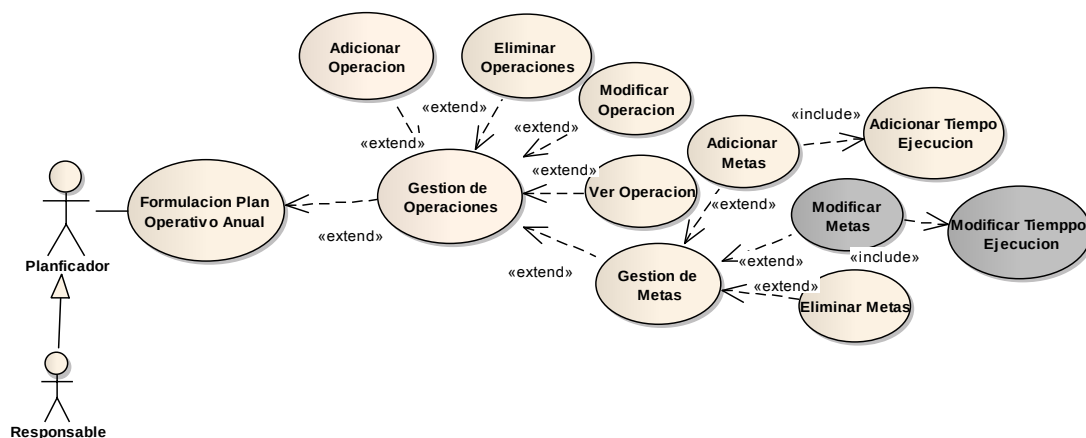


Figura 23 Especificación de casos de uso Modificar Meta – Tiempo de Ejecución

Caso de uso:	modificarMeta
Descripción: Permite modificar una Meta registrado en el sistema.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y presionar Modificar.	
Flujo Normal: 1. El actor presiona Modificar. 2. El sistema le muestra formulario con datos a modificar. 3. El actor introduce datos de Operación en el formulario. 4. Selecciona botón Guardar. 5. Se mostrar otro formulario (Modificar Tiempo de Ejecución), 6. El actor selecciona los meses a ejecutarse la meta en el formulario. 7. Se valida datos del formulario que son obligatorios estén introducidos. 8. El sistema guarda datos de los formularios y muestra la Pantalla de Gestión de Metas.	Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la Pantalla de Gestión de Metas. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: ninguno.	

Cuadro 10 Especificación de casos de uso Modificar Meta – Tiempo de Ejecución

Pos condiciones: Ninguno

Cuadro 11 Especificación de casos de uso Gestión de Recursos

II.1.1.3.3.2.11. Especificación de casos de uso Adicionar Recursos – Selección de Recursos

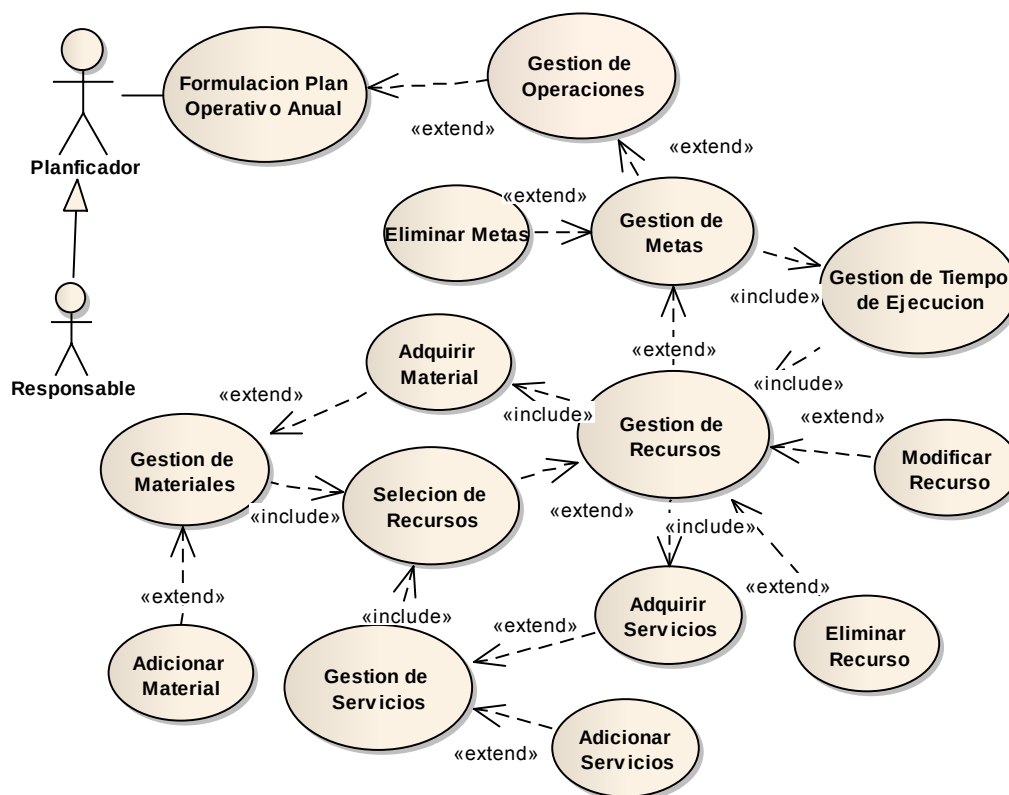


Figura 25 Especificación de casos de uso Adicionar Recursos – Selección de Recursos

Caso de uso:	Adicionar Recursos
Descripción: Permite registrar un nuevo recurso en el sistema.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y seleccionar una meta o tiempo de ejecución.	
Flujo Normal: 1. El actor presiona Adicionar. 2. El sistema muestra una pantalla para seleccionar un recurso para ser	Flujo Alternativo: 1. Por cancelar el sistema no realizará ninguna acción. 2. Por Atrás volverá a la pantalla

adicionado. 3. Se mostrar una Gestión de Material o Servicio y seleccionara un recurso. 4. El actor introduce datos de Recursos en el formulario. 5. Selecciona botón Guardar. 6. Se valida datos del formulario que son obligatorios estén introducidos. 7. El sistema guarda datos delos formularios y muestra la Pantalla de Gestión de Metas.	Gestión de Recursos
Pos condiciones: ninguno.	

Cuadro 12Especificación de casos de uso Adicionar Recursos – Selección de Recursos

II.1.1.3.3.2.12. Especificación de casos de uso Adicionar Material o Servicio

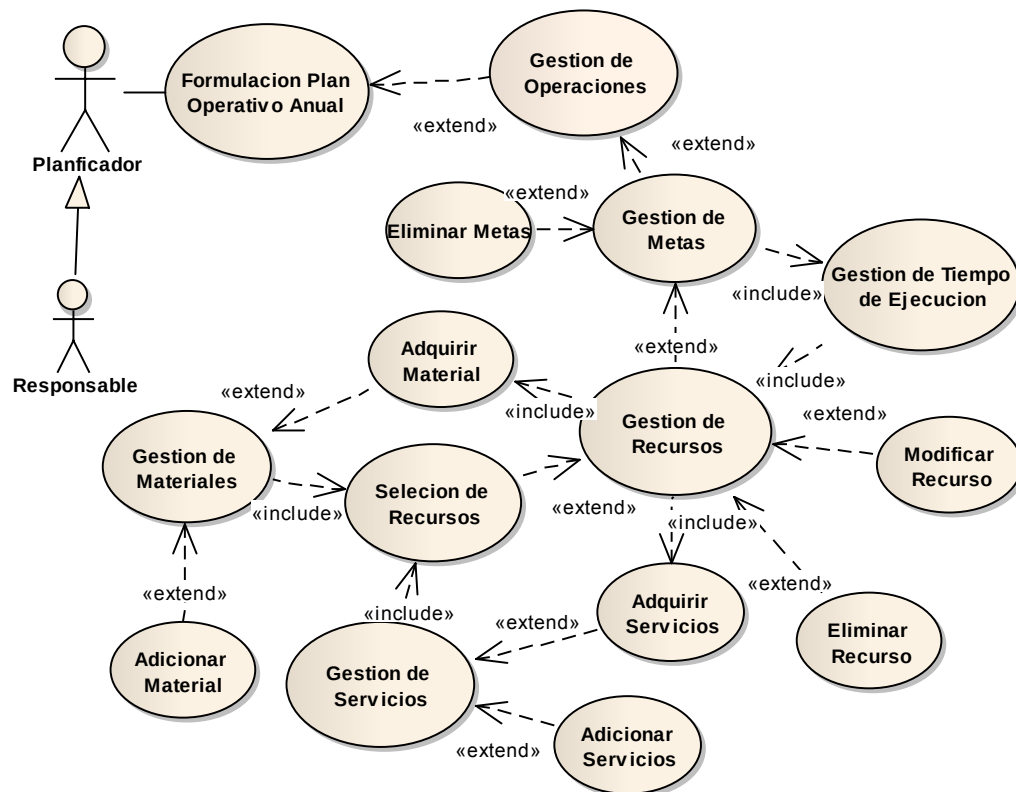


Figura 26Especificación de casos de uso Adicionar Material o Servicio

Caso de uso:	Adicionar Material o Servicio
Descripción: Permite registrar un nuevo material o servicio para después el Enc. Almacenes valide la adición en el sistema.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y seleccionar un recurso.	
Flujo Normal: 1. El actor presiona Adicionar. 2. Se mostrar una formulario 3. El actor introduce datos de Recursos material o servicio en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra la Pantalla de Gestión de Material o Servicio	Flujo Alternativo: 1. Por cancelar el sistema no realizará ninguna acción. 2. Por Atrás volverá a la pantalla Gestión de Recursos 3. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: ninguno.	

Cuadro 13Especificación de casos de uso Adicionar Material o Servicio

II.1.1.3.3.2.13. Especificación de casos de usoGestión de Ejecución

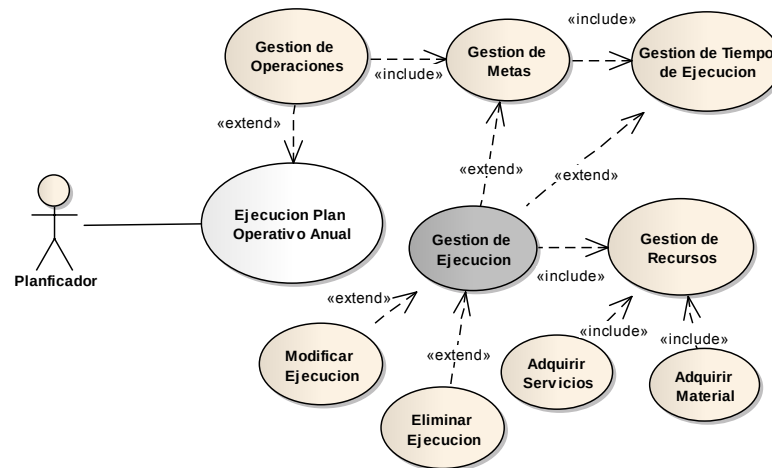


Figura 27Especificación de casos de uso Gestión de Ejecución

Caso de uso:	Gestión de Ejecución
Descripción: Permite Listar, Adicionar, Modificar, Eliminar y generar Reportes de los Recursos.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión y detalles (Recursos).	
Flujo Normal: 1. El sistema muestra Listado de los Recurso Ejecutado de una Meta en el sistema. 2. El actor realiza la Búsqueda en el Listado. 3. El sistema muestra las operaciones que cumple(n) con los Criterios de Búsqueda.	Flujo Alternativo: 1. Por Atrás volverá a la pantalla anterior de la Gestión.
Pos condiciones: Ninguno	

Cuadro 14Especificación de casos de uso Gestión de Ejecución

II.1.1.3.3.2.14. Especificación de casos de uso Adquirir Recurso

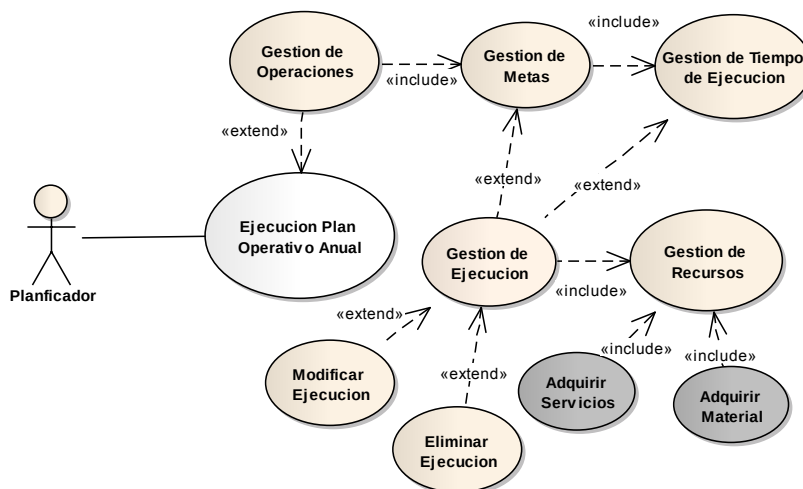


Figura 28Especificación de casos de uso Adquirir Recurso

Caso de uso:	Adquirir Recurso
Descripción: Permite Adquirir un Recurso en el sistema.	
Actores: Planificador.	

Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: 1. El actor presiona Recurso. 2. Se mostrar un listado de los recursos de una meta. 3. Al presionar Adquirir se mostrara un formulario. 4. El actor introduce datos de ejecución en el formulario. 5. Presiona guardar. 6. Se valida datos del formulario que son obligatorios estén introducidos. 7. El sistema muestra en la pantalla de gestión de Ejecución.	Flujo Alternativo: 1. Presionar Atrás se volverá a la pantalla de Gestión de Ejecución. 2. Selecciona botón Cancelar, se regresa a la pantalla de gestión de recurso. 3. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 4 para que el actor pueda ingresar la información que falta.
Pos condiciones: ninguno	

Cuadro 15 Especificación de casos de uso Adquirir Recurso

II.1.1.3.3.2.15. Especificación de casos de uso Modificar Ejecución

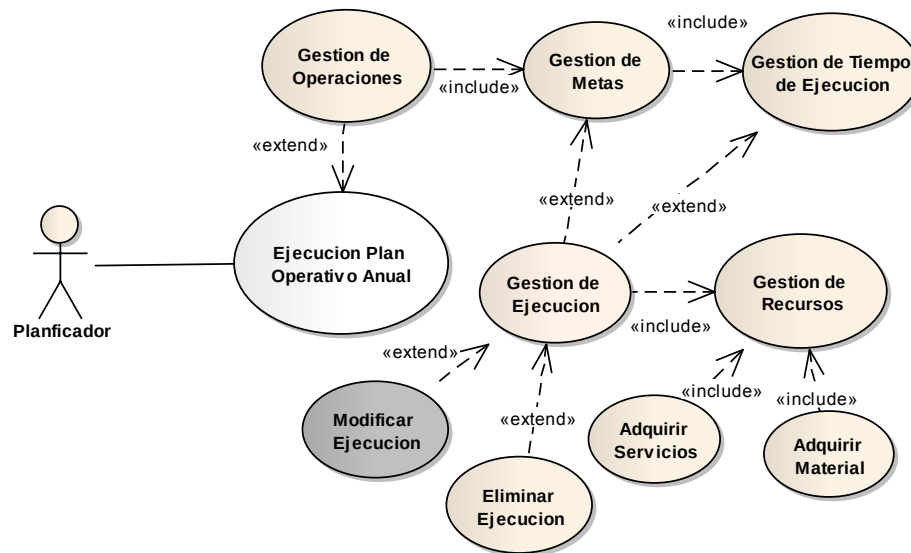


Figura 29 Especificación de casos de uso Modificar Ejecución

Caso de uso:	Modificar Ejecución
Descripción: Permite seleccionar un Recurso para ser modificado en el sistema.	
Actores: Planificador.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: 1. El actor presiona Modificar. 2. Se mostrar un formulario 3. El actor introduce datos en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra la Pantalla de Gestión de Ejecución.	
Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la Pantalla de Gestión de Ejecución.	
Pos condiciones: Agregar Tiempo de Ejecución.	

Cuadro 16 Especificación de casos de uso Modificar Ejecución

II.1.1.3.3.2.16. Especificación de casos de uso Eliminar Ejecución

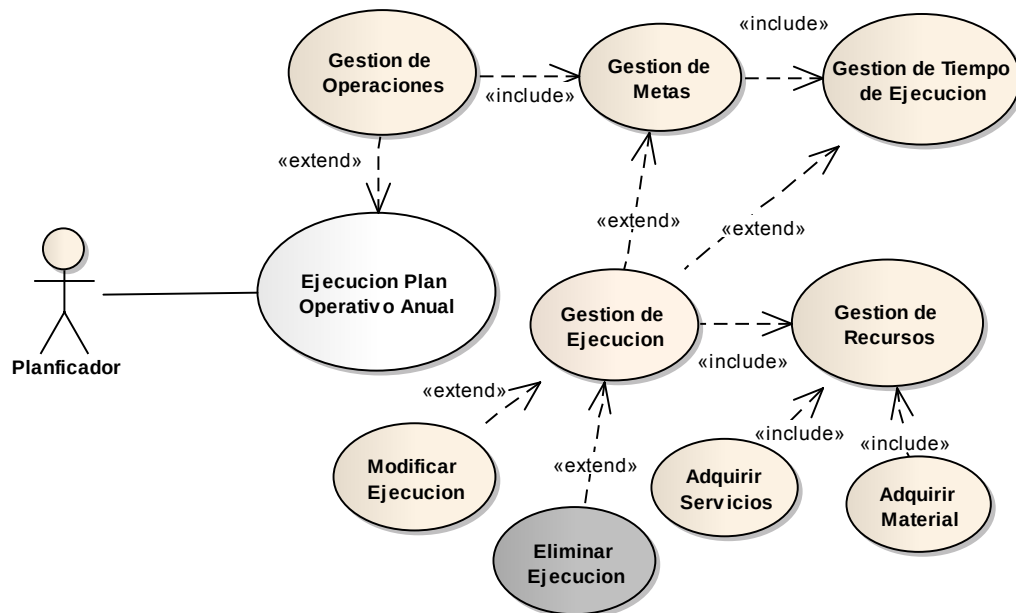


Figura 30 Especificación de casos de uso Eliminar Ejecución

Caso de uso:	Eliminar Ejecución		
Descripción: Permite eliminar un Recurso en Ejecución registrado en el sistema.			
Actores: Planificador.			
Precondiciones: Actor debe haber iniciado sesión.			
Flujo Normal: 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Aceptar, El sistema realiza la eliminación de un Recurso. 4. El sistema regresa a la pantalla de Gestión de Ejecución.		Flujo Alternativo: 1. Por cancelar el sistema no realizará ninguna acción.	
Pos condiciones: ninguno.			

Cuadro 17Especificación de casos de uso Eliminar Ejecución

II.1.1.3.3.Especificación de casos de uso del Administrador

II.1.1.3.3.1.Especificación de casos de uso Evaluación Plan Operativo Anual Selección de Área

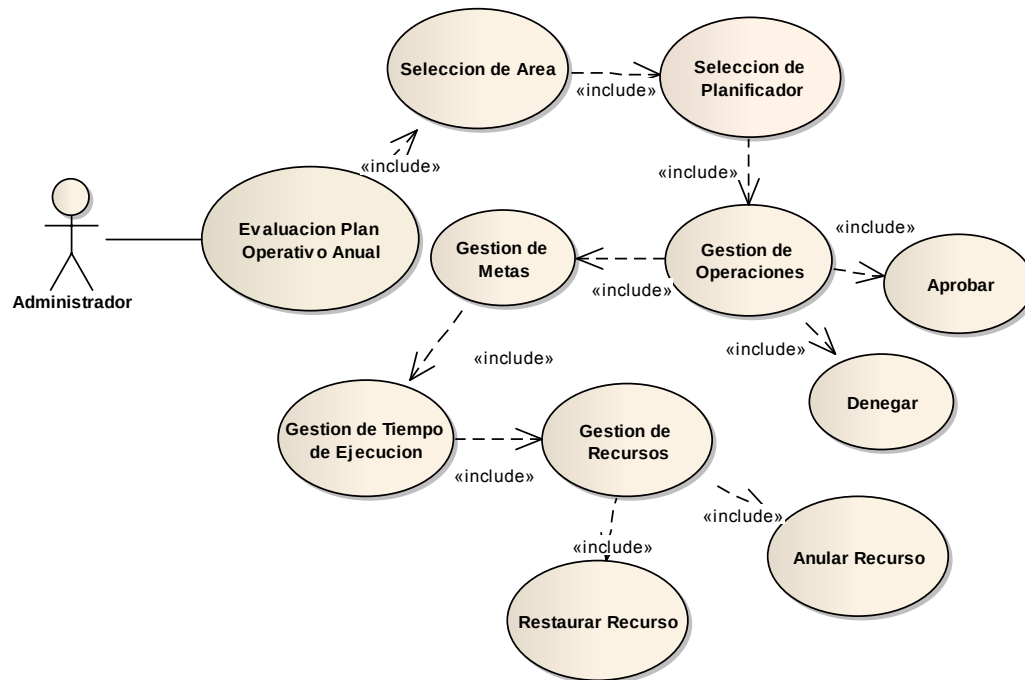


Figura 31Especificación de casos de uso Evaluación Plan Operativo Anual Selección de Área

Caso de uso:	Evaluación Plan Operativo Anual
Descripción: Permite Listar las áreas, Planificadores, Operaciones, Metas, Tiempo de Ejecución y Recursos.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: 1. El sistema muestra Listado de las Áreas en el sistema. 2. El Actor presiona Planificador y se mostrar un listado de los Planificadores del área. 3. El Actor presiona Objetivos y se mostrar un listado de los Objetivos del planificador podrá o denegar un objetivo.	Flujo Alternativo: 1. Por Atrás volverá a la pantalla anterior de la Gestión.

4. El Actor presiona Metas y se mostrar un listado de las Metas del objetivo. 5. El Actor presiona Tiempo de Ejecución y se mostrar un listado de los meses a efectuarse las metas. 6. El Actor presiona Recursos y se mostrar un listado de los recursos de la meta podrá anular o restaurar un recurso en el sistema.	
Pos condiciones: Ninguno	

Cuadro 18Especificación de casos de uso Evaluación Plan Operativo Anual Selección de Área

II.1.1.3.3.2. Especificación de casos de uso Determinación Techo Presupuestario

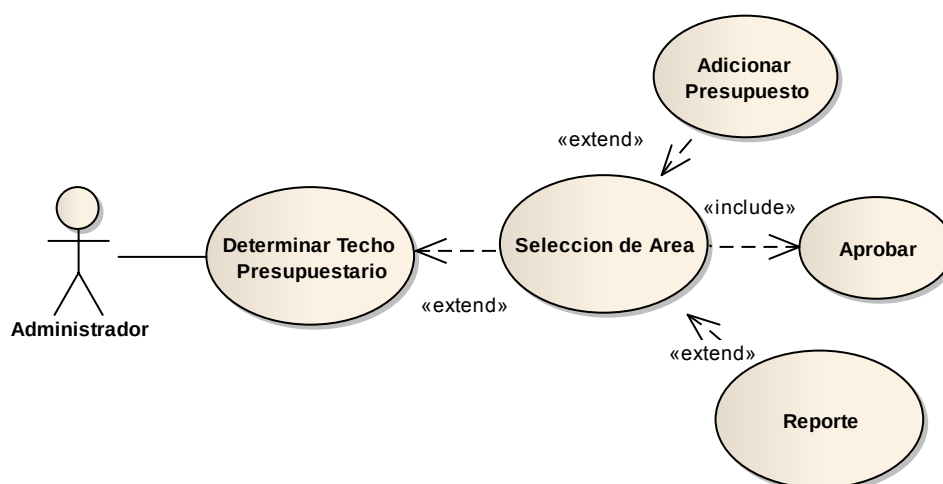


Figura 32Especificación de casos de uso Determinación Techo Presupuestario

Caso de uso:	Determinación Techo Presupuestario
Descripción:	Permite listar áreas con los montos presupuestario en el sistema.
Actores:	Administrador.
Precondiciones:	Actor debe haber iniciado sesión y haber seleccionado una Operación.
Flujo Normal: 1. El sistema muestra Listado de las Áreas en el sistema con sus respectivos montos que algunas áreas estén asignado. 2. Podrá adicionar el monto para el área con tiempo límite de realizar la formulación del	Flujo Alternativo:

POA. 3. Podrá realizar la aprobación del techo presupuestario para toda las áreas. 4. Tendrá la opción de realizar el reporte.	
Pos condiciones: Ninguno	

Cuadro 19 Especificación de casos de uso Determinación Techo Presupuestario

II.1.1.3.3.3. Especificación de casos de uso Adicionar Presupuesto

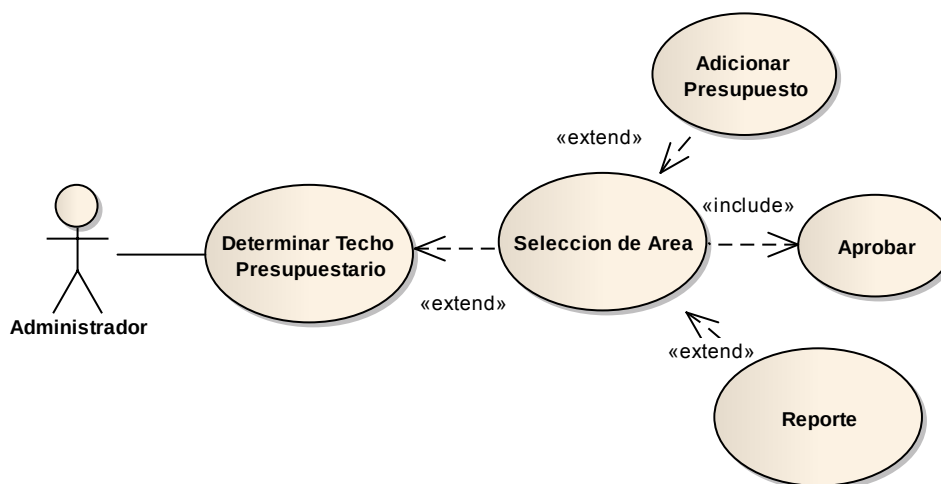


Figura 33 Especificación de casos de uso Adicionar Presupuesto

Caso de uso:	Adicionar Presupuesto	
Descripción:	Permite adicionar presupuesto en el sistema.	
Actores:	Administrador.	
Precondiciones:	Actor debe haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo: 1. Por cancelar el sistema no realizará ninguna acción.	
1. El actor presiona Adicionar. 2. El sistema muestra el formulario. 3. Introduce el monto y la fecha para el rango de tolerancia. 4. Por Aceptar, El sistema realiza la validación de los datos. 5. El sistema regresa a la pantalla de		

selección de área.	
Pos condiciones: ninguno.	

Cuadro 20Especificación de casos de uso Adicionar Presupuesto

II.1.1.3.3.5. Especificación de casos de uso Generar Reportes de información

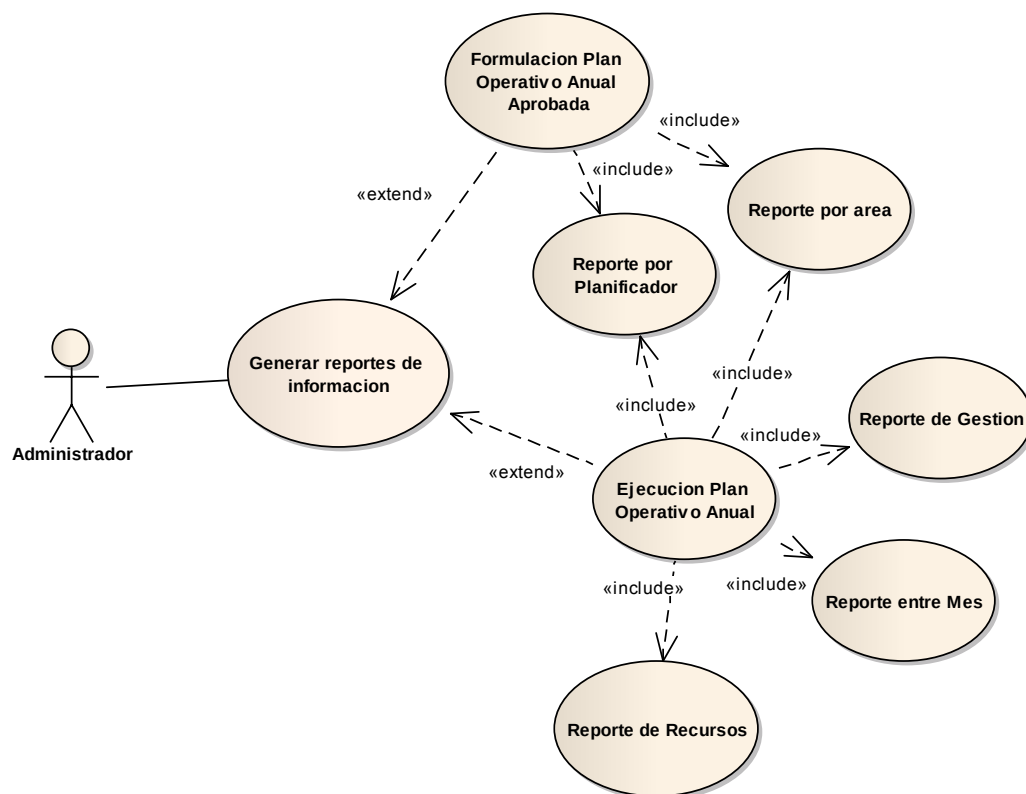


Figura 57Especificación de casos de uso Generar Reportes de información

Caso de uso:	Generar Reportes de información
Descripción: Permitevisualizar las opciones que podrán generar reportesen el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo:
1. Se mostrara una lista de opciones de reportes. 2. El actor presiona Reporte y descargara en archivo PDF del	

reporte seleccionado.	
Pos condiciones: ninguno.	

Cuadro 21Especificación de casos de uso Generar Reportes de información

Cuadro 25

II.1.1.3.3.6. Especificación de casos de uso Reporte – Formulación Aprobada

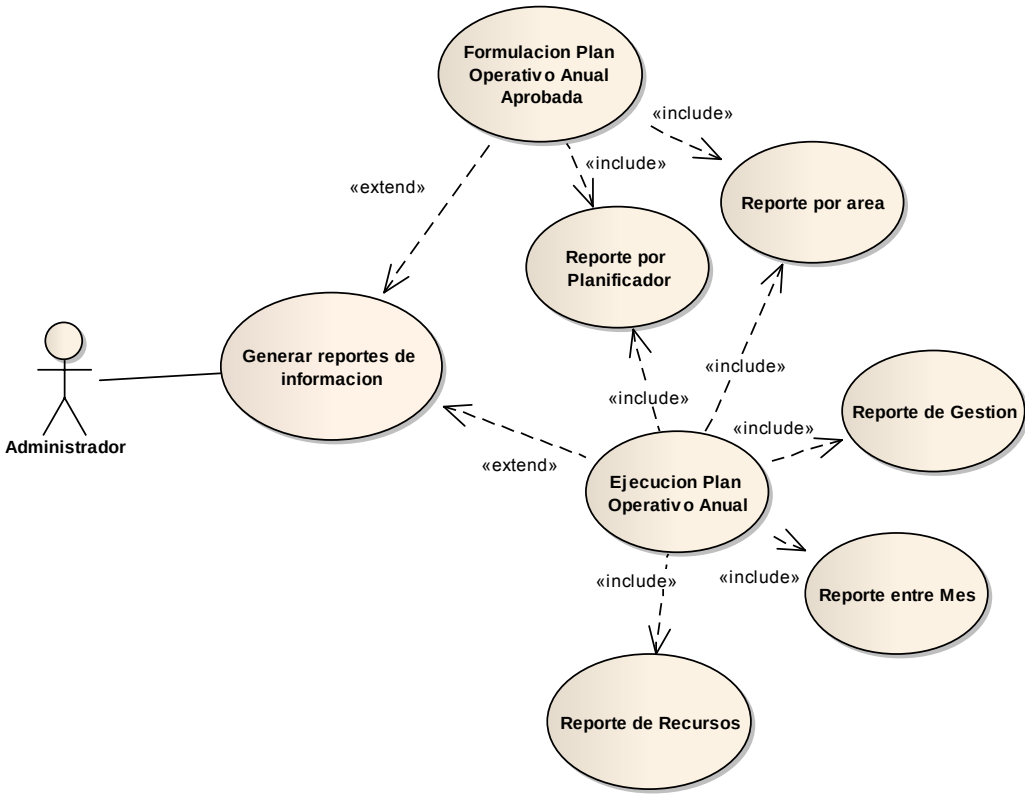


Figura 34Especificación de casos de uso Reporte – Formulación Aprobada

Caso de uso:	Reporte – Formulación Aprobada	
Descripción:	Permite visualizar opciones de reporte de las formulaciones en el sistema.	
Actores:	Administrador.	
Precondiciones:	Actor debe haber iniciado sesión.	
Flujo Normal:	1. El actor presiona Reporte:	Flujo Alternativo:

- Reporte por Área. - Reporte por Planificador. 2. Se descargar el archivo en PDF	
Pos condiciones: ninguno.	

Cuadro 22Especificación de casos de uso Reporte – Formulación Aprobada

II.1.1.3.3.3.7. Especificación de casos de uso Reporte – Ejecución Aprobada

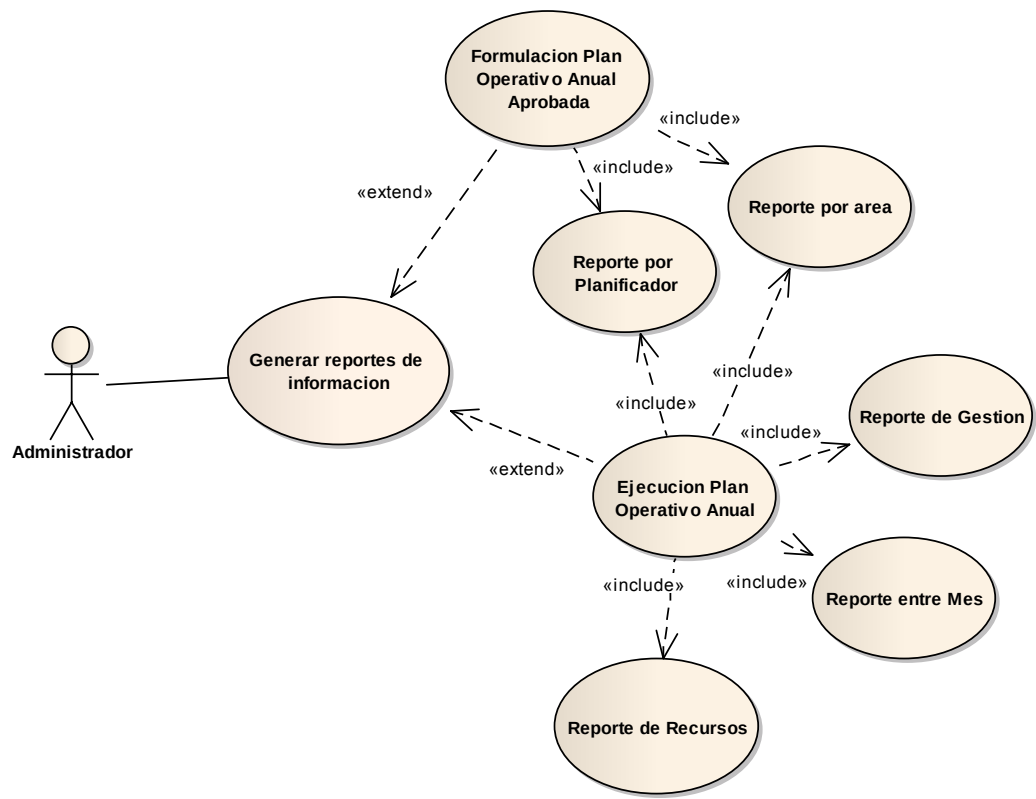


Figura 35Especificación de casos de uso Reporte – Ejecución Aprobada

Caso de uso:	Reporte – Ejecución Aprobada	
Descripción:	Permite visualizar opciones de reporte de las Ejecuciones en el sistema.	
Actores:	Administrador.	
Precondiciones:	Actor debe haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo:	
1. El actor presiona Reporte:		

- Reporte por Área. - Reporte por Planificador. - Reporte por Mes - Reporte de Gestión	
2. Se descargar el archivo en PDF	
Pos condiciones: ninguno.	

Cuadro 23Especificación de casos de uso Reporte – Ejecución Aprobada

II.1.1.3.3.4.Especificación de casos de uso del Enc. Almacenes

II.1.1.3.3.4.1. Especificación de casos de uso Gestión de Materiales

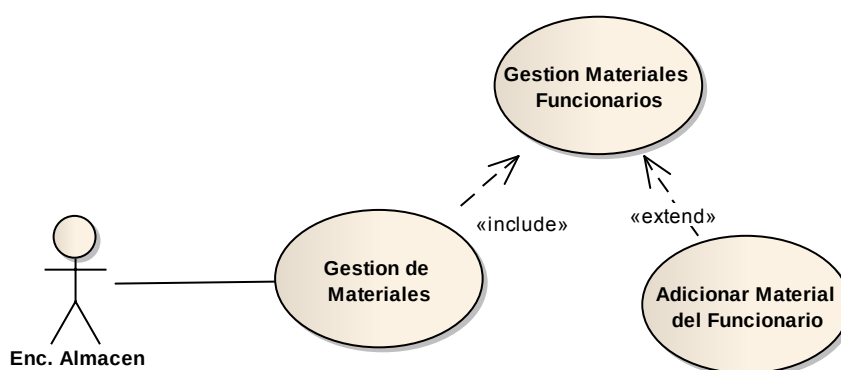


Figura 36Especificación de casos de uso Gestión de Materiales

Caso de uso:	Gestión de Materiales	
Descripción:	Permite Listar y AdicionaMateriales del Funiconario.	
Actores:	Enc. Almacenes.	
Precondiciones:	Actor debe haber iniciado sesión.	
Flujo Normal:	1. El sistema muestra Listado de los Materiales en el sistema. 2. El actor realiza la Búsqueda de los materiales en el Listado. 3. El actor realiza Activo/Inactivo de los Materiales. 4. El sistema muestra las operaciones que cumple(n) con los Criterios de Búsqueda,	Flujo Alternativo:

Activo e Inactivo.	
Pos condiciones: Ninguno	

Cuadro 24Especificación de casos de uso Gestión de Materiales

II.1.1.3.3.4.2. Especificación de casos de uso Adicionar Material del Funcionario

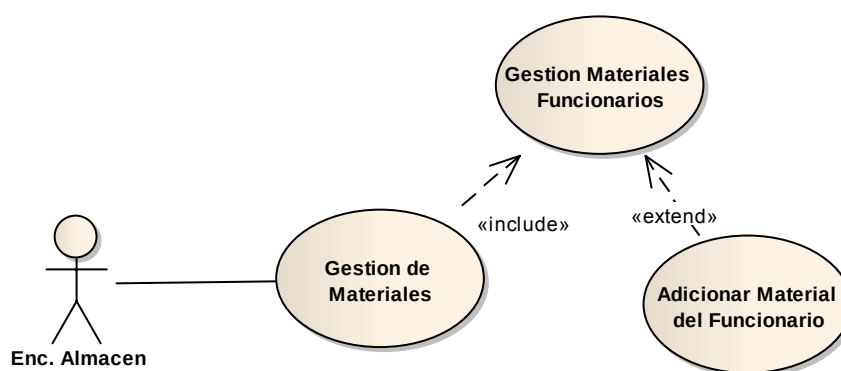


Figura 37Especificación de casos de uso Adicionar Material del Funcionario

Caso de uso:	AdicionarMaterial del Funcionario	
Descripción:	Permite registrar nuevo Material adicionado por el Funcionario en el sistema.	
Actores:	Enc. Almacenes.	
Precondiciones:	Actor debe haber iniciado sesión.	
Flujo Normal:	1. El actor presiona botón Adicionar del Funcionario. 2. El sistema le muestra un listado de los materiales ingresado por los funcionarios. 3. Al presionar Adicionar. 4. Se mostrara un formulario. 5. El actor introduce datos del material en el formulario. 6. Selecciona botón Guardar.	
Flujo Alternativo:	1. Selecciona botón Cancelar, se regresa a la Pantalla de Gestión de Materiales. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve	

7. Se valida datos del formulario que son obligatorios estén introducidos.	al paso 5 para que el actor pueda ingresar la información que falta.
8. El sistema guarda datos del formulario y muestra la Pantalla de Gestión de Materiales.	
Pos condiciones: ninguno.	

Cuadro 25Especificación de casos de uso Adicionar Material del Funcionario

II.1.1.3.3.5.Especificación de casos de uso Jefe de R.R.H.H.

II.1.1.3.3.5.1. Especificación de casos de uso Gestión de Servicios

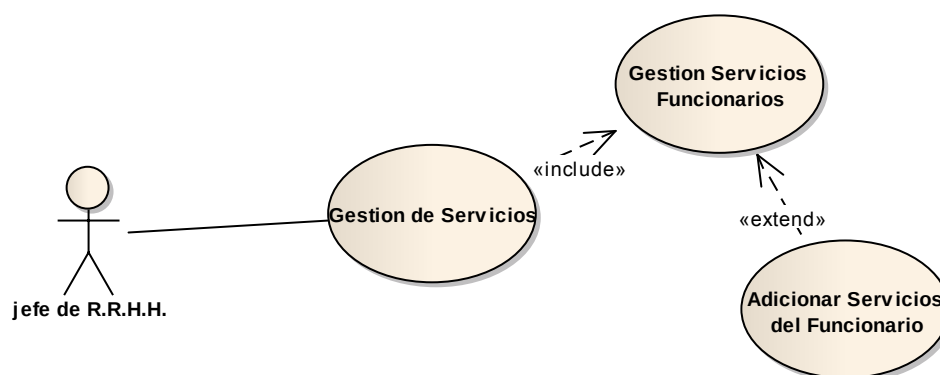


Figura 38Especificación de casos de uso Gestión de Servicios

Caso de uso:	Gestión de Servicios	
Descripción:	Permite Listar y Adicionar Servicios del Funcionario.	
Actores:	Jefe R.R.H.H.	
Precondiciones:	Actor debe haber iniciado sesión.	
Flujo Normal:	1. El sistema muestra Listado de los Servicios en el sistema. 2. El actor realiza la Búsqueda de los Servicios en el Listado. 3. El actor realiza Activo/Inactivo de los Servicios. 4. El sistema muestra las operaciones que cumple(n) con los Criterios de Búsqueda, Activo e Inactivo.	Flujo Alternativo:

Pos condiciones: Ninguno

Cuadro 26 Especificación de casos de uso Gestión de Servicios

II.1.1.3.3.5.2. Especificación de casos de uso Adicionar Material del Funcionario

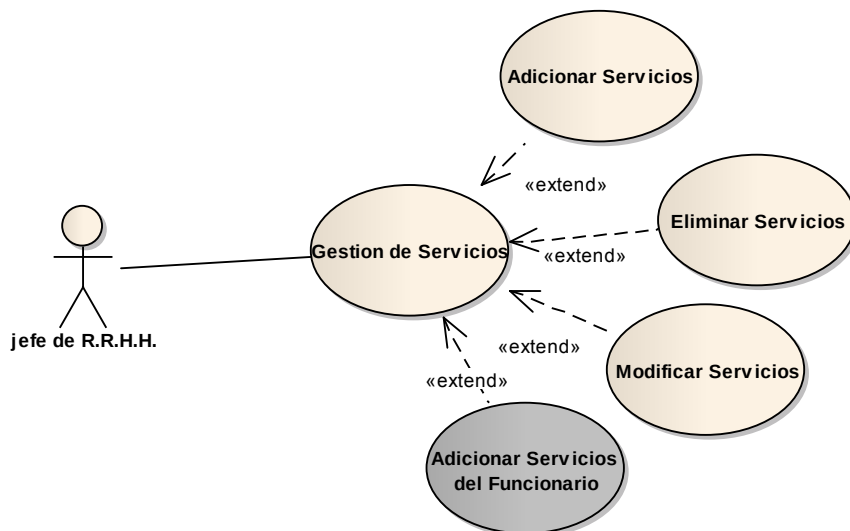


Figura 39 Especificación de casos de uso Adicionar Material del Funcionario

Caso de uso:	AdicionarServiciodel Funcionario
Descripción:	Permite registrar nuevo Servicioadicionado por el Funcionario en el sistema.
Actores:	Jefe R.R.H.H.
Precondiciones:	Actor debe haber iniciado sesión.
Flujo Normal: 1. El actor presiona botón Adicionar del Funcionario. 2. El sistema le muestra un listado de los Servicioingresado por los funcionarios. 3. Al presionar Adicionar. 4. Se mostrara un formulario. 5. El actor introduce datos del Servicioen el formulario. 6. Selecciona botón Guardar. 7. Se valida datos del formulario que son obligatorios estén introducidos. 8. El sistema guarda datos del formulario y muestra la	Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la Pantalla de Gestión de Servicio. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 5 para que el actor pueda ingresar la información que falta.

Pantalla de Gestión de Servicio.	
Pos condiciones: ninguno.	

Cuadro 27 Especificación de casos de uso Adicionar Material del Funcionario

II.1.1.3.4. Prototipo de Pantallas

II.1.1.3.4.1. Pantalla Ingresar al Sistema

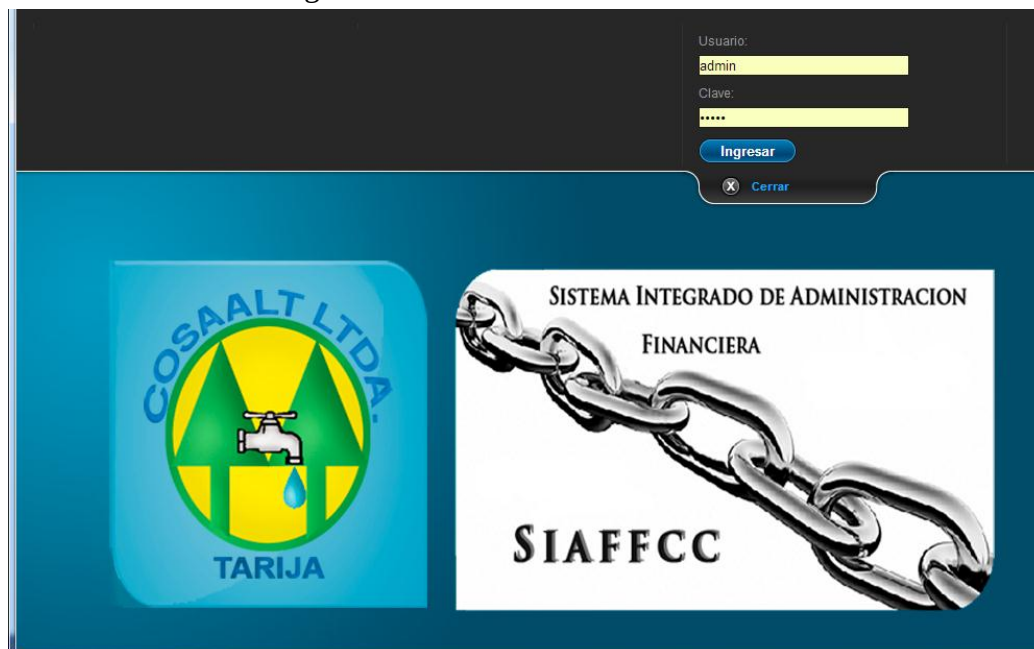


Figura 40 Pantalla Ingresar al Sistema

II.1.1.3.4.2. Pantalla Principal

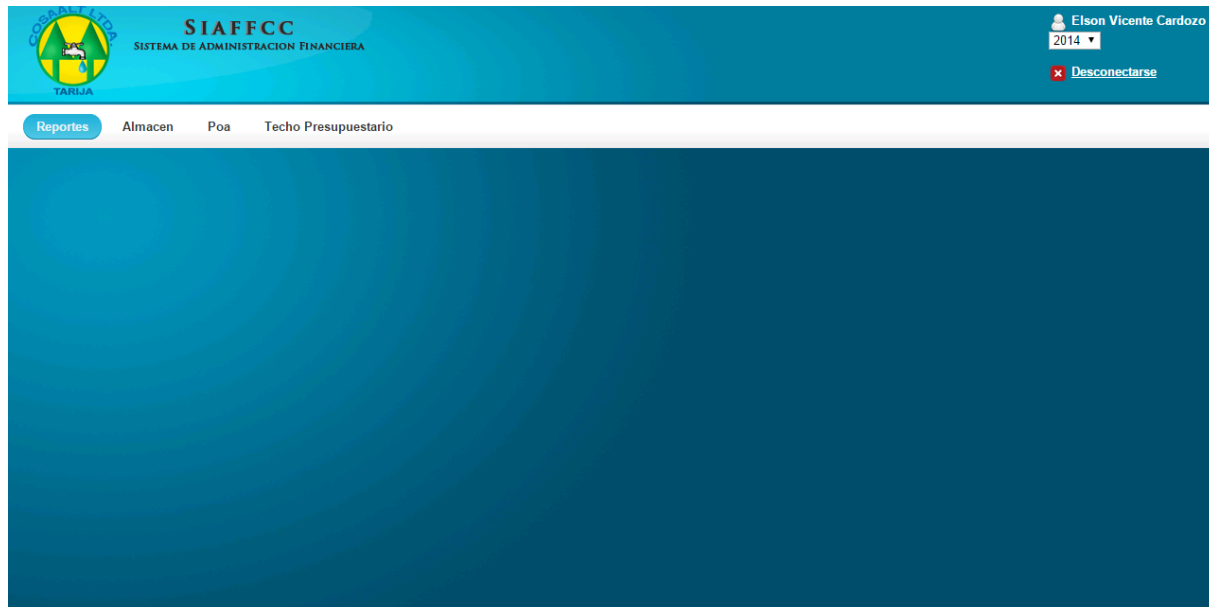


Figura 41 Pantalla Principal

II.1.1.3.4.3. Pantalla Determinación de Techo Presupuestario

II.1.1.3.4.3.1. Pantalla Selección de Área



Figura 42 Pantalla Selección de Área

II.1.1.3.4.3.2. Pantalla Adicionar Monto

Figura 43 Pantalla Adicionar Monto

II.1.1.3.4.3.3. Pantalla Aprobar Techo Presupuestario

n°	Monto
1 CONSEJO D	Falta Monto
2 CONSEJO D	Falta Monto
3 GERENTE GENERAL	Falta Monto
4 AUDITORIA INTERNA	Falta Monto
5 ASESORIA LEGAL	Falta Monto
6 PLANIFICACION	Falta Monto
7 GERENCIA ADMINISTRATIVA	Falta Monto
8 DPTO. RECURSOS HUMANOS	Falta Monto

Figura 44 Pantalla Aprobar Techo Presupuestario

II.1.1.3.4.4. Pantalla Formulación Plan Operativo Anual (POA)

II.1.1.3.4.4.1. Pantalla Gestión de Operaciones

Nº	Descripción	Cant. Planif.	Monto Planif.	Proceso
1	Realizar el mantenimiento y desarrollo de Sistema (Hardware y Software).	0 %	12310.0	Aprobado
2	Realizar el mantenimiento periodico de medidores.	0 %	0.0	En Espera
3	Mejorar la recaudacion diaria y dar mayor comodidad a los usuarios incrementando los puntos de cobranza.	0 %	3300.0	Modificar Eliminar Metas Enviar
Total de las actividades			15610.0	

Figura 45 Pantalla Gestión de Operaciones

II.1.1.3.4.4.2. Pantalla Adicionar Operaciones

Adicionar Determinacion de Objetivos

Descripcion:

[Informacion Poa](#)

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

Figura 46 Pantalla Adicionar Operaciones

II.1.1.3.4.4.3. Pantalla Modificar Operaciones

The screenshot displays the SIAFFCC (Sistema de Administración Financiera) web application. The header includes the COSANU LTDA logo, the system name, and user information for Elson Vicente Cardozo in 2014, with a 'Desconectarse' link. A navigation bar contains 'Reportes', 'Almacen', 'Poa', and 'Techo Presupuestario'. The main content area is titled 'Modificar Deteminacion de Objetivos' and features a form with a description field containing the text: 'Mejorar la recaudacion diaria y dar mayor comodidad a los usuarios incrementando los puntos de cobranza.' Below the form are 'Guardar' and 'Cancelar' buttons. A tab labeled 'Informacion Poa' is visible on the right side of the form.

Figura 47 Pantalla Modificar Operaciones

II.1.1.3.4.4.4. Pantalla Eliminar Operaciones

COSEANT LTDA
TARIJA

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
Desconectarse

Reportes Almacen Poa Techo Presupuestario

DETERMINACION DE ACTIVIDADES EMPRESARIAL

eliminando

Seguro que quiere eliminar la actividad?

OK Cancel

Nº	Descripcion	Cant. Planif.	Monto Planif.	Proceso
1	Realizar el mantenimiento y desarrollo de Sistema (Hardware y Software).	0 %	12310.0	Aprobado
2	Realizar el mantenimiento periodico de medidores.	0 %	0.0	En Espera
3	Mejorar la recaudacion diaria y dar mayor comodidad a los usuarios incrementando los puntos de cobranza.	0 %	3300.0	Modificar Eliminar Metas Enviar
Total de las actividades			15610.0	

Figura 48 Pantalla Eliminar Operaciones

II.1.1.3.4.4.5. Pantalla Gestión Metas

COSEANT LTDA
TARIJA

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
Desconectarse

Reportes Almacen Poa Techo Presupuestario

METAS DE GESTION

10 filas por pagina

Buscar:

Atras Adicionar

Inactivo Siguiente

Objetivo de Gestion Consejo de Administracion		Plazos		Programacion		Proceso
Nº	Descripcion	Inicio	Fin	Cant. Planif.	Monto Planif.	
1	Mantenimiento de Harware y Software	01/01/2015	31/12/2015	100 %	11950.0	Modificar Eliminar Recursos
2	Mantenimiento de red	01/01/2015	31/12/2015	100 %	360.0	Modificar Eliminar Recursos
3	Mantenimiento Sistema Area Comercial	01/01/2015	31/12/2015	100 %	0.0	Modificar Eliminar Recursos
Total de las actividades					12310.0	

Mostrando 1 a 3 de 3 entradas

Atras 1 Siguiente

Figura 49 Pantalla Gestión Metas

II.1.1.3.4.4.6. Pantalla Adicionar Metas

Adicionar Actividad Poa

Descripcion:

Cantidad Planificado:

Fecha de inicio:

Fecha Final:

Figura 50 Pantalla Adicionar Metas

II.1.1.3.4.4.7. Pantalla Adicionar Tiempo de Ejecución

Adicionar Tiempo de Ejecucion de Actividad Poa

Tiempo de Ejecuciones:

☐	☐	☒
Enero	Mayo	Septiembre
☒	☐	☒
Febrero	Junio	Octubre
☐	☒	☐
Marzo	Julio	Noviembre
☐	☒	☐
Abril	Agosto	Diciembre

Figura 51 Pantalla Adicionar Tiempo de Ejecución

II.1.1.3.4.4.8. Pantalla Eliminar Meta

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

METAS DE GESTION

10 filas por pagina

Buscar:

[Atras](#) [Adicionar](#) [Inactivo](#) [Siguiete](#)

Nº	Descripcion	Inicio	Fin	Cant. Planif.	Monto Planif.	Proceso
1	Mantenimiento de Harware y Software	01/01/2015	31/12/2015	100 %	11950.0	Modificar Eliminar Recursos
2	Mantenimiento de red	01/01/2015	31/12/2015	100 %	360.0	Modificar Eliminar Recursos
3	Mantenimiento Sistema Area Comercial	01/01/2015	31/12/2015	100 %	0.0	Modificar Eliminar Recursos
Total de las actividades					12310.0	

Mostrando 1 a 3 de 3 entradas

[← Atras](#) [1](#) [Siguiete →](#)

Figura 52 Pantalla Eliminar Meta

II.1.1.3.4.4.9. Pantalla Modificar Meta

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

Modificar Actividad Poa

Informacion Poa

Descripcion:

Cantidad Planificado:

Fecha de inicio:

Fecha Final:

[Enviar](#) [Cancelar](#)

Figura 53 Pantalla Modificar Meta

II.1.1.3.4.4.10. Pantalla Modificar Tiempo de Ejecución

Figura 54 Pantalla Modificar Tiempo de Ejecución

II.1.1.3.4.4.11. Pantalla Gestión Tiempo de Ejecución

Operacion de fucionamiento o inversion	Tiempo Previsto de ejecucion												Programacion		Procesos		
Descripcion	Ene.	Feb.	Mar.	Abr.	May.	Jun.	Jul.	Ago.	Sep.	Oct.	Nov.	Dic.	Cant. Planif.	Monto Planif.			
Mantenimiento de Harware y Software	X	X	X	X	X	X	X	X	X	X	X	X	100 %	11950.0	☒ Modificar	Eliminar	Recursos
Mantenimiento de red	X	X	X	X	X	X	X	X	X	X	X	X	100 %	360.0	☒ Modificar	Eliminar	Recursos
Mantenimiento Sistema Area Comercial	X	X	X	X	X	X	X	X	X	X	X	X	100 %	0.0	☒ Modificar	Eliminar	Recursos
Total de las actividades														0			

Figura 55 Pantalla Gestión Tiempo de Ejecución

II.1.1.3.4.4.12. Pantalla Eliminar Tiempo de Ejecución

COGAFYT-LDA **SIAFFCC** SISTEMA DE ADMINISTRACION FINANCIERA Elson Vicente Cardozo
2014 Desconectarse

Reportes Almacen Poa Techo Presupuestario

DETERMINACION DE OPERACIONES (Funciones o inversion)

Atras Inactivo

Operacion de fucionamiento o inversion	Ene. Feb. Mar.			Programacion		Procesos		
	Descripcion	Ene.	Feb.	Mar.	Cant. Planif.	Monto Planif.	Modificar	Eliminar
Mantenimiento de Harware y Software	X	X	X		100 %	11950.0	<a>Modificar	<a>Eliminar
Mantenimiento de red	X	X	X	X	100 %	360.0	<a>Modificar	<a>Eliminar
Mantenimiento Sistema Area Comercial	X	X	X	X	100 %	0.0	<a>Modificar	<a>Eliminar
Total de las actividades						0		

eliminando
Seguro que quiere eliminar la Meta?
OK Cancel

Figura 56 Pantalla Eliminar Tiempo de Ejecución

II.1.1.3.4.4.13. Pantalla Gestión Recursos

COGAFYT-LDA **SIAFFCC** SISTEMA DE ADMINISTRACION FINANCIERA Elson Vicente Cardozo
2014 Desconectarse

Reportes Almacen Poa Techo Presupuestario

ESTIMACION DEL COSTO REFERENCIAL POR OPERACIONES

10 filas por pagina
Buscar:

Atras Adicionar Inactivo

Tipo de recursos			Descripcion de los recursos	Uni.	Cant.	Precio unitario (Bs)	Costo (Bs)	Partida	Observacion	Procesos
R.R.H.H.	Materiales	Servicios								
	X		Monitores	PIEZA	5	1800.0	9000.0	39800		<a>Modificar <a>Eliminar
	X		Mouse	PIEZA	5	80.0	400.0	39800		<a>Modificar <a>Eliminar
	X		Teclados	PIEZA	5	150.0	750.0	39800	marca Delux	<a>Modificar <a>Eliminar
	X		Memoria Ram	PIEZA	10	180.0	1800.0	39800	1GB.	<a>Modificar <a>Eliminar
Total de los recursos							11950.0			

Mostrando 1 a 4 de 4 entradas
Atras 1 Siguiente

Figura 57 Pantalla Gestión Recursos

II.1.1.3.4.4.14. Pantalla Adicionar Recursos

II.1.1.3.4.4.14.1. Pantalla Selección Recurso



Figura 58 Pantalla Selección Recurso

II.1.1.3.4.4.14.2. Pantalla Gestión Recurso - Material



Figura 59 Pantalla Gestión Recurso - Material

II.1.1.3.4.4.14.3. Pantalla adicionar Material Funcionario

COMANDO EN JEFE FUERZA ARMADA
SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

Adicionar Material

Informacion Material

Descripcion: *

Costo: *

Unidad : ▼

Enviar Cancelar

Figura 60 Pantalla adicionar Material Funcionario

II.1.1.3.4.4.14.4. Pantalla adicionar Recurso – Material

COMANDO EN JEFE FUERZA ARMADA
SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

Adicionar Recursos Poa

Informacion Recursos Poa

Descripcion: Fuente de poder de 500 w. ✓

Partida: 39800 ✓

Unidad : PIEZA ✓

Precio Unitario : 250.0

Cantidad : Ingresar la cantidad deseada

Observacion : ✓

Enviar Cancelar

Figura 61 Pantalla adicionar Recurso – Material

II.1.1.3.4.5. Pantalla Evaluación Plan Operativo Anual

II.1.1.3.4.5.1. Pantalla Selección de Área



n°	Area	Seleccion
1	CONSEJO DE ADMINISTRACION	Funcionario
2	CONSEJO DE VIGILANCIA	Funcionario
3	GERENTE GENERAL	Funcionario
4	AUDITORIA INTERNA	Funcionario
5	ASESORIA LEGAL	Funcionario
6	PLANIFICACION	Funcionario
7	GERENCIA ADMINISTRATIVA	Funcionario
8	DPTO. RECURSOS HUMANOS	Funcionario
9	DPTO. CONTABILIDAD Y FINANZAS	Funcionario
10	DPTO. LICITACIONES Y ADQUISICIONES	Funcionario

Figura 62 Pantalla Selección de Área

II.1.1.3.4.5.2. Pantalla Selección de Funcionario



Area		
n°	Apellido y Nombre	Seleccion
1	Lorenzo Muñoz Muñoz	POA
2	Elson Vicente Cardozo	POA

Figura 63 Pantalla Selección de Funcionario

II.1.1.3.4.5.3. Pantalla Gestión de Objetivos - Evaluación

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo

2014 ▼

[Desconectarse](#)

Reportes

Almacen

Poa

Techo Presupuestario

EVALUACION DE OBJETIVO

Funcionario				Q Ver
Objetivo de Gestion Empresarial			Meta (Resultado Esperado)	Procesos
Nº	Descripcion	Cantidad	Monto	
1	Realizar el mantenimiento y desarrollo de Sistema (Hardware y Software).	0 %	12310.0	Metas Aprobado Denegar
2	Realizar el mantenimiento periodico de medidores.	0 %	0.0	Aprobado
3	Mejorar la recaudacion diaria y dar mayor comodidad a los usuarios incrementando los puntos de cobranza.	0 %	3300.0	Metas Aprobado Denegar
Total de las actividades			15610.0	

Figura 64 Pantalla Gestión de Objetivos - Evaluación

II.1.1.3.4.5.4. Pantalla Gestión de Metas - Evaluación

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo

2014 ▼

✕ Desconectarse

Reportes

Almacen

Poa

Techo Presupuestario

EVALUACION DE METAS

10 ▼ filas por pagina

Buscar:

◀ Atras		Siguiente ▶				
Objetivo de Gestion Consejo de Administracion		Plazos		Programacion		Proceso ▴
Nº	Descripcion ▴	Inicio ▴	Fin ▴	Cantidad ▴	Monto ▴	
1	Mantenimiento de Hardware y Software	01/01/2015	31/12/2015	100 %	11950.0	Recursos
2	Mantenimiento de red	01/01/2015	31/12/2015	100 %	360.0	Recursos
3	Mantenimiento Sistema Area Comercial	01/01/2015	31/12/2015	100 %	0.0	Recursos
Total de las actividades					12310.0	

Mostrando 1 a 3 de 3 entradas

← Atras

1

Siguiente →

Figura 65 Pantalla Gestión de Metas - Evaluación

II.1.1.3.4.5.5. Pantalla Gestión de Tiempo de Ejecución - Evaluación

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

EVALUACION DE TIEMPO DE EJECUCION

10 filas por pagina
Buscar:

[Atras](#)

Operacion de funcionamiento o inversion		Tiempo Previsto de ejecucion												Programacion		Procesos
Cod	Descripcion	Enf.	Feb.	Mar.	Abr.	May.	Jun.	Jul.	Ag.	Sep.	Oct.	Nov.	Dic.	Cantidad	Mont	
	Mantenimiento de Harware y Software	X	X	X	X	X	X	X	X	X	X	X	X	100 %	11950.0	Recursos
	Mantenimiento de red	X	X	X	X	X	X	X	X	X	X	X	X	100 %	360.0	Recursos
	Mantenimiento Sistema Area Comercial	X	X	X	X	X	X	X	X	X	X	X	X	100 %	0.0	Recursos
Total de las actividades															12310.0	

Mostrando 1 a 3 de 3 entradas

[← Atras](#) [1](#) [Siguiete →](#)

Figura 66 Pantalla Gestión de Tiempo de Ejecución - Evaluación

II.1.1.3.4.5.6. Pantalla Gestión de Recursos - Evaluación

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

EVALUACION DE RECURSOS

10 filas por pagina
Buscar:

[Atras](#)

Operacion	Tipo de recursos			Descripcion de los recursos	Unid.	Cant.	Precio unitario (Bs)	Costo (Bs)	Partida	Observacion	Procesos
	R.R.H.H.	Materiales	Servicios								Modificar
\$menu.CodAct		X		Monitores	PIEZA	5	1800.0	9000.0	39800		Devolver
\$menu.CodAct		X		Mouse	PIEZA	5	80.0	400.0	39800		Quitar
\$menu.CodAct		X		Teclados	PIEZA	5	150.0	750.0	39800	marca Delux	Devolver
\$menu.CodAct		X		Memoria Ram	PIEZA	10	180.0	1800.0	39800	1GB.	Quitar
Total de los recursos								11950.0			

Mostrando 1 a 4 de 4 entradas

[← Atras](#) [1](#) [Siguiete →](#)

Figura 67 Pantalla Gestión de Recursos - Evaluación

II.1.1.3.4.6. Pantalla Ejecución Plan Operativo Anual

II.1.1.3.4.6.1. Pantalla Gestión de Objetivos - Ejecución

DETERMINACION DE OBJETIVOS DE GESTION EMPRESARIAL

Nº	Descripcion	Cantidad	Monto Plan.	proceso
1	Realizar el mantenimiento periodico de medidores.	0 %	0.0	Metas

Figura 68 Pantalla Gestión de Objetivos - Ejecución

II.1.1.3.4.6.2. Pantalla Gestión de Metas - Ejecución

OBJETIVO DE GESTION

10 filas por pagina

Buscar:

Objetivo de Gestion Consejo de Administracion		Plazos		Cantidad	Monto plan.	Proceso
Nº	Descripcion	Inicio	Fin			
1	Realizar el mantenimiento correctivo de medidores	01/01/2015	31/12/2015	100 %	0.0	Ejecucion

Mostrando 1 a 1 de 1 entradas

← Atras 1 Siguiete →

Figura 69 Pantalla Gestión de Metas - Ejecución

II.1.1.3.4.6.3. Pantalla Gestión de Tiempo de Ejecución - Ejecución

GOBIERNO DE LA REPUBLICA DE GUAYANA
SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

DETERMINACION DE OPERACIONES (Funciones o inversion)

OBJETIVO DE GESTION

10 filas por pagina

Buscar:

Atras

Operacion de fucionamiento o inversion		Tiempo Previsto de ejecucion												cantidad	Procesos
Cod	Descripcion	En	Fe	Ma	Ab	Ma	Ju	Ju	Ag	Sep	Oct	Nov	Dic		
	Mantenimiento de Harware y Software	X	X	X	X	X	X	X	X	X	X	X	X	100 %	Ejecucion
	Mantenimiento de red	X	X	X	X	X	X	X	X	X	X	X	X	100 %	Ejecucion
	Mantenimiento Sistema Area Comercial	X	X	X	X	X	X	X	X	X	X	X	X	100 %	Ejecucion

Mostrando 1 a 3 de 3 entradas

Atras 1 Provento2013 - EA

Figura 70 Pantalla Gestión de Tiempo de Ejecución - Ejecución

II.1.1.3.4.6.4. Pantalla Gestión de Recursos Ejecutando - Ejecución

GOBIERNO DE LA REPUBLICA DE GUAYANA
SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

RECURSOS EN EJECUCION

10 filas por pagina

Buscar:

Atras Adicionar

Tipo de recursos		Descripcion de los recursos	Uni.	Cant	Precio unitario(Bs)	Costo (Bs)	Fecha	Procesos	
R.R.H.H.	Materiales							Modificar	Eliminar
	X	Memoria Ram	PIEZA	2	180.0	360.0	2014-11-22	Modificar	Eliminar
	X	Monitores	PIEZA	1	1800.0	1800.0	2014-11-22	Modificar	Eliminar
Total de los recursos						2160.0			

Mostrando 1 a 2 de 2 entradas

← Atras 1 Siguiente →

Figura 71 Pantalla Gestión de Recursos Ejecutando - Ejecución

II.1.1.3.4.6.5. Pantalla Gestión de Recursos - Ejecución

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

ESTIMACION DEL COSTO REFERENCIAL POR OPERACIONES

10 filas por pagina

Buscar:

Atras

Tipo de recursos			Descripcion de los recursos	Uni.	Cant.	Precio unitario(Bs)	Costo (Bs)	Partida	Proceso
R.R.H.H.	Materiales	Servicios							
	X		Mouse	PIEZA	5	80.0	400.0	39800	Adquirir
	X		Teclados	PIEZA	5	150.0	750.0	39800	Adquirir
Total de los recursos							1150.0		

Mostrando 1 a 2 de 2 entradas

← Atras 1 Siguiente →

Figura 72 Pantalla Gestión de Recursos - Ejecución

II.1.1.3.4.6.6. Pantalla Adquirir Recursos - Ejecución

SIAFFCC
SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
2014
[Desconectarse](#)

Reportes Almacen Poa Techo Presupuestario

Adicionar Recursos Poa

Informacion Recursos Poa

Descripcion: ✓

Unidad : ✓

Precio Unitario :

Cantidad :

Observacion :

Figura 73 Pantalla Adquirir Recursos - Ejecución

II.1.1.3.4.6.7. Pantalla Modificar Recurso Ejecutar - Ejecución

Modificar Recursos Poa

Información Recursos Poa

Descripción: Memoria Ram ✓

Unidad : PIEZA ✓

Precio Unitario : 180.0

Cantidad : 2

Observación :

 ✓

Figura 74 Pantalla Modificar Recurso Ejecutar - Ejecución

II.1.1.3.4.6.7. Pantalla Eliminar Recurso Ejecutar - Ejecución

RECURSOS EN EJECUCION

10 filas por pagina

Buscar:

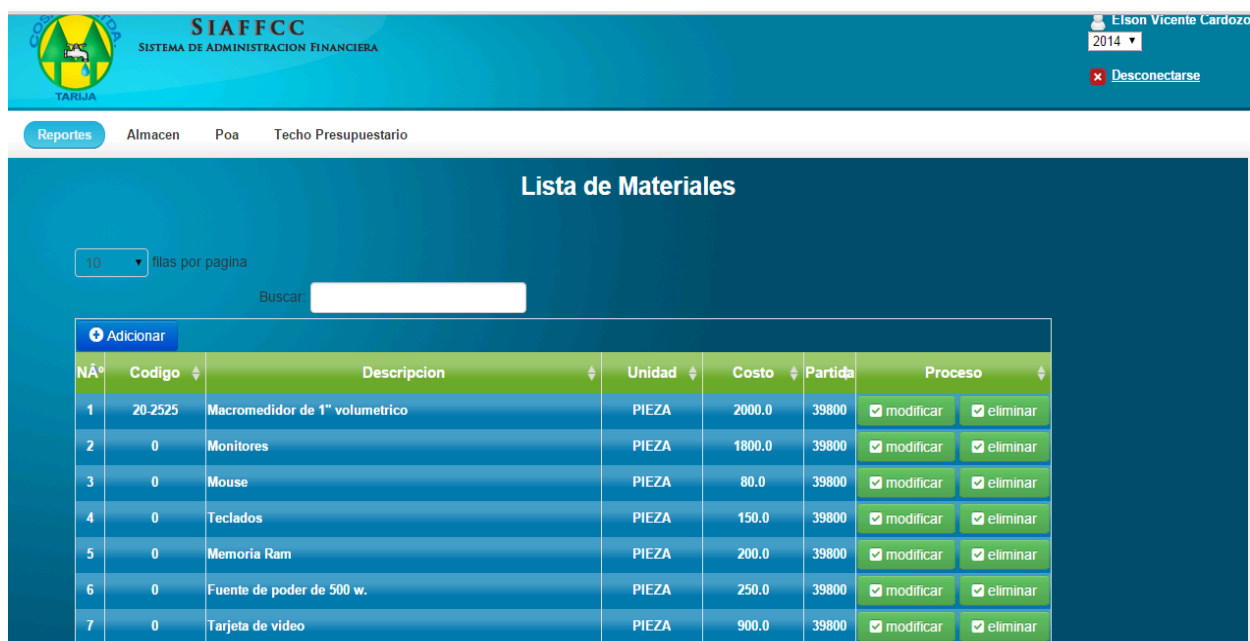
Tipo de recursos			Descripción de los recursos	Cantidad	Costo (Bs)	Fecha	Procesos
R.R.H.H.	Materiales	Servicios					
	X		Memoria Ram	PIEZA 2	180.0	2014-11-22	Modificar Eliminar
	X		Monitores	PIEZA 1	1800.0	2014-11-22	Modificar Eliminar
Total de los recursos						2160.0	

Mostrando 1 a 2 de 2 entradas

Figura 75 Pantalla Eliminar Recurso Ejecutar - Ejecución

II.1.1.3.4.7. Pantalla Gestión de Materiales

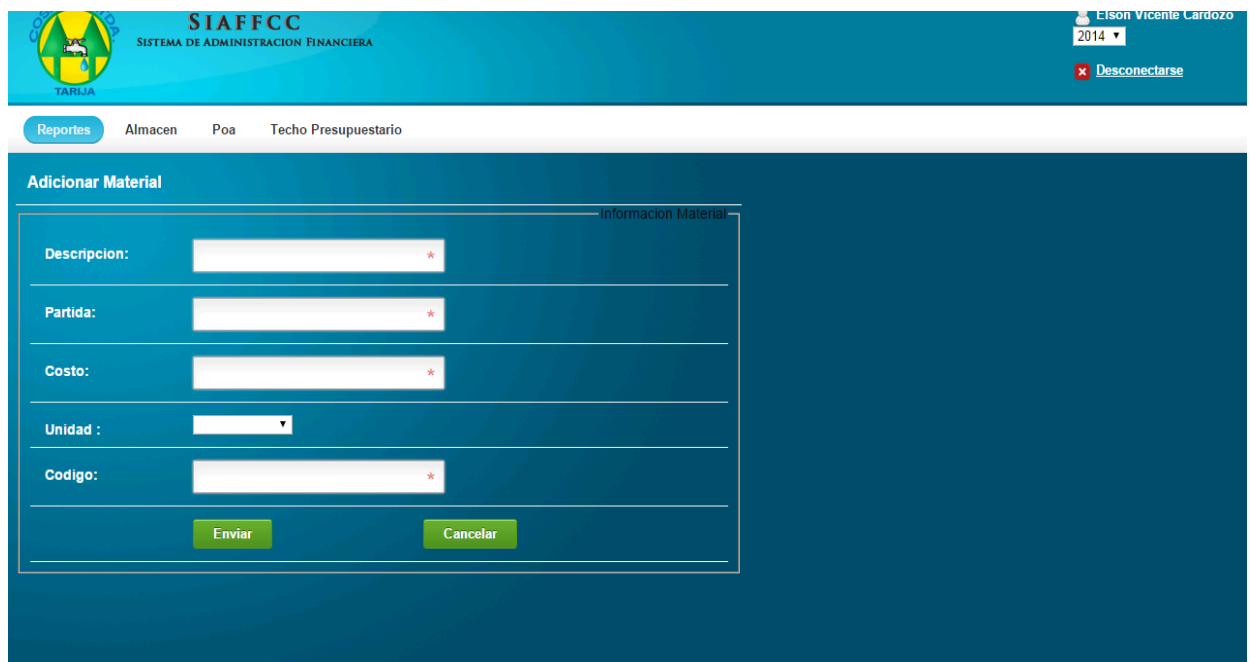
II.1.1.3.4.7.1. Pantalla Gestión de Materiales - Almacén



NÂ	Codigo	Descripcion	Unidad	Costo	Partida	Proceso
1	20-2525	Macromedidor de 1" volumetrico	PIEZA	2000.0	39800	☒ modificar ☒ eliminar
2	0	Monitores	PIEZA	1800.0	39800	☒ modificar ☒ eliminar
3	0	Mouse	PIEZA	80.0	39800	☒ modificar ☒ eliminar
4	0	Teclados	PIEZA	150.0	39800	☒ modificar ☒ eliminar
5	0	Memoria Ram	PIEZA	200.0	39800	☒ modificar ☒ eliminar
6	0	Fuente de poder de 500 w.	PIEZA	250.0	39800	☒ modificar ☒ eliminar
7	0	Tarjeta de video	PIEZA	900.0	39800	☒ modificar ☒ eliminar

Figura 76 Pantalla Gestión de Materiales - Almacén

II.1.1.3.4.7.2. Pantalla Adicionar Material - Almacén



Adicionar Material

Informacion Material

Descripcion: *

Partida: *

Costo: *

Unidad : *

Codigo: *

Figura 77 Pantalla Adicionar Material - Almacén

II.1.1.3.4.7.3. Pantalla Eliminar Material - Almacén


SIAFFCC
 SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
 2014
 Desconectarse

Reportes Almacen Poa Techo Presupuestario

Lista de Materiales

10 filas por pagina
 Buscar:

eliminando
 Seguro que quiere eliminar la Material?
 OK Cancel

NAº	Codigo	Descripcion	Partida	Proceso
1	20-2525	Macromedidor de 1" volumetrico	PIEZA 2000.0 39800	☒ modificar Eliminar
2	0	Monitores	PIEZA 1800.0 39800	☒ modificar Eliminar
3	0	Mouse	PIEZA 80.0 39800	☒ modificar Eliminar
4	0	Teclados	PIEZA 150.0 39800	☒ modificar Eliminar
5	0	Memoria Ram	PIEZA 200.0 39800	☒ modificar Eliminar
6	0	Fuente de poder de 500 w.	PIEZA 250.0 39800	☒ modificar Eliminar
7	0	Tarjeta de video	PIEZA 900.0 39800	☒ modificar Eliminar

Figura 78 Pantalla Eliminar Material - Almacén

II.1.1.3.4.7.4. Pantalla Modificar Material - Almacén


SIAFFCC
 SISTEMA DE ADMINISTRACION FINANCIERA

Elson Vicente Cardozo
 2014
 Desconectarse

Reportes Almacen Poa Techo Presupuestario

Modificar Material

Informacion Material

Descripcion: Macromedidor de 1 ☒

Partida: 39800 ☒

Costo: 2000.0 ☒

Unidad :

Codigo: 20-2525 ☒

Figura 79 Pantalla Modificar Material - Almacén

II.1.1.3.4.7.5. Pantalla Gestión Material - Funcionario

Lista de Materiales de los Funcionarios

10 filas por pagina

Buscar:

NA°	Codigo	Descripcion	Unidad	Costo	Partida	Proceso
1	Falta Codigo	Mesa	PIEZA	80.0	Falta Partida Pres.	☒ modificar eliminar incluir
2	Falta Codigo	silla	PIEZA	25.0	Falta Partida Pres.	☒ modificar eliminar incluir

Mostrando 1 a 2 de 2 entradas

← Atras 1 Siguiete →

Figura 80 Pantalla Gestión Material - Funcionario

II.1.1.3.4.7.6. Pantalla Modificar Material - Funcionario

Modificar Material

Información Material

Descripcion: ✓

Partida: ✓

Costo: ✓

Unidad :

Codigo: ✓

Figura 81 Pantalla Modificar Material - Funcionario

II.1.1.3.4.7.7. Pantalla Eliminar Material - Funcionario


SIAFFCC
 SISTEMA DE ADMINISTRACION FINANCIERA

2014 ▼
 Desconectarse

Reportes Almacen Poa Techo Presupuestario

Lista de Materiales de los Funcionarios

10 filas por pagina
 Buscar:

eliminando
 Seguro que quiere eliminar la Material?
 OK Cancel

Nº	Codigo	Descripcion	Partida	Proceso
1	Falta Codigo	Mesa	Falta Partida Pres.	☒ modificar Eliminar Incluir
2	Falta Codigo	silla	PIEZA 25.0	☒ modificar Eliminar Incluir

Mostrando 1 a 2 de 2 entradas

← Atras 1 Sigiente →

Figura 82 Pantalla Eliminar Material - Funcionario

II.1.1.3.5. Diagrama de Clases

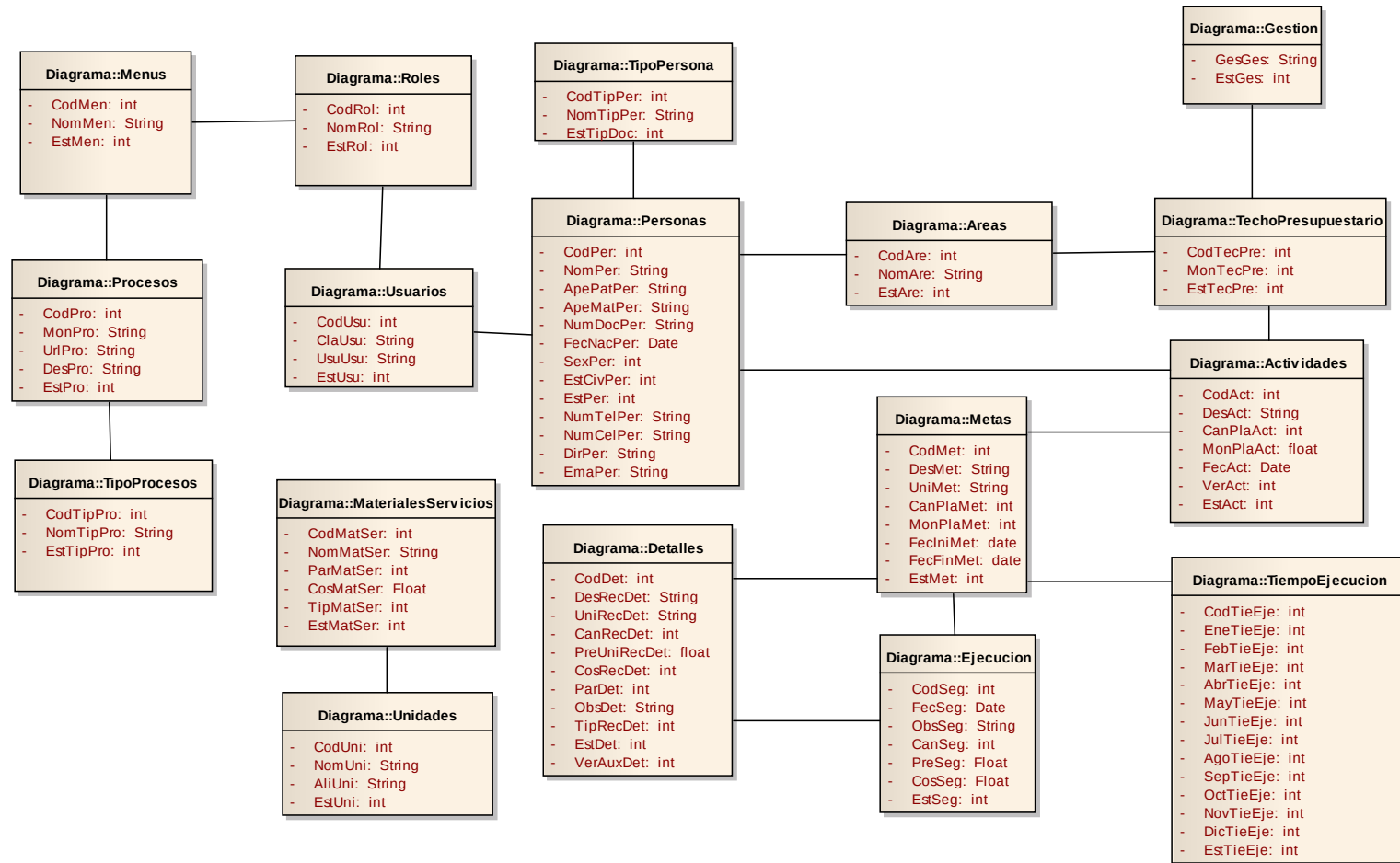


Figura 83 Diagrama de Clases

II.1.1.3.6. Diagrama de Actividad

II.1.1.3.6.1. Diagrama de Actividad: Caso de Uso Ingresar al Sistema

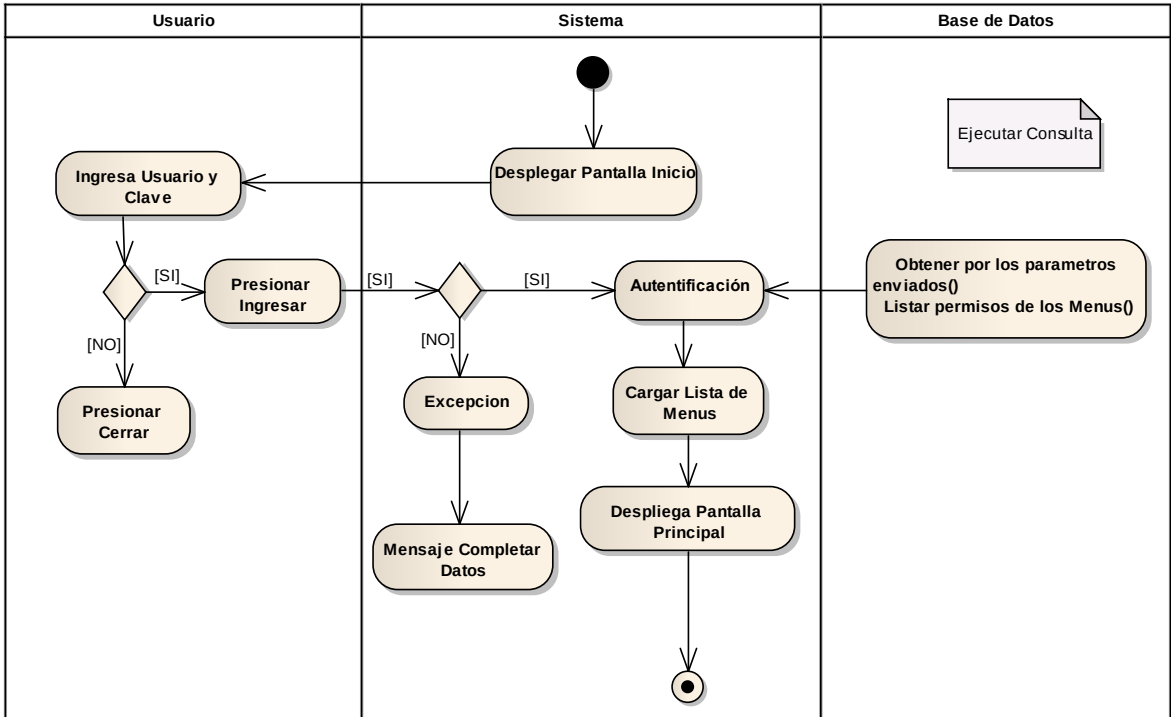


Figura 84Diagrama de Actividad: Caso de Uso Ingresar al Sistema

II.1.1.3.6.2. Diagrama de Actividad: Caso de Uso Formulación Plan Operativo Anual

II.1.1.3.6.2.1. Diagrama de Actividad: Caso de Uso Gestión Objetivos

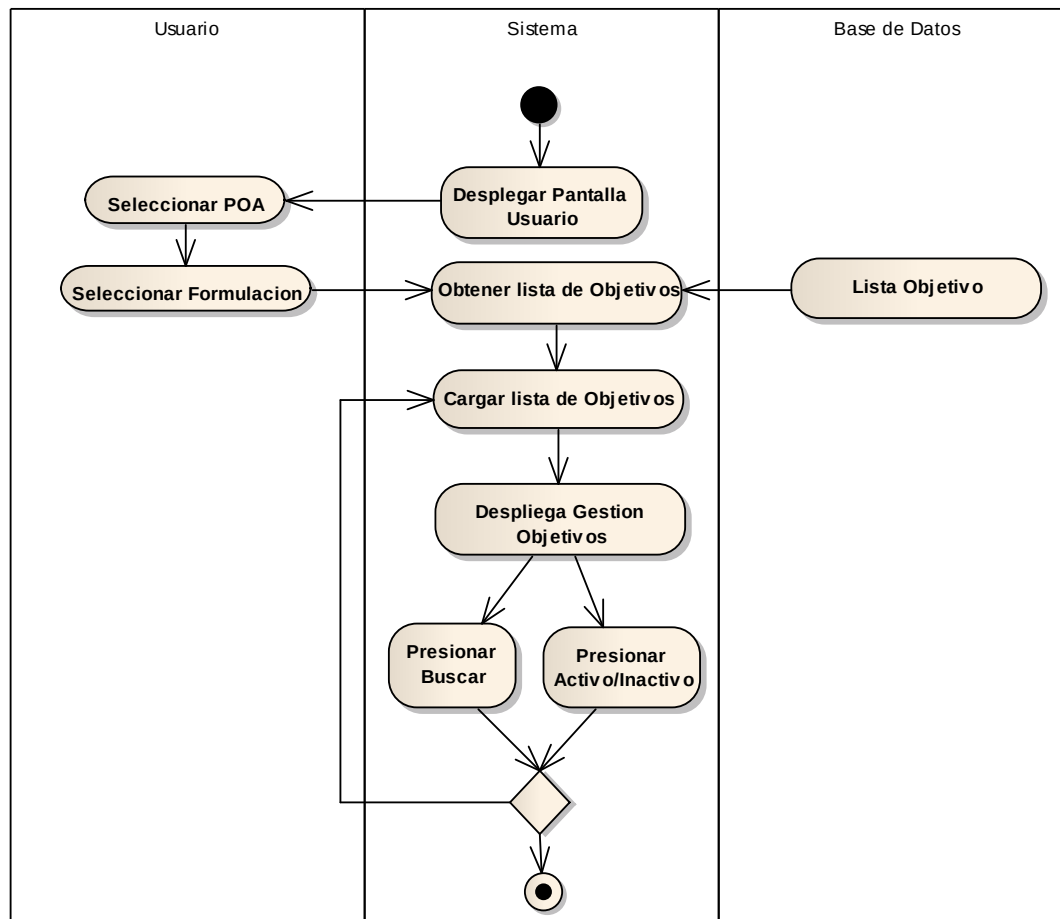


Figura 85 Diagrama de Actividad: Caso de Uso Gestión Objetivos

II.1.1.3.6.2.2. Diagrama de Actividad: Caso de Uso Adicionar Objetivos

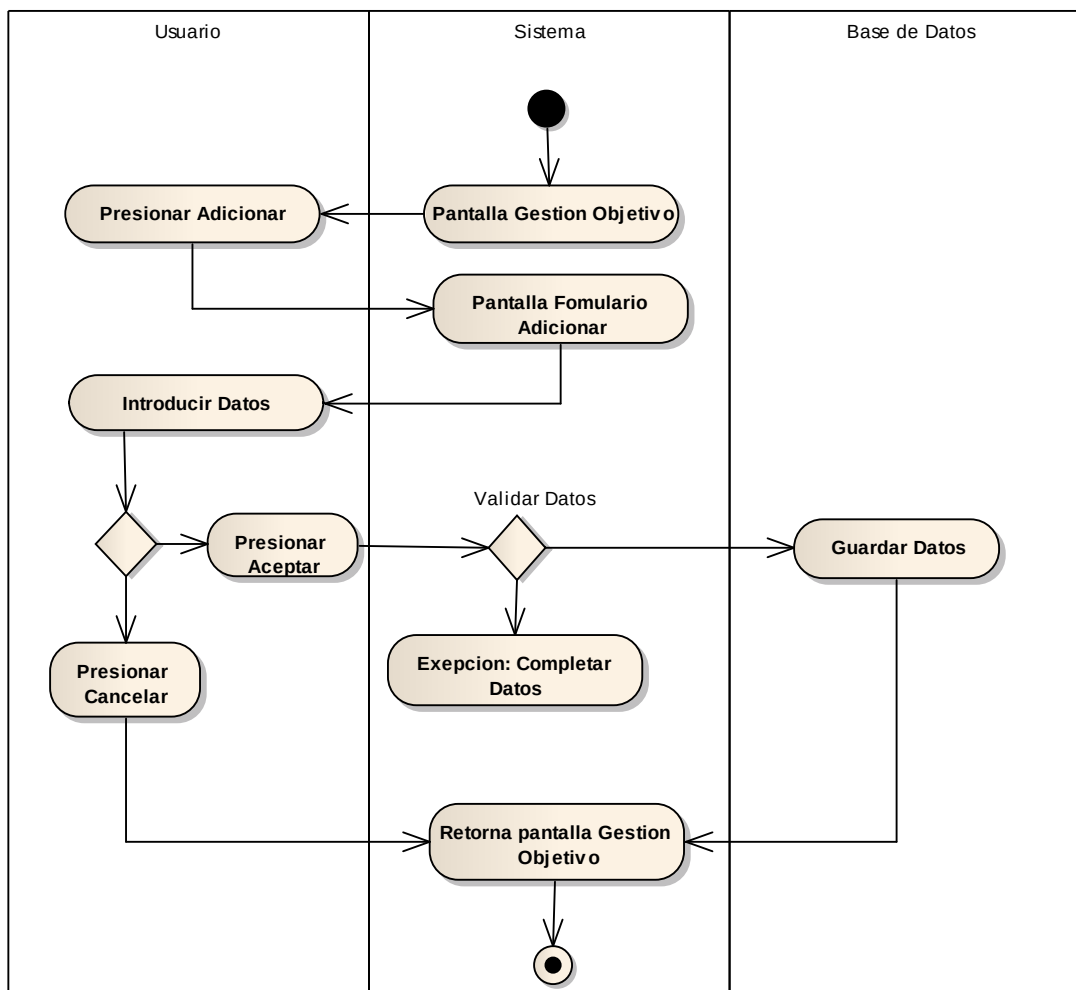


Figura 86Diagrama de Actividad: Caso de Uso Adicionar Objetivos

II.1.1.3.6.2.3. Diagrama de Actividad: Caso de Uso Eliminar Objetivos

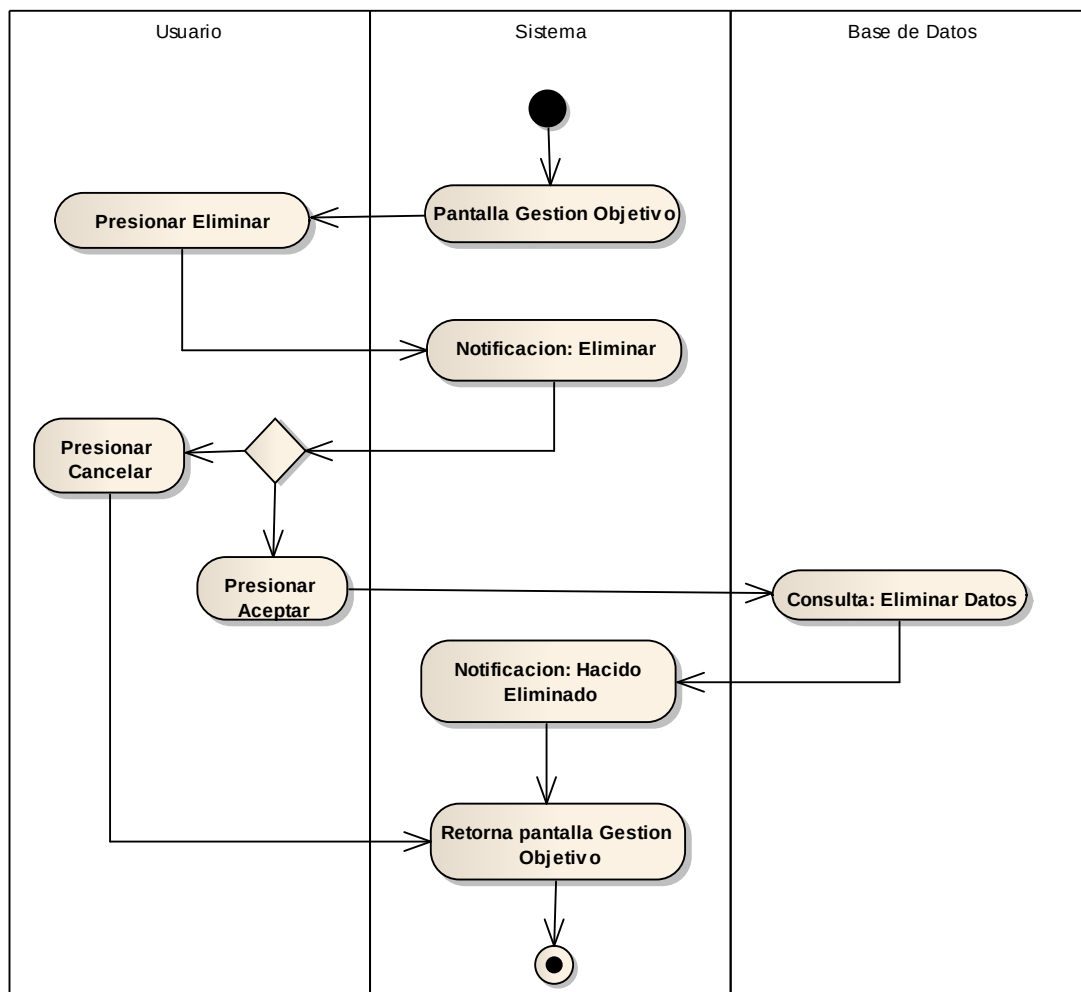


Figura 87 Diagrama de Actividad: Caso de Uso Eliminar Objetivos

II.1.1.3.6.2.4. Diagrama de Actividad: Caso de Uso Modificar Objetivos

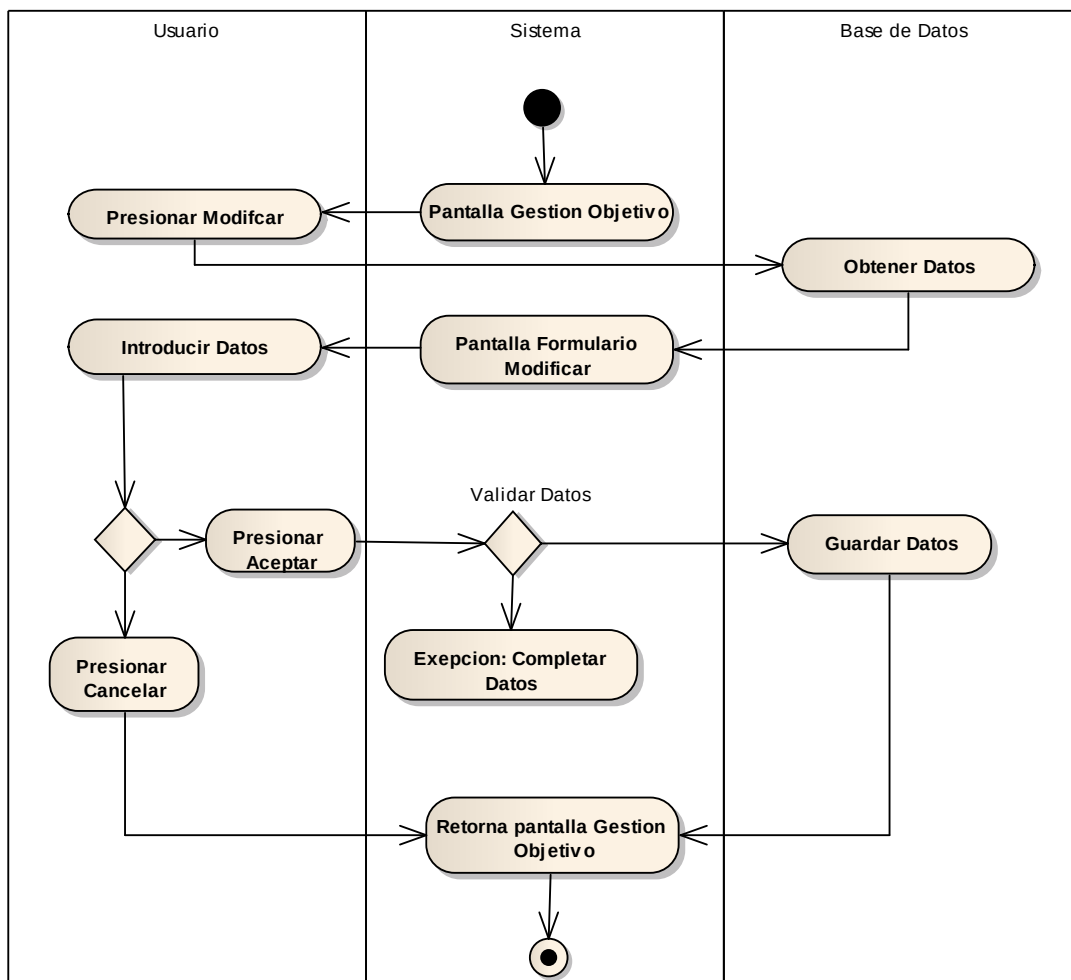


Figura 88 Diagrama de Actividad: Caso de Uso Modificar Objetivos

II.1.1.3.6.2.5. Diagrama de Actividad: Caso de Uso Gestión Metas

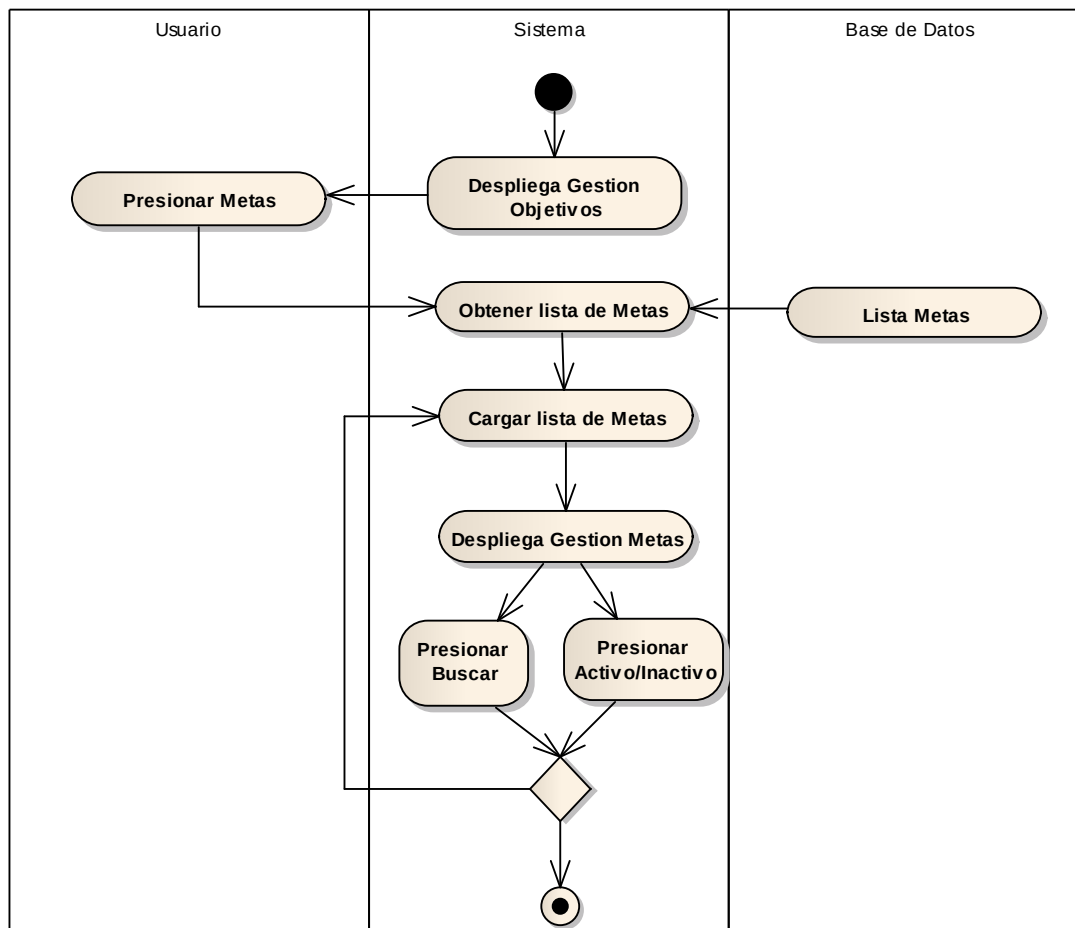


Figura 89 Diagrama de Actividad: Caso de Uso Gestión Metas

II.1.1.3.6.2.6. Diagrama de Actividad: Caso de Uso Adicionar Metas

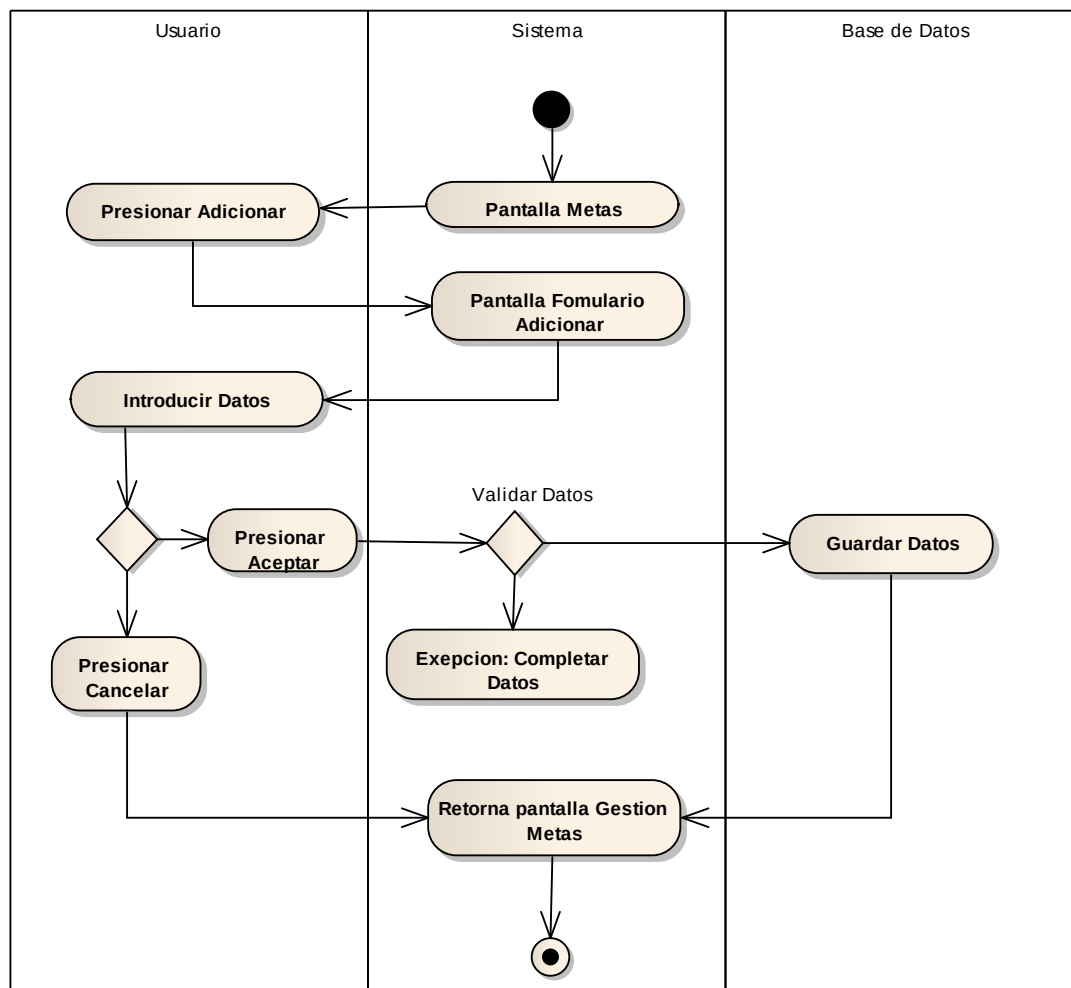


Figura 90 Diagrama de Actividad: Caso de Uso Adicionar Metas

II.1.1.3.6.2.7. Diagrama de Actividad: Caso de Uso Eliminar Metas

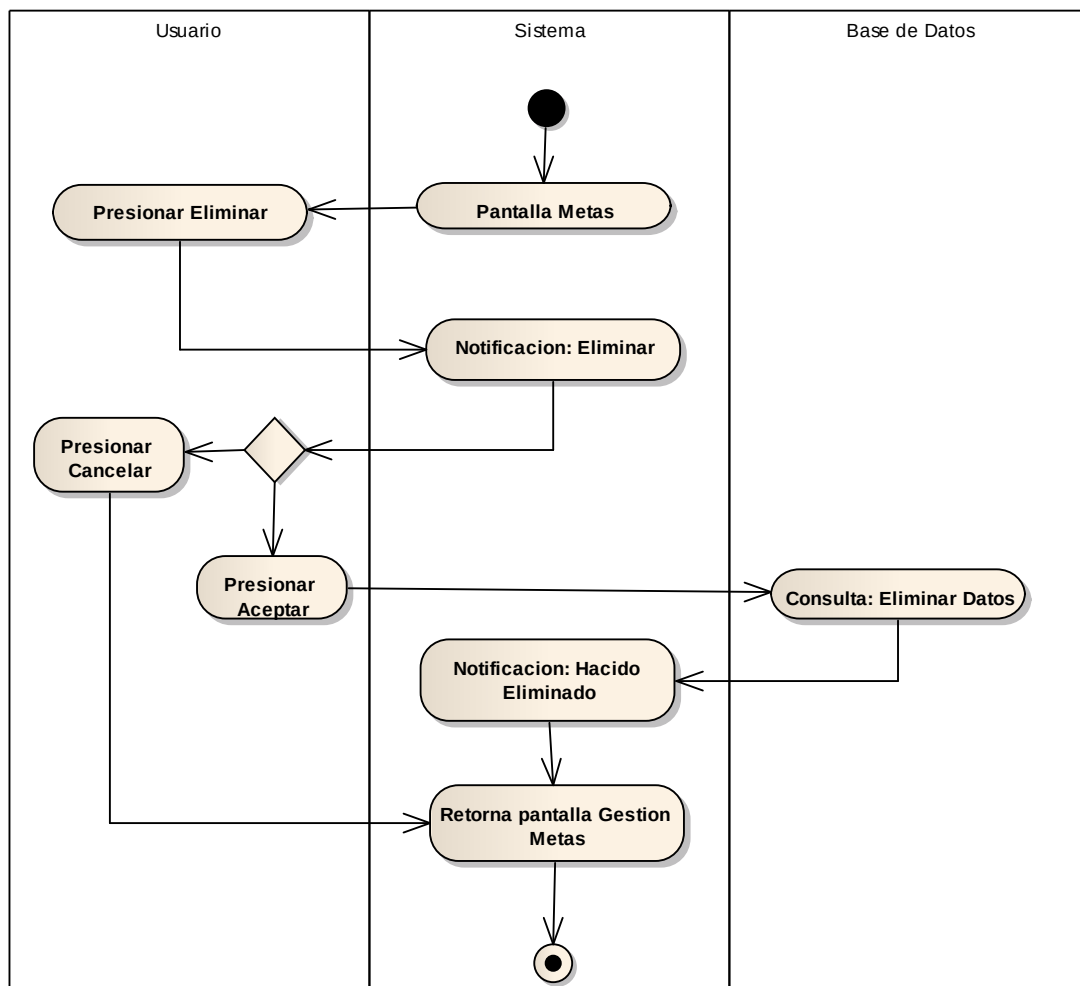


Figura 91 Diagrama de Actividad: Caso de Uso Eliminar Metas

II.1.1.3.6.2.8. Diagrama de Actividad: Caso de Uso Modificar Metas

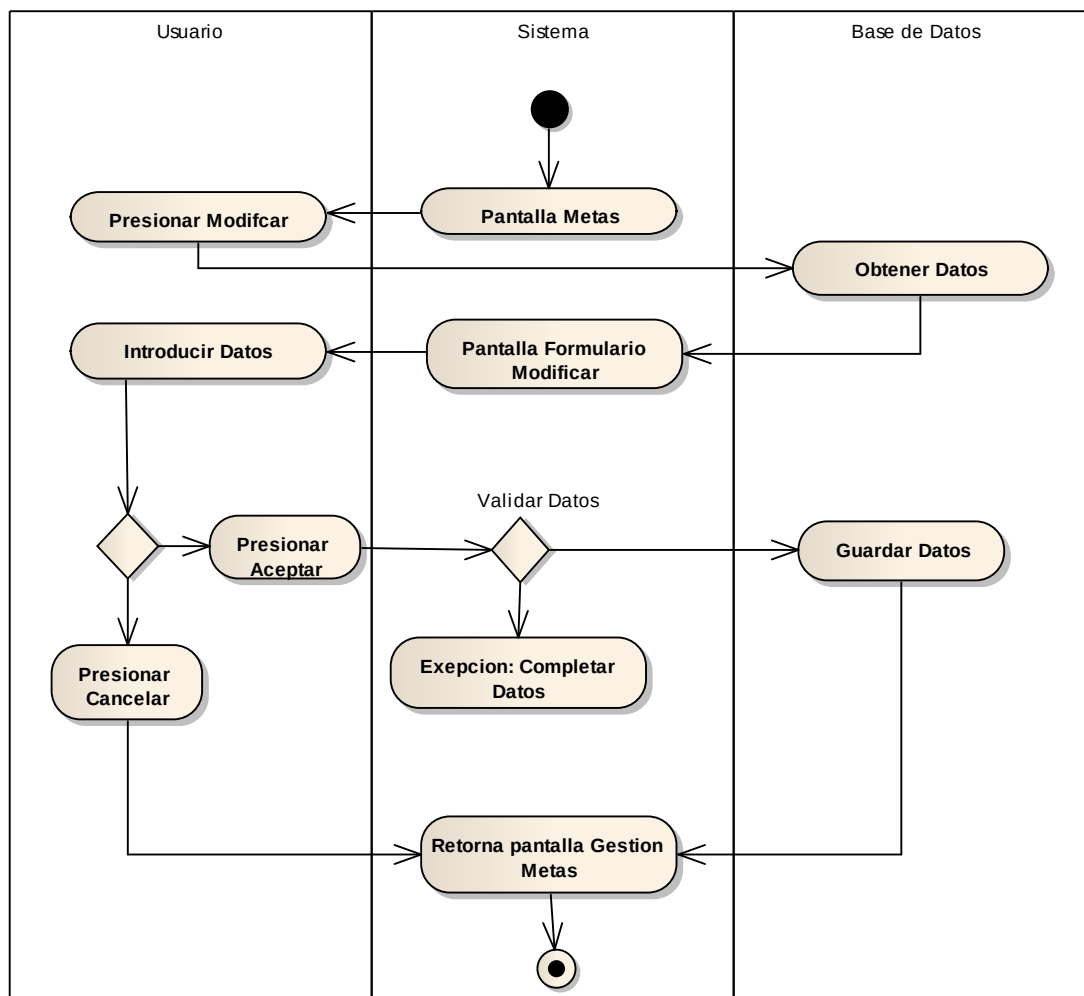


Figura 92 Diagrama de Actividad: Caso de Uso Modificar Metas

II.1.1.3.6.2.9. Diagrama de Actividad: Caso de Uso Gestión Recursos

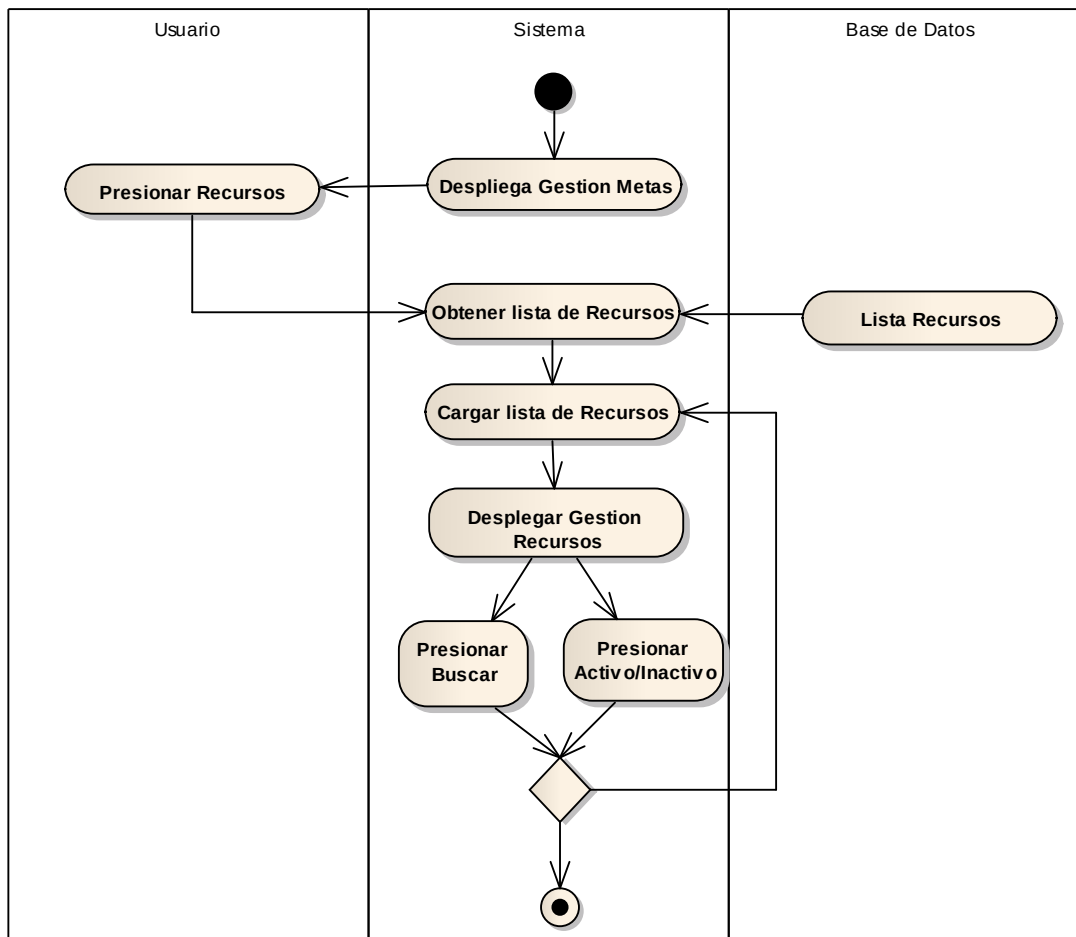


Figura 93 Diagrama de Actividad: Caso de Uso Gestión Recursos

II.1.1.3.6.2.10. Diagrama de Actividad: Caso de Uso Adicionar Recursos

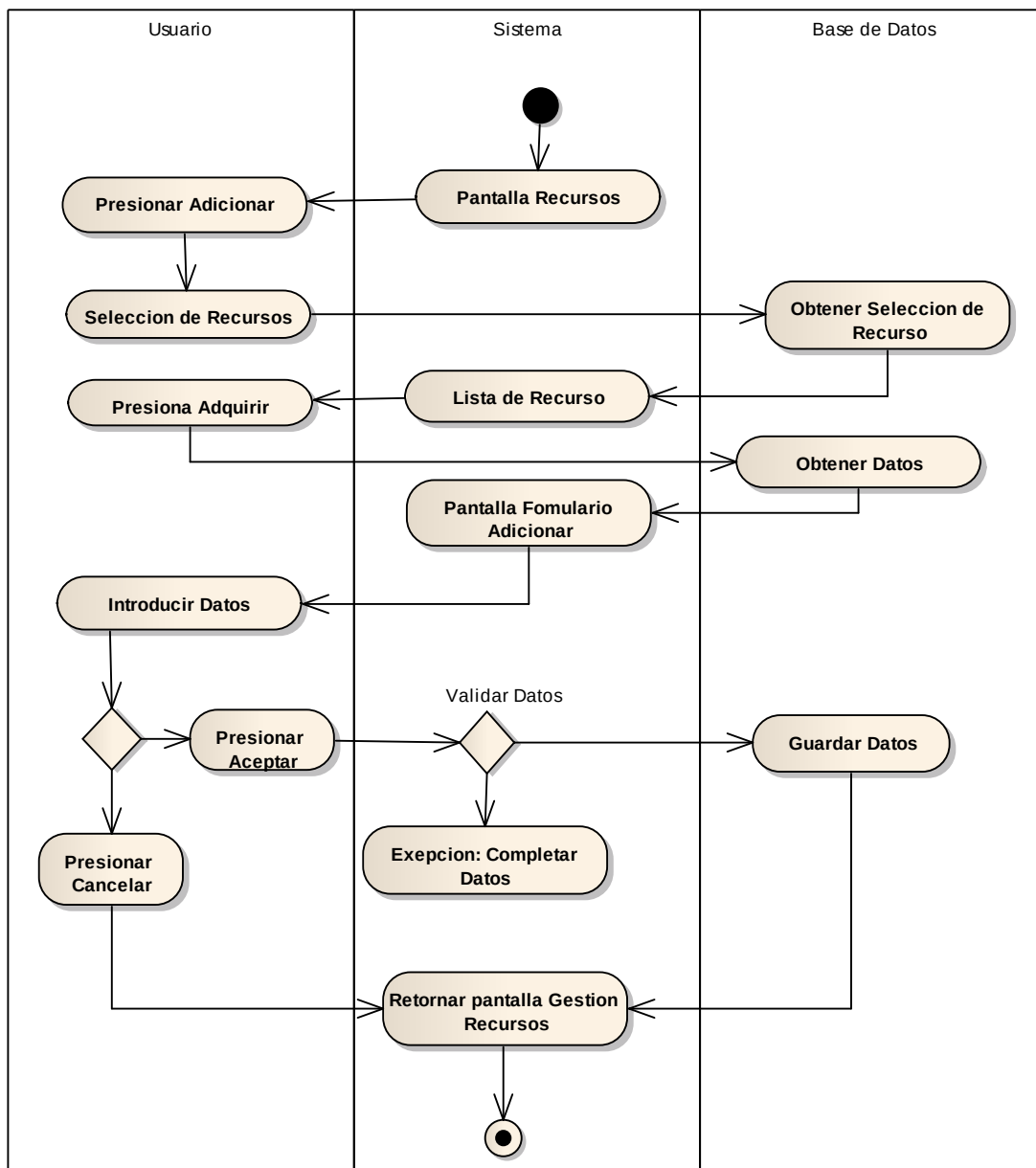


Figura 94 Diagrama de Actividad: Caso de Uso Adicionar Recursos

II.1.1.3.6.2.11. Diagrama de Actividad: Caso de Uso Eliminar Recursos

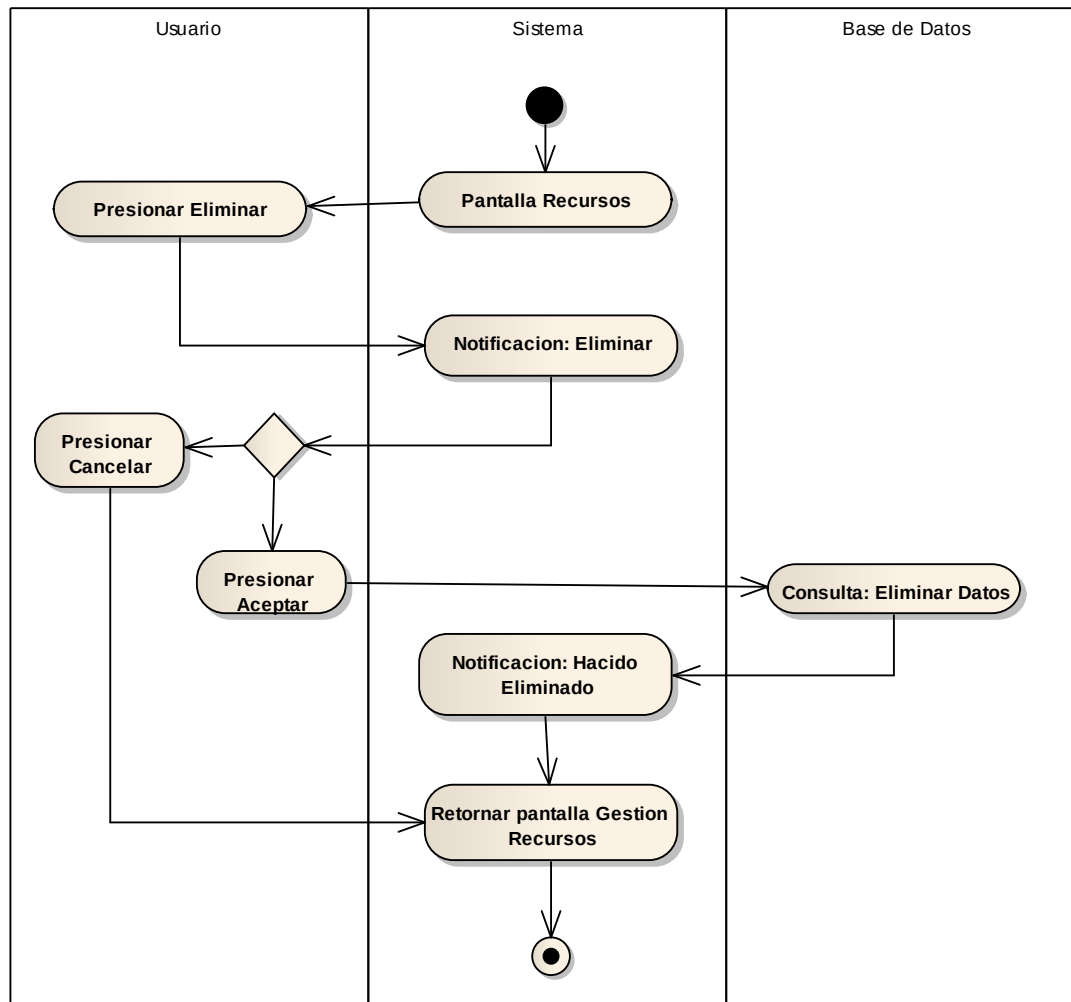


Figura 95 Diagrama de Actividad: Caso de Uso Eliminar Recursos

II.1.1.3.6.2.12. Diagrama de Actividad: Caso de Uso Modificar Recursos

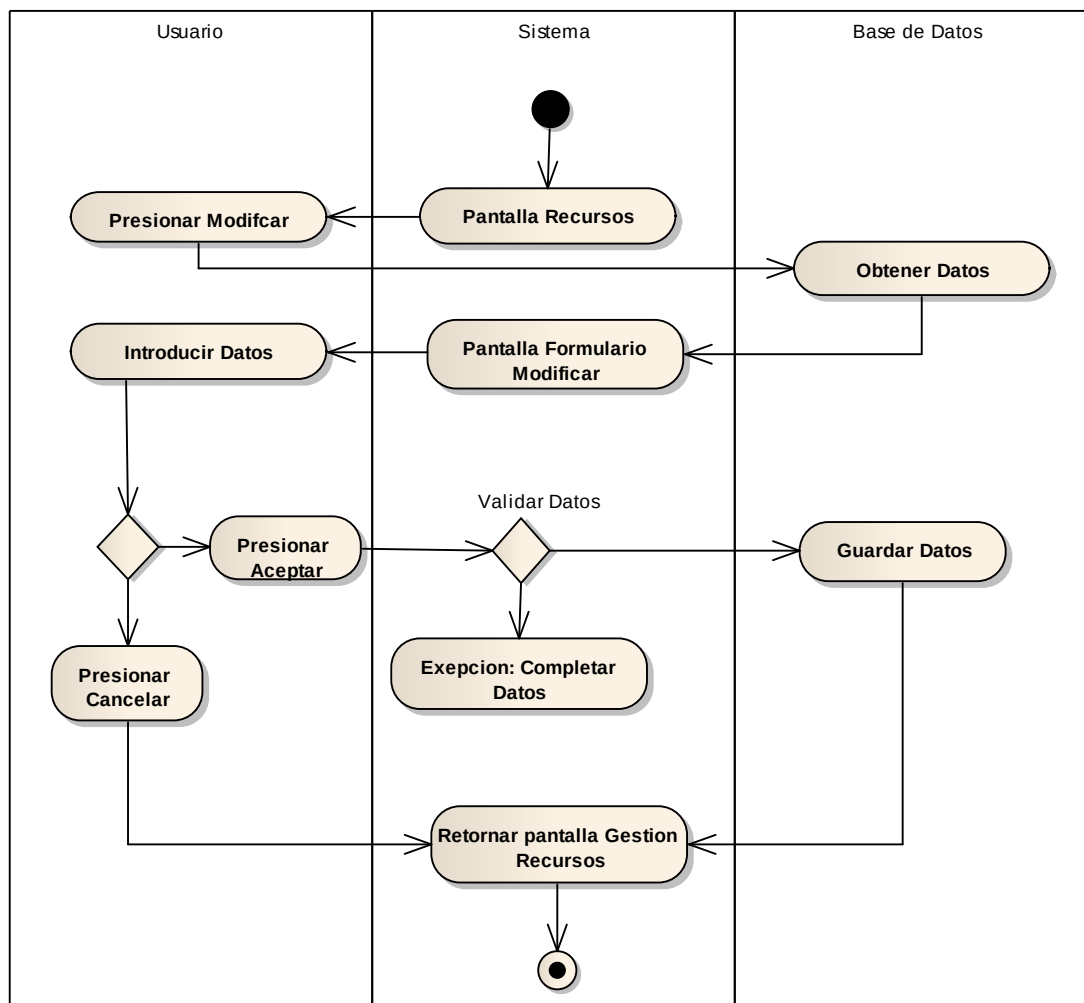
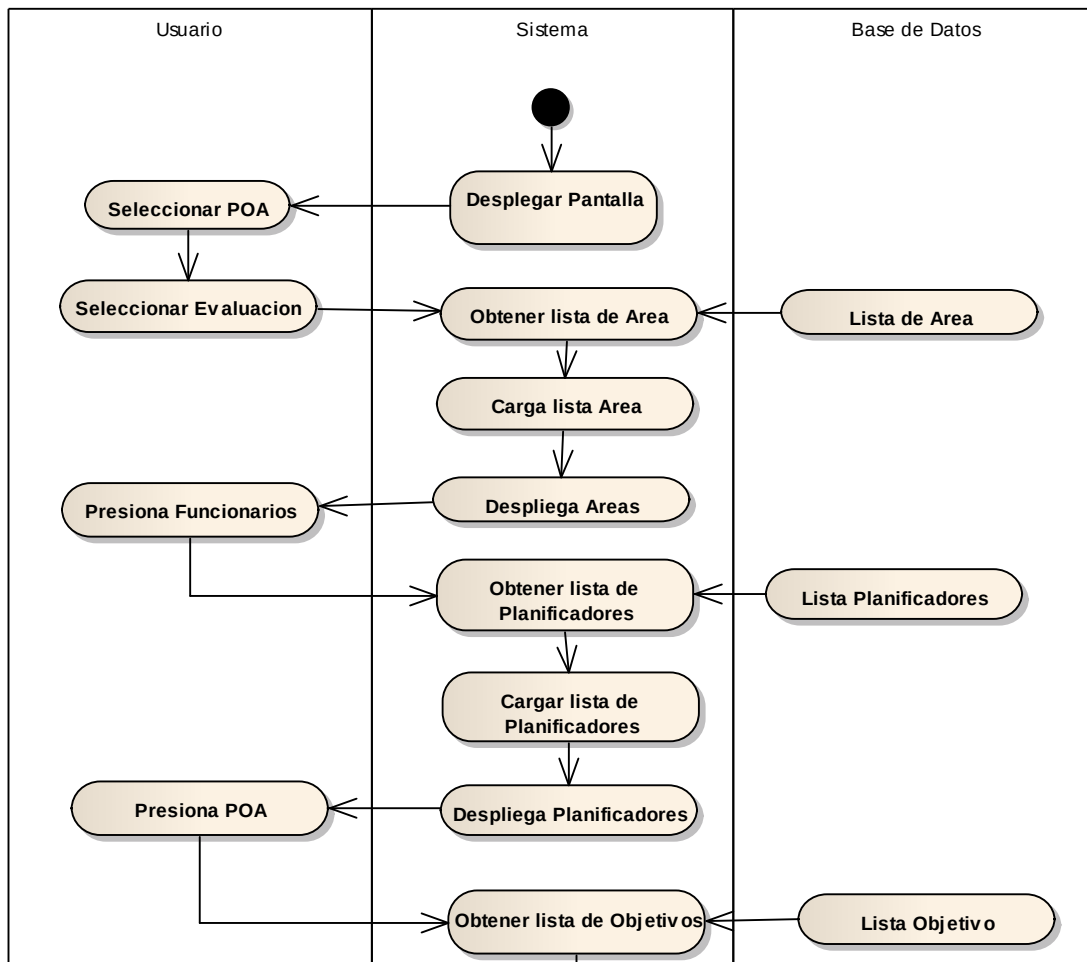


Figura 96 Diagrama de Actividad: Caso de Uso Modificar Recursos

II.1.1.3.6.3. Diagrama de Actividad: Caso de Uso Evaluación Plan Operativo Anual



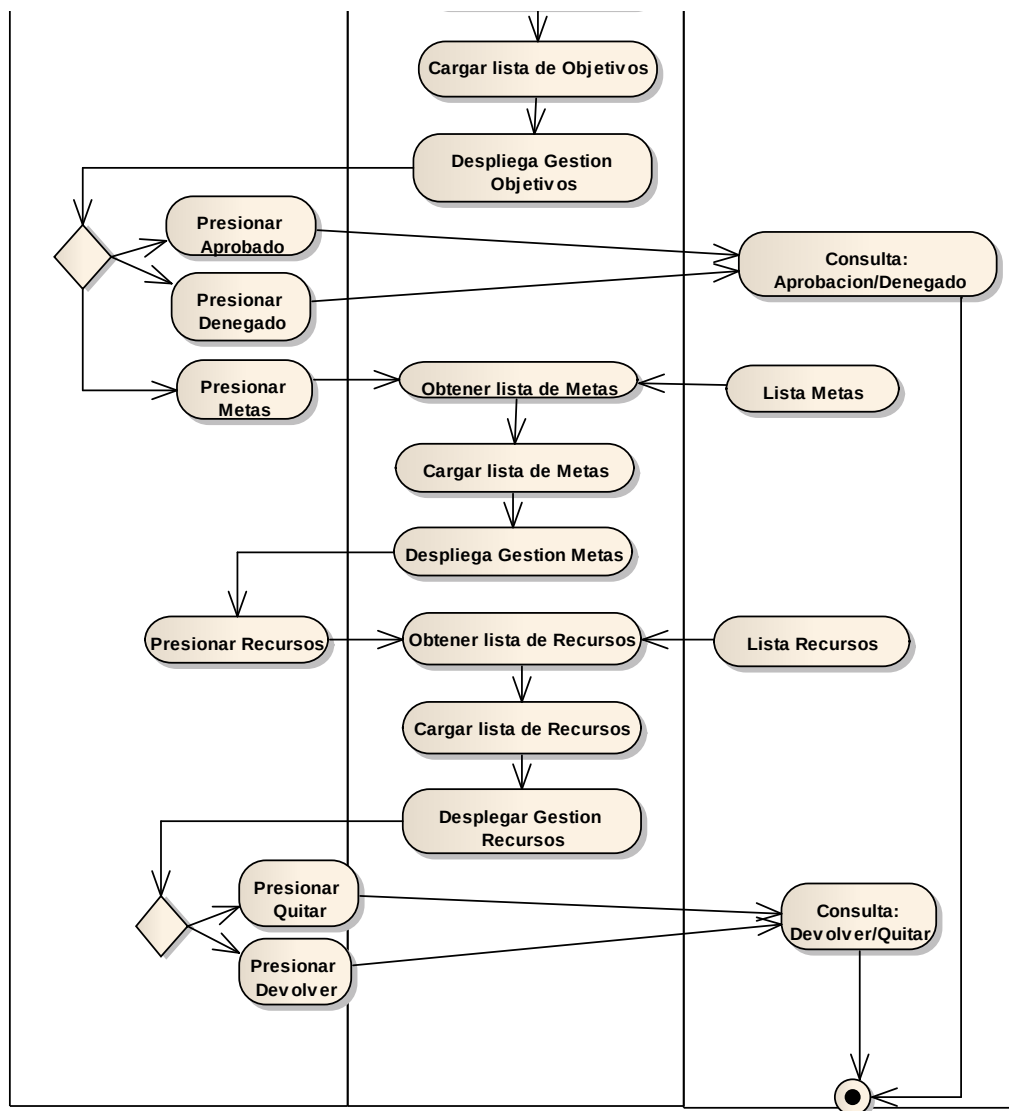


Figura 97 Diagrama de Actividad: Caso de Uso Evaluación Plan Operativo Anual

II.1.1.3.6.4. Diagrama de Actividad: Caso de Uso Techo Presupuestario

II.1.1.3.6.4.1. Diagrama de Actividad: Caso de Uso Gestión Techo Presupuestario

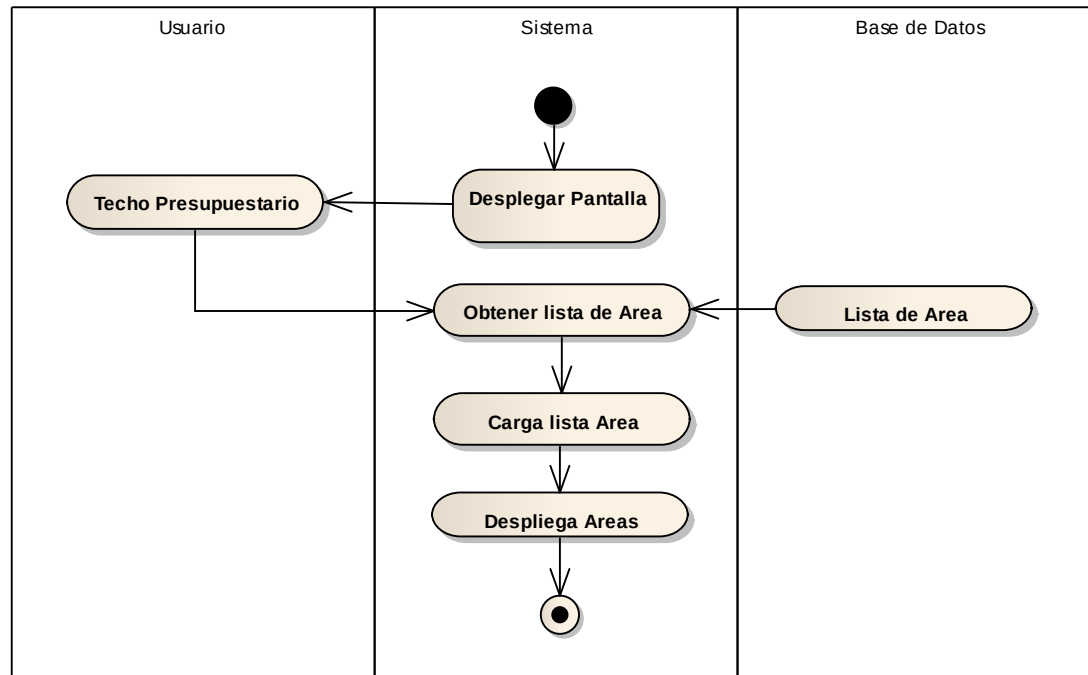


Figura 98 Diagrama de Actividad: Caso de Uso Gestión Techo Presupuestario

II.1.1.3.6.4.2. Diagrama de Actividad: Caso de Uso Adicionar Techo Presupuestario

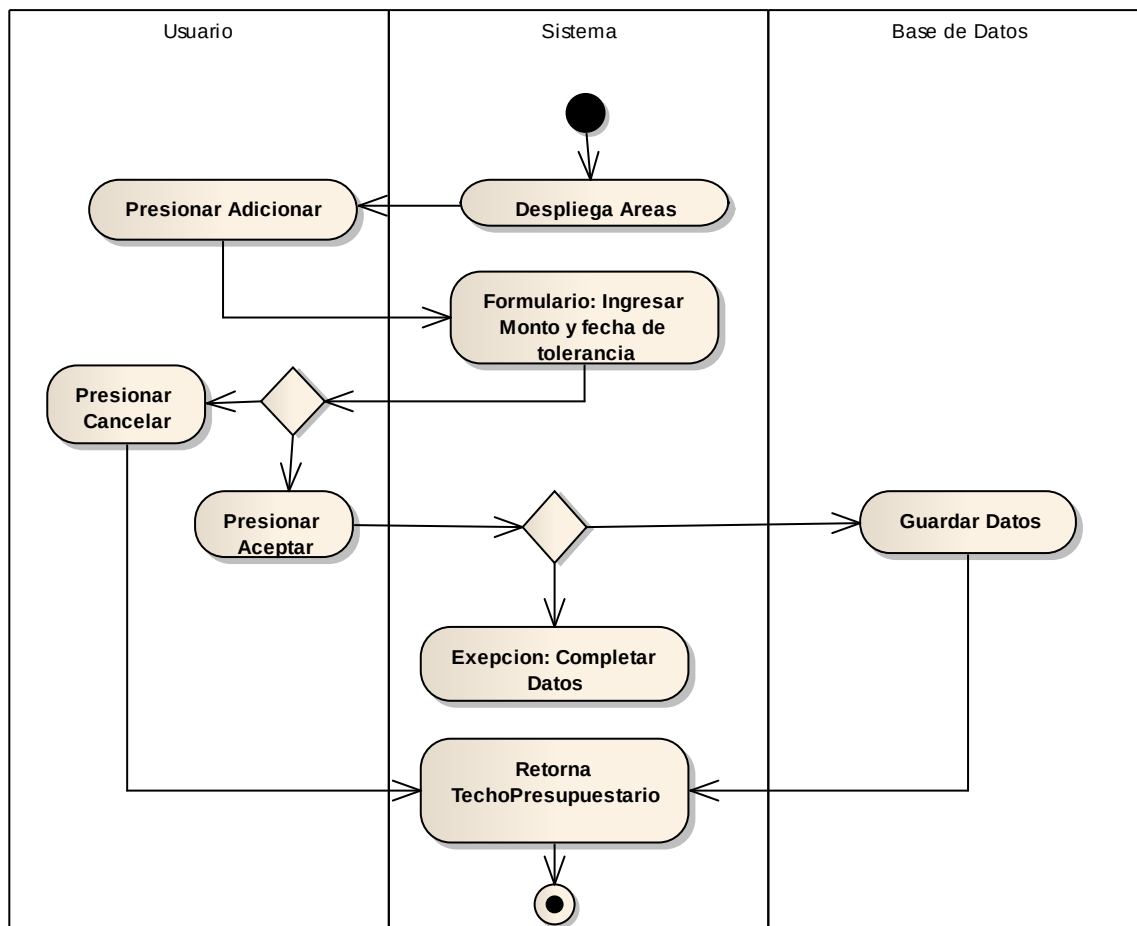


Figura 99 Diagrama de Actividad: Caso de Uso Adicionar Techo Presupuestario

II.1.1.3.6.4.3. Diagrama de Actividad: Caso de Uso Aprobar Techo Presupuestario

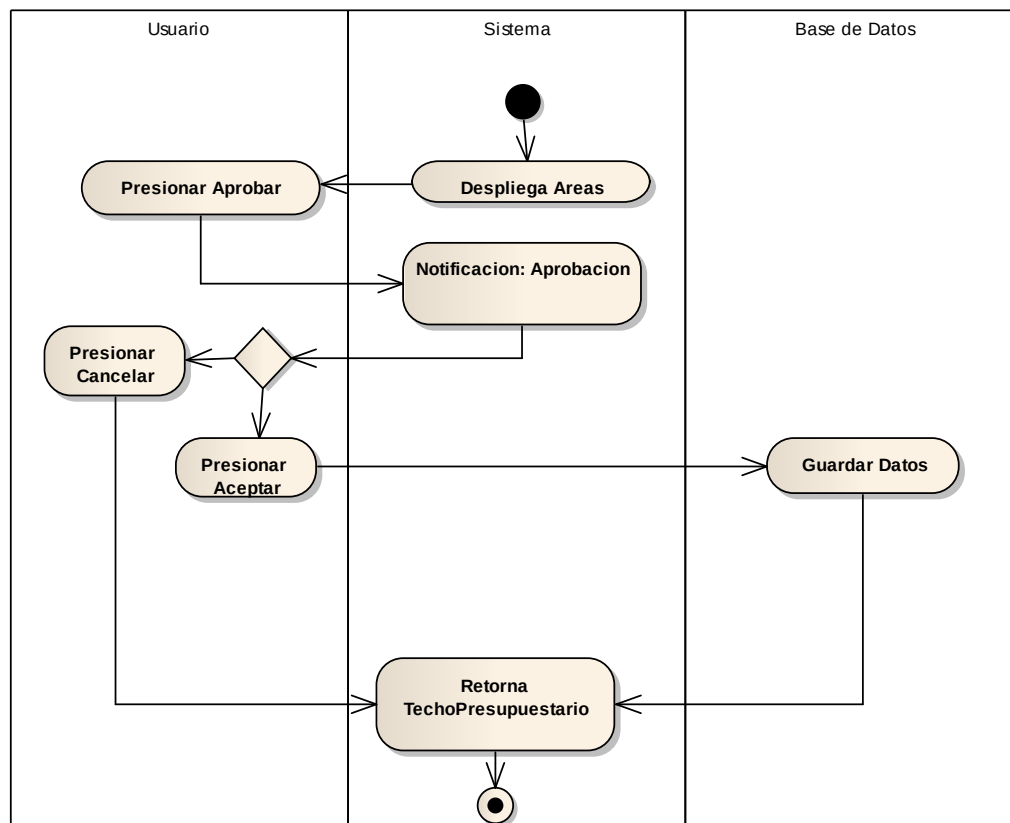


Figura 100 Diagrama de Actividad: Caso de Uso Aprobar Techo Presupuestario

II.1.1.3.6.4.4. Diagrama de Actividad: Caso de Uso Reporte Techo Presupuestario

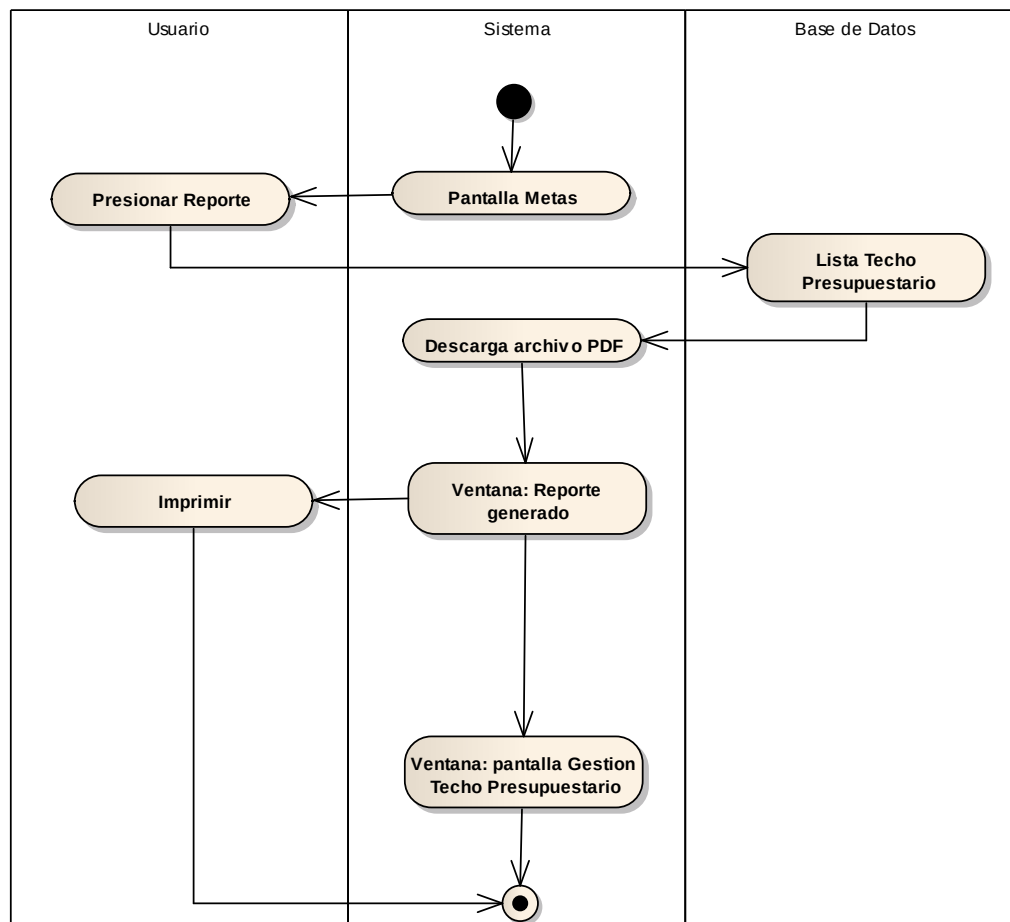


Figura 101 Diagrama de Actividad: Caso de Uso Reporte Techo Presupuestario

II.1.1.3.7. Diagrama de Secuencia

II.1.1.3.7.1. Diagrama de Secuencia: Caso de uso Ingresar al Sistema

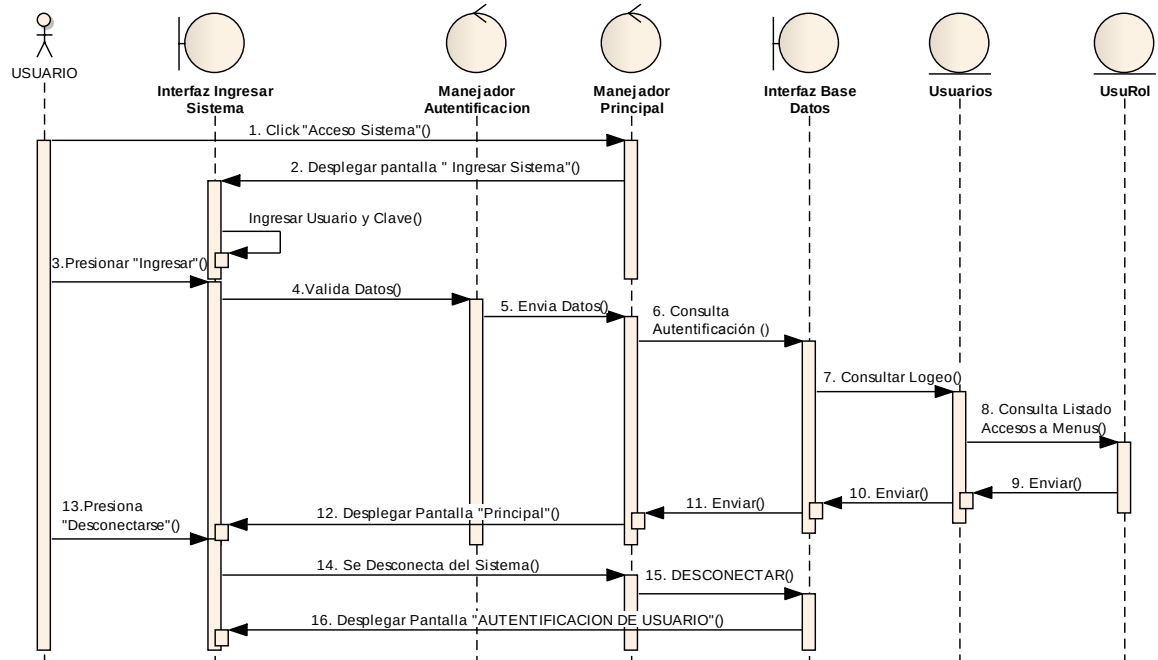


Figura 102 Diagrama de Secuencia: Caso de uso Ingresar al Sistema

II.1.1.3.7.2. Diagrama de Secuencia: casos de uso Formulación del Plan Operativo Anual (POA)

II.1.1.3.7.2.1. Diagrama de Secuencia: Caso de uso gestión de Objetivo

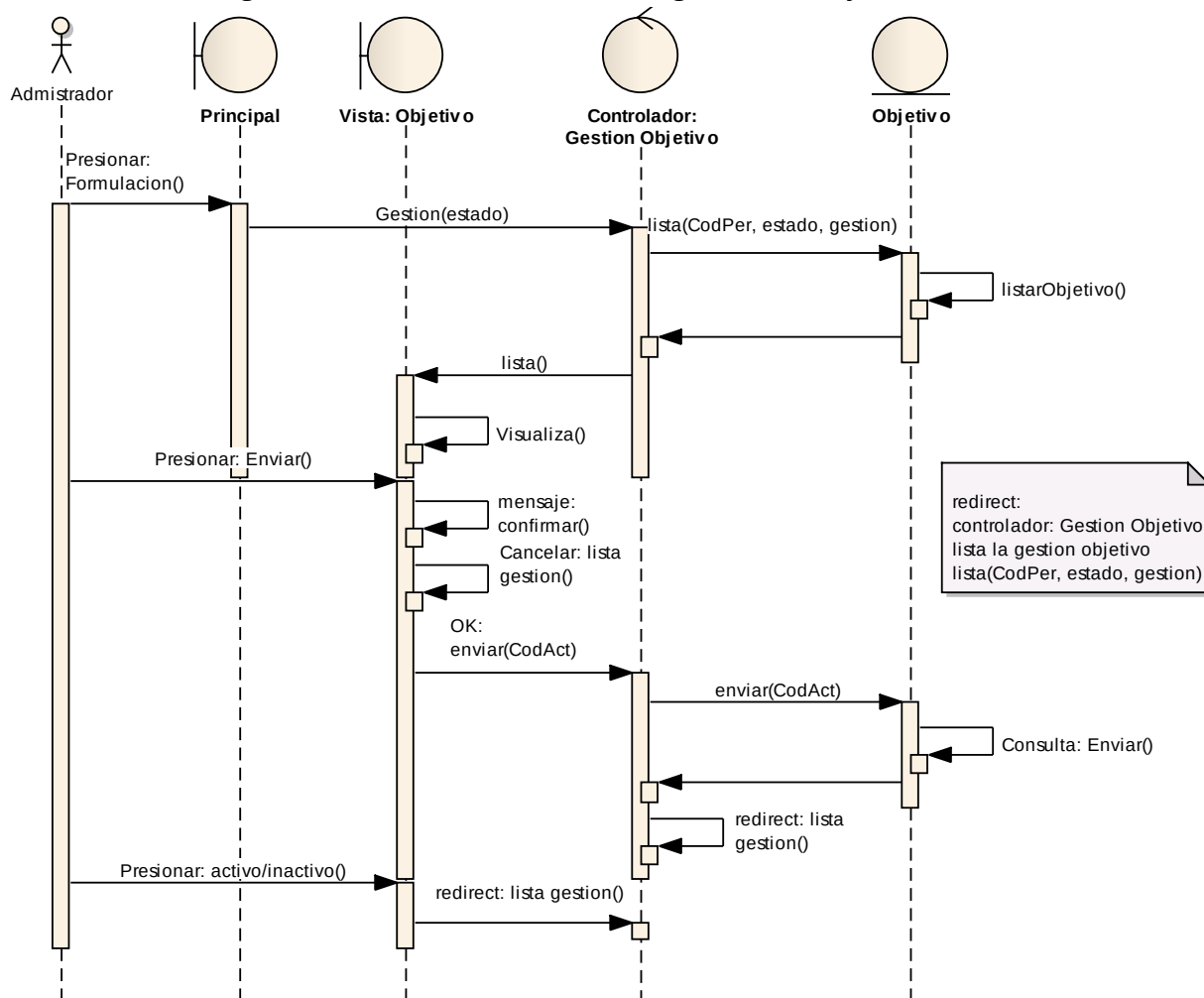


Figura 103 Diagrama de Secuencia: Caso de uso gestión de Objetivo

II.1.1.3.7.2.3. Diagrama de Secuencia: Caso de uso eliminar Objetivo

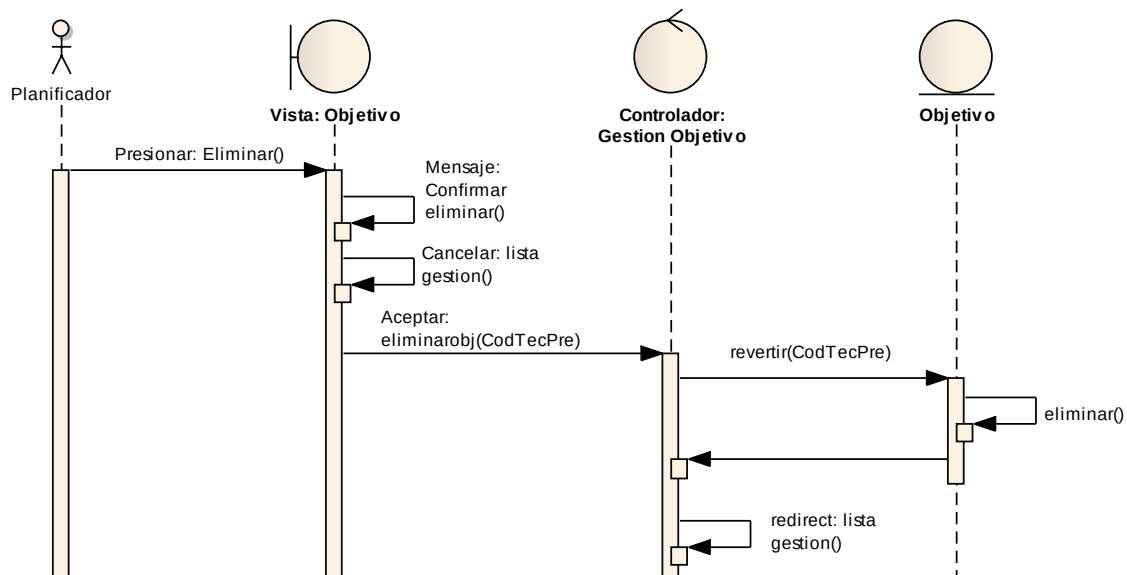


Figura 105 Diagrama de Secuencia: Caso de uso eliminar Objetivo

II.1.1.3.7.2.4. Diagrama de Secuencia: Caso de uso modificar Objetivo

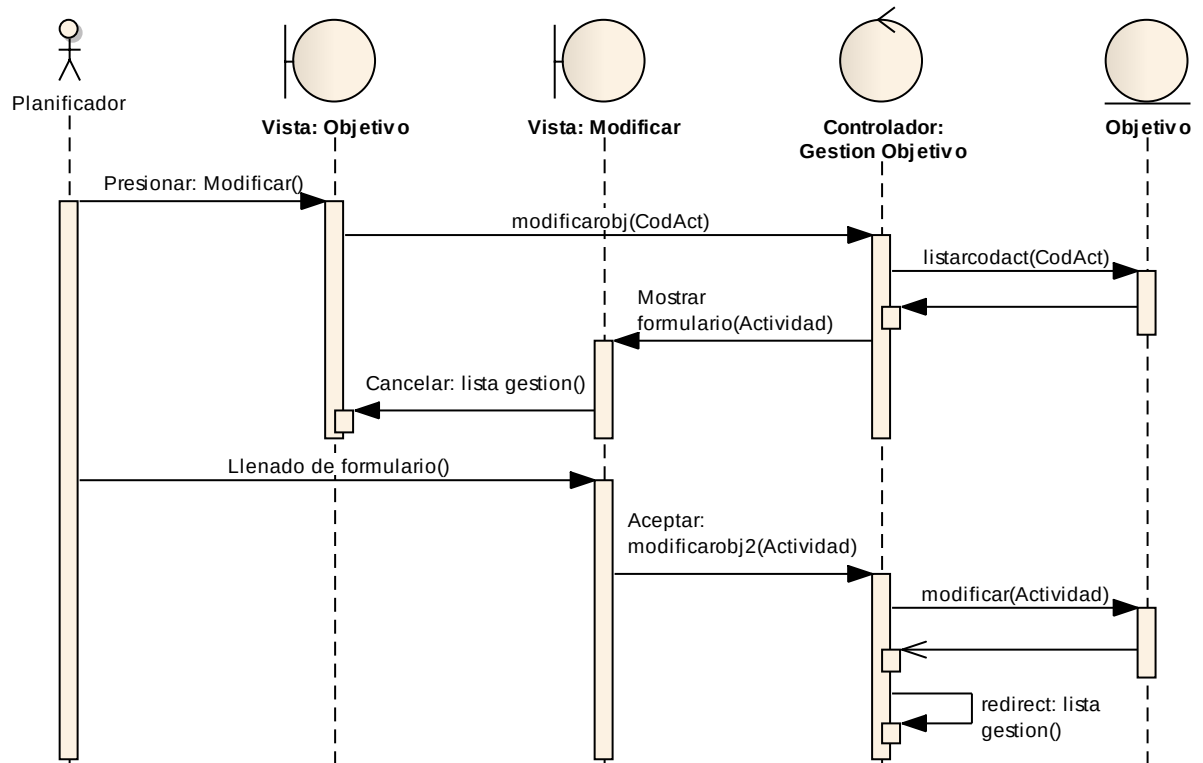


Figura 106 Diagrama de Secuencia: Caso de uso modificar Objetivo

II.1.1.3.7.2.5. Diagrama de Secuencia: Caso de uso gestión Metas

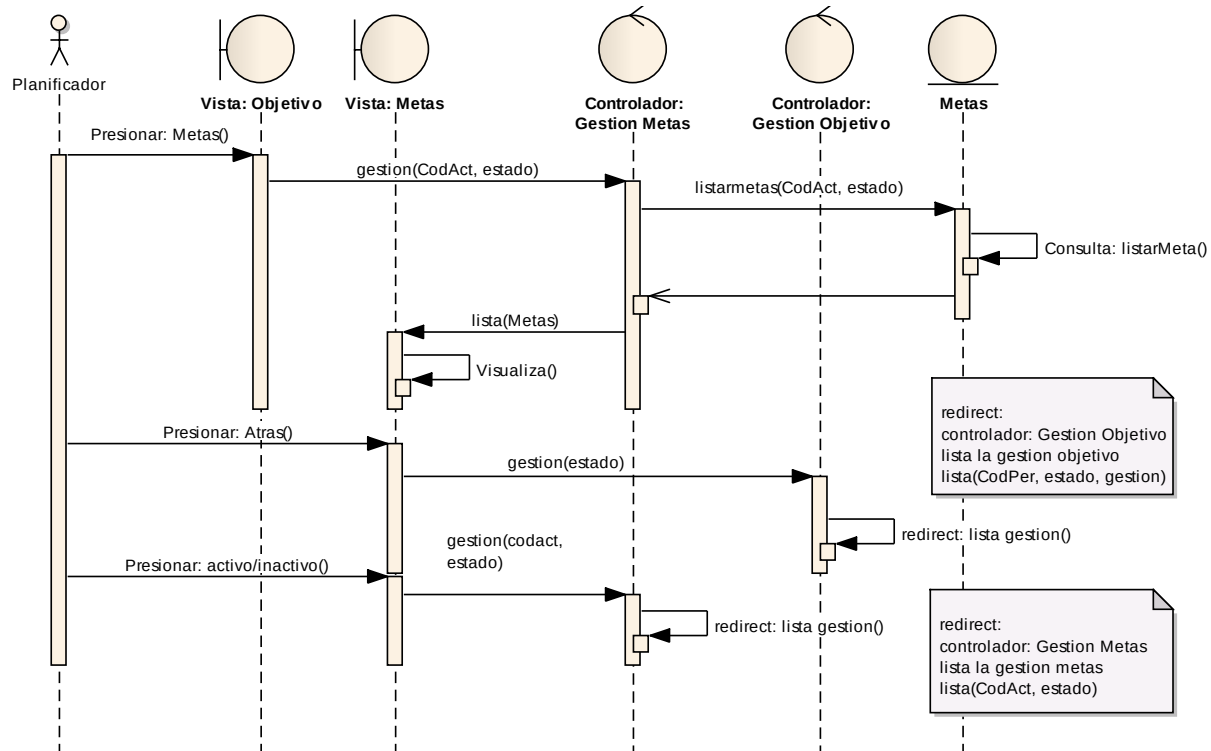


Figura 107 Diagrama de Secuencia: Caso de uso gestión Metas

II.1.1.3.7.2.6. Diagrama de Secuencia: Caso de uso adicionar Metas

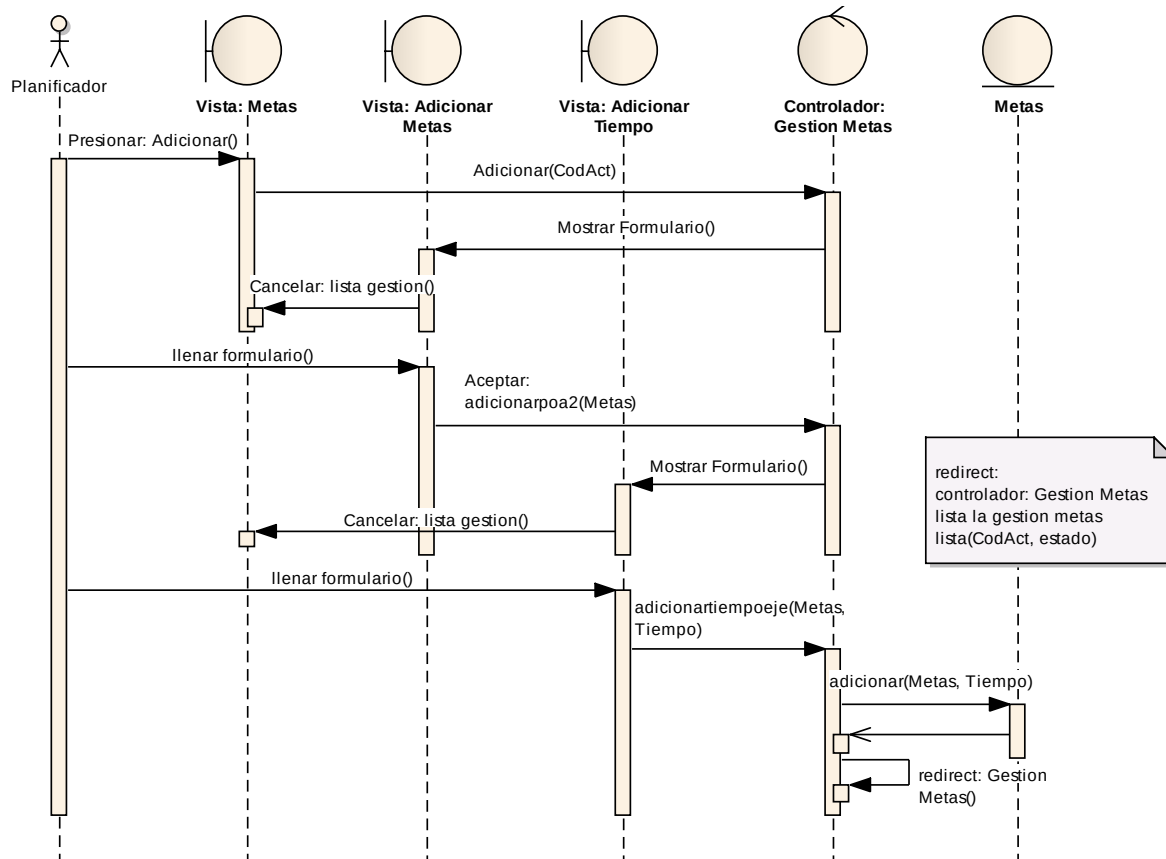


Figura 108 Diagrama de Secuencia: Caso de uso adicionar Metas

II.1.1.3.7.2.7. Diagrama de Secuencia: Caso de uso eliminar Metas

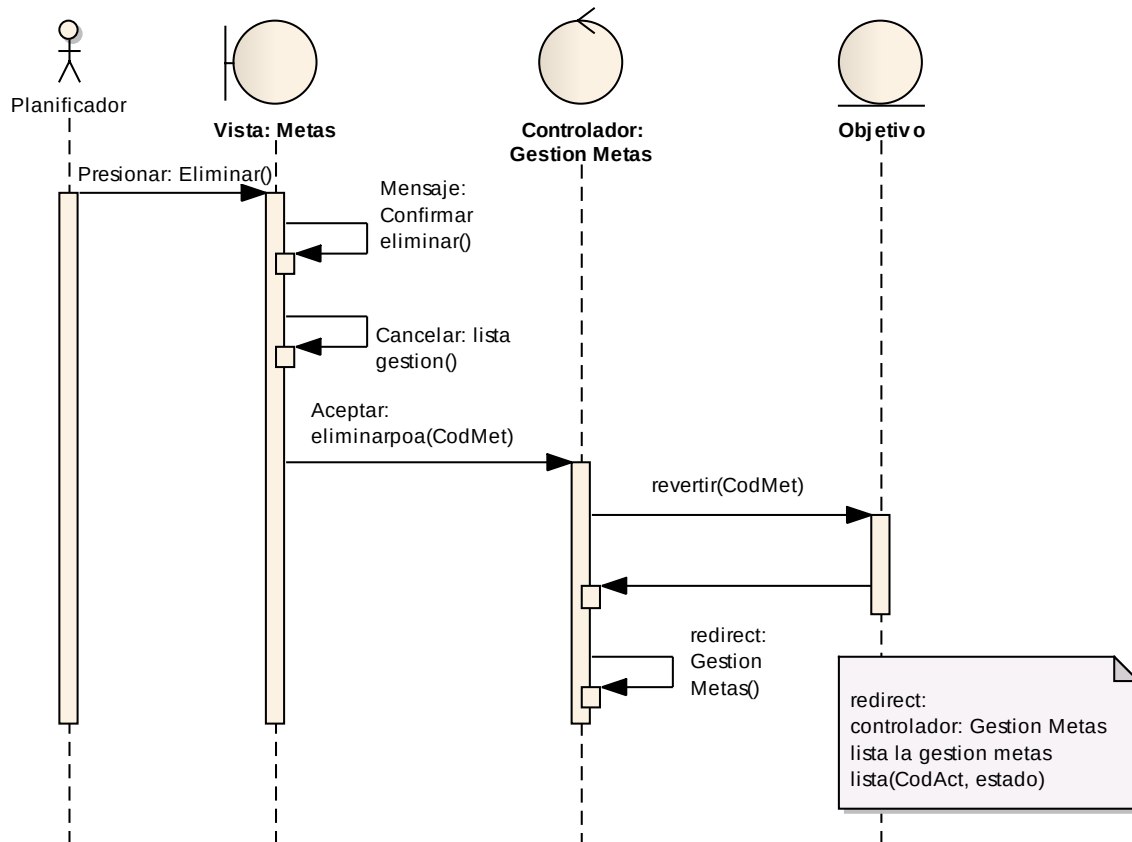


Figura 109 Diagrama de Secuencia: Caso de uso eliminar Metas

II.1.1.3.7.2.8. Diagrama de Secuencia: Caso de uso modificar Metas

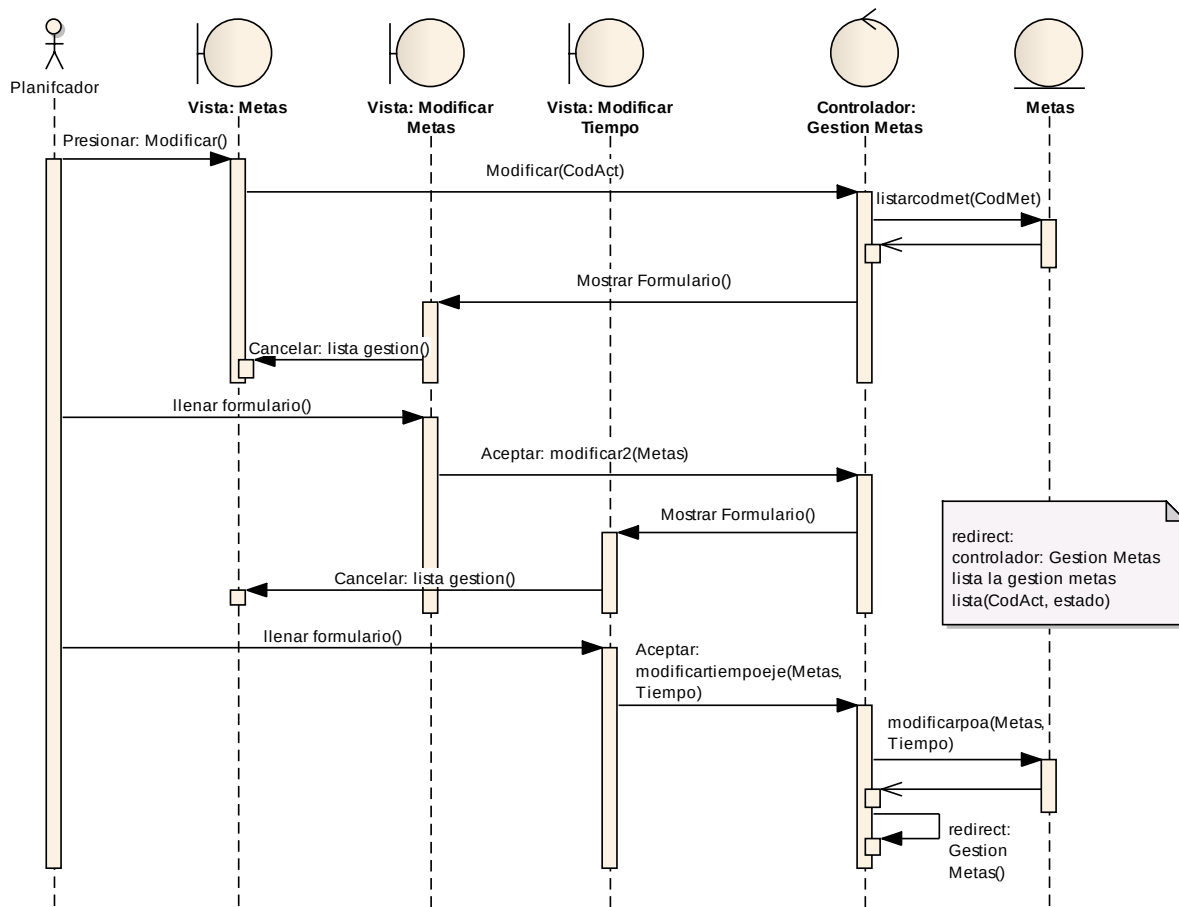


Figura 110 Diagrama de Secuencia: Caso de uso modificar Metas

II.1.1.3.7.2.9. Diagrama de Secuencia: Caso de uso gestión recursos

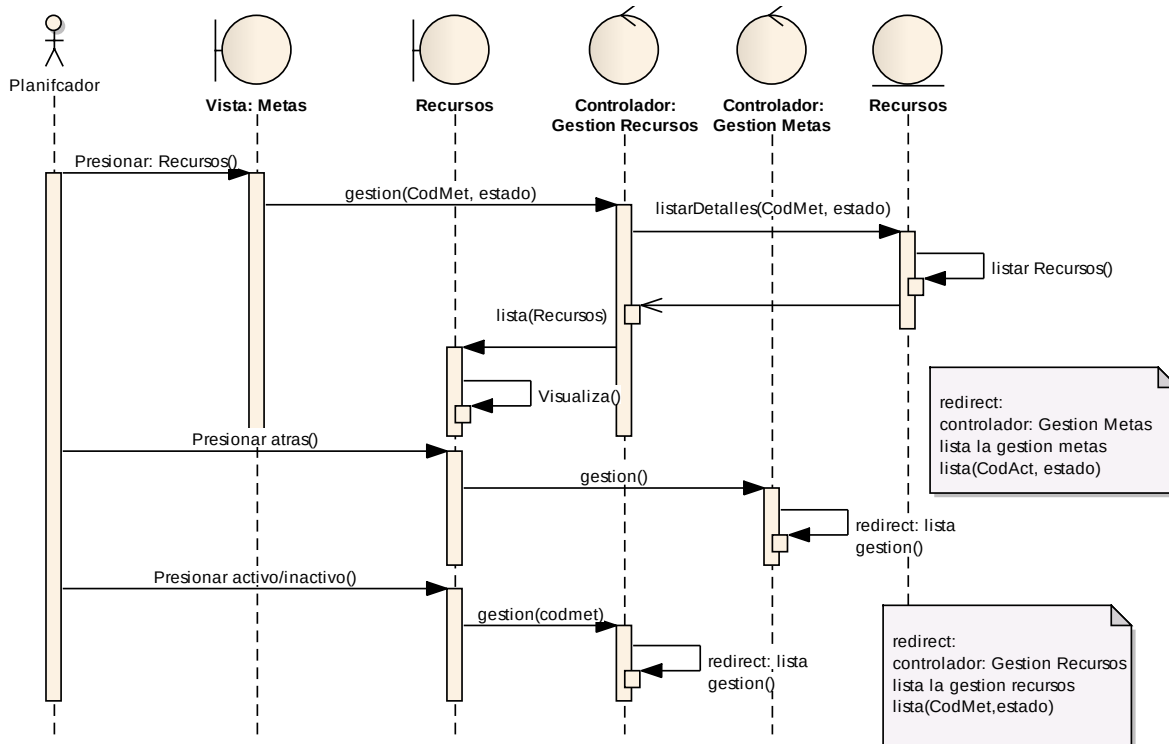


Figura 111 Diagrama de Secuencia: Caso de uso gestión recursos

II.1.1.3.7.2.10. Diagrama de Secuencia: Caso de uso adicionar recursos

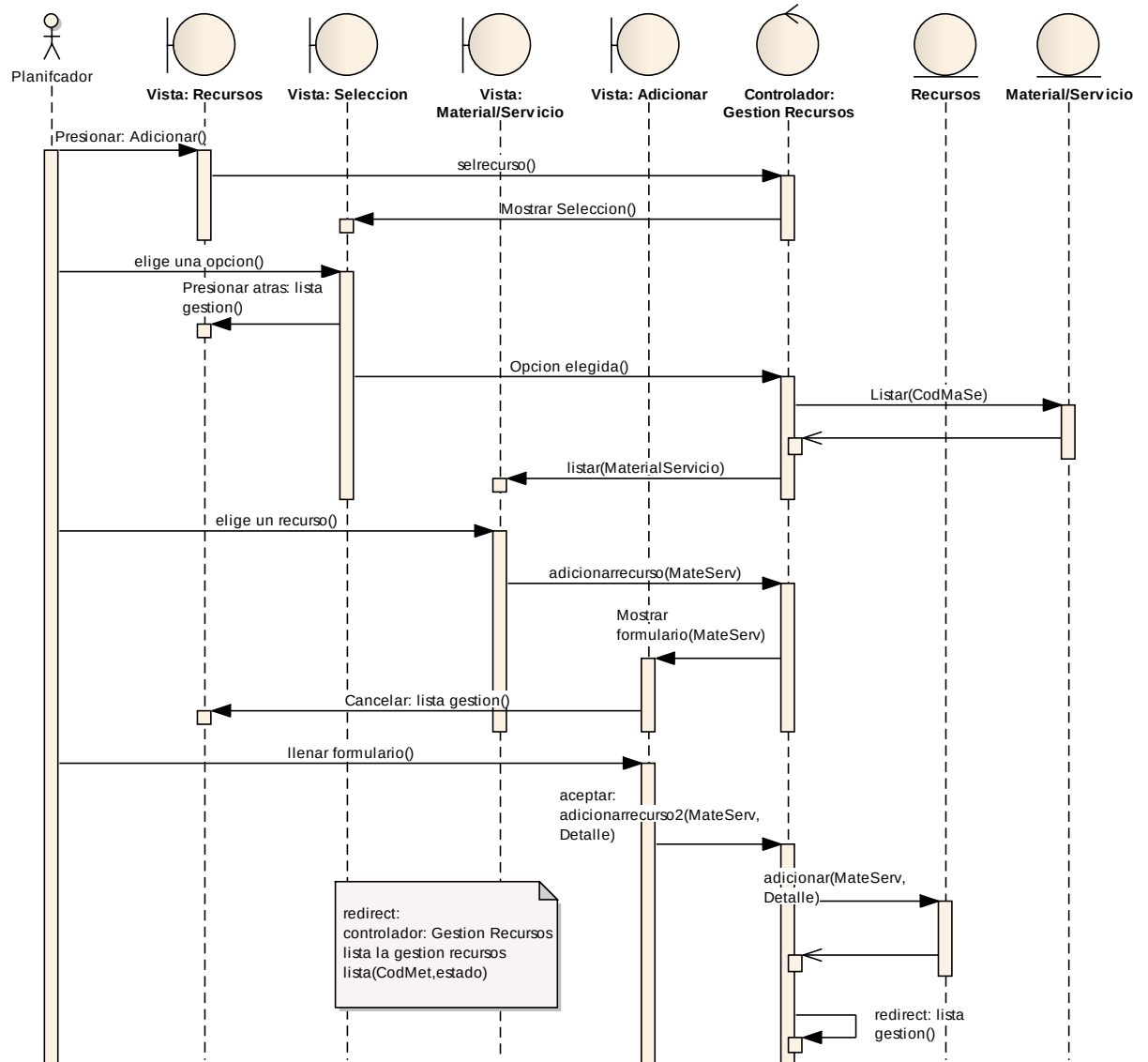


Figura 112 Diagrama de Secuencia: Caso de uso adicionar recursos

II.1.1.3.7.2.11.

Diagrama de Secuencia: Caso de uso eliminar recursos

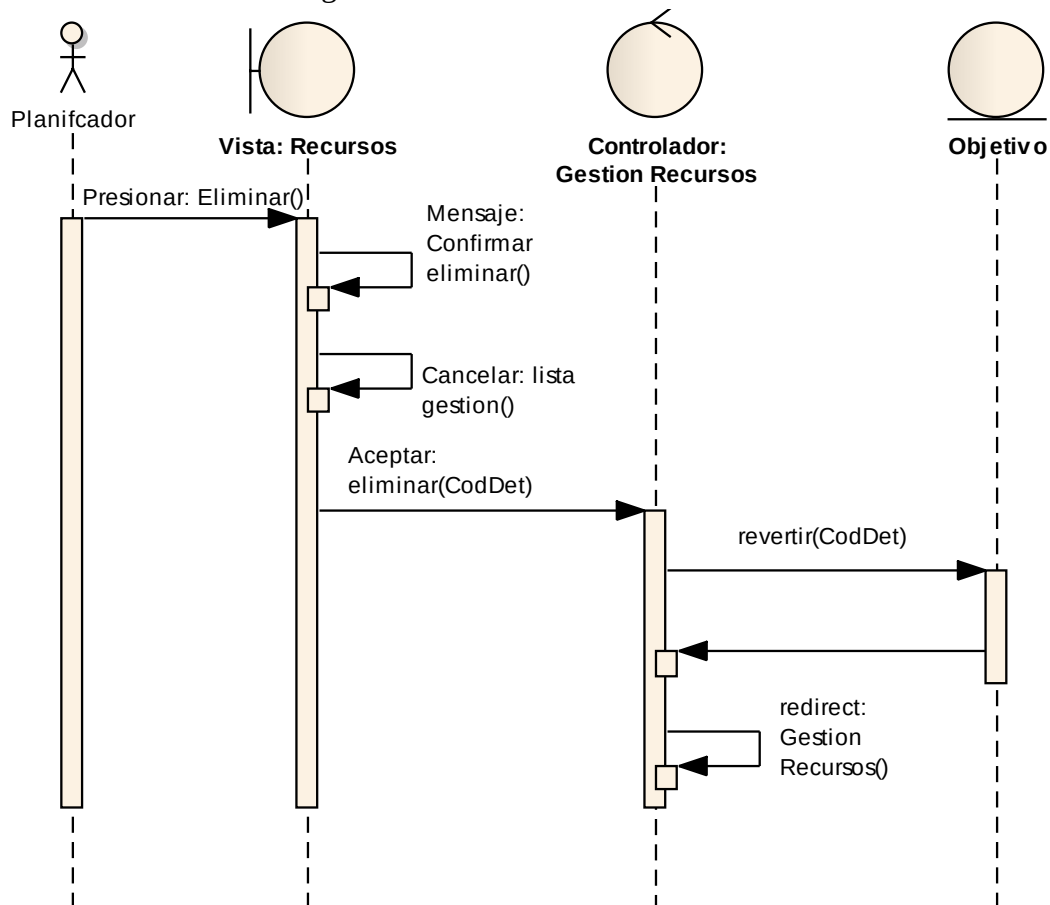


Figura 113 Diagrama de Secuencia: Caso de uso eliminar recursos

II.1.1.3.7.2.12. Diagrama de Secuencia: Caso de uso modificar recursos

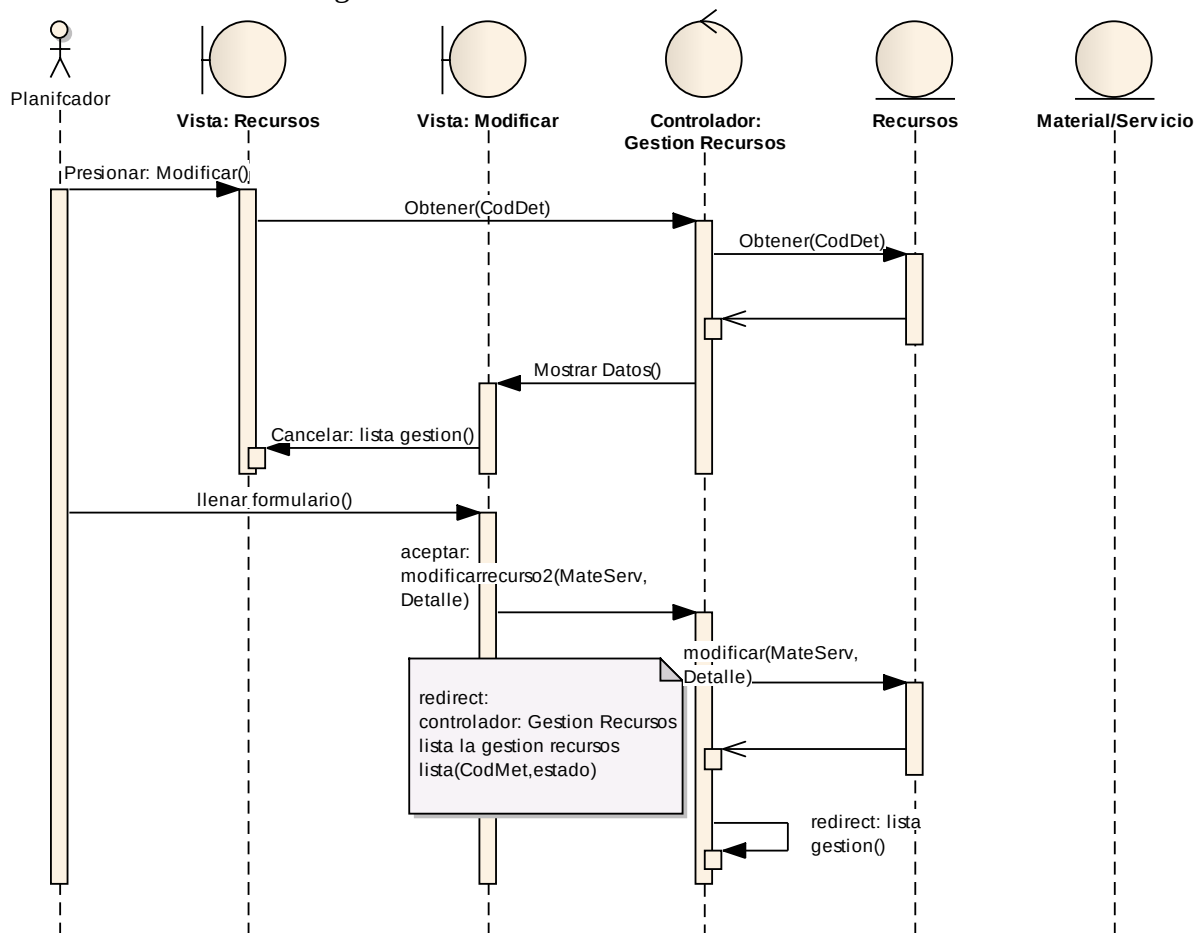
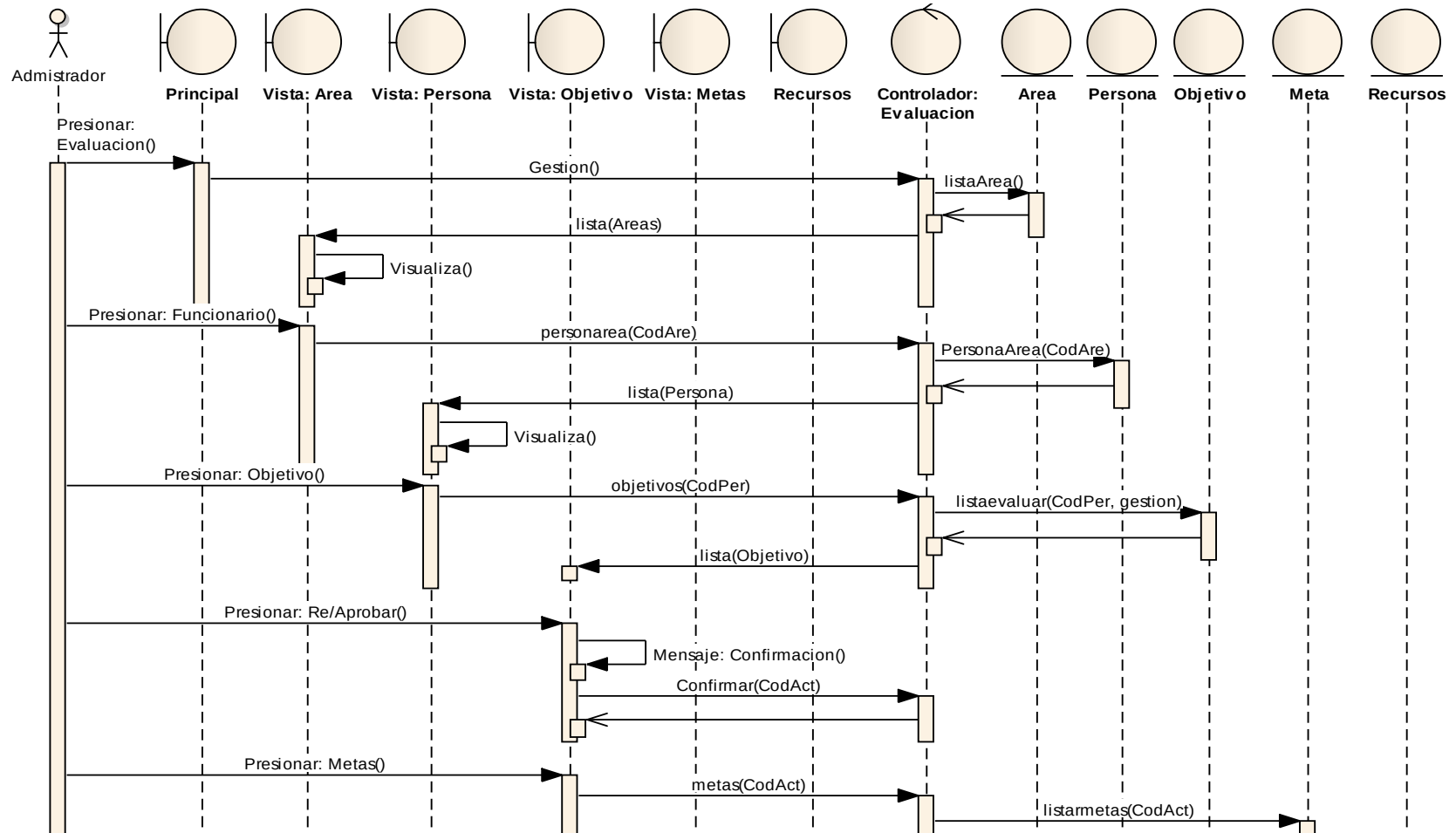


Figura 114 Diagrama de Secuencia: Caso de uso modificar recursos

II.1.1.3.7.3. Diagrama de Secuencia: casos de uso Evaluacion del Plan Operativo Anual (POA)



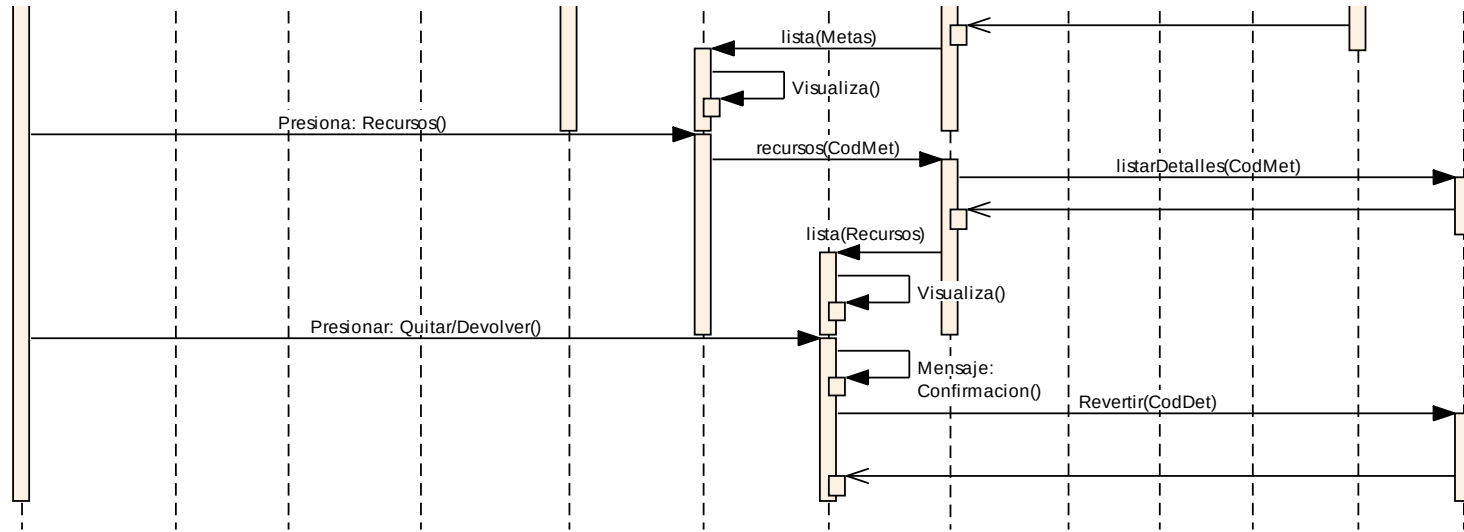


Figura 115 Diagrama de Secuencia: casos de uso Evaluacion del Plan Operativo Anual (POA)

II.1.1.3.7.4. Diagrama de Secuencia: casos de uso Techo Presupuestario

II.1.1.3.7.4.1. Diagrama de Secuencia: casos de uso Determinacion Techo Presupuestario

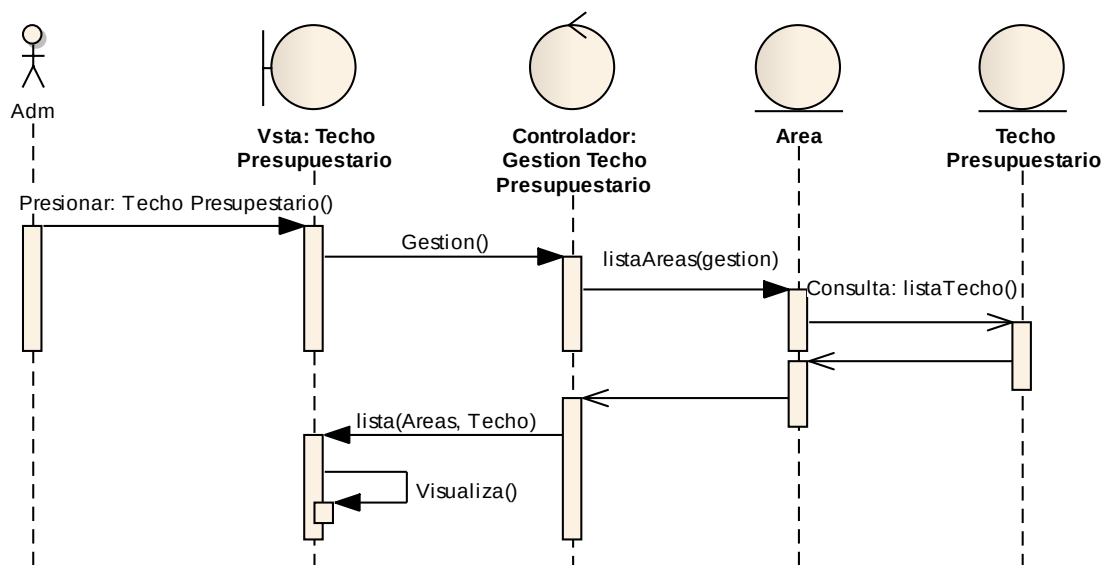


Figura 116 Diagrama de Secuencia: casos de uso Determinacion Techo Presupuestario

II.1.1.3.7.4.2. Diagrama de Secuencia: casos de uso adicionar Presupuesto

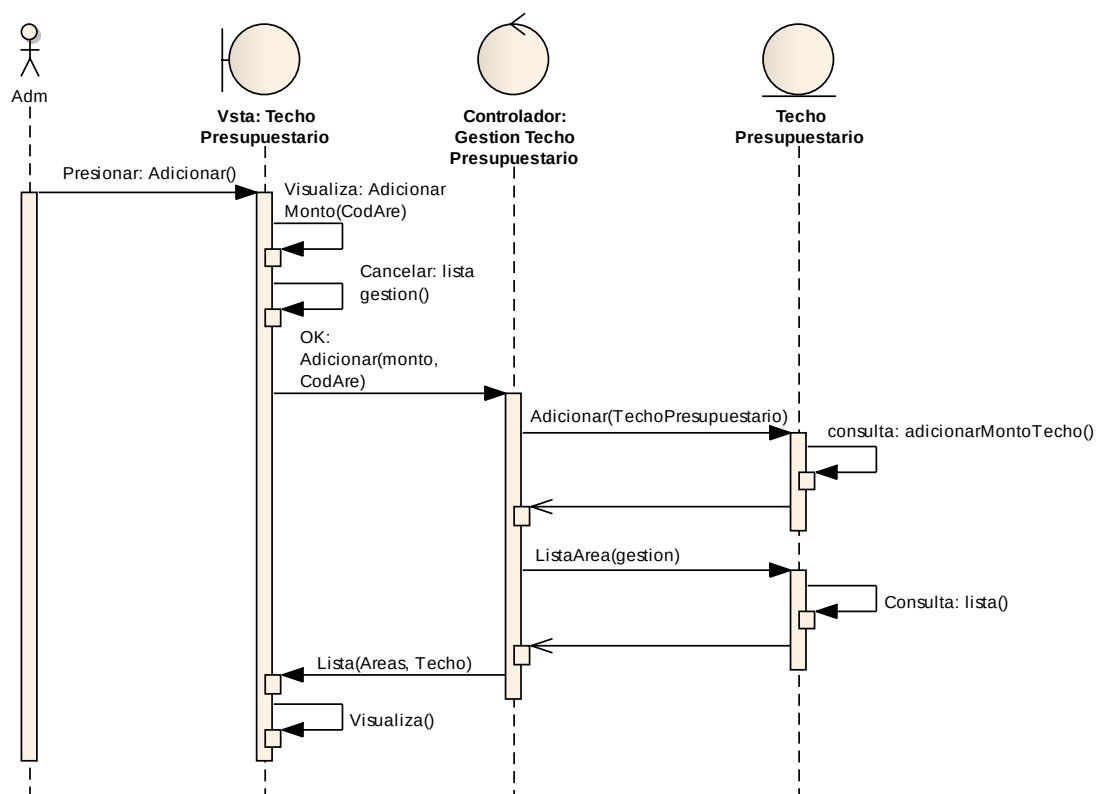


Figura 117 Diagrama de Secuencia: casos de uso adicionar Presupuesto

II.1.1.3.7.4.3. Diagrama de Secuencia: casos de uso Aprobar Presupuesto

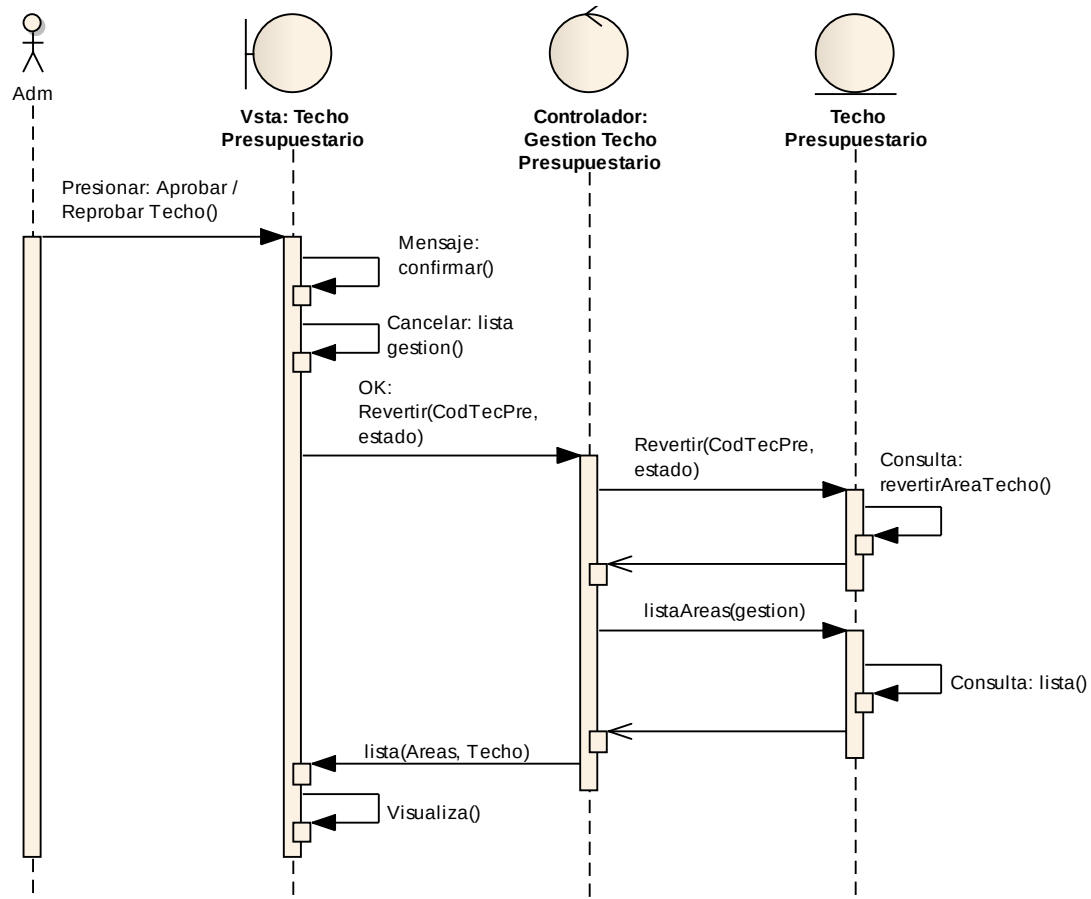


Figura 118 Diagrama de Secuencia: casos de uso Aprobar Presupuesto

II.1.1.3.7.4.4. Diagrama de Secuencia: casos de uso Reporte Presupuesto

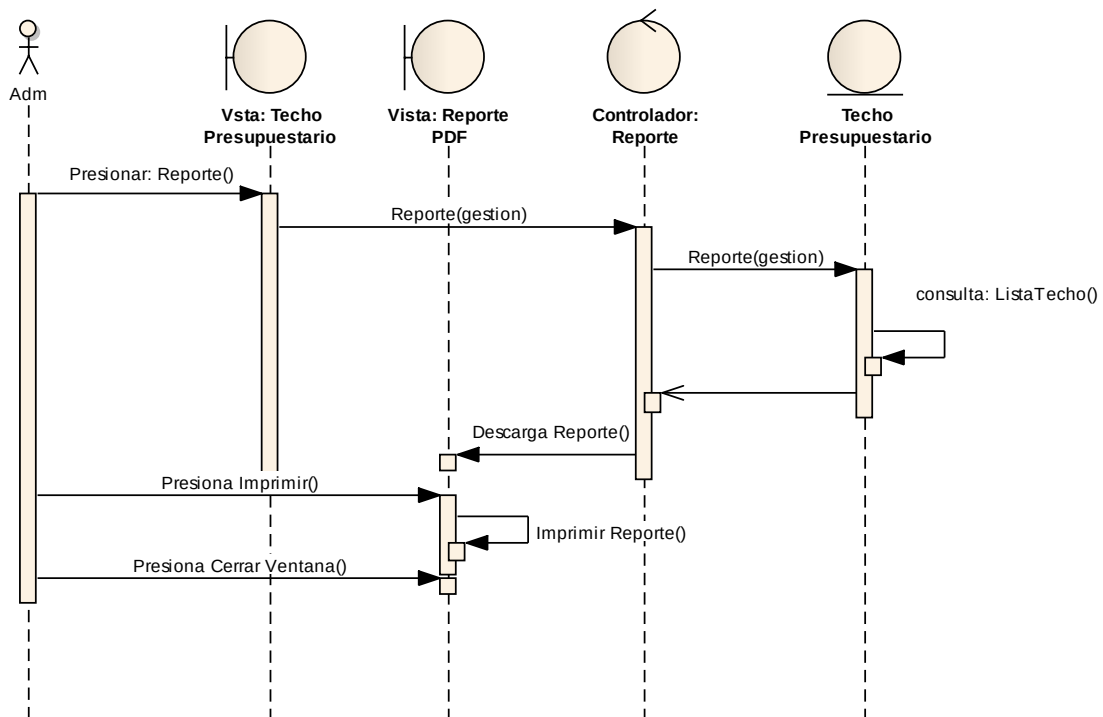


Figura 119 Diagrama de Secuencia: casos de uso Reporte Presupuesto

II.1.1.3.7.5. Diagrama de Secuencia: casos de uso Material / Servicio

II.1.1.3.7.5.1. Diagrama de Secuencia: casos de uso Gestion Material / Servicio

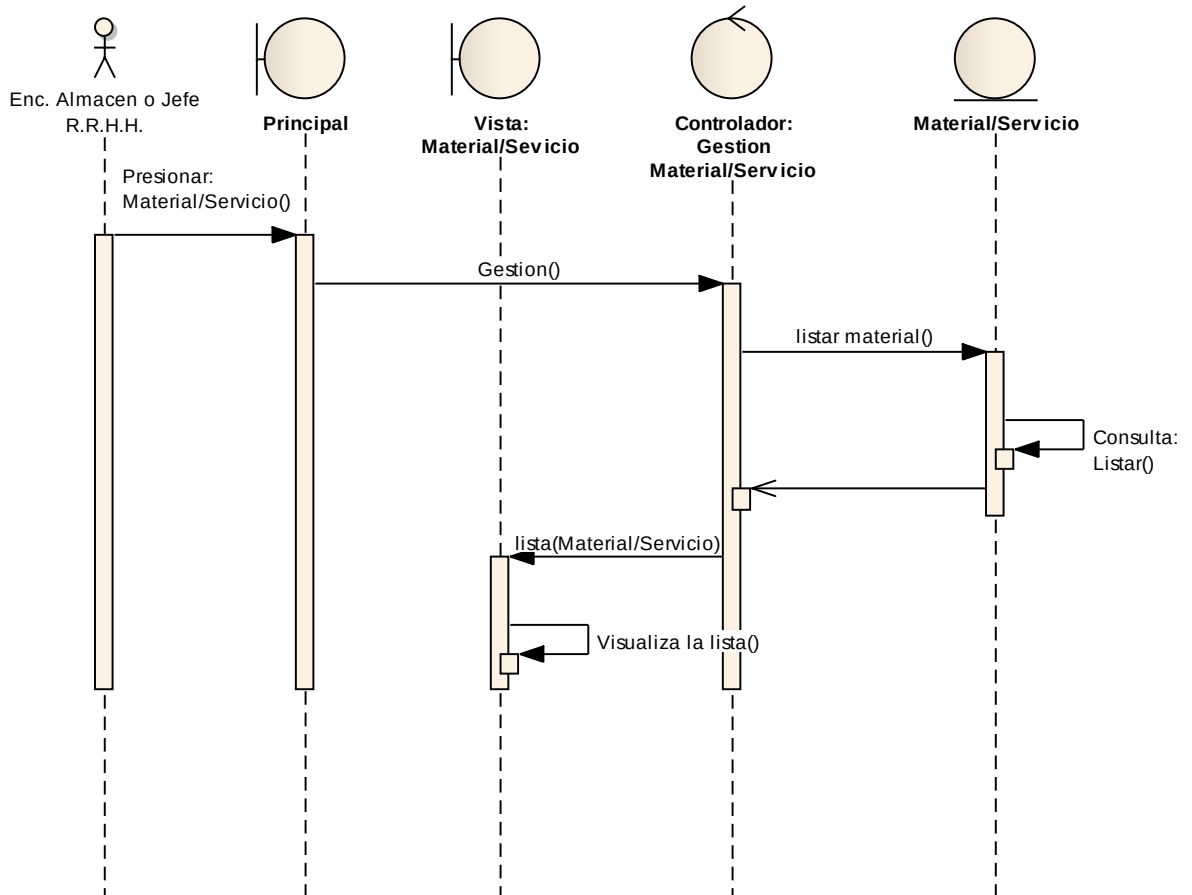


Figura 120 Diagrama de Secuencia: casos de uso Gestion Material / Servicio

II.1.1.3.7.5.2. Diagrama de Secuencia: casos de uso adicionar Material / Servicio

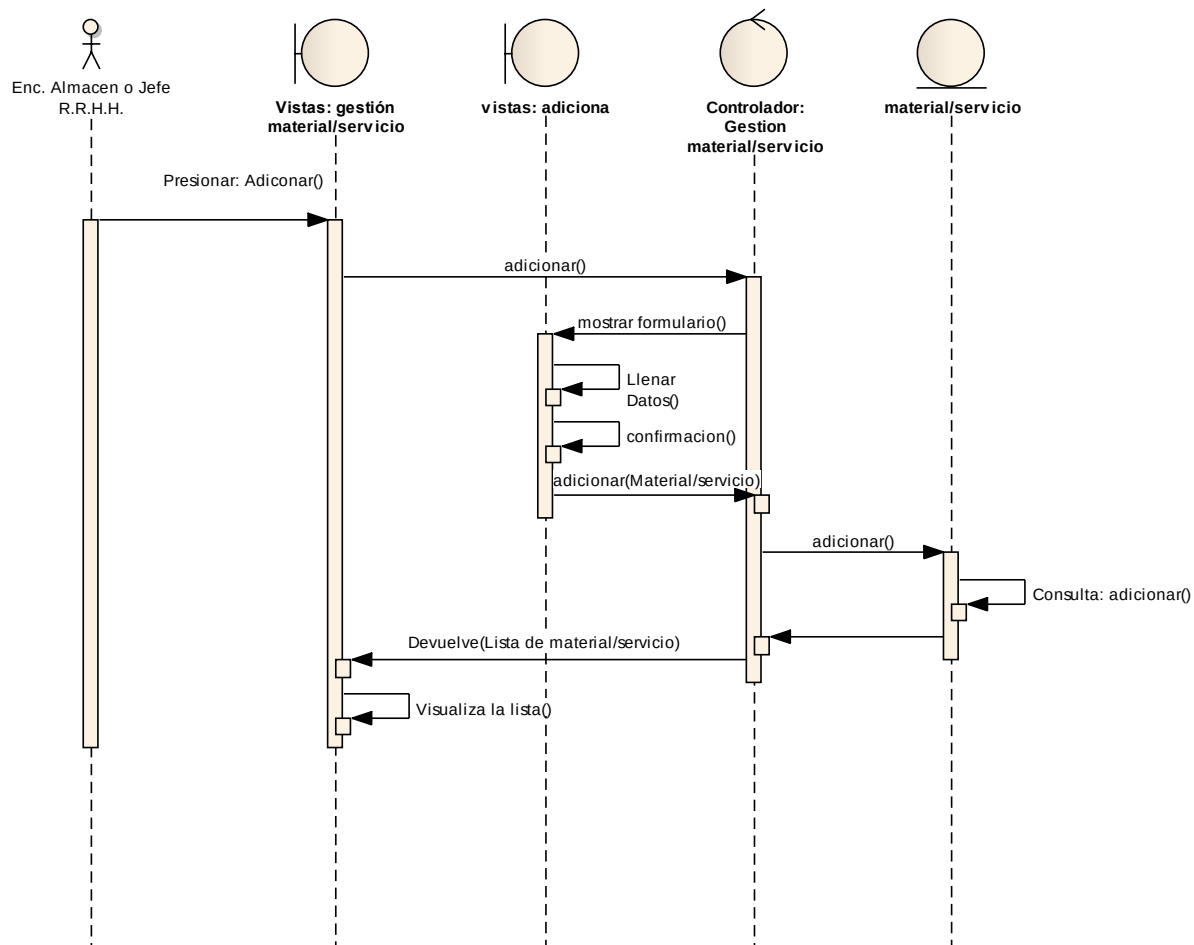


Figura 121 Diagrama de Secuencia: casos de uso adicionar Material / Servicio

II.1.1.3.8. Diagrama de Despliegue o Distribucion:

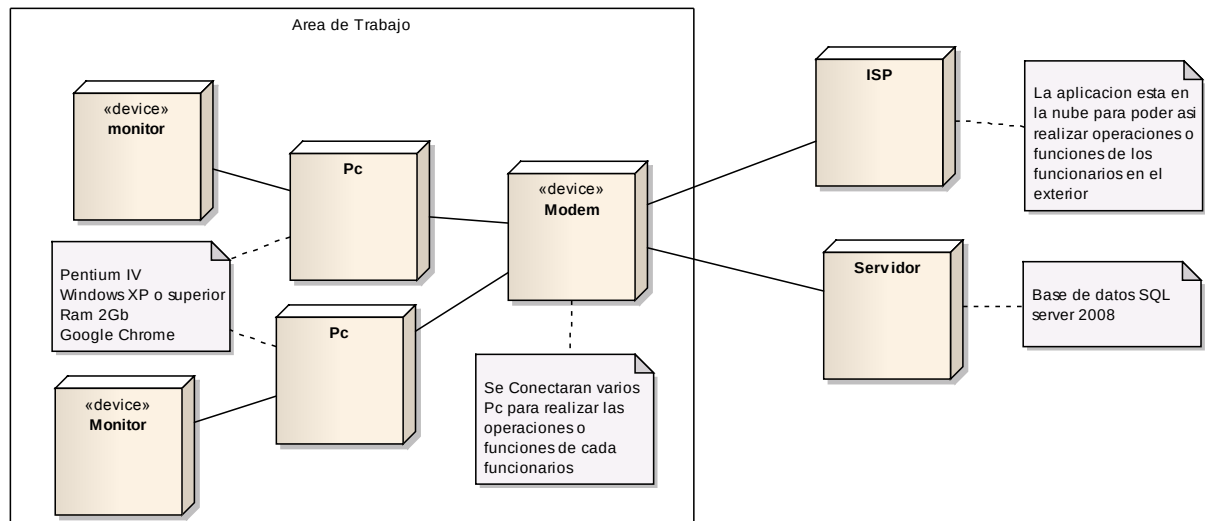


Figura 122 Diagrama de Despliegue o Distribucion:

II.1.1.3.9. Modelo Entidad Relación

II.1.1.3.9.1. Introducción

Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un ordenador; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas en el ordenador se debe realizar un modelo de datos.

Se suele cometer el error de ir creando nuevas tablas a medida que se van necesitando, haciendo así el modelo de datos y la construcción física de las tablas simultáneamente. El resultado de esto acaba siendo un sistema de información parcheado, con datos dispersos que terminan por no cumplir adecuadamente los requisitos necesarios.

II.1.1.3.9.2. Modelo Entidad / Relación

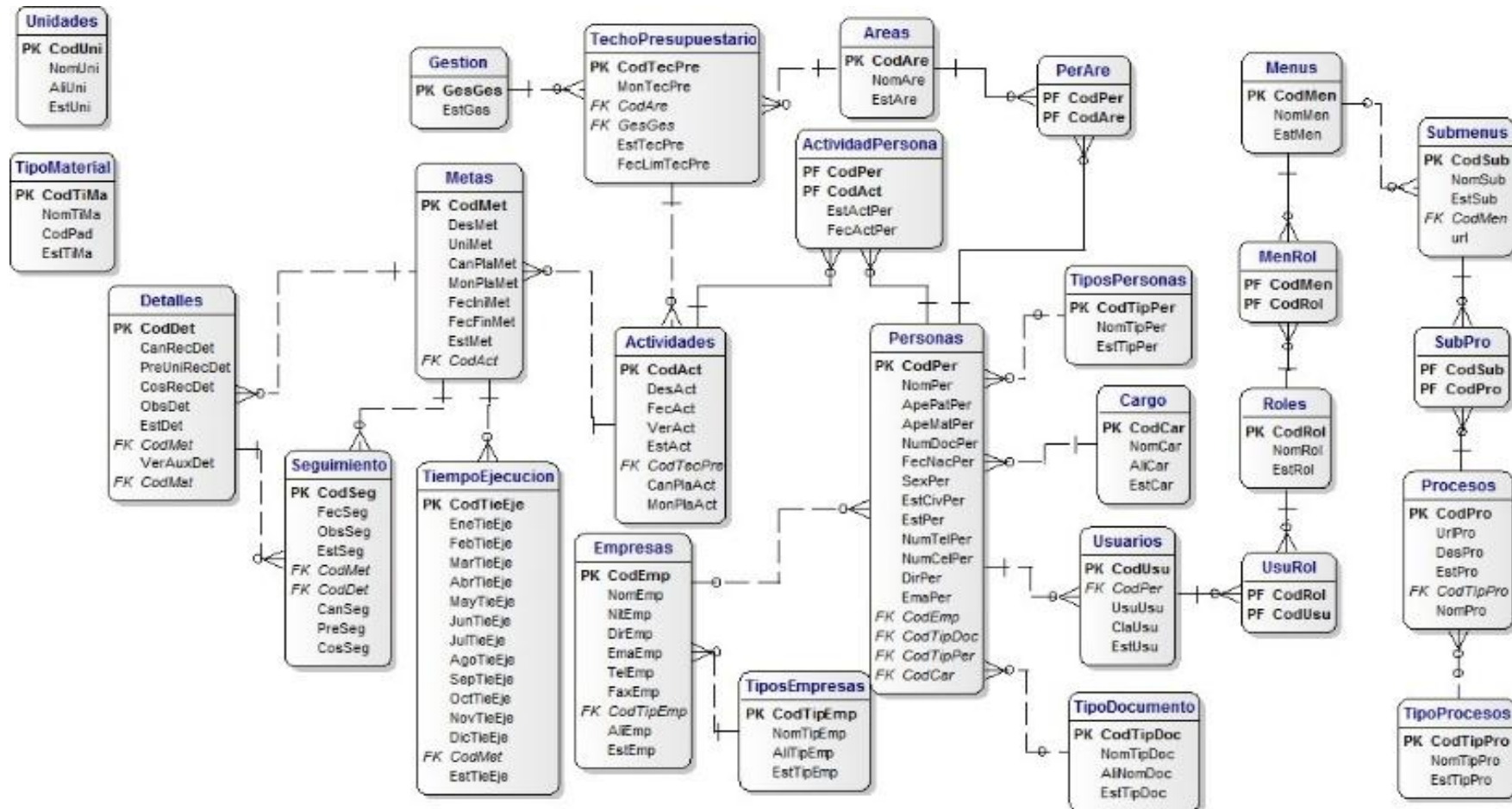


Figura 123 Modelo Entidad / Relación

II.1.1.3.9.3. Creación de la Base de Datos(Tablas y Procedimientos Almacenados)

```

USE [Siaffcc]
GO
/***** Object: StoredProcedimiento [dbo].[adicionarDetalle] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[adicionarDetalle]
    @DesRecDet as VARCHAR(150),
    @UniRecDet as VARCHAR(10),
    @CanRecDet as int,
    @PreUniRecDet as float,
    @ParDet as int,
    @ObsDet as VARCHAR(150),
    @TipRecDet as int,
    @CodAux as VARCHAR(12),
    @CodMet as int,
    @CodAct as int

AS
DECLARE @cod INT
DECLARE @costo FLOAT
BEGIN TRAN
BEGIN
    SET @cod=(SELECT MAX(CodDet) FROM Detalles)
    SET @cod=ISNULL(@cod, 0)+1
    set @costo=@CanRecDet*@PreUniRecDet
BEGIN TRY
insert into detalles(CodDet, DesRecDet, UniRecDet, CanRecDet, PreUniRecDet, CosRecDet,
ParDet, ObsDet, TipRecDet,CodAux, CodMet)
VALUES(@cod,@DesRecDet,@UniRecDet,@CanRecDet,@PreUniRecDet,CAST(@costo AS
DECIMAL(18,2)),@ParDet,@ObsDet,@TipRecDet
,@CodAux,@CodMet)

update Metas set MonPlaMet=(
        SELECT sum(CosRecDet)
        from Detalles
        where CodMet=@CodMet AND EstDet=1
    ) WHERE CodMet=@CodMet

update Actividades set MonPlaAct=(
        SELECT sum(MonPlaMet)

```

```

                                from Metas
                                where CodAct=@CodAct and EstMet=1
) WHERE CodAct=@CodAct

        COMMIT
END TRY
BEGIN CATCH
        ROLLBACK
        INSERT INTO LogError(descripcion) VALUES(ERROR_MESSAGE())
END CATCH
END
GO
/***** Object: StoredProcedimiento [dbo].[adicionarMaterial] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[adicionarMaterial]
    @NomMat as VARCHAR(80),
    @CodUni as int,
    @CosMat as float,
    @TipMat as NUMERIC,
    @ParMat as int,
    @CodAux as VARCHAR(12)
AS
    DECLARE @codmat INT
    BEGIN
        SET @codmat=(SELECT MAX(CodMat) FROM Materiales)
        SET @codmat=ISNULL(@codmat, 0)+1
        insert into materiales(CodMat, NomMat, CodUni, CosMat, TipMat,ParMat,CodAux)
        VALUES(@codmat,@NomMat,@CodUni,@CosMat,@TipMat,@ParMat,@CodAux)
    END
GO
/***** Object: StoredProcedimiento [dbo].[adicionarMaterialFun] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[adicionarMaterialFun]
    @NomMat as VARCHAR(80),
    @CodUni as int,
    @CosMat as float,
    @EstMat as NUMERIC,
    @TipMat as NUMERIC
AS
    DECLARE @codmat INT

```

```

BEGIN
SET @codmat=(SELECT MAX(CodMat) FROM Materiales)
    SET @codmat=ISNULL(@codmat, 0)+1
insert into materiales(CodMat, NomMat, CodUni, CosMat, EstMat, TipMat)
VALUES(@codmat,@NomMat,@CodUni,@CosMat,@EstMat,@TipMat)
END
GO
/***** Object: StoredProcedimiento [dbo].[adicionarMeta] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[adicionarMeta]
@DesMet as VARCHAR(250),
@CanPlaMet as int,
@FecIniMet as DATE,
@FecFinMet as DATE,
@CodAct as int,

@Ene as BIT,
@Feb as BIT,
@Mar as BIT,
@Abr as BIT,
@May as BIT,
@Jun as BIT,
@Jul as BIT,
@Ago as BIT,
@Sep as BIT,
@Oct as BIT,
@Nov as BIT,
@Dic as BIT

AS
DECLARE @codmet INT
DECLARE @codtie INT
BEGIN TRAN
BEGIN
    SET @codmet=(SELECT MAX(CodMet) FROM Metas)
    SET @codmet=ISNULL(@codmet, 0)+1
SET @codtie=(SELECT MAX(CodTieEje) FROM TiempoEjecucion)
    SET @codtie=ISNULL(@codtie, 0)+1
BEGIN TRY
insert into Metas(CodMet, DesMet, CanPlaMet, FecIniMet, FecFinMet, CodAct)
VALUES(@codmet,@DesMet,@CanPlaMet,@FecIniMet,@FecFinMet,@CodAct)
insert into TiempoEjecucion(CodTieEje,EneTieEje,FebTieEje,MarTieEje,AbrTieEje,
MayTieEje,JunTieEje,JulTieEje,AgoTieEje,SepTieEje,OctTieEje,NovTieEje,DicTieEje,CodMet)

```

```

values(@codtie,@Ene,@Feb,@Mar,@Abr,@May,@Jun,@Jul,@Ago,@Sep,@Oct,@Nov,@Dic,@codmet)
        COMMIT
END TRY
BEGIN CATCH
        ROLLBACK
        INSERT INTO LogError(descripcion) VALUES(ERROR_MESSAGE())
END CATCH
END
GO
/***** Object: StoredProcedimiento [dbo].[adicionarMontoTecho] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[adicionarMontoTecho]
    @CodAre as int,
    @Monto as int,
    @GesGes as VARCHAR(4)
AS
    DECLARE @cod INT
    BEGIN TRAN
    BEGIN
        SET @cod=(SELECT MAX(CodTecPre) FROM TechoPresupuestario)
        SET @cod=ISNULL(@cod, 0)+1

        insert into TechoPresupuestario(CodTecPre, CodAre,MonTecPre,GesGes)
        VALUES(@cod,@CodAre,@Monto,@GesGes)
        COMMIT

    END
    GO
/***** Object: StoredProcedimiento [dbo].[adicionarObjetivo] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[adicionarObjetivo]
    @DesAct As VARCHAR(250) ,
    @FecAct As DATE ,
    @CodPer AS int,
    @CodTecPre As INT
AS
    DECLARE @cod INT
    BEGIN TRAN
    BEGIN

```

```

        SET @cod=(SELECT MAX(CodAct) FROM Actividades)
        SET @cod=ISNULL(@cod, 0)+1
BEGIN TRY
    insert into Actividades(CodAct,DesAct,FecAct,CodPer,CodTecPre)
        values(@cod,@DesAct,@FecAct,@CodPer,@CodTecPre)
    COMMIT
END TRY
BEGIN CATCH
    ROLLBACK
    INSERT INTO LogError(descripcion) VALUES(ERROR_MESSAGE())
END CATCH
END
GO
/***** Object: StoredProcedimiento [dbo].[adicionarSeguimiento] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[adicionarSeguimiento]
@CodDet int,
@CanSeg int,
@PreSeg float,
@ObsSeg VARCHAR(150),
@CodMet int,
@fecha DATE,
@codact int
AS
DECLARE @cod INT
DECLARE @costo FLOAT
BEGIN
    SET @cod=(SELECT MAX(CodSeg) FROM Seguimiento)
    SET @cod=ISNULL(@cod, 0)+1
    set @costo=@CanSeg*@PreSeg
insert into Seguimiento(CodSeg, CanSeg, PreSeg, CosSeg,ObsSeg,FecSeg,CodMet,CodDet)
VALUES(@cod,@CanSeg,@PreSeg,CAST(@costo AS
DECIMAL(18,2)),@ObsSeg,@fecha,@CodMet,@CodDet)
END
GO
/***** Object: StoredProcedimiento [dbo].[aprobarDetalle] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[aprobarDetalle]
@coddet int,
@ver int

```

```

AS
BEGIN
update Detalles set VerAuxDet=@ver
WHERE CodDet=@CodDet
END
GO
/***** Object: StoredProcedimiento [dbo].[countMeta] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[countMeta]
@CodMet as int
AS
BEGIN
    SELECT COUNT(CodMet)
from Detalles
WHERE (CodMet=@CodMet and EstDet=1)
END
GO
/***** Object: StoredProcedimiento [dbo].[countObjetivo] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[countObjetivo]
@CodAct as int
AS
BEGIN
SELECT COUNT(CodAct)
from Metas
WHERE CodAct=@CodAct and EstMet=1
END
GO
/***** Object: StoredProcedimiento [dbo].[enviar] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[enviar]
    @CodAct as int,
    @est int
AS
BEGIN
update Actividades set VerAct=@est WHERE CodAct=@CodAct
END

```

```

GO
/***** Object: StoredProcedimiento [dbo].[incluirMaterial] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[incluirMaterial]
    @CodMat as int
AS
BEGIN
    update Materiales set EstMat=1
    WHERE CodMat=@CodMat
END
GO
/***** Object: StoredProcedimiento [dbo].[iniciarsesion] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[iniciarsesion]
    @UsuUsu as VARCHAR(50),
    @ClaUsu as VARCHAR(50)
AS
BEGIN
    select
    Personas.CodPer,Personas.NomPer,Personas.ApePatPer,Personas.ApeMatPer,Personas.NumDocPer,
    Personas.FecNacPer,Personas.SexPer,Personas.EstCivPer,Personas.EstPer,Personas.NumCelPer,Personas.NumTelPer,
    Personas.DirPer,Personas.EmaPer,Personas.CodEmp,Personas.CodTipPer,Personas.CodTipDoc
    from Personas JOIN Usuarios on Personas.CodPer=Usuarios.CodPer
    where  Usuarios.UsuUsu=@UsuUsu and Usuarios.ClaUsu=@ClaUsu and Usuarios.EstUsu=1
END
GO
/***** Object: StoredProcedimiento [dbo].[listaArea] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[listaArea]
AS
BEGIN
    SELECT CodAre,NomAre,EstAre
    FROM Areas
    WHERE EstAre=1

```

```

END
GO
/***** Object: StoredProcedimiento [dbo].[listaAreaPersona] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[listaAreaPersona]
@codare INT
AS
BEGIN
SELECT NomPer+' '+ApePatPer+' '+ApeMatPer as nombre, Personas.CodPer
FROM Personas JOIN PerAre ON Personas.CodPer=PerAre.CodPer
JOIN Areas on Areas.CodAre=PerAre.CodAre and Areas.CodAre=@codare
END
GO
/***** Object: StoredProcedimiento [dbo].[listaAreasTecho] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[listaAreasTecho]
@gestion VARCHAR(4)
AS
BEGIN
SELECT Areas.CodAre,Areas.NomAre,ISNULL(TechoPresupuestario.MonTecPre, 0) as
monto,TechoPresupuestario.EstTecPre,TechoPresupuestario.CodTecPre
from Areas LEFT OUTER JOIN TechoPresupuestario on
TechoPresupuestario.CodAre=Areas.CodAre and TechoPresupuestario.GesGes=@gestion
END
GO
/***** Object: StoredProcedimiento [dbo].[listaDetalleAprobado] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[listaDetalleAprobado]
@codmet INT
AS
BEGIN
select
Detalles.CodDet,DesRecDet,UniRecDet,CanRecDet,PreUniRecDet,CosRecDet,TipRecDet,Par
Det,CodMet
from Detalles where CodDet not in (select CodDet from Seguimiento) and EstDet=1 and
CodMet=@codmet
END

```

```

GO
/***** Object: StoredProcedimiento [dbo].[listaGestion] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[listaGestion]
AS
BEGIN
SELECT GesGes,EstGes
FROM Gestion
END
GO
/***** Object: StoredProcedimiento [dbo].[listaMaterialDetalle] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[listaMaterialDetalle]
@codmet int
AS
BEGIN
select m.CodMat,m.NomMat,m.ParMat,m.CosMat,m.tipMat,u.NomUni,m.CodAux
from Unidades u JOIN Materiales m on u.CodUni=m.CodUni and EstMat=1
WHERE CodMat not in (SELECT Detalles.CodAux

FROM Detalles JOIN Metas on
Metas.CodMet=Detalles.CodMet and Detalles.CodMet=@codmet )
GROUP BY m.CodMat,m.NomMat,m.ParMat,m.CosMat,m.tipMat,u.NomUni,m.CodAux
END
GO
/***** Object: StoredProcedimiento [dbo].[listaMetaSeguimiento] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[listaMetaSeguimiento]
@CodAct as int
AS
BEGIN
SELECT DISTINCT Metas.CodMet,DesMet,CanPlaMet,CONVERT(varchar(15), FecIniMet, 103)
as FecIniMet,
CONVERT(varchar(15), FecFinMet, 103) as FecFinMet,ISNULL(sum(Seguimiento.CosSeg), 0)
as monto
from Actividades JOIN Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1
and Metas.CodAct=@CodAct

```

```

LEFT OUTER JOIN Seguimiento on
Seguimiento.CodMet=Metas.CodMet and Seguimiento.EstSeg=1
GROUP BY Metas.CodMet,DesMet,CanPlaMet,FecIniMet,FecFinMet

END
GO
/***** Object: StoredProcedimiento [dbo].[listaObjetivoAprobado] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[listaObjetivoAprobado]
@CodPer As INT,
@gestion VARCHAR(4)
AS
BEGIN
SELECT DISTINCT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct,
Actividades.VerAct, Actividades.EstAct,
Actividades.CodPer, Actividades.CodTecPre, Actividades.CanPlaAct,
ISNULL(sum(Seguimiento.CosSeg), 0) as monto
from TechoPresupuestario JOIN Actividades on
Actividades.CodTecPre=TechoPresupuestario.CodTecPre and
TechoPresupuestario.EstTecPre=1
and TechoPresupuestario.GesGes=@gestion and
Actividades.CodPer=@CodPer and Actividades.VerAct=2
and Actividades.EstAct=1
JOIN Metas on Metas.CodAct=Actividades.CodAct and
Metas.EstMet=1
LEFT OUTER JOIN Seguimiento on
Seguimiento.CodMet=Metas.CodMet and Seguimiento.EstSeg=1
GROUP BY Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct,
Actividades.EstAct,
Actividades.CodPer, Actividades.CodTecPre, Actividades.CanPlaAct

END
GO
/***** Object: StoredProcedimiento [dbo].[listaObjetivoEvaluacion] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[listaObjetivoEvaluacion]
@CodPer As INT,
@gestion VARCHAR(4)
AS
BEGIN

```

```

SELECT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct,
Actividades.EstAct,
Actividades.CodPer, Actividades.CodTecPre, Actividades.MonPlaAct, Actividades.CanPlaAct
from TechoPresupuestario JOIN Actividades on
Actividades.CodTecPre=TechoPresupuestario.CodTecPre
and TechoPresupuestario.GesGes=@gestion and
Actividades.CodPer=@CodPer and Actividades.VerAct!=0

```

```
END
```

```
GO
```

```
/****** Object: StoredProcedimiento [dbo].[listaPersona] *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDIMIENTO [dbo].[listaPersona]
```

```
AS
```

```
BEGIN
```

```
SELECT NomPer+' '+ApePatPer+' '+ApeMatPer as nombre, CodPer
```

```
FROM Personas
```

```
WHERE EstPer=1
```

```
END
```

```
GO
```

```
/****** Object: StoredProcedimiento [dbo].[listarDetalle] *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDIMIENTO [dbo].[listarDetalle]
```

```
@CodMet as int,
```

```
@est as int
```

```
AS
```

```
BEGIN
```

```
SELECT CodMet,CodDet,DesRecDet,UniRecDet,CanRecDet,
```

```
PreUniRecDet,CosRecDet,ParDet,ObsDet,TipRecDet,EstDet,VerAuxDet
```

```
from Detalles
```

```
where (codmet=@CodMet)and(EstDet=@est)
```

```
END
```

```
GO
```

```
/****** Object: StoredProcedimiento [dbo].[listarmaterial] *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDIMIENTO [dbo].[listarmaterial]
```

```
AS
```

```

BEGIN
select m.CodMat,m.NomMat,m.ParMat,m.CosMat,m.tipMat,u.NomUni,m.CodAux
from Unidades u JOIN Materiales m on u.CodUni=m.CodUni and EstMat=1
END
GO
/***** Object: StoredProcedimiento [dbo].[listarMaterialFuncionario] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[listarMaterialFuncionario]
AS
BEGIN
select m.CodMat,m.NomMat,m.CosMat,m.tipMat,u.NomUni,m.ParMat,m.CodAux
from Unidades u JOIN Materiales m on u.CodUni=m.CodUni
where m.EstMat=3 or m.EstMat=4
END
GO
/***** Object: StoredProcedimiento [dbo].[listarMeta] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[listarMeta]
@CodAct as int,
@Est as int
AS
BEGIN
SELECT CodMet,DesMet,UniMet,CanPlaMet,MonPlaMet,CONVERT(varchar(15), FecIniMet,
103) as FecIniMet,
CONVERT(varchar(15), FecFinMet, 103) as FecFinMet,EstMet,CodAct
from Metas
WHERE CodAct=@CodAct and EstMet=@Est
END
GO
/***** Object: StoredProcedimiento [dbo].[listarObjetivo] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[listarObjetivo]
@CodPer As INT,
@EstAct AS INT,
@gestion VARCHAR(4)
AS
BEGIN

```

```

SELECT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct,
Actividades.EstAct,
Actividades.CodPer, Actividades.CodTecPre, Actividades.MonPlaAct, Actividades.CanPlaAct
from Gestion JOIN TechoPresupuestario on TechoPresupuestario.GesGes=Gestion.GesGes
and
TechoPresupuestario.GesGes=@gestion
JOIN Actividades on
Actividades.CodTecPre=TechoPresupuestario.CodTecPre
and Actividades.CodPer=@CodPer
and Actividades.EstAct=@EstAct

```

```
END
```

```
GO
```

```
/****** Object: StoredProcedimiento [dbo].[listarTechoPresupuestario] *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDIMIENTO [dbo].[listarTechoPresupuestario]
```

```
@Gestion VARCHAR(4)
```

```
AS
```

```
BEGIN
```

```
SELECT TechoPresupuestario.CodAre as CodAre, Areas.NomAre,
```

```

TechoPresupuestario.MonTecPre, TechoPresupuestario.GesGes, TechoPresupuestari
o.EstTecPre

```

```
FROM Areas JOIN TechoPresupuestario on Areas.CodAre=TechoPresupuestario.CodAre
```

```
WHERE TechoPresupuestario.GesGes=@Gestion and TechoPresupuestario.EstTecPre=1 or
TechoPresupuestario.EstTecPre=2
```

```
END
```

```
GO
```

```
/****** Object: StoredProcedimiento [dbo].[listarTiempoEjecucion] *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDIMIENTO [dbo].[listarTiempoEjecucion]
```

```
@CodAct as int,
```

```
@Est as int
```

```
AS
```

```
BEGIN
```

```
SELECT CodTieEje,
```

```
CASE WHEN EneTieEje=1 THEN 'X' ELSE ' ' END as EneTieEje,
```

```
CASE WHEN FebTieEje=1 THEN 'X' ELSE ' ' END as FebTieEje,
```

```
CASE WHEN MarTieEje=1 THEN 'X' ELSE ' ' END as MarTieEje,
```

```

CASE WHEN AbrTieEje=1 THEN 'X' ELSE ' ' END as AbrTieEje,
CASE WHEN MayTieEje=1 THEN 'X' ELSE ' ' END as MayTieEje,
CASE WHEN JunTieEje=1 THEN 'X' ELSE ' ' END as JunTieEje,
CASE WHEN JulTieEje=1 THEN 'X' ELSE ' ' END as JulTieEje,
CASE WHEN AgoTieEje=1 THEN 'X' ELSE ' ' END as AgoTieEje,
CASE WHEN SepTieEje=1 THEN 'X' ELSE ' ' END as SepTieEje,
CASE WHEN OctTieEje=1 THEN 'X' ELSE ' ' END as OctTieEje,
CASE WHEN NovTieEje=1 THEN 'X' ELSE ' ' END as NovTieEje,
CASE WHEN DicTieEje=1 THEN 'X' ELSE ' ' END as DicTieEje,
DesMet,CanPlaMet,MonPlaMet,Metas.CodMet
FROM Metas JOIN TiempoEjecucion on Metas.CodMet=TiempoEjecucion.CodMet and
Metas.CodAct=@CodAct
and EstTieEje=@Est
END
GO
/***** Object: StoredProcedimiento [dbo].[listarUnidad] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[listarUnidad]
AS
BEGIN
    SELECT CodUni,NomUni,AliUni
        FROM Unidades
END
GO
/***** Object: StoredProcedimiento [dbo].[listaSeguimiento] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[listaSeguimiento]
@codmet INT,
@fecha INT
AS
BEGIN
    SELECT Detalles.DesRecDet,Detalles.UniRecDet,Detalles.TipRecDet,Detalles.ParDet,

        Seguimiento.CanSeg,Seguimiento.PreSeg,Seguimiento.CosSeg,

        Seguimiento.FecSeg,Seguimiento.ObsSeg,Seguimiento.CodMet,Seguimiento.CodSeg
    from Seguimiento JOIN Detalles ON Detalles.CodDet=Seguimiento.CodDet
        and DATEPART(mm, Seguimiento.FecSeg)=@fecha and
    Seguimiento.CodMet=@codmet
END

```

```

GO
/***** Object: StoredProcedimiento [dbo].[modificarDetalle] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[modificarDetalle]
    @CodDet as int,
    @CanRecDet as int,
    @PreUniRecDet as float,
    @ObsDet as VARCHAR(150),
    @CodMet as int,
    @CodAct as int
AS
    DECLARE @costo FLOAT
    BEGIN
        set @costo=@CanRecDet*@PreUniRecDet
        update Detalles set
        CanRecDet=@CanRecDet,PreUniRecDet=@PreUniRecDet,ObsDet=@ObsDet,CosRecDet=@c
        osto
        WHERE CodDet=@CodDet

        update Metas set MonPlaMet=(
            SELECT sum(CosRecDet)
            from Detalles
            where CodMet=@CodMet AND EstDet=1
        ) WHERE CodMet=@CodMet

        update Actividades set MonPlaAct=(
            SELECT sum(MonPlaMet)
            from Metas
            where CodAct=@CodAct and EstMet=1
        ) WHERE CodAct=@CodAct

    END
GO
/***** Object: StoredProcedimiento [dbo].[modificarMaterialFun] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[modificarMaterialFun]
    @CodMat as NUMERIC,
    @NomMat as VARCHAR(80),
    @CodUni as int,

```

```

@CosMat as float,
@CodAux as VARCHAR(12),
@ParMat as int
AS
BEGIN
    update Materiales set
    NomMat=@NomMat,CodUni=@CodUni,CosMat=@CosMat,EstMat=4,TipMat=1,CodAux=@
    CodAux,ParMat=@ParMat
    WHERE CodMat=@CodMat
END
GO
/****** Object: StoredProcedimiento [dbo].[modificarMeta] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[modificarMeta]
@CodMet as int,
@DesMet as VARCHAR(250),
@CanPlaMet as int,
@FecIniMet as DATE,
@FecFinMet as DATE,
@CodAct as int,

@Ene as BIT,
@Feb as BIT,
@Mar as BIT,
@Abr as BIT,
@May as BIT,
@Jun as BIT,
@Jul as BIT,
@Ago as BIT,
@Sep as BIT,
@Oct as BIT,
@Nov as BIT,
@Dic as BIT
AS
BEGIN
    update Metas set DesMet=@DesMet, CanPlaMet=@CanPlaMet,
    FecIniMet=@FecIniMet,FecFinMet=@FecFinMet
    WHERE CodMet=@CodMet

    update TiempoEjecucion set
    EneTieEje=@Ene,FebTieEje=@Feb,MarTieEje=@Mar,AbrTieEje=@Abr,MayTieEje=@May,

```

```

JunTieEje=@Jun,JulTieEje=@Jul,AgoTieEje=@Ago,SepTieEje=@Sep,OctTieEje=@Oct,

NovTieEje=@Nov,DicTieEje=@Dic
WHERE CodMet=@CodMet

END
GO
/***** Object: StoredProcedimiento [dbo].[modificarMontoTecho] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[modificarMontoTecho]
    @codtecpres as int,
    @Monto as int,
    @codare as int,
    @gesges as VARCHAR(4)
AS
BEGIN
    update TechoPresupuestario set MonTecPre=@Monto,CodAre=@codare,GesGes=@gesges
    WHERE CodTecPre=@codtecpres
END
GO
/***** Object: StoredProcedimiento [dbo].[modificarObjetivo] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[modificarObjetivo]
    @CodAct as int,
    @DesAct As VARCHAR(250) ,
    @FecAct As DATE
AS
BEGIN
    update Actividades set DesAct=@DesAct,FecAct=@FecAct WHERE CodAct=@CodAct
END
GO
/***** Object: StoredProcedimiento [dbo].[modificarSeguimiento] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[modificarSeguimiento]
    @CodSeg int,

```

```

@CanSeg int,
@PreSeg FLOAT,
@obsseg VARCHAR(1000)
AS
DECLARE @costo FLOAT
BEGIN
    set @costo=@CanSeg*@PreSeg
update Seguimiento set
CanSeg=@CanSeg,PreSeg=@PreSeg,CosSeg=@costo,ObsSeg=@obsseg
WHERE CodSeg=@CodSeg
END
GO
/***** Object: StoredProcedimiento [dbo].[obtenerCantidad] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[obtenerCantidad]
@coddet int
AS
BEGIN
SELECT ISNULL((Detalles.CanRecDet-(
                        SELECT sum(Seguimiento.CanSeg)
                        FROM Metas JOIN Detalles on
Detalles.CodMet=Metas.CodMet AND Detalles.CodDet=@coddet
                        LEFT JOIN Seguimiento on
Seguimiento.CodMet=Metas.CodMet and Detalles.CodDet=Seguimiento.CodDet)),0) as
cantidad
FROM Metas JOIN Detalles on Detalles.CodMet=Metas.CodMet AND
Detalles.CodDet=@coddet
LEFT JOIN Seguimiento on Seguimiento.CodMet=Metas.CodMet and
Detalles.CodDet=Seguimiento.CodDet
END
GO
/***** Object: StoredProcedimiento [dbo].[obtenerDetalle] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[obtenerDetalle]
@CodDet as int
AS
BEGIN
SELECT
DesRecDet,UniRecDet,CanRecDet,PreUniRecDet,ParDet,ObsDet,TipRecDet,CodAux,CodDet,
CodMet

```

```

FROM Detalles
WHERE CodDet=@CodDet
END
GO
/***** Object: StoredProcedimiento [dbo].[obtenerMaterial] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[obtenerMaterial]
@CodMat as VARCHAR(12)
AS
BEGIN
    select m.CodMat,m.NomMat,m.ParMat,m.CosMat,m.tipMat,u.NomUni,m.CodAux
    from Unidades u JOIN Materiales m on u.CodUni=m.CodUni and m.CodMat=@CodMat
END
GO
/***** Object: StoredProcedimiento [dbo].[obtenerMeta] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[obtenerMeta]
@CodMet as int
AS
BEGIN
    SELECT CodMet, DesMet, CanPlaMet, CONVERT(varchar(15), FecIniMet, 103) as 'from',
    CONVERT(varchar(15), FecFinMet, 103) as 'to', CodAct, MonPlaMet, UniMet
    FROM Metas
    WHERE CodMet=@CodMet
END
GO
/***** Object: StoredProcedimiento [dbo].[obtenerMontoTecho] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[obtenerMontoTecho]
@CodTecPre as int
AS
BEGIN
    SELECT CodTecPre,CodAre,MonTecPre,GesGes
    FROM TechoPresupuestario
    WHERE CodTecPre=@CodTecPre
END
GO

```

```

/***** Object: StoredProcedimiento [dbo].[obtenerObjetivo] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[obtenerObjetivo]
@CodAct as int
AS
BEGIN
    select * from Actividades where CodAct=@CodAct
END
GO
/***** Object: StoredProcedimiento [dbo].[obtenerSeguimiento] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[obtenerSeguimiento]
@codseg int
AS
BEGIN
SELECT
CodSeg,Seguimiento.CodDet,Seguimiento.CodMet,CodSeg,ObsSeg,FecSeg,PreSeg,CosSeg,CanSeg,EstSeg,Detalles.DesRecDet,Detalles.UniRecDet
FROM Seguimiento JOIN Detalles on Seguimiento.CodDet=Detalles.CodDet and
Seguimiento.CodSeg=@codseg
END
GO
/***** Object: StoredProcedimiento [dbo].[reporteDetallePersona] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[reporteDetallePersona]
@codper int
AS
BEGIN
    SELECT CodDet,DesRecDet,UniRecDet,CanRecDet,
PreUniRecDet,CosRecDet,ParDet,ObsDet,TipRecDet,EstDet,VerAuxDet
from Personas join Actividades on Actividades.CodPer=Personas.CodPer
and Actividades.EstAct=1 and Personas.CodPer=@codper
            join Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1
join Detalles on Detalles.CodMet=Metas.CodMet and Detalles.EstDet=1
END
GO
/***** Object: StoredProcedimiento [dbo].[ReporteMetasActividades] *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[ReporteMetasActividades]
@CodPer INT,
@CodMet int,
@gestion VARCHAR(4)
AS
BEGIN
    SELECT Actividades.CodAct,CodMet,DesMet,CanPlaMet,MonPlaMet,CONVERT(varchar(15),
    FecIniMet, 103) as FecIniMet,
    CONVERT(varchar(15), FecFinMet, 103) as FecFinMet,TechoPresupuestario.GesGes
    from Personas join Actividades on Actividades.CodPer=Personas.CodPer
                                and
    Actividades.EstAct=1 and Personas.CodPer=@CodPer
                                join Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1 and
    Metas.CodMet=@CodMet
                                JOIN TechoPresupuestario on
    TechoPresupuestario.CodTecPre=Actividades.CodTecPre
                                and TechoPresupuestario.GesGes=@gestion
END

GO
/***** Object: StoredProcedimiento [dbo].[ReporteMetasDetalles] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[ReporteMetasDetalles]
@CodPer INT,
@CodAct int,
@codmet int,
@gestion VARCHAR(4)
AS
BEGIN
    SELECT Actividades.CodAct,Metas.CodMet,CodDet,DesRecDet,UniRecDet,CanRecDet,
    PreUniRecDet,CosRecDet,ParDet,ObsDet,TipRecDet,EstDet,VerAuxDet
    from Personas join Actividades on Actividades.CodPer=Personas.CodPer
                                and
    Actividades.EstAct=1 and Personas.CodPer=@CodPer and Actividades.CodAct=@CodAct
                                join Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1
                                JOIN Detalles on Detalles.CodMet=Metas.CodMet and Detalles.CodMet=@codmet
                                JOIN TechoPresupuestario on
    TechoPresupuestario.CodTecPre=Actividades.CodTecPre
                                and TechoPresupuestario.GesGes=@gestion

```

```

END
GO
/***** Object: StoredProcedimiento [dbo].[ReporteObjetivo] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[ReporteObjetivo]
    @CodPer As INT,
    @version int,
    @gestion VARCHAR(4)
AS
BEGIN
    SELECT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct,
    Actividades.EstAct,
    Actividades.CodPer, Actividades.CodTecPre, Actividades.MonPlaAct, Actividades.CanPlaAct
    from Gestion JOIN TechoPresupuestario on TechoPresupuestario.GesGes=Gestion.GesGes
                                and
    TechoPresupuestario.GesGes=@gestion
                                JOIN Actividades on
    Actividades.CodTecPre=TechoPresupuestario.CodTecPre
                                and Actividades.CodPer=@CodPer
    and Actividades.EstAct=1 and Actividades.VerAct=@version

END
GO
/***** Object: StoredProcedimiento [dbo].[ReporteSeguimiento] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[ReporteSeguimiento]
    @codper int,
    @gestion VARCHAR(4)
AS
BEGIN
    SELECT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct,
    Actividades.EstAct,
    Actividades.CodPer, Actividades.CodTecPre, Actividades.MonPlaAct, Actividades.CanPlaAct,
    (SELECT SUM(CosSeg) as monto
    FROM Seguimiento JOIN Metas on Metas.CodMet=Seguimiento.CodMet
                                JOIN Actividades on Actividades.CodAct=Metas.CodAct
    JOIN Personas on Personas.CodPer=Actividades.CodPer and Actividades.CodPer=@CodPer
    )
    from Gestion JOIN TechoPresupuestario on TechoPresupuestario.GesGes=Gestion.GesGes

```

```

                                and
TechoPresupuestario.GesGes=@gestion
                                JOIN Actividades on
Actividades.CodTecPre=TechoPresupuestario.CodTecPre
                                and
Actividades.CodPer=@CodPer and Actividades.EstAct=1 and Actividades.VerAct=2

END
GO
/***** Object: StoredProcedimiento [dbo].[reporteTiempoEjecucionPersona] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[reporteTiempoEjecucionPersona]
    @codper int,
    @codact int,
    @gestion VARCHAR(4)
AS
BEGIN
    SELECT
    Actividades.CodAct,CodTieEje,EneTieEje,FebTieEje,MarTieEje,AbrTieEje,MayTieEje,JunTieEje
    ,
    JulTieEje,AgoTieEje,SepTieEje,OctTieEje,NovTieEje,DicTieEje,DesMet,CanPlaMet,MonPlaMet
    from Personas join Actividades on Actividades.CodPer=Personas.CodPer
                                and Actividades.EstAct=1 and Personas.CodPer=@codper
    and Actividades.CodAct=@codact
                                join Metas on Metas.CodAct=Actividades.CodAct and
    Metas.EstMet=1
                                join TiempoEjecucion on TiempoEjecucion.CodMet=Metas.CodMet
    and TiempoEjecucion.EstTieEje=1
                                JOIN TechoPresupuestario on
    TechoPresupuestario.CodTecPre=Actividades.CodTecPre
                                and TechoPresupuestario.GesGes=@gestion

END
GO
/***** Object: StoredProcedimiento [dbo].[revertirAreaTecho] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[revertirAreaTecho]
    @codtecpre int,
    @estado NUMERIC(1)
AS

```

```

BEGIN
update TechoPresupuestario set EstTecPre=@estado
WHERE CodTecPre=@codtecpre
END
GO
/***** Object: StoredProcedimiento [dbo].[revertirDetalle] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[revertirDetalle]
@CodDet as int,
@EstDet as int
AS
BEGIN
    update Detalles set EstDet=@EstDet
    WHERE CodDet=@CodDet
END
GO
/***** Object: StoredProcedimiento [dbo].[revertirMeta] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[revertirMeta]
@CodMet as int,
@EstMet as int
AS
BEGIN
    update Metas set EstMet=@EstMet
    WHERE CodMet=@CodMet
    update TiempoEjecucion set EstTieEje=@EstMet
    WHERE CodMet=@CodMet
END
GO
/***** Object: StoredProcedimiento [dbo].[revertirObjetivo] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDIMIENTO [dbo].[revertirObjetivo]
@CodAct as int,
@EstAct as int
AS
BEGIN
    update Actividades set EstAct=@EstAct

```

```

WHERE CodAct=@CodAct
END
GO
/***** Object: Table [dbo].[Actividades] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Actividades](
    [CodAct] [int] NOT NULL,
    [DesAct] [varchar](250) NOT NULL,
    [FecAct] [date] NOT NULL,
    [VerAct] [int] NOT NULL,
    [EstAct] [int] NOT NULL,
    [CodPer] [int] NOT NULL,
    [CodTecPre] [int] NOT NULL,
    [CanPlaAct] [int] NOT NULL,
    [MonPlaAct] [float] NOT NULL,
    CONSTRAINT [PK__Activida__986747194D5F7D71] PRIMARY KEY CLUSTERED
(
    [CodAct] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[ALMACEN$] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ALMACEN$](
    [m_codigo] [nvarchar](255) NOT NULL,
    [m_unidad] [varchar](15) NULL,
    [m_nombre] [nvarchar](255) NULL,
    [m_cantidad] [float] NULL,
    [m_costo] [float] NULL,
    [m_saldofis] [float] NULL,
    [m_codgasto-Partida] [nvarchar](255) NULL,
    [m_codcontable-Relacion Contable] [nvarchar](255) NULL,

```

```

        [m_afisico] [float] NULL,
        [m_avalorado] [float] NULL,
        [m_adolar] [float] NULL,
        [m_afecha] [nvarchar](255) NULL,
        [m_fingreso] [float] NULL,
PRIMARY KEY CLUSTERED
(
        [m_codigo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Areas] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Areas](
        [CodAre] [int] NOT NULL,
        [NomAre] [varchar](50) NOT NULL,
        [EstAre] [bit] NOT NULL,
CONSTRAINT [PK_Areas] PRIMARY KEY CLUSTERED
(
        [CodAre] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Cotizacion] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Cotizacion](
        [CodCot] [int] NOT NULL,
        [EstCot] [int] NULL,
        [BruCot] [int] NULL,
        [DesCot] [int] NULL,

```

```

        [NetCot] [int] NULL,
        [TotCot] [float] NULL,
        [CodSolCot] [int] NULL,
        [CodMat] [int] NULL,
        [CanCot] [int] NULL,
PRIMARY KEY CLUSTERED
(
        [CodCot] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Detalles] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Detalles](
        [CodDet] [int] NOT NULL,
        [DesRecDet] [varchar](150) NOT NULL,
        [UniRecDet] [varchar](10) NOT NULL,
        [CanRecDet] [int] NOT NULL,
        [PreUniRecDet] [float] NOT NULL,
        [CosRecDet] [float] NULL,
        [ParDet] [int] NOT NULL,
        [ObsDet] [varchar](150) NULL,
        [TipRecDet] [int] NOT NULL,
        [EstDet] [int] NOT NULL,
        [CodMet] [int] NOT NULL,
        [VerAuxDet] [int] NOT NULL,
        [CodAux] [varchar](12) NOT NULL,
PRIMARY KEY CLUSTERED
(
        [CodDet] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Empresas] *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Empresas](
    [CodEmp] [int] NOT NULL,
    [NomEmp] [varchar](100) NULL,
    [NitEmp] [varchar](20) NULL,
    [DirEmp] [varchar](50) NULL,
    [EmaEmp] [varchar](30) NULL,
    [TelEmp] [varchar](30) NULL,
    [FaxEmp] [varchar](30) NULL,
    [CodTipEmp] [int] NOT NULL,
    [AliEmp] [varchar](15) NOT NULL,
    [EstEmp] [bit] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [CodEmp] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Gestion] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Gestion](
    [GesGes] [varchar](4) NOT NULL,
    [EstGes] [bit] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [GesGes] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO

```

```

/***** Object: Table [dbo].[Marcas] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Marcas](
    [CodMar] [int] NOT NULL,
    [NomMar] [varchar](30) NULL,
    [AliMar] [varchar](15) NULL,
    [EstMar] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [CodMar] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[MateMedi] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MateMedi](
    [CodMat] [int] NOT NULL,
    [CodMed] [int] NOT NULL,
CONSTRAINT [PK__MateMedi__F19BFCA824134F1B] PRIMARY KEY CLUSTERED
(
    [CodMed] ASC,
    [CodMat] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Material] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO

```

```

CREATE TABLE [dbo].[Material](
    [CodMat] [numeric](5, 0) NOT NULL,
    [NomMat] [varchar](80) NOT NULL,
    [ParMat] [int] NOT NULL,
    [CodUni] [int] NOT NULL,
    [CosMat] [float] NOT NULL,
    [TipMat] [numeric](1, 0) NOT NULL,
    [EstMat] [numeric](1, 0) NOT NULL,
    [CodAux] [varchar](12) NOT NULL,
    CONSTRAINT [PK__Material__837CFD430DEF03D2] PRIMARY KEY CLUSTERED
(
    [CodMat] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Materiales] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Materiales](
    [CodMat] [numeric](5, 0) NOT NULL,
    [NomMat] [varchar](80) NOT NULL,
    [ParMat] [int] NOT NULL,
    [CodUni] [int] NOT NULL,
    [CosMat] [float] NOT NULL,
    [TipMat] [numeric](1, 0) NOT NULL,
    [EstMat] [numeric](1, 0) NOT NULL,
    [CodAux] [varchar](12) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodMat] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Medidas] *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Medidas](
    [CodMed] [int] NOT NULL,
    [NomMed] [varchar](30) NULL,
    [AliMed] [varchar](15) NULL,
    [EstMed] [int] NULL,
    PRIMARY KEY CLUSTERED
(
    [CodMed] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[MenRol] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MenRol](
    [CodMen] [int] NOT NULL,
    [CodRol] [int] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodMen] ASC,
    [CodRol] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Menus] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Menus](

```

```

        [CodMen] [int] NOT NULL,
        [NomMen] [varchar](50) NOT NULL,
        [EstMen] [bit] NOT NULL,
CONSTRAINT [PK_Menus] PRIMARY KEY CLUSTERED
(
        [CodMen] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Metas] *****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

SET ANSI_PADDING ON

```

```

GO

```

```

CREATE TABLE [dbo].[Metas](
        [CodMet] [int] NOT NULL,
        [DesMet] [varchar](250) NOT NULL,
        [UniMet] [varchar](10) NULL,
        [CanPlaMet] [int] NOT NULL,
        [MonPlaMet] [float] NULL,
        [FecIniMet] [date] NOT NULL,
        [FecFinMet] [date] NOT NULL,
        [EstMet] [int] NOT NULL,
        [CodAct] [int] NOT NULL,

```

```

PRIMARY KEY CLUSTERED

```

```

(

```

```

        [CodMet] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO

```

```

SET ANSI_PADDING OFF

```

```

GO

```

```

/***** Object: Table [dbo].[PedidoCompras] *****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

SET ANSI_PADDING ON

```

```

GO
CREATE TABLE [dbo].[PedidoCompras](
    [CodPedCom] [int] NOT NULL,
    [FecReaPedCom] [varchar](20) NULL,
    [FecEntPedCom] [varchar](20) NULL,
    [ResComPedCom] [varchar](80) NULL,
    [CodSolCot] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [CodPedCom] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[PerAre] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[PerAre](
    [CodPer] [int] NOT NULL,
    [CodAre] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [CodPer] ASC,
    [CodAre] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
/***** Object: Table [dbo].[Personas] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Personas](
    [CodPer] [int] NOT NULL,
    [NomPer] [varchar](50) NOT NULL,
    [ApePatPer] [varchar](30) NOT NULL,
    [ApeMatPer] [varchar](30) NULL,

```

```

        [NumDocPer] [varchar](20) NULL,
        [FecNacPer] [varchar](15) NULL,
        [SexPer] [int] NULL,
        [EstCivPer] [int] NULL,
        [EstPer] [int] NULL,
        [NumTelPer] [varchar](20) NULL,
        [NumCelPer] [varchar](20) NULL,
        [DirPer] [varchar](100) NULL,
        [EmaPer] [varchar](50) NULL,
        [CodEmp] [int] NULL,
        [CodTipDoc] [int] NULL,
        [CodTipPer] [int] NULL,
CONSTRAINT [PK_Personas] PRIMARY KEY CLUSTERED
(
    [CodPer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Procesos] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Procesos](
    [CodPro] [int] NOT NULL,
    [UrlPro] [varchar](150) NOT NULL,
    [DesPro] [varchar](100) NULL,
    [EstPro] [bit] NOT NULL,
    [CodTipPro] [int] NULL,
    [NomPro] [varchar](50) NULL,
CONSTRAINT [PK_Procesos] PRIMARY KEY CLUSTERED
(
    [CodPro] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO

```

```

/***** Object: Table [dbo].[Roles] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Roles](
    [CodRol] [int] NOT NULL,
    [NomRol] [varchar](50) NOT NULL,
    [EstRol] [bit] NOT NULL,
    CONSTRAINT [PK_Roles] PRIMARY KEY CLUSTERED
(
    [CodRol] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Seguimiento] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Seguimiento](
    [CodSeg] [numeric](5, 0) NOT NULL,
    [FecSeg] [date] NOT NULL,
    [ObsSeg] [varchar](1000) NULL,
    [EstSeg] [numeric](1, 0) NOT NULL,
    [CodMet] [int] NOT NULL,
    [CodDet] [int] NOT NULL,
    [CanSeg] [int] NOT NULL,
    [PreSeg] [float] NOT NULL,
    [CosSeg] [float] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodSeg] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

```

```

SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[SolComMat] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SolComMat](
    [CodSolCom] [int] NOT NULL,
    [CanSolComMat] [int] NOT NULL,
    [EstSolComMat] [int] NOT NULL,
    [CodMat] [int] NOT NULL,
    [CodSolComMat] [int] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodSolComMat] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[SolicitudCompra] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[SolicitudCompra](
    [CodSolCom] [int] NOT NULL,
    [CodAre] [int] NOT NULL,
    [EstSolCom] [bit] NOT NULL,
    [ResSolCom] [varchar](50) NOT NULL,
    [FecReaSolCom] [date] NULL,
    [FecNecSolCom] [date] NULL,
    CONSTRAINT [PK__Solicitu__1BFBE4BE1E6F845E] PRIMARY KEY CLUSTERED
(
    [CodSolCom] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[SolicitudCotizacion] *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[SolicitudCotizacion](
    [CodSolCot] [int] NOT NULL,
    [DesPagSolCot] [int] NULL,
    [ConPagSolCot] [varchar](50) NULL,
    [ResSolCot] [varchar](50) NOT NULL,
    [ProSolCot] [varchar](50) NOT NULL,
    [CodSolCom] [int] NOT NULL,
    [EstSolCot] [int] NULL,
    [FecReaSolCot] [varchar](20) NULL,
    [DomSolCot] [varchar](100) NULL,
    CONSTRAINT [PK__Solicitu__90FEF499382F5661] PRIMARY KEY CLUSTERED
(
    [CodSolCot] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Submenus] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Submenus](
    [CodSub] [int] NOT NULL,
    [NomSub] [varchar](50) NOT NULL,
    [EstSub] [bit] NOT NULL,
    [CodMen] [int] NULL,
    [url] [varchar](150) NULL,
    CONSTRAINT [PK_Submenus] PRIMARY KEY CLUSTERED
(
    [CodSub] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[SubPro] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SubPro](
    [CodSub] [int] NOT NULL,
    [CodPro] [int] NOT NULL,
    CONSTRAINT [PK_SubPro] PRIMARY KEY CLUSTERED
(
    [CodSub] ASC,
    [CodPro] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[TechoPresupuestario] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TechoPresupuestario](
    [CodTecPre] [int] NOT NULL,
    [MonTecPre] [int] NOT NULL,
    [CodAre] [int] NOT NULL,
    [GesGes] [varchar](4) NOT NULL,
    [EstTecPre] [numeric](1, 0) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodTecPre] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TiempoEjecucion] *****/
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TiempoEjecucion](
    [CodTieEje] [int] NOT NULL,
    [EneTieEje] [bit] NULL,
    [FebTieEje] [bit] NULL,
    [MarTieEje] [bit] NULL,
    [AbrTieEje] [bit] NULL,
    [MayTieEje] [bit] NULL,
    [JunTieEje] [bit] NULL,
    [JulTieEje] [bit] NULL,
    [AgoTieEje] [bit] NULL,
    [SepTieEje] [bit] NULL,
    [OctTieEje] [bit] NULL,
    [NovTieEje] [bit] NULL,
    [DicTieEje] [bit] NULL,
    [CodMet] [int] NOT NULL,
    [EstTieEje] [int] NULL,
    PRIMARY KEY CLUSTERED
(
    [CodTieEje] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[TipoDocumento] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TipoDocumento](
    [CodTipDoc] [int] NOT NULL,
    [NomTipDoc] [varchar](50) NOT NULL,
    [AliNomDoc] [varchar](3) NOT NULL,
    [EstTipDoc] [bit] NOT NULL,
    CONSTRAINT [PK__TipoDocu__56B86A1A498EEC8D] PRIMARY KEY CLUSTERED
(
    [CodTipDoc] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

```

```

SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TipoMateriales] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TipoMateriales](
    [CodTipMat] [int] NOT NULL,
    [NomTipMat] [varchar](30) NOT NULL,
    [EstTipMat] [int] NOT NULL,
    [AliTipMat] [varchar](20) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodTipMat] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TipoProcesos] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TipoProcesos](
    [CodTipPro] [int] NOT NULL,
    [NomTipPro] [varchar](50) NOT NULL,
    [EstTipPro] [bit] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodTipPro] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TiposEmpresas] *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TiposEmpresas](
    [CodTipEmp] [int] NOT NULL,
    [NomTipEmp] [varchar](30) NOT NULL,
    [AliTipEmp] [varchar](15) NOT NULL,
    [EstTipEmp] [bit] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodTipEmp] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TiposPersonas] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TiposPersonas](
    [CodTipPer] [int] NOT NULL,
    [NomTipPer] [varchar](30) NOT NULL,
    [EstTipPer] [bit] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [CodTipPer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Unidades] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Unidades](
    [CodUni] [int] NOT NULL,
    [NomUni] [varchar](30) NULL,
    [AliUni] [varchar](10) NULL,
    [EstUni] [int] NULL,
    PRIMARY KEY CLUSTERED
(
    [CodUni] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Usuarios] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Usuarios](
    [CodUsu] [int] NOT NULL,
    [CodPer] [int] NOT NULL,
    [UsuUsu] [varchar](50) NOT NULL,
    [ClaUsu] [varchar](50) NOT NULL,
    [EstUsu] [int] NULL,
    CONSTRAINT [PK_DatosUsuarios] PRIMARY KEY CLUSTERED
(
    [CodUsu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[UsuRol] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE TABLE [dbo].[UsuRol](
    [CodRol] [int] NOT NULL,
    [CodUsu] [int] NOT NULL,
    CONSTRAINT [PK_UsuRol] PRIMARY KEY CLUSTERED
(
    [CodRol] ASC,
    [CodUsu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[Actividades] ADD DEFAULT ((0)) FOR [VerAct]
GO
ALTER TABLE [dbo].[Actividades] ADD DEFAULT ((1)) FOR [EstAct]
GO
ALTER TABLE [dbo].[Actividades] ADD DEFAULT ((0)) FOR [CanPlaAct]
GO
ALTER TABLE [dbo].[Actividades] ADD CONSTRAINT [DF__Actividad__MonPl__407A839F]
DEFAULT ((0.0)) FOR [MonPlaAct]
GO
ALTER TABLE [dbo].[Areas] ADD DEFAULT ((1)) FOR [EstAre]
GO
ALTER TABLE [dbo].[Detalles] ADD DEFAULT (") FOR [ObsDet]
GO
ALTER TABLE [dbo].[Detalles] ADD DEFAULT ((1)) FOR [EstDet]
GO
ALTER TABLE [dbo].[Detalles] ADD DEFAULT ((0)) FOR [VerAuxDet]
GO
ALTER TABLE [dbo].[Gestion] ADD DEFAULT ((1)) FOR [EstGes]
GO
ALTER TABLE [dbo].[Material] ADD DEFAULT ((0)) FOR [ParMat]
GO
ALTER TABLE [dbo].[Material] ADD DEFAULT ((1)) FOR [TipMat]
GO
ALTER TABLE [dbo].[Material] ADD DEFAULT ((1)) FOR [EstMat]
GO
ALTER TABLE [dbo].[Material] ADD DEFAULT ((0)) FOR [CodAux]
GO
ALTER TABLE [dbo].[Materiales] ADD DEFAULT ((0)) FOR [ParMat]
GO
ALTER TABLE [dbo].[Materiales] ADD DEFAULT ((1)) FOR [TipMat]
GO
ALTER TABLE [dbo].[Materiales] ADD DEFAULT ((1)) FOR [EstMat]
GO
ALTER TABLE [dbo].[Materiales] ADD DEFAULT ((0)) FOR [CodAux]

```

```

GO
ALTER TABLE [dbo].[Metas] ADD DEFAULT (") FOR [UniMet]
GO
ALTER TABLE [dbo].[Metas] ADD CONSTRAINT [DF__Metas__MonPlaMet__4AF81212]
DEFAULT ((0.0)) FOR [MonPlaMet]
GO
ALTER TABLE [dbo].[Metas] ADD DEFAULT ((1)) FOR [EstMet]
GO
ALTER TABLE [dbo].[Seguimiento] ADD DEFAULT ((1)) FOR [EstSeg]
GO
ALTER TABLE [dbo].[SolComMat] ADD DEFAULT ((1)) FOR [EstSolComMat]
GO
ALTER TABLE [dbo].[TechoPresupuestario] ADD CONSTRAINT
[DF__TechoPres__EstTe__7F6BDA51] DEFAULT ((1)) FOR [EstTecPre]
GO
ALTER TABLE [dbo].[TiempoEjecucion] ADD DEFAULT ((1)) FOR [EstTieEje]
GO
ALTER TABLE [dbo].[Actividades] WITH CHECK ADD FOREIGN KEY([CodPer])
REFERENCES [dbo].[Personas] ([CodPer])
GO
ALTER TABLE [dbo].[Actividades] WITH CHECK ADD FOREIGN KEY([CodTecPre])
REFERENCES [dbo].[TechoPresupuestario] ([CodTecPre])
GO
ALTER TABLE [dbo].[Cotizacion] WITH CHECK ADD CONSTRAINT [CodSolCotizacion]
FOREIGN KEY([CodSolCot])
REFERENCES [dbo].[SolicitudCotizacion] ([CodSolCot])
GO
ALTER TABLE [dbo].[Cotizacion] CHECK CONSTRAINT [CodSolCotizacion]
GO
ALTER TABLE [dbo].[Detalles] WITH CHECK ADD FOREIGN KEY([CodMet])
REFERENCES [dbo].[Metas] ([CodMet])
GO
ALTER TABLE [dbo].[Empresas] WITH CHECK ADD FOREIGN KEY([CodTipEmp])
REFERENCES [dbo].[TiposEmpresas] ([CodTipEmp])
GO
ALTER TABLE [dbo].[MateMedi] WITH CHECK ADD CONSTRAINT [CodMedi] FOREIGN
KEY([CodMed])
REFERENCES [dbo].[Medidas] ([CodMed])
GO
ALTER TABLE [dbo].[MateMedi] CHECK CONSTRAINT [CodMedi]
GO
ALTER TABLE [dbo].[Material] WITH CHECK ADD FOREIGN KEY([CodUni])
REFERENCES [dbo].[Unidades] ([CodUni])
GO
ALTER TABLE [dbo].[Materiales] WITH CHECK ADD FOREIGN KEY([CodUni])
REFERENCES [dbo].[Unidades] ([CodUni])

```

```

GO
ALTER TABLE [dbo].[MenRol] WITH CHECK ADD CONSTRAINT [FK_MenusRoles_Menus]
FOREIGN KEY([CodMen])
REFERENCES [dbo].[Menus] ([CodMen])
GO
ALTER TABLE [dbo].[MenRol] CHECK CONSTRAINT [FK_MenusRoles_Menus]
GO
ALTER TABLE [dbo].[MenRol] WITH CHECK ADD CONSTRAINT [FK_MenusRoles_Roles]
FOREIGN KEY([CodRol])
REFERENCES [dbo].[Roles] ([CodRol])
GO
ALTER TABLE [dbo].[MenRol] CHECK CONSTRAINT [FK_MenusRoles_Roles]
GO
ALTER TABLE [dbo].[Metas] WITH CHECK ADD FOREIGN KEY([CodAct])
REFERENCES [dbo].[Actividades] ([CodAct])
GO
ALTER TABLE [dbo].[PedidoCompras] WITH CHECK ADD CONSTRAINT [CodSolCoti] FOREIGN
KEY([CodSolCot])
REFERENCES [dbo].[SolicitudCotizacion] ([CodSolCot])
GO
ALTER TABLE [dbo].[PedidoCompras] CHECK CONSTRAINT [CodSolCoti]
GO
ALTER TABLE [dbo].[PerAre] WITH CHECK ADD CONSTRAINT [Codareas] FOREIGN
KEY([CodAre])
REFERENCES [dbo].[Areas] ([CodAre])
GO
ALTER TABLE [dbo].[PerAre] CHECK CONSTRAINT [Codareas]
GO
ALTER TABLE [dbo].[PerAre] WITH CHECK ADD CONSTRAINT [codperson] FOREIGN
KEY([CodPer])
REFERENCES [dbo].[Personas] ([CodPer])
GO
ALTER TABLE [dbo].[PerAre] CHECK CONSTRAINT [codperson]
GO
ALTER TABLE [dbo].[Personas] WITH CHECK ADD CONSTRAINT [FK__Personas__CodEmp]
FOREIGN KEY([CodEmp])
REFERENCES [dbo].[Empresas] ([CodEmp])
GO
ALTER TABLE [dbo].[Personas] CHECK CONSTRAINT [FK__Personas__CodEmp]
GO
ALTER TABLE [dbo].[Personas] WITH CHECK ADD CONSTRAINT
[FK_Personas_TipoDocumento] FOREIGN KEY([CodTipDoc])
REFERENCES [dbo].[TipoDocumento] ([CodTipDoc])
GO
ALTER TABLE [dbo].[Personas] CHECK CONSTRAINT [FK_Personas_TipoDocumento]
GO

```

```

ALTER TABLE [dbo].[Personas] WITH CHECK ADD CONSTRAINT
[FK_Personas_TiposPersonas] FOREIGN KEY([CodTipPer])
REFERENCES [dbo].[TiposPersonas] ([CodTipPer])
GO
ALTER TABLE [dbo].[Personas] CHECK CONSTRAINT [FK_Personas_TiposPersonas]
GO
ALTER TABLE [dbo].[Procesos] WITH CHECK ADD FOREIGN KEY([CodTipPro])
REFERENCES [dbo].[TipoProcesos] ([CodTipPro])
GO
ALTER TABLE [dbo].[Seguimiento] WITH CHECK ADD FOREIGN KEY([CodDet])
REFERENCES [dbo].[Detalles] ([CodDet])
GO
ALTER TABLE [dbo].[Seguimiento] WITH CHECK ADD FOREIGN KEY([CodMet])
REFERENCES [dbo].[Metas] ([CodMet])
GO
ALTER TABLE [dbo].[SolComMat] WITH CHECK ADD CONSTRAINT [CodSolCompra] FOREIGN
KEY([CodSolCom])
REFERENCES [dbo].[SolicitudCompra] ([CodSolCom])
GO
ALTER TABLE [dbo].[SolComMat] CHECK CONSTRAINT [CodSolCompra]
GO
ALTER TABLE [dbo].[SolicitudCompra] WITH CHECK ADD CONSTRAINT [CodArea] FOREIGN
KEY([CodAre])
REFERENCES [dbo].[Areas] ([CodAre])
GO
ALTER TABLE [dbo].[SolicitudCompra] CHECK CONSTRAINT [CodArea]
GO
ALTER TABLE [dbo].[SolicitudCotizacion] WITH CHECK ADD CONSTRAINT [CodSolCom]
FOREIGN KEY([CodSolCom])
REFERENCES [dbo].[SolicitudCompra] ([CodSolCom])
GO
ALTER TABLE [dbo].[SolicitudCotizacion] CHECK CONSTRAINT [CodSolCom]
GO
ALTER TABLE [dbo].[Submenus] WITH CHECK ADD CONSTRAINT [FK_Submenus_Menus]
FOREIGN KEY([CodMen])
REFERENCES [dbo].[Menus] ([CodMen])
GO
ALTER TABLE [dbo].[Submenus] CHECK CONSTRAINT [FK_Submenus_Menus]
GO
ALTER TABLE [dbo].[SubPro] WITH CHECK ADD CONSTRAINT [FK_SubPro_Procesos]
FOREIGN KEY([CodPro])
REFERENCES [dbo].[Procesos] ([CodPro])
GO
ALTER TABLE [dbo].[SubPro] CHECK CONSTRAINT [FK_SubPro_Procesos]
GO

```

```

ALTER TABLE [dbo].[SubPro] WITH CHECK ADD CONSTRAINT [FK_SubPro_Submenus]
FOREIGN KEY([CodSub])
REFERENCES [dbo].[Submenus] ([CodSub])
GO
ALTER TABLE [dbo].[SubPro] CHECK CONSTRAINT [FK_SubPro_Submenus]
GO
ALTER TABLE [dbo].[TechoPresupuestario] WITH CHECK ADD CONSTRAINT [CodAre]
FOREIGN KEY([CodAre])
REFERENCES [dbo].[Areas] ([CodAre])
GO
ALTER TABLE [dbo].[TechoPresupuestario] CHECK CONSTRAINT [CodAre]
GO
ALTER TABLE [dbo].[TechoPresupuestario] WITH CHECK ADD FOREIGN KEY([GesGes])
REFERENCES [dbo].[Gestion] ([GesGes])
GO
ALTER TABLE [dbo].[TiempoEjecucion] WITH CHECK ADD FOREIGN KEY([CodMet])
REFERENCES [dbo].[Metas] ([CodMet])
GO
ALTER TABLE [dbo].[Usuarios] WITH CHECK ADD CONSTRAINT
[FK_DatosUsuarios_Personas] FOREIGN KEY([CodPer])
REFERENCES [dbo].[Personas] ([CodPer])
GO
ALTER TABLE [dbo].[Usuarios] CHECK CONSTRAINT [FK_DatosUsuarios_Personas]
GO
ALTER TABLE [dbo].[UsuRol] WITH CHECK ADD CONSTRAINT [FK_UsuRol_Roles] FOREIGN
KEY([CodRol])
REFERENCES [dbo].[Roles] ([CodRol])
GO
ALTER TABLE [dbo].[UsuRol] CHECK CONSTRAINT [FK_UsuRol_Roles]
GO
ALTER TABLE [dbo].[UsuRol] WITH CHECK ADD CONSTRAINT [FK_UsuRol_Usuarios]
FOREIGN KEY([CodUsu])
REFERENCES [dbo].[Usuarios] ([CodUsu])
GO
ALTER TABLE [dbo].[UsuRol] CHECK CONSTRAINT [FK_UsuRol_Usuarios]
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Codigo de Area' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Areas',
@level2type=N'COLUMN',@level2name=N'CodAre'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Nombre de Area'
, @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Areas',
@level2type=N'COLUMN',@level2name=N'NomAre'
GO

```

```

EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Estado de Area;
0=Inactivo, 1=Activo', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Areas',
@level2type=N'COLUMN',@level2name=N'EstAre'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Codigo de
Detalle', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Detalles',
@level2type=N'COLUMN',@level2name=N'CodDet'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Descripcion de
Detalle', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Detalles',
@level2type=N'COLUMN',@level2name=N'DesRecDet'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Unidad de
Detalle', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Detalles',
@level2type=N'COLUMN',@level2name=N'UniRecDet'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Cantidad de
Detalle', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Detalles',
@level2type=N'COLUMN',@level2name=N'CanRecDet'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Precio Unitario
de Detalle', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Detalles',
@level2type=N'COLUMN',@level2name=N'PreUniRecDet'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Costo del
recurso', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Detalles',
@level2type=N'COLUMN',@level2name=N'CosRecDet'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Partida
presupuestaria del recurso', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Detalles',
@level2type=N'COLUMN',@level2name=N'ParDet'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Observacion',
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Detalles',
@level2type=N'COLUMN',@level2name=N'ObsDet'
GO

```

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Tipo de Recurso'  
, @level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'Detalles',  
@level2type=N'COLUMN',@level2name=N'TipRecDet'  
GO  
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Estado del  
Detalle', @level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'Detalles',  
@level2type=N'COLUMN',@level2name=N'EstDet'  
GO  
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Codigo de Meta'  
, @level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'Detalles',  
@level2type=N'COLUMN',@level2name=N'CodMet'  
GO
```

II.1.1.3.10. Diccionario de Datos

Tabla: Actividades

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodAct	Si	INTEGER	Si	Codigo de Actividad
DesAct	No	VARCHAR(250)	Si	Descripcion de Actividad
FecAct	No	DATE	Si	Fecha de Actividad
VerAct	No	INTEGER	Si	Version de Actividad
EstAct	No	INTEGER	Si	Estado de Actividad
CodPer	No	INTEGER	Si	Codigo de Persona
CodTecPre	No	INTEGER	Si	Codigo de Techo Presupuestario
CanPlaAct	No	INTEGER	Si	Cantidad Planificada de Actividad
MonPlaAct	No	FLOAT(8)	Si	Monto Plnificado de Actividad

Cuadro 28Tabla: Actividades

Tabla: Areas

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodAre	Si	INTEGER	Si	Codigo de Area
NomAre	No	VARCHAR(50)	Si	Nombre de Area
EstAre	No	BIT	Si	Estado de Area; 0=Inactivo, 1=Activo

Cuadro 29Tabla: Areas

Tabla: Detalles

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodDet	Si	INTEGER	Si	Codigo de Detalle
DesRecDet	No	VARCHAR(150)	Si	Descripcion de Detalle
UniRecDet	No	VARCHAR(10)	Si	Unidad de Detalle
CanRecDet	No	INTEGER	Si	Cantidad de Detalle
PreUniRecDet	No	FLOAT(8)	Si	Precio Unitario de Detalle
CosRecDet	No	FLOAT(8)	No	Costo del recurso
ParDet	No	INTEGER	Si	Partida presupuestaria del recurso
ObsDet	No	VARCHAR(150)	No	Observacion

TipRecDet	No	INTEGER	Si	Tipo de Recurso
EstDet	No	INTEGER	Si	Estado del Detalle
CodMet	No	INTEGER	Si	Codigo de Meta
VerAuxDet	No	INTEGER	Si	Version de Detalle
CodAux	No	VARCHAR(12)	Si	Codigo Auxiliar de Material

Cuadro 30Tabla: Detalles

Tabla: Empresas

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
Codigomp	Si	INTEGER	Si	Codigo de Empresa
NomEmp	No	VARCHAR(100)	No	Nombre de Empresa
NitEmp	No	VARCHAR(20)	No	Numero Identificacion Tributaria de Empresa
DirEmp	No	VARCHAR(50)	No	Direccion de Empresa
EmaEmp	No	VARCHAR(30)	No	Correo electronico de Empresa
TelEmp	No	VARCHAR(30)	No	Telefono de Empresa
FaxEmp	No	VARCHAR(30)	No	Fax de Empresa
CodTipEmp	No	INTEGER	Si	Codigo tipo de Empresa
AliEmp	No	VARCHAR(15)	Si	Alias de Empresa
EstEmp	No	BIT	Si	Estado de Empresa

Cuadro 31Tabla: Empresas

Tabla: Gestion

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
GesGes	Si	VARCHAR(4)	Si	Codigo de Gestion
EstGes	No	BIT	Si	Estado de Gestion

Cuadro 32Tabla: Gestion

Tabla: Materiales

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodMat	Si	NUMERIC(5)	Si	Codigo Material
NomMat	No	VARCHAR(80)	Si	Nombre de Material
ParMat	No	INTEGER	Si	Partida Presupuestaria de Material
CodUni	No	INTEGER	Si	Codigo de Unidad
CosMat	No	FLOAT(8)	Si	Costo del Material
TipMat	No	NUMERIC(1)	Si	Tipo de Material

EstMat	No	NUMERIC(1)	Si	Esatdo de Material
CodAux	No	VARCHAR(12)	Si	Codigo Auxiliar de Material

Cuadro 33Tabla: Materiales

Tabla: MenRol

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodMen	Si	INTEGER	Si	Codigo Menus
CodRol	Si	INTEGER	Si	Codigo Rol

Cuadro 34Tabla: MenRol

Tabla: Menus

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodMen	Si	INTEGER	Si	Codigo Menus
NomMen	No	VARCHAR(50)	Si	Nombre Menus
EstMen	No	BIT	Si	Estado Menus

Cuadro 35Tabla: Menus

Tabla: Metas

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodMet	Si	INTEGER	Si	Codigo de metas
DesMet	No	VARCHAR(250)	Si	Descripcion de metas
UniMet	No	VARCHAR(10)	No	Unidad de Metas
CanPlaMet	No	INTEGER	Si	Cantidad Planificada de Metas
MonPlaMet	No	FLOAT(8)	No	Monto Planificado de Metas
FecIniMet	No	DATE	Si	Fecha Inicial de Metas
FecFinMet	No	DATE	Si	FechaFinal de Metas
EstMet	No	INTEGER	Si	Esatdo de Metas
CodAct	No	INTEGER	Si	Codigo de Actividad

Cuadro 36Tabla: Metas

Tabla: PerAre

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodPer	Si	INTEGER	Si	Codigo de Persona
CodAre	Si	INTEGER	Si	Codigo de Area

Cuadro 37Tabla: PerAre

Tabla: Personas

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodPer	Si	INTEGER	Si	Codigo de Actividad
NomPer	No	VARCHAR(50)	Si	Nombre de Actividad
ApePatPer	No	VARCHAR(30)	Si	Apellido Paterno
ApeMatPer	No	VARCHAR(30)	No	Apellido Materno
NumDocPer	No	VARCHAR(20)	No	Numero de documento
FecNacPer	No	VARCHAR(15)	No	Fecha de nacimiento
SexPer	No	INTEGER	No	Sexo de la persona (masculino/femenino)
EstCivPer	No	INTEGER	No	Estado Civil
EstPer	No	INTEGER	No	
NumTelPer	No	VARCHAR(20)	No	Numero telefonico
NumCelPer	No	VARCHAR(20)	No	Numero celular
DirPer	No	VARCHAR(100)	No	Direccion de la Persona
EmaPer	No	VARCHAR(50)	No	Correo Electronico
Codigomp	No	INTEGER	No	Codigo de Empresa
CodTipDoc	No	INTEGER	No	Codigo de Tipo de documento
CodTipPer	No	INTEGER	No	

Cuadro 38Tabla: Personas

Tabla: Procesos

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodPro	Si	INTEGER	Si	Codigo Proceso
UrlPro	No	VARCHAR(150)	Si	Direccion URL
DesPro	No	VARCHAR(100)	No	Descripcion Proceso
EstPro	No	BIT	Si	Estado Proceso
CodTipPro	No	INTEGER	No	Codigo tipo de Proceso
NomPro	No	VARCHAR(50)	No	Nombre de Proceso

Cuadro 39Tabla: Procesos

Tabla: Roles

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodRol	Si	INTEGER	Si	Codigo de Rol
NomRol	No	VARCHAR(50)	Si	Nombre de Rol
EstRol	No	BIT	Si	Estado de Rol

Cuadro 40Tabla: Roles

Tabla: Seguimiento

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodSeg	Si	NUMERIC(5)	Si	Codigo de Seguimiento
FecSeg	No	DATE	Si	Fecha de Seguimiento
ObsSeg	No	VARCHAR(1000)	No	Observcion de Seguimiento
EstSeg	No	NUMERIC(1)	Si	Estado de Seguimiento
CodMet	No	INTEGER	Si	Codigo Metas
CodDet	No	INTEGER	Si	Codigo de Detalle
CanSeg	No	INTEGER	Si	Cantidad de Material de Seguimiento
PreSeg	No	FLOAT(8)	Si	Precio Unitario de Material de Seguimiento
CosSeg	No	FLOAT(8)	Si	Costo de Material de Seguimiento

Cuadro 41Tabla: Seguimiento

Tabla: Submenus

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodSub	Si	INTEGER	Si	Codigo SubMenus
NomSub	No	VARCHAR(50)	Si	Nombre de SubMenus
EstSub	No	BIT	Si	Estado de SubMenus
CodMen	No	INTEGER	No	Codigo de Menus
url	No	VARCHAR(150)	No	Direccion URL del Menu

Cuadro 42Tabla: Submenus

Tabla: SubPro

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodSub	Si	INTEGER	Si	Codigo Submenus
CodPro	Si	INTEGER	Si	Codigo Proceso

Cuadro 43Tabla: SubPro

Tabla: TechoPresupuestario

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodTecPre	Si	INTEGER	Si	Codigo de Techo Presupuestario
MonTecPre	No	INTEGER	Si	Monto de Techo Presupuestario
CodAre	No	INTEGER	Si	Codigo de Area
GesGes	No	VARCHAR(4)	Si	Codigo de Gestion
EstTecPre	No	NUMERIC(1)	Si	Estado de Techo Presupuestario

Cuadro 44Tabla: TechoPresupuestario

Tabla: TiempoEjecucion

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodTieEje	Si	INTEGER	Si	Codigo de Tiempo de Ejecucion
EneTieEje	No	BIT	No	Mes de Enero de Tiempo de Ejecucion
FebTieEje	No	BIT	No	Mes de Febrero de Tiempo de Ejecucion
MarTieEje	No	BIT	No	Mes de Marzo de Tiempo de Ejecucion
AbrTieEje	No	BIT	No	Mes de Abril de Tiempo de Ejecucion
MayTieEje	No	BIT	No	Mes de Mayo de Tiempo de Ejecucion
JunTieEje	No	BIT	No	Mes de Junio de Tiempo de Ejecucion
JulTieEje	No	BIT	No	Mes de Julio de Tiempo de Ejecucion
AgoTieEje	No	BIT	No	Mes de Agosto de Tiempo de Ejecucion
SepTieEje	No	BIT	No	Mes de Septiembre de Tiempo de Ejecucion

OctTieEje	No	BIT	No	Mes de Octubre de Tiempo de Ejecucion
NovTieEje	No	BIT	No	Mes de noviembre de Tiempo de Ejecucion
DicTieEje	No	BIT	No	Mes de Diciembre de Tiempo de Ejecucion
CodMet	No	INTEGER	Si	Codigo de Metas
EstTieEje	No	INTEGER	No	Esatdo de Tiempo de Ejecucion

Cuadro 45Tabla: TiempoEjecucion

Tabla: TipoDocumento

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodTipDoc	Si	INTEGER	Si	Codigo tipo de documento
NomTipDoc	No	VARCHAR(50)	Si	Nombre de Tipo de documento
AliNomDoc	No	VARCHAR(3)	Si	Alias Tipo de documento
EstTipDoc	No	BIT	Si	Estado Tipo de Documento

Cuadro 46Tabla: TipoDocumento

Tabla: TipoMateriales

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodTipMat	Si	INTEGER	Si	
NomTipMat	No	VARCHAR(30)	Si	
EstTipMat	No	INTEGER	Si	
AliTipMat	No	VARCHAR(20)	Si	

Cuadro 47Tabla: TipoMateriales

Tabla: TipoProcesos

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodTipPro	Si	INTEGER	Si	Codigo Tipo de Proceso
NomTipPro	No	VARCHAR(50)	Si	Nombre Tipo de Proceso
EstTipPro	No	BIT	Si	Estado Tipo de Proceso

Cuadro 48Tabla: TipoProcesos

Tabla: TiposEmpresas

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodTipEmp	Si	INTEGER	Si	Codigo Tipo de Empresa
NomTipEmp	No	VARCHAR(30)	Si	Nombre Tipo de Empresa
AliTipEmp	No	VARCHAR(15)	Si	Alias Tipo de Empresa
EstTipEmp	No	BIT	Si	Estado Tipo de Empresa

Cuadro 49Tabla: TiposEmpresas

Tabla: TiposPersonas

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodTipPer	Si	INTEGER	Si	Codigo de Tipo de persona
NomTipPer	No	VARCHAR(30)	Si	Nombre de Tipo de persona
EstTipPer	No	BIT	Si	Estado Tipo de persona

Cuadro 50Tabla: TiposPersonas

Tabla: Unidades

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodUni	Si	INTEGER	Si	Codigo de Unidad
NomUni	No	VARCHAR(30)	No	Nombre de Unidad
AliUni	No	VARCHAR(10)	No	Alias de Unidad
EstUni	No	INTEGER	No	Esatdo de Unidad

Cuadro 51 Tabla: Unidades

Tabla: Usuarios

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodUsu	Si	INTEGER	Si	Codigo Usuario
CodPer	No	INTEGER	Si	Codigo de Persona
UsuUsu	No	VARCHAR(50)	Si	Usuario
ClaUsu	No	VARCHAR(50)	Si	Clave
EstUsu	No	INTEGER	No	Estado de Usuario

Cuadro 52Tabla: Usuarios

Tabla: UsuRol

Atributo:

Nombre de Columna	Clave Primaria	Tipo de Datos	Nulos	Descripcion
CodRol	Si	INTEGER	Si	Codigo de Rol
CodUsu	Si	INTEGER	Si	Codigo de Usuario

Cuadro 53Tabla: UsuRol

Procedimiento: adicionarDetalle

Nombre	adicionarDetalle
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[adicionarDetalle] @DesRecDet as VARCHAR(150), @UniRecDet as VARCHAR(10), @CanRecDet as int, @PreUniRecDet as float, @ParDet as int, @ObsDet as VARCHAR(150), @TipRecDet as int, @CodAux as VARCHAR(12), @CodMet as int, @CodAct as int AS DECLARE @cod INT DECLARE @costo FLOAT BEGIN TRAN BEGIN SET @cod=(SELECT MAX(CodDet) FROM Detalles) SET @cod=ISNULL(@cod, 0)+1 set @costo=@CanRecDet*@PreUniRecDet BEGIN TRY insert into detalles(CodDet, DesRecDet, UniRecDet, CanRecDet, PreUniRecDet, CosRecDet, ParDet, ObsDet, TipRecDet, CodAux, CodMet) VALUES(@cod, @DesRecDet, @UniRecDet, @CanRecDet, @PreUniRecDet, CAST(@costo AS DECIMAL(18, 2)), @ParDet, @ObsDet, @TipRecDet , @CodAux, @CodMet) update Metas set MonPlaMet=(SELECT sum(CosRecDet) from Detalles where CodMet=@CodMet AND EstDet=1) WHERE CodMet=@CodMet update Actividades set MonPlaAct=(</pre>

	<pre> SELECT sum(MonPlaMet) from Metas where CodAct=@CodAct and EstMet=1) WHERE CodAct=@CodAct COMMIT END TRY BEGIN CATCH ROLLBACK INSERT INTO LogError(descripcion) VALUES(ERROR_MESSAGE()) END CATCH END </pre>
--	---

Procedimiento: adicionarMaterial	
Nombre	adicionarMaterial
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[adicionarMaterial] @NomMat as VARCHAR(80), @CodUni as int, @CosMat as float, @TipMat as NUMERIC, @ParMat as int, @CodAux as VARCHAR(12) AS DECLARE @codmat INT BEGIN SET @codmat=(SELECT MAX(CodMat) FROM Materiales) SET @codmat=ISNULL(@codmat, 0)+1 insert into materiales(CodMat, NomMat, CodUni, CosMat, TipMat, ParMat, CodAux) VALUES(@codmat, @NomMat, @CodUni, @CosMat, @TipMat, @ParMat, @CodAux) END </pre>

Procedimiento: adicionarMaterialFun	
Nombre	adicionarMaterialFun
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[adicionarMaterial] @NomMat as VARCHAR(80), @CodUni as int, @CosMat as float, @EstMat as NUMERIC, @TipMat as NUMERIC AS DECLARE @codmat INT BEGIN SET @codmat=(SELECT MAX(CodMat) FROM Materiales) SET @codmat=ISNULL(@codmat, 0)+1 insert into materiales(CodMat, NomMat, CodUni, CosMat, </pre>

	EstMat, TipMat) VALUES(@codmat, @NomMat, @CodUni, @CosMat, @EstMat, @TipMat) END
--	---

Procedimiento: adicionarMeta	
Nombre	adicionarMeta
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[adicionarMeta] @DesMet as VARCHAR(250), @CanPlaMet as int, @FecIniMet as DATE, @FecFinMet as DATE, @CodAct as int, @Ene as BIT, @Feb as BIT, @Mar as BIT, @Abr as BIT, @May as BIT, @Jun as BIT, @Jul as BIT, @Ago as BIT, @Sep as BIT, @Oct as BIT, @Nov as BIT, @Dic as BIT AS DECLARE @codmet INT DECLARE @codtie INT BEGIN TRAN BEGIN SET @codmet=(SELECT MAX(CodMet) FROM Metas) SET @codmet=ISNULL(@codmet, 0)+1 SET @codtie=(SELECT MAX(CodTieEje) FROM TiempoEjecucion) SET @codtie=ISNULL(@codtie, 0)+1 BEGIN TRY insert into Metas(CodMet, DesMet, CanPlaMet, FecIniMet, FecFinMet, CodAct) VALUES(@codmet, @DesMet, @CanPlaMet, @FecIniMet, @FecFinMet, @CodAct) insert into TiempoEjecucion(CodTieEje, EneTieEje, FebTieEje, MarTieEje, AbrTieEje, MayTieEje, JunTieEje, JulTieEje, AgoTieEje, SepTieEje, OctTieEje, NovTieEje, DicTieEje, CodMet) values(@codtie, @Ene, @Feb, @Mar, @Abr, @May, @Jun, @Jul, @Ago, @Sep, @Oct, @Nov, @Dic, @codmet) COMMIT </pre>

	END TRY BEGIN CATCH ROLLBACK INSERT INTO LogError(descripcion) VALUES(ERROR_MESSAGE()) END CATCH END
--	--

Procedimiento: adicionarMontoTecho	
Nombre	adicionarMontoTecho
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[adicionarMontoTecho] @CodAre as int, @Monto as int, @GesGes as VARCHAR(4) AS DECLARE @cod INT BEGIN TRAN BEGIN SET @cod=(SELECT MAX(CodTecPre) FROM TechoPresupuestario) SET @cod=ISNULL(@cod, 0)+1 insert into TechoPresupuestario(CodTecPre, CodAre, MonTecPre, GesGes) VALUES(@cod, @CodAre, @Monto, @GesGes) COMMIT END </pre>

Procedimiento: adicionarObjetivo	
Nombre	adicionarObjetivo
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[adicionarObjetivo] @DesAct As VARCHAR(250) , @FecAct As DATE , @CodPer AS int, @CodTecPre As INT AS DECLARE @cod INT BEGIN TRAN BEGIN SET @cod=(SELECT MAX(CodAct) FROM Actividades) SET @cod=ISNULL(@cod, 0)+1 BEGIN TRY insert into Actividades(CodAct, DesAct, FecAct, CodPer, CodTecPre) values(@cod, @DesAct, @FecAct, @CodPer, @CodTecPre) COMMIT END TRY </pre>

	BEGIN CATCH ROLLBACK INSERT INTO LogError(descripcion) VALUES(ERROR_MESSAGE()) END CATCH END
--	---

Procedimiento: adicionarSeguimiento	
Nombre	adicionarSeguimiento
Codigo	CREATE PROCEDIMIENTO [dbo].[adicionarSeguimiento] @CodDet int, @CanSeg int, @PreSeg float, @ObsSeg VARCHAR(150), @CodMet int, @fecha DATE, @codact int AS DECLARE @cod INT DECLARE @costo FLOAT BEGIN SET @cod=(SELECT MAX(CodSeg) FROM Seguimiento) SET @cod=ISNULL(@cod, 0)+1 set @costo=@CanSeg*@PreSeg insert into Seguimiento(CodSeg, CanSeg, PreSeg, CosSeg, ObsSeg, FecSeg, CodMet, CodDet) VALUES(@cod, @CanSeg, @PreSeg, CAST(@costo AS DECIMAL(18, 2)), @ObsSeg, @fecha, @CodMet, @CodDet) END

Procedimiento: aprobarDetalle	
Nombre	aprobarDetalle
Codigo	CREATE PROCEDIMIENTO [dbo].[aprobarDetalle] @coddet int, @ver int AS BEGIN update Detalles set VerAuxDet=@ver WHERE CodDet=@CodDet END

Procedimiento: countMeta	
Nombre	countMeta
Codigo	CREATE PROCEDIMIENTO [dbo].[countMeta] @CodMet as int AS BEGIN

	SELECT COUNT(CodMet) from Detalles WHERE (CodMet=@CodMet and EstDet=1) END
--	---

Procedimiento: countObjetivo	
Nombre	countObjetivo
Codigo	CREATE PROCEDIMIENTO [dbo].[countObjetivo] @CodAct as int AS BEGIN SELECT COUNT(CodAct) from Metas WHERE CodAct=@CodAct and EstMet=1 END

Procedimiento: enviar	
Nombre	enviar
Codigo	CREATE PROCEDIMIENTO [dbo].[enviar] @CodAct as int, @est int AS BEGIN update Actividades set VerAct=@est WHERE CodAct=@CodAct END

Procedimiento: fn_diagramobjects	
Nombre	fn_diagramobjects
Codigo	CREATE FUNCTION [dbo].[fn_diagramobjects]() RETURNS int WITH EXECUTE AS N'dbo' AS BEGIN declare @id_upgraddiagrams int declare @id_sysdiagrams int declare @id_helpdiagrams int declare @id_helpdiagramdefinition int declare @id_creatediagram int declare @id_reNombrediagram int declare @id_alterdiagram int declare @id_dropdiagram int declare @InstalledObjects int select @InstalledObjects = 0 select @id_upgraddiagrams =

	<pre> object_id(N'dbo.sp_upgraddiagrams'), @id_sysdiagrams = object_id(N'dbo.sysdiagrams'), @id_helpdiagrams = object_id(N'dbo.sp_helpdiagrams'), @id_helpdiagramdefinition = object_id(N'dbo.sp_helpdiagramdefinition'), @id_creatediagram = object_id(N'dbo.sp_creatediagram'), @id_reNombrediagram = object_id(N'dbo.sp_reNombrediagram'), @id_alterdiagram = object_id(N'dbo.sp_alterdiagram'), @id_dropdiagram = object_id(N'dbo.sp_dropdiagram') if @id_upgraddiagrams is Nulos select @InstalledObjects = @InstalledObjects + 1 if @id_sysdiagrams is Nulos select @InstalledObjects = @InstalledObjects + 2 if @id_helpdiagrams is Nulos select @InstalledObjects = @InstalledObjects + 4 if @id_helpdiagramdefinition is Nulos select @InstalledObjects = @InstalledObjects + 8 if @id_creatediagram is Nulos select @InstalledObjects = @InstalledObjects + 16 if @id_reNombrediagram is Nulos select @InstalledObjects = @InstalledObjects + 32 if @id_alterdiagram is Nulos select @InstalledObjects = @InstalledObjects + 64 if @id_dropdiagram is Nulos select @InstalledObjects = @InstalledObjects + 128 return @InstalledObjects END </pre>
--	--

Procedimiento: incluirMaterial	
Nombre	incluirMaterial
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[incluirMaterial] @CodMat as int AS BEGIN update Materiales set EstMat=1 WHERE CodMat=@CodMat END </pre>

Procedimiento: iniciarsesion	
Nombre	iniciarsesion
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[iniciarsesion] @UsuUsu as VARCHAR(50), @ClaUsu as VARCHAR(50) AS BEGIN </pre>

	<pre> select Personas.CodPer, Personas.NomPer, Personas.ApePatPer, Personas.ApeMatPer, Personas.NumDocPer, Personas.FecNacPer, Personas.SexPer, Personas.EstCivPer, Personas.EstPer, Personas.NumCelPer, Personas.NumTelPer, Personas.DirPer, Personas.EmaPer, Personas.Codigomp, Personas.CodTipPer, Personas.CodTipDoc from Personas JOIN Usuarios on Personas.CodPer=Usuarios.CodPer where Usuarios.UsuUsu=@UsuUsu and Usuarios.ClaUsu=@ClaUsu and Usuarios.EstUsu=1 END </pre>
--	--

Procedimiento: listaArea	
Nombre	listaArea
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[listaArea] AS BEGIN SELECT CodAre, NomAre, EstAre FROM Areas WHERE EstAre=1 END </pre>

Procedimiento: listaAreaPersona	
Nombre	listaAreaPersona
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[listaAreaPersona] @codare INT AS BEGIN SELECT NomPer+' '+ApePatPer+' '+ApeMatPer as nombre, Personas.CodPer FROM Personas JOIN PerAre ON Personas.CodPer=PerAre.CodPer JOIN Areas on Areas.CodAre=PerAre.CodAre and Areas.CodAre=@codare END </pre>

Procedimiento: listaAreasTecho	
Nombre	listaAreasTecho
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[listaAreasTecho] @gestion VARCHAR(4) AS BEGIN SELECT Areas.CodAre, Areas.NomAre, ISNULL(TechoPresupuestario.MonTecPre, 0) as monto, TechoPresupuestario.EstTecPre, TechoPresupuestario.CodTecPre from Areas LEFT OUTER JOIN TechoPresupuestario on </pre>

	TechoPresupuestario.CodAre=Areas.CodAre and TechoPresupuestario.GesGes=@gestion END
--	---

Procedimiento: listaDetalleAprobado	
Nombre	listaDetalleAprobado
Codigo	CREATE PROCEDIMIENTO [dbo].[listaDetalleAprobado] @codmet INT AS BEGIN select Detalles.CodDet, DesRecDet, UniRecDet, CanRecDet, PreUniRecDet, CosRecDet, TipRecDet, ParDet, CodMet from Detalles where CodDet not in (select CodDet from Seguimiento) and EstDet=1 and CodMet=@codmet END

Procedimiento: listaGestion	
Nombre	listaGestion
Codigo	CREATE PROCEDIMIENTO [dbo].[listaGestion] AS BEGIN SELECT GesGes, EstGes FROM Gestion END

Procedimiento: listaMaterialDetalle	
Nombre	listaMaterialDetalle
Codigo	CREATE PROCEDIMIENTO [dbo].[listaMaterialDetalle] @codmet int AS BEGIN select m.CodMat, m.NomMat, m.ParMat, m.CosMat, m.tipMat, u.NomUni, m.CodAux from Unidades u JOIN Materiales m on u.CodUni=m.CodUni and EstMat=1 WHERE CodMat not in (SELECT Detalles.CodAux FROM Detalles JOIN Metas on Metas.CodMet=Detalles.CodMet and Detalles.CodMet=@codmet) GROUP BY m.CodMat, m.NomMat, m.ParMat, m.CosMat, m.tipMat, u.NomUni, m.CodAux END

Procedimiento: listaMetaSeguimiento	
Nombre	listaMetaSeguimiento
Codigo	CREATE PROCEDIMIENTO [dbo].[listaMetaSeguimiento] @CodAct as int

	AS BEGIN SELECT DISTINCT Metas.CodMet, DesMet, CanPlaMet, CONVERT(varchar(15), FecIniMet, 103) as FecIniMet, CONVERT(varchar(15), FecFinMet, 103) as FecFinMet, ISNULL(sum(Seguimiento.CosSeg), 0) as monto from Actividades JOIN Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1 and Metas.CodAct=@CodAct LEFT OUTER JOIN Seguimiento on Seguimiento.CodMet=Metas.CodMet and Seguimiento.EstSeg=1 GROUP BY Metas.CodMet, DesMet, CanPlaMet, FecIniMet, FecFinMet END
--	---

Procedimiento: listaObjetivoAprobado	
Nombre	listaObjetivoAprobado
Codigo	CREATE PROCEDIMIENTO [dbo].[listaObjetivoAprobado] @CodPer As INT, @gestion VARCHAR(4) AS BEGIN SELECT DISTINCT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct, Actividades.EstAct, Actividades.CodPer, Actividades.CodTecPre, Actividades.CanPlaAct, ISNULL(sum(Seguimiento.CosSeg), 0) as monto from TechoPresupuestario JOIN Actividades on Actividades.CodTecPre=TechoPresupuestario.CodTecPre and TechoPresupuestario.EstTecPre=1 and TechoPresupuestario.GesGes=@gestion and Actividades.CodPer=@CodPer and Actividades.VerAct=2 and Actividades.EstAct=1 JOIN Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1 LEFT OUTER JOIN Seguimiento on Seguimiento.CodMet=Metas.CodMet and Seguimiento.EstSeg=1 GROUP BY Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct, Actividades.EstAct, Actividades.CodPer, Actividades.CodTecPre, Actividades.CanPlaAct END

Procedimiento: listaObjetivoEvaluacion	
Nombre	listaObjetivoEvaluacion
Codigo	CREATE PROCEDIMIENTO [dbo].[listaObjetivoEvaluacion] @CodPer As INT,

	<pre>@gestion VARCHAR(4) AS BEGIN SELECT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct, Actividades.EstAct, Actividades.CodPer, Actividades.CodTecPre, Actividades.MonPlaAct, Actividades.CanPlaAct from TechoPresupuestario JOIN Actividades on Actividades.CodTecPre=TechoPresupuestario.CodTecPre and TechoPresupuestario.GesGes=@gestion and Actividades.CodPer=@CodPer and Actividades.VerAct!=0 END</pre>
--	--

Procedimiento: listaPersona	
Nombre	listaPersona
Codigo	<pre>CREATE PROCEDIMIENTO [dbo].[listaPersona] AS BEGIN SELECT NomPer+' '+ApePatPer+' '+ApeMatPer as nombre, CodPer FROM Personas WHERE EstPer=1 END</pre>

Procedimiento: listarDetalle	
Nombre	listarDetalle
Codigo	<pre>CREATE PROCEDIMIENTO [dbo].[listarDetalle] @CodMet as int, @est as int AS BEGIN SELECT CodMet, CodDet, DesRecDet, UniRecDet, CanRecDet, PreUniRecDet, CosRecDet, ParDet, ObsDet, TipRecDet, EstDet, VerAuxDet from Detalles where (codmet=@CodMet)and(EstDet=@est) END</pre>

Procedimiento: listarmaterial	
Nombre	listarmaterial
Codigo	<pre>CREATE PROCEDIMIENTO [dbo].[listarmaterial] AS BEGIN select m.CodMat, m.NomMat, m.ParMat, m.CosMat, m.tipMat, u.NomUni, m.CodAux from Unidades u JOIN Materiales m on u.CodUni=m.CodUni</pre>

	and EstMat=1 END
--	---------------------

Procedimiento: listarMaterialFuncionario	
Nombre	listarMaterialFuncionario
Codigo	CREATE PROCEDIMIENTO [dbo].[listarMaterialFuncionario] AS BEGIN select m.CodMat, m.NomMat, m.CosMat, m.tipMat, u.NomUni, m.ParMat, m.CodAux from Unidades u JOIN Materiales m on u.CodUni=m.CodUni where m.EstMat=3 or m.EstMat=4 END

Procedimiento: listarMeta	
Nombre	listarMeta
Codigo	CREATE PROCEDIMIENTO [dbo].[listarMeta] @CodAct as int, @Est as int AS BEGIN SELECT CodMet, DesMet, UniMet, CanPlaMet, MonPlaMet, CONVERT(varchar(15), FecIniMet, 103) as FecIniMet, CONVERT(varchar(15), FecFinMet, 103) as FecFinMet, EstMet, CodAct from Metas WHERE CodAct=@CodAct and EstMet=@Est END

Procedimiento: listarObjetivo	
Nombre	listarObjetivo
Codigo	CREATE PROCEDIMIENTO [dbo].[listarObjetivo] @CodPer As INT, @EstAct AS INT, @gestion VARCHAR(4) AS BEGIN SELECT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct, Actividades.EstAct, Actividades.CodPer, Actividades.CodTecPre, Actividades.MonPlaAct, Actividades.CanPlaAct from Gestion JOIN TechoPresupuestario on TechoPresupuestario.GesGes=Gestion.GesGes and TechoPresupuestario.GesGes=@gestion JOIN Actividades on Actividades.CodTecPre=TechoPresupuestario.CodTecPre and Actividades.CodPer=@CodPer and

	Actividades.EstAct=@EstAct
	END

Procedimiento: listarTechoPresupuestario	
Nombre	listarTechoPresupuestario
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[listarTechoPresupuestario] @Gestion VARCHAR(4) AS BEGIN SELECT TechoPresupuestario.CodAre as CodAre, Areas.NomAre, TechoPresupuestario.MonTecPre, TechoPresupuestario.GesGes, TechoPresupuestario.EstTecPre FROM Areas JOIN TechoPresupuestario on Areas.CodAre=TechoPresupuestario.CodAre WHERE TechoPresupuestario.GesGes=@Gestion and TechoPresupuestario.EstTecPre=1 or TechoPresupuestario.EstTecPre=2 END </pre>

Procedimiento: listarTiempoEjecucion	
Nombre	listarTiempoEjecucion
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[listarTiempoEjecucion] @CodAct as int, @Est as int AS BEGIN SELECT CodTieEje, CASE WHEN EneTieEje=1 THEN 'X' ELSE ' ' END as EneTieEje, CASE WHEN FebTieEje=1 THEN 'X' ELSE ' ' END as FebTieEje, CASE WHEN MarTieEje=1 THEN 'X' ELSE ' ' END as MarTieEje, CASE WHEN AbrTieEje=1 THEN 'X' ELSE ' ' END as AbrTieEje, CASE WHEN MayTieEje=1 THEN 'X' ELSE ' ' END as MayTieEje, CASE WHEN JunTieEje=1 THEN 'X' ELSE ' ' END as JunTieEje, CASE WHEN JulTieEje=1 THEN 'X' ELSE ' ' END as JulTieEje, CASE WHEN AgoTieEje=1 THEN 'X' ELSE ' ' END as AgoTieEje, CASE WHEN SepTieEje=1 THEN 'X' ELSE ' ' END as SepTieEje, CASE WHEN OctTieEje=1 THEN 'X' ELSE ' ' END as OctTieEje, CASE WHEN NovTieEje=1 THEN 'X' ELSE ' ' END as NovTieEje, CASE WHEN DicTieEje=1 THEN 'X' ELSE ' ' END as DicTieEje, DesMet, CanPlaMet, MonPlaMet, Metas.CodMet FROM Metas JOIN TiempoEjecucion on Metas.CodMet=TiempoEjecucion.CodMet and Metas.CodAct=@CodAct and EstTieEje=@Est END </pre>

Procedimiento: listarUnidad	
Nombre	listarUnidad
Codigo	<pre>CREATE PROCEDIMIENTO [dbo].[listarUnidad] AS BEGIN SELECT CodUni, NomUni, AliUni FROM Unidades END</pre>

Procedimiento: listaSeguimiento	
Nombre	listaSeguimiento
Codigo	<pre>CREATE PROCEDIMIENTO [dbo].[listaSeguimiento] @codmet INT, @fecha INT AS BEGIN SELECT Detalles.DesRecDet, Detalles.UniRecDet, Detalles.TipRecDet, Detalles.ParDet, Seguimiento.CanSeg, Seguimiento.PreSeg, Seguimiento.CosSeg, Seguimiento.FecSeg, Seguimiento.ObsSeg, Seguimiento.CodMet, Seguimiento.CodSeg from Seguimiento JOIN Detalles ON Detalles.CodDet=Seguimiento.CodDet and DATEPART(mm, Seguimiento.FecSeg)=@fecha and Seguimiento.CodMet=@codmet END</pre>

Procedimiento: modificarDetalle	
Nombre	modificarDetalle
Codigo	<pre>CREATE PROCEDIMIENTO [dbo].[modificarDetalle] @CodDet as int, @CanRecDet as int, @PreUniRecDet as float, @ObsDet as VARCHAR(150), @CodMet as int, @CodAct as int AS DECLARE @costo FLOAT BEGIN set @costo=@CanRecDet*@PreUniRecDet update Detalles set CanRecDet=@CanRecDet, PreUniRecDet=@PreUniRecDet, ObsDet=@ObsDet, CosRecDet=@costo WHERE CodDet=@CodDet update Metas set MonPlaMet=(</pre>

	<pre> SELECT sum(CosRecDet) from Detalles where CodMet=@CodMet AND EstDet=1) WHERE CodMet=@CodMet update Actividades set MonPlaAct=(SELECT sum(MonPlaMet) from Metas where CodAct=@CodAct and EstMet=1) WHERE CodAct=@CodAct END </pre>
--	--

Procedimiento: modificarMaterialFun	
Nombre	modificarMaterialFun
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[modificarMaterialFun] @CodMat as NUMERIC, @NomMat as VARCHAR(80), @CodUni as int, @CosMat as float, @CodAux as VARCHAR(12), @ParMat as int AS BEGIN update Materiales set NomMat=@NomMat, CodUni=@CodUni, CosMat=@CosMat, EstMat=4, TipMat=1, CodAux=@CodAux, ParMat=@ParMat WHERE CodMat=@CodMat END </pre>

Procedimiento: modificarMeta	
Nombre	modificarMeta
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[modificarMeta] @CodMet as int, @DesMet as VARCHAR(250), @CanPlaMet as int, @FecIniMet as DATE, @FecFinMet as DATE, @CodAct as int, @Ene as BIT, @Feb as BIT, @Mar as BIT, @Abr as BIT, @May as BIT, @Jun as BIT, @Jul as BIT, </pre>

	<pre> @Ago as BIT, @Sep as BIT, @Oct as BIT, @Nov as BIT, @Dic as BIT AS BEGIN update Metas set DesMet=@DesMet, CanPlaMet=@CanPlaMet, FecIniMet=@FecIniMet, FecFinMet=@FecFinMet WHERE CodMet=@CodMet update TiempoEjecucion set EneTieEje=@Ene, FebTieEje=@Feb, MarTieEje=@Mar, AbrTieEje=@Abr, MayTieEje=@May, JunTieEje=@Jun, JulTieEje=@Jul, AgoTieEje=@Ago, SepTieEje=@Sep, OctTieEje=@Oct, NovTieEje=@Nov, DicTieEje=@Dic WHERE CodMet=@CodMet END </pre>
--	---

Procedimiento: modificarMontoTecho	
Nombre	modificarMontoTecho
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[modificarMontoTecho] @codtecpres as int, @Monto as int, @codare as int, @gesges as VARCHAR(4) AS BEGIN update TechoPresupuestario set MonTecPre=@Monto, CodAre=@codare, GesGes=@gesges WHERE CodTecPre=@codtecpres END </pre>

Procedimiento: modificarObjetivo	
Nombre	modificarObjetivo
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[modificarObjetivo] @CodAct as int, @DesAct As VARCHAR(250) , @FecAct As DATE AS BEGIN update Actividades set DesAct=@DesAct, FecAct=@FecAct WHERE CodAct=@CodAct END </pre>

Procedimiento: modificarSeguimiento	
Nombre	modificarSeguimiento
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[modificarSeguimiento] @CodSeg int, @CanSeg int, @PreSeg FLOAT, @obsseg VARCHAR(1000) AS DECLARE @costo FLOAT BEGIN set @costo=@CanSeg*@PreSeg update Seguimiento set CanSeg=@CanSeg, PreSeg=@PreSeg, CosSeg=@costo, ObsSeg=@obsseg WHERE CodSeg=@CodSeg END </pre>

Procedimiento: obtenerCantidad	
Nombre	obtenerCantidad
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[obtenerCantidad] @coddet int AS BEGIN SELECT ISNULL((Detalles.CanRecDet-(SELECT sum(Seguimiento.CanSeg) FROM Metas JOIN Detalles on Detalles.CodMet=Metas.CodMet AND Detalles.CodDet=@coddet LEFT JOIN Seguimiento on Seguimiento.CodMet=Metas.CodMet and Detalles.CodDet=Seguimiento.CodDet)), 0) as cantidad FROM Metas JOIN Detalles on Detalles.CodMet=Metas.CodMet AND Detalles.CodDet=@coddet LEFT JOIN Seguimiento on Seguimiento.CodMet=Metas.CodMet and Detalles.CodDet=Seguimiento.CodDet END </pre>

Procedimiento: obtenerDetalle	
Nombre	obtenerDetalle
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[obtenerDetalle] @CodDet as int AS BEGIN SELECT DesRecDet, UniRecDet, CanRecDet, PreUniRecDet, ParDet, ObsDet, TipRecDet, CodAux, CodDet, CodMet FROM Detalles WHERE CodDet=@CodDet END </pre>

Procedimiento: obtenerMaterial	
Nombre	obtenerMaterial
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[obtenerMaterial] @CodMat as VARCHAR(12) AS BEGIN select m.CodMat, m.NomMat, m.ParMat, m.CosMat, m.tipMat, u.NomUni, m.CodAux from Unidades u JOIN Materiales m on u.CodUni=m.CodUni and m.CodMat=@CodMat END </pre>

Procedimiento: obtenerMeta	
Nombre	obtenerMeta
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[obtenerMeta] @CodMet as int AS BEGIN SELECT CodMet, DesMet, CanPlaMet, CONVERT(varchar(15), FecIniMet, 103) as 'from', CONVERT(varchar(15), FecFinMet, 103) as 'to', CodAct, MonPlaMet, UniMet FROM Metas WHERE CodMet=@CodMet END </pre>

Procedimiento: obtenerMontoTecho	
Nombre	obtenerMontoTecho
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[obtenerMontoTecho] @CodTecPre as int AS BEGIN SELECT CodTecPre, CodAre, MonTecPre, GesGes FROM TechoPresupuestario WHERE CodTecPre=@CodTecPre END </pre>

Procedimiento: obtenerObjetivo	
Nombre	obtenerObjetivo
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[obtenerObjetivo] @CodAct as int AS BEGIN select * from Actividades where CodAct=@CodAct END </pre>

Procedimiento: obtenerSeguimiento	
Nombre	obtenerSeguimiento
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[obtenerSeguimiento] @codseg int AS BEGIN SELECT CodSeg, Seguimiento.CodDet, Seguimiento.CodMet, CodSeg, ObsSeg, FecSeg, PreSeg, CosSeg, CanSeg, EstSeg, Detalles.DesRecDet, Detalles.UniRecDet FROM Seguimiento JOIN Detalles on Seguimiento.CodDet=Detalles.CodDet and Seguimiento.CodSeg=@codseg END </pre>

Procedimiento: reporteDetallePersona	
Nombre	reporteDetallePersona
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[reporteDetallePersona] @codper int AS BEGIN SELECT CodDet, DesRecDet, UniRecDet, CanRecDet, PreUniRecDet, CosRecDet, ParDet, ObsDet, TipRecDet, EstDet, VerAuxDet from Personas join Actividades on Actividades.CodPer=Personas.CodPer and Actividades.EstAct=1 and Personas.CodPer=@codper join Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1 join Detalles on Detalles.CodMet=Metas.CodMet and Detalles.EstDet=1 END </pre>

Procedimiento: ReporteMetasActividades	
Nombre	ReporteMetasActividades
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[ReporteMetasActividades] @CodPer INT, @CodMet int, @gestion VARCHAR(4) AS BEGIN SELECT Actividades.CodAct, CodMet, DesMet, CanPlaMet, MonPlaMet, CONVERT(varchar(15), FecIniMet, 103) as FecIniMet, CONVERT(varchar(15), FecFinMet, 103) as FecFinMet, TechoPresupuestario.GesGes from Personas join Actividades on Actividades.CodPer=Personas.CodPer and Actividades.EstAct=1 and Personas.CodPer=@CodPer join Metas on Metas.CodAct=Actividades.CodAct and </pre>

	Metas.EstMet=1 and Metas.CodMet=@CodMet JOIN TechoPresupuestario on TechoPresupuestario.CodTecPre=Actividades.CodTecPre and TechoPresupuestario.GesGes=@gestion END
--	---

Procedimiento: ReporteMetasDetalles	
Nombre	ReporteMetasDetalles
Codigo	CREATE PROCEDIMIENTO [dbo].[ReporteMetasDetalles] @CodPer INT, @CodAct int, @codmet int, @gestion VARCHAR(4) AS BEGIN SELECT Actividades.CodAct, Metas.CodMet, CodDet, DesRecDet, UniRecDet, CanRecDet, PreUniRecDet, CosRecDet, ParDet, ObsDet, TipRecDet, EstDet, VerAuxDet from Personas join Actividades on Actividades.CodPer=Personas.CodPer and Actividades.EstAct=1 and Personas.CodPer=@CodPer and Actividades.CodAct=@CodAct join Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1 JOIN Detalles on Detalles.CodMet=Metas.CodMet and Detalles.CodMet=@codmet JOIN TechoPresupuestario on TechoPresupuestario.CodTecPre=Actividades.CodTecPre and TechoPresupuestario.GesGes=@gestion END

Procedimiento: ReporteObjetivo	
Nombre	ReporteObjetivo
Codigo	CREATE PROCEDIMIENTO [dbo].[ReporteObjetivo] @CodPer As INT, @version int, @gestion VARCHAR(4) AS BEGIN SELECT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct, Actividades.EstAct, Actividades.CodPer, Actividades.CodTecPre, Actividades.MonPlaAct, Actividades.CanPlaAct from Gestion JOIN TechoPresupuestario on TechoPresupuestario.GesGes=Gestion.GesGes and TechoPresupuestario.GesGes=@gestion JOIN Actividades on

	Actividades.CodTecPre=TechoPresupuestario.CodTecPre and Actividades.CodPer=@CodPer and Actividades.EstAct=1 and Actividades.VerAct=@version
	END

Procedimiento: ReporteSeguimiento	
Nombre	ReporteSeguimiento
Codigo	CREATE PROCEDIMIENTO [dbo].[ReporteSeguimiento] @codper int, @gestion VARCHAR(4) AS BEGIN SELECT Actividades.CodAct, Actividades.DesAct, Actividades.FecAct, Actividades.VerAct, Actividades.EstAct, Actividades.CodPer, Actividades.CodTecPre, Actividades.MonPlaAct, Actividades.CanPlaAct, (SELECT SUM(CosSeg) as monto FROM Seguimiento JOIN Metas on Metas.CodMet=Seguimiento.CodMet JOIN Actividades on Actividades.CodAct=Metas.CodAct JOIN Personas on Personas.CodPer=Actividades.CodPer and Actividades.CodPer=@CodPer) from Gestion JOIN TechoPresupuestario on TechoPresupuestario.GesGes=Gestion.GesGes and TechoPresupuestario.GesGes=@gestion JOIN Actividades on Actividades.CodTecPre=TechoPresupuestario.CodTecPre and Actividades.CodPer=@CodPer and Actividades.EstAct=1 and Actividades.VerAct=2 END

Procedimiento: reporteTiempoEjecucionPersona	
Nombre	reporteTiempoEjecucionPersona
Codigo	CREATE PROCEDIMIENTO [dbo].[reporteTiempoEjecucionPersona] @codper int, @codact int, @gestion VARCHAR(4) AS BEGIN SELECT Actividades.CodAct, CodTieEje, EneTieEje, FebTieEje, MarTieEje, AbrTieEje, MayTieEje, JunTieEje, JulTieEje, AgoTieEje, SepTieEje, OctTieEje, NovTieEje, DicTieEje, DesMet, CanPlaMet, MonPlaMet from Personas join Actividades on

	Actividades.CodPer=Personas.CodPer and Actividades.EstAct= 1 and Personas.CodPer=@codper and Actividades.CodAct=@codact join Metas on Metas.CodAct=Actividades.CodAct and Metas.EstMet=1 join TiempoEjecucion on TiempoEjecucion.CodMet=Metas.CodMet and TiempoEjecucion.EstTieEje=1 JOIN TechoPresupuestario on TechoPresupuestario.CodTecPre=Actividades.CodTecPre and TechoPresupuestario.GesGes=@gestion END
--	--

Procedimiento: revertirAreaTecho	
Nombre	revertirAreaTecho
Codigo	CREATE PROCEDIMIENTO [dbo].[revertirAreaTecho] @codtecpre int, @estado NUMERIC(1) AS BEGIN update TechoPresupuestario set EstTecPre=@estado WHERE CodTecPre=@codtecpre END

Procedimiento: revertirDetalle	
Nombre	revertirDetalle
Codigo	CREATE PROCEDIMIENTO [dbo].[revertirDetalle] @CodDet as int, @EstDet as int AS BEGIN update Detalles set EstDet=@EstDet WHERE CodDet=@CodDet END

Procedimiento: revertirMeta	
Nombre	revertirMeta
Codigo	CREATE PROCEDIMIENTO [dbo].[revertirMeta] @CodMet as int, @EstMet as int AS BEGIN update Metas set EstMet=@EstMet WHERE CodMet=@CodMet update TiempoEjecucion set EstTieEje=@EstMet WHERE CodMet=@CodMet END

Procedimiento: revertirObjetivo	
Nombre	revertirObjetivo
Codigo	<pre>CREATE PROCEDIMIENTO [dbo].[revertirObjetivo] @CodAct as int, @EstAct as int AS BEGIN update Actividades set EstAct=@EstAct WHERE CodAct=@CodAct END</pre>

Procedimiento: sp_alterdiagram	
Nombre	sp_alterdiagram
Codigo	<pre>CREATE PROCEDIMIENTO [dbo].[sp_alterdiagram] (@diagramNombre sysNombre, @owner_id int = null, @version int, @definition varbinary(max)) WITH EXECUTE AS 'dbo' AS BEGIN set nocount on declare @theId int declare @retval int declare @IsDbo int declare @UIDFound int declare @DiagId int declare @ShouldChangeUID int if(@diagramNombre is null) begin RAISERROR ('Invalid ARG', 16, 1) return -1 end execute as caller; select @theId = DATABASE_PRINCIPAL_ID(); select @IsDbo = IS_MEMBER(N'db_owner'); if(@owner_id is null) select @owner_id = @theId; revert; select @ShouldChangeUID = 0</pre>

	<pre> select @DiagId = diagram_id, @UIDFound = principal_id from dbo.sysdiagrams where principal_id = @owner_id and Nombre = @diagramNombre if(@DiagId IS NULL or (@IsDbo = 0 and @theId <> @UIDFound)) begin RAISERROR ('Diagram does not exist or you do not have permission.', 16, 1); return -3 end if(@IsDbo <> 0) begin if(@UIDFound is null or USER_NOMBRE(@UIDFound) is null) -- invalid principal_id begin select @ShouldChangeUID = 1 ; end end -- update dds data update dbo.sysdiagrams set definition = @definition where diagram_id = @DiagId ; -- change owner if(@ShouldChangeUID = 1) update dbo.sysdiagrams set principal_id = @theId where diagram_id = @DiagId ; -- update dds version if(@version is Nulos) update dbo.sysdiagrams set version = @version where diagram_id = @DiagId ; return 0 END </pre>
--	---

Procedimiento: sp_creatediagram	
Nombre	sp_creatediagram
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[sp_creatediagram] (@diagramNombre sysNombre, @owner_id int = null, @version int, @definition varbinary(max)) WITH EXECUTE AS 'dbo' AS </pre>

```

BEGIN
set nocount on

declare @theId int
declare @retval int
declare @IsDbo int
declare @userNombre sysNombre
if(@version is null or @diagramNombre is null)
begin
RAISERROR (N'E_INVALIDARG', 16, 1);
return -1
end

execute as caller;
select @theId = DATABASE_PRINCIPAL_ID();
select @IsDbo = IS_MEMBER(N'db_owner');
revert;

if @owner_id is null
begin
select @owner_id = @theId;
end
else
begin
if @theId <> @owner_id
begin
if @IsDbo = 0
begin
RAISERROR (N'E_INVALIDARG', 16, 1);
return -1
end
select @theId = @owner_id
end
end
-- next 2 line only for test, will be removed after define
Nombre unique
if EXISTS(select diagram_id from dbo.sysdiagrams where
principal_id = @theId and Nombre = @diagramNombre)
begin
RAISERROR ('The Nombre is already used.', 16, 1);
return -2
end

insert into dbo.sysdiagrams(Nombre, principal_id , version,
definition)
VALUES(@diagramNombre, @theId, @version, @definition) ;

select @retval = @@IDTABLA
return @retval
END

```

Procedimiento: sp_dropdiagram	
Nombre	sp_dropdiagram
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[sp_dropdiagram] (@diagramNombre sysNombre, @owner_id int = null) WITH EXECUTE AS 'dbo' AS BEGIN set nocount on declare @theId int declare @IsDbo int declare @UIDFound int declare @DiagId int if(@diagramNombre is null) begin RAISERROR ('Invalid value', 16, 1); return -1 end EXECUTE AS CALLER; select @theId = DATABASE_PRINCIPAL_ID(); select @IsDbo = IS_MEMBER(N'db_owner'); if(@owner_id is null) select @owner_id = @theId; REVERT; select @DiagId = diagram_id, @UIDFound = principal_id from dbo.sysdiagrams where principal_id = @owner_id and Nombre = @diagramNombre if(@DiagId IS NULL or (@IsDbo = 0 and @UIDFound <> @theId)) begin RAISERROR ('Diagram does not exist or you do not have permission.', 16, 1) return -3 end delete from dbo.sysdiagrams where diagram_id = @DiagId; return 0; END </pre>

Procedimiento: sp_helpdiagramdefinition	
Nombre	sp_helpdiagramdefinition

Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[sp_helpdiagramdefinition] (@diagramNombre sysNombre, @owner_id int = null) WITH EXECUTE AS N'dbo' AS BEGIN set nocount on declare @theId int declare @IsDbo int declare @DiagId int declare @UIDFound int if(@diagramNombre is null) begin RAISERROR (N'E_INVALIDARG', 16, 1); return -1 end execute as caller; select @theId = DATABASE_PRINCIPAL_ID(); select @IsDbo = IS_MEMBER(N'db_owner'); if(@owner_id is null) select @owner_id = @theId; revert; select @DiagId = diagram_id, @UIDFound = principal_id from dbo.sysdiagrams where principal_id = @owner_id and Nombre = @diagramNombre; if(@DiagId IS NULL or (@IsDbo = 0 and @UIDFound <> @theId)) begin RAISERROR ('Diagram does not exist or you do not have permission.', 16, 1); return -3 end select version, definition FROM dbo.sysdiagrams where diagram_id = @DiagId ; return 0 END </pre>
--------	---

Procedimiento: sp_helpdiagrams	
Nombre	sp_helpdiagrams
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[sp_helpdiagrams] (@diagramNombre sysNombre = NULL, </pre>

	<pre> @owner_id int = NULL) WITH EXECUTE AS N'dbo' AS BEGIN DECLARE @user sysNombre DECLARE @dboLogin bit EXECUTE AS CALLER; SET @user = USER_NOMBRE(); SET @dboLogin = CONVERT(bit, IS_MEMBER('db_owner')); REVERT; SELECT [Database] = DB_NOMBRE(), [Nombre] = Nombre, [ID] = diagram_id, [Owner] = USER_NOMBRE(principal_id), [OwnerID] = principal_id FROM sysdiagrams WHERE (@dboLogin = 1 OR USER_NOMBRE(principal_id) = @user) AND (@diagramNombre IS NULL OR Nombre = @diagramNombre) AND (@owner_id IS NULL OR principal_id = @owner_id) ORDER BY 4, 5, 1 END </pre>
--	--

Procedimiento: sp_reNombredidiagram	
Nombre	sp_reNombredidiagram
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[sp_reNombredidiagram] (@diagramNombre sysNombre, @owner_id int = null, @new_diagramNombre sysNombre) WITH EXECUTE AS 'dbo' AS BEGIN set nocount on declare @theId int declare @IsDbo int declare @UIDFound int declare @DiagId int declare @DiagIdTarg int declare @u_Nombre sysNombre </pre>

```

if((@diagramNombre is null) or (@new_diagramNombre is
null))
begin
RAISERROR ('Invalid value', 16, 1);
return -1
end

EXECUTE AS CALLER;
select @theId = DATABASE_PRINCIPAL_ID();
select @IsDbo = IS_MEMBER(N'db_owner');
if(@owner_id is null)
select @owner_id = @theId;
REVERT;

select @u_Nombre = USER_NOMBRE(@owner_id)

select @DiagId = diagram_id, @UIDFound = principal_id
from dbo.sysdiagrams where principal_id = @owner_id and
Nombre = @diagramNombre
if(@DiagId IS NULL or (@IsDbo = 0 and @UIDFound <>
@theId))
begin
RAISERROR ('Diagram does not exist or you do not have
permission.', 16, 1)
return -3
end

-- if((@u_Nombre is Nulos) and (@new_diagramNombre =
@diagramNombre)) -- nothing will change
-- return 0;

if(@u_Nombre is null)
select @DiagIdTarg = diagram_id from dbo.sysdiagrams
where principal_id = @theId and Nombre =
@new_diagramNombre
else
select @DiagIdTarg = diagram_id from dbo.sysdiagrams
where principal_id = @owner_id and Nombre =
@new_diagramNombre

if((@DiagIdTarg is Nulos) and @DiagId <> @DiagIdTarg)
begin
RAISERROR ('The Nombre is already used.', 16, 1);
return -2
end

if(@u_Nombre is null)
update dbo.sysdiagrams set [Nombre] =
@new_diagramNombre, principal_id = @theId where
diagram_id = @DiagId
else

```

	update dbo.sysdiagrams set [Nombre] = @new_diagramNombre where diagram_id = @DiagId return 0 END
--	---

Procedimiento: sp_upgraddiagrams	
Nombre	sp_upgraddiagrams
Codigo	<pre> CREATE PROCEDIMIENTO [dbo].[sp_upgraddiagrams] AS BEGIN IF OBJECT_ID(N'dbo.sysdiagrams') IS NULOS return 0; CREATE TABLE dbo.sysdiagrams (Nombre sysNombre NULOS, principal_id int NULOS, -- we may change it to varbinary(85) diagram_id int CLAVE PRIMARIA IDTABLA, version int, definition varbinary(max) CONSTRAINT UK_principal_Nombre UNIQUE (principal_id, Nombre)); /* Add this if we need to have some form of extended properties for diagrams */ /* IF OBJECT_ID(N'dbo.sysdiagram_properties') IS NULL BEGIN CREATE TABLE dbo.sysdiagram_properties (diagram_id int, Nombre sysNombre, value varbinary(max) NULOS) END */ IF OBJECT_ID(N'dbo.dtproperties') IS NULOS begin insert into dbo.sysdiagrams ([Nombre], [principal_id], </pre>

	<pre> [version], [definition]) select convert(sysNombre, dgm.[uvalue]), DATABASE_PRINCIPAL_ID(N'dbo'), -- will change to the sid of sa 0, -- zero for old format, dgdef.[version], dgdef.[lvalue] from dbo.[dtproperties] dgm inner join dbo.[dtproperties] dggd on dggd.[property] = 'DtgSchemaGUID' and dggd.[objectid] = dgm.[objectid] inner join dbo.[dtproperties] dgdef on dgdef.[property] = 'DtgSchemaDATA' and dgdef.[objectid] = dgm.[objectid] where dgm.[property] = 'DtgSchemaNOMBRE' and dggd.[uvalue] like N'_EA3E6268-D998-11CE-9454- 00AA00A3F36E_' return 2; end return 1; END </pre>
--	--

II.2. Componente:Capacitacion

II.2.1 Introducción

Ante los avances tecnológicos y humanos, el término de Capacitación y Sistemas de Información están modificando la forma de trabajo de las empresas, los sistemas de Información ayudan a acelerar los procesos, por tanto, las organizaciones que los implantan logran ventajas competitivas al adoptarlos en sus funciones. Pero si el personal no está debidamente capacitado la producción de las empresa será deficiente y en mucho casos incompleto.

La capacitación se refiere a las metodologías que se usan para proporcionar a las personas dentro de la empresa, las habilidades necesarias que necesitan para realizar su trabajo de una manera más eficiente, esto contempla desde pequeños cursos que le permitan al usuario entender el funcionamiento básico del sistema nuevo, hasta capacitación más profunda y avanzadas a bases de prácticas y material didáctico como libros (Tutoriales) y otros.

La capacitación es un proceso que lleva a la mejora continua de la producción y con esto implantar nuevas formas de trabajo, como en este caso el manejo adecuado del Sistema Informático “SIAF” que ayuda a agilizar la centralización manual de la

información de los files de los funcionarios y su difusión, así el control administrativo y oportuno de la generación de los mismos.

II.2.2. Objetivos

II.2.2.1. Objetivo General

Lograr la participación y aportación por parte de las personas, haciendo uso de las nuevas tecnologías de Información y Comunicación

II.2.2.2. Objetivo Específico

- Fomentar la utilización del Sistema Web desarrollado.
- Preparar a los funcionarios para la ejecución de las diversas tareas y responsabilidades de la organización.

II.2.3 Aspectos Pedagógicos de la Capacitación

II.2.3.1. Definición de Capacitación

La capacitación es un proceso educacional de carácter estratégico aplicado de manera organizada y sistémica, mediante el cual los funcionarios adquieren o desarrollan conocimientos y habilidades específicas relativas al trabajo, y modifica sus actitudes frente a los que hacerse de la organización, el puesto o el ambiente laboral.

II.2.3.2. Importancia de la Capacitación

Permite que:

- Consolidación en la integración de los funcionarios de la organización.
- Mayor identificación con la cultura organizacional.
- Disposición desinteresada por el logro de la misión empresarial.
- Entrega total de esfuerzo por llegar a cumplir con las actividades, ejecución, aprobación.
- Mayor desempeño laboral de los funcionarios.
- Alta productividad.
- Promueve la creatividad, innovación y disposición para el trabajo.
- Desarrollo de una mejor comunicación entre los miembros de una organización.
- Reducción de tiempos.

- Aumento de la armonía, el trabajo en equipo y por ende de la cooperación y coordinación.
- Obtener información de fuente confiable, como son los funcionarios.

II.2.3.3. Beneficios de la Capacitación

Como beneficia la capacitación a la empresa:

- Conduce a rentabilidad más alta y a actitudes más positivas.
- Mejora el conocimiento del puesto a todos los niveles.
- Eleva la moral de la fuerza de trabajo.
- Ayuda al personal a identificarse con los objetivos de la organización.
- Crea mejor imagen.
- Mejora la relación jefes-subordinados.
- Es un auxiliar para la comprensión y adopción de políticas.
- Se agiliza la toma de decisiones y la solución de problemas.
- Incrementa la productividad y calidad del trabajo.
- Elimina los costos de recurrir a consultores externos

II.2.3.4. Metodología

Para realizar la capacitación acerca del uso del Sistema Informático “SIAF”, se hizo uso de técnicas de capacitación aplicada en la empresa, más propiamente la técnica conocida como capacitación en el puesto. Esta técnica contempla que una persona aprenda una responsabilidad mediante su desempeño real en el Sistema Informático “SIAF” ya sea un Administrador o Usuario, aplicando esta metodología, la capacitación se realizó en las instalaciones de la empresa COSAALT.

Lo pasos que se contemplaron para realizar una capacitación en el puesto son:

- Preparación del aprendiz.
- Presentación del Sistema Informático Sistema Integrado Administrativo y Financiero “SIAF”
- Seguimiento

II.2.3.4.1. Contenido

El contenido de la capacitación se subdivide en 3 módulos la cuales se detallan a continuación:

- Módulo 1: La Internet

En esta unidad de aprendizaje se realiza una exposición, en donde el capacitador explica: sobre La Internet mediante los siguientes puntos:

1. Introducción
 2. ¿De dónde proviene?
 3. Instalación del Navegador Google Chrome
 4. Concepto básico del Navegador Google Chrome
 5. Configuración de Google Chrome
 6. Página Web
- Módulo 2: Presentación del Sistema Informático SIAF “Sistema Integrado Administrativo y Financiero”

En esta unidad de aprendizaje se hace la presentación del Sistema Informático “SIAF”.

- Módulo 3: Capacitación del Sistema SIAF

En esta unidad de aprendizaje se explica los pasos necesarios para administrar el sitio Web dirigido a las Acciones que puede solicitar los funcionarios.

El contenido de la capacitación se realizó de acuerdo al siguiente cronograma:

Fecha	Hora	Actividad	Expositor
25 de Noviembre del 2013	15:00 – 16:30	Presentación de los participantes y Objetivo de la capacitación. Módulo 1: La Internet Módulo 2: Presentación, Manejo y Capacitación del Sistema Informático “SIAF”. Dirigido para todos los funcionarios.	Andrea Jesica Sanchez Cardozo
25 de Noviembre del 2013	15:00 – 16:00	Módulo 3: Presentación, Manejo y Capacitación del Sistema Informático “SIAF”. Dirigido para todos los funcionarios del área de RRHH y personal involucrado.	Andrea Jesica Sanchez Cardozo

II.2.3.4.2. Medios a utilizar

II.2.3.4.2.1. Aspectos Técnicos

- 1 Computador con el Sistema Desarrollado.

II.2.3.4.2. Aspectos Logísticos

- Material digital informativo (dvds)

II.2.3.5. Conclusiones

Realizada la capacitación y según la ejecución del contenido planteado, se pudo diferenciar tipos de cambio en el personal, como por ejemplo el desarrollo de destreza y conocimiento de manera directa respecto al tema de Internet; además se desarrolló un cambio del personal por insertar las TIC en la Empresa COSAALT, así también se pudo identificar nuevos requerimientos que se podrán implementar a futuro, es así como la culminación de la capacitación terminó despejando todas la dudas del personal capacitado.

II.2.3.6. Medios de Verificación

- Certificado de capacitación en tecnologías de información y comunicación (TIC) para el Mejoramiento para la planificación de los formularios de la elaboración de los POA's de las áreas de COSAALT, firmados por director del proyecto.

Capítulo III: Conclusiones y Recomendaciones

III.1. Conclusiones

Culminado el presente proyecto, se detallan las siguientes conclusiones:

- Se logró cumplir el propósito del proyecto, a través de su indicador reducido el número de errores en un 70% en los procesos de Formulación, Evaluación y Ejecución.
- El sistema fue desarrollado por el Framework Spring Source, el que garantizó la integridad y consistencia de datos, utilizando así el Modelo Vista Controlador (MVC), lo que permitió agilizar la conclusión del sistema.
- Se aplicó la metodología RUP para el desarrollo de software, lo que implicó apoyarse también en la metodología de prototipos, para que la etapa de desarrollo pueda abarcar con los requerimientos que no se pudieron identificar en la etapa de análisis.
- Todas las herramientas de software descritas en el documento han sido de gran utilidad habiendo facilitado el desarrollo del sistema en sus diferentes etapas.
- Por último, se puede afirmar que la aplicación de las Tecnologías de Información y Comunicación (TIC) han sido definitivamente útiles y determinantes en el mejoramiento para la planificación de la elaboración del Plan Operativo Anual (POA) de la Cooperativa de servicios de agua potable y alcantarillado de Tarija (COSAALT-LTDA).
- Se realizó la capacitación la que asegura que tanto el administrador del sistema, como los funcionarios de la empresa COSAALT, operarán correctamente el sistema, reduciendo los errores.

III.2 Recomendaciones

- Se aplicó la herramienta Spring Source la que permitió optimizar código fuente, evitar mapeos e instancias en clases externas, facilitó la configuración con el servidor Apache Tomcat, así también su estructura de Modelo Vista Controlador proporcionó una visualización más descriptiva para el entorno de la programación, por estos motivos se recomienda aplicar Spring Source.

- Para el desarrollo de un sistema, se recomienda estudiar las ventajas de la metodología RUP, como de otras para así elegir de acuerdo a los alcances de lo propuesto la aplicación de la metodología correcta.
- Para la implementación y explotación del proyecto se recomienda realizar periódicamente copias de respaldo de información (backups), por el motivo de evitar pérdidas de información.
- Se recomienda ampliar el sistema desarrollado para contemplar nuevos módulos que abarquen marcaciones por medios biométricos, control de descuentos adicionales, considerando que el diseño es flexible y sujeto de escalamiento. Por último se recomienda realizar la capacitación a cada nuevo funcionario para que este pueda utilizar el sistema sin mayores complicaciones y de manera correcta.