

**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA DE CUMPLIMIENTO
REGULATORIO: CUMPLE**

T E S I S

que para obtener el título de
INGENIERO EN COMPUTACIÓN

presenta
GABRIELA LÓPEZ LÓPEZ

**DIRECTOR DE TESIS:
ING. CARLOS ROMÁN ZAMITIZ**



MEXICO, D.F.

2009

...agradezco a Dios por permitirme sentir esta maravillosa vida...

Le dedico esta tesis a mi madre, a mi padre, a mi hermana y a mi hermano, quienes me han mostrado el camino desde su propio punto de vista, siempre dejándome escoger el mío.

Esta tesis significa para mí, la culminación de una etapa llena de retos, de preparación, y de alegrías.

Agradezco a mis amigos y compañeros de la universidad, especialmente a Iván y su familia.

A Saitosoft que me brindó la oportunidad de aprender y crecer con ellos.

Gracias a Alberto, Uriel y Mike, que me alentaron en todo momento.

Gracias Facultad de ingeniería.

CONTENIDO

| | |
|---|-----------|
| INTRODUCCIÓN | 4 |
| ANTECEDENTES | 6 |
| 1.1 INGENIERÍA DE SOFTWARE | 6 |
| 1.2 CICLOS DE VIDA | 7 |
| MODELO EN CASCADA (CICLO DE VIDA CLÁSICO). | 7 |
| MODELO DE PROTOTIPOS (EVOLUTIVO) | 7 |
| MODELO EN ESPIRAL | 8 |
| 1.3 PROCESO UNIFICADO | 9 |
| DIRIGIDO POR CASOS DE USO | 9 |
| CENTRADO EN LA ARQUITECTURA | 10 |
| ENFOCADO EN LOS RIESGOS | 10 |
| CICLO DE VIDA ITERATIVO E INCREMENTAL DEL PROCESO UNIFICADO | 10 |
| 1.4 FASES DEL CICLO DE VIDA DEL PROCESO UNIFICADO | 11 |
| 1.5 INTERNET Y LA WORLD WIDE WEB | 12 |
| 1.6 ARQUITECTURA CLIENTE-SERVIDOR | 13 |
| 1.7 APACHE TOMCAT | 13 |
| 1.8 JAVA | 14 |
| PLATAFORMA JAVA | 15 |
| PRINCIPALES CARACTERÍSTICAS | 16 |
| 1.9 CMMI | 18 |
| NIVELES DE MADUREZ (MODELO ESCALONADO) | 20 |
| 1.10 UML | 22 |
| DEFINICIÓN | 22 |
| ELEMENTOS | 22 |
| MODELO DINÁMICO | 24 |
| MODELO LÓGICO | 27 |
| MODELO FÍSICO | 29 |
| MODELO DE CASOS DE USO. | 30 |
| 1.11 ARQUITECTURA MODELO-VISTA-CONTROLADOR (MVC) | 31 |
| 1.12 ESTIMACIÓN POR PUNTOS DE CASOS DE USO. | 33 |
| 1.13 CONCEPTOS DE BASES DE DATOS | 37 |
| DEFINICIÓN DE BASES DE DATOS | 37 |
| SISTEMA DE GESTIÓN DE BASES DE DATOS | 37 |
| FASES DEL DISEÑO DE BASES DE DATOS | 38 |
| MODELOS DE DATOS | 38 |
| DIAGRAMA ENTIDAD-RELACIÓN (E-R) | 39 |
| CONCEPTOS | 39 |

ETAPA DE CONCEPCIÓN **41**

| | | |
|------------|--|-----------|
| 2.1 | DEFINICIÓN DEL PROBLEMA | 42 |
| | ¿POR QUÉ LA NECESIDAD DE UN SISTEMA DE CONTROL DE DISPOSICIONES LEGALES? | 42 |
| 2.2 | OBJETIVOS PRINCIPALES | 42 |
| 2.3 | ALCANCE DEL SISTEMA | 42 |
| 2.4 | FACTIBILIDAD DEL SISTEMA | 43 |
| 2.5 | FLUJO DE CUMPLIMIENTO REGULATORIO ACTUAL | 44 |
| | PROCESO AS IS: | 44 |
| 2.6 | ANÁLISIS DE RIESGOS | 46 |
| 2.7 | ALTERNATIVA DE SOLUCIÓN | 47 |
| | PROCESO TO BE: | 47 |
| 2.8 | METODOLOGÍA DE DESARROLLO PROPUESTA. | 48 |

ETAPA DE ELABORACIÓN **50**

| | | |
|------------|--|-----------|
| 3.1 | ANÁLISIS DE LAS NECESIDADES DEL SISTEMA | 51 |
| | ENTREVISTAS CON EL CLIENTE | 51 |
| 3.2 | REQUERIMIENTOS FUNCIONALES | 52 |
| | A. ADMINISTRACIÓN DEL SISTEMA. | 53 |
| | B. CONFIGURACIÓN DE MARCO REGULATORIO. | 54 |
| | C. SEGUIMIENTO DE TAREAS. | 59 |
| | D. ADMINISTRACIÓN DE RIESGOS. | 59 |
| | E. MAPA DE SEGUIMIENTO. | 60 |
| 3.3 | REQUERIMIENTOS NO FUNCIONALES | 60 |
| 3.4 | ESTIMACIÓN | 61 |
| | ESTIMACIÓN POR CASOS DE USO | 61 |
| 3.5 | PLAN DE PROYECTO | 64 |
| 3.6 | DISEÑO DEL SISTEMA | 65 |
| | MODELO CONCEPTUAL | 65 |
| | ARQUITECTURA DEL SISTEMA | 67 |
| | DIAGRAMA DE CLASES | 69 |
| | DIAGRAMAS DE SECUENCIA | 71 |
| | DIAGRAMA DE COMPONENTES | 73 |
| | DIAGRAMA DE DESPLIEGUE | 74 |
| 3.7 | MODELO RELACIONAL | 74 |
| | DISEÑO DE LA BASE DE DATOS | 74 |
| | DICCIONARIO DE DATOS | 77 |

ETAPA DE CONSTRUCCIÓN **94**

| | | |
|------------|--|-----------|
| 4.1 | SISTEMA DE CUMPLIMIENTO REGULATORIO | 95 |
| | ÁREAS DE APLICACIÓN | 95 |

| | |
|--|------------|
| OBJETIVOS | 95 |
| 4.2 DETALLES DE LA METODOLOGÍA EMPLEADA. | 95 |
| ETAPA 1. CONFIGURACIÓN | 95 |
| ETAPA 2. ASIGNACIÓN DE RESPONSABLES | 96 |
| ETAPA 3. CONTROL Y DOCUMENTACIÓN | 96 |
| ETAPA 4. ADMINISTRACIÓN DEL RIESGO | 96 |
| 4.3 ESTRUCTURA DE LA APLICACIÓN | 97 |
| MENÚ PRINCIPAL DEL SISTEMA | 98 |
| FUNCIONALIDADES IMPORTANTES | 105 |
| ÁREA DE ADMINISTRACIÓN | 115 |
| 4.4 CODIFICACIÓN | 115 |
| GIROEMPRESA.JAVA | 115 |
| MANEJAGIROEMPRESA.JAVA | 116 |
| CATALOGOGIROEMPRESA.JSP | 117 |
| CAPTURAGIROEMPRESA.JSP | 119 |
| GUARDAGIROEMPRESA.JSP | 120 |
| | |
| <i>ETAPA DE TRANSICIÓN</i> | 121 |
| | |
| 5.1 CARACTERÍSTICAS GENERALES DEL SISTEMA | 122 |
| SISTEMA DE CUMPLIMIENTO REGULATORIO | 122 |
| 5.2 REQUISITOS DE INSTALACIÓN | 122 |
| REQUERIMIENTOS DE SOFTWARE DEL SERVIDOR | 122 |
| REQUERIMIENTOS DE HARDWARE DEL SERVIDOR | 122 |
| REQUERIMIENTOS PARA EQUIPOS CLIENTE | 122 |
| 5.3 INSTALACIÓN Y CONFIGURACIÓN | 123 |
| 5.4 PRODUCTOS DE TRABAJO | 125 |
| 5.5 GLOSARIO DE TÉRMINOS | 127 |
| | |
| CONCLUSIONES | 130 |
| | |
| BIBLIOGRAFÍA | 132 |

Introducción

En la actualidad todas las empresas u organizaciones están sometidas a lineamientos de carácter legal, dependiendo de su lugar de establecimiento y giro. Los lineamientos legales son puntos (reglas) escritos por una autoridad jurídica (Secretaría de Energía, Secretaría del Trabajo, etc.), establecidos para tener el control del buen uso y funcionamiento. Cuando una empresa se crea, tiene la responsabilidad de conocer cuales lineamientos debe cumplir y realizar las adecuaciones necesarias para que la autoridad verifique el cumplimiento.

Es importante que los lineamientos legales se cumplan en el tiempo previsto y de manera total, de no ser de esta manera, la autoridad responsable de verificar el cumplimiento puede decidir el cierre de las actividades en la empresa debido al incumplimiento. En algunos casos, se presentan multas para las empresas, o clausuras parciales, sin embargo, de no cumplirse con estos puntos, la pérdida financiera para las empresas es exponencial al tiempo transcurrido.

Algunos lineamientos legales son: las leyes federales, los tratados internacionales, las leyes estatales y municipales, cada una de ellas, define a quién va dirigida, quién es el encargado de ejercerla, y que consecuencias tiene el incumplimiento de la misma.

Este trabajo de tesis está enfocado a describir el proceso completo para elaborar un sistema que resuelva la problemática real de una empresa, mediante las herramientas aprendidas durante la carrera de Ingeniero en Computación, durante mi vida como estudiante y mis años como profesionalista.

Para este propósito escribí 6 capítulos, basados en las etapas el ciclo de vida del Proceso Unificado: Concepción, Elaboración, Construcción, y Transición.

En el Capítulo 1 (Antecedentes), esbozo el marco general y teórico que necesitaremos para la realización del sistema.

En el Capítulo 2 (Descripción del Sistema de Interés) me refiero a la primera etapa del ciclo de vida: Concepción, donde defino el problema real de la empresa, listo las necesidades generales de la empresa, establezco la situación económica, el alcance del sistema, la factibilidad del sistema y la administración de riesgos.

En el Capítulo 3 (Elaboración) describo el proceso de análisis en base a una lista de requerimientos obtenida de las entrevistas con el cliente, realizo la estimación, el plan del proyecto, el diseño de la arquitectura del sistema, y realizo diagramas generales de diseño así como el diagrama entidad-relación de la base de datos.

En el Capítulo 4 (Sistema de Cumplimiento Regulatorio CUMPLE) refiriéndome a la etapa de construcción, listo las funcionalidades más importantes del sistema, su estructura y las ventajas de usarlo.

En el Capítulo 5 (Transición). Describo de manera general cómo se lleva a cabo la transición con los usuarios de la primera versión del Sistema.

En el Capítulo 6 (Conclusiones). Desarrollo las conclusiones del Sistema.

Capítulo 1

Antecedentes

Para tener un marco histórico y global de lo que significa el desarrollo de un software, y tener todos los antecedentes necesarios que están involucrados en su desarrollo, escribiré el primer capítulo de antecedentes. Cada uno de los temas tiene una aplicación directa en el desarrollo del Sistema de Cumplimiento Regulatorio.

Antecedentes

El ciclo de vida de un software estrictamente comienza con su concepción y termina en el momento que se desinstala, sin embargo, comúnmente llamamos ciclo de vida del software a un subconjunto de este proceso, a las etapas entre la concepción y la entrega del producto (muchas veces se continúa el ciclo de vida con una etapa más de mantenimiento o reingeniería). Un ciclo de vida establece el orden de las etapas del proceso de desarrollo del software y los criterios a tener en cuenta para poder pasar de una etapa a la siguiente. Durante este trabajo solo contemplaremos el ciclo de vida hasta la entrega del producto. Para ayudarnos a definir el proceso de desarrollo por donde un software de calidad debe pasar nos ayudaremos de la ingeniería de software.

1.1 Ingeniería de Software

La ingeniería de software es la aplicación de la ingeniería para lograr soluciones correctas, utilizables y económicas a los problemas de desarrollo de software, su objetivo es construir un producto de software o mejorar uno existente. La ingeniería de software permite a los que la practican, controlar el proceso de desarrollo y les aporta bases para construir software de alta calidad de una forma productiva.

Como definición formal de la **Ingeniería de Software** tenemos la propuesta por Fritz Bauer: "El establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico que sea fiable y funcione de manera eficiente sobre máquinas reales."

El campo de aplicación de la ingeniería de software es amplio. Alguno de los aspectos de la ingeniería de software pueden clasificarse como ciencia matemática o de cómputo; otros aspectos entran en las áreas de economía, administración o psicología.

El **proceso de desarrollo de software** descrito por Jacobson: "*es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo*".

Concretamente el proceso de desarrollo de software define quién hace qué, cuándo y cómo. Para hacer esto, la ingeniería de software contempla una serie de modelos que abarcan los métodos, las herramientas y los procedimientos para controlar y definir todo el proceso. Estos modelos se denominan frecuentemente paradigmas de la ingeniería del software y no son más que una visión particular de la realidad para que todos los involucrados en el desarrollo del software tengan un patrón a seguir y puedan organizar su comunidad. La elección de un paradigma se realiza básicamente de acuerdo al tipo del proyecto y de la aplicación, teniendo en cuenta las entregas a realizar durante el desarrollo del software.

Las principales diferencias entre los distintos modelos de ciclo de vida están en:

- Su alcance, es decir, las etapas que abarca. Un proyecto puede comprender un simple estudio de viabilidad del desarrollo de un producto, el desarrollo completo hasta sus pruebas de aceptación o, llevar toda la historia del producto con su desarrollo, fabricación, y modificaciones posteriores hasta su retirada del mercado.
- Las tareas que se deben realizar durante las etapas en que se divide el ciclo. Esto puede depender del propio asunto o tema al que se refiere el proyecto, no son las mismas tareas que se deben realizar para proyectar un ERP que el mantenimiento a un módulo de un sistema.
- Y la secuencia en la que se realizan las etapas.

Entre los paradigmas más conocidos se encuentran el modelo en cascada, modelo en espiral y modelo

de prototipos.

1.2 Ciclos de Vida

Modelo en cascada (ciclo de vida clásico).

El modelo en cascada es un enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior es decir, progresa a través del análisis, diseño, codificación, pruebas y mantenimiento. El modelo permite la retroalimentación con la etapa inmediatamente anterior, de esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo. El ciclo de vida clásico o en cascada presenta problemas en los proyectos reales, ya que no siguen el flujo secuencial que propone el modelo y por consiguiente crea problemas en la aplicación del paradigma. Entre los detalles a tomar en cuenta para la selección de este modelo se encuentra que normalmente es difícil para el cliente, establecer todos los requisitos al inicio del desarrollo por lo que durante el proceso se pueden tener cambios no esperados. Otro detalle, es que no se tendrá una versión operativa del sistema hasta llegar a las etapas finales del ciclo, por lo que cuando se tenga una versión operativa, si no se realizó un análisis y diseño correcto lo que se tendrá son errores desastrosos. Normalmente se usa en proyectos pequeños o para productos muy similares que ya se hayan construido.

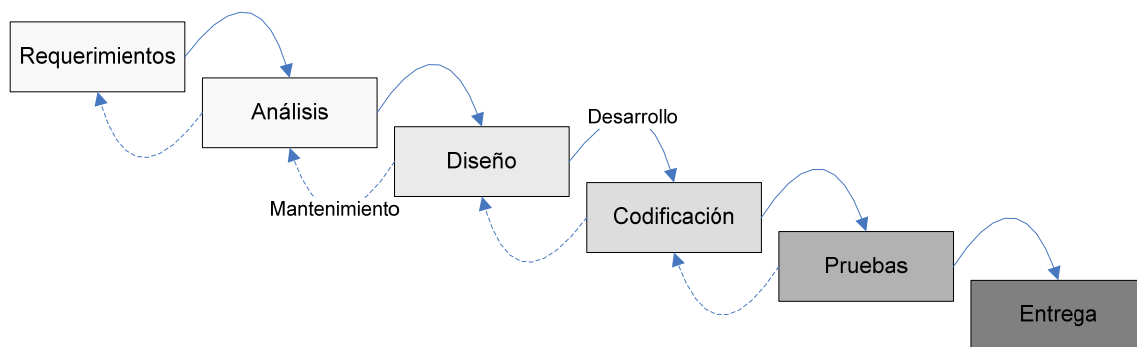


Figura 1 Modelo en Cascada

Modelo de prototipos (evolutivo)

El modelo de prototipos se refiere a elaborar un prototipo con las necesidades generales del cliente para el software, sin identificar los requisitos detallados de entrada, procesamiento o salida. Se inicia con la definición de los objetivos globales del cliente, luego se identifican los requisitos conocidos y las áreas en donde es necesaria más definición. Entonces se plantea con rapidez un diseño de prototipo y se construye. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o usuario final, el cual es evaluado por él mismo para una retroalimentación temprana, gracias a ésta se refinan los requisitos del software que se desarrollará. La iteración ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

La realización de prototipos es una herramienta en la que se apoyan diferentes modelos de ciclo de vida. Un prototipo debe tener el objetivo de mostrar al cliente o a la gerencia del proyecto el resultado que se obtendrá en la implementación de a cada uno de los requerimientos del cliente una vez terminado el desarrollo.

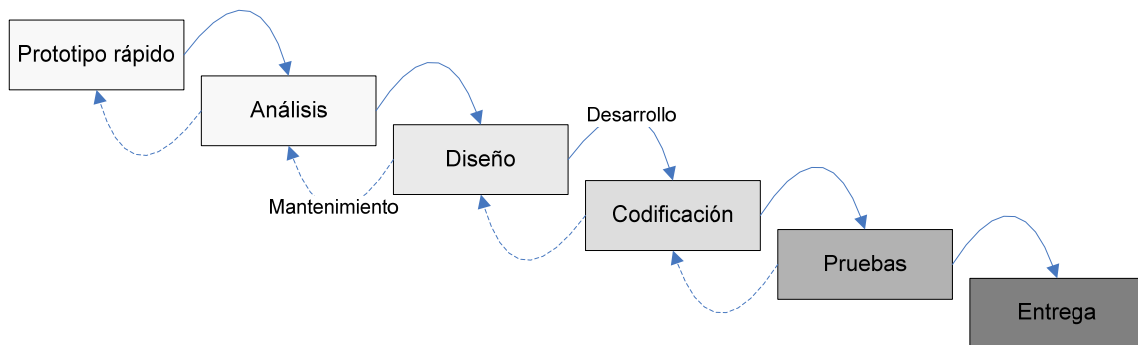


Figura 2 Modelo de Prototipos

Modelo en espiral

El modelo en espiral es un modelo de ciclo de vida desarrollado por Barry Boehm en 1985, ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo análisis de riesgo, que falta en esos paradigmas.

El modelo, representado mediante la espiral define cuatro actividades principales, representadas por los cuatro cuadrantes de la figura:

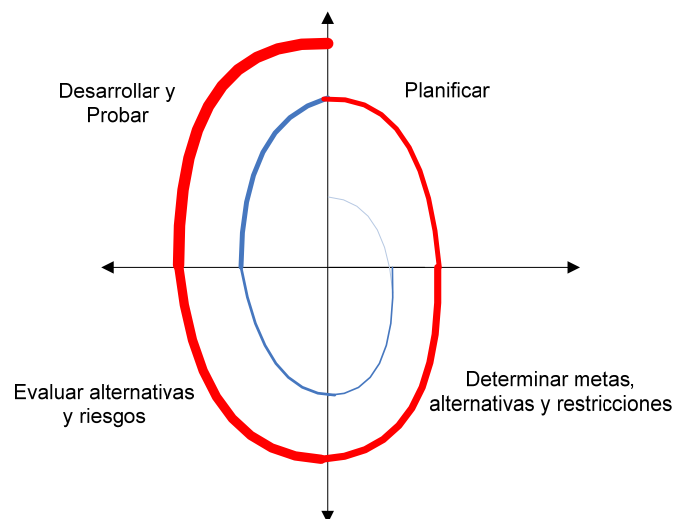


Figura 3 Modelo en Espiral

Planificación: *determinación de objetivos, alternativas y restricciones.*

Análisis de riesgo: *análisis de alternativas e identificación/resolución de riesgos.*

Ingeniería: *desarrollo del producto de "siguiente nivel"*

Evaluación del cliente: *valoración de los resultados de la ingeniería*

Un aspecto muy importante a considerar del modelo en espiral, es que con cada iteración alrededor del centro de la espiral se construye una versión más completa del software.

Durante la primera vuelta alrededor de la espiral se definen los objetivos, las restricciones, se

descubren y analizan los riesgos y se contemplan las alternativas, si el análisis de los riesgos conduce a que existe una incertidumbre en los requisitos, se continua con la etapa de ingeniería en el modelo, donde se puede usar la creación de prototipos, simuladores y/o otros modelos para definir tanto con el cliente como con el desarrollador hasta el total entendimiento y aceptación del requisito.

El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación del cliente) y puede sugerir modificaciones, en base a estos comentarios del cliente, se produce la siguiente fase de planeación y de análisis de riesgo.

En este modelo cada bucle alrededor de la espiral tiene un paso decisivo con el final del análisis de riesgo ya que puede resultar en una decisión de “seguir o no seguir” es decir, si los riesgos son demasiado grandes, se puede dar por terminado el proyecto.

Sin embargo, en la mayoría de los casos, se sigue avanzando alrededor del camino de la espiral, y ese camino lleva a los desarrolladores hacia fuera, hacia un modelo más completo del sistema, y al final, al propio sistema operacional. Cada vuelta alrededor de la espiral requiere ingeniería, que se puede llevar a cabo mediante el enfoque del ciclo de vida clásico o de la creación de prototipos. Debe tenerse en cuenta que el número de actividades de desarrollo que ocurren en el cuadrante inferior derecho aumenta al alejarse del centro de la espiral.

El paradigma del modelo en espiral para la ingeniería del software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque “evolutivo” para la ingeniería del software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en reducción del riesgo, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto. Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico, pero incorporándola dentro de un marco de trabajo interactivo que refleja de forma más realista el mundo real.

Pero, al igual que otros paradigmas, el modelo en espiral no es un modelo al cien por ciento exitoso por si solo, tiene como puntos claves que puede ser difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable y requiere una considerable habilidad para la valoración del riesgo, ya que si no se descubre un riesgo importante, indudablemente surgirán problemas.

La solución de los problemas presentados por las metodologías tradicionales se logra con una gran evolución del modelo espiral. El proceso unificado propone la elaboración de varios ciclos de desarrollo, donde cada uno finaliza con la entrega al cliente de un producto terminado. Este se enmarca entre los conocidos modelos iterativo incremental.

1.3 Proceso Unificado

El proceso unificado es un proceso de desarrollo de software. Es una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo) que tiene como objetivo asegurar la producción de software de calidad dentro del plazo y presupuesto definido, está dirigido por casos de uso, está centrado en la arquitectura y es iterativo (mini-proyectos) e incremental (versiones). Explicaré cada una de las características antes mencionadas:

Dirigido por casos de uso

El proceso unificado está dirigido por *casos de uso*, éstos describen la funcionalidad del sistema y se usan como un documento de entrega al cliente (artefacto) básico para establecer el comportamiento deseado del mismo. Además sirven para verificar y validar la arquitectura, realizar las pruebas y para tener una adecuada comunicación entre las personas involucradas en el desarrollo del software (proyecto).

Los desarrolladores crean modelos de diseño e implementación basándose en el modelo de casos de uso, revisan cada uno de los modelos creados para comprobar si corresponden con lo especificado. Igualmente los responsables de las pruebas del sistema comprueban la implementación para asegurarse que corresponde con lo establecido en los casos de uso. Los casos de uso no solo inician el proceso software sino que permanecen ligados a él durante todo el ciclo de vida.

Centrado en la arquitectura

La arquitectura del sistema se usa como artefacto básico para conceptualizar, construir, administrar y guiar el desarrollo del sistema. Nos da una perspectiva clara del sistema global, la cual es necesaria para controlar su desarrollo.

El papel de la arquitectura software, es el mismo del papel que juega la arquitectura en la construcción de un edificio. El edificio se ve desde distintos puntos de vista: estructura, calefacción, fontanería, electricidad, entre otras. Esto permite al constructor tener una visión completa del edificio antes de comenzar la construcción. De forma similar la arquitectura software se describe en función de las diferentes perspectivas del sistema a construir.

Enfocado en los riesgos

El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

Ciclo de vida iterativo e incremental del Proceso Unificado

Como ya lo habíamos visto en el modelo en espiral, un enfoque iterativo propone una comprensión incremental del problema a través de refinamientos sucesivos, permitiendo una flexibilidad para incorporar nuevos requisitos o cambios, identificar y resolver los riesgos lo más pronto posible y tener un crecimiento incremental de una solución efectiva a través de varios ciclos.

Como los proyectos software son largos es común dividir el trabajo en proyectos más pequeños, esto constituye una iteración completa que resulta en un incremento o versión del sistema. Las iteraciones se refieren al flujo del trabajo y deben de llevarse a cabo con una planeación anterior, de forma que cada una de las iteraciones constituye un mini proyecto de software.

El proceso unificado consta de cuatro fases:

- I. Inicio o concepción
- II. Elaboración
- III. Construcción
- IV. Transición

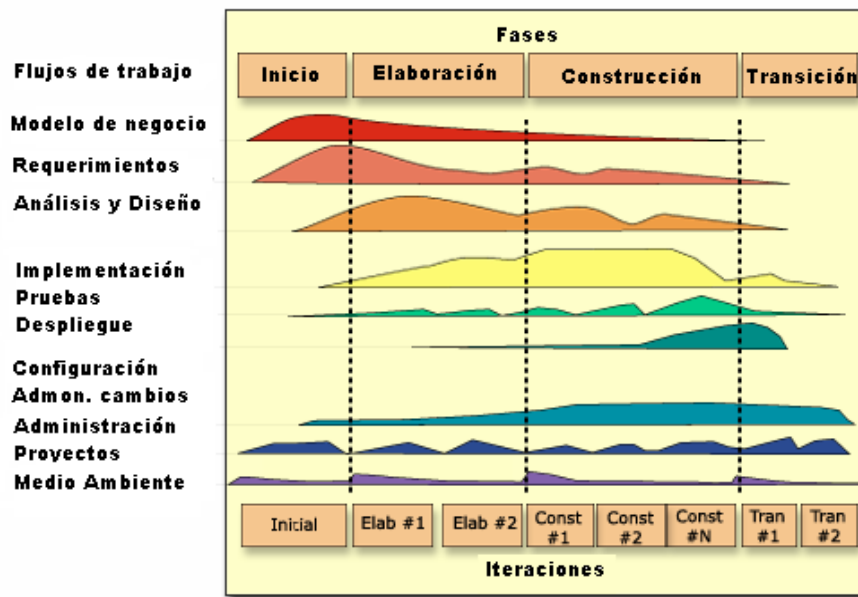


Figura 4 Diagrama general del proceso unificado

Cada fase a su vez se subdivide en iteraciones. Una iteración es un ciclo de vida completo, tiene una duración fija, aplica a todas las fases pero se usa principalmente en construcción. Para cada iteración se planea un conjunto de casos de uso o escenarios. Cada fin de una iteración se debe considerar un hito. Los casos de uso más importantes y riesgosos se atacan en las primeras iteraciones.

1.4 Fases del ciclo de vida del Proceso Unificado

I. Concepción o inicio

Objetivo: Determinar si el producto de software propuesto es económicamente viable.

Para determinar si un software propuesto es viable, se requiere hacer un estudio general de lo que necesitamos hacer, para comenzar se requiere conocer el dominio de la aplicación, después conocer el proceso de negocio del cliente y delimitar los objetivos y alcance del software que se construirá.

De manera general, durante la fase de concepción, se desarrolla una descripción del producto final a partir de una buena idea y se presenta el *análisis del negocio* para el producto. En esta fase se obtienen las principales funciones del sistema para los usuarios más importantes, un bosquejo de la arquitectura del sistema, el plan del proyecto y cuanto costará desarrollar el producto. Aquí se obtiene el modelo de casos de uso simplificado, conteniendo los casos de uso más críticos. La arquitectura muestra los subsistemas más importantes, se identifican y priorizan los riesgos, se planifica en detalle la fase de elaboración y se estima el proyecto de manera aproximada. Es una fase breve. Se considera que tiene una duración de 5 a 10% máximo del proyecto.

Algunas cuestiones que se deben resolver en esta etapa son:

- ¿El producto de software propuesto tiene una elevada relación costo beneficio?
- ¿Se puede entregar a tiempo el producto de software propuesto?
- ¿Qué riesgos acechan al desarrollar el producto de software propuesto y cómo pueden ser mitigados estos riesgos?

- ¿Los miembros del equipo de desarrollo tienen los conocimientos necesarios?

II. Elaboración

Objetivo: Afinar los requerimientos iniciales, afinar la arquitectura, vigilar los riesgos, afinar las prioridades y producir un plan de administración de proyecto de software.

Durante esta fase se identifican a detalle la mayoría de los casos de uso del producto y se afina la arquitectura del sistema. La arquitectura se expresa en forma de vistas de todos los modelos del sistema, los cuales juntos representan el sistema entero. Por tanto hay una vista arquitectónica del modelo de casos de uso, del modelo de análisis, del modelo de diseño, del modelo de implementación y del modelo de despliegue. Al final de esta fase ya se puede planificar las actividades y estimar los recursos necesarios para terminar el proyecto. Se estima que tiene una duración entre 25 y 30 % máximo del proyecto.

III. Construcción

Objetivo: Desarrollar de forma incremental una versión beta del sistema.

Durante esta fase se crea el producto. Al final de esta fase el producto contiene todos los casos de uso que se han acordado con el cliente.

Esta fase es un proceso de manufactura, programación de todos los artefactos de software, en el cual se debe poner énfasis en la administración de los recursos y el control de costos, agenda y calidad. Es la fase más prolongada de todas y en la que se realizan la mayoría de las iteraciones. Se estima una duración máxima de un 50% del proyecto.

IV. Transición

Objetivo: Capacitar al usuario, realizar pruebas de aceptación y liberar el sistema de producción.

Cubre el período durante el cual el producto se convierte en versión beta. Se prueba por usuarios con experiencia e informan de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas mejoras sugeridas.

Esta fase comienza cuando el producto está suficientemente maduro para ser entregado. Se llama transición porque se transfiere a las manos del usuario, pasando del entorno de desarrollo al de producción. Esta fase es clave para que la transición del producto se opere por el cliente que es en donde se debe hacer las recomendaciones a los clientes para que la organización adapte la nueva solución. Se estima que la duración es un 10% máximo del proyecto.

1.5 Internet y la World Wide Web

Internet (una red global de computadoras) se desarrolló hace más de cuatro décadas, y su patrocinio estuvo a cargo del Departamento de Defensa de los Estados Unidos. Diseñada en un principio para conectar los sistemas de cómputo principales de aproximadamente una docena de universidades y organizaciones de investigación, Internet es ahora utilizada por cientos de millones de computadoras en todo el mundo.

Con la introducción de World Wide Web (que permite a los usuarios de computadoras localizar y ver documentos basados en multimedia, sobre casi cualquier tema, a través de Internet), Internet se ha convertido explosivamente en uno de los principales mecanismos de comunicación en todo el mundo.

Internet y World Wide Web se encuentran entre las creaciones más importantes y profundas de la humanidad. En el pasado, la mayoría de las aplicaciones de computadora se ejecutaban en equipos que no estaban conectados entre sí. Las aplicaciones de la actualidad pueden diseñarse para comunicarse entre los cientos de millones de computadoras en todo el mundo. Internet mezcla las

tecnologías de la computación y las comunicaciones. Facilita nuestro trabajo. Hace que la información esté accesible en forma instantánea y conveniente para todo el mundo. Hace posible que los individuos y negocios pequeños locales obtengan una exposición mundial. Está cambiando la naturaleza de la manera en que se hacen los negocios. La gente puede buscar mejores precios para casi cualquier producto o servicio. Los miembros de las comunidades con intereses especiales pueden mantenerse en contacto unos con otros. Los investigadores pueden estar inmediatamente al tanto de los últimos descubrimientos.

Muchos de los servicios de Internet están basados en la arquitectura cliente-servidor.

1.6 Arquitectura Cliente-Servidor

Un servidor es una computadora que lleva a cabo un servicio que normalmente requiere mucha potencia de procesamiento. Un cliente es una computadora que solicita los servicios que proporciona uno o más servidores y que también lleva a cabo algún tipo de procesamiento por sí mismo pero en menor escala. El objetivo de esta arquitectura es equilibrar el proceso de una red hasta que se comparta la potencia de procesamiento entre computadoras que llevan a cabo servicios especializados tales como acceder a bases de datos (servidores), y aquellos que llevan a cabo tareas tales como la visualización.

Existen diferentes categorías de servidores, entre algunos de ellos tenemos:

Servidores de archivos. Se encarga de almacenar archivos de todo tipo para después proporcionarlos a los clientes.

Servidores de bases de datos. Los servidores de bases de datos son computadoras que almacenan grandes colecciones de datos estructurados, es decir, bases de datos.

Servidores WEB. Son los encargados de almacenar como páginas los documentos Web. Cuando se abre un documento Web existente, el servidor responde enviando una página al cliente que puede ser visualizada por un navegador. Para que exista esta comunicación se utiliza el protocolo conocido como HTTP.

Servidores de aplicaciones. Un servidor de aplicaciones se dedica a una aplicación única. Es el encargado de traducir la petición del cliente a una página entendible por el navegador.

Servidores de correo. Un servidor de correo gestiona el envío y recepción de correo de un grupo de usuarios. Existen varios protocolos para el correo electrónico pero un servidor de correo estará especializado en solo uno de ellos.

La comunicación entre el cliente y el servidor se lleva a través de al menos un software entre ellos, a este software se le llama middleware. Por ejemplo: el software para llevar a cabo comunicaciones utilizando el protocolo TCP/IP y otros protocolos de red, un software que permite a bases de datos diferentes conectarse a una red cliente-servidor. Probablemente el mejor ejemplo de este tipo de software sea la ODBC (Conectividad abierta de bases de datos) producida por Microsoft. Esta permite que existan felizmente en una red la vasta mayoría de sistemas de gestión de bases de datos.

1.7 Apache Tomcat

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

Tomcat es un servidor web con soporte de servlets y JSPs. no es un servidor de aplicaciones, como JBoss. Incluye el compilador Jasper, que compila JSPs convirtiéndolos en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo, en sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones, hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

La jerarquía de directorios de instalación de Tomcat incluye:

- bin** - arranque, cierre, y otros scripts y ejecutables
- common** - clases comunes que pueden utilizar Catalina y las aplicaciones web
- conf** - ficheros XML y los correspondientes DTD para la configuración de Tomcat
- logs** - logs de Catalina y de las aplicaciones
- server** - clases utilizadas solamente por Catalina
- shared** - clases compartidas por todas las aplicaciones web
- webapps** - directorio que contiene las aplicaciones web
- work** - almacenamiento temporal de ficheros y directorios

1.8 Java

La tecnología Java es tanto un lenguaje de programación y una plataforma.

El lenguaje de programación Java es un lenguaje de alto nivel que se caracteriza por todas las consignas siguientes:

- Simple
- Arquitectura neutral
- Orientado a objetos
- Portátil
- Distribuido
- Alto rendimiento
- Multiproceso
- Robusto
- Dinámico

En el lenguaje de programación Java, el código fuente es el primer texto escrito en los archivos que terminan con la extensión de .java. Los archivos de código fuente se traducen por el compilador javac que genera un archivo que no contiene código nativo del procesador que los genera, si no que en su lugar contiene *bytecodes* - el lenguaje de máquina de la máquina virtual de Java (JVM). Con una instancia de máquina virtual de java se ejecuta la aplicación. Esta representación se ilustra en la siguiente figura:



Figura 5 Una visión general del proceso de desarrollo de software

Debido a que la máquina virtual de Java está disponible para muchos sistemas operativos diferentes, el Archivo con extensión .class es capaz de ejecutarse en Microsoft Windows, en el sistema operativo Solaris TM (Solaris OS), Linux, o Mac OS (Fig 6). Algunas máquinas virtuales, tales como la máquina de la virtual Java HotSpot, se encarga de realizar pasos adicionales en tiempo de ejecución para dar a su solicitud un aumento de rendimiento. Esto incluye diversas tareas tales como encontrar los cuellos de

botella de rendimiento y volver a compilar (a código nativo), utilizan con frecuencia las secciones de código.

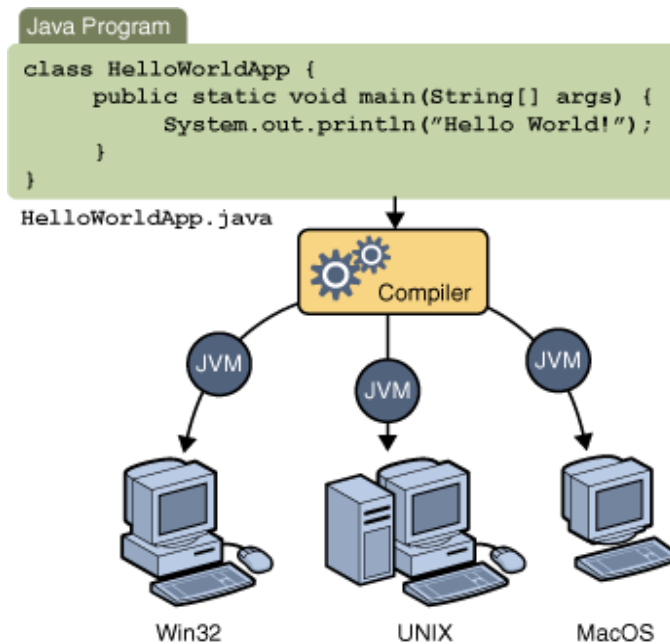


Figura 6 A través de la máquina virtual de Java, la misma aplicación es capaz de correr en múltiples plataformas.

Plataforma Java

Una *plataforma* es el hardware o el entorno de software en el que se ejecuta un programa. La mayoría de las plataformas pueden ser descritas como una combinación del sistema operativo y el hardware subyacente. La plataforma Java (Fig 7) difiere de la mayoría de las otras plataformas en que se trata de una única plataforma de software que se ejecuta en la parte superior de otro hardware basadas en las plataformas.

La plataforma Java tiene dos componentes:

- La *máquina virtual de Java*
- La *interfaz de programación de aplicaciones (API)* de Java

El API es una gran colección de componentes de software ya preparado que ofrecen muchas funciones útiles. Se agrupan en las bibliotecas de clases relacionadas e interfaces; estas bibliotecas son conocidas como *paquetes*.

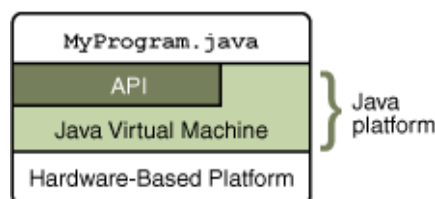


Figura 7 Estructura de la plataforma Java

Como plataforma de ambiente independiente, la plataforma Java puede ser un poco más lento que el código nativo. Sin embargo, los avances en el compilador y las tecnologías de máquina virtual están llevando rendimiento cercano al de código nativo sin poner en peligro la portabilidad.

Principales características

Simple

Las principales características del lenguaje de programación Java incluyen un lenguaje *sencillo* que puede ser programado sin formación de programador extensa, mientras se esté en sintonía con las prácticas de software actual. Los conceptos fundamentales de la tecnología Java son captados rápidamente y los programadores pueden ser productivos desde el principio.

Tecnología de Objetos en Java

Para ser verdaderamente considerados como "orientado a objetos", un lenguaje de programación deben apoyar a un mínimo de cuatro características:

Encapsulación. Ocultación de información y modularidad (abstracción)

Polimorfismo El mismo mensaje enviado a diferentes objetos, resulta en diferentes comportamientos de los objetos ya que es dependiente de la naturaleza del objeto que recibe el mensaje

Herencia - Definir nuevas clases y comportamientos basados en las clases existentes para rehusar código y organizar el código.

Enlace dinámico - objetos podrían provenir de cualquier lugar, posiblemente a través de la red. Tienes que ser capaz de enviar mensajes a los objetos sin tener que conocer su tipo específico en el momento de escribir el código.

Java cumple estos requisitos muy bien, y añade ejecutar considerable apoyo durante el tiempo para hacer su trabajo de desarrollo de software más fácil.

Arquitectura Neutral y portátil

La tecnología Java está diseñada para soportar aplicaciones que serán desplegados en entornos de red heterogéneos. En esos entornos, las aplicaciones deben ser capaces de ejecutar en una variedad de arquitecturas de hardware. Dentro de esta variedad de plataformas de hardware, las aplicaciones se deben ejecutar sobre una variedad de sistemas operativos e interoperar con interfaces de programación en varios lenguajes.

Para acomodar la diversidad de ambientes de funcionamiento, el producto del compilador de Java TM genera *bytecodes* - una *arquitectura de* formato intermedio *neutral* diseñado para transportar el código eficientemente a las plataformas de hardware y de software. La naturaleza de la tecnología Java de ser interpretado resuelve el problema de la distribución de binarios y el problema de la versión, el lenguaje de programación Java mismo códigos byte se pueden ejecutar en cualquier plataforma.

Arquitectura neutral es sólo una parte de un sistema verdaderamente *portátil*. La tecnología Java lleva la portabilidad un paso más allá al ser estrictos en su definición base del lenguaje. La tecnología Java define y especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos. Sus programas son los mismos en todas las plataformas - no hay incompatibilidades de tipo de datos a través de arquitecturas de hardware y software.

Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un

sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, Pc o Mac.

Rendimiento

El rendimiento es siempre una consideración. La plataforma Java logra un rendimiento superior mediante la adopción de un régimen por el cual el intérprete puede funcionar a plena velocidad, sin necesidad de comprobar el tiempo de ejecución de medio ambiente. El *recolector de basura automático* se ejecuta como un hilo de baja prioridad de fondo, garantizando una alta probabilidad de que la memoria está disponible cuando sea necesario, dando lugar a un mejor rendimiento. Las aplicaciones que requieren grandes cantidades de poder de cómputo pueden ser diseñadas de tal manera que calcular las secciones intensivas pueden ser reescritas en código de máquina nativo según sea necesario. En general, los usuarios perciben que las aplicaciones interactivas responder con rapidez a pesar de que son interpretados.

Interpretado

El *intérprete de Java* puede ejecutar bytecodes de Java directamente en cualquier máquina a la que el intérprete y el sistema en tiempo de ejecución han sido portados.

Distribuido

La tecnología Java está diseñada para funcionar en entornos distribuidos, lo que significa que la *seguridad* es de vital importancia. Con las características de seguridad diseñadas en el lenguaje y la ejecución del sistema en tiempo, la tecnología Java te permite construir aplicaciones que no se pueden invadir desde el exterior.

Java proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando, se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Robusto

Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.

También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java.

Además, para asegurar el funcionamiento de la aplicación, realiza una verificación de los byte-codes, que son el resultado de la compilación de un programa Java. Es un código de máquina virtual que es interpretado por el intérprete Java. No es el código máquina directamente entendible por el hardware, pero ya ha pasado todas las fases del compilador: análisis de instrucciones, orden de operadores, etc., y ya tiene generada la pila de ejecución de órdenes.

Multithreaded

Al ser multithreaded (multihilvanado, en mala traducción), Java permite muchas actividades simultáneas en un programa. Los threads (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los threads construidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++.

El beneficio de ser multithreaded consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.), aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento.

Cualquiera que haya utilizado la tecnología de navegación concurrente, sabe lo frustrante que puede ser esperar por una gran imagen que se está trayendo. En Java, las imágenes se pueden ir trayendo en un thread independiente, permitiendo que el usuario pueda acceder a la información en la página sin tener que esperar por el navegador.

Dinámico

Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las librerías nuevas o actualizadas no paralizarán las aplicaciones actuales (siempre que mantengan el API anterior).

Java también simplifica el uso de protocolos nuevos o actualizados. Si su sistema ejecuta una aplicación Java sobre la red y encuentra una pieza de la aplicación que no sabe manejar, tal como se ha explicado en párrafos anteriores, Java es capaz de traer automáticamente cualquiera de esas piezas que el sistema necesita para funcionar.

Java, para evitar que los módulos de byte-codes o los objetos o nuevas clases, haya que estar trayéndolos de la red cada vez que se necesiten, implementa las opciones de persistencia, para que no se eliminen cuando se limpie la caché de la máquina.

1.9 CMMI

En México existe un Centro de Excelencia para la Industria del Software (ESICenter), éste es parte de una red Mundial de ESICenters, y tiene como misión contribuir al desarrollo de la industria en México, Centro y Norte América, incrementando su competitividad a través de la difusión, la mejora continua y el conocimiento en tecnologías de información.

Cada ESICenter cuenta con 4 áreas de interés: Capacitación, Evaluaciones, Consultoría, e Investigación.

Las organizaciones exitosas han aprendido que aún los mejores profesionistas necesitan un ambiente organizado, disciplinado y estructurado y las organizaciones que no proveen dicho ambiente, arriesgan a su personal a interminables horas de trabajo para solucionar problemas.

El ESICenter se enfoca en la Mejora de Procesos, ya que ésta es parte fundamental en el desarrollo de un software, porque el esfuerzo y las capacidades de las personas pueden desperdiciarse si no hay un proceso productivo. Con un proceso bien definido, podemos reducir el número de defectos en productos entregados al cliente, nos permitiría también tener calendarios más realistas, incrementar la productividad de los equipos de desarrollo, y por lo tanto, mejorar el retorno a la inversión. Sin embargo, tenemos muchos problemas comunes en los procesos o tenemos la ausencia total de ellos, lo que da como consecuencia proyectos de desarrollo de software cancelados, productos entregados con gran cantidad de defectos a clientes, o se reescriben altas cantidades de líneas de código, y en más de los casos, entregas tardías.

Un Proceso puede ser definido como la secuencia de pasos realizados con un propósito específico, y para disminuir los costos de no conformidad (costos generados por lo clientes cuando rechazan el funcionamiento del software entregado) en la entrega de proyectos en el desarrollo de software, se recomienda establecer un proceso de mejora, con la cual mejorar la productividad e incrementar la satisfacción del cliente y del equipo de trabajo.

El modelo de CMMI (Modelo de Madurez de Capacidades Integradas) sucesor del Modelo CMM (Modelo de Madurez de Capacidades), se gesta a solicitud del Departamento de Defensa de los Estados Unidos, fue desarrollado por el Instituto de Ingeniería de Software (SEI) de la Universidad de Carnegie Mellon. Es un modelo o marco de referencia para el crecimiento de capacidades y madurez, es decir, se busca definir una capacidad (un proceso efectivo) y madurarlo (hacerlo constante y más eficaz).

El modelo de CMMI se basa en niveles de madurez, cada nivel de madurez es una capa en el camino a la madurez de la organización, cada nivel es la base del nivel superior y cada nivel superior incluye los atributos de los niveles inferiores (acumulativo).

Este marco de referencia pretende hacer que las organizaciones cuenten con un proceso de desarrollo de software capaz de mejorar el ciclo de vida en los proyectos de software, obtener mayor calidad, mayor productividad, aumentar la satisfacción del cliente, mejorar la moral del personal y reducir costos. Proporciona un enfoque disciplinado para mejorar los procesos de una organización.

Para alcanzar un nivel de madurez, el modelo de CMMI, descompone cada nivel en diferentes objetivos y los relaciona entre sí, a cada objetivo particular le llama Área de Proceso.

Un área de proceso es una agrupación de prácticas relacionadas entre sí, cuya ejecución permite alcanzar una serie de objetivos.

Las organizaciones no pueden ser certificadas CMMI. Por el contrario, una organización es evaluada (por ejemplo, usando un método de evaluación como SCAMPI) y recibe una calificación de nivel 1-5 si sigue los niveles de Madurez (si bien se comienza con el nivel 2). En caso de que quiera la organización, puede coger áreas de proceso y en vez de por niveles de madurez puede obtener los niveles de capacidad en cada una de las Áreas de Proceso, obteniendo el "Perfil de Capacidad" de la Organización.

CMMI identifica 25 áreas de procesos (22 en la versión que no integra IPD).

Vistas desde la representación continua (Fig. 8) del modelo, se agrupan en 4 categorías según su finalidad: Gestión de proyectos, Ingeniería, Gestión de procesos y Soporte a las otras categorías.

Vistas desde la representación escalonada (Fig. 8), se clasifican en los 5 niveles de madurez. Al nivel de madurez 2 pertenecen las áreas de proceso cuyos objetivos debe lograr la organización para alcanzarlo, ídem con el 3, 4 y 5.

| Área de proceso | Categoría | Nivel de madurez |
|---------------------------------------|--------------|------------------|
| Análisis y resolución de problemas | Soporte | 5 |
| Gestión de la configuración | Soporte | 2 |
| Análisis y resolución de decisiones | Soporte | 3 |
| Gestión integral de proyecto | G. Proyectos | 3 |
| Gestión integral de proveedores | G. Proyectos | 3 |
| Gestión de equipos | G. Proyectos | 3 |
| Medición y análisis | Soporte | 2 |
| Entorno organizativo para integración | Soporte | 3 |

| | | |
|--|--------------|---|
| Innovación y desarrollo | G. Procesos | 5 |
| Definición de procesos | G. Procesos | 3 |
| Procesos orientados a la organización | G. Procesos | 3 |
| Rendimiento de los procesos de la org. | G. Procesos | 4 |
| Formación | G. Procesos | 3 |
| Integración de producto | Ingeniería | 3 |
| Monitorización y control de proyecto | G. Proyecto | 2 |
| Planificación de proyecto | G. Proyecto | 2 |
| Gestión calidad procesos y productos | Soporte | 2 |
| Gestión cuantitativa de proyectos | G. Proyectos | 4 |
| Desarrollo de requisitos | Ingeniería | 3 |
| Gestión de requisitos | Ingeniería | 2 |
| Gestión de riesgos | G. Proyectos | 3 |
| Gestión y acuerdo con proveedores | G. Proyectos | 2 |
| Solución técnica | Ingeniería | 3 |
| Validación | Ingeniería | 3 |
| Verificación | Ingeniería | 3 |

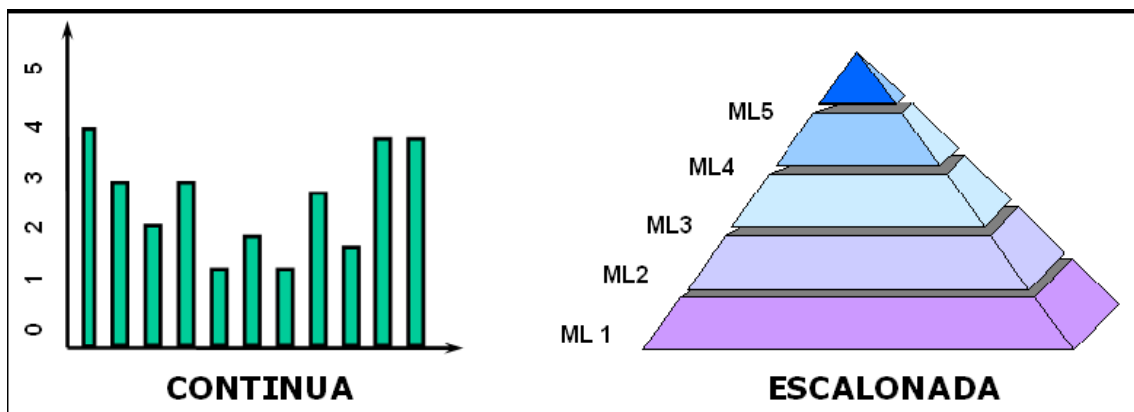


Figura 8 Representación continua y escalonada

Niveles de Madurez (Modelo escalonado)

Nivel 1. INICIAL

Todas las empresas dedicadas al desarrollo de software comienzan en esta etapa. Esta etapa se define por la implantación de procesos, pero no todas las personas de la empresa conocen, y mucho menos aplican estos procesos. No son procesos bien definidos y los resultados dependen de la competencia y de esfuerzos heroicos de las personas. Es posible que se consiga calidad pero es necesario asignar a las mejores personas para cada tarea. Y lo más importante es que es difícil predecir resultados.

Nivel 2. ADMINISTRADO

En este nivel se establecen y se siguen políticas organizacionales (aun en tiempos de estrés), los recursos son adecuados, se asignan responsabilidades y autoridades a lo largo de la vida de los proyectos, existe una administración disciplinada de proyectos, los éxitos de proyectos anteriores pueden ser repetidos en nuevos proyectos y la dirección tiene la visibilidad de las actividades y productos en puntos definidos. En el nivel 2 se definen 7 Áreas de Proceso.

Nivel 3. DEFINIDO

Este nivel se construye sobre los cimientos de la administración de proyectos establecida en el nivel 2, es decir, ya se tiene un proceso definido para todo el ciclo de vida de un proyecto, los procesos definidos y documentados pueden ser aplicados por cualquier persona de la organización, incluso por una nueva persona y con menor tiempo en su curva de aprendizaje, los procesos siguen en mejora continua y pueden ser adaptados a cualquier proyecto, se identifican y resuelven las necesidades de capacitación, en pocas palabras se cuenta con una organización más productiva. En el nivel 3 se definen 14 Áreas de Proceso.

Nivel 4. ADMINISTRADO CUANTITATIVAMENTE

En este nivel, se utilizan datos con técnicas estadísticas y técnicas cuantitativas para la administración y control de los procesos. Se utilizan dichas técnicas en toda la organización para entender los resultados pasados, predecir los futuros y mejorar el desempeño de los procesos. En el nivel 4 se definen 2 Áreas de Proceso.

Nivel 5. OPTIMIZADO

En este nivel se identifican, evalúan e implantan mejoras a los procesos y tecnologías que objetivamente incrementan las capacidades de los procesos (mejora continua). En el nivel 5 se definen 2 Áreas de Proceso.

1.10 UML

Definición

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 específica y por las reglas semánticas relacionadas para la construcción de sistemas de software, soporta un conjunto rico en elementos de notación gráficos. El UML en sí mismo no prescribe ni aconseja cómo usar esta notación en el proceso de desarrollo o como parte de una metodología de diseño orientada a objetos. Describe la notación para clases, componentes, nodos, actividades, flujos de trabajo, casos de uso, objetos, estados y cómo modelar la relación entre esos elementos.

El UML provee beneficios significativos para los ingenieros de software y las organizaciones al ayudarles a construir modelos rigurosos, trazables y mantenibles, que soporten el ciclo de vida de desarrollo de software completo.

Elementos

Interfaces

Las interfaces son clases abstractas con sólo comportamientos. Especifican el contrato operacional que un implementador acepta satisfacer. Por ejemplo, se puede definir una interfaz llamada *IPerson* con los métodos para acceder a la dirección (*Address*) y a la Edad (*Age*) como en el ejemplo a continuación:

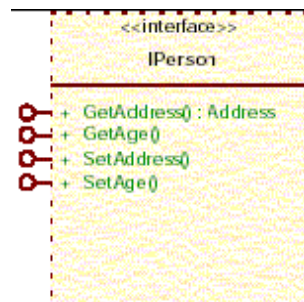


Figura 9 Interface

La interfaz no se puede instanciar en tiempo de ejecución; es un elemento abstracto. En su lugar, se puede definir una clase que 'implemente' esta interfaz. De esta forma, la clase se obliga a proveer implementaciones de los métodos de la interfaz con la firma exacta que se especifica en la definición de la Interfaz. Esto permite que numerosas clases de diferentes jerarquías y con diferentes superclases y diferentes comportamientos sean tratadas todas como del tipo *IPerson* (si implementan la interfaz) en tiempo de ejecución.

Relaciones

Los elementos lógicos se pueden relacionar en una gran variedad de maneras. Las siguientes son las más comunes:

- **Asociación**

Las relaciones de asociación capturan las relaciones estáticas entre las entidades. Generalmente se relacionan a un objeto que tiene una instancia de otro como atributo o estando relacionado en términos de posesión (pero no estando compuesto de). Por ejemplo, un *Vendedor* tiene una asociación a sus *Cientes* (como en el dibujo de más abajo), pero un *Vendedor* no está 'compuesto de' *Cientes*. Ambos extremos del enlace de la asociación pueden asumir un 'rol' como parte de la asociación; por ejemplo el

Vendedor puede asumir el rol de 'Vendedor de alfombras' si hay condiciones especiales que impliquen este rol.



Figura 10 Asociación

- **Agregación**

Las relaciones de agregación definen relaciones todo/parte. Por ejemplo, un viaje en tren puede estar compuesto por varios tramos del viaje. Así, el viaje completo agrega o está compuesto de tramos separados. La forma más fuerte de agregación es la composición e infiere que una clase no sólo colecciona a otra clase sino que realmente está compuesta de esta colección. El ejemplo de más abajo muestra que una tienda (*Store*) agrega o reúne una cantidad de departamentos (*Departments*) y que hay una relación todo/parte; es decir, una tienda está compuesta de (in partes) esta colección de departamentos.



Figura 11 Agregación

- **Herencia**

La herencia describe la relación jerárquica entre las clases. Describe el 'árbol genealógico'. Las clases pueden heredar los atributos y el comportamiento de una clase padre (que, a su vez, puede ser el hijo de otra clase). Este árbol de características y comportamiento heredados brinda al diseñador la posibilidad de coleccionar la funcionalidad común en clases raíces (ancestros) y refinar y especializar este comportamiento en una o más generaciones (hijos). Los especificadores de alcance (público, protegido y privado) determinan qué elementos se pueden heredar y cuáles no son visibles. El siguiente diagrama muestra un nivel de herencia:

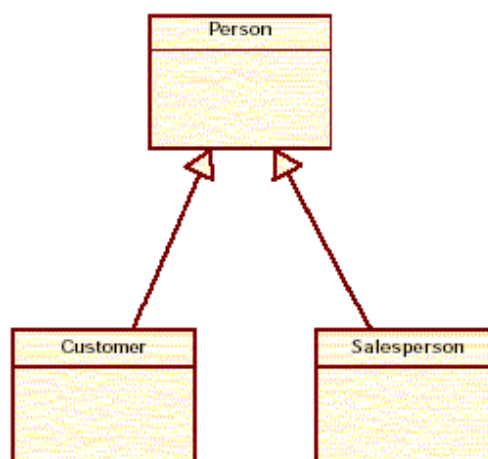


Figura 12 Herencia

• **Dinámicas**

Las relaciones dinámicas son los mensajes que se pasan entre las clases (los objetos durante el tiempo de ejecución). Un diagrama de secuencias ilustra este paso de mensajes y la secuencia en la que ocurre. El dibujo de más abajo muestra la secuencia de mensajes que se pasan a lo largo del tiempo cuando un usuario ingresa a una tienda de libros en línea. Estos elementos del modelo tienen una asociación entre sí que es reflejada por el paso de mensajes en tiempo de ejecución.

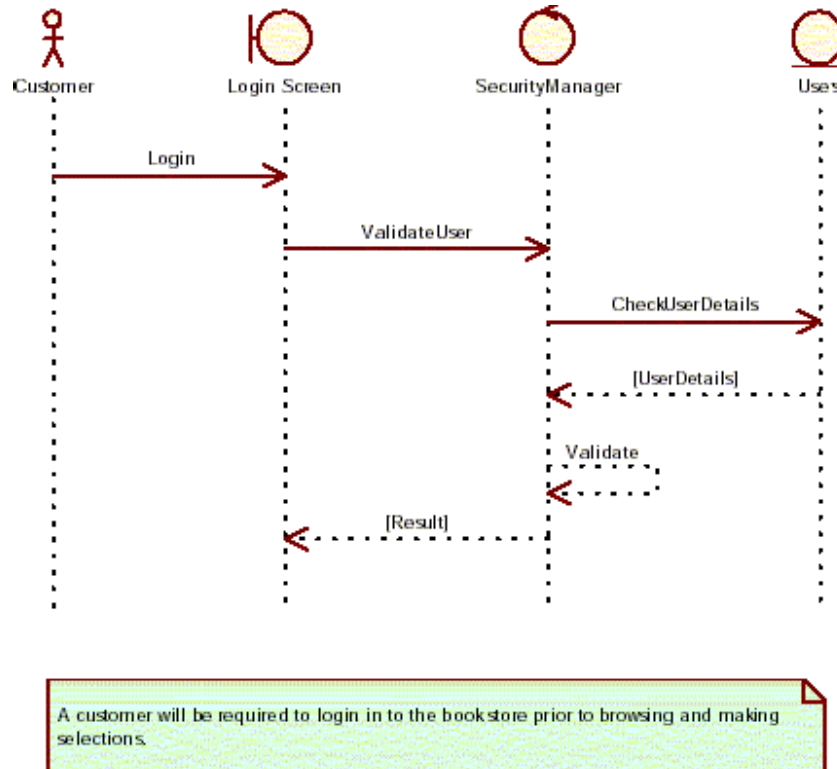


Figura 13 Diagrama de secuencia

Un elemento del modelo lógico y los atributos, asociaciones y operaciones que posee pueden ser especificados mucho más con las restricciones, los requisitos y los escenarios.

Restricciones

Son las reglas contractuales que se aplican a un elemento o a sus características. Típicamente caen en uno de tres tipos:

1. Pre-condiciones; deben ser verdaderas antes de que opere o de que exista un objeto
2. Post-condiciones; deben ser verdaderas luego del uso o de la destrucción de un objeto
3. Invariantes; deber ser siempre verdaderas para la vida y uso de un objeto

Modelo dinámico

El modelo dinámico se usa para expresar y modelar el comportamiento del sistema a lo largo del tiempo.

Modelo de Procesos de negocio

Un proceso de negocio:

1. Tiene un objetivo

2. Tiene entradas específicas
3. Tiene salidas específicas
4. Emplea recursos
5. Tiene un número de actividades que se llevan a cabo en algún orden
6. Puede afectar más de una unidad organizacional. Impacto organizacional horizontal
7. Crea valores de algún tipo para el cliente. El cliente puede ser interno o externo.

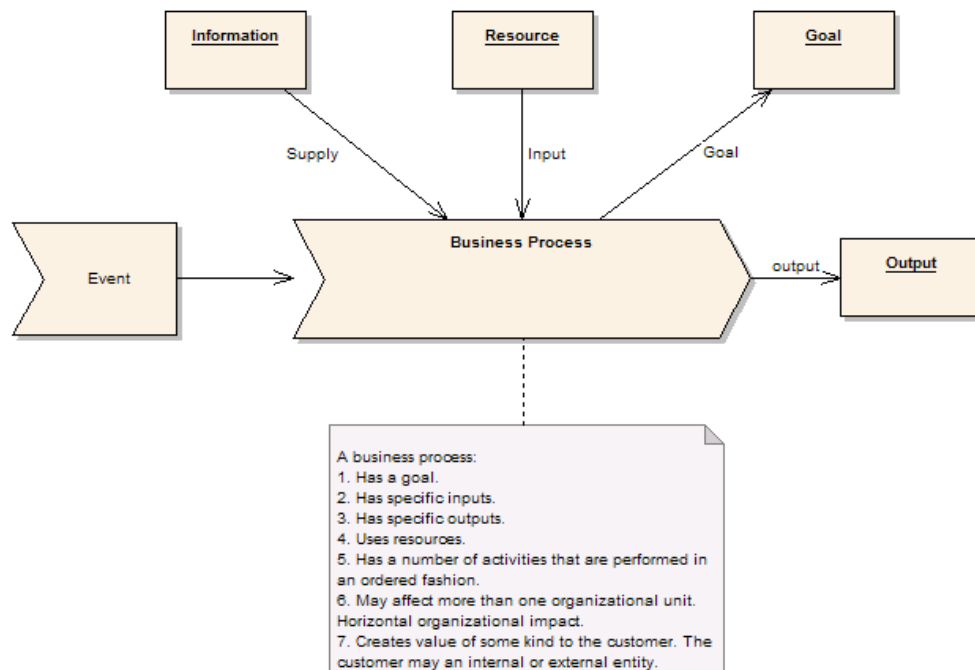


Figura 14 Modelo de negocio

Un proceso de negocio es una colección de actividades diseñadas para producir una salida específica para un cliente o mercado particular. Implica un fuerte énfasis en 'cómo' se hace el trabajo en una organización, en contraposición al enfoque en 'qué' de producto. Así, un proceso es un ordenamiento específico de actividades de trabajo a través del tiempo y del espacio, con un comienzo, un fin, entradas y salidas claramente identificados: una estructura para la acción.

Conexiones

- Vínculo de provisión desde el objeto Información. Un vínculo de provisión indica que la información u objeto vinculado al proceso no se utiliza en la fase de procesamiento. Por ejemplo, se pueden utilizar plantillas de órdenes una y otra vez para proveer nuevas órdenes de un cierto estilo -las plantillas no se alteran ni se consumen como parte de la actividad-
- Vínculo de provisión desde el objeto Recurso. Un vínculo de entrada indica que el objeto o recurso adjunto se consume durante el procedimiento del proceso. Como ejemplo, a medida que se procesan las órdenes de un cliente se completan y firman y típicamente se utiliza sólo una vez por cada recurso único (orden).
- Vínculo de objetivo al objeto Objetivo. Un vínculo de objetivo indica que el objeto adjunto al proceso de negocio describe su objetivo. Un objetivo es la justificación de negocio para desarrollar la actividad.
- Vínculo de flujo de estados al objeto Salida

- Vínculo de flujo de estados desde el evento Event. Un vínculo de flujo de estado indica que se pasa algún objeto al proceso de negocio. Captura el paso de control a otra entidad o proceso, con el paso implícito del estado o de información desde una actividad a otra.

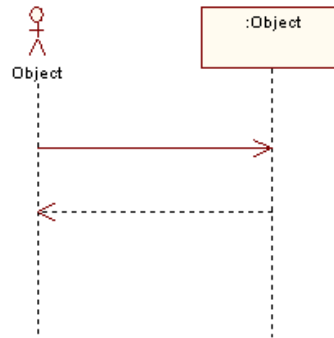


Figura 15 Flujo de Trabajo

Diagrama de Actividades

En UML un diagrama de actividades se usa para mostrar la secuencia de actividades. Los diagramas de actividades muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Estos también pueden usarse para detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas actividades. Los Diagramas de Actividades son útiles para el Modelado de Negocios donde se usan para detallar el proceso involucrado en las actividades de negocio.

Un ejemplo de un diagrama de actividades se muestra a continuación:

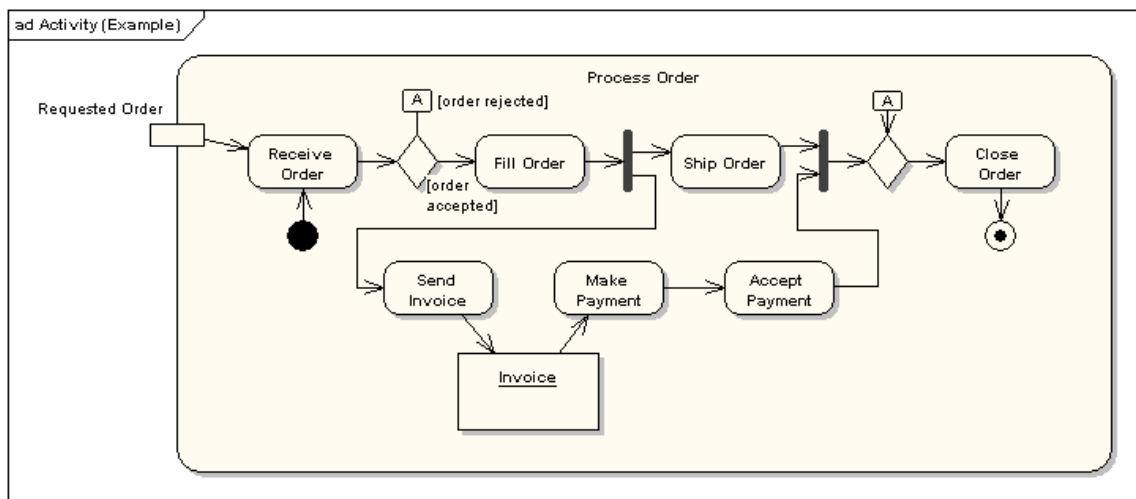


Figura 16 Diagrama de actividad

Diagrama de Secuencia

El UML provee un medio gráfico para representar la interacción entre los objetos a lo largo del tiempo en los diagramas de secuencia. Éstos muestran típicamente a un usuario o a un actor y los objetos y componentes con los que interactúen durante la ejecución de un Caso de Uso. Un diagrama de secuencia representa típicamente un único escenario de Caso de Uso o flujo de eventos.

Los diagramas son una vía excelente para documentar los escenarios de uso, para capturar los objetos necesarios de manera temprana en el análisis y para verificar el uso de los objetos más tarde en el diseño. Los diagramas de secuencia muestran el flujo de mensajes de un objeto a otro y, como tales, representan los métodos y los eventos soportados por un/a objeto/clase.

El diagrama ilustrado abajo muestra un ejemplo de un diagrama de secuencia, con el usuario o actor a la izquierda iniciando un flujo de eventos y mensajes que corresponden al escenario del caso de uso. Los mensajes que pasan entre objetos se convertirán en operaciones de clases en el modelo final.

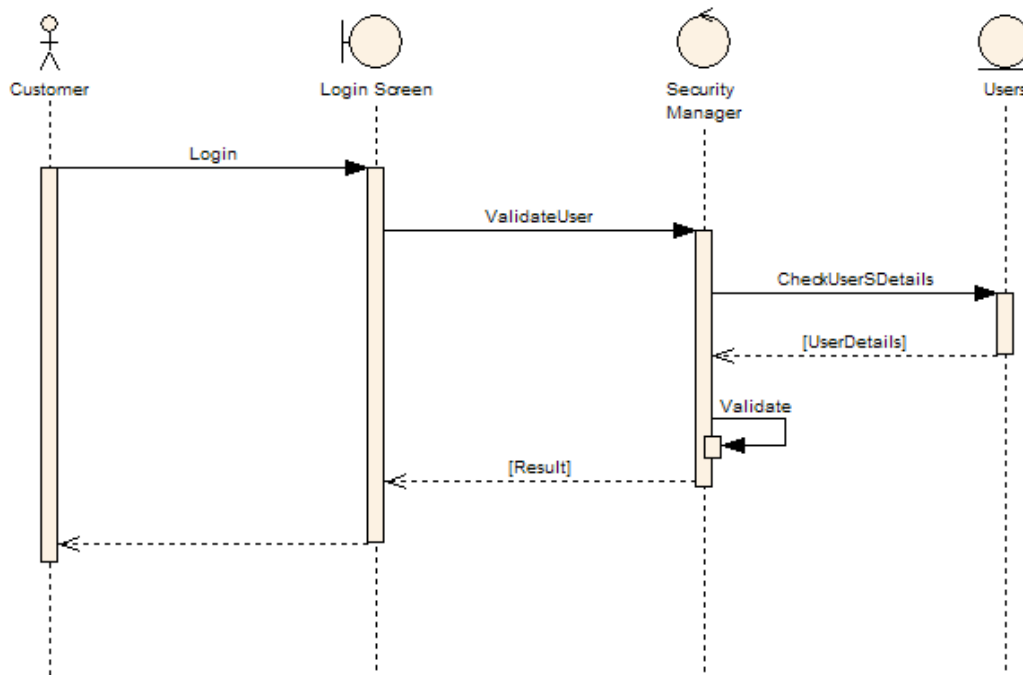


Figura 17 Diagrama de secuencia

Modelo lógico

Un modelo lógico es una vista estática de los objetos y las clases que cubren el espacio de análisis y diseño. Típicamente, un modelo de dominio es una vista más pobre, de alto nivel de los objetos de negocio y de las entidades, mientras que el modelo de clases es un modelo más riguroso y enfocado al diseño. Esta discusión describe principalmente el modelo de clases.

El Modelo de Clases

Una clase es un elemento estándar del UML, que se usa para especificar el patrón del que se producirán los objetos en tiempo de ejecución. Una clase es una especificación; un objeto es una instancia de una clase. Las clases se pueden heredar de otras clases (es decir, heredan todo el comportamiento y el estado de sus padres y agregan nueva funcionalidad propia), pueden tener otras clases como atributos, pueden delegar sus responsabilidades a otras clases e implementar interfaces abstractas.

El modelo de clases está en el núcleo del desarrollo y del diseño orientados a objetos; expresa el estado persistente y el comportamiento del sistema. Una clase encapsula el estado (los atributos) y ofrece los servicios para manipularlo (el comportamiento). Un buen diseño orientado a objetos limita el acceso directo a los atributos de la clase y ofrece los servicios que manipulan a solicitud del solicitante. Este ocultamiento de los datos y exposición de los servicios asegura que las modificaciones de los datos se realizan sólo en un lugar y de acuerdo con reglas específicas; para grandes sistemas la cantidad de

código que tiene acceso directo a los elementos de datos en muchos sitios es extremadamente alto.

Las clases se representan usando la siguiente notación:

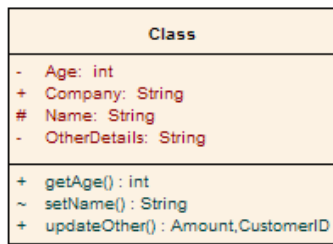


Figura 18 Clase

Observe que la clase tiene tres áreas diferentes:

1. El nombre de la clase (y el estereotipo, si corresponde)
2. El área de los atributos (es decir los elementos de datos internos)
3. El comportamiento; privado y público

Los atributos y los métodos pueden ser marcados como:

- Privados (private), indicando que no son visibles para los solicitantes fuera de la clase
- Protegidos (protected), son visibles sólo para las clases hijas
- Públicos (public), son visibles para todos.

La herencia mostrada a continuación consiste en: una clase abstracta en este caso, es el padre de dos clases hijos, las cuales heredan las características y extienden sus comportamientos.

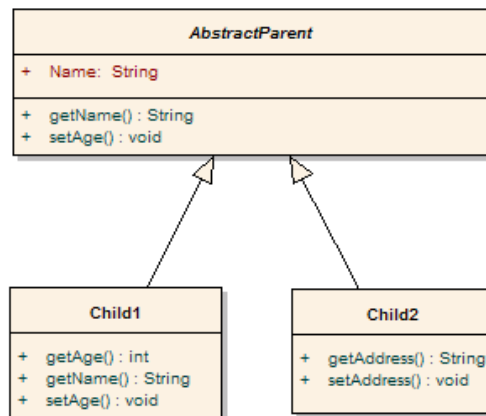


Figura 19 Herencia

Las clases se pueden agrupar en unidades lógicas o paquetes. Ellos juntan elementos relacionados entre sí. El diagrama siguiente ilustra esto.

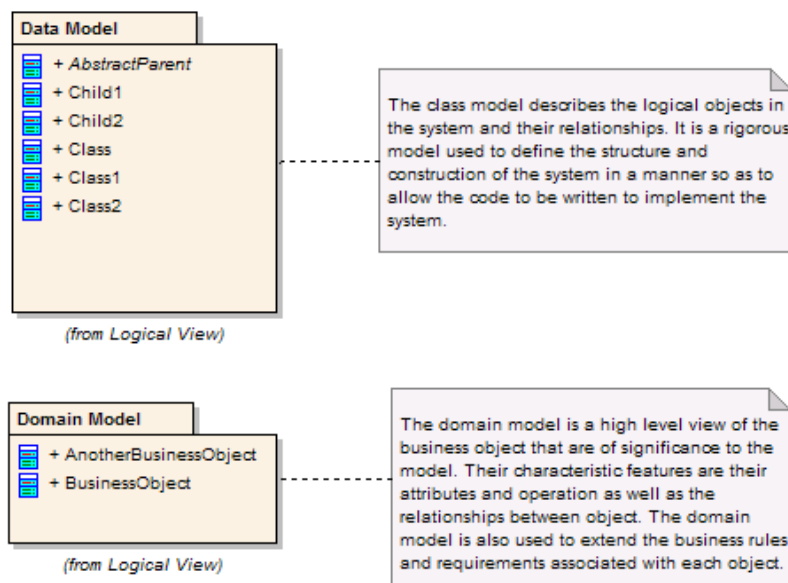


Figura 20 Paquetes

Modelo físico

El Modelo Físico/de Despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto.

Vista de Despliegue

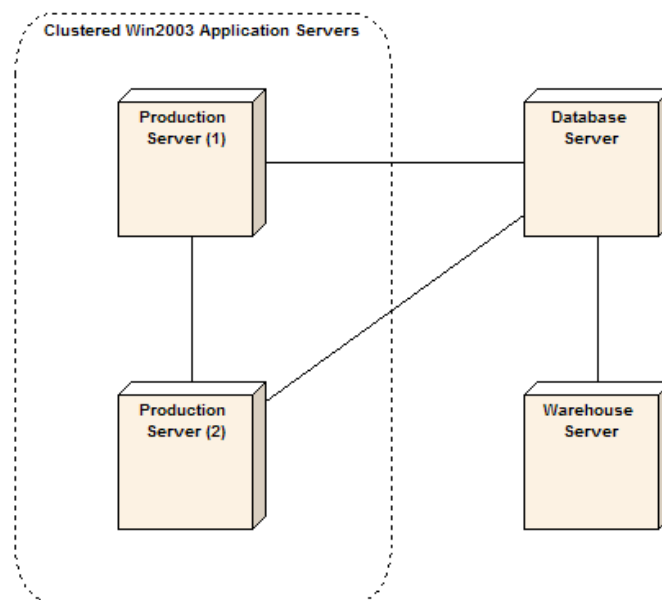


Figura 21 Vista de despliegue

El modelo físico muestra dónde y cómo se desplegarán los componentes. Es un mapa específico de la

instalación física del sistema. Un diagrama de despliegue ilustra el despliegue físico del sistema en un ambiente de producción (o prueba). Muestra dónde se ubicarán los componentes, en qué servidores, máquinas o hardware. Puede ilustrar vínculos de red, ancho de banda de LAN, etc.

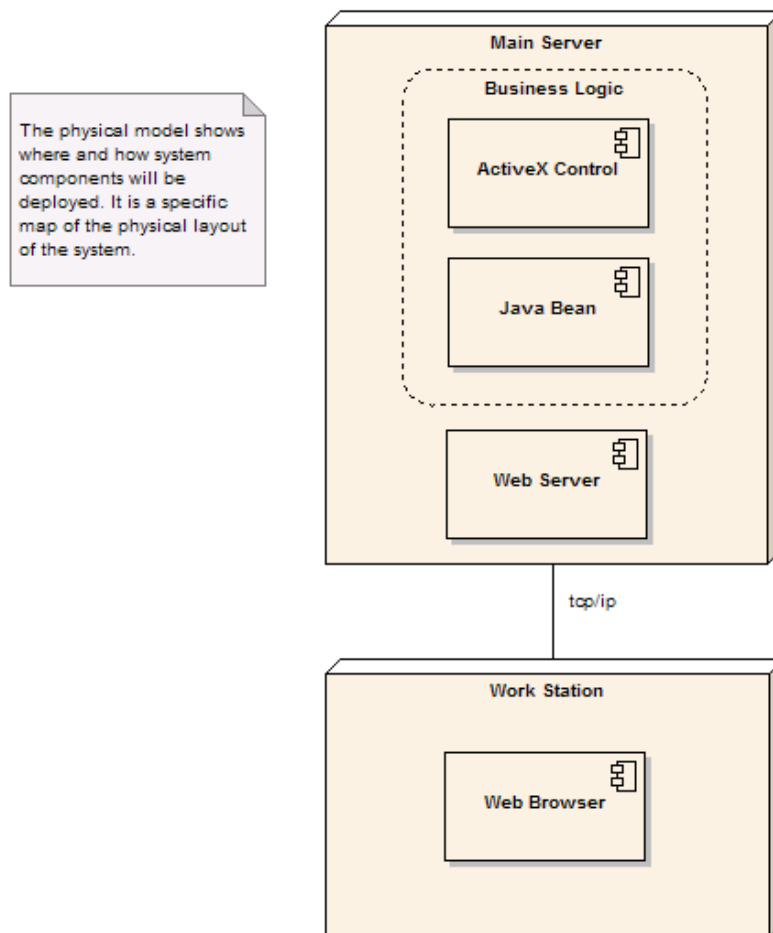


Figura 22 Diagrama de despliegue

Se utiliza un nodo para identificar cualquier servidor, terminal de trabajo u otro hardware host que se utiliza para desplegar componentes en el ambiente de producción.

Modelo de Casos de Uso.

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Un Caso de Uso es una unidad simple de trabajo significativo; por ejemplo, "Validarse en el sistema", "Registrarse en el sistema" y "Crear un pedido" son todos casos de uso.

Cada caso de uso tiene una nota que describe la funcionalidad que se construirá en el sistema propuesto. Un caso de uso puede "incluir" la funcionalidad de otro caso de uso o "extender" a otro caso de uso con su propio comportamiento.

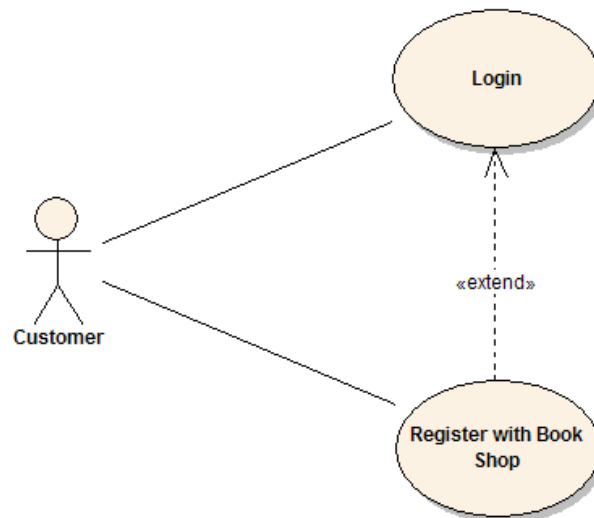


Figura 23 Diagrama de casos de uso

Una descripción de caso de uso generalmente incluirá:

- Comentarios generales y notas describiendo el caso de uso
- Requisitos -cosas que el caso de uso debe permitir hacer al usuario, tales como <capacidad para actualizar pedido>, <capacidad para modificar pedido>, etc.
- Restricciones -reglas acerca de qué se puede y qué no se puede hacer-. Incluye:
 - Pre-condiciones que deben ser verdaderas antes de que el caso de uso se ejecute, por ejemplo <crear pedido> debe preceder a <modificar pedido>
 - Post-condiciones que deben ser verdaderas una vez que el caso de uso se ejecutó, por ejemplo <el pedido está modificado y es consistente>
 - invariantes: éstas son siempre verdaderas -por ejemplo, un pedido debe tener siempre un número de cliente.
- Escenarios -descripciones secuenciales de los pasos que se toman para llevar a cabo el caso de uso. Pueden incluir escenarios múltiples, para satisfacer circunstancias excepcionales y caminos de proceso alternativos
- Diagramas de escenarios -diagramas de secuencia para describir el flujo de trabajo- similar al punto 4 pero descrito gráficamente.
- Atributos adicionales como fase de implementación, número de versión, rango de complejidad, estereotipo y estado

1.11 Arquitectura Modelo-Vista-Controlador (MVC)

El diagrama muestra la relación entre el modelo, la vista y el controlador. Las líneas sólidas indican una asociación directa, y las punteadas una indirecta.

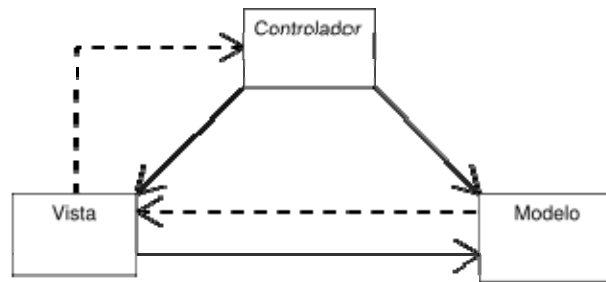


Figura 24 Arquitectura MVC

Definición de las partes

El **Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones (lógica del negocio). El **Modelo** no tiene conocimiento específico de los **Controladores** o de las **Vistas**, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La **Vista** es el objeto que maneja la presentación visual de los datos representados por el **Modelo**. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El **Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el **Modelo**. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista.

En una aplicación web la vista es la página HTML, el CSS y todo lo que el usuario vea en la pantalla. El controlador es el código que provee de datos dinámicos a la página, es el que modifica la vista y el modelo según sea necesario. Y el modelo contiene clases representativas de la aplicación o sea los datos de la base de datos.

Esta arquitectura ha demostrado ser muy apropiada para las aplicaciones web y especialmente adaptarse bien a las tecnologías proporcionadas por la plataforma J2EE, de manera que:

- El modelo, conteniendo lógica de negocio, sería modelado por un conjunto de clases Java, existiendo dos claras alternativas de implementación, utilizando objetos Java tradicionales llamados POJOs (*Plain Old Java Objects*) o bien utilizando EJB (*Enterprise JavaBeans*) en sistemas con unas mayores necesidades de concurrencia o distribución.
- La vista proporcionará una serie de páginas web dinámicamente al cliente, siendo para él simples páginas HTML. Existen múltiples frameworks que generan estas páginas web a partir de distintos formatos, siendo el más extendido el de páginas JSP (*JavaServer Pages*), que mediante un conjunto de tags XML proporcionan un interfaz sencillo y adecuado a clases Java y objetos proporcionados por el servidor de aplicaciones. Esto permite que sean sencillas de desarrollar por personas con conocimientos de HTML. Entre estos tags tienen mención especial la librería estándar JSTL (*JavaServer Pages Standard Tag Library*) que proporciona una gran funcionalidad y versatilidad.
- El controlador en la plataforma J2EE se desarrolla mediante servlets, que hacen de intermediarios entre la vista y el modelo, más versátiles que los JSP para esta función al estar escritos como clases Java normales, evitando mezclar código visual (HTML, XML...) con código Java. Para facilitar la implementación de estos servlets también existe una serie de frameworks que proporcionan soporte a los desarrolladores, entre los que cabe destacar Struts, que con una amplia comunidad de usuarios se ha convertido en el estándar *de facto* en este rol.

Con todo lo anterior, el funcionamiento de una aplicación web J2EE que utilice el patrón arquitectural MVC se puede descomponer en una serie de pasos:

1. El usuario realiza una acción en su navegador, que llega al servidor mediante una petición HTTP y es recibida por un servlet (controlador). Esa petición es interpretada y se transforma en la ejecución de código java que delegará al modelo la ejecución de una acción de éste.
2. El modelo recibe las peticiones del controlador, a través de un interfaz o fachada que encapsulará y ocultará la complejidad del modelo al controlador. El resultado de esa petición será devuelto al controlador.
3. El controlador recibe del modelo el resultado, y en función de éste, selecciona la vista que será mostrada al usuario, y le proporcionará los datos recibidos del modelo y otros datos necesarios para su transformación a HTML. Una vez hecho esto el control pasa a la vista para la realización de esa transformación.
4. En la vista se realiza la transformación tras recibir los datos del controlador, elaborando la respuesta HTML adecuada para que el usuario la visualice.

Esta arquitectura de aplicaciones otorga varias ventajas clave al desarrollo de aplicaciones web, destacando:

- Al separar de manera clara la lógica de negocio (modelo) de la vista permite la reusabilidad del modelo, de modo que la misma implementación de la lógica de negocio que maneja una aplicación pueda ser usado en otras aplicaciones, sean éstas web o no.
- Permite una sencilla división de roles, dejando que sean diseñadores gráficos sin conocimientos de programación o desarrollo de aplicaciones los que se encarguen de la realización de la capa vista, sin necesidad de mezclar código Java entre el código visual que desarrollen (tan sólo utilizando algunos tags, no muy diferentes de los usados en el código HTML).

1.12 Estimación por Puntos de Casos de Uso.¹

El método de Puntos de Casos de Uso (UCP), está basado en los tradicionales Puntos Función. Es un método originado de la tesis de master de Gustav Karner, bajo supervisión de Ivar Jacobson (creador de los casos de uso). La técnica ha sido usada por la empresa Rational (posteriormente adquirida por IBM) durante varios años y con buenos resultados.

En esencia, la técnica de Karner es similar a las técnicas de Punto Función en las que se realizan las siguientes actividades:

- Se deben revisar los aspectos clave de los requerimientos para calcular un recuento de Puntos Caso de Uso sin ajustar (UUCP).
- Estudiar los factores técnicos y el entorno para crear los factores de ajuste.
- Ajustar los factores para llegar a obtener los Puntos Caso de Uso ajustados (UCP), que posteriormente se transformarán en una estimación de esfuerzo (horas/hombre).

1. Cálculo de los puntos caso de uso sin ajustar (UUCP)

¹ Fuente: www.kybeleconsulting.com

Para realizar el cálculo de los Puntos Caso de Uso sin ajustar, se tienen que realizar los tres pasos definidos a continuación.

2. Clasificar cada interacción entre actor y caso de uso según su complejidad y asignarle un peso.

Para clasificar la complejidad de los actores se debe determinar la forma en la que cada actor interactúa con el sistema que se va a desarrollar. En concreto, los actores se clasifican en 3 categorías diferentes, simple, medio y complejo. Un actor simple representa otro sistema con una API definida, un actor medio es otro sistema que interactúa a través de un protocolo como por ejemplo TCP/IP o es una persona interactuando a través de una interfaz por línea de comandos, y un actor complejo interactúa a través de una interfaz gráfica.

Una vez clasificado cada actor según su tipo de interacción, se le asigna el peso correspondiente asociado a dicha interacción. En la Tabla 1, se presenta un resumen del procedimiento de clasificación de los actores.

| Tipo de interacción | Peso |
|---|------|
| Simple (a través de un API) | 1 |
| Medio (a través de un protocolo) | 2 |
| Complejo (a través de una interfaz gráfica) | 3 |

Tabla 1. Clasificación de los Actores

3. Calcular la complejidad de cada caso de uso según el número de transacciones o pasos del mismo.

Para realizar el cálculo de la complejidad de un caso de uso se debe determinar el número de transacciones, incluyendo los caminos alternativos. Una transacción es un conjunto de actividades atómicas, donde se ejecutan todas ellas o ninguna. En este contexto, cada caso de uso se debe clasificar en una de las siguientes categorías: "simple", "medio" o "complejo". En concreto, un caso de uso simple tiene 3 o menos transacciones, un caso de uso medio de 4 a 7 transacciones, y un caso de uno complejo más de 7 transacciones.

Una vez clasificado cada caso de uso, según el número de transacciones, se le asigna el peso asociado a dicho número de transacciones. En la Tabla 2 se presenta un resumen del procedimiento de clasificación de los casos de uso.

| Tipo de Caso de Uso | Numero de Transacciones | Peso |
|---------------------|-------------------------|------|
| Simple | 3 o menos | 1 |
| Medio | De 4 a 7 | 2 |
| Complejo | 7 o más | 3 |

Tabla 2. Clasificación de los Casos de Uso

4. Calcular los Puntos Caso de Uso no ajustados (UUCP)

Los UUCP se calculan sumando la dificultad de las interacciones y la complejidad de los casos de uso, es decir, sumando el total de los pesos de los actores (clasificados en el paso 1) y el total de los pesos para los casos de uso (clasificados en el paso 2). Ejemplo:

- 2 interacciones por Web: $2 * 3 = 6$

- _ 4 UCP complejos: $4 * 3 = 12$
- _ UUCP = $6 + 12 = 18$

5. Cálculo de los factores técnicos (TCF)

Para ajustar los UUCP (Puntos Caso de Uso no ajustados) calculados en los pasos anteriores, se deben tener en cuenta factores de ajuste, tanto factores técnicos, como factores de entorno.

En el caso de los factores técnicos (TCF), a cada factor definido en la Tabla 3 (Ri) se le asigna un valor entre 0 y 5, dependiendo de su influencia en el proyecto. En este sentido, asignar un valor 0 significa que el factor es irrelevante para el proyecto, un valor 3 es promedio y un valor 5 significa que el factor es esencial.

Una vez que todos los factores técnicos tienen asignado el valor de la influencia, se procede al cálculo de los resultados de cada factor, es decir, se realiza una multiplicación entre la influencia del factor y su peso asociado, ver en la Tabla 3 la columna “Resultado”.

Cuando se han calculado los resultados de cada uno de los factores técnicos, se aplica la expresión descrita a continuación, donde el sumatorio se corresponde a la suma de los resultados de los factores técnicos.

$$TCF = 0,6 + (0,01 \times \sum_{r=1}^{r=13})$$

En la Tabla 3 se presenta un resumen del procedimiento del cálculo de los factores técnicos, siendo Ri los factores concretos.

| Factor | Descripción | Peso | Influencia | Resultado |
|---------------|--------------------------------------|-------------|-------------------|------------------|
| R1 | Sistema Distribuido | 2 | n1 | R1=2*n1 |
| R2 | Objetivos de rendimiento | 2 | n2 | R2=2*n2 |
| R3 | Eficiencia respecto al usuario final | 1 | n3 | R3=1*n3 |
| R4 | Procesamiento complejo | 1 | n4 | R4=1*n4 |
| R5 | Código reutilizable | 1 | n5 | R5=1*n5 |
| R6 | Instalación sencilla | 0,5 | n6 | R6=0,5*n6 |
| R7 | Fácil utilización | 0,5 | n7 | R7=0,5*n7 |
| R8 | Portabilidad | 2 | n8 | R8=2*n8 |
| R9 | Fácil de cambiar | 1 | n9 | R9=1*n9 |
| R10 | Uso concurrente | 1 | n10 | R10=1*n10 |
| R11 | Características de seguridad | 1 | n11 | R11=1*n11 |
| R12 | Accesible por terceros | 1 | n12 | R12=1*n12 |
| R13 | Se requiere formación especial | 1 | n13 | R13=1*n13 |

Tabla 3. Cálculo de los factores técnicos (TCF)

6. Cálculo de los factores de entorno (EF)

Además de tener en cuenta los factores técnicos para el ajuste de los UUCP (Puntos Caso de Uso no ajustados), en segundo lugar se deben contabilizar los factores de entorno. Para ello, a cada factor de entorno definido en la Tabla 4 (Ri) se le asigna un valor entre 0 y 5 dependiendo de su influencia en el proyecto. Asignar un valor 0 significa que el factor es irrelevante para el proyecto, un valor 3 es

promedio y un valor 5 significa que el factor es esencial.

Una vez que todos los factores de entorno tienen asignado el valor de la influencia, se procede al cálculo de los resultados de cada factor, es decir, se realiza una multiplicación entre la influencia del factor y su peso asociado, ver en la Tabla 4 la columna “Resultado”.

Cuando se han calculado los resultados de cada uno de los factores, se aplica la expresión descrita a continuación, donde el sumatorio se corresponde a la suma de los resultados de los factores de entorno.

$$EF = 1,4 + (-0,03 \times \sum_{r1}^{r8})$$

En la Tabla 4 se presenta un resumen del procedimiento del cálculo de los factores de entorno, siendo Ri los factores concretos.

| Factor | Descripción | Peso | Influencia | Resultado |
|--------|--------------------------------------|------|------------|-----------|
| R1 | Familiar con RUP | 1,5 | n1 | R1=1,5*n1 |
| R2 | Experiencia en la aplicación | 0,5 | n2 | R2=0,5*n2 |
| R3 | Experiencia en orientación a objetos | 1 | n3 | R3=1*n3 |
| R4 | Capacidades de análisis | 0,5 | n4 | R4=0,5*n4 |
| R5 | Motivación | 1 | n5 | R5=1*n5 |
| R6 | Requisitos estables | 2 | n6 | R6=2*n6 |
| R7 | Trabajadores a tiempo parcial | -1 | n7 | R7=-1*n7 |
| R8 | Lenguaje complejo | -1 | n8 | R8=-1*n8 |

Tabla 4. Clasificación de los factores de entorno (FE)

7. Cálculo de los puntos de caso de uso ajustados (UCP)

Finalmente, para obtener los Puntos Caso de Uso ajustados (UCP) se utilizan los datos obtenidos en los pasos anteriores, Puntos Caso de Uso no ajustados (UUCP) y factores de ajuste (TCF y EF), haciendo uso de la expresión que se presentan a continuación.

$$UCP = UUCP * TCF * EF$$

Se debe tener en cuenta que a través del cálculo de esta expresión obtenemos una estimación del tamaño y no del esfuerzo.

8. Estimación del esfuerzo

Como ocurre en otros métodos de estimación, una vez obtenido el tamaño, se puede obtener el esfuerzo. Para ello, se utiliza la siguiente expresión:

$$\text{Esfuerzo} = UCP * \text{Factor de Productividad}$$

El método originario propone usar un factor de ajuste (Factor de Productividad) similar al que se usa en el método de Puntos Función clásico, si bien Karner propone concretamente 20 personas – hora por cada Punto Caso de Uso (UCP). Otras propuestas son las de Barnerjee que propone un rango entre 15 y 30 horas, o la de Scheider y Winters, que sugiere un refinamiento de los factores de entorno (EF), en concreto, proponen seguir el procedimiento que se presenta a continuación:

- Contar los factores de entorno entre R1 y R6 cuya influencia es inferior a 3 (influencia promedio) y los

factores de entorno entre R7 y R8 que son superiores a 3.
Ver “factores de entorno” en la Tabla 4.

• Entonces:

- 20 horas-hombre por UCP si el valor es ≤ 2
- 28 horas-hombre por UCP si el valor es ≤ 4
- 36 horas-hombre por UCP si el valor es ≥ 5 , en este caso se debería replantear el proyecto.

Se debe destacar, que el valor del esfuerzo estimado, calculado mediante la expresión presentada anteriormente, no cubre todas las fases del ciclo de vida del proyecto, sino que se refiere únicamente a las horas-hombre invertidas en el desarrollo de la funcionalidad especificada en los casos de uso (fase de codificación). La fase de codificación representa generalmente un 40% del esfuerzo total del proyecto. En este sentido, para obtener el esfuerzo total del proyecto, se puede realizar un nuevo ajuste que consiste en sumar a la estimación de esfuerzo obtenida por UCP, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo del software, que se pueden distribuir de la siguiente forma: análisis 10%, diseño 20%, codificación 40%, pruebas 15% y sobrecarga 15%.

1.13 Conceptos de bases de datos

Definición de bases de datos

Un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información determinados.

Sistema de gestión de bases de datos

Los **sistemas de gestión de bases de datos** o **SGBD** (en inglés *database management system*, abreviado *DBMS*) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Existen distintos objetivos que deben cumplir los SGBD.

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios *niveles de abstracción*.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia.** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad determinada que tiene determinadas condiciones, por ejemplo que los menores de edad no pueden tener licencia de conducir. El sistema no debería aceptar datos de un conductor menor de edad. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Manejo de transacciones.** Una Transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.

- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Fases del Diseño de Bases de Datos

El diseño de bases de datos, representa un enfoque orientado a los datos para el desarrollo de sistemas de información (fase de diseño): la atención se centra en los datos y sus propiedades. Estas fases son:

Diseño Conceptual

Parte de los requerimientos y su resultado es el esquema conceptual de la base de datos. Un **esquema conceptual** es una descripción de alto nivel de la estructura de la base de datos, independiente del software del DBMS (Data Base Management System) que se use para manipularla.

Los diagramas de datos más ampliamente usados para del diseño conceptual de base de datos son los diagramas entidad – relación (ER), UML (Unified Modeling Language).

Diseño Lógico

Parte del esquema conceptual y da como resultado un esquema lógico. Un **esquema lógico** es una descripción de la estructura de una base de datos que puede procesar el software de DBMS.

Diseño Físico

Parte del esquema lógico y da como resultado un esquema físico. Un **esquema físico** es una descripción de la implantación de la base de datos en la memoria secundaria; describe las estructuras de almacenamiento y los métodos usados para tener un acceso efectivo a los datos, por lo anterior el esquema físico se adapta al DBMS específico.

Modelos de datos

Es una colección de conceptos que se emplean para describir la estructura de una base de datos. Esa colección de conceptos incluye entidades, atributos y relaciones.

La mayoría de los modelos de datos poseen un conjunto de operaciones básicas para especificar consultas y actualizaciones de la base de datos. Los modelos de datos más representativos son los siguientes:

Modelos lógicos basados en objetos:

- **Modelo entidad – relación:** Es el **modelo de datos** mediante el cual se pretende visualizar los objetos que pertenecen a la Base de Datos como **entidades** (que corresponde al concepto de objeto de la Programación Orientada a Objetos) las cuales tienen unos atributos y se vinculan mediante *relaciones*.
- **Modelo Orientado a objetos:** Al igual que el modelo E-R, se basa en una colección de objetos. Un objeto contiene valores almacenados en variables de instancia dentro de un objeto. Un objeto contiene partes de código que operan sobre el objeto, estas partes se llaman métodos. Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan en clases

Modelos lógicos basados en registros:

- **Modelo de red:** Los datos se representan mediante colecciones de registros y las relaciones se representan mediante enlaces o ligas, los cuales pueden verse como punteros. Los registros en la base de datos se organizan como colecciones de grafos arbitrarios
- **Modelo Jerárquico:** Es similar al modelo de red en el sentido de que los datos y las

relaciones entre los datos se representan mediante registros y enlaces, respectivamente. Se diferencia del modelo de red en que los registros están organizados como colecciones de árboles en vez de grafos arbitrarios.

- **Modelo Relacional:** El modelo relacional representa los datos y las relaciones entre los datos mediante una colección de tablas, cada una de las cuales tiene un número de columnas con nombres únicos. La relación se indica mediante campos en los mismos archivos.

Diagrama Entidad-Relación (E-R)

El diagrama entidad-relación se basa en "una percepción del mundo real que consiste en una colección de objetos básicos llamados entidades y relaciones entre estos objetos".

El diagrama entidad-relación es el modelo más ampliamente usado para el diseño conceptual de bases de datos y fue introducido por Peter Chen en 1976.

Los elementos básicos son entidades, interrelaciones y atributos.

Entidades. Una entidad es un objeto que se distingue de otros objetos por medio de un conjunto específico de atributos. Las entidades representan clases de objetos de la realidad. PERSONA, MUNICIPIO, PAÍS. Las entidades se representan gráficamente por rectángulos.

Relaciones. Es una asociación entre varias entidades. Se representan por rombos.

Atributos. Representan las propiedades básicas de las entidades. Se representan por elipses. El Diagrama E-R, muestra además propiedades de opcionalidad y cardinalidad.

Opcionalidad u obligatoriedad. Una entidad puede tener o no relaciones de pertenencia u ocurrencia con relación a otra entidad.

Cardinalidad. Indica el grado de relación de entre las entidades, que puede ser: Uno a Uno, Uno a muchos, o muchos a muchos.

Conceptos

Normalización

Conjunto de reglas que sirven para ayudar a los diseñadores a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede.

La normalización también se puede definir como el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener.

Grados de normalización.

Existen varios niveles de normalización, sin embargo son tres los niveles básicos: Primera Forma Normal (1FN), Segunda Forma Normal (2FN) y Tercera Forma Normal (3FN). Cada una de estas formas tiene sus propias reglas.

- Primera Forma Normal (1FN): Incluye la eliminación de todos los grupos repetidos.
- Segunda Forma Normal (2FN): Asegura que todas las columnas que no son **llave** sean completamente dependientes de la llave primaria (PK).
- Tercera Forma Normal (3FN): Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

Integridad Referencial

La integridad referencial es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad.

Redundancia

La redundancia hace referencia al almacenamiento de los mismos datos varias veces en diferentes lugares. La redundancia de datos puede provocar problemas como:

- Incremento del trabajo: como un mismo dato está almacenado en dos o más lugares, esto hace que cuando se graben o actualicen los datos, deban hacerse en todos los lugares a la vez.
- Desperdicio de espacio de almacenamiento: ya que los mismos datos están almacenados en varios lugares distintos, ocupando así más bytes del medio de almacenamiento. Este problema es más evidente en grandes bases de datos.
- Inconsistencia de datos: esto sucede cuando los datos redundantes no son iguales entre sí. Esto puede suceder, por ejemplo, cuando se actualiza el dato en un lugar, pero el dato duplicado en otro lugar no es actualizado.

Capítulo 2

Descripción del Sistema de Interés

Etapa de Concepción

Objetivo: Definir la necesidad o problema, limitar el alcance del sistema, listar los riesgos y determinar si el producto de software propuesto es económicamente viable.

Durante esta etapa, también se da a conocer el marco teórico del flujo de negocio al equipo de trabajo (especialmente al arquitecto de sistemas) para hacer un análisis previo y esbozar una posible solución con el fin de, comprender el concepto y poder definir el alcance del sistema y poder con esto, hacer una propuesta económica inicial al cliente.

Con estas premisas, el cliente y el equipo de desarrollo, contarán con las bases para determinar si el proyecto es viable o no.

2.1 Definición del Problema

¿Por qué la necesidad de un sistema de control de disposiciones legales?

Se ha encontrado que en empresas con una aparente administración óptima, su área de control de cumplimiento legal (en muchos casos: áreas jurídicas) tiene deficiencias que les han acarreado desde simples sanciones de carácter administrativo, hasta contingencias de consecuencias considerables derivadas del incumplimiento. Esto en su mayoría es consecuencia de que la información que se debe manejar para el total cumplimiento de las disposiciones legales que le aplican a la empresa, son una enorme cantidad de tareas, fechas límite y documentos que se deben de cumplir en tiempo y forma por lo que llevar un control individual y desorganizado propicia el olvido o retraso del cumplimiento, y esto en muchas ocasiones se traduce para una empresa, en sanciones administrativas o incluso en cierres parciales o totales de las mismas.

La necesidad de mejorar la administración de los trámites que se deben de cumplir en tiempo y forma en las empresas es evidente. El reto es unir en un solo sistema, todas las disposiciones legales que le aplican a una empresa (dependiendo de sus necesidades) y proporcionar medios visuales y tecnológicos para tener una mejor administración y apoyar a la toma de decisiones con información exacta y oportuna.

La definición de nuestro problema es sin duda, el alto número de disposiciones legales que hay que administrar para cumplir en tiempo y forma, las cuales sin ayuda tecnológica, con el tiempo se transforman en un trabajo imposible de realizar, por lo cual, como objetivo principal tenemos la necesidad de construir un sistema que administre las disposiciones legales necesarias y aplicables a una empresa u organización en particular, y nos indique día a día la lista de tareas sin cumplir.

En el Grupo de Gas LP (en adelante GGLP), se han presentado multas por la falta del cumplimiento de algunas disposiciones y cierres parciales de sus plantas, inclusive situaciones de fallas técnicas (como explosiones de Autotanques¹) debido a la mala administración de los insumos.

2.2 Objetivos principales

- A. Identificar el riesgo diario en que se encuentra cada una de las plantas del grupo de Gas LP, de ser clausurada o multada por alguna auditoría legal.
- B. Reducir el número de multas generadas por el incumplimiento de las disposiciones legales en una Auditoría (en cada una de las plantas del Corporativo Gas LP).
- C. Administrar el conjunto de disposiciones legales que le aplican al Cliente.
- D. Identificar una disposición legal, el responsable de resolverla, y monitorear el estado de la misma.

2.3 Alcance del Sistema

El alcance del sistema se limitará con los 8 puntos siguientes, los cuales son validados y firmados por el

¹ Explosión de Autotanque en Lindavista por una manguera en mal estado.
<http://www.eluniversal.com.mx/notas/451363.html>

cliente.

Necesidades básicas requeridas

- Administrar un banco de información ordenado del marco legal aplicable a la organización, donde se incluyan: leyes federales, leyes estatales, leyes municipales, normas, reglamentos, certificaciones, entre otras.
- Gestionar los activos fijos de la organización y llevar el control del marco legal aplicable a cada uno de ellos.
- Establecer responsables y fechas compromiso a partir de las obligaciones que se deben cumplir y darles seguimiento hasta que lleguen a su término.
- Una vez realizada la configuración inicial, el sistema debe ser capaz de generar automáticamente las siguientes obligaciones de acuerdo a la periodicidad establecida.
- Los responsables de las obligaciones, reportarán avances, delegarán tareas, y podrán obtener reportes de sus actividades por un rango de fechas.
- Llevar el control de los documentos generados por el cumplimiento de las obligaciones, de forma física y digital.
- Medir el nivel de riesgo de la organización en función de las obligaciones que se tienen que cumplir en una determinada fecha.
- Monitoreo de múltiples organizaciones.

2.4 Factibilidad del sistema

Factibilidad Económica

El GGLP enfrenta una larga lista de incumplimientos, por lo que de éstos se desprende que:

1. Sea presa fácil de las multas e incluso de clausuras de las plantas.
2. Tenga pérdida económica ocasionada por multas.
3. Tenga pérdida económica ocasionada por cierre de actividades.
4. Tenga pérdida económica ocasionada por el tiempo en que los trabajadores vuelven a tramitar las evidencias que se traspapelan o se pierden.
5. Tenga pérdida económica ocasionada por tramitar una evidencia con carácter de urgente.
6. Tenga pérdida económica ocasionada por asignar tiempo al área jurídica para localizar al responsable de alguna evidencia, en muchos casos sin dar con el responsable porque no exista o porque no se cuenta con los datos.
7. Tenga pérdida económica ocasionada por pagos de tiempo extra al área jurídica en tiempos de auditorías.
8. El área jurídica labore largas jornadas.

Debido a lo anterior, el costo por el desarrollo del Sistema es superado por el beneficio que se obtiene de él.

Para el GGLP esta situación se sale de control, ya que solo existe un área jurídica para 11 plantas, 2

hoteles y 2 almacenes repartidos a lo largo de la República Mexicana, por lo que ha decidido invertir en un sistema de cómputo a través del cual pueda dar seguimiento a cada una de las obligaciones en tiempo y forma, con una total responsabilidad.

Factibilidad Técnica

El Grupo de Gas LP cuenta con un área de Sistemas, la cual tiene los recursos necesarios para la correcta implementación del Sistema vía WEB. El área de Sistemas se encarga de configurar y mantener una INTRANET entre todas las empresas del grupo, por lo tanto es factible la instalación, configuración y mantenimiento del sistema.

2.5 Flujo de Cumplimiento Regulatorio Actual

Proceso AS IS:

En la etapa de levantamiento de requerimientos o elaboración, se requiere conocer el ámbito de nuestro cliente, conocer el proceso que usa habitualmente para realizar la tarea que queremos administrar y automatizar.

El Sistema de Cumplimiento Regulatorio se encargará de administrar en su totalidad disposiciones legales (de cualquier tipo) que tengan que cumplir cada una de las empresas del grupo y posteriormente se encargará de administrar las auditorías y/o certificaciones.

Para cumplir con las obligaciones que le aplican a cualquiera de sus plantas, almacenes u hoteles, y estar en regla con las autoridades jurídicas, el corporativo divide esta tarea en dos procesos:

El primero es el proceso de Monitoreo de disposiciones legales, el cual consiste en averiguar y estar al pendiente de cualquier normatividad NUEVA o MODIFICACIÓN a una existente, que aplique directamente a las plantas, almacenes, hoteles, autotransportes de reparto y/o automóviles particulares con los que cuenta el GGLP, al identificar la normatividad, ésta debe ser revisada y enviada al departamento correspondiente de la empresa, para tramitar o llevar a cabo las acciones necesarias y cumplir con la disposición, evitando así alguna sanción.

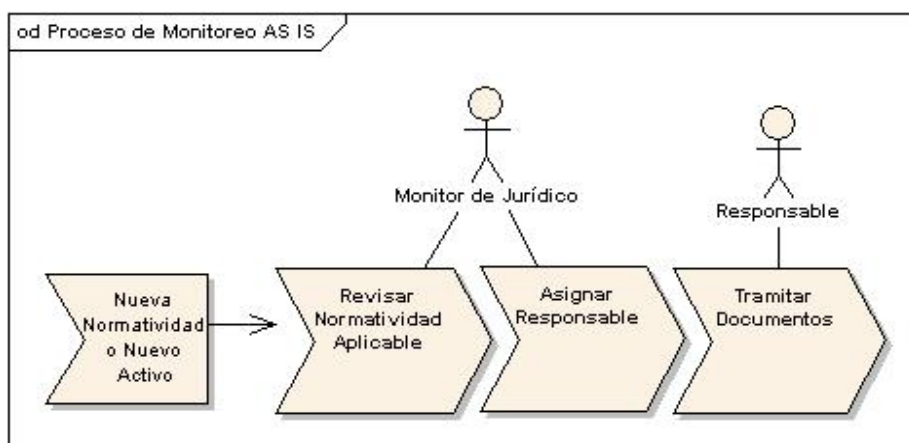


Fig. 2 - 1 Proceso de Monitoreo Actual

Durante este proceso también existe la posibilidad de adquirir un nuevo Activo para cualquiera de las empresas del grupo, como ejemplo tenemos la compra de un nuevo Autotank, y con esto el área

jurídica tendría que llevar un nuevo registro de las obligaciones que adquiere con este activo fijo.

El problema radica cuando el área jurídica delega la obligación al área correspondiente, la cual puede incluso estar en otra de las empresas del Grupo, y la mayoría de las veces la obligación llega a tener un responsable, sin embargo, el proceso solo llega al punto de resolver la disposición, pero la evidencia de haber cumplido rara vez está documentada o se traspapela y no llega al lugar correcto en el momento oportuno de una auditoría. Por otra parte, los directivos del Grupo no cuentan con el porcentaje de riesgo en que se encuentra alguna de las plantas de ser sancionadas, y con esto se tienen pérdidas monetarias muy grandes.

El proceso actual de monitoreo de disposiciones legales, tiene desventajas debido a que el área jurídica no es capaz de llevar el control de todos los activos de la empresa y mucho menos de las disposiciones que debe cumplir cada uno de ellos. Este proceso se realiza durante todo el año, y muchas veces se contrata a una empresa externa para que le proporcione al Grupo la lista de las normatividades que le aplican, dependiendo de un estudio de los activos fijos de su empresa y de sus procesos de flujo de negocio y giro.

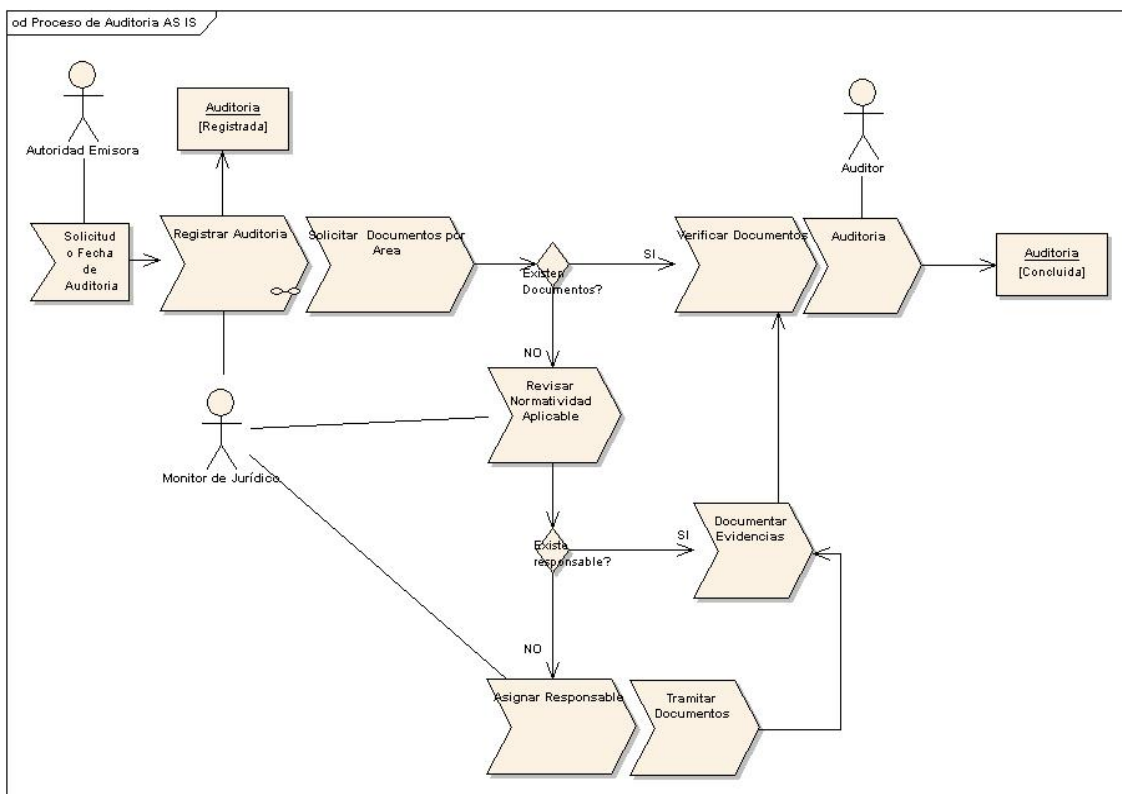


Fig. 2 - 2 Proceso de Auditoría Actual

En segundo lugar tenemos el proceso de Auditoría actual, el cual se realiza de la siguiente manera:

Cada una de las Autoridades² que emiten una normatividad realiza el proceso de diferente manera pero el objetivo es el mismo. Una secretaría puede comunicarse con el área jurídica de la empresa para notificar que se realizará una visita de revisión presencial y anexa una serie de documentos que se requieren para la auditoría, entre ellos se declara a las personas que pueden llegar a auditar, con

² Cualquier Autoridad emisora de normas que competen a la empresa en cuestión. Por ejemplo: Secretaria de Economía (SE), Secretaría del Trabajo y Previsión Social (STPS)

fotografía y los posibles días en que se llevará a cabo, se adjunta la lista de documentos que se requieren para ser revisados y junto con la visita se realiza una inspección de instalaciones y procesos.

Durante la inspección se agrega al informe los detalles de la revisión, los puntos que se han resuelto en su totalidad o cuales fueron cumplidos parcialmente. Cuando existe una disposición legal que la ley marca como necesaria para el funcionamiento de la planta y esta no está completamente aceptada por el auditor, se puede hasta cerrar la planta en ese momento. Lo cual ya ha sucedido en el pasado.

El área jurídica muchas veces no sabe quién es la persona encargada de resolver ciertas cuestiones específicas o existen cosas que se olvidan o se traspapelan, siendo blancos perfectos para las auditorías sin previo aviso.

En el caso de las certificaciones como la industria limpia, la Procuraduría Federal de Protección al Ambiente (PROFEPA) encargada de dicha certificación, provee una guía de auto-evaluación ambiental con el fin de conocer el estado de la empresa en material ambiental³, posteriormente un auditor lista las tareas necesarias para obtener la certificación, el área jurídica de la empresa se encarga de repartir a cada área los puntos a cumplir. Cuando se acerca el día de la revisión, el área jurídica o en este caso el área de calidad se encarga de hacer una pre-revisión en cada área. Cada una de éstas cuenta con un auditor y un auditado, cuando se encuentran con puntos sin cumplir, se levantan no conformidades, que se resumen en la descripción de lo que no está cumplido o falta por cumplir, se les da un plazo para cumplirlo y se hace firmar al responsable del área para que este lo tenga listo en la siguiente revisión definitiva.

El área jurídica genera una vista gerencial para dar a conocer el porcentaje de cumplimiento a los directivos de la empresa, actualmente se basan en un documento donde tienen cargados todos los puntuales a cumplir y por área van escribiendo el porcentaje que ellos creen que cumplen. Al final de la revisión generan gráficas con los datos obtenidos y ese documento es lo que se muestra en la reunión gerencial los días lunes.

El proceso que llevan en el GGLP no es eficiente ya que ni la dirección del Grupo, ni las gerencias de cada una de las empresas en él, sabe qué nivel de cumplimiento tienen en un determinado día. Sin estos datos es imposible tomar decisiones para evitar sanciones o incluso para resolver contingencias urgentes.

En resumen, el área jurídica, puede recibir en cualquier momento una persona encargada de una revisión, por lo que es imprescindible tener en orden los documentos y las constancias de cumplimiento.

2.6 Análisis de Riesgos

El Sistema de Cumplimiento Regulatorio es un proyecto que conlleva la interacción de diversas áreas en cada empresa del Grupo de Gas LP, por lo que se presentan diversos riesgos listados a continuación.

Por parte del cliente:

- a. Disponibilidad de tiempo (persona o personas encargadas de recopilar la información necesaria, seleccionarla y transmitirla al equipo de trabajo).
- b. Disponibilidad de tiempo para presentación de avances y pruebas.
- c. Disponibilidad de información suficiente por parte del cliente.
- d. Acertar en la selección del personal adecuado para la definición de cada módulo.

³ Página principal de PROFEPA para obtener información acerca de la certificación de la industria limpia. <http://www.profepa.gob.mx/PROFEPA/AuditoriaAmbiental/ProgramaNacionaldeAuditoriaAmbiental/>

Por parte del equipo de desarrollo:

- a. Curva de aprendizaje del equipo de trabajo en la tecnología sugerida⁴.

2.7 Alternativa de Solución

Proceso To Be:

El proceso de Monitoreo de disposiciones legales que se propone es el siguiente:

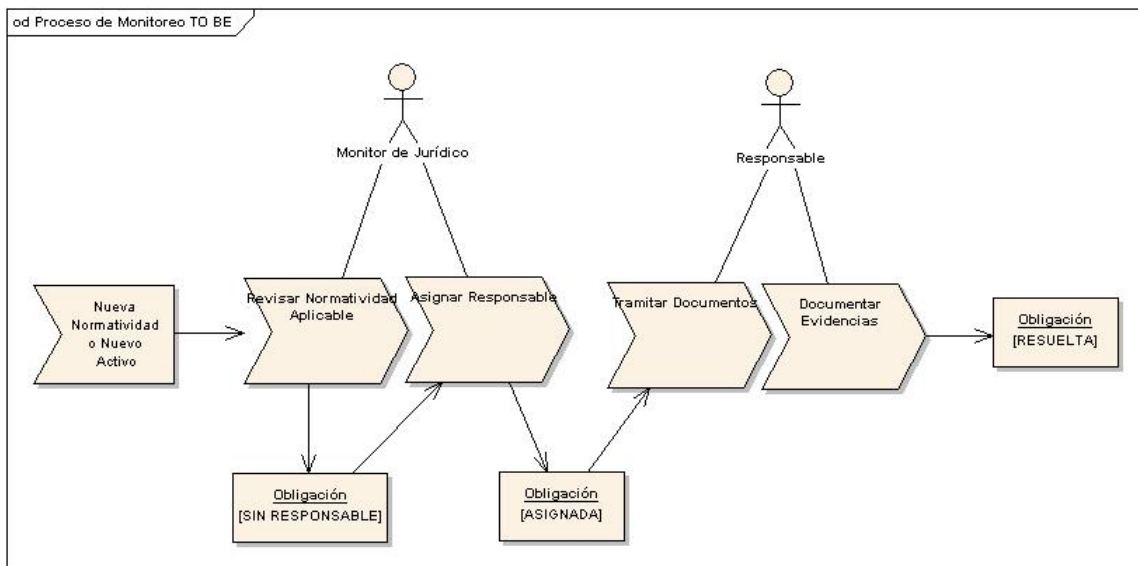


Fig. 2 - 3 Proceso de monitoreo de disposiciones legales propuesto

Con este flujo se asegurará que la información generada por los responsables quedará documentada y a la vista de todos los involucrados.

Por otro lado, el proceso de Auditoría tiene un cambio mayor (Fig. 2-4), debido a que podemos modificarlo para ganar tiempo, disminuir el esfuerzo y aumentar los beneficios económicos, al contar con un sistema que centralice la información y nos permita dar seguimiento a las obligaciones como tareas comunes.

⁴ Definida en el capítulo 3 como requerimiento del cliente.

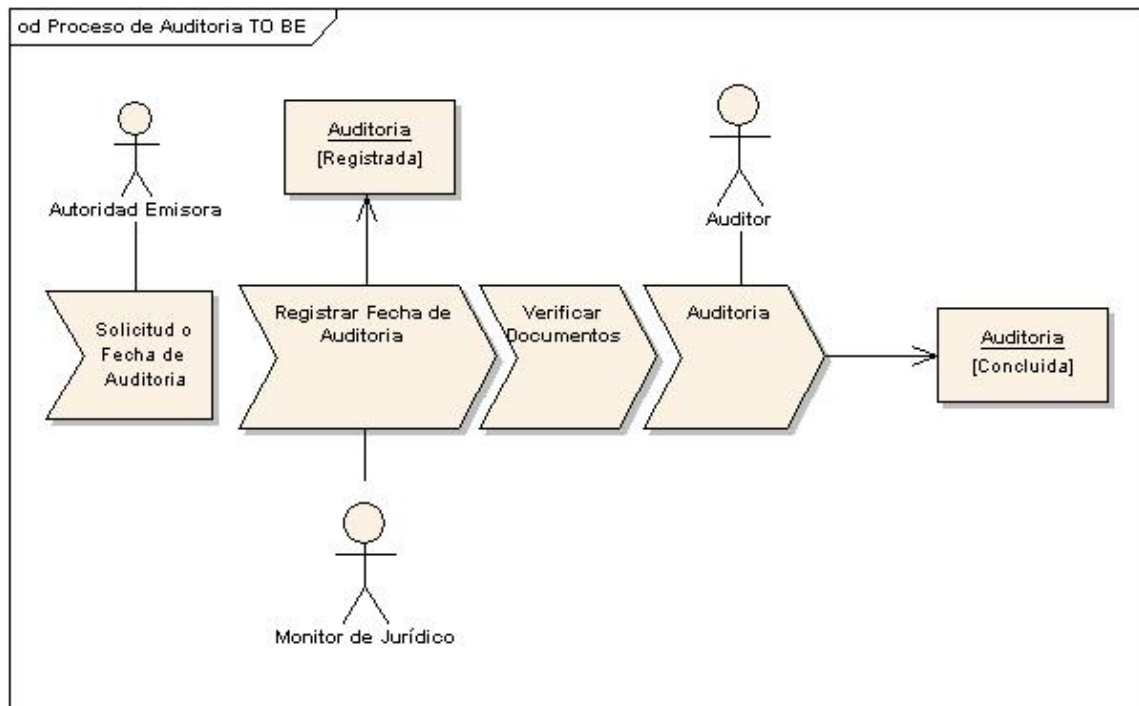


Fig. 2 - 4 Proceso de Auditoria Nuevo

2.8 Metodología de desarrollo propuesta.

Etapa 1: Para versión 1.0

Configuración del Sistema

1. Carga del marco regulatorio aplicable al cliente.
2. Asociación de la normatividad aplicable a cada tipo de activo.
3. Carga de los activos fijos de la empresa que se desean administrar.
4. Asignar responsables.

Monitoreo

1. Seguimiento de tareas.
2. Levantamiento de no conformidades.

Nuevas entradas

1. Nueva normatividad.
2. Nuevo tipo de activo.
3. Nuevo activo fijo en alguna de las empresas del grupo.

Configuración del Sistema

El sistema contará con un catálogo llamado Marco Regulatorio el cual será cargado con las normatividades aplicables al cliente. Cada normatividad será administrada y actualizada cada vez que las Autoridades Responsables así lo señalen (por lo general cada año).

Una vez que tenemos toda la información que la empresa debe cumplir, se requiere asociar cada uno de los puntos con el tipo de activo que lo debe cumplir. Es decir, los activos de la empresa que sean de tipo Autotanque, deben cumplir con el punto XXX de la Norma MMMMMM, independientemente de cuál sea

la placa del Autotanque todos deben cumplir con ese punto de la norma, a este proceso le llamamos Asociación de la Normatividad con el Tipo Activo.

De esta manera, cuando se realiza la carga de los activos fijos de la empresa, cada uno ya tiene automáticamente asignados los puntuales de la normatividad que debe cumplir y solo requerirán el encargado de hacerlo.

Este proceso se realizará al inicio del sistema, y en el futuro sólo se actualizarán los cambios en las normatividades.

Monitoreo periódico

En este punto, cada responsable puede entrar con su propia sesión al sistema y ver las tareas que tiene asignadas, la fecha compromiso, el nivel de riesgo que representa la tarea, y la descripción del punto que debe cumplir.

El monitor de la normatividad es capaz de dar seguimiento al avance de los responsables e incluso dar por resuelta una tarea o levantar no conformidades⁵ en la misma.

Nuevas Entradas

Cuando se cuenta con una nueva Normatividad (decretada por alguna Autoridad), el monitor será el encargado de darla de alta en el catálogo de Normatividades y asociarla al tipo de activo correspondiente. El sistema se encargará de generar las tareas automáticamente para cada uno de los activos fijos que están dados de alta en el sistema.

Cuando se cuenta con un tipo de activo nuevo, el monitor será el encargado de dar de alta el tipo de activo y asociar las normatividades que le corresponden.

Cuando se cuenta con un activo fijo nuevo en la empresa, el monitor será el encargado de dar de alta el activo fijo y asignar un responsable de cumplir con estas tareas.

⁵ Una no conformidad sucede cuando un monitor considera que el punto de la normatividad no está cumplido al 100% y asigna una tarea más al responsable.

Capítulo 3

Elaboración

Etapa de Elaboración

Esta etapa consiste en afinar los requerimientos iniciales, definir la arquitectura, vigilar los riesgos, y producir un plan de administración de proyecto de software.

Como parte primordial del proceso unificado se define el número de iteraciones que se realizan durante el ciclo de vida del proyecto, estas iteraciones se reflejan en el plan de proyecto. Para el fin de esta tesis, abarco la primera iteración del sistema, la cual consiste en tener una versión inicial del sistema (versión 1.0) funcional y basada en los puntos principales para iniciar con el uso del sistema.

3.1 *Análisis de las necesidades del Sistema*

Entrevistas con el cliente

El proceso de descubrir las necesidades del sistema se llevará a cabo mediante entrevistas con el cliente, éstas minutas generadas se analizan y se documentan en casos de uso mediante escenarios, una vez que están completos se llevan con cliente para que sean aprobados y firmados.

Puntos a considerarse para la realización de las entrevistas con el cliente:

- Organigrama y directorio de contactos involucrados por parte del cliente.
- Agendar reuniones con la gente indicada (usuario que conoce el proceso que se requiere gestionar) y guardar la confirmación de las mismas.
- Seleccionar o generar cuestionarios específicos de acuerdo al proyecto con anterioridad a las entrevistas.
- Verificar que los requerimientos del cliente sean congruentes con los objetivos y el alcance del proyecto.
- Minuta de la reunión, firmada por el cliente.
- En caso de que utilicen un formato específico para cualquier documento que se desea generar por el sistema, se debe pedir una copia de cada uno de ellos.

Para realizar el levantamiento de requerimientos por medio de entrevistas, utilizo el sistema Acuérdate WEB¹ especializado en la gestión de reuniones.

Como ejemplo de los documentos generados por el proceso de entrevistas con el cliente, presento un fragmento de la reunión de definición de requerimientos técnicos (Figura 3 - 1).

¹ Acuérdate es una aplicación Web con una metodología dirigida a dar apoyo en planeación de reuniones, convocatoria a reuniones, establecimiento de acuerdos, creación de minutas, seguimiento de actividades y control estadístico de productividad.

Informe de acuerdos - Narrativa

Página 1

| |
|--|
| Título: Reunión para la obtención de requerimientos no funcionales. |
| Lugar: Por Definir |
| Objetivo: Obtener requerimientos no funcionales. |

| Tipo Reunión: | Fecha | Hora Inicial | Hora Final | Duración | Carácter |
|---|------------|--------------|------------|----------|-----------|
| Reunión para la obtención de requerimientos no funcionales. | 16/01/2006 | 09:00 AM | 10:00 AM | 1.0 hrs. | ORDINARIA |

| |
|---|
| Convocante: Sin Empresa: null luna null dhi+. |
| Asistentes: Corporativo de Gas LP: Ing. Morales Ramírez Iván+. |

Avances**Puntos:****1.- Requerimientos no funcionales.**

Entrevista para obtener los requerimientos no funcionales para la elaboración del Sistema de Cumplimiento Regulatorio Cumple.

Los requerimientos recaudados en esta reunión se utilizarán para analizar la forma en que se deberá construir la aplicación, los estándares que se deben seguir para su desarrollo y el lenguaje de desarrollo requerido.

Persona que se entrevista: Ing. Iván Morales.
 Área y Puesto: Área de Sistemas, Gerente del Área.
 Persona que entrevista: Gabriela López.

1.1.- Requerimientos Técnicos.

- 1.- Definir el lenguaje de programación que se debe usar para el desarrollo del Sistema.
Responsable(s): Morales Ramírez Iván 05/03/2009 No Resuelto
 Avance(s): 17/01/2006 El sistema se desarrollará con el lenguaje de programación JAVA versión 1.4
- 2.- Definir los estándares que se deben seguir a lo largo del proyecto.
Responsable(s): Morales Ramírez Iván 17/01/2006 No Resuelto
 Avance(s): 17/01/2006 Los estándares del lenguaje de programación.
- 3.- Si es Aplicación Web: Definir ¿Cual Browser se va a usar?
Responsable(s): Morales Ramírez Iván 17/01/2006 No Resuelto
 Avance(s): 17/01/2006 Internet Explorer 6.0

Fig. 3 - 1 Minuta de reunión para obtención de requerimientos técnicos

3.2 Requerimientos funcionales

“Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer” (Sommerville 2005).

Mediante el análisis de los datos obtenidos en las entrevistas con el cliente, se definen los siguientes módulos:

- a. Administración del Sistema.
- b. Configuración de Marco Regulatorio.
- c. Seguimiento de Tareas.
- d. Administración de Riesgos.
- e. Control Documental.

Cada módulo está representado por un diagrama de casos de uso que identifica a los requerimientos funcionales. Para este trabajo de tesis explico de manera general el objetivo de uno y a detalle los escenarios de módulo de configuración del marco regulatorio.

a. Administración del Sistema.

El módulo de "Administración del Sistema" (Fig. 3 - 2), tiene por objetivo mantener actualizados los datos de los usuarios, la lista de perfiles, las funcionalidades activas de los módulos, y los catálogos generales que se utilizan en el sistema.

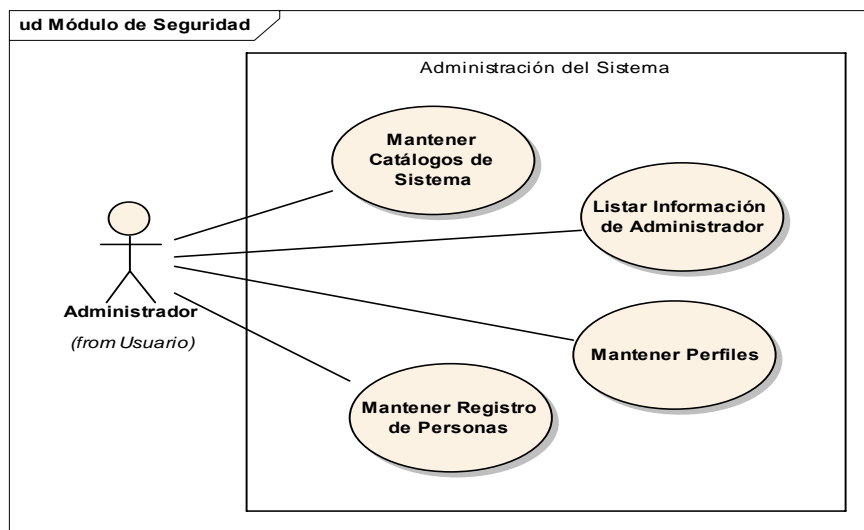


Fig. 3 - 2 Diagrama de casos de uso: Administración del Sistema

La Fig. 3 -3 muestra los catálogos del sistema que son gestionados por el administrador.

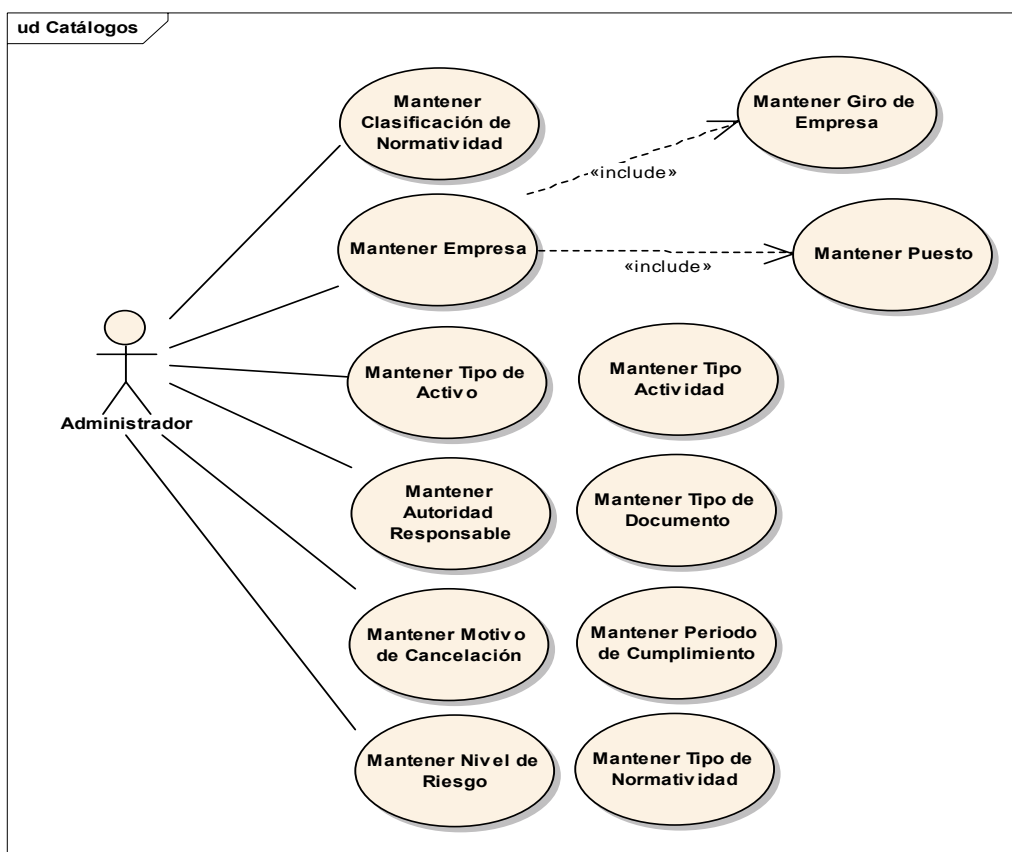


Fig. 3 - 3 Diagrama de casos de uso: Catálogos del Sistema

b. Configuración de Marco Regulatorio.

El Módulo de Configuración de Marco Regulatorio (Fig.3 - 4), tiene por objetivo guardar el listado de las disposiciones legales² que se aplican a la empresa y/o grupo de empresas que administrará el sistema, por ejemplo, los puntos para conseguir una certificación, se guardan de una manera fácil y ordenada en el catálogo y se pueden dividir en puntos informativos u obligatorios. Los puntos informativos no necesitan tener una evidencia física o digital para asegurar que están resueltos, en cambio, los puntos obligatorios serán tratados de otra manera, y muy importante será ingresar los datos de las evidencias que se requieren tener o conseguir para dar por cumplido el puntual.

También cuenta con un catálogo de todos los activos fijos de cada empresa, que tienen que cumplir con alguna disposición legal. Estos activos fijos se deben agrupar en tipos de activos (siempre y cuando compartan las mismas obligaciones).

² El sistema puede administrar cualquier listado que tenga puntos a cumplir para obtener un objetivo específico, por ejemplo: Normas, Certificaciones, Tareas.

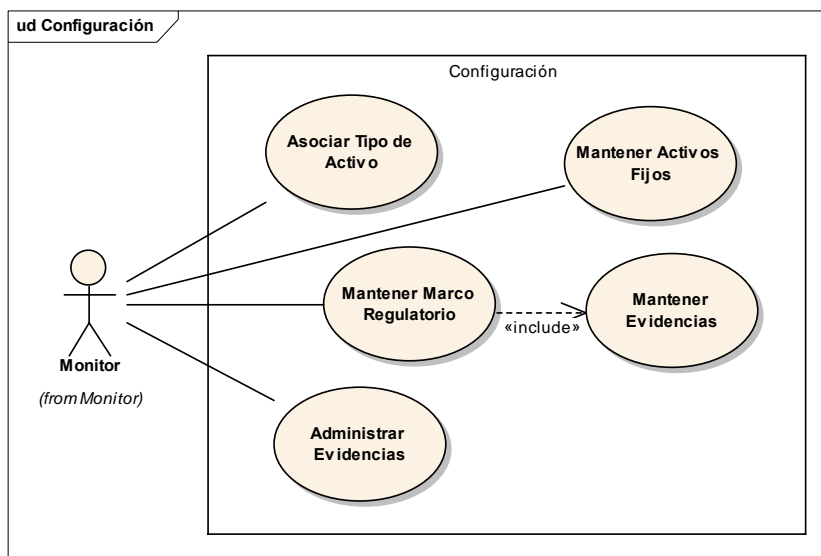


Fig. 3 - 4 Diagrama de casos de uso: Configuración de Marco Regulatorio

Una vez capturados los tipos de activos y las disposiciones legales de carácter obligatorio, se asocian entre sí, para que el sistema genere una tarea por cada evidencia con la que cuente la disposición legal. El siguiente paso es asignar un responsable a cada tarea.

En las siguientes tablas desgloso los casos de uso para el Módulo de Configuración de Marco Regulatorio.

| | |
|--------------------------|---|
| Caso de Uso: | <i>Mantener Marco Regulatorio</i> |
| Módulo: | d. Configuración |
| Objetivo: | Mantener actualizado el marco legal aplicable a la empresa. El caso de uso cuenta con las funcionalidades de agregar, modificar, y eliminar normatividades. |
| Actores: | Monitor. |
| Restricciones: | <i>Pre-condición.</i> El usuario ha iniciado sesión en el sistema. <i>Pre-condición.</i> El usuario cuenta con los permisos suficientes para realizar la opción. <i>Pre-condición.</i> El usuario cuenta con la información necesaria para capturar los datos requeridos. <i>Post-condición.</i> El sistema tiene el marco regulatorio actualizado. |
| <i>Escenarios</i> | |
| {Flujo Básico} | <u>Mantener Marco Regulatorio</u> |
| | <ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario solicita al sistema mantener una nueva normatividad. 2. El sistema despliega los tipos de normatividad existentes en el sistema. 3. El usuario elige el tipo de normatividad que desea mantener. 4. El sistema despliega: <ul style="list-style-type: none"> • Las normatividades existentes para ese tipo de normatividad. • Las opciones de agregar, modificar, y eliminar una normatividad. |

| | |
|---------------------|--|
| | <ol style="list-style-type: none"> 5. El usuario elige la opción de agregar una normatividad. (Ver Flujo Alterno) 6. El sistema despliega los datos de: Punto, Descripción, Clasificación, Nivel de Riesgo, Frecuencia de Cumplimiento, Desarrollo del Punto, para ser capturados. 7. El usuario captura los datos y solicita guardar los datos. 8. El sistema guarda lo datos y recarga la información mostrada. (Ver Caso de Uso Mantener Evidencias) 9. El caso de uso termina. |
| {Flujo Alterno} | <u>5. Modificar Normatividad</u> |
| | <ol style="list-style-type: none"> a. El usuario elige la opción de modificar una normatividad. b. El sistema despliega los datos existentes para punto, descripción, clasificación, nivel de riesgo, frecuencia de cumplimiento, desarrollo del punto, y estatus, para que el usuario los modifique. c. El usuario modifica los datos y solicita guardar la información. d. El sistema verifica si el usuario ha cambiado la clasificación de normatividad: <ul style="list-style-type: none"> • Caso 1: De obligación a informativa: Para cada activo fijo asociado al puntual, se eliminará la asociación cuando todos los activos fijos asociados a él, no cuenten con responsables en cada uno de los acuerdos. (En caso contrario no se permitirá cambiar de obligación a informativa). • Caso 2. De informativa a obligación: Para cada activo fijo asociado al puntual obligatorio, se creará un acuerdo por cada evidencia del puntual. e. El flujo continúa en el paso 8 del flujo básico. |
| {Flujo Alterno}. | <u>5. Eliminar Normatividad</u> |
| | <ol style="list-style-type: none"> a. El usuario selecciona una normatividad y elige la opción de eliminar una normatividad. (Ver Flujo Excepcional) b. El sistema elimina la normatividad. c. El flujo continúa en el paso 9 del flujo básico. |
| {Flujo Excepcional} | <u>5a. Selección Inválida.</u> |
| | <ol style="list-style-type: none"> i. El usuario elige la opción de eliminar sin previamente seleccionar la normatividad que desea eliminar. ii. El sistema muestra el mensaje "Por favor seleccione una de las opciones". iii. El caso de uso termina. |
| {Flujo Excepcional} | <u>5b. El registro no se puede eliminar</u> |
| | <ol style="list-style-type: none"> i. El sistema muestra un mensaje de: "El registro no se puede eliminar porque existen dependencias". ii. El caso de uso termina. |

| | |
|----------------|--|
| Caso de Uso: | <i>Administrar Evidencias</i> |
| Módulo: | d. Configuración |
| Objetivo: | Este caso de uso, asigna responsables a los acuerdos derivados de las evidencias que debe cumplir un activo fijo. |
| Actores: | Monitor. |
| Restricciones: | <i>Pre-condición.</i> El usuario ha elegido un activo fijo para administrar sus evidencias. <i>Pre-condición.</i> El usuario ha iniciado sesión en el sistema. <i>Pre-condición.</i> El usuario cuenta con los permisos suficientes para realizar la acción. |

| Escenarios | |
|---------------------|--|
| {Flujo Básico} | <u>Administrar evidencias</u> |
| | <ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario solicita al sistema administrar evidencias de un activo fijo. 2. El sistema lista las evidencias existentes para el activo fijo y las opciones de: <ul style="list-style-type: none"> ▪ Uno a uno. ▪ Asignar un responsable a DIFERENTES evidencias. ▪ Asignar un responsable a todos los activos fijos (mismo tipo, misma evidencia, misma empresa). <p>El sistema muestra (como aviso) el número de las disposiciones legales que no cuentan con acuerdos.</p> 3. El usuario elige la opción uno a uno. (Ver Flujo Alterno) 4. El usuario elige un acuerdo. 5. El sistema despliega la lista de las personas disponibles en el sistema, un calendario para seleccionar la fecha compromiso, y pide al usuario una instrucción opcional para el responsable seleccionado. 6. El usuario captura los datos de los usuarios que quedarán asignados para cumplir con ese acuerdo, la fecha compromiso y una instrucción opcional y solicita al sistema guardar los datos. 7. El sistema guarda los datos y refresca la información. 8. El caso de uso termina. |
| {Flujo Alterno} | <u>3. Asignar un Responsables a DIFERENTES evidencias</u> |
| | <ol style="list-style-type: none"> a. El usuario elige la opción de Asignar un responsable a DIFERENTES evidencias. b. El usuario elige uno o varios acuerdos y solicita al sistema enviar los datos. c. El sistema despliega la lista de las personas disponibles en el sistema, un calendario para seleccionar la fecha compromiso, y pide al usuario una instrucción opcional para el responsable seleccionado. d. El usuario captura los datos de la persona que quedará asignada al o los acuerdos previos seleccionados, la fecha compromiso y una instrucción opcional y solicita al sistema guardar los datos. e. El flujo continúa en el punto 7 del flujo básico. |
| {Flujo Alterno}. | <u>3. Asignar un responsable a todos los activos fijos del mismo tipo.</u> |
| | <ol style="list-style-type: none"> a. El usuario elige la opción de Asignar un responsable a todos los activos fijos (mismo tipo, misma evidencia, misma empresa). b. El usuario elige un acuerdo y solicita al sistema enviar los datos. (Ver Flujo Excepcional) c. El sistema despliega la lista de las personas disponibles en el sistema, un calendario para seleccionar la fecha compromiso, y pide al usuario una instrucción opcional para el responsable seleccionado. d. El usuario captura los datos de la persona que quedará asignada al acuerdo previo seleccionado, la fecha compromiso y una instrucción opcional y solicita al sistema guardar los datos. e. El flujo continúa en el punto 7 del flujo básico. |
| {Flujo Excepcional} | <u>3b. Solicitud inválida.</u> |
| | <ol style="list-style-type: none"> i. El usuario solicita enviar los datos sin haber seleccionado un acuerdo. ii. El sistema envía un mensaje de: "Seleccione una de las opciones" iii. El caso de uso termina. |

| | |
|--------------|---|
| Caso de Uso: | <i>Mantener Activos Fijos</i> |
| Módulo: | d. Configuración |
| Objetivo: | Este caso de uso contiene la información de los Activos Fijos de una empresa. Los activos |

| | |
|----------------------|--|
| | fijos identifican específicamente un activo de la empresa. <i>Por ejemplo: Computadora DELL2600 No. Serie "GLOPEZ123", Edificio A, Vehículo de transporte interno CORSA placas: MAM123, etc.</i> |
| Actores: | Monitor. |
| Restricciones: | <p><i>Pre-condición.</i> El usuario ha iniciado sesión en el sistema.</p> <p><i>Pre-condición.</i> El usuario cuenta con los permisos suficientes para realizar la opción.</p> <p><i>Pre-condición.</i> El usuario cuenta con la información necesaria para capturar los datos requeridos.</p> <p><i>Post-condición.</i> El sistema tiene el catálogo de activos fijos actualizado.</p> |
| Escenarios | |
| {Flujo Básico} | <u>Mantener activo fijo</u> |
| | <ol style="list-style-type: none"> 1. El usuario solicita dar de alta un nuevo activo fijo. 2. El sistema despliega la información de la descripción del activo fijo: 3. El usuario solicita guardar la información capturada. |
| {Flujo Alternativo} | <u>5. Modificar Normatividad</u> |
| | <ol style="list-style-type: none"> f. El usuario elige la opción de modificar una normatividad. g. El sistema despliega los datos existentes para punto, descripción, clasificación, nivel de riesgo, frecuencia de cumplimiento, desarrollo del punto, y estatus, para que el usuario los modifique. h. El usuario modifica los datos y solicita guardar la información. i. El sistema verifica si el usuario ha cambiado la clasificación de normatividad: <ul style="list-style-type: none"> • Caso 1: De obligación a informativa: Para cada activo fijo asociado al puntual, se eliminará la asociación cuando todos los activos fijos asociados a él, no cuenten con responsables en cada uno de los acuerdos. (En caso contrario no se permitirá cambiar de obligación a informativa). • Caso 2: De informativa a obligación: Para cada activo fijo asociado al puntual obligatorio, se creará un acuerdo por cada evidencia del puntual. j. El flujo continúa en el paso 8 del flujo básico. |
| {Flujo Alternativo}. | <u>5. Eliminar Normatividad</u> |
| | <ol style="list-style-type: none"> d. El usuario selecciona una normatividad y elige la opción de eliminar una normatividad. (Ver Flujo Excepcional) e. El sistema elimina la normatividad. f. El flujo continúa en el paso 9 del flujo básico. |
| {Flujo Excepcional} | <u>5a. Selección Inválida.</u> |
| | <ol style="list-style-type: none"> iv. El usuario elige la opción de eliminar sin previamente seleccionar la normatividad que desea eliminar. v. El sistema muestra el mensaje "Por favor seleccione una de las opciones". vi. El caso de uso termina. |
| {Flujo Excepcional} | <u>5b. El registro no se puede eliminar</u> |
| | <ol style="list-style-type: none"> iii. El sistema muestra un mensaje de: "El registro no se puede eliminar porque existen dependencias". iv. El caso de uso termina. |

c. Seguimiento de Tareas.

El módulo de seguimiento de tareas (Fig. 3 - 5), tiene por objetivo que un usuario en este caso llamado "Solicitante" pueda revisar los avances de las tareas que ha asignado a una persona, calificarlas o modificar las fechas compromiso. En el mismo módulo un usuario "Responsable" puede escribir los avances de una tarea y/o delegarla a otra persona.

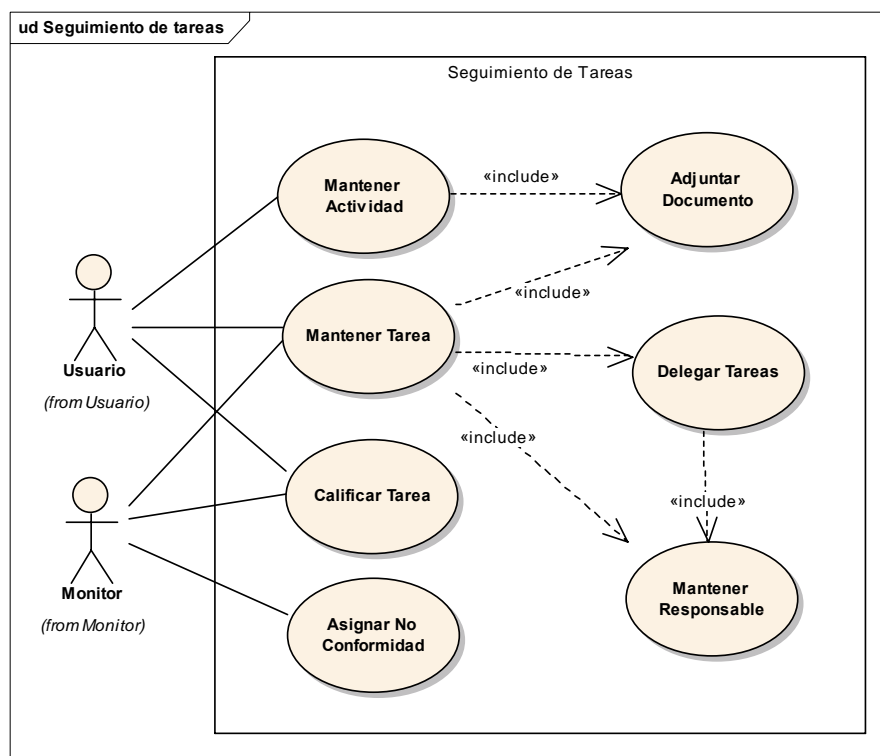


Fig. 3 - 5 Diagrama de caso de uso: Seguimiento de Tareas

Este módulo contendrá también un apartado especial para las no conformidades, que son traducidas como tareas generadas por el no cumplimiento de una tarea anterior y que deben ser atendidas a la mayor brevedad posible.

d. Administración de Riesgos.

El módulo de administración de Riesgos (Fig. 3 -6), tiene por objetivo tener un indicador gráfico (a manera de semáforo) que muestre las disposiciones legales que están sin resolver, y por lo tanto poner más atención en ellas. Se mostrarán totales de los puntuales que se han cumplido en color verde, que están en tiempo para ser resueltas en color ámbar o las tareas que están sin resolverse y fuera de tiempo en color rojo. Y por cada total se debe desplegar el detalle de cada una, hasta finalmente poder ver quién es el encargado de la tarea y los avances que se lleva de la misma.

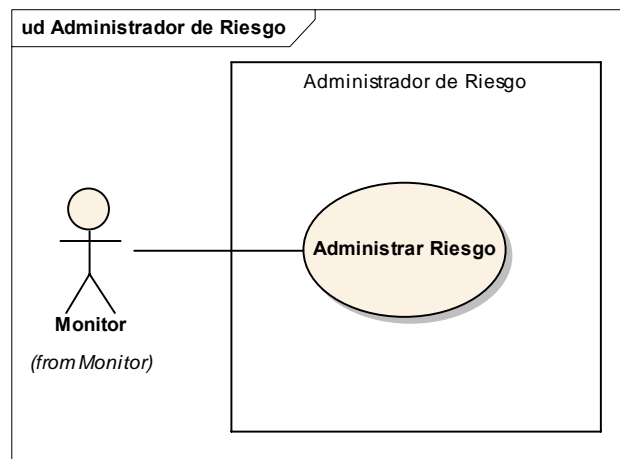


Fig. 3 - 6 Diagrama de caso de uso: Administrador de Riesgo

El Administrador de riesgo debe mostrar a todas las empresas pertenecientes al grupo corporativo, con la intención de que los directivos y monitores accedan a él y puedan tomar decisiones reales para el buen manejo de su empresa.

e. Mapa de Seguimiento.

El módulo de Mapa de Seguimiento tiene por objetivo listar los documentos necesarios para cumplir una disposición legal, con el fin de contar con la información concentrada y así mostrarla en caso de una auditoría. (Fig. 3 - 7)

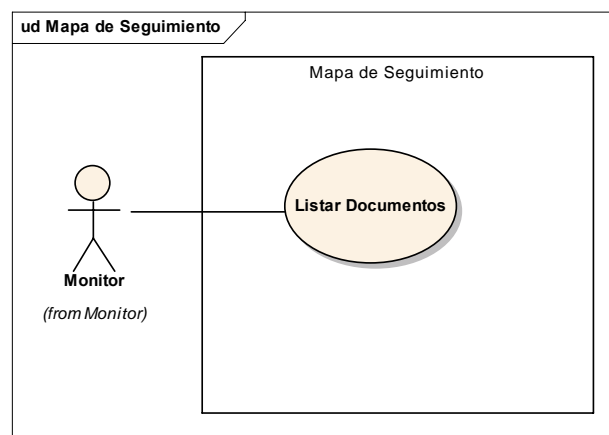


Fig. 3 - 7 Diagrama de caso de uso: Mapa de Seguimiento

3.3 Requerimientos no funcionales

“Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad.” (Sommerville 2005).

Debido a que los requerimientos no funcionales surgen de las necesidades del usuario, se toman en

cuenta las restricciones en el presupuesto, las políticas de la organización, estándares de desarrollo, la interoperabilidad con otros sistemas de software o hardware y de acuerdo a las entrevistas con el cliente, se definió como requerimiento técnico la tecnología que se empleará en el desarrollo del sistema, presentada a continuación:

| | |
|-----------------------------|---|
| Lenguaje de programación: | Java, Sun Microsystems (j2sdk versión 1.4) Java Server Pages (JSP) |
| Servidor de aplicaciones: | Apache Tomcat v 4.1.31 (puerto 8080) |
| Sistema Operativo: | Windows 2000 Server SP4, Windows 2003 |
| Manejador de base de datos: | Microsoft SQLServer 2000 Español SP3 (puerto 1433) |
| Navegador: | Microsoft Internet Explorer v.6 |
| Configuración de pantalla: | 1024 x 768 píxeles. |

Esta estructura de software se define porque el cliente cuenta con todas las características mencionadas, lo cual se entiende como un costo prácticamente nulo en la infraestructura de despliegue del sistema. En el capítulo 1, se describen las características de cada uno de los elementos que conforman esta arquitectura de software.

3.4 Estimación

Estimación por Casos de Uso

Estimación: Puntos de Casos de Uso

Proyecto

Sistema de Cumplimiento Regulatorio CUMPLE

20 Marzo 2006

1.- Factor de peso de actores sin ajustar (UAW)

| Actores | Factor peso | Cantidad | Factor * Cantidad | Descripción |
|------------|-------------|----------|-------------------|--|
| Simple | 1 | 0 | 0 | Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación |
| Medio | 2 | 0 | 0 | Otro Sistema que interactúa con el sistema a desarrollar |
| Complejo | 3 | 3 | 9 | Una persona que interactúa con el sistema |
| UAW | | | 9 | |

2.- Factor de peso de los casos de uso sin ajustar (UUCW)

| Tipo caso de uso | Factor | Cantidad | Factor * Cantidad | Descripción |
|------------------|--------|----------|-------------------|---------------------|
| Simple | 5 | 27 | 135 | <=3 4 - 7 > 7 |
| Medio | 10 | 2 | 20 | |
| Complejo | 15 | 1 | 15 | |
| UUCW | | | 170 | |

3.- Puntos de Caso de uso sin ajustar UUCP = UAW + UUCW

| | |
|-------------|------------|
| UUCP | 179 |
|-------------|------------|

4.- Factor de Complejidad Técnica TCF = $0.6 + 0.01 \times \sum(\text{Peso}_i \times \text{Valor asignado}_i)$

| Factor | Descripción | Peso | Valor Asignado | Peso * Valor asignado | Comentario |
|--------|---|------|----------------|-----------------------|--|
| T1 | Sistema distribuido | 2 | 0 | 0 | El sistema es centralizado |
| T2 | Objetivos de performance o tiempo de respuesta | 1 | 0 | 0 | Velocidad limitada por entradas provistas por el usuario |
| T3 | Eficiencia del usuario final | 1 | 0 | 0 | Escasas restricciones de eficiencia |
| T4 | Procesamiento interno complejo | 1 | 1 | 1 | No hay cálculos complejos |
| T5 | El código debe ser reutilizable | 1 | 0 | 0 | No se requiere que el código sea reutilizable |
| T6 | Facilidad de instalación | 0.5 | 0 | 0 | Escasos requerimientos de facilidad de instalación |
| T7 | Facilidad de uso | 0.5 | 0 | 0 | Normal |
| T8 | Portabilidad | 2 | 0 | 0 | No se requiere que el sistema sea portable |
| T9 | Facilidad de cambio | 1 | 0 | 0 | Se requiere un costo moderado de mantenimiento |
| T10 | Concurrencia | 1 | 0 | 0 | No hay concurrencia |
| T11 | Incluye objetivos especiales de seguridad | 1 | 0 | 0 | Seguridad normal |
| T12 | Provee acceso directo a terceras partes | 1 | 0 | 0 | Los usuarios de Web tienen acceso directo |
| T13 | Se requieren facilidades especiales de entrenamiento a usuarios | 1 | 0 | 0 | Pocos usuarios internos, sistema fácil de usar |

1

TCF = 0.61

5.- Factor de ambiente EF = $1.4 - 0.03 \times \sum(\text{Peso}_i \times \text{Valor asignado}_i)$

| Factor | Descripción | Peso | Valor Asignado | Peso * Valor asignado | Comentario | Factor conversión |
|--------|--|------|----------------|-----------------------|---|-------------------|
| E1 | Familiaridad con el modelo de proyecto utilizado | 1.5 | 3 | 4.5 | El grupo está bastante familiarizado con el modelo | 0 |
| E2 | Experiencia en la aplicación | 0.5 | 5 | 2.5 | La mayoría del grupo ha trabajado mucho tiempo en la aplicación | 1 |

| | | | | | | |
|----|---|-----|---|-----|---|---|
| E3 | Experiencia en OO | 1 | 5 | 5 | La mayoría del grupo programa en objetos | 0 |
| E4 | Capacidad del analista lider | 0.5 | 5 | 2.5 | Se contrato a un especialista | 1 |
| E5 | Motivación | 1 | 5 | 5 | El grupo está altamente motivado | 0 |
| E6 | Estabilidad de los requerimientos | 2 | 4 | 8 | No se esperan cambios durante las primeras entregas | 0 |
| E7 | Personal part-time | -1 | 0 | 0 | 1 persona de tiempo completo | 0 |
| E8 | Dificultad del lenguaje de programación | -1 | 1 | -1 | Se usará lenguaje JAVA | 0 |

26.5

| | | | |
|----|-------|----|----|
| EF | 0.605 | CF | 10 |
| | | CF | OK |

6.- Puntos de Caso de uso ajustados UCP = UUCP x TCF x EF

UCP= 66

7.- Esfuerzo en horas

E = Esfuerzo en horas hombre para desarrollo de funcionalidad, CF = factor de conversión

E = UCP x CF 660

8.- Duración del proyecto

| Etapa | Actividad | % | Horas Hombre |
|--------------|--------------------|-----|----------------|
| Concepción | Prospecto | 0 | 0.00 |
| Elaboración | Análisis | 15 | 220.00 |
| Elaboración | Diseño | 15 | 220.00 |
| Construcción | Programación | 45 | 660.00 |
| Transición | Pruebas | 10 | 146.67 |
| | Configuración | 0 | 0.00 |
| | Control de cambios | 0 | 0.00 |
| | PPQA | 0 | 0.00 |
| | MA | 0 | 0.00 |
| | Otras actividades | 15 | 220.00 |
| Total | | 100 | 1466.67 |

3.5 Plan de Proyecto

Con base a las actividades y la estimación de cada etapa, podemos comenzar el esbozo de la planeación del proyecto (Fig. 3 - 8), para dar una mejor visibilidad del avance y control del mismo. Durante todo el ciclo de vida del proyecto este plan se va afectando con los datos reales, y sufriendo ajustes.

| Id | Nombre de tarea | Duración | Comienzo | Fin |
|----|---|-----------|--------------|--------------|
| 1 | Sistema de Cumplimiento Regulatorio | 200 días? | lun 02/01/06 | vie 06/10/06 |
| 2 | Ciclo 1 (v.1.0) | 200 días? | lun 02/01/06 | vie 06/10/06 |
| 3 | Concepción | 13 días | lun 02/01/06 | mié 18/01/06 |
| 4 | Prospecto de Negocio | 1 día | lun 02/01/06 | lun 02/01/06 |
| 5 | Definición del Problema | 5 días | mar 03/01/06 | lun 09/01/06 |
| 6 | Evaluación de factibilidad | 1 día | mar 10/01/06 | mar 10/01/06 |
| 7 | Proceso AS IS (Flujo Actual) | 2 días | mié 11/01/06 | jue 12/01/06 |
| 8 | Proceso TO BE (Propuesta de Solución) | 1 día | vie 13/01/06 | vie 13/01/06 |
| 9 | Administración de Proyecto | 13 días | lun 02/01/06 | mié 18/01/06 |
| 10 | Elaboración | 77 días? | jue 19/01/06 | vie 05/05/06 |
| 11 | Análisis | 43 días | jue 19/01/06 | lun 20/03/06 |
| 12 | Planear entrevistas con cliente | 1.5 días | jue 19/01/06 | vie 20/01/06 |
| 13 | Realizar entrevistas | 5 días | vie 20/01/06 | vie 27/01/06 |
| 14 | Elaborar Casos de uso | 9.5 días | vie 27/01/06 | jue 09/02/06 |
| 34 | Definición de interfaz gráfica | 10 días | vie 10/02/06 | jue 23/02/06 |
| 35 | Validación de los Casos de Uso (Con cliente) | 10 días | vie 24/02/06 | jue 09/03/06 |
| 36 | Ajuste de los escenarios de los Casos de Uso | 2 días | vie 10/03/06 | lun 13/03/06 |
| 37 | Estimación de los Caso de uso | 5 días | mar 14/03/06 | lun 20/03/06 |
| 38 | Diseño | 31 días | mar 21/03/06 | mar 02/05/06 |
| 39 | Definición de la arquitectura del Sistema | 5 días | mar 21/03/06 | lun 27/03/06 |
| 40 | Generación de Modelo Conceptual | 5 días | mar 28/03/06 | lun 03/04/06 |
| 41 | Generación de diagrama de clases | 7 días | mar 04/04/06 | mié 12/04/06 |
| 42 | Generación de diagrama de secuencias | 7 días | jue 13/04/06 | vie 21/04/06 |
| 43 | Generación de Diseño de Base de datos | 3 días | lun 24/04/06 | mié 26/04/06 |
| 44 | Generación de Diccionario de Datos | 2 días | jue 27/04/06 | vie 28/04/06 |
| 45 | Ajustes a Diseño | 2 días | lun 01/05/06 | mar 02/05/06 |
| 46 | Administración de Proyecto | 77 días? | jue 19/01/06 | vie 05/05/06 |
| 47 | Construcción | 85 días? | lun 08/05/06 | vie 01/09/06 |
| 48 | Desarrollo de Casos de Uso | 66.5 días | lun 08/05/06 | mar 08/08/06 |
| 68 | Pruebas | 15 días | mar 08/08/06 | mar 29/08/06 |
| 69 | Administración del Proyecto | 85 días? | lun 08/05/06 | vie 01/09/06 |
| 70 | Transición | 25 días? | lun 04/09/06 | vie 06/10/06 |
| 71 | Preparar Discos de Instalación | 1 día | lun 04/09/06 | lun 04/09/06 |
| 72 | Instalación con el Usuario | 1 día | mar 05/09/06 | mar 05/09/06 |
| 73 | Capacitación al Usuario | 3 días | mié 06/09/06 | vie 08/09/06 |
| 74 | Pruebas de Aceptación del Usuario | 5 días | lun 11/09/06 | vie 15/09/06 |
| 75 | Optimización, corrección y control de errores | 3 días | lun 18/09/06 | mié 20/09/06 |
| 76 | Manual de Usuario | 5 días | jue 21/09/06 | mié 27/09/06 |
| 77 | Manual de instalación | 1 día | jue 28/09/06 | jue 28/09/06 |
| 78 | Liberación del Sistema (firmar carta de aceptación y entregar a | 1 día | vie 29/09/06 | vie 29/09/06 |
| 79 | Administración del Proyecto | 25 días? | lun 04/09/06 | vie 06/10/06 |

Fig. 3 - 8 Plan de Proyecto "Sistema de Cumplimiento Regulatorio"

3.6 Diseño del Sistema

Modelo Conceptual

Para la generación del modelo conceptual se parte del flujo básico de cada caso de uso y se analizan los sustantivos para generar las clases conceptuales y sus campos.

Caso de uso: Mantener Marco Regulatorio

Mantener Marco Regulatorio. {Flujo Básico}.

1. El caso de uso comienza cuando el **usuario** solicita al **sistema** mantener una nueva **normatividad**.
2. El **sistema** despliega los **tipos de normatividad** existentes en el **sistema**.
3. El **usuario** elige el **tipo de normatividad** que desea mantener.
4. El **sistema** despliega las **normatividades** existentes para ese **tipo de normatividad**, y las opciones de agregar, modificar, y eliminar una **normatividad**.
5. El **usuario** elige la opción de agregar una **normatividad**.
6. El **sistema** despliega los datos de:
Punto, Descripción, Clasificación, Nivel de Riesgo, Frecuencia de Cumplimiento, Desarrollo del Punto, para ser capturados.
7. El **usuario** captura los datos y solicita guardar los datos.
8. El **sistema** guarda lo datos y recarga la información desplegada.
9. El caso de uso termina.

| Sustantivos | Categoría |
|----------------------------|--------------------------------|
| Usuario | Actor |
| Sistema | Clase |
| Normatividad | Clase |
| Tipo de Normatividad | Clase |
| Punto | Atributo de Clase Normatividad |
| Descripción | Atributo de Clase Normatividad |
| Clasificación | Atributo de Clase Normatividad |
| Nivel de Riesgo | Atributo de Clase Normatividad |
| Frecuencia de Cumplimiento | Atributo de Clase Normatividad |
| Desarrollo del Punto | Atributo de Clase Normatividad |

La figura 3 - 9 muestra el diagrama conceptual para el caso de uso "Mantener Marco Regulatorio" del sistema Cumple.

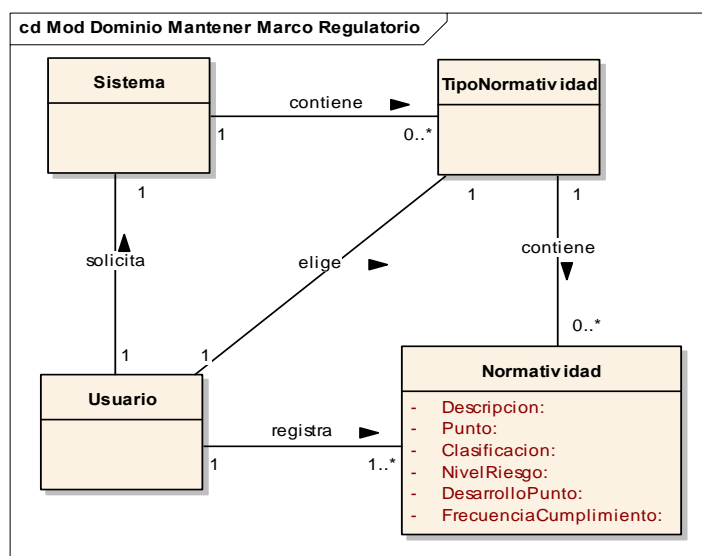


Fig. 3 - 9 Diagrama conceptual: Mantener Marco Regulatorio

Caso de uso: Administrar Evidencias

Administrar evidencias {Flujo Básico}.

1. El caso de uso comienza cuando el **usuario** solicita al **sistema** administrar **evidencias** de un **activo fijo**.
2. El **sistema** lista las **evidencias** existentes para el **activo fijo** y las opciones de:
 - Uno a uno.
 - Asignar un **responsable** a DIFERENTES **evidencias**.
 - Asignar un **responsable** a todos los **activos fijos** (mismo **tipo**, misma **evidencia**, misma **empresa**). El **sistema** muestra también el número de las **disposiciones legales** que no cuentan con el **acuerdo** de la **evidencia** que se debe administrar (solo como aviso).
3. El **usuario** elige la opción uno a uno.
4. El **usuario** elige un **acuerdo**.
5. El **sistema** despliega la lista de las **personas** disponibles en el **sistema**, un calendario para seleccionar la **fecha compromiso**, y pide al **usuario** una **instrucción** opcional para el **responsable** seleccionado.
6. El **usuario** captura los datos de él o los **usuarios** que quedarán asignados para cumplir con ese **acuerdo**, la **fecha compromiso** y una **instrucción** opcional y requiere guardar los datos al **sistema**.
7. El **sistema** guarda los datos y refresca la información.
8. El caso de uso termina.

| Sustantivos | Categorías |
|---------------------------|--------------------------------|
| Usuario | Actor |
| Sistema | Clase |
| Evidencia | Clase |
| Activo Fijo | Clase |
| Responsable (Persona) | Clase |
| Empresa | Clase |
| Disposición Legal (Punto) | Atributo de Clase Normatividad |
| Acuerdo | Clase |
| Fecha compromiso | Atributo de Clase Responsable |
| Instrucción | Atributo de Clase Responsable |

La figura 3 -10 muestra el diagrama conceptual para el caso de uso "Administrar Evidencias" del

sistema Cumple.

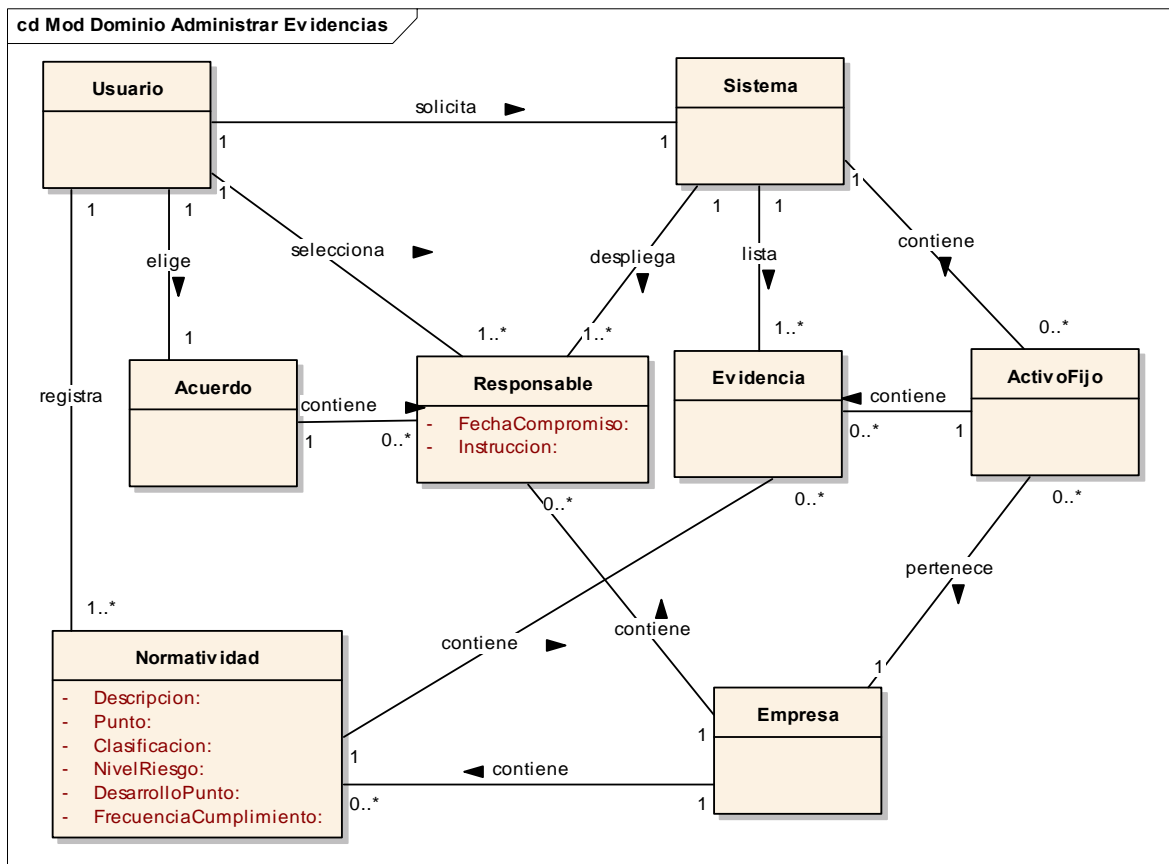


Fig. 3 - 10 Diagrama conceptual: Administrar Evidencias

Arquitectura del Sistema

La arquitectura del sistema estará basada en el MVC (modelo-vista-controlador)³.

Utilizaremos HTML (Lenguaje de Marcas de Hipertexto) y CSS (Hojas de estilo en cascada) para la presentación de los datos mediante un navegador web (capa de presentación o vista), estos datos serán procesados por una clase que controlará qué operaciones se deben realizar (controlador), y actuará sobre los datos representados por el modelo.

El modelo está compuesto por el paquete de clases Bean que contienen la estructura de los datos de cada objeto, el paquete de clases ManejaBean que gestionan la transformación de los datos de cada objeto y una librería (alx.jar) que controla el acceso a la base de datos y permite pasar datos entre tablas y objetos.

Las clases ManejaXXX (en el paquete ManejaBean) tienen implementadas las operaciones básicas de:

Insertar: inserta un registro de tipo XXX a la base de datos.

Actualizar: actualiza un registro de la tabla XXX en la base de datos.

³ Ver Capítulo 1.

Eliminar: elimina un registro de la tabla XXX en la base de datos.

Trae: obtiene un registro de la tabla XXX, donde se cumple la condición dada.

Busca: Obtiene una lista de registros de la tabla XXX, que cumplen con la condición dada.

Donde XXX, representa una tabla en la base de datos.

Por ejemplo:

| Tabla (Entidad) | Clases a generar: | |
|-----------------|-------------------|-------------------------|
| | Paquete Bean | Paquete ManejaBean |
| Persona | Persona.java | ManejaPersona.java |
| Estado | Estado.java | ManejaEstado.java |
| Normatividad | Normatividad.java | ManejaNormatividad.java |

El diagrama de la arquitectura se muestra en la Fig. 3 - 11.

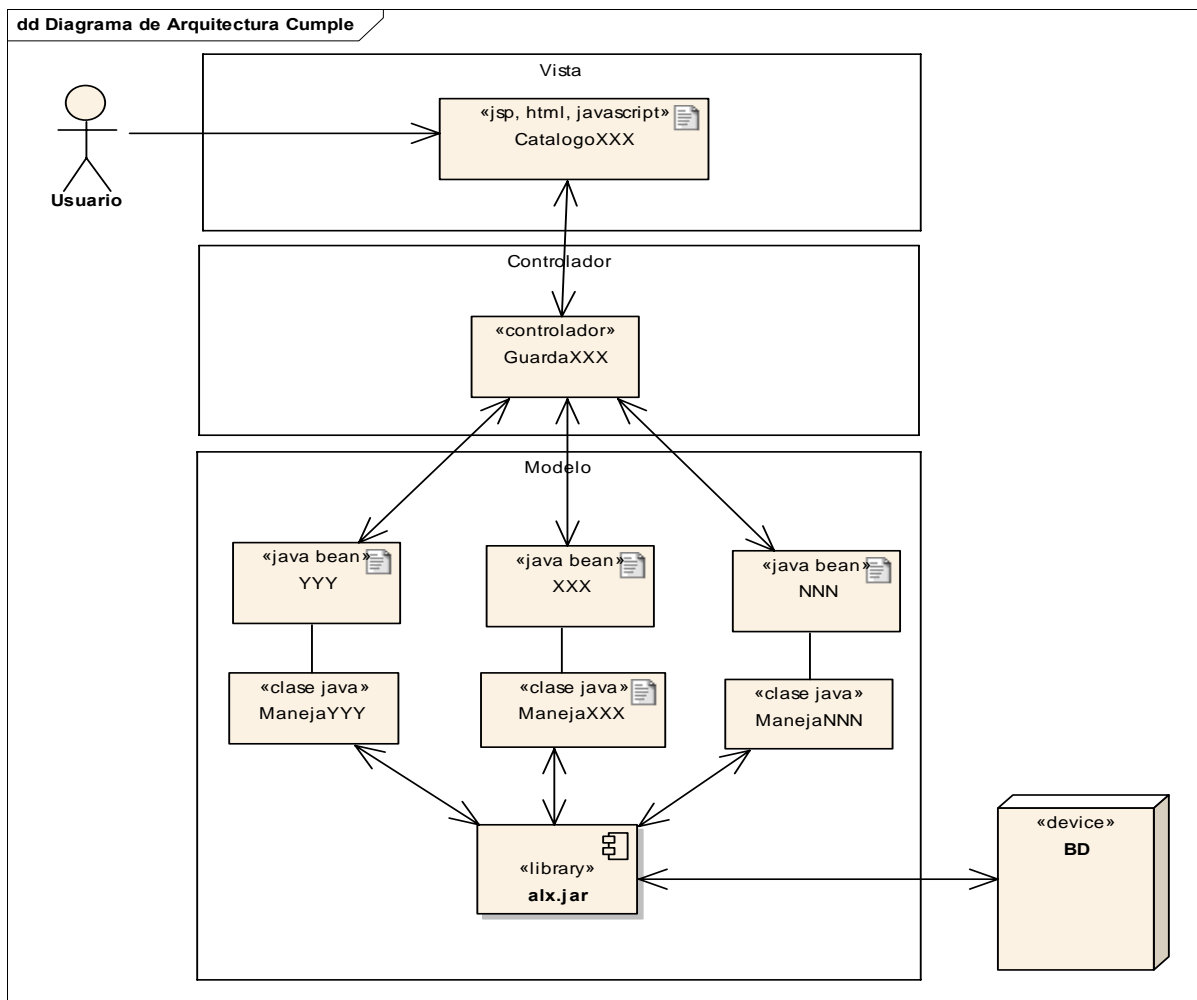


Fig. 3 - 11 Diagrama de Arquitectura del Sistema

Diagrama de clases

Las figuras 3 -12, y 3 -13 muestran el diagrama de clases para el caso de uso “Mantener Marco Regulatorio” del sistema Cumple.



Fig. 3 - 12 Diagrama de clases: Maneja Marco Regulatorio (paquete Bean)

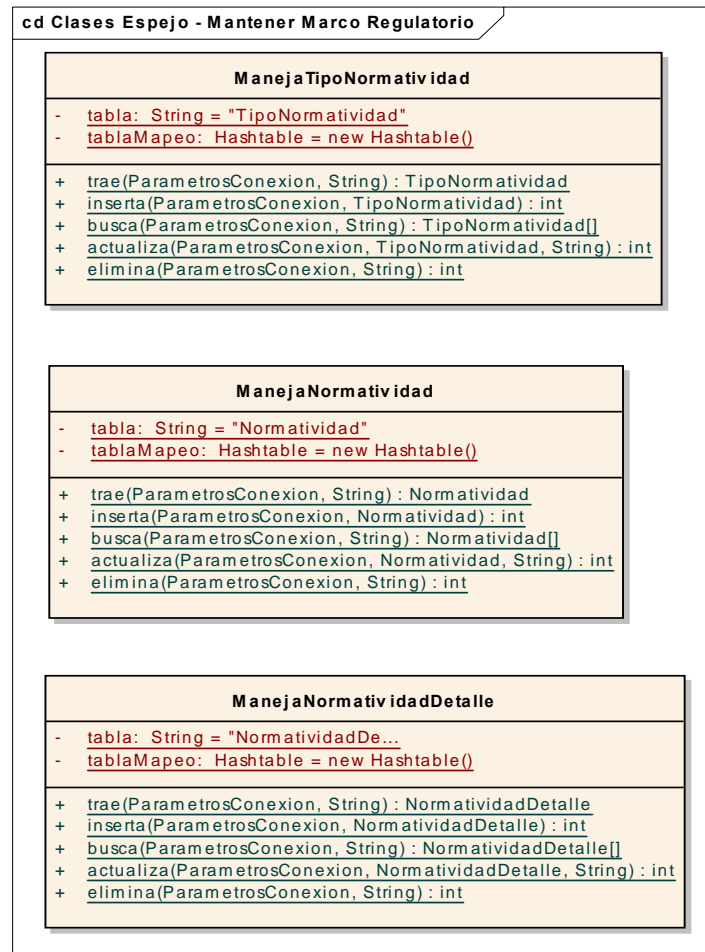


Fig. 3 - 13 Diagrama de clases: Mantener Marco Regulatorio (paquete ManejaBean)

Diagramas de Secuencia

Diagrama de Secuencia de Administrar Evidencias

La figura 3 -14 muestra el diagrama de secuencia para el caso de uso "Mantener Evidencias" del sistema Cumple.

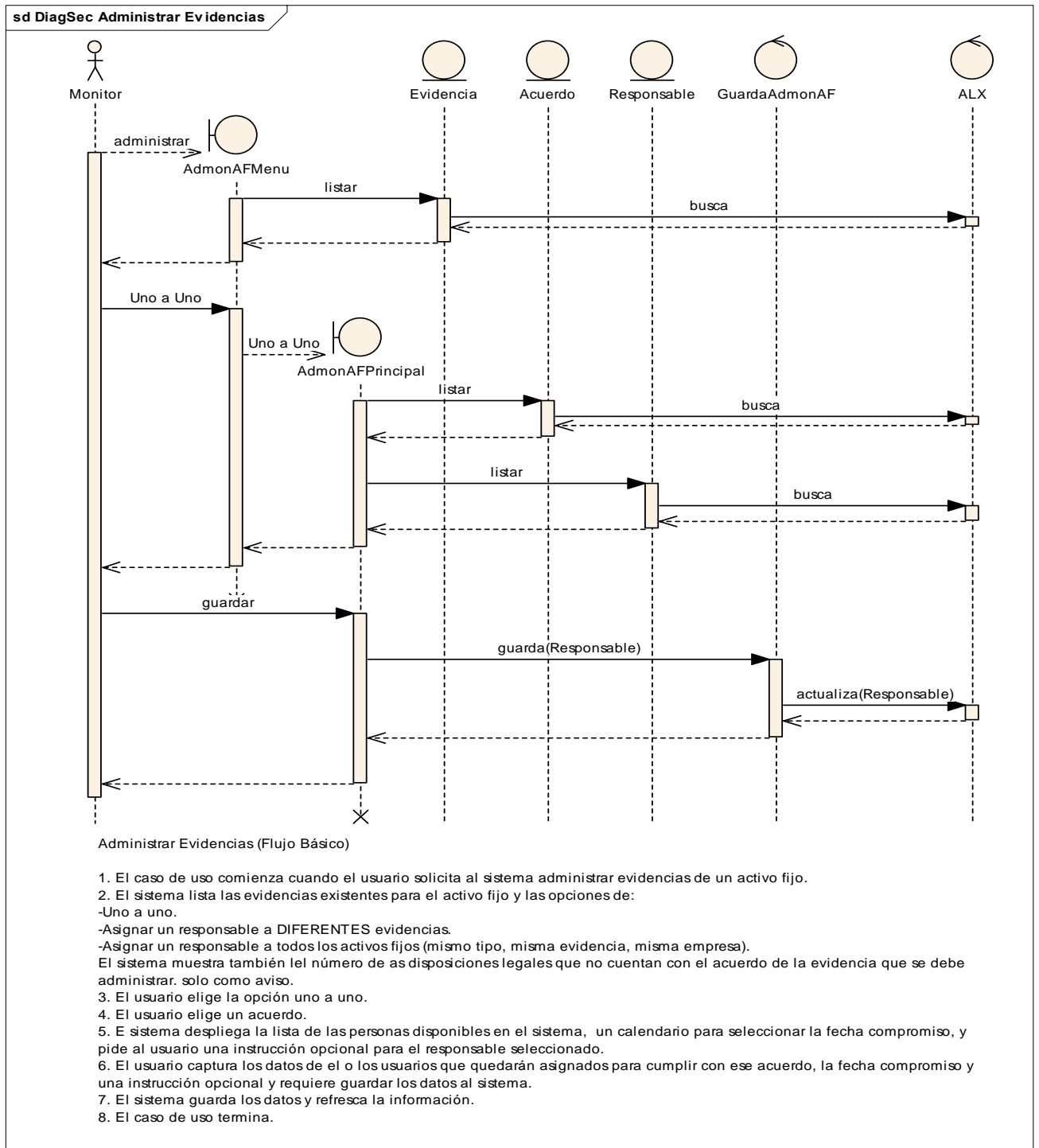


Fig. 3 - 14 Diagrama de secuencias: Administrar Evidencias

Diagrama de Secuencia de Mantener Marco Regulatorio

La figura 3 -15 muestra el diagrama de secuencia para el caso de uso "Mantener Marco Regulatorio" del sistema Cumple.

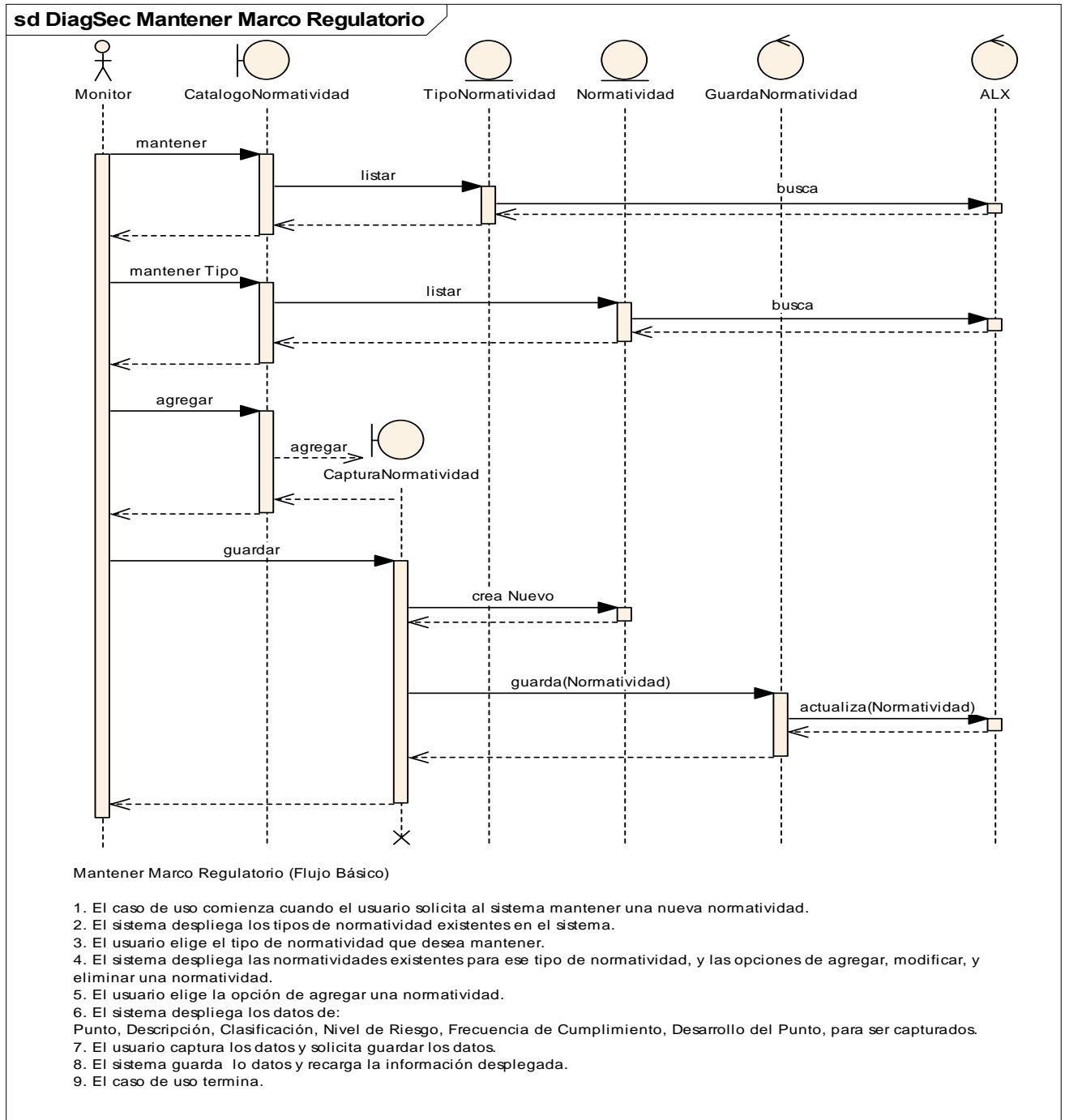


Fig. 3 - 15 Diagrama de secuencias: Mantener Marco Regulatorio

Diagrama de Componentes

La figura 3 -16 muestra el diagrama de componentes para el sistema Cumple.

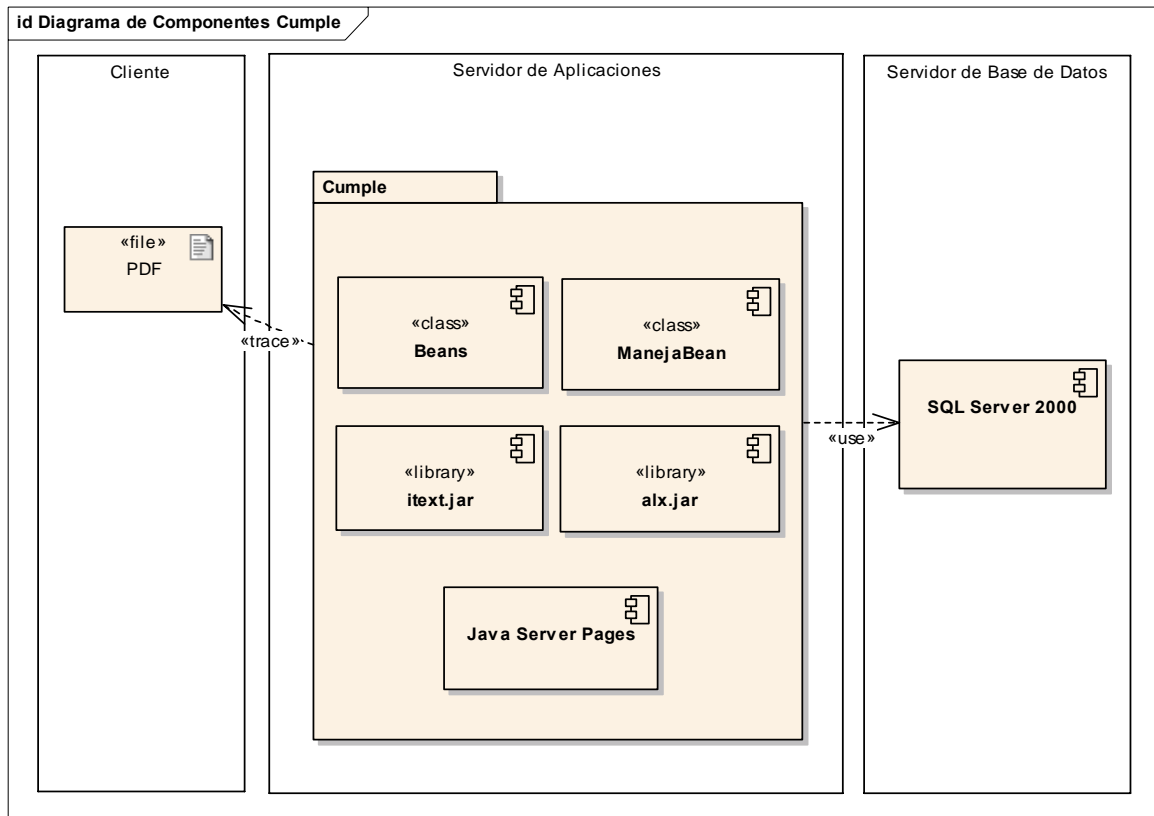


Fig. 3 - 16 Diagrama de Componentes

Diagrama de Despliegue

La figura 3 -17 muestra el diagrama de despliegue para el sistema Cumple.

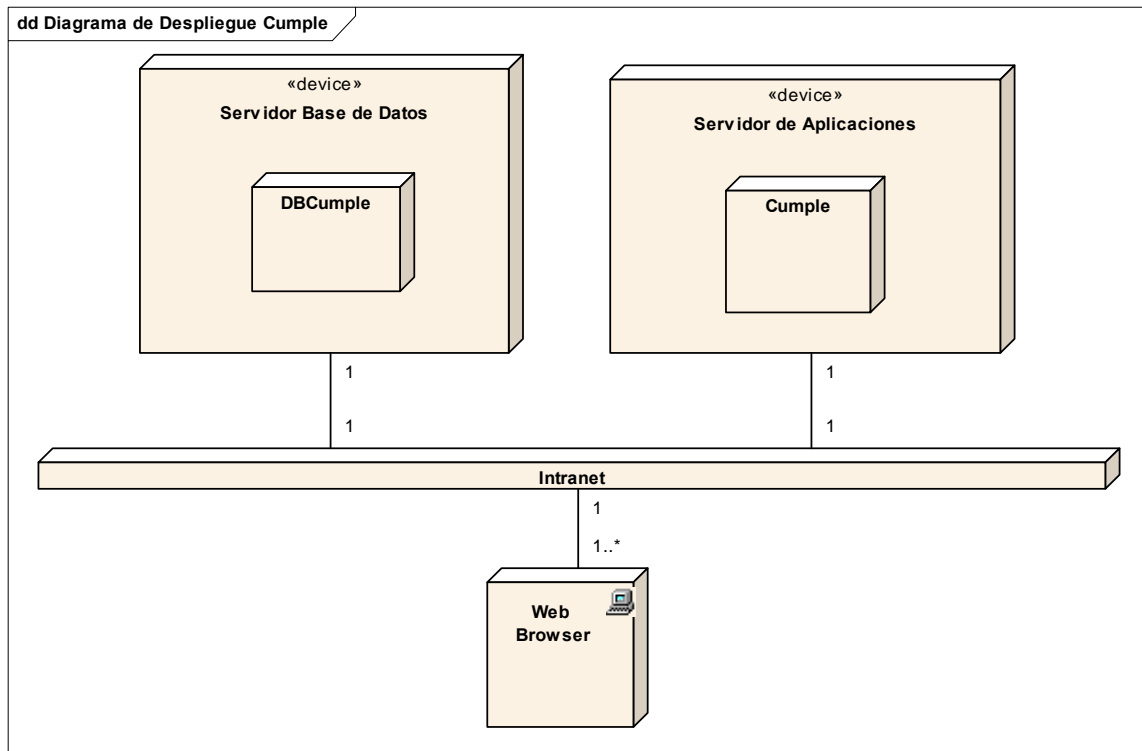


Fig. 3 - 17 Diagrama de Despliegue

3.7 Modelo Relacional

Diseño de la Base de Datos

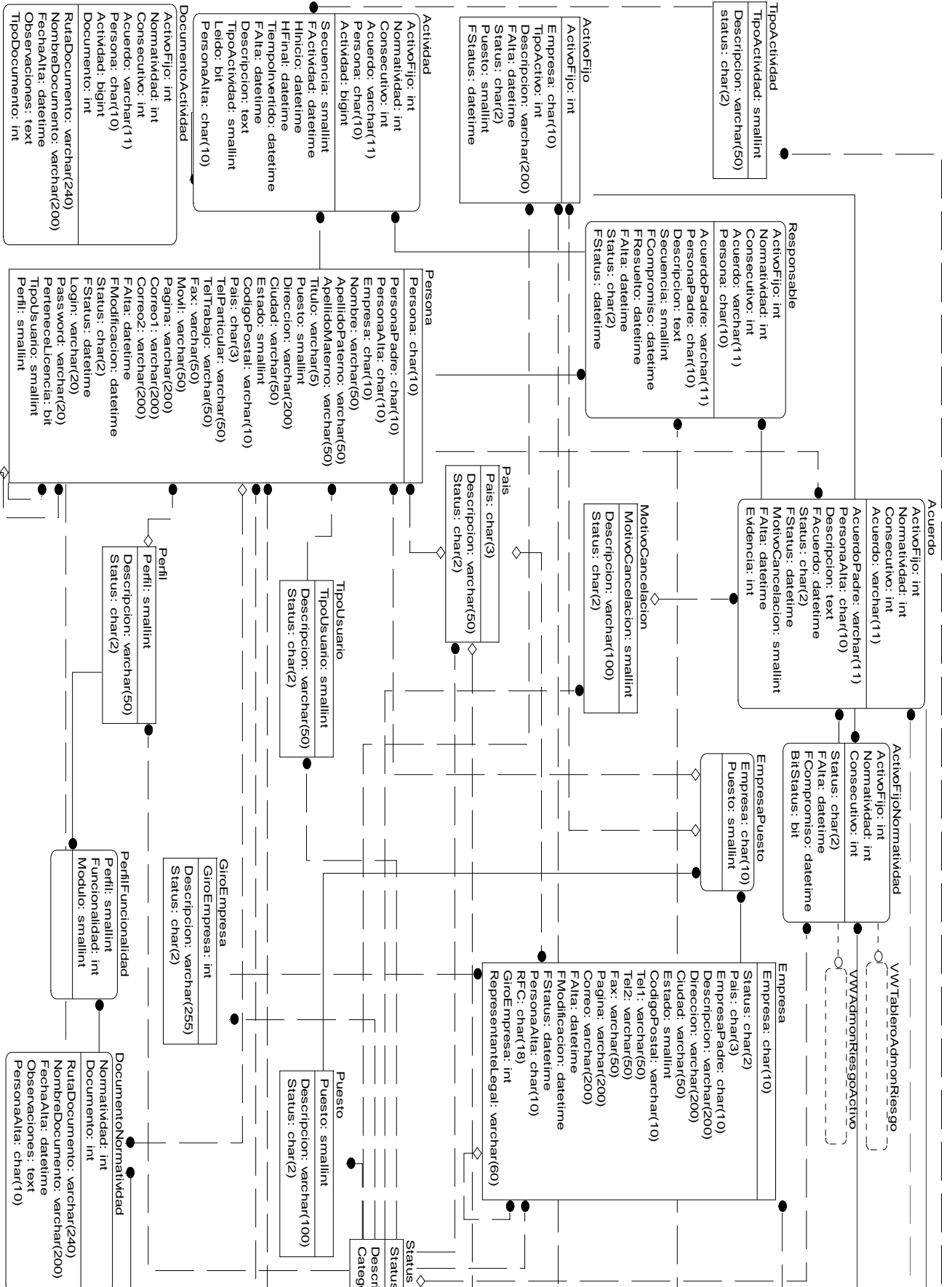
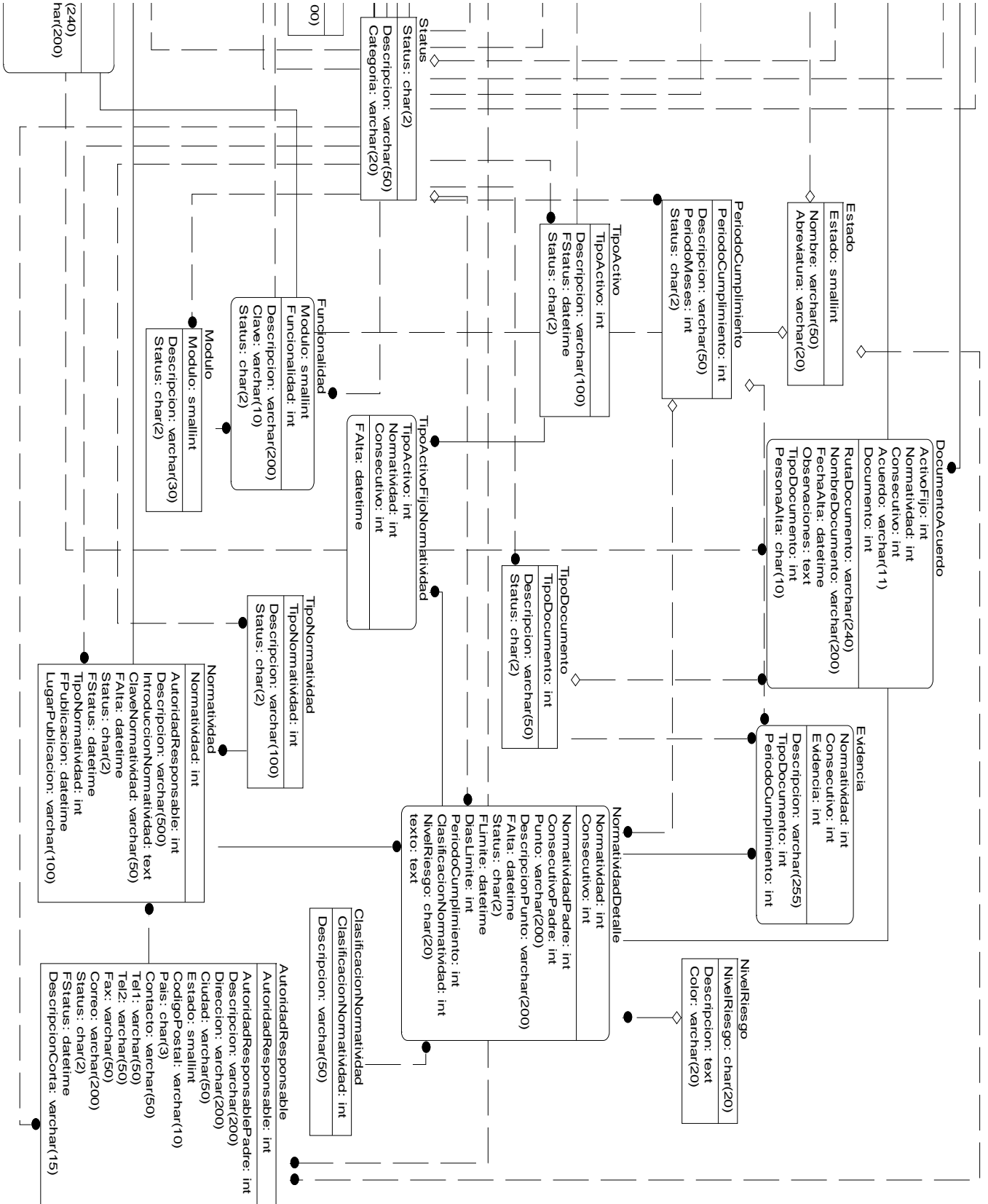


Fig. 3 - 18 Diagrama Entidad-Relación



Diccionario de Datos

Diagrama Entidad Relación

Tipo: **Paquete**

Paquete: Modelo de datos.

Detalle: Creado 05/01/2006.

Descripción: Diagrama entidad relación de la base de datos del sistema CUMPLE.

Actividad

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa el detalle de las acciones realizadas por los responsables en función de un acuerdo determinado para el seguimiento de evidencias.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-----------------|----------|-----------|---|
| Falso | Persona | char | Falso | Identificador del Responsable de la actividad realizada. |
| Falso | Acuerdo | varchar | Falso | Identificador del acuerdo al que está ligada la actividad. |
| Verdadero | Actividad | bigint | Verdadero | Identificador de la actividad. |
| Falso | Secuencia | smallint | Verdadero | Consecutivo de captura de la actividad. |
| Falso | FActividad | datetime | Falso | Fecha de realización de la actividad. |
| Falso | HInicio | datetime | Falso | Hora de Inicio de la actividad. |
| Falso | HFinal | datetime | Falso | Hora de término de la actividad. |
| Falso | TiempoInvertido | datetime | Falso | Tiempo de desarrollo de la actividad. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de registro de la actividad. |
| Falso | Descripcion | text | Verdadero | Descripción de las acciones realizadas. |
| Falso | TipoActividad | smallint | Verdadero | Identificador de la Clasificación de actividad realizada. |
| Falso | Leido | bit | Falso | Indica si la actividad ya fue leída por la gestión del acuerdo. |
| Falso | PersonaAlta | char | Verdadero | Usuario que da de alta la actividad. |

ActivoFijo

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de activos fijos (propiedades, bienes materiales o derechos) sobre los cuales se aplicara las normatividades existentes, para una empresa determinada.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|------------|------|-----------|---|
| Verdadero | ActivoFijo | int | Verdadero | Identificador del activo fijo. |
| Falso | Empresa | char | Verdadero | Identificador de la empresa a la que está relacionado el activo |

| | | | | |
|-------|-------------|----------|-----------|---|
| | | | | fijo. |
| Falso | TipoActivo | int | Verdadero | Identificador de la clasificación del activos fijos. |
| Falso | Descripcion | varchar | Falso | Descripción del activo fijo. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de registro del activo fijo. |
| Falso | Status | char | Verdadero | Estado activo o inactivo del activo fijo en la empresa. |
| Falso | Puesto | smallint | Falso | Identificador del puesto del activo fijo. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado del activo fijo. |

ActivoFijoAcuerdo

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la relación de los acuerdos capturados para una determinada normatividad ligada a los activos fijos de la empresa.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|--------------|---------|-----------|--|
| Verdadero | ActivoFijo | int | Verdadero | Identificador del activo fijo. |
| Verdadero | Normatividad | int | Verdadero | Identificador de la normatividad. |
| Verdadero | Consecutivo | int | Verdadero | Consecutivo de captura de acuerdos en función de Activo fijo y normatividad. |
| Verdadero | Acuerdo | varchar | Verdadero | Identificador del acuerdo. |

ActivoFijoNormatividad

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la relación de los activos fijos de una empresa ligados a una determinada normatividad.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|--------------|----------|-----------|---|
| Verdadero | ActivoFijo | int | Verdadero | Identificador del activo fijo. |
| Verdadero | Normatividad | int | Verdadero | Identificador de la normatividad. |
| Verdadero | Consecutivo | int | Verdadero | Consecutivo de captura de los activos fijos ligados a una normatividad. |
| Falso | Status | char | Falso | Estado de la relación del activo fijo en función de la normatividad. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de registro de la relación del activo fijo a la normatividad. |
| Falso | FInicial | datetime | Falso | Fecha de Inicio de gestión de normatividad del activo fijo relacionado. |

ActivoFijoNormatividadResponsable

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la relación de los responsables del seguimiento y gestión de la normatividad aplicada a un activo fijo de una empresa determinada.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------------|----------|-----------|--|
| Verdadero | ActivoFijo | int | Verdadero | Identificador del activo fijo. |
| Verdadero | Normatividad | int | Verdadero | Identificador de la normatividad. |
| Verdadero | Consecutivo | int | Verdadero | Consecutivo de captura de los activos fijos ligados a una normatividad. |
| Verdadero | Empresa | char | Verdadero | Identificador de la empresa. |
| Verdadero | Puesto | smallint | Verdadero | Identificador del puesto de la persona responsable en la empresa. |
| Falso | TipoResponsabilidad | int | Falso | Identificador del tipo de responsabilidad del activo fijo en la empresa. |

Acuerdo

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista de los acuerdos que se capturan para el seguimiento de evidencias en el cumplimiento de las normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------------|----------|-----------|--|
| Verdadero | Acuerdo | varchar | Verdadero | Identificador del acuerdo. |
| Falso | AcuerdoPadre | varchar | Falso | Identificador del acuerdo al que pueden pertenecer uno o más acuerdos derivados del seguimiento. |
| Falso | PersonaAlta | char | Verdadero | Identificador de la persona que da de alta el acuerdo. |
| Falso | Descripcion | text | Verdadero | Descripción del acuerdo. |
| Falso | FAcuerdo | datetime | Verdadero | Fecha de compromiso del acuerdo. |
| Falso | Status | char | Verdadero | Estado de seguimiento del acuerdo Resuelto, No Resuelto, Cancelado. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado del acuerdo. |
| Falso | Prioridad | int | Verdadero | Prioridad de resolución del acuerdo. |
| Falso | MotivoCancelacion | smallint | Falso | Identificador del motivo de cancelación de un acuerdo si es al caso. |
| Falso | Proyecto | varchar | Verdadero | Identificador del proyecto del acuerdo. |
| Falso | Tema | varchar | Verdadero | Identificador del tema al que se refiere el acuerdo. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de captura del acuerdo. |

| | | | | |
|-------|-------------|-----|-------|--|
| Falso | TipoAcuerdo | int | Falso | Identificador de la clasificación del acuerdo. |
| Falso | Evidencia | int | Falso | Identificador de la evidencia para el seguimiento del acuerdo. |

AutoridadResponsable

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de autoridades responsables de crear, cumplir y dar seguimiento a las normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------------------|----------|-----------|---|
| Verdadero | AutoridadResponsable | int | Verdadero | Identificador de la autoridad responsable. |
| Falso | AutoridadResponsablePadre | int | Falso | Identificador de la autoridad responsable a la que pueden pertenecer una o más autoridades responsables creadoras de normatividades |
| Falso | Descripcion | varchar | Verdadero | Nombre de la autoridad responsable. |
| Falso | DescripcionCorta | varchar | Verdadero | Son las siglas de la autoridad responsable. |
| Falso | Direccion | varchar | Falso | Descripción de la dirección donde se ubica la autoridad responsable. |
| Falso | Ciudad | varchar | Falso | Descripción de la ciudad donde se ubica la autoridad responsable. |
| Falso | Estado | smallint | Falso | Identificador del estado donde se ubica la autoridad responsable. |
| Falso | CodigoPostal | varchar | Falso | Identificador del país donde se ubica la autoridad responsable. |
| Falso | Pais | char | Falso | Descripción del país donde se ubica la autoridad responsable. |
| Falso | Contacto | varchar | Falso | Nombre del contacto dentro de las instalaciones de la autoridad responsable. |
| Falso | Tel1 | varchar | Falso | Número de teléfono de la autoridad responsable. |
| Falso | Tel2 | varchar | Falso | Número de teléfono de la autoridad responsable. |
| Falso | Fax | varchar | Falso | Número de fax de la autoridad responsable. |
| Falso | Correo | varchar | Falso | Correo electrónico de contacto de la autoridad responsable. |
| Falso | Status | char | Verdadero | Estado activo o inactivo de la autoridad responsable. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado de la autoridad responsable. |

ClasificacionNormatividad

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista del tipo o clasificación de las normatividades capturadas.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------------------|---------|-----------|--|
| Verdadero | ClasificacionNormatividad | int | Verdadero | Identificador de clasificación de normatividad. |
| Falso | Descripcion | varchar | Verdadero | Descripción de la clasificación de la normatividad (Informativa u obligatoria) |

Directorio

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la estructura de organigrama de la empresa, para el seguimiento de normatividades, en un proyecto determinado.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-------|-----------------|----------|-----------|--|
| Falso | Directorio | varchar | Verdadero | Identificador del directorio de la empresa. |
| Falso | Proyecto | varchar | Falso | Identificador del proyecto de la empresa. |
| Falso | Empresa | char | Falso | Identificador de la empresa. |
| Falso | Tipo | int | Falso | Identificador del tipo de directorio. |
| Falso | DirectorioPadre | varchar | Falso | Identificador del directorio de la empresa ha el que pueden pertenecer uno o más directorios por proyecto en la empresa. |
| Falso | PersonaAlta | char | Falso | Identificador de la persona que captura el directorio. |
| Falso | Descripcion | varchar | Verdadero | Descripción del directorio por proyecto. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de captura del directorio. |

Documento

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista y relación de los documentos o archivos ligados a un acuerdo para el seguimiento de evidencias en el cumplimiento de normatividades relacionadas a un activo fijo.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-----------------|----------|-----------|---|
| Verdadero | Documento | int | Verdadero | Identificador del documento. |
| Falso | RutaDocumento | varchar | Verdadero | Ruta del repositorio de los documentos. |
| Falso | Actividad | bigint | Falso | Identificador de la actividad. |
| Falso | NombreDocumento | varchar | Verdadero | Nombre y extensión del archivo. |
| Falso | FechaAlta | datetime | Verdadero | Fecha de alta de captura del |

| | | | | |
|-------|---------------|---------|-----------|--|
| | | | | documento. |
| Falso | Observaciones | text | Falso | Descripción del contenido del documento. |
| Falso | Acuerdo | varchar | Falso | Identificador del acuerdo. |
| Falso | Privado | int | Verdadero | Bandera que indica si el documento es de carácter privado. |
| Falso | PersonaAlta | char | Verdadero | Persona que da de alta el documento. |
| Falso | TipoDocumento | int | Falso | Identificador de la clasificación del documento. |
| Falso | Evidencia | int | Falso | Identificador de la evidencia. |
| Falso | ActivoFijo | int | Falso | Identificador del activo fijo. |
| Falso | Normatividad | int | Falso | Identificador de la normatividad. |

Empresa

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de las empresas capturadas para el seguimiento de normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------|----------|-----------|---|
| Verdadero | Empresa | char | Verdadero | Identificador de la empresa. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo de la empresa. |
| Falso | Pais | char | Falso | Identificador del país donde se ubica la empresa. |
| Falso | EmpresaPadre | char | Falso | Identificador de la empresa ha el que pueden pertenecer una o más empresas. |
| Falso | Descripcion | varchar | Verdadero | Nombre de la empresa. |
| Falso | Direccion | varchar | Falso | Descripción de la dirección donde se ubica la empresa. |
| Falso | Ciudad | varchar | Falso | Descripción de la ciudad donde se ubica la empresa. |
| Falso | Estado | smallint | Falso | Identificador de el estado donde se ubica la empresa. |
| Falso | CodigoPostal | varchar | Falso | Descripción de la ciudad donde se ubica la empresa. |
| Falso | Tel1 | varchar | Falso | Número de teléfono de la empresa. |
| Falso | Tel2 | varchar | Falso | Número de teléfono de la empresa. |
| Falso | Fax | varchar | Falso | Número de fax de la empresa. |
| Falso | Pagina | varchar | Falso | Dirección del sitio de internet de la empresa. |
| Falso | Correo | varchar | Falso | Dirección de correo electrónico de contacto de la empresa. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de captura de la empresa. |
| Falso | FModificacion | datetime | Falso | Fecha de modificación de datos de la empresa. |

| | | | | |
|-------|--------------------|----------|-----------|---|
| Falso | FStatus | datetime | Verdadero | Fecha de estado de la empresa. |
| Falso | PersonaAlta | char | Falso | Identificador de la persona que captura la empresa. |
| Falso | RFC | char | Falso | Descripción del Registro Federal de Contribuyentes de la empresa. |
| Falso | GiroEmpresa | int | Verdadero | Identificador del giro de la empresa. |
| Falso | TipoEmpresa | int | Falso | Identificador de la clasificación de la empresa. |
| Falso | RepresentanteLegal | varchar | Falso | Nombre del representante legal de la empresa. |

EmpresaPuesto

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la relación de puestos en el organigrama de la empresa.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------|----------|-----------|---|
| Verdadero | Empresa | char | Verdadero | Identificador de la empresa. |
| Verdadero | Puesto | smallint | Verdadero | Identificador del puesto en el organigrama de la empresa. |

Estado

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista de estados de la república para la ubicación de empresas y autoridades responsables.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|----------|-----------|---|
| Verdadero | Estado | smallint | Verdadero | Identificador del estado de la república. |
| Falso | Nombre | varchar | Verdadero | Nombre del estado de la república. |
| Falso | Abreviatura | varchar | Verdadero | Abreviatura del estado de la república. |

Evidencia

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista y relación de las evidencias para el cumplimiento de normatividades capturadas.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------|---------|-----------|---|
| Verdadero | Evidencia | int | Verdadero | Identificador de la evidencia. |
| Falso | Normatividad | int | Verdadero | Identificador de la normatividad. |
| Falso | Consecutivo | int | Verdadero | Consecutivo de captura de las evidencias. |
| Falso | Descripcion | varchar | Falso | Descripción de los datos y contenido de la evidencia. |
| Falso | TipoDocumento | int | Verdadero | Identificador de la clasificación de la evidencia |

| | | | | |
|-------|---------------------|-----|-------|--|
| Falso | PeriodoCumplimiento | int | Falso | Identificador del periodo de cumplimiento de la evidencia. |
|-------|---------------------|-----|-------|--|

Funcionalidad

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de funcionalidades por modulo en el sistema.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------|----------|-----------|---|
| Verdadero | Modulo | smallint | Verdadero | Identificador del Modulo del sistema. |
| Verdadero | Funcionalidad | int | Verdadero | Identificador de la funcionalidad. |
| Falso | Descripcion | varchar | Verdadero | Descripción de la funcionalidad del sistema. |
| Falso | Clave | varchar | Verdadero | Identificador asignado para acceder a la funcionalidad |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo de la funcionalidad del sistema. |

GiroEmpresa

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los giros o funciones de las empresas.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|---------|-----------|--|
| Verdadero | GiroEmpresa | int | Verdadero | Identificador del giro de la empresa. |
| Falso | Descripcion | varchar | Verdadero | Descripción del giro de la empresa. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del giro de la empresa. |

Modulo

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los módulos en los que está dividido el sistema.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|----------|-----------|--|
| Verdadero | Modulo | smallint | Verdadero | Identificador del modulo. |
| Falso | Descripcion | varchar | Verdadero | Descripción del nombre del modulo del sistema. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del modulo del sistema. |

MotivoCancelacion

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los motivos de cancelación de acuerdos.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------------|----------|-----------|---|
| Verdadero | MotivoCancelacion | smallint | Verdadero | Identificador del motivo de cancelación. |
| Falso | Descripcion | varchar | Verdadero | Descripción del motivo de cancelación. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del motivo de cancelación. |

NivelRiesgo

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista de niveles de riesgo por incumplimiento de normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|---------|-----------|--|
| Verdadero | NivelRiesgo | char | Verdadero | Identificador del nivel de riesgo. |
| Falso | Descripcion | varchar | Verdadero | Descripción del nivel de riesgo. |
| Falso | Color | varchar | Falso | Descripción del color asignado al nivel de riesgo para visualización de indicadores. |

Normatividad

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista de los datos generales de normatividades capturadas para seguimiento.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|--------------------------|----------|-----------|---|
| Verdadero | Normatividad | int | Verdadero | Identificador de la normatividad. |
| Falso | AutoridadResponsable | int | Verdadero | Identificador de la autoridad responsable de dar seguimiento a la normatividad. |
| Falso | Descripcion | varchar | Verdadero | Descripción del título de la normatividad. |
| Falso | IntroduccionNormatividad | text | Falso | Descripción general del contenido de la normatividad. |
| Falso | ClaveNormatividad | varchar | Falso | Descripción de la clave de la normatividad. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de captura de la normatividad. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo de la normatividad. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado de la normatividad. |
| Falso | TipoNormatividad | int | Verdadero | Identificador de la clasificación de la normatividad. |
| Falso | FPublicacion | datetime | Verdadero | Fecha de publicación de la normatividad. |
| Falso | LugarPublicacion | varchar | Falso | Descripción del lugar de publicación de la normatividad. |

NormatividadDetalle

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista y relación de los datos específicos (puntos) de las normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------------------|----------|-----------|---|
| Verdadero | Normatividad | int | Verdadero | Identificador de la normatividad. |
| Verdadero | Consecutivo | int | Verdadero | Identificador del consecutivo de captura de la normatividad. |
| Falso | NormatividadPadre | int | Falso | Identificador de la normatividad al que pueden pertenecer una o más normatividades. |
| Falso | ConsecutivoPadre | int | Falso | Identificador del consecutivo de captura de normatividades al que pueden pertenecer uno o más consecutivos de normatividades. |
| Falso | Punto | varchar | Falso | Descripción del Número de Punto en el listado de la normatividad. |
| Falso | DescripcionPunto | varchar | Falso | Descripción general del contenido del punto de la normatividad. |
| Falso | FALta | datetime | Verdadero | Fecha de alta de captura del punto de la normatividad. |
| Falso | Status | char | Falso | Estado Activo o Inactivo del punto de la normatividad. |
| Falso | FLimite | datetime | Falso | Fecha límite de cumplimiento del punto de la normatividad. |
| Falso | NivelRiesgo | char | Falso | Identificador del nivel de riesgo de cumplimiento de la normatividad. |
| Falso | DiasLimite | int | Falso | Número de días para el cumplimiento del punto de la normatividad. |
| Falso | PeriodoCumplimiento | int | Falso | Identificador del periodo de cumplimiento de la normatividad. |
| Falso | ClasificacionNormatividad | int | Verdadero | Identificador de la clasificación del punto de la normatividad. |

NormatividadDetalleTexto

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa el contenido de las especificaciones de las normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|--------------|------|-----------|-----------------------------------|
| Verdadero | Normatividad | int | Verdadero | Identificador de la normatividad. |
| Verdadero | Consecutivo | int | Verdadero | Identificador del consecutivo |

| | | | | |
|-------|-------|------|-----------|--|
| | | | | de captura de la normatividad. |
| Falso | Texto | text | Verdadero | Descripción del contenido de los puntos de las normatividades. |

Pais

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de países para la ubicación de empresas y autoridades responsables.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|---------|-----------|------------------------------------|
| Verdadero | Pais | char | Verdadero | Identificador del país. |
| Falso | Descripcion | varchar | Verdadero | Nombre del país. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del país. |

Perfil

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de perfiles para el acceso de los usuarios al sistema.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|----------|-----------|--------------------------------------|
| Verdadero | Perfil | smallint | Verdadero | Identificador del perfil. |
| Falso | Descripcion | varchar | Verdadero | Nombre del perfil. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del perfil. |

PerfilFuncionalidad

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la relación de los perfiles de acceso con las funcionalidades y módulos del sistema.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------|----------|-----------|--|
| Verdadero | Perfil | smallint | Verdadero | Identificador del perfil de acceso. |
| Verdadero | Funcionalidad | int | Verdadero | Identificador de la funcionalidad del sistema. |
| Verdadero | Modulo | smallint | Verdadero | Identificador del modulo del sistema |

PeriodoCumplimiento

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de periodos de cumplimiento para normatividades y evidencias.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------------|---------|-----------|--|
| Verdadero | PeriodoCumplimiento | int | Verdadero | Identificador del periodo de cumplimiento. |
| Falso | Descripcion | varchar | Verdadero | Descripción del periodo de cumplimiento. |

| | | | | |
|-------|--------------|------|-----------|---|
| Falso | PeriodoMeses | int | Falso | Numero de meses en el que aplica el periodo de cumplimiento |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del periodo de cumplimiento. |

Persona

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los datos de los usuarios del sistema y/o personas responsables.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-----------------|----------|-----------|---|
| Verdadero | Persona | char | Verdadero | Identificador de la persona. |
| Falso | PersonaPadre | char | Falso | Identificador de la persona a la que pueden estar ligadas una o más personas. |
| Falso | PersonaAlta | char | Verdadero | Identificador de la persona que da de alta la captura del usuario. |
| Falso | Empresa | char | Verdadero | Identificador de la empresa a la que pertenece la persona. |
| Falso | Nombre | varchar | Verdadero | Descripción del nombre de la persona. |
| Falso | ApellidoPaterno | varchar | Verdadero | Descripción del apellido paterno de la persona. |
| Falso | ApellidoMaterno | varchar | Falso | Descripción del apellido materno de la persona. |
| Falso | Titulo | varchar | Falso | Descripción del título de la persona. |
| Falso | Puesto | smallint | Falso | Identificador del puesto de la persona. |
| Falso | Direccion | varchar | Falso | Descripción de la dirección donde se ubica la persona. |
| Falso | Ciudad | varchar | Falso | Descripción de la ciudad donde se ubica la persona. |
| Falso | Estado | smallint | Falso | Identificador del estado de la república donde se ubica la persona. |
| Falso | CodigoPostal | varchar | Falso | Descripción del código postal donde se ubica la empresa. |
| Falso | Pais | char | Falso | Identificador del país donde se ubica la persona. |
| Falso | TelParticular | varchar | Falso | Número de teléfono particular de la persona. |
| Falso | TelTrabajo | varchar | Falso | Número de teléfono laboral de la persona. |
| Falso | Fax | varchar | Falso | Número de fax de la persona. |
| Falso | Movil | varchar | Falso | Número de teléfono móvil de la persona. |
| Falso | Página | varchar | Falso | Dirección del sitio de internet de la persona. |
| Falso | Correo1 | varchar | Falso | Dirección de correo electrónico de la persona. |

| | | | | |
|-------|-------------------|----------|-----------|--|
| Falso | Correo2 | varchar | Falso | Dirección de correo electrónico de la persona. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de captura de la persona. |
| Falso | FModificacion | datetime | Verdadero | Fecha de modificación de datos de la persona. |
| Falso | Status | char | Verdadero | Estado activo o inactivo de la persona. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado de la persona. |
| Falso | Login | varchar | Falso | Descripción de la clave de acceso de usuario en el sistema. |
| Falso | Password | varchar | Falso | Descripción de la contraseña de acceso de usuario en el sistema. |
| Falso | PerteneceLicencia | bit | Verdadero | Bandera que indica el licenciamiento del usuario en el sistema. |
| Falso | TipoUsuario | smallint | Verdadero | Identificador de la clasificación del usuario. |
| Falso | Perfil | smallint | Falso | Identificador del perfil de acceso. |

Proyecto

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de proyectos de las empresas.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-------|---------------|----------|-----------|---|
| Falso | Proyecto | varchar | Verdadero | Identificador del proyecto. |
| Falso | ProyectoPadre | varchar | Falso | Identificador del proyecto al que pueden estar ligados uno o más proyectos. |
| Falso | FInicio | datetime | Falso | Fecha de Inicio de proyecto. |
| Falso | FFinal | datetime | Falso | Fecha de término de proyecto. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de captura del proyecto. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del proyecto. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado del proyecto. |
| Falso | Descripcion | varchar | Verdadero | Descripción del proyecto. |
| Falso | PersonaAlta | char | Verdadero | Identificador del usuario que da de alta el proyecto. |

Puesto

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de puestos ligados a las personas de las empresas.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|----------|-----------|---------------------------|
| Verdadero | Puesto | smallint | Verdadero | Identificador del puesto. |
| Falso | Descripcion | varchar | Verdadero | Descripción del puesto. |

| | | | | |
|-------|--------|------|-----------|--------------------------------------|
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del puesto. |
|-------|--------|------|-----------|--------------------------------------|

Responsable

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista y relación de las responsabilidades asignadas a personas para el seguimiento del cumplimiento de normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|--------------|----------|-----------|--|
| Verdadero | Acuerdo | varchar | Verdadero | Identificador del acuerdo. |
| Verdadero | Persona | char | Verdadero | Identificador de la persona. |
| Falso | AcuerdoPadre | varchar | Falso | Identificador del acuerdo al que pueden estar relacionados uno o más acuerdos. |
| Falso | PersonaPadre | char | Falso | Identificador de la persona al que pueden estar ligados una o más personas. |
| Falso | Descripcion | text | Verdadero | Descripción de la responsabilidad. |
| Falso | Secuencia | smallint | Verdadero | Consecutivo de captura de responsable. |
| Falso | FCompromiso | datetime | Falso | Fecha de compromiso de la responsabilidad. |
| Falso | FResuelto | datetime | Falso | Fecha de solución de la responsabilidad. |
| Falso | FAlta | datetime | Verdadero | Fecha de alta de captura de la responsabilidad. |
| Falso | Status | char | Verdadero | Estado de Resuelto o No resuelto de la responsabilidad. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado de la responsabilidad. |

Status

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los estados de los registros de las diferentes entidades del sistema.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|---------|-----------|--|
| Verdadero | Status | char | Verdadero | Identificador del estatus. |
| Falso | Descripcion | varchar | Verdadero | Descripción del estatus. |
| Falso | Categoria | varchar | Falso | Clasificación de la entidad del estatus. |

Tema

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los temas de los proyectos de la empresa.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-------|--------|---------|-----------|-------------------------|
| Falso | Tema | varchar | Verdadero | Identificador del tema. |

| | | | | |
|-------|-------------|----------|-----------|------------------------------------|
| Falso | FAlta | datetime | Verdadero | Fecha de alta de captura del tema. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del tema. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado del tema. |
| Falso | Descripcion | varchar | Verdadero | Descripción del tema. |

TipoActividad

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los tipos de actividad de las acciones ligadas a los acuerdos.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------|----------|-----------|---|
| Verdadero | TipoActividad | smallint | Verdadero | Identificador del tipo de actividad. |
| Falso | Descripcion | varchar | Verdadero | Descripcion del tipo de actividad. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del tipo de actividad. |

TipoActivo

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los tipos de activo fijo.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|----------|-----------|---|
| Verdadero | TipoActivo | int | Verdadero | Identificador del tipo de activo fijo. |
| Falso | Descripcion | varchar | Verdadero | Descripción del tipo de activo fijo. |
| Falso | FStatus | datetime | Verdadero | Fecha de estado del tipo de activo fijo. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del tipo de activo fijo. |

TipoActivoFijoNormatividad

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la relación de los tipos de activo fijo y normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|--------------|----------|-----------|--|
| Verdadero | TipoActivo | int | Verdadero | Identificador del tipo de activo fijo. |
| Verdadero | Normatividad | int | Verdadero | Identificador de la normatividad. |
| Verdadero | Consecutivo | int | Verdadero | Consecutivo de captura de la normatividad. |
| Falso | FAlta | datetime | Falso | Fecha de alta de captura de la relación de tipo de activo fijo y normatividad. |

TipoDocumento

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista de los tipos de documentos para el seguimiento de evidencias, acuerdos y normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|---------------|---------|-----------|---|
| Verdadero | TipoDocumento | int | Verdadero | Identificador del tipo de documento. |
| Falso | Descripcion | varchar | Verdadero | Descripción del tipo de documento. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del tipo de documento. |

TipoEmpresa

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista de tipos de empresa.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-------|-------------|---------|-----------|---|
| Falso | TipoEmpresa | int | Verdadero | Identificador del tipo de empresa. |
| Falso | Descripcion | varchar | Verdadero | Descripción del tipo de empresa. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del tipo de empresa. |

TipoNormatividad

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista de tipos de normatividades.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|------------------|---------|-----------|--|
| Verdadero | TipoNormatividad | int | Verdadero | Identificador del tipo de normatividad. |
| Falso | Descripcion | varchar | Verdadero | Descripción del tipo de normatividad |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del tipo de normatividad. |

TipoPrioridad

Base de datos: SQL Server 2000, Tipo: tabla

Descripción: Representa la lista de los tipos de prioridad para los acuerdos.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-------|---------------|---------|-----------|--------------------------------------|
| Falso | TipoPrioridad | int | Verdadero | Identificador del tipo de prioridad. |
| Falso | Descripcion | varchar | Verdadero | Descripción del tipo de prioridad. |
| | | | | |

TipoUsuario

Base de datos: SQL Server 2000, *Tipo:* tabla

Descripción: Representa la lista de los tipos de usuario.

Columnas

| PK | Nombre | Tipo | No Nulo | Descripción |
|-----------|-------------|----------|-----------|---|
| Verdadero | TipoUsuario | smallint | Verdadero | Identificador del tipo de usuario. |
| Falso | Descripcion | varchar | Verdadero | Descripción del tipo de usuario. |
| Falso | Status | char | Verdadero | Estado Activo o Inactivo del tipo de activo fijo. |

Capítulo 4

Sistema de Cumplimiento Regulatorio

Etapa de Construcción

Esta etapa consiste en obtener una versión Beta del Sistema de Cumplimiento Regulatorio. En este capítulo describo la metodología y las principales funcionalidades con las que cuenta el sistema.

4.1 Sistema de Cumplimiento Regulatorio

El Sistema de Cumplimiento Regulatorio CUMPLE, es una aplicación WEB, creada para registrar, administrar y controlar el cumplimiento de las disposiciones legales aplicables a una o varias empresas, ante las diversas autoridades que las rigen.

Áreas de aplicación

Entre las principales áreas de aplicación destacan:

- Gestión y control de las disposiciones legales
- Control documental asociado a disposiciones legales y/o certificaciones.
- Aplicación de auditorías internas y externas
- Control de certificaciones

Objetivos

- ✓ Eliminar la posibilidad de sanciones por parte de la autoridad competente debido a incumplimientos de las disposiciones legales aplicables a la empresa.
- ✓ Conocer el nivel de cumplimiento y de riesgo por empresa con la finalidad de tomar mejores decisiones de negocio previas a cualquier contingencia que pudiera presentarse
- ✓ Llevar un control estadístico del cumplimiento por empresa.
- ✓ Llevar un control estadístico de cumplimiento por responsable.
- ✓ Contar con información oportuna y vigente para su consulta en un solo repositorio.
- ✓ Definir responsables y fechas compromiso por disposición legal, dando seguimiento hasta que estas se concluyan.
- ✓ Agilizar la gestión de trámites y documentar las evidencias de cumplimiento.

4.2 Detalles de la Metodología empleada.

Etapas 1. Configuración

| | |
|----------------------------------|---|
| Responsable de | Monitor y Administrador. |
| ¿En qué consiste? | Consiste en listar y capturar los aspectos a administrar en el sistema, los |
| Área del Sistema que lo soporta: | <p>Áreas de Administrador para los catálogos diversos.</p> <p>Área de Catálogos generales: CONFIGURACIÓN, Documentos, Giros de empresas, Puentes, Personas, Tipos de Activos, para las Normatividades y Evidencias que se administrarán con el sistema.</p> <p>El submenú TIPO ACTIVO FIJO para la asociación dependiendo de su configuración.</p> <p>El submenú Catálogos para capturar a los posibles responsables.</p> <p>El marco legal aplicable a cada empresa.</p> <p>La evidencia o evidencias requeridas o sugeridas para cumplir una disposición legal.</p> <p>Asociación del marco legal con los Tipos de Activos de la empresa.</p> |

Etapa 2. Asignación de Responsables

| | |
|----------------------------------|---|
| Objetivo: | Asignación de responsables. |
| ¿En qué consiste? | Consiste en asignar todas evidencias a un responsable de cumplirla. Nota: El responsable es un puesto de la empresa. |
| Responsable de captura: | Monitor. |
| Área del Sistema que lo soporta: | Área del usuario en Menú CONFIGURACIÓN: <ul style="list-style-type: none"> ➤ Submenú ACTIVOS FIJOS para asignar un responsable |

Etapa 3. Control y Documentación

| | |
|----------------------------------|---|
| Objetivo: | Dar seguimiento al cumplimiento regulatorio. |
| ¿En qué consiste? | Consiste en dar seguimiento a los responsables de cumplir con las evidencias asignadas. Esto incluye: <ul style="list-style-type: none"> ➤ Control de pendientes propios ➤ Control de pendientes por usuario ➤ Generar acuerdos libres ➤ Documentación de Evidencias |
| Responsable de captura: | Monitor y Responsable del cumplimiento. |
| Área del Sistema que lo soporta: | Área del usuario en Menú PRINCIPAL: <ul style="list-style-type: none"> ➤ Submenú Tareas para revisar los pendientes propios. ➤ Submenú Seguimiento para revisar los pendientes por usuario. ➤ Submenú Seguimiento en Tareas Libres (para levantar no conformidades o acuerdos libres) Área del usuario en Menú ATLAS DE RIESGO: <ul style="list-style-type: none"> ➤ Submenú Estudios y Programas para documentar evidencias. |

Etapa 4. Administración del Riesgo

| | |
|----------------------------------|---|
| Objetivo: | Conocer el estado de Riesgo de una empresa ante una auditoría externa. |
| ¿En qué consiste? | Consiste en dar seguimiento al estado de Riesgo de la empresa, es decir, verificar el cumplimiento, o no cumplimiento de manera general de las disposiciones legales de la empresa. Esto incluye: <ul style="list-style-type: none"> ➤ Control de riesgo por empresa ➤ Control de riesgo por autoridad competente |
| Responsable de captura: | Monitor. |
| Área del Sistema que lo soporta: | Área del usuario en Menú ADMINISTRADOR DE RIESGO: <ul style="list-style-type: none"> ➤ Submenú Administrar Riesgo para administrar el riesgo por empresa y por autoridad competente. |

4.3 Estructura de la aplicación



Fig. 4 - 1: Pantalla de inicio de sesión de CUMPLE

Al iniciar una sesión en el sistema CUMPLE se desplegará la pantalla principal que se muestra en la figura 4-1:

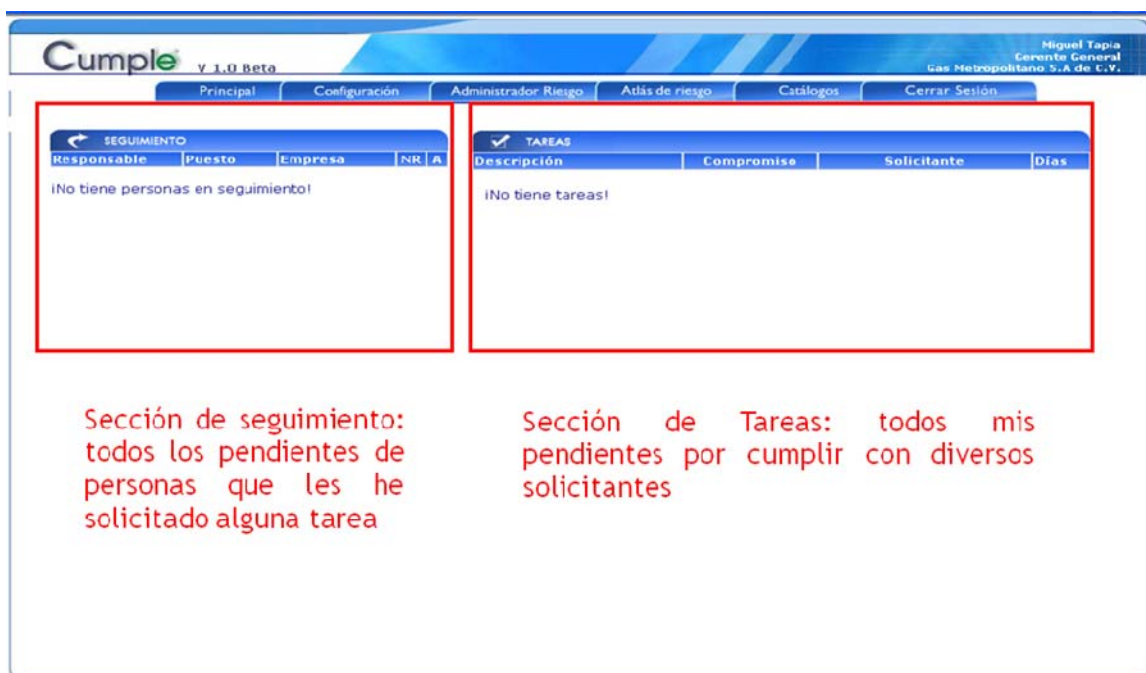


Fig. 4 - 2 Pantalla principal del Sistema.

En la pantalla podemos identificar el usuario que ha iniciado sesión, la versión del sistema y dos secciones importantes de información:

Sección de Seguimiento: Lista la información de las personas que tienen tareas pendientes, así como el número de ellas que están en tiempo para poder ser resueltas y el número de las tareas que ya caducaron, las cuales son más riesgosas.

Sección de Tareas: Lista una breve descripción de las tareas que la persona en sesión tiene pendientes, de cualquier solicitante, la fecha compromiso y los días que restan para poder cumplir la tarea.

Menú Principal del Sistema

PRINCIPAL

El primer elemento que encontramos en el menú principal es PRINCIPAL, que nos dirige a la pantalla principal desde cualquier lugar del sistema. Como submenús, tenemos las opciones de Tareas y Seguimiento. Fig. 4 - 3.

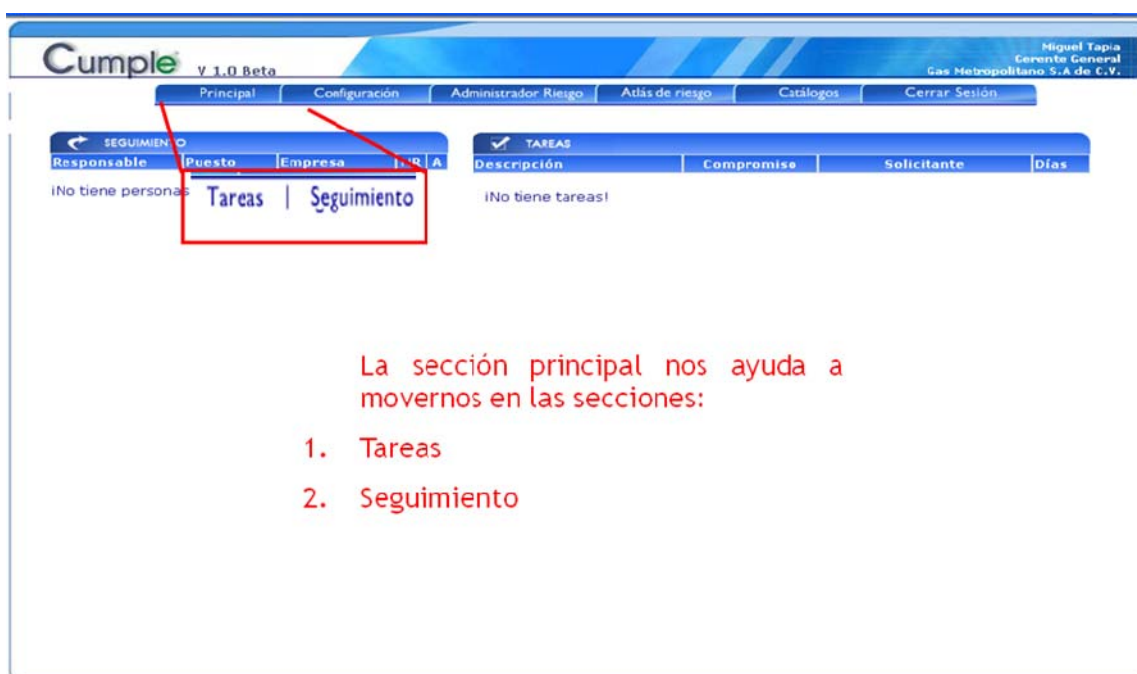


Fig. 4 - 3 Menú PRINCIPAL del Sistema.

El submenú Tareas:

Lista un histórico de forma detallada de las tareas que la persona en sesión tiene pendientes, completas, atrasadas, así como el solicitante de cada una de ellas, la descripción de la tarea completa, a que activo fijo se asocia, la normatividad que cumple, la fecha compromiso establecida, así como los documentos adjuntos sí fuera el caso. Con las opciones de capturar alguna actividad realizada o comentario, y adjuntar documentos. Fig 4 - 4.

El submenú Seguimiento:

De manera similar al submenú Tareas, muestra de forma detallada las tareas pendientes, completas, atrasadas y las características de estas, con la diferencia de que éstas son solicitadas por el usuario en sesión y los responsables son diversos.

Cumple V 1.0.0

Principal | Configuración | Administrador Riesgo | Adías de riesgo | Catálogos | Cerrar Sesión

Tareas | Seguimiento

TAREAS de Aide Marín

SOLICITANTE: Marín Aide | ACTIVIDAD: Todas | TIPO ACTIVO: Todos

RESPONSABLE: Marín Aide | CLAVE: Todos | EMPRESA: Todos

ESTATUS: No Resuelto | ACTIVO: Todos

08/11/2006 AL 08/11/2006

Ordenar por: Activo Fijo

Total : 2 acuerdos

| Solicitante | Tarea | Activo Fijo | Normatividad | CompromisoDías | |
|---|--|--|---|------------------------------|-------------|
| 1 Marín Aide Gerente de Ventas Saitosoft S. A. de C.V. | Tramitar Documentos que exige la norma | Almacén Planta baja Almacén Saitosoft S. A. de C.V. | - NOM-017-STPS-2001 - EQUIPO DE PROTECCION PERSONAL - SELECCION, USO Y MANEJO EN LOS CENTROS DE TRABAJO | 30 noviembre 2006, jueves 22 | Delegar Act |
| 2 Marín Aide Gerente de Ventas Saitosoft S. A. de C.V. | Tramitar Etiquetas para EPP | Almacén Planta baja Almacén Saitosoft S. A. de C.V. | - NOM-017-STPS-2001 - EQUIPO DE PROTECCION PERSONAL - SELECCION, USO Y MANEJO EN LOS CENTROS DE TRABAJO | 30 noviembre 2006, jueves 22 | Delegar Act |

Para facilitar la búsqueda de las tareas se pueden filtrar con las opciones del recuadro superior

Fig. 4 - 4 Submenú TAREAS del Sistema.

CONFIGURACIÓN

El segundo elemento que encontramos en el menú principal es CONFIGURACIÓN, que nos dirige a la sección para configurar el Marco Regulatorio y su asociación con los Tipos de Activo de la empresa, así como también nos permite llevar a cabo la asignación de responsable para que lleve a cabo su cumplimiento. Como submenús, tenemos las opciones de Normatividad, Tipo Activo Fijo y Activos Fijos. Fig. 4 - 5.

El submenú Normatividad:

Lista de manera detallada las Normatividades capturadas, permite la captura de nuevas y la configuración de Evidencias necesarias para cumplir con la disposición legal señalada. Fig. 4 - 6.

El submenú Tipo Activo Fijo:

Lista de manera detallada la clasificación de Activos Fijos (Tipos de Activo) y sus respectivas disposiciones legales asociadas. Esta sección tiene por objetivo llevar a cabo la asociación entre el marco regulatorio y los activos fijos de la empresa para que una vez que están asociados se puedan asignar a un responsable para cumplirlos. Fig. 4 - 7.



Fig. 4 - 5 Menú CONFIGURACIÓN del Sistema.

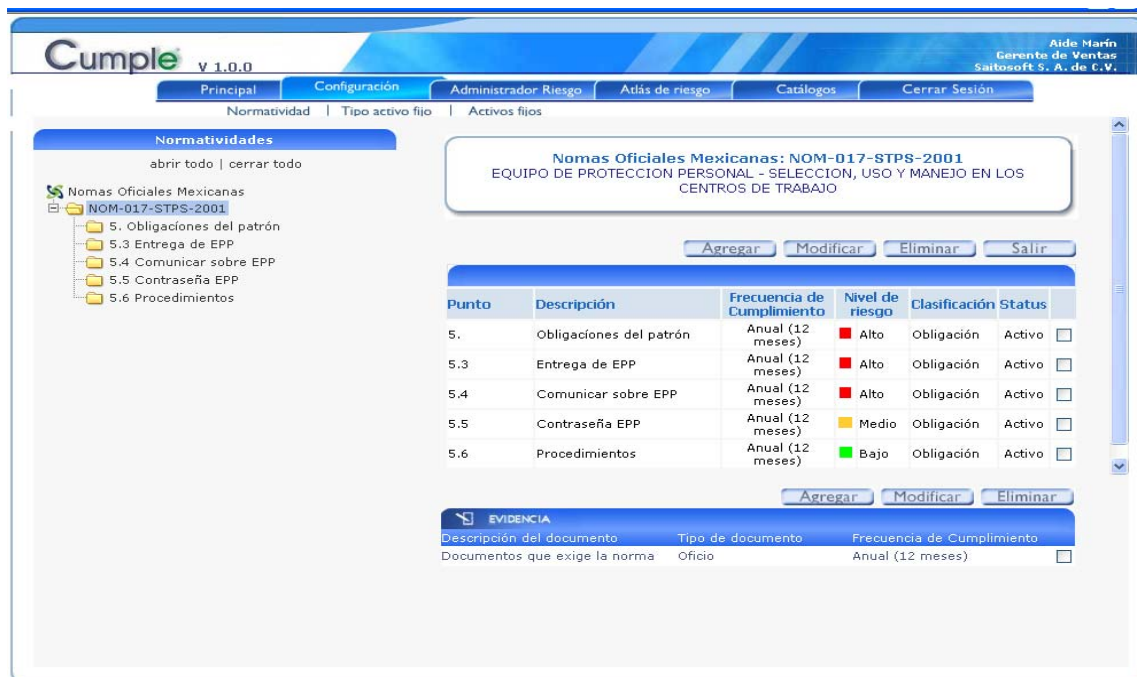


Fig. 4 - 6 Submenú NORMATIVIDAD del Sistema.

Cumple v 1.0.0

Principal | Configuración | Administrador Riesgo | Atlas de riesgo | Catálogos | Cerrar Sesión

Normatividad | Tipo activo fijo | Activos fijos

Tipos de Activo [Agregar] [Salir]

| Tipo | |
|------------------------|-------------------------------------|
| Almacén | <input checked="" type="checkbox"/> |
| Centro de Distribución | <input type="checkbox"/> |
| Edificio | <input type="checkbox"/> |
| Quirófano | <input type="checkbox"/> |

Seleccionar "Almacén"

NORMATIVIDADES APLICADAS AL ACTIVO: Almacén

POR NORMATIVIDAD: NOM-017-STPS-2001 [Eliminar]

| Punto | Descripción | Riesgo | <input type="checkbox"/> |
|-------|-------------------------|--------|--------------------------|
| 5.1 | Obligaciones del patrón | Alto | <input type="checkbox"/> |
| 5.3 | Entrega de EPP | Alto | <input type="checkbox"/> |
| 5.4 | Comunicar sobre EPP | Alto | <input type="checkbox"/> |
| 5.5 | Contraseña EPP | Medio | <input type="checkbox"/> |
| 5.6 | Procedimientos | Bajo | <input type="checkbox"/> |

Despliega las normativas asociadas

Fig. 4 - 7 Submenú TIPO ACTIVO FIJO del Sistema.

El submenú Activos Fijos:

Lista de manera detallada los Activos Fijos de la empresa especificada, permite asignar un responsable por default para cualquier disposición legal que se requiera cumplir. Esta sección tiene por objetivo agregar cuantos activos fijos tenga la empresa de manera particular para un control unitario, y llevar a cabo la asignación de un responsable para cumplir con las disposiciones que le corresponde a cada activo. Fig. 4 - 8.

Cumple v 1.0.0

Principal | Configuración | Administrador Riesgo | Atlas de riesgo | Catálogos | Cerrar Sesión

Normatividad | Tipo activo fijo | Activos fijos

ACTIVOS FIJOS

EMPRESA: Todos [v] [Buscar]

TIPO ACTIVO: Todos [v] **Filtros** Ordenar por: Activo [v]

Total: 0 activos [Agregar] [Modificar] [Eliminar] [Salir]

| Activo | Tipo | Responsable | Empresa |
|----------------------|------|-------------|---------|
| Activos Fijos | | | |

Agregar: Permite añadir activos fijos

Modificar: Permite realizar correcciones a los activos fijos

Eliminar: Suprime los activos sin responsables

Salir: Cerrar la sección

Fig. 4 - 8 Submenú ACTIVOS FIJOS del Sistema.

ADMINISTRADOR DE RIESGO

El tercer elemento que encontramos en el menú principal es ADMINISTRADOR RIESGO, que nos dirige a la sección que nos permite evaluar el nivel de cumplimiento de las disposiciones legales de la empresa Fig. 4 - 9. Como submenús, tenemos la opción de Administrador Riesgo.

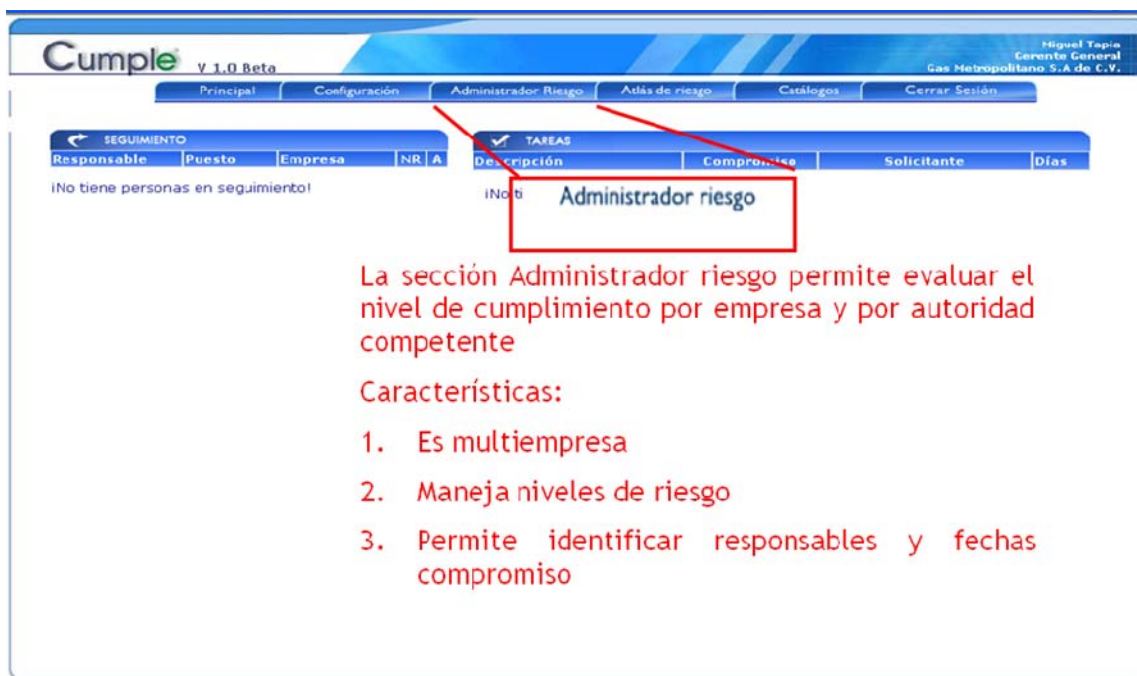


Fig. 4 - 9 Menú ADMINISTRADOR RIESGO del Sistema.

El submenú Administrador Riesgo:

Lista de manera detallada cada una de las empresas configuradas así como su nivel de cumplimiento (nivel de riesgo), ofrece una descripción ejecutiva de lado izquierdo por empresa, para ayudar en la toma de decisiones a manera de semáforo indicando con ROJO el riesgo mayor, con ÁMBAR un riesgo medio, y con VERDE que no se tiene ningún riesgo. En la parte derecha se muestra de manera general las cantidades de disposiciones legales No cumplidas, Atrasadas o No Asignadas a un responsable de cada nivel de Riesgo y por autoridad emisora Fig. 4 -10.

En cada autoridad emisora listada, se puede tener acceso al detalle de las disposiciones legales mostradas, así como el responsable asignado, el activo al que corresponde y todos los datos necesarios para el seguimiento puntual. Fig. 4 - 11



Fig. 4 - 10 Submenú ADMINISTRADOR RIESGO del Sistema.



Fig. 4 - 11 Submenú ADMINISTRADOR RIESGO del Sistema. Detalle por Autoridad Emisora.

En la fig. 4 – 11, se muestra el detalle por Autoridad Emisora de la disposición legal, para revisar las actividades pendientes se debe seleccionar el activo que se quiere revisar, con el botón Edit (Fig. 4 – 12).

El administrador de riesgo es de suma importancia para las empresas, debido a que de manera ágil se puede saber qué no se cumple, quién no cumple, qué nivel de riesgo se tiene por día. Y esto facilita de gran manera la visión jurídica de una empresa.

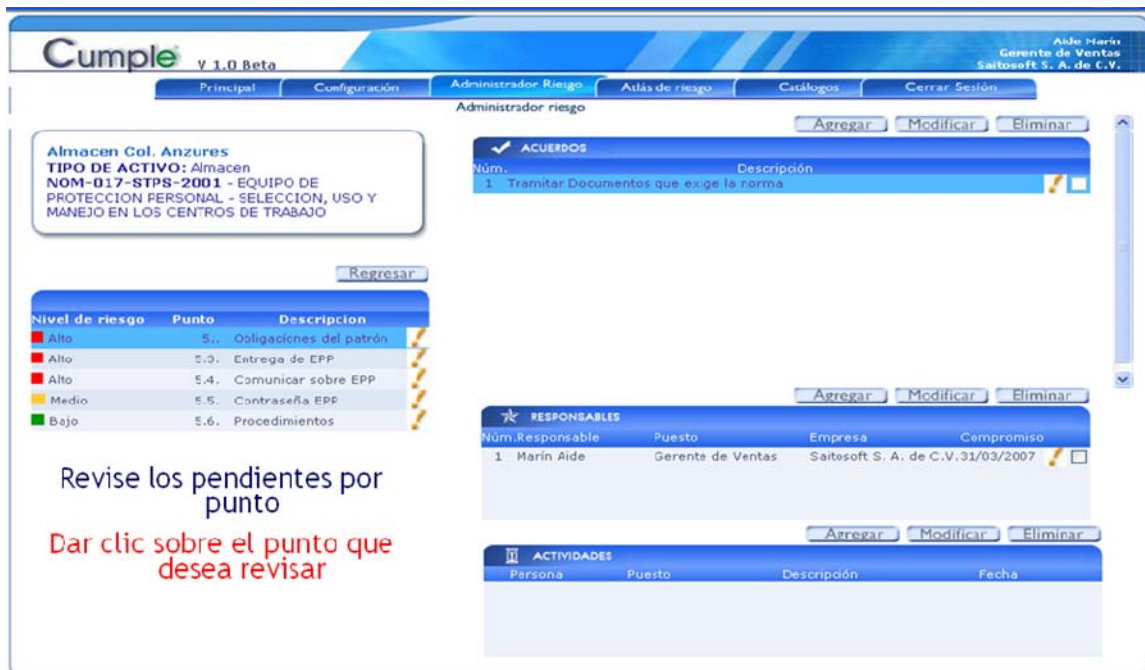
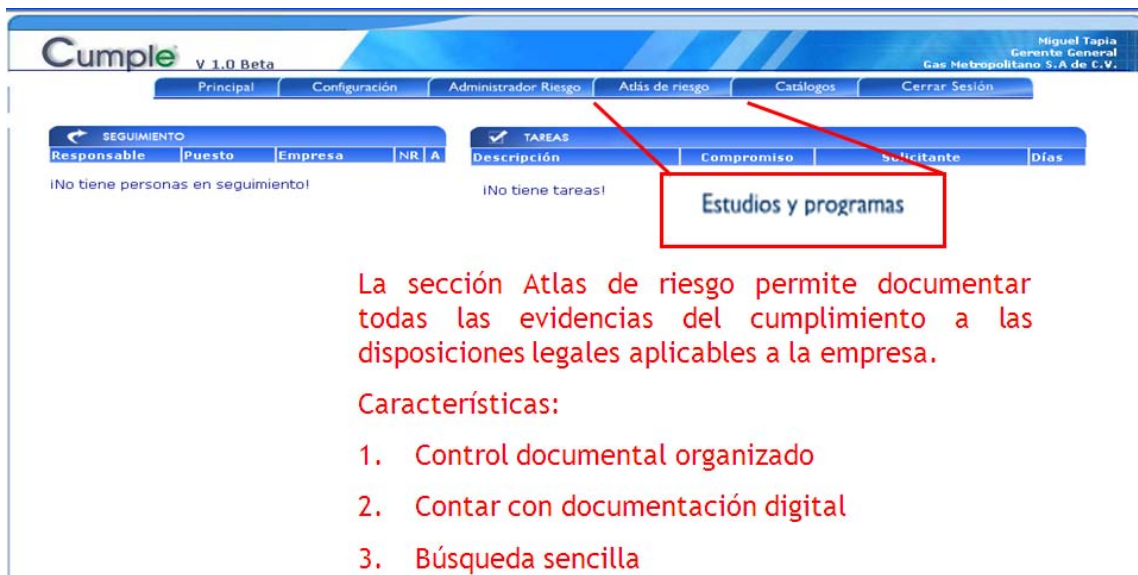


Fig. 4 - 12 Submenú ADMINISTRADOR RIESGO del Sistema. Detalle del Activo Fijo.

ATLAS DE RIESGO

El cuarto elemento que encontramos en el menú principal es ATLAS DE RIESGO, que nos dirige a la sección que nos permite revisar la documentación que se requiere para avalar el cumplimiento de cada disposición legal que debe cumplir la empresa de acuerdo a sus activos fijos. Como submenús, tenemos las opciones de Estudios y Programas. Fig. 4 – 13.



La sección Atlas de riesgo permite documentar todas las evidencias del cumplimiento a las disposiciones legales aplicables a la empresa.

Características:

1. Control documental organizado
2. Contar con documentación digital
3. Búsqueda sencilla

Fig. 4 - 13 Menú ATLAS DE RIESGO del Sistema.

El submenú Estudios y Programas:

Lista de manera detallada los documentos requeridos para avalar el cumplimiento de la disposición legal aplicada a un activo fijo de la empresa. En esta sección podemos adjuntar los documentos o revisarlos una vez adjuntos. Fig. 4-14.

ESTUDIOS Y PROGRAMAS

EMPRESA: Saitosoft S.A. de C.V. ACTIVO: Todos
 NORMATIVIDAD: Todos AUTORIDAD: Secretaría de Trabajo y Previs...
 Ordenar por: Estudios y Programas

Total : 5 estudios

| Empresa | Estudios y Programas | Autoridad Competente | Cumplimiento Legal | Activo | Período |
|-------------------------|--|--|--------------------|--------|----------------------------|
| Saitosoft S. A. de C.V. | Documentos que exige la norma | Secretaría de Trabajo y Previsión Social | NOM-017-STPS-2001 | 5. | Almacen Col. Anzures Anual |
| Saitosoft S. A. de C.V. | Manual EPP | Secretaría de Trabajo y Previsión Social | NOM-017-STPS-2001 | 5.4 | Almacen Col. Anzures Anual |
| Saitosoft S. A. de C.V. | Etiquetas para EPP | Secretaría de Trabajo y Previsión Social | NOM-017-STPS-2001 | 5.5 | Almacen Col. Anzures Anual |
| Saitosoft S. A. de C.V. | Procedimientos por escrito | Secretaría de Trabajo y Previsión Social | NOM-017-STPS-2001 | 5.6 | Almacen Col. Anzures Anual |
| Saitosoft S. A. de C.V. | Lista de recepción de EPP por los trabajadores | Secretaría de Trabajo y Previsión Social | NOM-017-STPS-2001 | 5.3 | Almacen Col. Anzures Anual |

Documentación electrónica en función a las evidencias

Fig. 4 - 14 Submenú ESTUDIOS Y PROGRAMAS del Sistema

CATÁLOGOS

El quinto elemento que encontramos en el menú principal es CATÁLOGOS. Como submenús, tenemos el catálogo de Contactos. Fig. 4 - 15.

CATÁLOGOS

SEGUIMIENTO TAREAS

Responsable Puesto Empresa NR A Descripción Compromiso Contactos Días

¡No tiene personas en seguimiento!

¡No tiene tareas!

La sección Catálogos permite ingresar Contactos al sistema, con a finalidad de que sean nombrados responsables de un trámite u obligación.

Fig. 4 - 15 Menú CONTACTOS del Sistema

Funcionalidades importantes

DELEGAR TAREAS

En la pantalla de tareas personales, se cuenta con la funcionalidad de delegar esa tarea (Fig. 4 - 16) a una o varias personas (Fig. 4 - 17), siempre y cuando la nueva persona encargada no la tenga previamente asignada.

Cumple v 1.0.0

Principal Configuración Administrador Riesgo Atlas de riesgo Catálogos Cerrar Sesión

Tareas | Seguimiento

TAREAS de Aide Marín

SOLICITANTE: Marin Aide RESPONSABLE: Marin Aide ESTATUS: No Resuelto ACTIVIDAD: Todas CLAVE: Todos ACTIVO: Todos TIPO ACTIVO: Todos EMPRESA: Todos

08/11/2006 AL 08/11/2006 Ordenar por: Activo Fijo

Total : 2 acuerdos

| Solicitante | Tarea | Activo Fijo | Normatividad | CompromisoDías | Delegar | Act |
|--|--|---|---|---------------------------|---------|-----|
| Marin Aide Gerente de Ventas Saitosoft S. A. de C.V. | Tramitar Documentos que exige la norma | Almacen Planta baja Almacen Saitosoft S. A. de C.V. | - NOM-017-STPS-2001 - EQUIPO DE PROTECCION PERSONAL - SELECCION, USO Y MANEJO EN LOS CENTROS DE TRABAJO | 30 noviembre 2006, jueves | Delegar | Act |
| Marin Aide Gerente de Ventas Saitosoft S. A. de C.V. | Tramitar Etiquetas para EPP | Almacen Planta baja Almacen Saitosoft S. A. de C.V. | - NOM-017-STPS-2001 - EQUIPO DE PROTECCION PERSONAL - SELECCION, USO Y MANEJO EN LOS CENTROS DE TRABAJO | 30 noviembre 2006, jueves | Delegar | Act |

Para delegar una tarea:
1. Dar clic en la opción “Delegar”

Fig. 4 - 16 Delegar tareas.

Cumple v 1.0.0

Principal Configuración Administrador Riesgo Atlas de riesgo Catálogos Cerrar Sesión

Tareas | Seguimiento

Tarea que se va a delegar

DELEGADOS

Tarea : Tramitar Etiquetas para EPP
Solicitante : Aide Marín
Compromiso : 30/11/2006
Instrucción :

PERSONAS DISPONIBLES

Buscar: Ir
Ordenar Por: Apellido Paterno Todos
Empresa: Saitosoft S. A. de C.V.

Nuevo Contacto

| Nombre | Puesto | Empresa | Seleccionar |
|----------------|-------------------|-------------------------|-------------------------------------|
| Lopez Gabriela | Lider de Proyecto | Saitosoft S. A. de C.V. | <input checked="" type="checkbox"/> |

Fecha Compromiso Instrucción: 23/11/2006

Generar la orden de compra y adquirir

Agregar

1. Seleccionar el nuevo responsable
2. Asignar una fecha compromiso
Opcional: Instrucción
3. Dar clic en la opción “Agregar”.

Fig. 4 - 17 Delegar tarea a una o varias personas

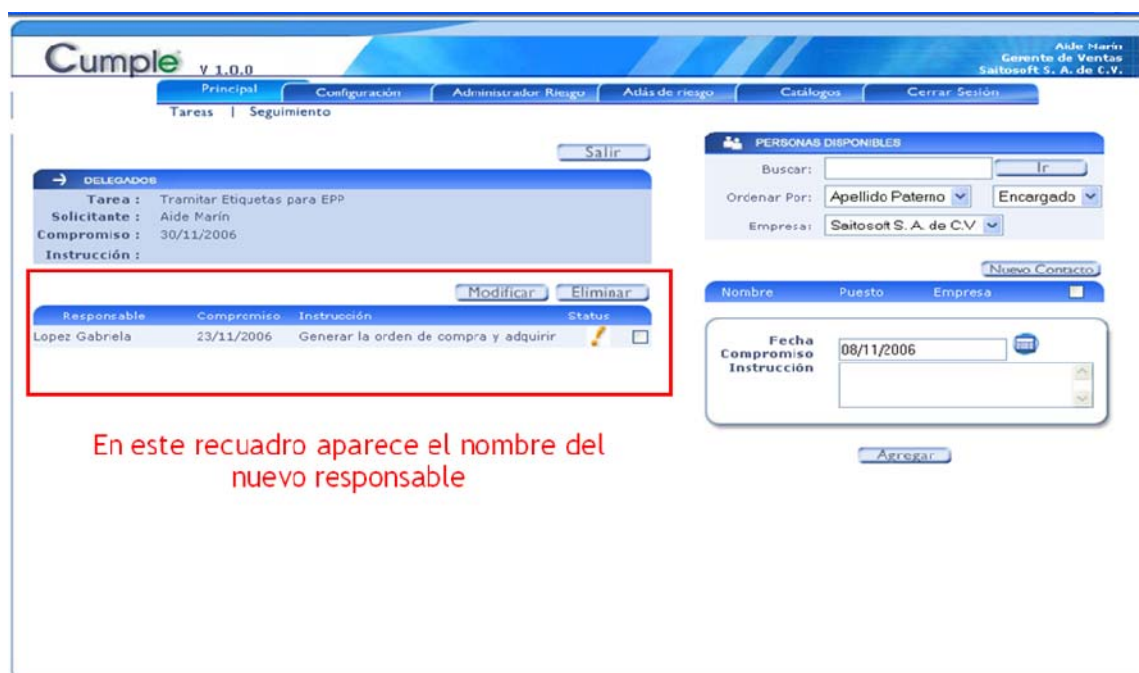


Fig. 4 - 18 Delegar Tarea.

ADJUNTAR DOCUMENTOS

La funcionalidad de adjuntar documentos se puede realizar desde varios lugares del sistema. (Fig. 4 - 19 a 23)

1. Al configurar una Normatividad se tiene la opción de capturar documentos anexos a la normatividad, con el fin de adjuntar el documento original de donde proviene la Normatividad.
2. Al configurar una evidencia se tiene la opción de capturar uno o varios documentos por evidencia con el fin de indicar que esa evidencia requiere un documento especial para ser válido.
3. En el área de estudios y programas, se puede capturar uno o varios documentos con el fin de indicar las evidencias del cumplimiento de una disposición legal.
4. Al capturar una actividad, se tiene la opción de capturar uno o varios documentos, con el fin de indicar que se realizó alguna actividad que produjo un documento físico.

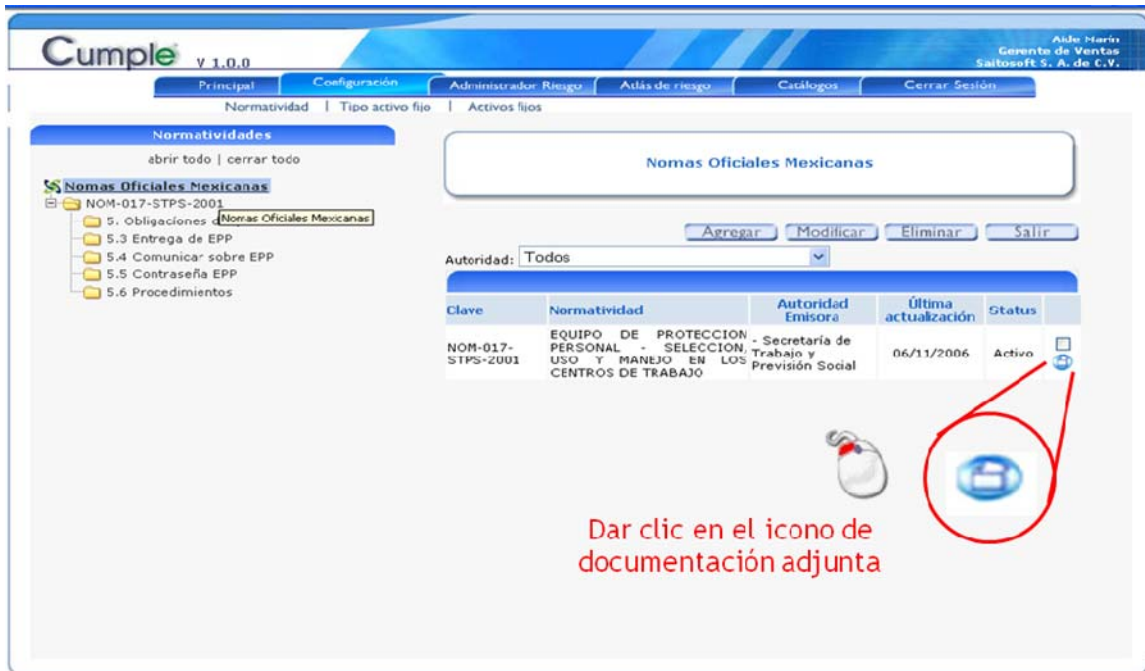


Fig. 4 - 20 Adjuntar documento. Paso 1.

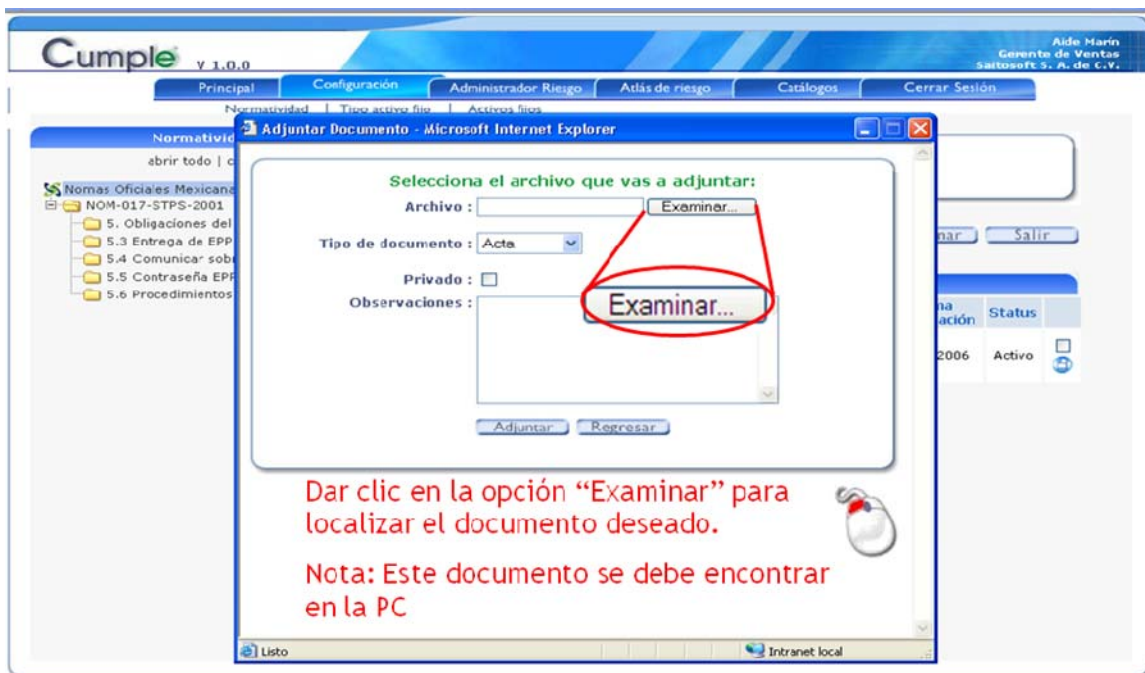


Fig. 4 - 21 Adjuntar documento. Paso 2.

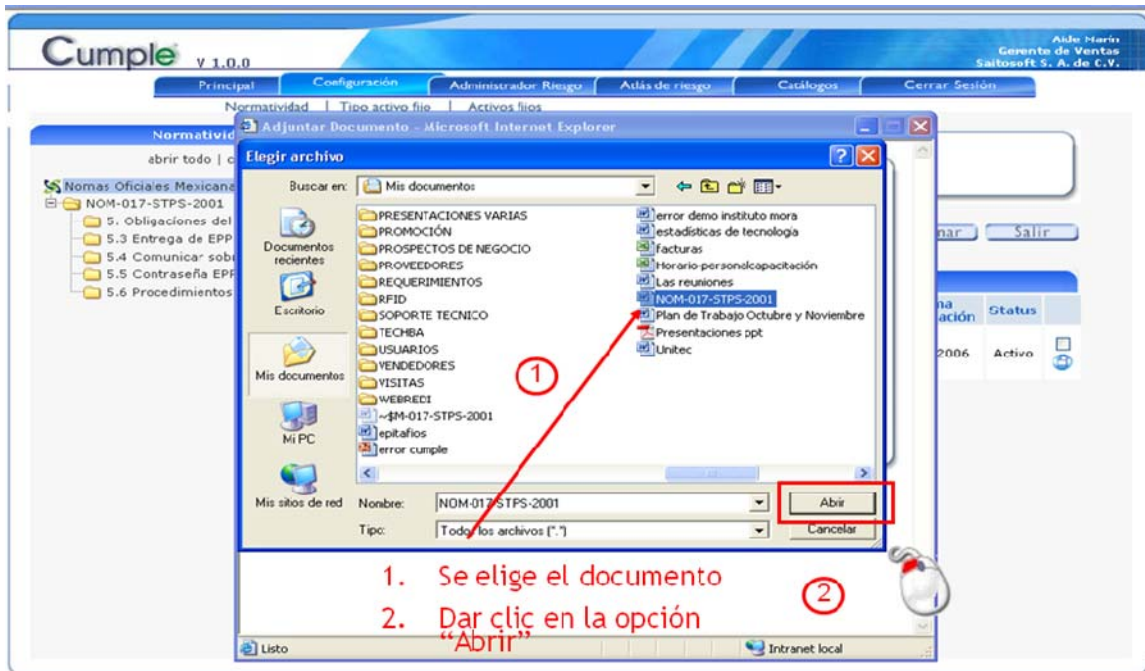


Fig. 4 - 22 Adjuntar documento. Paso 3.

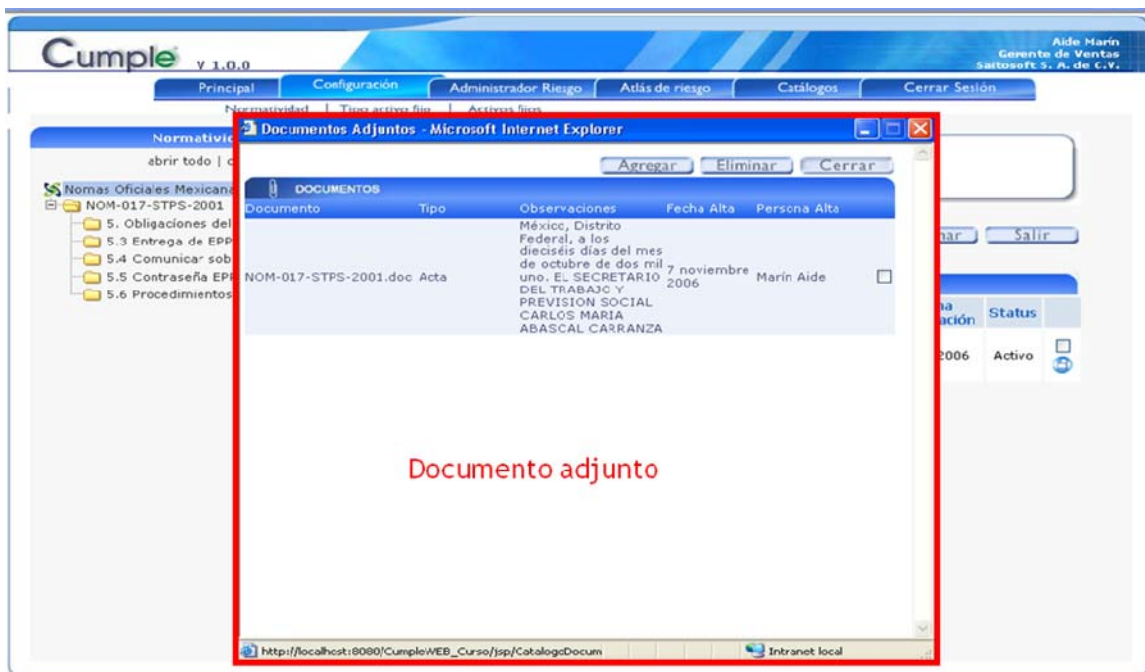


Fig. 4 - 23 Adjuntar documento. Paso 4.

CAPTURAR ACTIVIDADES

En la sección de Tareas se encuentra la opción de capturar actividades. Para cada una de las tareas pendientes del usuario en sesión, se tiene la posibilidad de capturar avances o comentarios acerca del término de la tarea. Esto con el fin de mantener la comunicación entre el solicitante y el responsable de la tarea. Al solicitante en la sección de Seguimiento le aparecerá la actividad o actividades que el responsable haya capturado. Fig. 4 – 24 a 26.

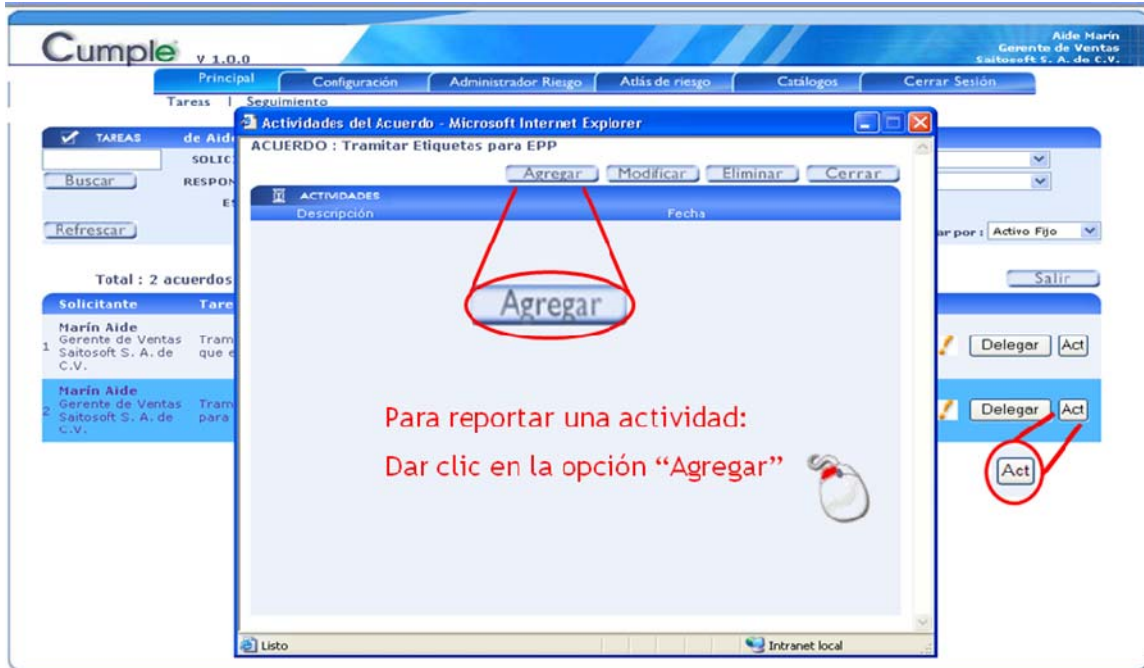


Fig. 4 - 24 Capturar Actividades. (Paso 1)

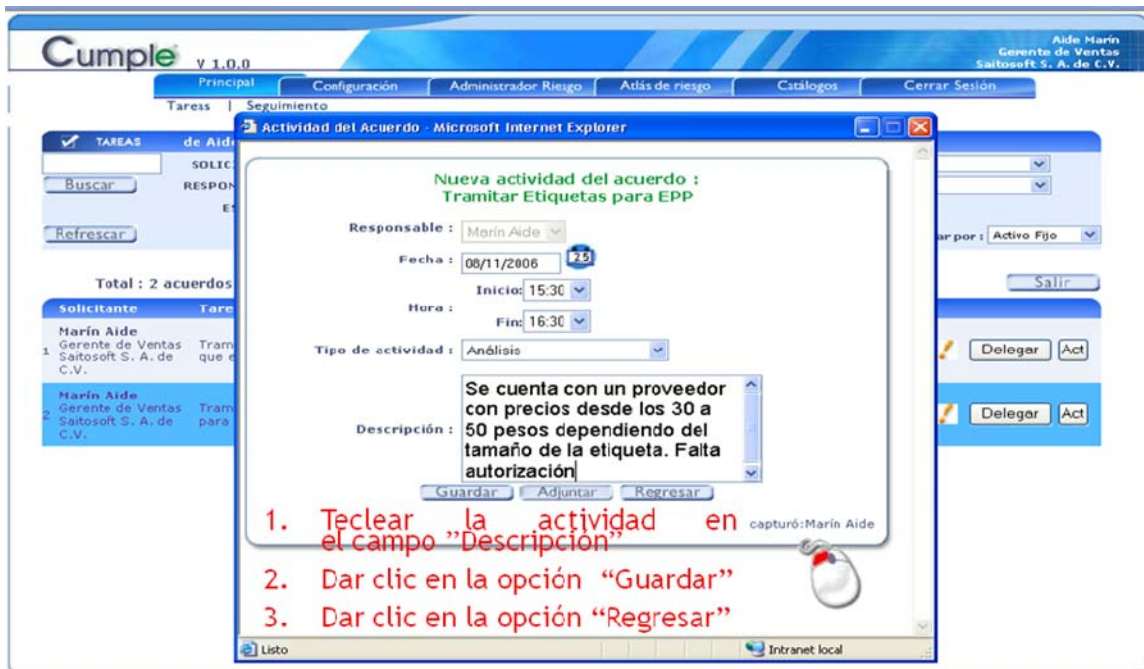
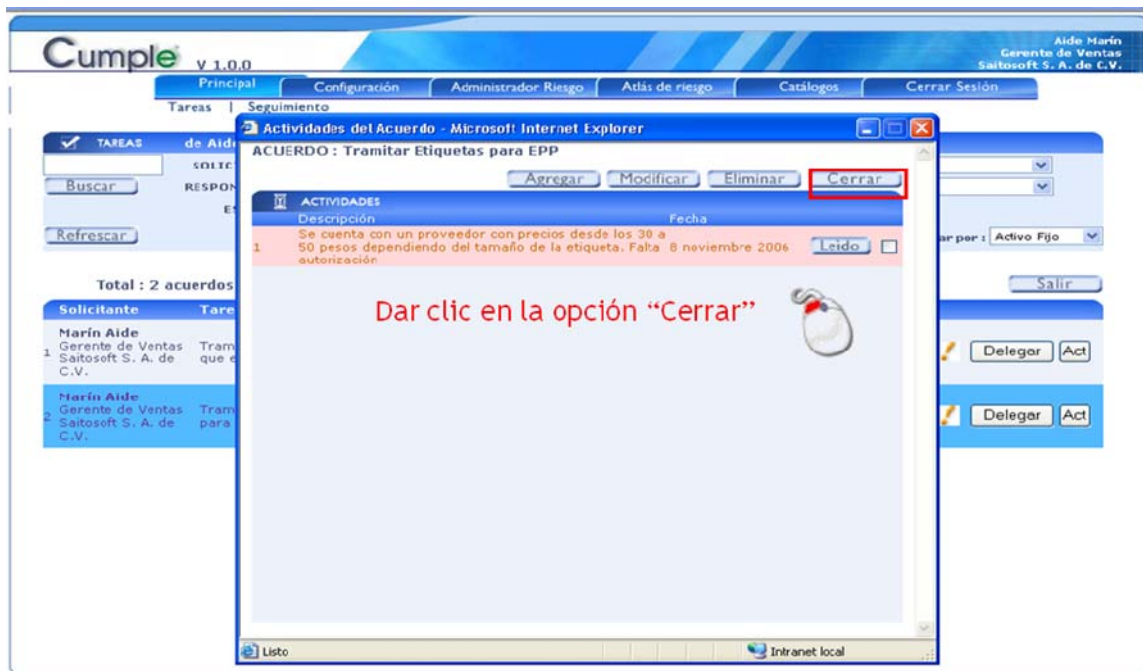


Fig. 4 - 25 Capturar Actividades (Paso 2)



TAREAS LIBRES

Las tareas libres o No Conformidades, tienen por objetivo asignar una tarea de manera directa a un responsable, sin embargo debe cumplir con la relación entre el activo fijo de la empresa y la normatividad asociada.

Esta funcionalidad se utiliza mucho cuando se realizan certificaciones o auditorías, para indicar que se reportaron No conformidades en el ejercicio de la auditoría y se requiere el rápido y puntual cumplimiento de la disposición legal indicada. (Fig. 4 – 27 a 29)

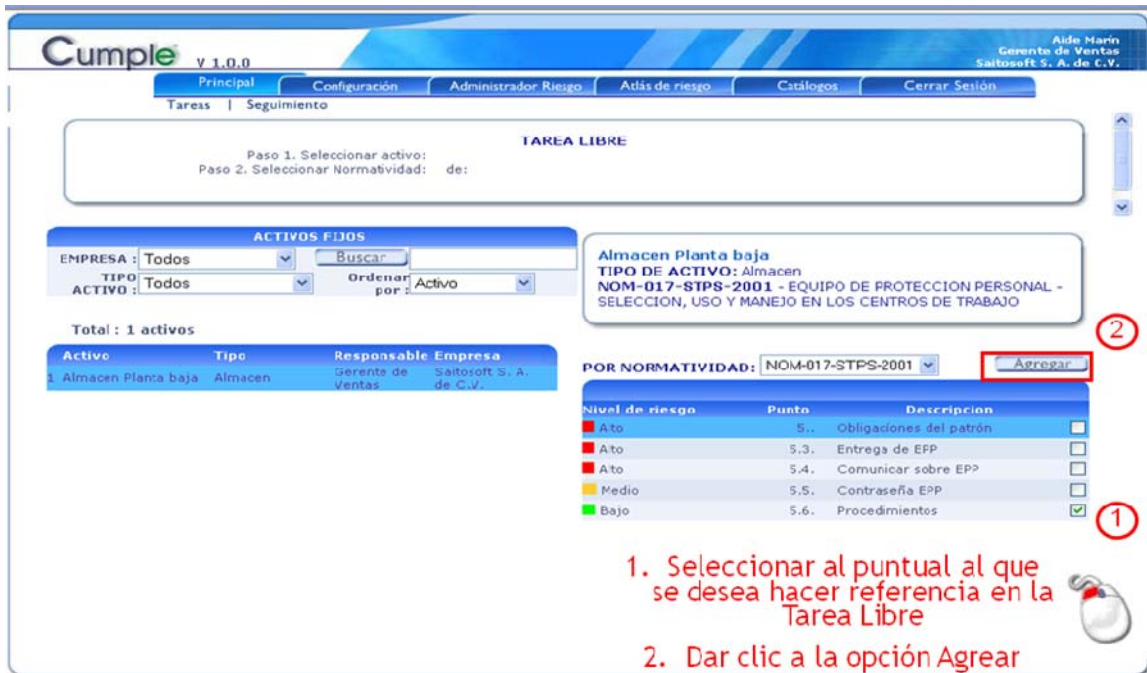


Fig. 4 - 27 Tarea Libre (Paso 1)

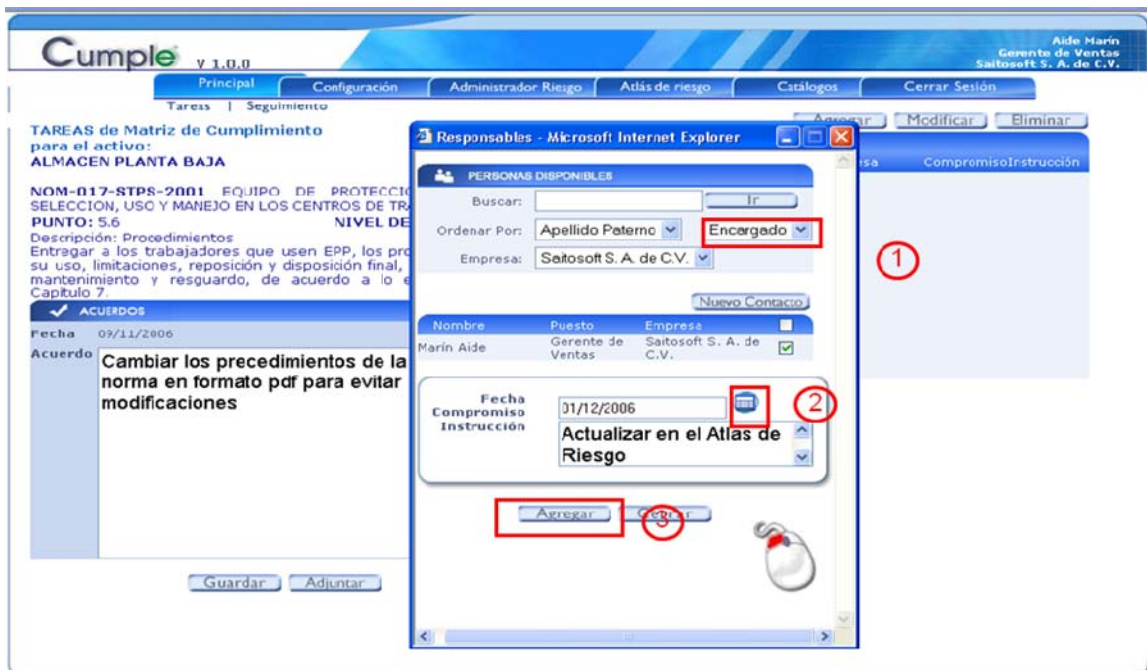


Fig. 4 - 28 Tarea Libre (Paso 2)

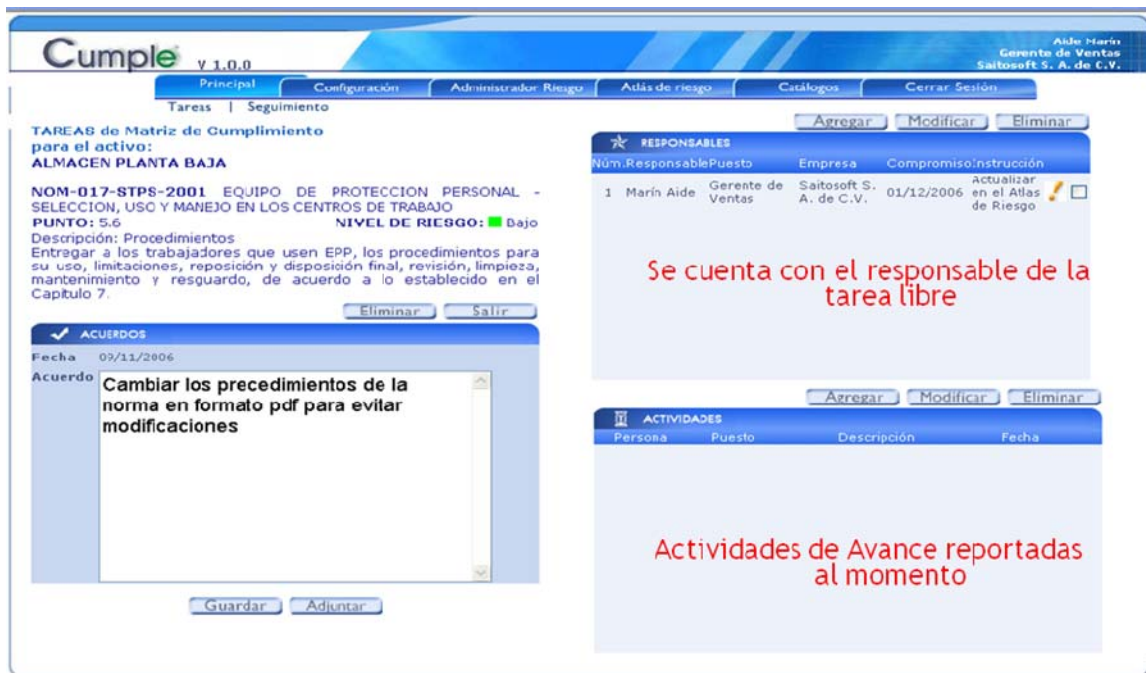


Fig. 4 - 29 Tarea Libre (Paso 3)

RESOLVER TAREAS

La solución de Tareas es una parte muy importante del sistema, debido a que marca la solución correcta de una disposición legal. Solo puede dar por concluida una disposición legal el solicitante de la tarea que en la mayoría de los casos será el monitor de la empresa seleccionada. (Fig. 4 - 30 a 32)



Fig. 4 - 30 Resolver tareas (Paso 1)

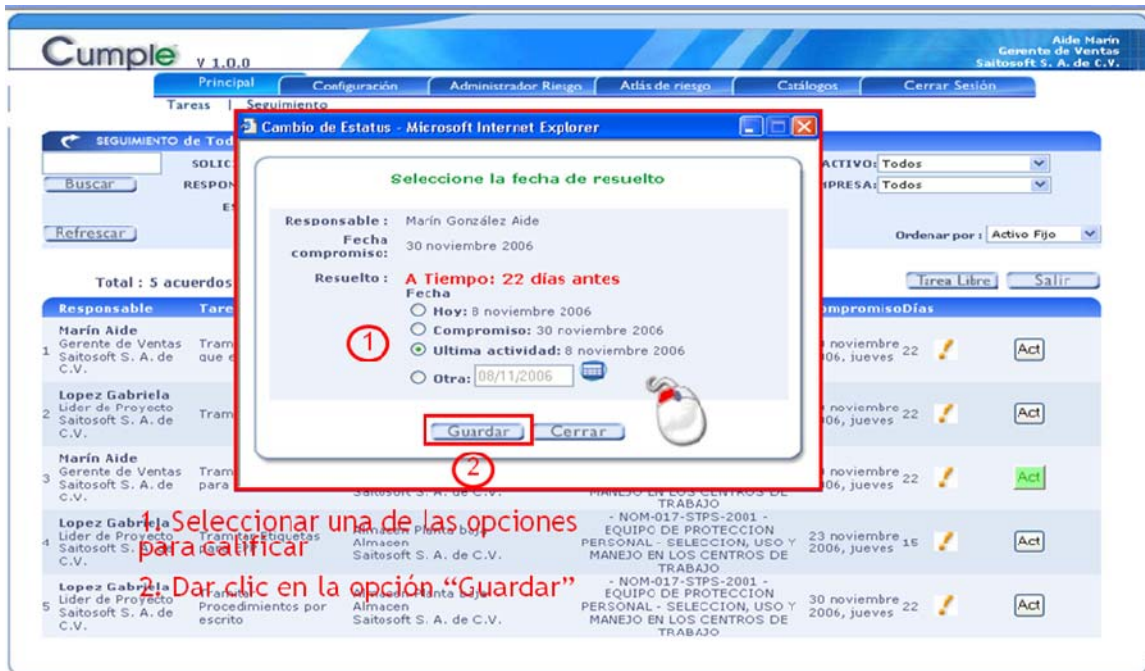


Fig. 4 - 31 Resolver tareas (Paso 2)



Fig. 4 - 32 Resolver tareas (Paso 3)

Área de Administración

MENÚ: CATÁLOGOS

En esta sección el administrador del sistema lleva a cabo la administración de los catálogos de Autoridades Emisoras, Clasificaciones de Normatividad, Periodos de cumplimiento, Tipos de documentos, Tipos de Normatividad, Motivos de cancelación, Tipos de Actividad, Giros de Empresas, Puestos, Tipos de Activo (clasificación de Activos Fijos,) y Empresas. Todos estos datos son utilizados en el área del Usuario. Cuando falte algún concepto el usuario está obligado a pedirle al administrador del Sistema Regulatorio que lo capture para su futura selección.

MENÚ: ADMINISTRADOR

Se listan los datos del Administrador.

MENÚ: USUARIOS

Se listan y capturan las personas que tienen acceso al sistema.

MENÚ: PERFILES

Se listan y capturan los perfiles definidos por los privilegios de acceso a cada área del sistema.

4.4 Codificación

La codificación de todo el sistema se lleva a cabo de manera similar al fragmento de código siguiente, donde presento la implementación para el catálogo de giros de empresas del sistema.

Las clases involucradas son: GiroEmpresa.java (bean), GiroEmpresa.java (Maneja), CatalogoGiroEmpresa.jsp (Catálogo para la vista), CapturaGiroEmpresa.jsp (Formulario de captura), Guarda GiroEmpresa.jsp (para guardar cambios)

GiroEmpresa.java

```
package beans;

public class GiroEmpresa
{
    private int giroEmpresa;
    private java.lang.String descripcion;
    private java.lang.String status;

    public int getGiroEmpresa()
    {
        return giroEmpresa;
    }

    /** @param newGiroEmpresa */
    public void setGiroEmpresa(int newGiroEmpresa)
    {
        giroEmpresa = newGiroEmpresa;
    }
}
```

```

public java.lang.String getDescripcion()
{
    return descripcion;
}

/** @param newDescripcion */
public void setDescripcion(java.lang.String newDescripcion)
{
    descripcion = newDescripcion;
}

public java.lang.String getStatus()
{
    return status;
}

/** @param newStatus */
public void setStatus(java.lang.String newStatus)
{
    status = newStatus;
}
}

```

ManejaGiroEmpresa.java

```

package maneja;

import BD.ConexionBD;
import BD.ParametrosConexion;
import java.util.Hashtable;
import alxs.utils.BD.ManejaBean;
import beans.GiroEmpresa;

public class ManejaGiroEmpresa extends ManejaBean
{
    private static String tabla="GiroEmpresa";
    private static Hashtable tablaMapeo=new Hashtable();
    static{
        tablaMapeo.put("giroEmpresa","giroEmpresa");
        tablaMapeo.put("descripcion","descripcion");
        tablaMapeo.put("status","status");
    }

    public static GiroEmpresa trae(ParametrosConexion param, String condiciones) throws Exception
    {
        GiroEmpresa giroempresa= new GiroEmpresa();
        return
(GiroEmpresa)trae(giroempresa,tablaMapeo,tabla,condiciones,ConexionBD.getConnection(param));
    }

    public static int inserta(ParametrosConexion param, GiroEmpresa giroempresa)throws Exception
    {
        Hashtable tmp= (Hashtable)tablaMapeo.clone();
        tmp.remove("giroEmpresa");
        return inserta(giroempresa,tabla, tmp, ConexionBD.getConnection(param));
    }
}

```



```

    }

    public static GiroEmpresa[] busca(ParametrosConexion param, String condiciones) throws Exception
    {
        return (GiroEmpresa[]) busca(GiroEmpresa.class , tablaMapeo, tabla,
condiciones,ConexionBD.getConnection(param));
    }

    public static int actualiza(ParametrosConexion param,GiroEmpresa giroempresa,String
condiciones)throws Exception
    {
        Hashtable tmp= (Hashtable)tablaMapeo.clone();
        tmp.remove("giroEmpresa");
        return actualiza(giroempresa, tmp, tabla, condiciones,ConexionBD.getConnection(param) );
    }

    public static int elimina(ParametrosConexion param, String condiciones) throws Exception
    {
        return elimina( tabla, condiciones,ConexionBD.getConnection(param));
    }
}

```

CatalogoGiroEmpresa.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="beans.Persona,
beans.Status, maneja.ManejaStatus, beans.GiroEmpresa, maneja.ManejaGiroEmpresa, java.util.Hashtable,
java.util.Enumeration" %>

<%
String clave="CTGECN";
String repetido=request.getParameter("repetido");
Persona persona= (Persona) session.getAttribute("atributosPersona");
%>
<html>
<head>
<link href="../css/CumpleWEB.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="../js/ChecarC.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>CatalogoGiroEmpresa</title>

<script language="javascript" type="text/javascript">
function modificarN(giremp)
{
    document.location.href="../CapturaGiroEmpresa.jsp?giremp="+giremp
}
...

function eliminar()
{
    var cadena="";
    var ge=arregloRegistros.length;
    for(i=0;i<this.arregloRegistros.length;i++)
        cadena+="\n\t"+arregloRegistros[i];

    if (confirm("¿Desea eliminar los siguientes Registros " + cadena + " ?"))
    {
        document.location.href="/GuardaGiroEmpresa.jsp?elimina=Si&ge="+ge+"&aR="+arregloRegistros
    }
}
...
</script>
<body
onLoad="mensaje('<%=repetido%>');MM_preloadImages('../images/agregar2.jpg','../images/modificar2.jpg
','../images/eliminar2.jpg','../images/regresar2.jpg')">

<%if (persona==null)

```

```

{
    out.println("No se ha logeado al sistema");
}
else
{%>
    <%@ include file="Claves.jsp" %>
    <%if (temp)
    {
        GiroEmpresa[] catalogo=ManejaGiroEmpresa.busca(param, " ORDER BY Descripcion");
        Status status;%>
<div id="Layer2" style="position:absolute; left:10%; top:2%; width:80%; height:100%; z-index:2">
    <form action="" method="post" name="checkbox_form" target="mainFrame">
    <table width="100%" border="0" cellspacing="0">
        <tr bgcolor="#FFFFFF">
            <td width="28%"><span class="Estilo1">Giro de empresas </span></td>
            <td width="72%"><div align="right">

                <!--Botones-->

                <a href="#" onMouseOut="MM_swapImgRestore()"
                onMouseOver="MM_swapImage('btnAgregar','', '../images/agregar2.jpg',1)"></a>

                <a href="#" onMouseOut="MM_swapImgRestore()"
                onMouseOver="MM_swapImage('btnModificar','', '../images/modificar2.jpg',1)"></a>

                <a href="#" onMouseOut="MM_swapImgRestore()"
                onMouseOver="MM_swapImage('btnEliminar','', '../images/eliminar2.jpg',1)"></a>

                ...

            </div></td>
        </tr>
    </table>

    <table width="100%" cellpadding="0" cellspacing="0">
        <tr class="encabezado">
            <td height="20" colspan="2" background="../images/fondo_tit.jpg"></td>
            <td background="../images/fondo_tit.jpg"><div align="right"></div></td>
        </tr>
        <tr class="encabezado">
            <td width="73%" height="16" background="../images/fondo_tit2.jpg">Giro Empresa</td>
            <td width="24%" background="../images/fondo_tit2.jpg"> Status </td>
            <td width="3%" background="../images/fondo_tit2.jpg">&nbsp;</td>
        </tr>
    <% for (int i=0; i< catalogo.length ;i++)
    {%>
        <tr class="normal" style='cursor:pointer' onMouseOver=this.className='seleccion2'
        onDbClick=modificarN("<%= catalogo[i].getGiroEmpresa()>")
        onMouseOut=this.className='normal'>
        <td ><%= catalogo[i].getDescripcion()></td>
        <td><div align="center">
            <% status=ManejaStatus.trae(param, " WHERE Status='"+catalogo[i].getStatus()+"'"); %>
            <%= status.getDescripcion() %></div>
        </td>
        <td align="center">
            <input id="<%=catalogo[i].getDescripcion()%" name="checkbox" type="checkbox"
            value="<%=catalogo[i].getGiroEmpresa()%"></td>
        </tr>
    <%}>
</table>

```

```

    </form>
</div>
</p>
<div align="center">
<%temp=false;
}&
else
{%>
                No tiene acceso a esta funcionalidad
<input name="Regresar" type="button" id="Regresar" onClick="history.back()" value="Regresar">
<%}
}%>
</div>
</body>
</html>

```

CapturaGiroEmpresa.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="beans.Persona, beans.GiroEmpresa, maneja.ManejaGiroEmpresa, beans.Status,
maneja.ManejaStatus" %>

<%
Persona persona= (Persona) session.getAttribute("atributosPersona");
String ge=request.getParameter("giremp");

. . .

GiroEmpresa modificar=ManejaGiroEmpresa.trae(param," WHERE GiroEmpresa="+ge+"");
Status[] statuss=ManejaStatus.busca(param,"WHERE Categoria='Catalogo' ORDER BY Descripcion");
%>

<form action="GuardaGiroEmpresa.jsp" method="post" name="datos" id="datos">
. . .
<%
if (ge==null)
{%>
        Nuevo Giro de Empresa
<%}else{%>
        Modificar Giro Empresa:  <%= modificar.getDescripcion() %>
<%}%>
. . .
<strong>Giro de empresa:</strong>
<input name="descripcion" type="text" id="descripcion"
        <% if (ge!= null) {%>value="<%= modificar.getDescripcion() %>"<% } %>
        size="50" maxlength="250">
. . .
<strong>Status:</strong>
<% if (ge!= null)
{%>
        <select name="status" id="status">
        <% for (int i=0; i< statuss.length ;i++)
        {%>
                <option value='<%=statuss[i].getStatus().trim()%>'
                <% if(ge!=null){
                if (statuss[i].getStatus().equals(modificar.getStatus())){%>selected <%}}%>
                ><%= statuss[i].getDescripcion() %> </option>
        <%}%>
        </select>
<% }else{%>
        <strong>Activo</strong>
        <input type="hidden" name="status" value='AC'>
<%} %>
. . .

<a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('btnGuardar','','../images/guardar2.jpg',1)"></a>

```

...

GuardaGiroEmpresa.jsp

```

<%@page import="maneja.ManejaGiroEmpresa, beans.Persona, beans.GiroEmpresa"%>
<jsp:useBean id="giroempresa" class="beans.GiroEmpresa">
<jsp:setProperty name="giroempresa" property="*/>
</jsp:useBean>

<%
Persona persona= (Persona) session.getAttribute("atributosPersona");
String ge=request.getParameter("ge");
String pagina="CatalogoGiroEmpresa.jsp";
String clave="";
String elimina=request.getParameter("elimina");
...

if (ge!=null)
{
    if (elimina.equals("Si"))
    {
        String aR=request.getParameter("aR");
        String aRtemp1="";
        if (aR.indexOf(",")==-1){
            ManejaGiroEmpresa.elimina(param," WHERE Descripcion='"+aR.trim()+"'");
        }else{
            int inicio_cad=0;
            int cont=Integer.parseInt(ge);
            aR=aR+",";
            for (int i=0; i < cont ;i++){
                inicio_cad=aR.indexOf(",");
                aRtemp1=aR.substring(0, inicio_cad);
                ManejaGiroEmpresa.elimina(param," WHERE Descripcion='"+aRtemp1.trim()+"'");
                aR=aR.substring(inicio_cad+1,aR.length());
            }
        }
    }
    else
    {
        ...
        ManejaGiroEmpresa.actualiza(param,giroempresa," WHERE GiroEmpresa="+ge);
        ...
    }
    else
    {
        ...
        ManejaGiroEmpresa.inserta(param,giroempresa);
        ...
    }
}
...

```

Capítulo 5

Transición

Etapa de transición

Esta etapa consiste en transferir el producto a sus usuarios. Este capítulo describo los productos de trabajo generados para la entrega del sistema.

5.1 Características Generales del Sistema

Sistema de Cumplimiento Regulatorio

Cumple v1.0

- Lenguaje de desarrollo: Java
- Plataforma: J2EE (Java 2 Enterprise Edition)
- Componentes: SERVLETS, JPS, JAVA BEANS.
- Arquitectura: MVC (Modelo - Vista - Controlador).

5.2 Requisitos de Instalación

Los siguientes requerimientos técnicos están listados como fueron requeridos en la instalación del 31 de Julio del 2006, para posibles actualizaciones de software se requiere previas pruebas para asegurar el funcionamiento correcto del Sistema.

Requerimientos de software del servidor

- ✓ Sistema Operativo Windows 2000 Server SP4, Windows 2003
- ✓ Apache Tomcat v 4.1.31
- ✓ J2sdk v 1.4.1_01 (Sun Microsystems)
- ✓ Microsoft SQLServer 2000 Español SP3

Requerimientos de hardware del servidor

- ✓ Procesador Pentium 4, equivalente o superior.
- ✓ Memoria RAM mínima requerida: 512 MB
- ✓ Espacio en disco duro: 80 MB
 - Apache Tomcat 50 MB
 - J2sdk 10 MB
 - Aplicación WEB 10 MB
 - Base de Datos 5 MB (Inicial)

Requerimientos para equipos cliente

- ✓ Navegador Microsoft Internet Explorer v.6 o superior.
- ✓ Macromedia Flash Player 7 o superior.
- ✓ Opciones de bloqueo de menús emergentes desactivada solo para el sitio de la aplicación.
- ✓ Permitir ejecutar lenguaje Javascript.
- ✓ Puerto 8080 habilitado en Firewall.
- ✓ En caso de tener SQL Server 2000 habilitar puerto 1433 en Firewall.
- ✓ Configuración de pantalla: deseable 1024x768 p.
- ✓ Adobe Reader 6.0 o superior.

5.3 Instalación y Configuración

Para llevar a cabo la instalación del Sistema de Cumplimiento Regulatorio es recomendable cerrar todas las aplicaciones y seguir los siguientes pasos:

| | |
|--------|---|
| Paso 1 | Instalar Microsoft SQLServer 2000 |
| Paso 2 | <p>Ponerle el service pack 3 para Microsoft SQLServer 2000: esn_sql2ksp3</p> <p>A) Desempaquetar esn_sql2ksp3 B) Ejecutar el SETUP.exe C) Reiniciar.</p> |
| Paso 3 | Ejecutar j2sdk_1_4_1_01-windows-i586.exe. |
| Paso 4 | <p>Crear variable de Sistema: Nombre: CLASSPATH Valor: . (Punto)</p> <p>Para hacer esto: Seleccionar Propiedades con el botón derecho sobre Mi PC, entrar a opciones avanzadas, clic en variables de entorno y crear nueva variable de sistema.</p> |
| Paso 5 | <p>Ejecutar: jakarta-tomcat-4.1.31.exe</p> <p>Al ejecutarlo, aparecerá una ventana reconociendo el jdk instalado.</p> |
| Paso 6 | <p>Instalar Apache Tomcat con: Usuario: admin Password: admin.</p> <p>Y como NTService. (para hacer esto: seleccionar la casilla que dice NTService, dejar las otras opciones como están; esto quiere decir que cuando se levante el sistema operativo también se levantará el servicio de tomcat y ya no lo tendremos que hacer manualmente)</p> |
| Paso 7 | <p>Crear: Variable de Sistema: JAVA_HOME=C:\j2sdk1.4.1_01 (o la ruta donde se instaló el paso 3)</p> |
| Paso 8 | <p>Copiar los archivos proporcionados: alxs.jar, itextl.3.1.jar y jtids-0.9.jar en:</p> |

| | |
|---------|---|
| | C:\Archivos de programa\Apache Group\Tomcat 4.1\common\lib |
| Paso 9 | <p>Detener el Servicio de tomcat.</p> <p>Para hacer esto: Entrar a Panel de Control → Herramientas Administrativas → Servicios y ahí buscar en la lista el Servicio de Apache Tomcat 4.1, seleccionarlo y clic en DETENER.</p> |
| Paso 10 | <p>Copiar Carpeta CumpleWEB en:</p> <p>C:\Archivos de programa\Apache Group\Tomcat 4.1\webapps</p> |
| Paso 11 | <p>Nota :</p> <p>Para crear la base de datos se tienen 2 opciones, Restaurar la base de datos previamente creada que contiene los catálogos precargados o crear la base de datos a mano y cargar después los catálogos. Para la guía de instalación se cuenta con la base de datos creada con anticipación por cuestiones de negocio de la empresa que brinda el Sistema.</p> <p>Restaurar base de datos: Crear la base de datos CumpleWEB a mano. Seleccionarla y dar clic derecho, en todas las tareas, restaurar base de datos. Desde dispositivos, agregar y buscar en C el archivo de respaldo que dice: CumpleWEB. Aceptar 2 veces, y en la otra pestaña "Opciones", seleccionar la casilla de - Forzar la restauración a la base de datos existente-.</p> |
| Paso 12 | <p>Entrar a la base de datos, seleccionar usuarios, borrar CumpleWEB y administradorc.</p> <p>Ir al apartado de Seguridad, con botón derecho inicios de sesión: y crear un nuevo inicio:</p> <p>Nombre: Administradorc Seleccionar autenticación de SQLServer Password: admin.</p> <p>Base de datos: CumpleWEB Idioma: spanish</p> <p>Pestaña Funciones de Servidor: Seleccionar primera casilla.</p> <p>Pestaña: Acceso a Base de de datos: Seleccionar CumpleWEB y abajo, seleccionar db_ower</p> <p>Aceptar y poner de nuevo la contraseña: admin..</p> |
| Paso 13 | Crear otro inicio de sesión: |

| | |
|---------|---|
| | Nombre: CumpleWEB Seleccionar autenticación de SQLServer Password: 50elpmuc02 |
| Paso 14 | Iniciar el Servicio de tomcat. Para hacer esto: Entrar a Panel de Control → Herramientas Administrativas → Servicios y ahí buscar en la lista el Servicio de Apache Tomcat 4.1, seleccionarlo y darle INICIAR. |
| Paso 15 | Probar: http://localhost:8080/CumpleWEB usuario: administradorc contraseña: admin |

5.4 Productos de Trabajo

Los productos de trabajo se refieren a todos los documentos que se crean a lo largo del ciclo de vida del software, en cada etapa se generan documentos importantes los cuales pueden ser entregados o no al cliente. Esta sección tiene por objetivo listar y describir brevemente los productos de trabajo por cada etapa del desarrollo del Sistema de Cumplimiento Regulatorio que se recomiendan entregar al cliente en la Etapa de Transición.

Cabe señalar que los productos de trabajo que listo a continuación se debe a que el Sistema de Cumplimiento Regulatorio se traduce en una necesidad de una empresa en particular la cual por diseño y desarrollo particular por cuestiones de negocio se entiende que se debe entregar el código completo y el diseño anexo a éste, lo que también por cuestiones de negocio no se entregarían en su totalidad cuando un sistema es vendido como un producto propietario de la empresa que brinda el servicio.

- Para la Etapa de Concepción:

Contrato

El contrato es un documento legal que se realiza entre la empresa que presta el servicio y el cliente que lo solicita.

Matriz de Necesidades del Cliente

La matriz de necesidades del cliente es en general un documento que abarca la lista de requerimientos de usuario para definir el alcance del sistema.
Se incluyen requerimientos internos establecidos para la mejor construcción del sistema.

- Para la etapa de Elaboración:

Diagramas de Casos de Uso

Los diagramas de casos de uso son en general un documento que plasma de manera gráfica las funcionalidades del sistema y describe cada uno de los escenarios posibles para llevar a cabo la operación del sistema, la cual debe corresponder a la operación necesaria y requerida por el cliente.

Modelos conceptuales

Los modelos conceptuales son la representación de conceptos en el dominio de un problema. Muestran los conceptos relevantes e identifican los conceptos del negocio que son la base para el desarrollo orientado a objetos. Se relacionan mediante Asociaciones entre Conceptos que muestran la relación lógica o física que existe en el mundo real entre dos conceptos, se pueden nombrar a las asociaciones para clarificar el modelo.

Plan del Proyecto

El plan de proyecto es en general un documento que describe el plan de trabajo del desarrollo del software.

Diseño de Arquitectura

El diseño de la arquitectura detalla las capas lógicas de la aplicación y de cómo se implementarán en la plataforma de ejecución seleccionada. Describen la solución en términos de paquetes y clases de negocio.

Diseño de la base de datos

El diseño de la base de datos es en general un documento que detalla de manera gráfica la estructura lógica del sistema.

Diccionario de Datos

El diccionario de datos es en general un listado de cada elemento de la base de datos, con la descripción detallada de lo que representa ese elemento para el sistema.

- Para la etapa de Construcción:

Diagramas de Secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

Código de Sistema

El código del Sistema se refiere en su totalidad a la programación del Sistema.

Matriz de Pruebas del Sistema

La matriz de pruebas del sistema en general se refiere a un documento que plasma las pruebas que se realizaron durante la construcción del sistema.

- Para la etapa de Transición:

Manuales del Sistema

Los manuales del Sistema se refieren a un documento o documentos que detallan de forma específica las funcionalidades del Sistema.

Guía Rápida del Sistema.

La guía rápida del Sistema se refiere a un documento que plasma los mínimos pasos necesarios para llevar a cabo el correcto funcionamiento del Sistema.

Documentos técnicos de configuración

En este apartado tenemos documentos que ayudan a la configuración de sistema o documentos

que facilitan el uso del Sistema. Las fuentes son variables en cada sistema.

Para el Sistema de Cumplimiento regulatorio tenemos:

- Guía para realizar un respaldo de la base de datos en SQLServer 2000.
- Guía para realizar una restauración de base de datos en SQLServer 2000.
- Guía para permitir Pop Ups en el Sistema.
- Mensajes comunes en Cumple

Glosario de Términos

Documentar las definiciones de términos y acrónimos ayuda a otros documentos a ser más concisos y precisos. Un glosario compartido ayuda a prevenir malos entendidos y hace más fácil a miembros nuevos del equipo a ser productivos.

5.5 Glosario de Términos

| | |
|--------------------------|--|
| Activo Fijo | <p>Las propiedades, bienes materiales o derechos que en el curso normal de los negocios no están destinados a la venta, sino que representan la inversión de capital o patrimonio de una dependencia o entidad en las cosas usadas o aprovechadas por ella, de modo periódico, permanente o semi-permanente, en la producción o en la fabricación de artículos para venta o la prestación de servicios a la propia entidad, a su clientela o al público en general.</p> <p>El "activo fijo" se clasifica en tres grupos:</p> <p>a) "tangible", que comprende las propiedades o bienes susceptibles de ser tocados, tales como los terrenos, los edificios, la maquinaria, etc.;</p> <p>b) "intangibles", que incluye cosas que no pueden ser tocadas materialmente, tales como los derechos de patente, los de vía, el crédito mercantil, el valor de ciertas concesiones, etc.;</p> <p>c) las inversiones en compañías afiliadas.</p> |
| Acuerdo | Es una decisión tomada en común por dos o más personas, por una junta, asamblea o tribunal. También se denomina así a un pacto, tratado o resolución de organizaciones, instituciones, empresas públicas o privadas. |
| Auditor | Persona encargada de verificar, de manera independiente, la calidad e integridad del trabajo que se ha realizado en un área particular. |
| Auditoría | Proceso planificado y sistemático en el cual un auditor obtiene evidencias objetivas que le permitan emitir un juicio informado sobre el estado y efectividad de una organización. |
| Autoridad | El poder ejercido por una persona legitimada por una institución o razón conforme a unas funciones que le son generalmente reconocidas. |
| Cumplimiento | Actuación que se lleva a cabo como consecuencia de una obligación, una promesa o una orden, puede derivar en el fin de un plazo o un periodo determinado. |
| Dictamen | Opinión, parecer, juicio acerca de alguna cosa que emite alguna persona o corporación. De esta manera es el documento escrito que acredita y justifica el cumplimiento de un requisito de trámite en proceso. |
| Disposición Legal | Colocación de una o más cosas según un orden o en una posición adecuada y conveniente conforme a la ley o prescrito por ella. |
| Evidencia | Es la actividad necesaria que implica demostrar la verdad de un hecho, su existencia o contenido según los medios establecidos por la ley. |

| | |
|-----------------------|--|
| Gestión | Es la actividad que se encarga de ejecutar todas aquellas órdenes planificadas por la administración, en este caso, de empresas. Más específicamente, se trata de realización de diligencias enfocadas a la obtención de algún beneficio, tomando a las personas que trabajan en la compañía como recursos activos para el logro de los objetivos. |
| Instrucción | Conjunto de reglas o indicaciones que se dan para hacer una actividad o para manejar un objeto. |
| Legislación | Un cuerpo de leyes que regulan una determinada materia o ciencia. |
| Ley | Es una norma jurídica dictada por el legislador. Es decir, un precepto establecido por la autoridad competente, en que se manda o prohíbe algo en consonancia con la justicia. Su incumplimiento trae aparejada una sanción. |
| Lineamientos | Conjunto de acciones específicas que determinan la forma, lugar y modo para llevar a cabo una política en materia legal relacionados con la misma. |
| Marco Legal | Son las bases sobre las cuales las instituciones construyen y determinan el alcance y naturaleza de sus actividades, determinando la legislación aplicable dependiendo de las mismas. |
| No Conformidad | Situación aislada que, basada en evidencias objetivas, demuestra el incumplimiento de algún aspecto de un requerimiento de control que permita dudar de la adecuación de las medidas para preservar la confidencialidad, integridad o disponibilidad de información o requisito. |
| Norma Jurídica | Es una regla u ordenación del comportamiento humano dictado por la autoridad competente del caso, con un criterio de valor y cuyo incumplimiento trae aparejado una sanción. Generalmente, impone deberes y confiere derechos. Se diferencia de otras normas de conducta en su carácter heterónomo (impuesto por otro), bilateral (frente al sujeto obligado a cumplir la norma, existe otro facultado para exigir su cumplimiento), coercible (exigible por medio de sanciones tangibles) Y externo (importa el cumplimiento de la norma, no el estar convencido de la misma). |
| Normatividad | Es un conjunto de criterios o fórmulas, con las que se rige la conducta humana, Pueden ser éstas de carácter voluntario, (del orden moral) o pueden ser obligatorias (del orden jurídico). Toda normatividad para ser válida debe cumplir con los siguientes requisitos : a) Ser proclamada por autoridad reconocida y empleando la difusión necesaria. b) Tener esa autoridad, suficiente ascendente moral. c) Debe comprender reglas claras y de aplicación general. d) Especificar la penalidad por las infracciones. e) Tener métodos de coerción o convencimiento. f) Estar fincada en una tesis de carácter ético o moral. Las normatividades derivan de las disposiciones legales aplicables a la actividad económica de la empresa y estas pueden ser Municipales, |

| | |
|--------------------------|--|
| | Estatales, Federales o Internacionales. |
| Punto | Parte o aspecto de una materia o de un asunto del que se trata. |
| Reglamentos | Es una norma jurídica de carácter general dictada por el poder ejecutivo. Su rango en el orden jerárquico es inmediatamente inferior a la ley, y generalmente la desarrolla. |
| Responsable | Obligado a responder de alguna cosa, actividad o persona. |
| Riesgo | Es la contingencia de un daño. A su vez contingencia significa que el daño en cualquier momento puede materializarse o no hacerlo nunca. El riesgo es sólo la posibilidad de que el daño creado por un agente pueda ocurrir. |
| Servlet | Pequeño programa que corre dentro del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad. |
| Tramite | Cada uno de los pasos que se realizan de manera sucesiva para solucionar un asunto. |
| Unidad de Negocio | Cada parte de una organización empresarial que cuenta con su propia estrategia, dirección y presupuesto. |

Capítulo 6

Conclusiones

Conclusiones

La gran cantidad de tareas legales necesarias para mantener a una empresa en regla se convierte en una necesidad básica por administrar. El objetivo del sistema es contar con una herramienta web capaz de facilitar el manejo de las tareas por cumplir y reducir los tiempos y costos presentados por el incumplimiento. Con el Sistema de Cumplimiento Regulatorio CUMPLE, se cuenta con una interfaz amigable, que nos proporciona las herramientas para administrar y dar seguimiento rápido y eficaz a las auditorías de carácter legal e internas de una empresa. Reduce los tiempos de entrega al mantener la información centralizada y disponible, y nos proporciona bases para una mejor toma de decisiones.

Con el uso de Cumple se encuentran los siguientes beneficios:

- Se conoce el riesgo diario en cada empresa o planta administrada por el sistema, de ser multada o clausurada.
- Se reduce el número de multas generadas por el incumplimiento.
- Se cuenta con la información detallada de cada lineamiento legal que se debe cumplir y los pasos necesarios para hacerlo.
- Se conoce el responsable de resolver cada una de las tareas, las actividades realizadas y los documentos que validan el cumplimiento.
- El tiempo invertido en re-trabajar se reduce.
- Las auditorías son llevadas en menor tiempo debido a que la información está organizada y de fácil acceso.
- Se cuenta con un panorama de las próximas actividades a realizar, lo que contribuye a una mejor planeación.
- Incrementa la colaboración entre grupos de trabajo y mejora la comunicación.
- Puede ser consultado en cualquier lugar con acceso a Internet.

Durante el desarrollo de este software, tuve la oportunidad de presentar este material a varias empresas, de diferentes giros, con lo cual, pude darme cuenta de que la metodología empleada para organizar y dar seguimiento a las actividades necesarias para cumplir un objetivo particular, se adapta con mucha facilidad a cualquier giro, lo que hace de Cumple una herramienta versátil y con gran potencial de venta.

Considero que las etapas más importantes para la construcción de un software, en particular para Cumple, son la etapa de concepción, donde se identifica el alcance del sistema y la etapa de elaboración, debido a que la definición de la metodología y la forma en que se implementará esa metodología son la base para el éxito de un software de alta calidad y de fácil mantenimiento.

Durante el desarrollo de esta tesis, cumplo con el objetivo de plasmar la información que he ido obteniendo durante mis estudios y al practicar felizmente mi profesión, en la cual debo mencionar, que se realizaron pruebas para cambiar el manejador de base de datos, donde los únicos inconvenientes fueron la transformación de formatos de fechas entre diferentes motores de bases de datos.

Sin duda, siempre encontraremos maneras de mejorar un sistema de software, pero lo más importante es siempre tener en cuenta el objetivo de cada cliente y adaptar una solución para satisfacer esas necesidades, tomando en cuenta la mayor parte de los factores influyentes en esta decisión como pueden ser: el tiempo, el número de recursos, el presupuesto, las herramientas definidas por el cliente, la infraestructura del cliente, etc.

Casos de éxito

- Grupo Metropolitano
- Corporativo Hospital Satélite

Bibliografía

Libros:

Stephen R. Schach
Ingeniería de Software Clásica y Orientada a Objetos
Sexta edición
Mc Graw Hill, 2006

Pressman, Roger S.
Ingeniería del Software. Un enfoque práctico.
Quinta edición
Mc Graw Hill, 2002

Jacobson, Ivar
Booch, Grady
Rumbaugh, James
El Proceso Unificado de Desarrollo de Software
Addison - Wesley, 2000

Martin, James
Odell, James J
Análisis y Diseño Orientado a Objetos
Prentice Hall, 1992

Cursos:

Planeación de Proyectos (PP)
ESI Center (Centro de Excelencia para la Industria del Software)
Tecnológico de Monterrey, 2006

Monitoreo y Control de Proyectos (PMC)
ESI Center (Centro de Excelencia para la Industria del Software)
Tecnológico de Monterrey, 2006

Internet:

<http://www.sun.com/>

<http://tomcat.apache.org/>

<http://www.uml.org/>

<http://es.wikipedia.org/>

<http://www.kybeleconsulting.com/>

<http://www.sparxsystems.com/uml-tutorial.html>

<http://oness.sourceforge.net/proyecto/html/index.html>

http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/bd_conceptosgenerales.htm

<http://www.ingenierosoftware.com/calidad/cmm-cmmi.php>