

**UNIVERSIDAD GABRIELA MISTRAL
FACULTAD DE INGENIERIA**

**APLICACIÓN DE GESTIÓN
DOCUMENTAL EN EL AREA DE
SEGURIDAD Y SALUD OCUPACIONAL**

Memoria para optar al título de Ingeniero de Ejecución en Informática

Autor : Cristian Alejandro Vargas Martinez
Profesor Guía : Roberto Caru Cisternas.
Profesor Integrante : Jorge Tapia Castillo.

Santiago – Chile
Octubre, 2016

INDICE

CAPITULO 1	5
INTRODUCCIÓN	5
DESCRIPCION DE LA EMPRESA	7
ANTECEDENTES GENERALES DE LA EMPRESA	7
ORGANIZACIÓN	8
SERVICIOS	9
HERRAMIENTAS DE DESARROLLO	12
CAPITULO 2	14
MARCO TEORICO	14
DESCRIPCIÓN DE LA SITUACIÓN ACTUAL	15
FLUJOGRAMA SITUACION ACTUAL	19
DEFINICIÓN DEL PROBLEMA	20
HIPOTESIS	24
REQUERIMIENTOS	26
FUNCIONALIDAD DE LOS REQUERIMIENTOS	30
PROPUESTA DE SOLUCIÓN 1	34
PROPUESTA DE SOLUCIÓN 2	35
ARQUITECTURA DE LA PROPUESTA 2 (.NET)	36
CAPITULO 3	37
FACTIBILIDAD	37
FACTIBILIDAD LEGAL	37
FACTIBILIDAD TECNICA	39
PROPUESTA DE SOLUCIÓN TÉCNICA 1	40
PROPUESTA DE SOLUCIÓN TÉCNICA 2	43
ARQUITECTURA .NET	46
FACTIBILIDAD OPERACIONAL	49
FACTIBILIDAD ECONOMICA	51
ALTERNATIVA DE SOLUCIÓN ECONÓMICA 1	52
FLUJO DE CAJA DE PROPUESTA 1	56
VAN Y EL TIR DE LA PROPUESTA 1	57
ALTERNATIVA DE SOLUCIÓN ECONÓMICA 2	58

FLUJO DE CAJA PROPUESTA 2	60
VAN Y TIR	62
ELECCION DE ALTERNATIVA	62
CAPITULO 4	64
DESARROLLO DEL TRABAJO	64
INTRODUCCION	65
SISTEMA GESTION DOCUMENTAL	66
ACTORES SISTEMA GESTION DOCUMENTAL	67
CASOS DE USO DEL NEGOCIO	69
DIAGRAMA DE CASOS DE USO SISTEMA GESTION DOCUMENTAL	70
Caso De Uso Gestionar Actividad Asignada	72
Caso De Uso Gestionar Investigaci3n de Accidentes	76
Caso De Uso Gestionar Comit3 paritario	80
Caso De Uso Gestionar Auditoria de calidad	82
Caso De Uso Crear Perfiles de Acceso	85
ARQUITECTURA SISTEMA GESTION DOCUMENTAL	88
ARQUITECTURA FUNCIONAL	89
DIAGRAMA DE CAPAS SISTEMA DE GESTION DOCUMENTAL	91
DIVISION DE LAS CAPAS SISTEMA GESTION DOCUMENTAL	92
CAPA DE PRESENTACION O CAPA CLIENTE	93
CAPA CONTROL DE PETICIONES SISTEMA GESTION DOCUMENTAL	96
MODELO (CAPA LOGICA DE NEGOCIO)	100
CAPA DE ACCESO A LOS DATOS O PERSISTENCIA	107
CAPITULO 5	110
HALLAZGOS	110
ARQUITECTURA FISICA	111
DIAGRAMA DE COMPONENTES	112
DIAGRAMA DE COMPONENTES SISTEMA GESTION DOCUMENTAL	113
DIAGRAMA DE COMPONENTES DEL SISTEMA GESTION DOCUMENTAL	115
TECNOLOGÍA DEL SISTEMA GESTION DOCUMENTAL	118
CONCLUSION	119
CAPITULO 6	122
BIBLIOGRAFÍA	122

TEXTOS _____	123
SITIOS WEB _____	124
CAPITULO 7 _____	126
GLOSARIO _____	126
Caso De Uso _____	127
ARQUITECTURA J2EE _____	128
CAPA DE PRESENTACION (View) _____	131
CAPA DE CONTROL DE PETICION _____	132
STRUTS _____	133
CAPA DEL MODELO DE NEGOCIO (MODEL BUSSINESS) _____	135
ACTION _____	137
LA ARQUITECTURA EJB _____	137
Tipos de Componentes _____	141
COMPARACIÓN ARQUITECTURA J2EE V/S .NET _____	148
CONCEPTO DE SAP / R3 _____	149

CAPITULO 1

INTRODUCCIÓN

Este informe tiene como misión mostrar las distintas etapas que se llevaron a cabo para poder realizar el proyecto Sistema de Gestión Documental perteneciente a la empresa Electroandina S.A.

Este proyecto se basa en la necesidad de integrar los distintos módulos de procesos del área de seguridad y salud ocupacional, para que estos puedan tener una comunicación con la plataforma SAP de Electroandina.

Para esto se hará un exhaustivo análisis que comprende distintas etapas que tendrán que recoger los requerimientos para el sistema de gestión documental, Para tal efecto se ocuparan distintas herramientas como J2EE (Java), UML, Arquitectura J2EE.

Las cuales permitirán dar posibles soluciones para poder capturar la lógica del negocio del sistema,

El enfoque que tendrá este proyecto será el de la Arquitectura j2ee, esto implica hacer un análisis que define claramente el diseño del proyecto SISTEMA GESTION DOCUMENTAL.

Para el análisis se necesita realizar toda la documentación necesaria para recoger los requerimientos y antecedentes del SISTEMA GESTION DOCUMENTAL, además de realizar un diagnóstico del problema para poder dar una solución que pueda satisfacer a cabalidad estos requerimientos, además de un estudio de factibilidad del proyecto SISTEMA GESTION DOCUMENTAL, Siguiendo con un diseño lógico enfocado a la arquitectura y siguiendo el estándar MDA (Domain Model

Architecture) de J2EE, es decir poder restringir el dominio y la solución del problema para el SISTEMA GESTION DOCUMENTAL.

A continuación se desarrollaran las distintas etapas del proyecto para la unidad de negocio SISTEMA GESTION DOCUMENTAL.

DESCRIPCION DE LA EMPRESA

ANTECEDENTES GENERALES DE LA EMPRESA

NOMBRE : Domo Chile Consultores

R U T : 76.463.002-7

GIRO : Asesoría y servicios computacionales

DIRECCION : Los jardines de colina 8

COMUNA : Colina, Santiago

TELEFONO : (02) 2642216

PAGINA Web : <http://www.domoconsultores.com/>

EMAIL : consultas@domoconsultores.com

ORGANIZACIÓN

Domo Chile consultores. Ha experimentado un sostenido y vigoroso crecimiento, con una sana posición financiera, y ha adecuado su estructura organizacional a la realidad de cambios provenientes del avance tecnológico y de mercados cada vez más exigentes y competitivos.

Su representante legal es Heber Morales Velásquez, y desde su inicio ha liderado proyectos innovadores, de valor agregado para el medio nacional, y que además, han incorporado el uso de las más avanzadas tecnologías del rubro computacional. Domo Chile consultores ha logrado desarrollar una gran experiencia en áreas como: Proyectos de Integración de Sistemas, Soluciones Informáticas, Servicios de Computación, Soporte Técnico y Comercialización de Aplicaciones de Software y Plataforma de Hardware.

Además el área de Consultoría está abocada a entregar las herramientas necesarias para la Puesta en Marcha de los sistemas administrativos y aplicaciones empresariales, ofreciéndole a sus clientes distintas alternativas de Capacitación para su personal, con el fin de lograr un conocimiento cabal de la operatoria y objetivos de ellos, llevándolo a optimizar en un 100% el uso de los sistemas.

Hacia el área Web, atiende demandas del mercado, brinda herramientas, robustas, confiables y vigentes que ayude automatizar las tareas administrativas, es decir, análisis, diseño, reingeniería, prototipos y capacitación específica.

SERVICIOS

Domo Chile da múltiples servicios y soluciones basándose en lenguaje java, en base a Análisis, diseño y documentación utilizando lenguaje de modelamiento UML y desarrollado en la plataforma corporativa java J2EE¹, utilizando herramientas de desarrollo que simplifica el diseño y construcción de componentes como servlets, páginas JSP², EJB³. Algunos de los servicios que presta son:

- ◆ Servicio para el desarrollo de aplicaciones: DOMO CHILE realiza servicios de aplicaciones altamente calificados y geográficamente distribuidos para desempeñar funciones de administración y mantenimiento con herramientas open source.
- ◆ Seguridad: Las soluciones de seguridad de Domo Chile se enfocan en la administración de seguridad, una administración automatizada de identidad y administración de eventos de seguridad. Esta solución lo ayuda a alcanzar el rol al permitir que los usuarios, sistemas y aplicativos estén online rápidamente, al mismo tiempo que administra los usuarios, de forma eficaz, accede a los derechos y preferencias de privacidad en todo el ciclo de vida de la identidad.

¹ Ver Anexos Arquitectura J2ee

² Ver Anexos Capa De Presentación

³ Ver Anexos Capa De Lógica De Negocio.

- ◆ Además permite una administración de eventos de seguridad lo ayuda a monitorear, correlacionar y reaccionar a incidentes de seguridad del negocio.

Domo Chile: es una empresa dedicada a la consultoría informática en gestión empresarial, el outsourcing y la prestación de servicios tecnológicos tales como el desarrollo, implantación o mantenimiento de aplicaciones y sistemas informáticos. El objetivo es dotar a los clientes de soluciones tecnológicas innovadoras que les permitan mejorar su forma de hacer negocios y de competir en sus mercados. Para cumplir este objetivo, Domo Chile usa de diversas tecnologías (gestión de procesos de negocio, soluciones móviles), Abordan proyectos de todos los tamaños y son capaces de adaptarse a los requerimientos de los clientes más exigentes. Centrados en la plataforma J2EE¹ y seguidores de sus prácticas de diseño y desarrollo pone a disposición, los servicios de Consultores, que brindarán su apoyo en el proceso de toma de decisiones en torno a tecnologías informáticas, permitiendo reducir los plazos de las evaluaciones y minimizar los riesgos.

- ◆ El proyecto integral de Domo Chile engloba la elaboración completa del sistema informático y su mantenimiento después de la implantación. Cuando el cliente solicita un presupuesto, se hace un primer estudio de valoración del proyecto denominado "análisis previo"

¹ Ver Anexos De Arquitectura J2ee

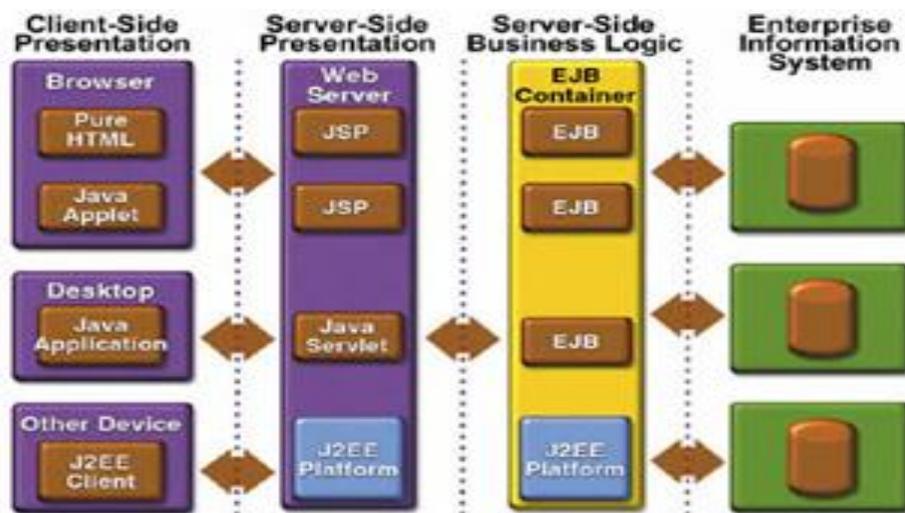
sin compromiso alguno por parte del cliente; en él, dependiendo de la complejidad del proyecto, se dan unas cifras orientativas del costo de las diferentes fases que se han de seguir hasta la finalización del proyecto, así como de los resultados y documentos que se obtienen en cada uno de ellos.

Si el cliente está de acuerdo con los términos de dicho documento, se procede a la realización del análisis detallado donde se determinan las características y los costos definitivos. Con una exhaustiva documentación por parte del análisis y diseño de la plataforma J2EE¹.

¹ Ver Anexos De Arquitectura J2ee

HERRAMIENTAS DE DESARROLLO

DOMO CHILE Asociados crea soluciones a partir del lenguaje java y j2ee¹, que es una herramienta de desarrollo, esta es una especificación abierta que define una plataforma de desarrollo de aplicaciones distribuidas orientadas al negocio. Esta se basa en la separación de las capas, esta separación de capas permite una delimitación de responsabilidades a la vez satisface los requisitos no funcionales de este tipo de aplicaciones (escalabilidad, extensibilidad, flexibilidad, etc.) y disminuye el acoplamiento de las mismas. El número de capas de una aplicación j2ee variara según su complejidad y/o necesidades.



¹ Ver Anexos De Arquitectura J2ee

Esta estructura de capas se ha constituido es un estándar a la hora de desarrollar aplicaciones distribuidas para sistemas empresariales dejando obsoleto el modelo cliente-servidor. Esta estandarización es bastante importante ya que si la arquitectura es clara y está diseñada en términos de alto nivel, sin basarse en código explícito de cada plataforma, Por lo tanto será mucho más fácil una posible migración en la etapa del diseño del sistema.

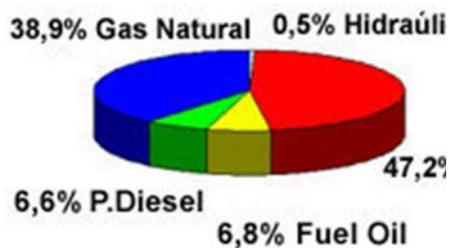
CAPITULO 2

MARCO TEORICO

DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

Electroandina es la empresa líder en generación y comercialización de energía eléctrica del Sistema Interconectado del Norte Grande de Chile -SING-, con una capacidad instalada actual de 1029 MW, que equivale al 29 % de la potencia instalada total, un 27% del total de líneas de transmisión y un 42% de participación en las ventas de energía a clientes.

Las centrales de generación se encuentran ubicadas en la ciudad de Tocopilla, 200 Km. al Norte de Antofagasta, estas Unidades de Generación Eléctrica utilizan petróleo, carbón y gas natural como materia prima para la generación de energía eléctrica, esta base de combustibles que utiliza contribuye altamente a la calidad de servicio a sus clientes. Al tener un parque de 10 unidades generadoras basadas en distintos combustibles, la empresa cuenta con un respaldo permanente para el suministro de energía eléctrica.



Este gráfico muestra la variedad de fuentes de energía que utiliza Electroandina para dar servicio a sus clientes.

Los clientes de Electroandina son los principales yacimientos cupríferos de la segunda región, como son: Chuquicamata, El Abra y Radomiro Tomic. A través del desarrollo de un proyecto muy importante y novedoso, ya que este cliente requiere energía eléctrica en 60 ciclos (en Chile se utilizan 50 ciclos como norma para los servicios eléctricos).

Hay que considerar que la energía constituye el 10% de los costos de extracción de estas compañías. Además, se trabaja en lugares tan alejados de los centros urbanos, que los servicios adicionales que puede prestar una empresa eléctrica como Electroandina, son fundamentales para lograr la mayor satisfacción del cliente.

La compañía Electroandina creó una unidad de negocio aparte a lo que se refiere la empresa en sí, esta unidad de servicios industriales (Sistema GSI) que nace como una unidad de negocios independiente que aprovecha la vasta experiencia adquirida en el trabajo con los clientes mineros e industriales de la región y con el gran respaldo de una empresa sólida y eficiente. Las áreas de negocio de Electroandina servicios industriales son:

SERVICIOS PORTUARIOS: Construido en 1987 como una unidad de apoyo a la generación eléctrica, hoy la División Puerto de Tocopilla, se alza como el principal puerto privado del norte. Producto de la visión empresarial, este importante terminal portuario está transformado en una división de negocio de la compañía y maximiza el uso de su capacidad instalada, ofreciendo servicios integrales de recepción, almacenamiento y despacho de carga, a quienes requieran un servicio rápido eficiente y confiable.

Este puerto está diseñado para el manejo de cargas generales, gráneles sólidos y líquidos. Posee un muelle mecanizado con un cabezal de 80 metros y un puente de acceso de 200 metros de longitud, apto para el tráfico de camiones.

SERVICIO DE MANTENIMIENTO INDUSTRIAL: Esta unidad de Servicios de Mantenimiento, se constituye como un área de

mantenimiento integral, que permite mantener operativo su parque de 1029 MW. El objetivo principal de esta unidad es mantener una alta disponibilidad de la infraestructura productiva, producto del crecimiento sucesivo de las instalaciones y las consecuentes modernizaciones de éstas.

Esta unidad cuenta con una maestranza amplia y moderna, talleres de mantenimiento de generadores de vapor, caldereros, soldadura y un área especializada en protecciones eléctricas y medidas de seguridad.

SERVICIO DE MANTENIMIENTO DE LINEAS DE TRANSMISION Y SUBESTACIONES DE ALTA TENSION: tiene como principal función lavar líneas y subestaciones energizadas y adicionalmente efectuar reparaciones de estructuras, cambios de aisladores y montajes en subestaciones. Este lavado evita que los componentes del sistema de transmisión pierdan su calidad de aislamiento por efecto de la contaminación y así se disminuyen las pérdidas de energía por transmisión. La transmisión de electricidad en el norte se asocia inmediatamente al mantenimiento que supone esa enorme infraestructura de que está dotado el SING. Miles de kilómetros de líneas, decenas de subestaciones y una intrincada ingeniería que permite el flujo de grandes volúmenes de energía entre generadores y clientes, y éstos entre sí, hablan de una tarea que además de esfuerzos, requiere de un manejo complejo, preciso y oportuno de las empresas que la tienen a cargo.

El problema de la misma lejanía geográfica de las unidades de negocios de Electroandina (GSI) hace que exista una fuerte dependencia a través del correo electrónico para la realización de sus operaciones y planificación de sus actividades, no contando con un

“sistema de gestión documental” de servicios para sus jefes de sucursales y responsables de área, para poder solicitar la distinta gama de servicios a electroandina en su área de seguridad, no existe un sistema que pueda gestionar estos servicios en el área de seguridad y salud ocupacional, además de otorgar funciones para sus distintos administradores, en las distintas áreas RRHH de Electroandina. Tampoco hay un back up de la gran cantidad de documentos que se generan, con formato estándar y que puedan ser respaldados en una base de datos.

A modo de ejemplo describiremos lo que ocurre, respecto a uno de los servicios que se gestionan en Electroandina el área de seguridad y salud ocupacional:

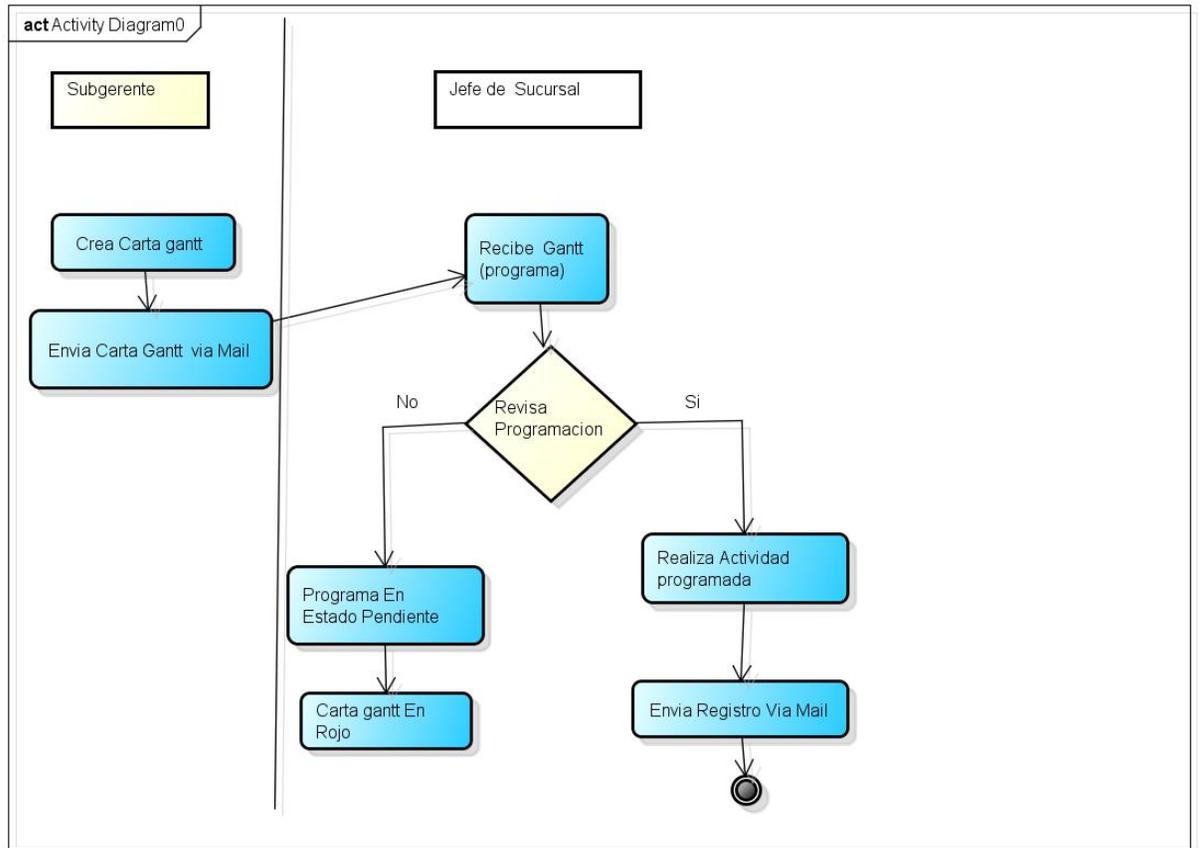
- ◆ La subgerencia de prevención de la casa matriz, pueden enviar una pauta de actividades (carta Gantt) a realizar a través del correo corporativo de electroandina, el subgerente de la casa matriz llena un documento estándar y lo adjunta al correo con otras plantillas de documentos que deben ser llenados para concretar la actividad designada, esta viaja a través del correo corporativo.

- ◆ Esta pauta de actividades es recibida por un Jefe de sucursal. Cabe también señalar que se va generando una cantidad de documentos adjuntos importantes ya que, este jefe de sucursal debe enviar un registro de cumplimiento de las actividades detalladas en la carta Gantt que envía el subgerente de la casa matriz.

- ◆ Electroandina también posee una plataforma que administra las operaciones de la empresa, este sistema está hecho en SAP R/3⁷, software de negocios que ofrece soluciones estándares para las necesidades de los distintos departamentos como finanzas, recursos humanos, contabilidad, etc. Además de estas soluciones estándares, el ambiente de desarrollo de SAP y su sistema de información, proveen a los clientes con poderosas herramientas para desarrollo y adaptación del sistema a los requerimientos de la empresa. Por otra parte una desventaja es que el sistema SAP/R3 no está integrado a al sistema de gestión documental, que se necesita crear.

FLUJOGRAMA SITUACION ACTUAL

⁷ Ver Anexos Conceptos De SAP



DEFINICIÓN DEL PROBLEMA

La plataforma de Electroandina es SAP/R3⁸ esta posee múltiples capacidades que se adecuan al negocio de la empresa, como la flexibilidad que tiene para adaptarse a nuevos cambios del método operacional y las diferentes áreas de la empresa, con un grado de apertura que el sistema posee para integrarse a otros sistemas informáticos.

Además de tener certificaciones de calidad en el software y al fabricante que lo implante, y por ultimo asegurar la fiabilidad de los datos y controlar el acceso y uso de la información.

- ◆ Pese a estas garantías que posee la plataforma de Electroandina, se necesita tener una comunicación con el nuevo sistema de gestión documental que se debe implantar. esta comunicación es necesaria para la integración de SAP R/3 con las demás áreas de Seguridad y salud ocupacional, ya que SAP contiene la base de datos, de todo el personal de electroandina con sus respectivos cargos y roles dentro de la organización, cualquier proyecto necesitara de un acceso, para la identificación del trabajador y la actualización de sus antecedentes en caso de cual siniestro que pudiera ocurrir en su jornada laboral.

- ◆ Como se detalla en la situación actual de la empresa, los subgerentes de la casa matriz, envían una carta Gantt con la programación de las actividades vía mail corporativo, además se copian documentos como planillas Excel y otros documentos tipo, con permiso de lectura, que se dejan en una carpeta

⁸ Ver Anexos Conceptos De SAP

compartida del subgerente, para que los jefes de sucursales respalden en su notebook.

- ◆ Cuando esta carta Gantt es recibida por un jefe de sucursal, este se encarga de revisar el programa de actividades, y planifica estas actividades asignando un responsable para que se cumpla la actividad, esta actividad se valida por medio de una firma y nombre de los asistentes, con un OK que fue realizada, por lo tanto de esta manera se genera un registro, el cual es enviado al subgerente de la casa matriz. Este registro será revisado por el subgerente y se marcará en verde como actividad OK realizada.
- ◆ En caso de que la actividad de la carta Gantt o programa, no fuese realizado, el subgerente revisará el registro enviado por el jefe de sucursal, este se marcará con rojo como NOOK, como parte de los ítems de la carta Gantt como actividad no realizada, quedando en estado pendiente.
- ◆ En tanto el jefe de sucursal, no tiene la certeza de la actualización de su programa (carta Gantt) es recibida o rechaza por el subgerente de la casa matriz, ya que toda esta respuesta del flujo se hace vía mail corporativo.
- ◆ Tomando en cuenta que electroandina posee 36 sucursales, por cada sucursal, se generan un número de 4 cartas Gantt distintas,

por jefes de sucursales, Por lo tanto el número de registros enviados por mail corporativo, crece de forma exponencial, si a eso le sumamos la cantidad gigante de documentos adjuntos, se hace engorroso buscar el respaldo, de una gestión de carta Gantt específica, ya que tiene una fuerte dependencia con el correo de cada subgerente o jefe de sucursal.

- ◆ No existe un proceso efectivo de respaldo de documentos, como por ej. una base de datos, en la cual se guarde, la gran cantidad de documentos, para luego digitalizarlos y que tenga un respaldo. En caso de alguna contingencia u accidente ya sea siniestro, etc.
- ◆ Uno de los puntos más críticos, es la generación de indicadores mensuales de cumplimiento, que se hace imposible por la lentitud, en la revisión de los distintos mail enviados por los jefes de sucursales, y los mail enviados por los subgerentes, para poder realizar un seguimiento confiable y dar como resultado un indicador también preciso, y tener una consistencia fiable en los datos. A su vez una base de datos, resguardada frente a cualquier siniestro u accidente.
- ◆ Un punto importante ocurre cuando el subgerente envía la carta Gantt, con las actividades, además envía un vínculo de una carpeta compartida, con documentos adjuntos o documentos tipos, como una especie de repositorio. esto provoca un colapso en la red, ya que muchos usuarios tratan de acceder a una carpeta compartida de un subgerente, Además de la poca seguridad de que los archivos puedan ser borrados por cualquier

usuario, no habiendo un control de versiones y los riesgos de sufrir ataques de virus, etc.

- ◆ Un punto importante a destacar es, que todas las actividades que se asignen en una carta Gantt o programa necesitan además de una verificación en terreno, por lo tanto se adjuntan una serie de documentos que son digitalizados y enviados como adjuntos en los correos, estos documentos son de suma importancia ya que son firmados por los distintos jefes de área, y a su vez deben estar disponible ante cualquier auditoria por parte del seremi de salud o la inspección del trabajo, que tienen carácter de norma por lo tanto, cualquier incumpliendo puede llevar a una sanción.

HIPOTESIS

Tomando como base la descripción del problema y recopilando información exhaustiva en la unidad de seguridad y salud ocupacional, de electroandina a través de distintas entrevistas, encuestas tipo, además de observar los distintos procesos involucrados en el negocio se pudieron definir los siguientes objetivos que darán las señales concretas para encontrar los requerimientos para el sistema de gestión que se necesita crear.

Objetivos del sistema:

- ◆ Mejora en el tratamiento de la gestión de las cartas Gantt, que incluye realizar un sistema que entregue una mejor atención a los usuarios subgerentes y jefes de sucursales, además de los responsable de área, con el objetivo de llevar métricas online, sobre los distintos indicadores de cumplimiento de acuerdo al programa que se esté llevando a cabo.
- ◆ Ser capaz de tener una aplicación modular y flexible en el tiempo, ya que este proyecto persigue ir incorporando más procesos, en el ámbito de seguridad y salud ocupacional y una permanente integración con las demás áreas de la compañía, por ejemplo RR.HH.
- ◆ Ampliar los mercados a más zonas de Chile y la misma América Latina por la necesidad de nuevos mercados implica la necesidad de tener de forma urgente una tecnología de punta que ayude con herramientas flexibles a la toma de decisiones a nivel estratégico y dar una ventaja comparativa frente a la competencia en búsqueda de realizar un mejor servicio y negocio.

REQUERIMIENTOS

Respecto a los problemas que se plantearon para la creación de un sistema de seguridad y salud ocupacional, en electroandina, se necesita mejorar todo el proceso de la gestión de una carta Gantt de actividades, y la gestión de accidentes laborales esto implica una integración fuerte con la plataforma SAP R/3, estas mejoras en estos procesos serán explicadas a continuación:

- ◆ El subgerente de la casa matriz, envía una carta Gantt o programa de actividades a realizar en este caso charlas de capacitación o verificaciones de estas mismas vía mail, además de un vínculo para que el usuario receptor pueda obtener los documentos adjuntos, el usuario receptor es el jefe de la sucursal, el cual revisa el programa enviado designando las actividades a sus distintos equipos de trabajo, por lo general son 4 programas o cartas Gantt distintas, que envía el subgerente a al jefe de sucursal,

esta carta Gantt debe ser actualizada por los jefes de sucursal e informada al cierre de actividades al subgerente, utilizando distintos colores para clasificar el estado de estas, como por ej. un semáforo. Las distintas tareas van acompañadas de una serie de documentos adjuntos que son impresos por los jefes de sucursal y que necesitan ser firmados por los trabajadores y nuevamente digitalizados como imagen o documento Pdf, para su envío con la actualización de la carta Gantt, y cambiando el estado de la actividad y llevando un registro de cumplimiento de la misma dentro de un mes, dependiendo de la duración de tarea. Con toda esta información concentrada va repitiendo el ciclo de actualización de carta Gantt, y enviando documentos adjuntos vía mail y depositando estos documentos en una carpeta compartida, que deja el subgerente de sucursal, generándose una gran cantidad de información y difícil manejo y control en el monitoreo de esta, y una gran cantidad de mails, y actividades con estado pendiente, lo que dificulta la creación de indicadores de cumplimientos de estas actividades asignadas, lo que hace engorrosa la espera del cierre de una actividad asignada, que puede tardarse, días, semanas o meses, y estar vulnerable en cualquier momento a una fiscalización por parte de la autoridad, porque se maneja mucha normativa de salud ocupacional que son decretos que debe cumplir toda empresa como la OHSAS 18001 y ISO 9001, que son fiscalizadas por la seremi de salud , inspección del trabajo, etc.

- ◆ Algo similar ocurre cuando, un jefe directo de un trabajador accidentado se le envía documentación, para que este haga una visita en terreno frente a un accidente laboral o siniestro del afectado, estos documentos necesitan ser firmados por el trabajador o testigos del accidente y se debe ingresar toda esta información en un proceso de gestión, que sea capaz de generar un reporte para realizar el control y seguimiento de este incidente. Toda esta información se va guardando en archivos físicos de storebox, que van generando una gran cantidad de papeles que no son digitalizados ni respaldados en un repositorio SVN confiable o base de datos, lo cual genera una pérdida de control en su gestión.

- ◆ La generación de actividades asignadas mediante carta Gantt, obedece a una planificación anual que realiza electronadina en su área de seguridad y salud ocupacional, esto permite tener una serie de indicadores y métricas que facilitan la visión global de la compañía y poder realizar cruces de información en las BD que se alimentan diariamente de la información que es recaudada en estos procesos, que se deben implantar en la gestión documental.

- ◆ Frente a la necesidad de darle un mayor control a los usuarios y llevar un monitoreo constante de las actividades que se realizan en el área, es necesario tener un módulo específico al área de auditoria,

por la importancia de la información de requerimientos normativos, en cual se deben identificar cualquier hallazgo que genere una desviación en los procedimientos normativos que se ejecutan. identificando claramente la sucursal y áreas responsables, que ejecutan e implementan estos procedimientos, ya que distintas autoridades del gobierno pueden requerir esta información y si no está bajo la normativa, se expone a grandes multas y clausura por los distintos decretos que la rigen.

- ◆ Otro requerimiento, el sistema debe contar con una administración de perfiles y roles, ya que el volumen de documentación que se maneja es amplio y creciente en el tiempo. considerando el gran número de sucursales y usuarios, que puedan optar a distintas vistas y pestañas con permisos manejados por un usuario administrador.

- ◆ Un punto relevante que se mencionó en la descripción del problema, es la necesidad de contar con sistema de gestión documental en el área de seguridad y salud ocupacional, ya que la demanda es bastante alta, si se piensa que los principales clientes que tiene son Soquimich, Codelco, estos cuentan con una gran cantidad de trabajadores que prestan

servicios en sus faenas y necesitan de una supervisión constante y remota durante su jornada laboral, por el tipo de trabajo que realizan deben contar con toda una implementación de seguridad que debe regir las normativas. Lo que implica que toda la información que se maneje, debe tener alta disponibilidad frente a cualquier accidente dentro de faenas en los clientes, como en las mismas áreas internas de electroandina, Lo que implica el cambio de tecnología y la necesidad de implementar una lógica de negocio consistente, escalable en el tiempo, es decir con componentes reusables que garantizan mayor adaptación a cualquier módulo o mejora que se quiera implantar en los procesos ya descritos. por último es la necesidad de comunicación que necesita Electroandina con su plataforma de SAP, esta comunicación va a ser posible con la participación de otra tecnología que tenga una arquitectura de n-capas y muchas más ventajas de las mencionadas.

FUNCIONALIDAD DE LOS REQUERIMIENTOS

Se requieren para el sistema de gestión documental y los distintos procesos separarlos en módulos de trabajo para delimitar el contexto del problema, según los requerimientos se han identificado los siguientes módulos:

- ◆ Gestión actividades asignadas: En este módulo el jefe de sucursal debe tener la planificación anual de las actividades, podrá tener una medición mensual del cumplimiento de estas

mediante un porcentaje, se podrá visualizar las charlas y verificaciones como actividades, seleccionando al responsable y sucursal, tendrá actividades de higiene ocupacional, actividades de emergencias de simulacros, además de las verificaciones de los informes, generación de acciones correctivas y visualizar los responsables de las acciones correctivas y fechas.

- ◆ Gestión e investigación de accidentes: este módulo tendrá campos predeterminados de la ficha de personal, estos datos serán obtenidos de un proceso que se comunicara con SAP que contiene la base de datos del personal, este módulo debe retroalimentar la matriz de peligro, el usuario visualiza el proceso del accidente, deberá consultar al sistema todos los peligros que están identificados, si se identifica el peligro que afecto al trabajador, se debe seleccionar los controles que pudiesen haber sido ineficaces, si el trabajador se accidento por otro motivo se debe agregar un nuevo peligro, por lo tanto se agregara como nuevo control de peligro, el usuario hace una sugerencia de control operacional y actualiza la matriz de peligro, con esto se identificara un nuevo peligro y una nueva medida de control.
- ◆ Gestión de Comité paritario: En este módulo el administrador de perfiles de usuario, dará los accesos solo al personal representante del comité paritario, el usuario del comité paritario adjuntara un acta de reunión mensual, con los acuerdos de seguridad de la empresa, mejora de acciones correctivas, este debe permitir asignar un responsable y una fecha de los acuerdos, este módulo los usaran todas las sucursales ya que tiene una normativa legal por el decreto

DS54 , además deberá tener usuarios fiscalizadores que harán un control y monitoreo cuando las tareas no se cumplen y están pendientes de las 32 sucursales y actas mensuales que no se realizan, etc.

- ◆ Gestión de auditoría de calidad(OHSAS 18001) y seguridad(ISO 9001): en este módulo habrá un actor que se encargará de auditar el cumplimiento de estas normas , se deberá ingresar los hallazgos de auditoría, es decir cuando el usuario auditor identifique que hay una desviación de los requisitos normativos, podrá seleccionar el área de servicio, condiciones del terreno, que tienen incumpliendo, se podrá asignar el proceso de la sucursal auditada, y se comunicara con SAP para validar el personal responsable y se podrá ingresar el hallazgo y poder clasificarlo como no conforme, observación, oportunidad de mejora y deberá permitir seleccionar el incumpliendo normativo y adjuntar la evidencia del hallazgo, debe permitir guardar distintos hallazgos de distintos procesos realizando una selección y permitir guardar el levantamiento y realizar una notificación de los hallazgos al responsable del proceso y poder revisar el hallazgo e ingresar una acción correctiva.

- ◆ Uno de los requerimientos fundamentales para llevar a cabo este análisis es la herramienta UML que es un lenguaje de

modelamiento de datos bastante flexible que es capaz de identificar todos los procesos de negocios, además de sus actores del sistema pasando por todos los escenarios donde se cumpla algún proceso automatizado, además de ubicar toda la importancia de la fase de análisis del negocio, es fundamental para la etapa del diseño ya que posee metamodelos capaces de recoger toda la arquitectura del sistema SISTEMA GESTION DOCUMENTAL y además tomar todos los requerimientos no funcionales utilizando patrones de diseño,etc.

PROPUESTA DE SOLUCIÓN 1

La propuesta 1 consiste en usar el lenguaje JAVA con su versión J2EE¹ (ENTERPRICE EDITION), esta plataforma creada por SUN SySTEMS en 1997 que sugiere la arquitectura basada en componentes con n-capas como: capa de presentación, capa de aplicación, capa de servicios de negocios y capa de persistencia (Datos), en la arquitectura multicapa sugerida incorpora las mejores prácticas de arquitecturas y tecnologías en la industria. Esta arquitectura en su capa de presentación está basada en un Framework como Struts, La arquitectura Struts establece una implementación del estándar MVC (Model-View-Controller), esta arquitectura incluye un atractivo número de estándares basados esenciales para la industria del software. Además introduce Patrones de diseño del conector J2EE en la capa de servicios de negocios (Session, ServiceLocator, JMS, etc.).

Los patrones de diseño son un nuevo estándar de desarrollo estos están diseñados para producir reusabilidad al nivel de negocio. Esta posee muchísimas ventajas como: que es soportada en múltiples sistemas operativos, ya que es ejecutada por la máquina virtual de java, es un sistema que ocupa herramientas “open source” (soluciones libres) enfocado bajo la licencia Common Public License, la cual es admitida por la Open Source Initiative (OSI) como licencia libre que se pueden bajar de la Web, teniendo una ventaja comparativa superior ya que disminuye los costos significativamente de implementación.

¹ Ver Anexos De Arquitectura J2ee

Esta plataforma ya tiene 17 años de vida y es usada por las mejores empresas de software en el mundo, entre otras ventajas la plataforma j2ee está controlada por el JCP, Un organismo de control formado por más de 500 empresas en el mundo informático (SUN, IBM, ORACLE, SAP, HP, AOL, etc) lo que garantiza la evolución constante del sistema en si con los mejores estándares del mundo. También posee una herramienta para comprobar la compatibilidad con el estándar asegurando implementaciones del estándar compatibles entre sí que aumenta su portabilidad.

PROPUESTA DE SOLUCIÓN 2

La propuesta 2 es el lenguaje PHP pero la versión de la misma compañía (MICROSOFT) que es .NET.

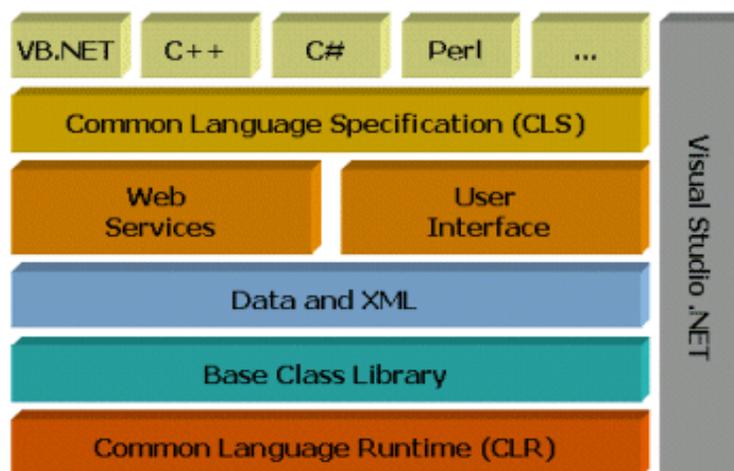
Este lenguaje está construido con scripts como otros lenguajes similares como visual.net, colfusion, etc. En primer lugar, ASP.NET es un lenguaje relativamente nuevo, diseñado desde cero con el fin único de diseñar aplicaciones Web. Esto quiere decir que las tareas más habituales en el desarrollo de estas aplicaciones, pueden hacerse con .NET de forma fácil, rápida y efectiva.

En segundo lugar, .NET es un lenguaje multiplataforma, y no propietario. Un script ASP normal puede ejecutarse sin cambiar ni una sola línea de código en cualquier servidor que interprete ASP, es decir, en servidores Windows, Linux, etc., ASP.NET es un lenguaje propietario de Microsoft.

ASP.NET combina excelentemente con otras inmejorables herramientas, como son el servidor apache y la base de datos mysql (o msql, o postgres), MICROSOFT.NET trabaja con solo un tipo de tecnología en el acceso a las bases de datos como ADO.NET, posee una escalabilidad balanceada es decir, que permite dar un mejor servicio con la misma integridad a más usuarios que requieran el servicio, pero solo puede ser soportada por maquinas basadas en WIN 32, esto quiere decir solo maquinas que posean sistemas operativos de Microsoft como WINDOWS 2000, XP, 2003, etc.

Una de las principales ventajas es seguir con la misma plataforma que es la última tecnología de MICROSOFT.NET pero el costo del estándar es elevado ya que el precio de las licencias por implementación, además que este estándar ha sido creado solo para soportar muchos lenguajes hechos solo por Microsoft como: Visual basic.net, Coldfusion, C++, c # etc.

ARQUITECTURA DE LA PROPUESTA 2 (.NET)



CAPITULO 3

FACTIBILIDAD

FACTIBILIDAD LEGAL

Desde el plano jurídico la empresa electroandina cuenta con todas las disposiciones legales para implementar su sitio de la unidad de negocio SISTEMA GESTION DOCUMENTAL, que por lo demás

ya lleva varios años en la Web, esto significa que cuenta con la reglamentación del Departamento de Ciencias de la Computación de la Universidad de Chile que estableció, jurídicamente mediante un contrato de prestación de servicios que sólo genera "derechos personales" -o créditos- para los cientos de miles de solicitantes del registro de un nombre de dominio con extensión "punto Cl.". Quedando el sitio de la electroandina como www.electroandina.cl, este sistema regula en caso de conflictos de 2 solicitantes de un mismo nombre de dominio, hace una mediación y regula que todos los nombres de las empresas que requieran tener un sitio en la Web sean diferentes generando un arbitraje para cualquiera de las partes según estipulen las cláusulas de NIC chile.cl. Quedando claro que el marco legal del sitio del Sistema de Gestión Documental, está en el marco legal de la legislación chilena.

En cuanto a la protección de los datos personales de los clientes de Electroandina está protegido por lo que dice la ley 19.628 sobre protección de datos personales y vigente desde el año 1998, que garantiza y posibilita un equilibrio adecuado entre el Derecho a la Privacidad y el Derecho a la Información de las personas naturales, y concretamente, no permite que los titulares de los datos y/o antecedentes que se procesan computacionalmente

controlen y auto determinen el uso con fines de lucro de sus atributos, notas, datos o antecedentes personales.

Esta ley asegura un buen uso de los datos de todos los usuarios que integren al sitio de Electroandina en cualquier tipo de transacción de se requiera realizar ya que se encuentran protegidos en las bases de datos de Electroandina.

FACTIBILIDAD TECNICA

En esta etapa se consideran las propuestas para la solución del problema en el aspecto tecnológico, evaluar la elección de un equipo de trabajo, la orientación técnica de los distintos componentes, la variedad de hardware que se necesita. Además de un diseño o arquitectura de construcción de cada dispositivo para la solución, y es en función de esas pautas que se determinan el tamaño, capacidad, velocidad, etc., de cada componente del sistema.

Esta etapa también implica analizar la mejor alternativa para el procesamiento de las aplicaciones que pensamos desarrollar para la solución de nuestra propuesta para el sistema de gestión documental y su integración con SAP, por conceptos en la resolución de las alternativas, ambas son tecnologías orientadas al objeto aquí se detallara cada una de ellas y su funcionalidad para satisfacer la lógica del negocio:

PROPUESTA DE SOLUCIÓN TÉCNICA 1

La plataforma J2EE¹ define un mecanismo estándar basado en el concepto portabilidad de Java. Pretende facilitar y disminuir el costo de desarrollo del software en múltiples capas que permitan adaptarse a los requerimientos actuales del futuro sistema de gestión documental, Fomenta la separación en capas y el desarrollo especializado de componentes. Disminuye el código necesario para desarrollar los sistemas a través de: servicios ya provistos y APIs que implementan las funcionalidades normalmente requeridas por la plataforma. J2EE, maneja una

¹ Ver Anexos De Arquitectura J2ee

arquitectura que actúa con distintos patrones de diseño en cada capa, uno de estos patrones es MVC¹.

La arquitectura MVC (*Model/View/Controller*) fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, la Vista y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

Requisitos Técnicos

- ◆ Sistema operativo Linux Redhat Enterprise Linux 6, como plataforma de servidor con JRE1.7.x
- ◆ Sistema Operativo Linux Redhat Enterprise Linux 6, como plataforma de trabajo con JSDK 1.7.x.
- ◆ Se deberá contar un servidor Web de aplicación JBoss y un servidor Web TOMCAT.
- ◆ Base de Datos ORACLE 11g

Requisito Equipo de Desarrollo

- ◆ 1 Jefe de proyecto
- ◆ 1 Analista de Sistemas

¹ Ver Anexos Capa De Control De Peticiones

- ◆ 1 Desarrollador

Requisito Hardware Minimo

Electroandina contiene todos los requerimientos para el servidor, servidor Web y todo el hardware que se utilizaría para implementar el Sistema gestión documental.

Como ya se ha mencionado las propiedades y las características de la plataforma a usar, se determinó que el servidor debería tener las siguientes características mínimas:

- ◆ Procesador INTEL XEON E5-2603 v3 1.6 GHZ QUAD CORE 15MB Cache
- ◆ Memoria RAM 8GB LRDIMM DDR4 1600 MHZ
- ◆ 2 TB 7200 RPM SATA Disco Duro
- ◆ Unidad DVD Super Multi Integrado
- ◆ Logitech System Mouse (Gris)

- ◆ Teclado Estándar para Windows
- ◆ Tower 4.5 U

PROPUESTA DE SOLUCIÓN TÉCNICA 2

.NET es una plataforma que se beneficia de un framework que da soporte a la aplicación con aspectos como balance de carga, transacción. Se trata de un conjunto de herramientas de un único proveedor (MICROSOFT), no de un estándar en cual se pueden implantar múltiples proveedores, tiene una sintaxis moderna y productiva con una tecnología multipropósito que puede ser utilizada para aplicaciones Web o aplicaciones con interfaz de Windows.

.NET ofrece independencia de lenguaje e interoperabilidad entre lenguajes. Existen ya más de 20 compiladores (ADA, RPG, COBOL, Fortran, Eiffel, PERL, Component Pascal, C#, J#, Visual.NET, Prolog). El código (sin importar el lenguaje en que esté) es traducido al MS Intermediate Language (MSIL o IL) - analogía con Java bytecode. La traducción del código IL a

lenguaje de máquina es hecha por el Common Language RunTime (CLR) - analogía con JRE.

Es necesario tener instalado el .NET Framework para poder correr las aplicaciones desarrolladas para ésta tecnología, tanto si es sobre plataforma Windows (Win 7 a 8) u otra. Existen prototipos de implementaciones del Framework para Linux y FreeBSD (Proyecto Mono). Tiene 2 componentes principales: el CLR y la .NET Framework class library.

Provee un entorno consistente de programación orientada a objetos donde el código objeto pueda ser:

- ◆ Almacenado y ejecutado localmente,
- ◆ Ejecutado localmente pero distribuido en Internet o Ejecutado remotamente.
- ◆ Ofrecer una experiencia consistente al desarrollador sin importar si está implementando aplicaciones para Windows o el Web.

Provee un entorno de ejecución de código que:

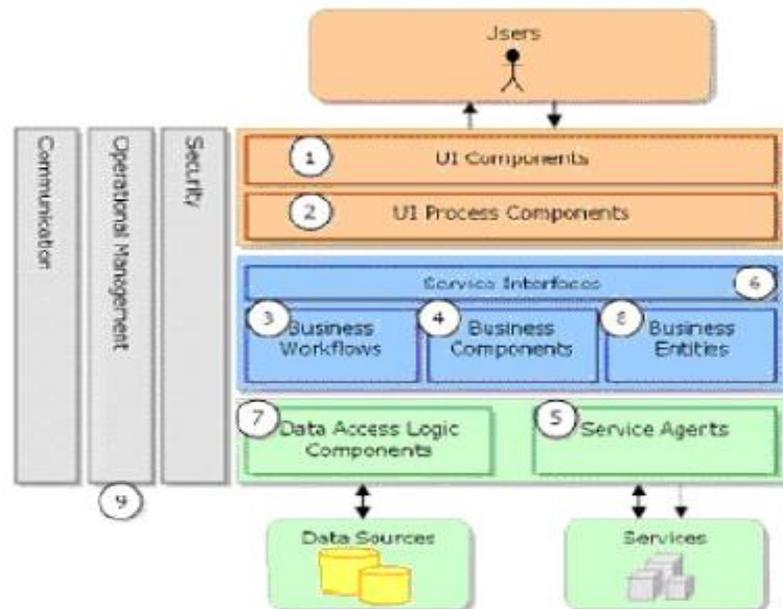
- ◆ Minimice el deployment (desarrollo) de los programas y resuelva los conflictos de versión.
- ◆ Garantiza la ejecución segura del código, incluyendo el código generado por desconocidos o terceras partes.
- ◆ Elimina los problemas de performance (rendimiento) en entornos con lenguaje batch o interpretado.

Dentro de las capas que trabaja ASP.NET están:

- ◆ A nivel de la capa de presentación construye toda la comunicación sobre estándares de la industria para asegurar que el código basado en .NET Framework sea integrable a otro código. Es una implementación completamente nueva de .NET, escrita desde cero en C#.
- ◆ Las aplicaciones son compiladas en el servidor, y las páginas son generadas en HTML específicamente para el browser que hizo la invocación.
- ◆ Tiene un conjunto de herramientas completa y un IDE común para diseño (VisualStudio.Net).
- ◆ La .NET Framework class library, la mensajería, y las soluciones de acceso a datos son accesibles completamente desde el Web en forma transparente.

ARQUITECTURA

.NET



A nivel de la capa de negocio consta de componentes gestionados por .NET (managed Objects), se conecta a bases de

datos usando Active Data Objects (ADO .NET) y a sistemas existentes usando servicios proporcionados por MICROSOFT HOST INTEGRATION SERVER 2000, como el COM Transaction Integrador (COM TI).

Requisito Hardware

Se necesita servidor donde se almacenará la base de datos de los usuarios que ocuparán el sistema.

Requisitos Técnicos

- ◆ Sistema Operativo Windows 2012 Server con plataforma de Servidor.
- ◆ Sistema Operativo Windows 8 profesional, como plataforma de trabajo.
- ◆ Se deberá contar un servidor Web de aplicación utilizando el Internet Information Server (IIS) versión 8.0.
- ◆ Base de Datos SQL Server 2012
- ◆ Licencia Microsoft .net.

Requisito Equipo de Desarrollo

El equipo humano para desarrollar esta alternativa, para la tecnología .NET de Microsoft, deberá ser estar capacitados para trabajar con esta plataforma

- ◆ 1 Jefe de proyecto
- ◆ 1 Analista de Sistemas
- ◆ 2 Desarrolladores

Requisito Hardware Minimo

El Sistema de Gestión Documental posee toda los requerimientos de servidor con de hardware necesario para implementar el sistema.

- ◆ Procesador Intel® CORE 2 QUAD Q6600
- ◆ Memoria RAM 8GB LRDIMM DDR4 1600 MHZ
- ◆ 1 x 2 TB 7200 RPM SATA Disco Duro
- ◆ Unidad DVD Super Multi Integrado
- ◆ Genius System Mouse (Gris)
- ◆ Teclado Estándar para Windows
- ◆ Tower 4.5 U

FACTIBILIDAD OPERACIONAL

Electroandina ya tiene las bases para tomar la decisión de desarrollar el proyecto de gestión documental para su área de seguridad y salud ocupacional, ya que por todas las razones ya expuestas en el estudio de esta propuesta tiene una directriz clara de integración con SAP para la misma expansión de la empresa a mercados futuros y la misma integración para aumentar la comunicación, ya que el proyecto Gestión documental tiene múltiples etapas. Electroandina ha dado la mayor información para sus usuarios, llámese subgerentes, distintos jefes de sucursales, usuarios internos y con los distintos perfiles que se han dado para el sistema de gestión documental, mediante los analistas a cargo del proyecto por medio de la elaboración de una encuesta en la cual aparecen diversas consultas tomándolas opiniones de estos mismos, tanto en la toma de requerimientos como en la decisión de la solución del problema, Electroandina por su parte ha decidido apoyar toda la documentación que han dado las distintas soluciones ya que si bien son propuestas distintas, ambas propuestas son orientadas a la programación orientada al objeto, tanto la propuesta de J2EE¹² como la alternativa .NET. Los usuarios por su parte colaboraron activamente en la toma de requerimientos con las distintas encuestas tipo que se les proporciono, logrando captar muchas necesidades en los

¹² Ver Anexos De Arquitectura J2ee

distintos niveles tanto operacional como niveles netamente técnicos.

Dentro de los procesos que se utilizan actualmente en la empresa se hace imprescindible la creación de un sistema de gestión documental para el área de seguridad y salud ocupacional, en todo lo que significa la demora de los procesos de asignar actividades, para los distintos usuarios jefes de sucursal de la empresa y también los usuarios externos. con la implementación del sistema nuevo se lograr solucionar los problemas de tiempo de respuesta, la definición de distintos perfiles, la identificación de los distintos actores, sistemas, subsistemas ,componentes, etc. y algunos conflictos de roles de algunos usuarios que interactúan con los procesos, para tomar todas estas necesidades de SISTEMA GESTION DOCUMENTAL, consultores DOMO CHILE le propina un fuerte énfasis en la etapa de análisis del problema generando manuales de usuario bastante amigables con ejemplos claros mostrando las distintas ayudas y los posibles caminos para aclarar cualquier inquietud en el manejo del futuro sistema, a su vez elabora toda la documentación técnica para los distintos administradores del sistema de gestión documental y todo el personal de soporte llámese programadores o analistas propios de Electroandina.

DOMO CHILE por su parte tiene un plan elaborado de capacitación que requiere el desarrollo del proyecto Gestión

documental con sus distintas áreas de outsourcing, generando equipos de trabajo en los distintos proyectos ya realizados por la empresa DOMO CHILE. Estos equipos toman la decisión de encabezar grupos de trabajo en toda la marcha blanca del proyecto, y todo el tiempo que requiera capacitar firmemente a los distintos tipos de usuarios.

Como es de esperar en cualquier tipo de organización hay usuarios que no están contentos con el cambio de un sistema para eso DOMO CHILE tiene distintos planes y capacitaciones básicas con el personal que trabaja en el proyecto y muchos niveles de asesorías. Si se toma en cuenta que se cambiara la forma de trabajo a una plataforma Web el cambio nos será tan radical, ya sólo se cambiaran algunas interfaces, pero el proceso seguirá siendo el mismo y agregación de otros módulos más automatizados pero el cambio total se observara en la implementación de la lógica de negocio y arquitectura, por lo demás muchos cambios que a la vista de algunos usuarios son transparentes.

FACTIBILIDAD ECONOMICA

Esta etapa se refiere a los recursos económicos y financieros necesarios para desarrollar o llevar a cabo las actividades o procesos y/o para obtener los recursos básicos que deben considerarse son el Costo Operacional y Costo de Inversión para el proyecto SISTEMA GESTION DOCUMENTAL.

Generalmente la factibilidad económica es el elemento más importante ya que a través de él se solventan las demás carencias de otros recursos, es lo más difícil de conseguir y requiere de actividades adicionales cuando no se posee.

La factibilidad económica requiere ser presentada con todos los posibles beneficios para la unidad de negocio de Electroandina, pero sin descuidar ninguno de los elementos necesarios para que el proyecto SISTEMA GESTION DOCUMENTAL funcione.

Este análisis económico implica realizar previsiones de los gastos e ingresos del producto en los distintos periodos de tiempo del futuro. El análisis de los ingresos supone realizar previsiones de los costos en forma permanente y los costos que se realizan una sola vez. El análisis de los costos implica estudiar todos los componentes de estos costos.

Una vez estimados los beneficios previstos para los distintos años y las inversiones necesarias podemos calcular la rentabilidad esperada del SISTEMA GESTION DOCUMENTAL. Hay que mencionar que Electroandina posee todo el hardware necesario para implantar las 2 propuestas.

ALTERNATIVA DE SOLUCIÓN ECONÓMICA 1

En esta alternativa que es J2EE, no incluye el costo de la licencia del software y la base de datos ya que son "Open Source" (gratis), Sin embargo existe otros gastos que se explica a continuación.

Detalle de costos operacionales: estos costos operacionales son los de los profesionales a cargo del proyecto (personal de DOMO CHILE) anualmente.

Alternativa 1

Remuneraciones	MES	AÑO
Programador 1	800.000	9.600.000
Analista 1	1.100.000	13.200.000
Jefe de Proyectos	1.500.000	18.000.000
Total año 1	3.400.000	40.800.000

Remuneraciones	MES	AÑO
Programador 1	800.000	9.600.000
Analista 1	1.100.000	13.200.000
Jefe de Proyectos	1.500.000	18.000.000
Total año 2	3.400.000	40.800.000

Remuneraciones	MES	AÑO
Programador 1	800.000	9.600.000
Analista 1	1.100.000	13.200.000
Jefe de Proyectos	1.500.000	18.000.000

Total año 3	3.400.000	40.800.000

Remuneraciones	MES	AÑO
Programador 1	800.000	9.600.000
Analista 1	1.100.000	13.200.000
Jefe de Proyectos	1.500.000	18.000.000
Total año 4	3.400.000	40.800.000

Remuneraciones	MES	AÑO
Programador 1	800.000	9.600.000
Analista 1	1.100.000	13.200.000
Jefe de Proyectos	1.500.000	18.000.000
Total año 5	3.400.000	40.800.000

Estos costos operacionales son los que se harán por año que incluye el sueldo de los profesionales involucrados en el proyecto con una duración de 5 años y las demás tablas son los costos de capacitación de los usuarios, mantenimiento del sistema y servicios de energía.

Capacitación	700.000

Gastos Mantención	MES	AÑO
Mantención de Sistema	800.000	9.600.000
Total	800.000	9.600.000

Gastos Generales	MES	AÑO
Agua, Luz, Teléfono etc.	500.000	6.000.000
Total	500.000	6.000.000

Activos		Depreciación Año	Deprec. Acum.	Valor Residual
Muebles y Equipos	700	20.000.000	1.000.000	10.000.000

FLUJO DE CAJA DE PROPUESTA 1

	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
	0	1	2	3	4	5
Ingresos por Etapa N° 1		4.000.000	4.500.000	5.000.000	5.500.000	6.000.000
Costo por Servicio Otorgado		- 40.800.000	- 40.800.000	- 40.800.000	- 40.800.000	- 40.800.000
Margen Bruto		- 36.800.000	- 36.300.000	- 35.800.000	- 35.300.000	- 34.800.000
Gastos por Remuneración		- 40.800.000	- 40.800.000	- 40.800.000	- 40.800.000	- 40.800.000
Gastos de Mantención		-9.600.000	-9.600.000	-9.600.000	-9.600.000	-9.600.000
Gastos Generales		6.000.000	6.000.000	6.000.000	6.000.000	6.000.000
Depreciación		-2.000.000	-2.000.000	-2.000.000	-2.000.000	-2.000.000
Gasto de Licencia		0				
Utilidad antes de Impuesto		9.600.000	10.100.000	10.600.000	11.100.000	11.600.000
Impuesto. (19%)		-1.824.000	-1.919.000	-2.014.000	-2.109.000	-2.204.000
Utilidad después de Impuesto		7.776.000	8.181.000	8.586.000	8.991.000	9.396.000
Préstamo (+)Depreciación		1.000.000	1.000.000	1.000.000	1.000.000	1.000.000
Inversión Inicial	- 20.960.000					10.000.000
Capital de Trabajo	-3.600.000					3.600.000

Flujo de Caja	- 24.560.000	8.776.000	9.181.000	9.586.000	9.991.000	23.996.000
----------------------	-------------------------	------------------	------------------	------------------	------------------	-------------------

(*) Valores residuales.

VAN Y EL TIR DE LA PROPUESTA 1

V.A.N.	(17.383.313)
T.I.R.	33%
TASA	12,00%

ALTERNATIVA DE SOLUCIÓN ECONÓMICA 2

Esta propuesta que es de .NET incluye un gasto adicional de licencias de Microsoft .NET además de otro programador que desarrolle en .NET que trabajara para DOMO CHILE. Se detallan a continuación los costos operacionales de esta alternativa tanto mensual como anualmente.

Alternativa 2

Remuneraciones	MES	AÑO
Programador 2	1.200.000	14.400.000
Analista 1	1.400.000	16.800.000
Jefe de Proyectos	1.600.000	19.200.000
Total año 1	4.200.000	50.400.000

Remuneraciones	MES	AÑO
Programador 2	1.200.000	14.400.000
Analista 1	1.400.000	16.800.000
Jefe de Proyectos	1.600.000	19.200.000
Total año 2	4.200.000	50.400.000

Remuneraciones	MES	AÑO
Programador 2	1.200.000	14.400.000
Analista 1	1.400.000	16.800.000
Jefe de Proyectos	1.600.000	19.200.000
Total año 3	4.200.000	50.400.000

Remuneraciones	MES	AÑO
Programador 2	1.200.000	14.400.000
Analista 1	1.400.000	16.800.000
Jefe de Proyectos	1.600.000	19.200.000
Total año 4	4.200.000	50.400.000

Remuneraciones	MES	AÑO
Programador 2	1.200.000	14.400.000
Analista 1	1.400.000	16.800.000
Jefe de Proyectos	1.600.000	19.200.000
Total año 5	4.200.000	50.400.000

Gasto de Licencia	960.000
--------------------------	----------------

Capacitación	700.000
---------------------	---------

Gastos Mantención	MES	AÑO
Mantención de Sistema	900.000	10.800.000
	900.000	10.800.000

Gastos Generales	MES	AÑO
Agua, Luz, Teléfono etc.	500.000	6.000.000
	500.000	6.000.000

Activos	700	Depreciación Año	Deprec. Acum	Valor Residual
Muebles y Equipos	20.000.000	2.000.000	10.000.000	10.000.000

FLUJO DE CAJA PROPUESTA 2

Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
-------	-------	-------	-------	-------	-------

	0	1	2	3	4	5
Ingresos por Etapa N° 1		4.000.000	4.500.000	5.000.000	5.500.000	6.000.000
Costo por Servicio Otorgado		- 50.800.000	- 50.800.000	- 50.800.000	- 50.800.000	- 50.800.000
Margen Bruto		- 46.800.000	- 46.300.000	- 45.800.000	- 45.300.000	- 44.800.000
Gastos por Remuneración		- 50.800.000	- 50.800.000	- 50.800.000	- 50.800.000	- 50.800.000
Gastos de Mantenición		- 10.600.000	- 10.600.000	- 10.600.000	- 10.600.000	- 10.600.000
Gastos Generales		6.000.000	6.000.000	6.000.000	6.000.000	6.000.000
Depreciación		-2.000.000	-2.000.000	-2.000.000	-2.000.000	-2.000.000
Gasto de Licencia		960.000				
Utilidad antes de Impuesto		9.640.000	11.100.000	11.600.000	12.100.000	12.600.000
Impuesto. (19%)		-1.831.600	-2.109.000	-2.204.000	-2.299.000	-2.394.000
Utilidad después de Impuesto		7.808.400	8.991.000	9.396.000	9.801.000	10.206.000
Préstamo (+)Depreciación		2.000.000	2.000.000	2.000.000	2.000.000	2.000.000
Inversión Inicial	- 20.960.000					10000000
Capital de Trabajo	-3.600.000					3600000
Flujo de Caja	- 24.560.000	9.808.400	10.991.000	11.396.000	11.801.000	25.806.000

(*) Valores residuales

VAN Y TIR

V.A.N.	23.213.672
T.I.R.	40%
Tasa	12,00%

ELECCION DE ALTERNATIVA

Tomando en cuenta los estudios de factibilidad se decidió lo siguiente:

Alternativa 1

Considerando la factibilidad Técnica la alternativa J2ee¹ brinda muchas ventajas como ser un lenguaje que utiliza componentes que son reutilizables en el tiempo, conceptos de portabilidad esto quiere decir, que para la implementación de la segunda etapa todos los módulos que se quieran construir serán agregados sin dificultad a la plataforma ya construida lo que no incurriría en nuevos costos de software o de hardware para la empresa.

Esto resulta mucho más conveniente ya que j2ee posee herramientas open source (aplicaciones que se bajan directamente de la Web) lo que resulta un ahorro importante ya que por tener múltiples proveedores y no pagar una licencia como único fabricante, la escalabilidad y las múltiples ventajas en cuanto a arquitectura, capas, separación de componentes.

Se deduce también que j2ee es una plataforma empresarial que lleva 10 años en el mercado siendo una plataforma alimentada bajo las principales compañías del mundo como SUN, BORLAND, IBM, etc.

Respecto a la factibilidad económica los resultados de los cálculos y comparando el Van y el TIR, este último arroja un resultado más bajo, por lo tanto más cercano a cero, lo que concluye que esta alternativa es más viable que la segunda, por conclusión la alternativa elegida.

Alternativa 2

¹ Ver Anexos De Arquitectura J2ee

En cuanto a la factibilidad técnica, la plataforma .NET no tiene muchos años en el mercado, es una plataforma relativamente nueva, si bien posee características similares a J2ee¹, posee algunas desventajas en cuanto seguridad frente a virus informáticos, ya que si se piensa en la falta de encapsulamiento de .NET , se provocarían mayores costos de software y hardware para poder reemplazar los componentes dañados por estos virus, y por sobre todo los datos que se manejan de los usuarios los cuales podrían generar errores graves de consistencia en las bases de datos.

Si bien no es una alternativa mala implica asumir un costo mayor por cuanto se tiene que pagar una licencia a Microsoft por ser el único fabricante de la plataforma .NET

Elegir esta plataforma significa contratar el primer año de implementación a un desarrollador externo a DOMO CHILE que se especialice en .NET lo que significaría un costo doble lo que se ve reflejado claramente en el flujo de caja de la propuesta 2, aumentando claramente el Van y el TIR. Por lo tanto este último índice se dispara enormemente siendo menos viable para el proyecto SISTEMA GESTION DOCUMENTAL.

CAPITULO 4

DESARROLLO DEL TRABAJO

¹ Ver Anexos Arquitectura J2ee

INTRODUCCION

La siguiente etapa corresponde mostrar el diseño lógico de la alternativa elegida que es j2ee¹, para efectos del análisis se utilizara la herramienta UML para el modelado de los distintos escenarios de la toma de requisitos o casos de usos.

UML es un Lenguaje de Modelado Unificado basado en una notación gráfica la cual permite: especificar, construir, visualizar y documentar los objetos de un sistema programado.

¹ Ver Anexos De Arquitectura J2ee

Este lenguaje permite al analista definir claramente los procesos de negocio para el posterior diseño del sistema, UML es una poderosa herramienta que permite recoger a cabalidad los requerimientos funcionales del sistema, con una tendencia definitiva a la programación orientada a los objetos con una notación estándar que permite mostrar distintos escenarios de los procesos para modelar su uso completamente, identificando cuales van a ser los actores directos del SISTEMA GESTION DOCUMENTAL con sus correspondientes casos de uso, hasta llegar a una arquitectura funcional donde claramente se identificaran los distintos requerimientos no funcionales, además de las distintas capas y componentes en las que se encuentran divididos los procesos de la unidad de negocio SISTEMA GESTION DOCUMENTAL, se hará una hincapié en la arquitectura funcional del sistema que permite identificar una serie de componentes que interactúan en la unidad de negocio, y la separación de las distintas capas para una mejor portabilidad del sistema y darle una clara estructura , fundamental para el diseño de la lógica del negocio¹ del SISTEMA GESTION DOCUMENTAL.

SISTEMA GESTION DOCUMENTAL

Para lograr tomar los requerimientos y poder modelar estos mismos se necesita del UML, el cual modela al SISTEMA GESTION DOCUMENTAL de Electroandina mediante el uso de objetos que forman parte de él así como, las relaciones estáticas o dinámicas que existen entre ellos.

¹ Ver Anexos De Capa De Lógica Del Negocio

El futuro SISTEMA GESTION DOCUMENTAL será un software encargado de prestar distintos servicios a sus usuarios, este va a cumplir una labor de información y apoyo para la toma de decisiones de Electroandina. Su construcción hace necesario un análisis muy fuerte en la toma de los requerimientos funcionales del sistema.

Para poder lograr este objetivo DOMO CHILE usa un lenguaje o estándar llamado UML (lenguaje de modelamiento unificado) este se utiliza en cualquier metodología de análisis y diseño orientado a objetos, para expresar los diseños de los sistemas, los cuales ayudan a encerrar el ámbito del problema y los procesos asociados a este. Una de las propiedades fundamentales que posee UML son los diagramas de casos de uso, por medio de estos diagramas se identifican claramente los requerimientos funcionales del sistema.

Un diagrama de Casos de Uso¹ muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuario u otras aplicaciones).

Es una herramienta esencial para la captura de requerimientos y para la planificación y control de un proyecto interactivo lo cual se necesita para el proyecto SISTEMA GESTION DOCUMENTAL.

Cada caso de uso, es una operación completa desarrollada por los actores y por el proceso en un diálogo mutuo.

El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema.

ACTORES SISTEMA GESTION DOCUMENTAL

¹ Ver Anexos Casos De Uso

Ya explicada la metodología del UML y las herramientas que usa como los casos de uso y los actores, se necesita identificar estos actores en el Sistema de Gestión Documental y sus distintas funciones dentro de los casos de uso, se comenzara con listar los actores del SISTEMA GESTION DOCUMENTAL:

- ◆ **Subgerente Casa Matriz:** Este actor representa al usuario que realiza un monitoreo a la gestión de actividad asignada, esta actividad asignada como caso de uso, es la que se le delega al jefe de sucursal, el usuario subgerente supervisa en todo momento la gestión del jefe de sucursal, dependiendo de la actividad que esté realizando, y debe velar porque los indicadores de cumplimiento se ejecuten correctamente.
- ◆ **Jefe Sucursal:** Este actor está a cargo de gestionar las actividades asignadas es decir, asignar las distintas cartas Gantt mensuales de actividades al actor responsable de tarea, a su vez debe validar el cumplimiento, clasificando el estatus de estos registros mediante un mecanismo de alarmas que posee el sistema de gestión documental.
- ◆ **Responsable Tarea:** Este actor está a cargo de la ejecución de una actividad asignada por el actor jefe de sucursal, su rol es validar que el cumplimiento de la actividad de la carta Gantt, verificando en terreno que se realice dicha actividad, para después elaborar un registro que debe ingresar al sistema de gestión documental.

- ◆ **Administrador de Perfiles:** Este actor tiene como misión administrar los distintos perfiles de usuario, además de dar distintos privilegios para otros usuarios que necesiten tener acceso de lectura o visualizar los distintos módulos del sistema de gestión documental.

.

- ◆ **Jefe de Área:** Este actor tiene como misión gestionar la investigación de accidentes laborales, puede clasificar de acuerdo al listado de accidentes que el sistema entrega y puede asignar a un responsable de tarea sus acciones.

- ◆ **Usuario Auditor:** Este actor tiene como misión visualizar que las actas del comité paritario, estén ingresadas en el módulo gestión comité paritario y que las fechas de los acuerdos se cumplan, solo tiene un privilegio de lectura.
- ◆ **Usuario Comité:** Este actor tiene como misión ingresar los acuerdos del comité paritario y asignarle a un responsable de tarea, la ejecución de las actas en el módulo de gestión comité.
- ◆ **Auditor:** Este actor tiene como misión ingresar observaciones en el módulo auditoria de calidad, cuando visualiza el estado de acciones correctivas en el sistema, dependiendo del estado de las tareas.

CASOS DE USO DEL NEGOCIO

Para obtener una visión general de los diferentes procesos de negocio del SISTEMA GESTION DOCUMENTAL, Se construirá un diagrama de casos de uso donde cada caso de uso, representa un proceso de negocio y cada actor representa un rol.

Este diagrama permite mostrar los límites, contexto o entorno del SISTEMA GESTION DOCUMENTAL. Por tanto tan sólo se mostrarán actores del negocio correspondientes con sus roles dentro del sistema, de forma que quedaran graficado los procesos de negocio en los que sólo tomen parte roles del SISTEMA GESTION DOCUMENTAL.

Las funciones de estos actores serán definidas en un formato más expandido o caja negra la cual resume los escenarios donde interactúan con los distintos procesos (casos de uso) del modelado del negocio.

DIAGRAMA DE CASOS DE USO SISTEMA GESTION DOCUMENTAL

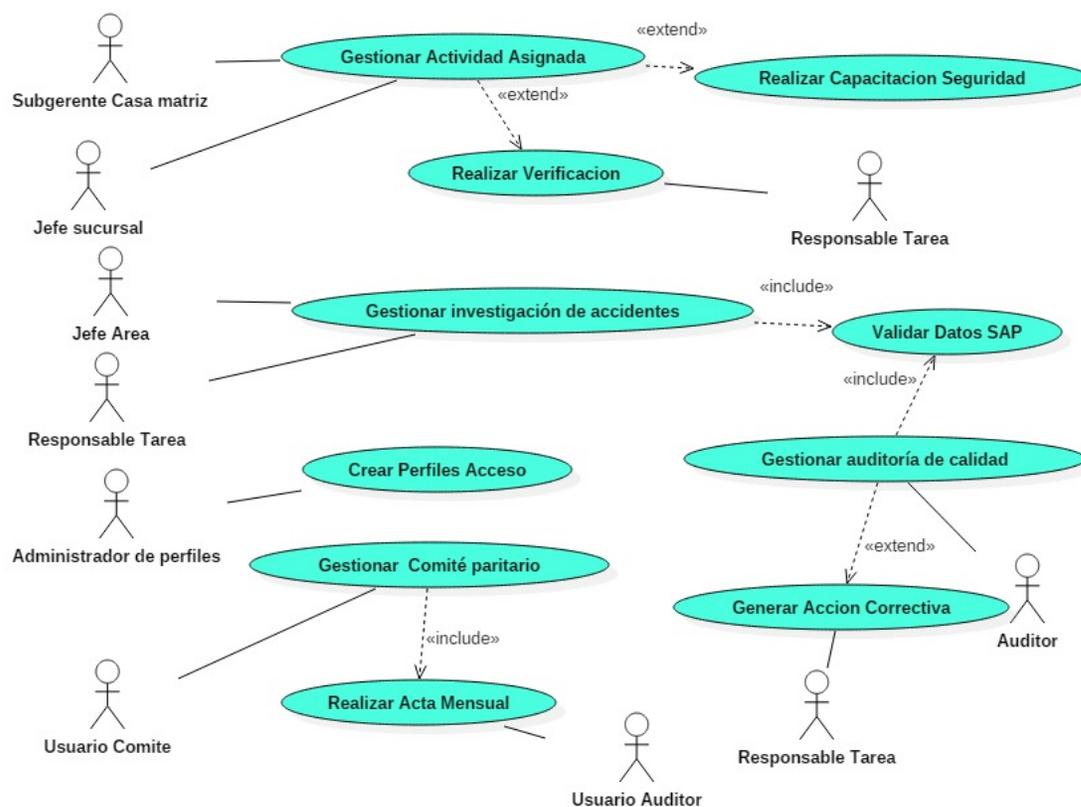
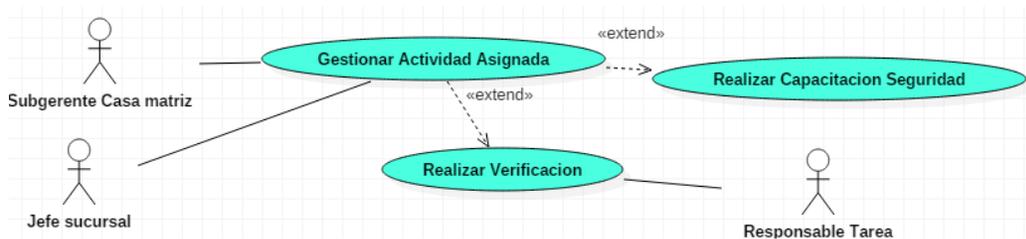


FIG. Se representan en el diagrama de casos de uso, todos los procesos involucrados en el negocio por medio de una elipse, el cual logra capturar cada requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema mediante un diálogo. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el SISTEMA GESTION DOCUMENTAL. Las relaciones como la “include” se manifiesta cuando un caso de uso contiene a otro, y el “extends” cuando el caso de uso es similar al otro o posee herencia (contiene funcionalidades).

Caso De Uso Gestionar Actividad Asignada



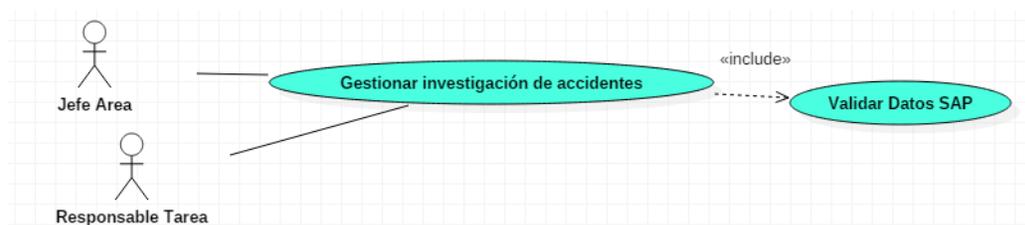
CASO DE USO 1	Gestionar Actividad Asignada
Objetivos en Contexto	Permitir al jefe de sucursal gestionar las actividades programadas para el mes en curso.
Ámbito y Nivel	Gestión de Actividad asignada
Precondiciones	El Jefe de sucursal debe estar validado como usuario responsable de programa de actividades y habilitado en el sistema.
Condiciones Éxito	El jefe de sucursal logra gestionar la actividad asignada generando un registro de la actividad.

Condición Fallo	El jefe de sucursal no logra procesar un registro de una actividad asignada.	
Actores Primarios, Secundarios	Jefe de sucursal , Responsable de tarea, Subgerente casa matriz,	
Iniciador	Escenario 1: El jefe de sucursal gestiona una actividad asignada Escenario 2: El responsable de tarea realiza una capacitación Escenario 3: El responsable de tarea realiza una verificación	
Descripción	Pasos	Acción
	Escenario 1	
1	El jefe de sucursal ingresa su usuario y clave	El sistema muestra un menú con las opciones del sistema por carta Gantt, que le corresponde realizar como actividad asignada y puede descargar el contenido de la carta Gantt.
2	El jefe de sucursal descarga la actividad asignada y pulsa clic sobre el link del nombre de la actividad.	El sistema descarga la actividad asignada en un formato pdf, para su impresión.
3	El jefe de sucursal realiza un comunicado del contenido de la carta Gantt a su personal	El sistema valida en SAP el listado de participantes a cargo del jefe de sucursal, y adjunta el documento, y enciende un icono en

	dando clic al botón enviar comunicado, a su vez selecciona el registro de los participantes y digitaliza el documento y pulsa adjuntar documento y pulsa guardar actividad.	verde, que significa que el registro se encuentra en un estado cargado.
	Escenario 2	
1	Actor responsable de tarea	Repite paso 1 escenario 1
2		Repite paso 2 escenario 1
3		Repite paso 3 escenario 1
	Escenario 3	
1	Actor responsable de tarea	Repite paso 1 escenario 2
2	Repite paso 2 escenario 2	Repite paso 2 escenario 2
3	El responsable de tarea con el documento impreso se dirige al área, que le designe la verificación y en terreno, valida el cumplimiento de	El sistema valida las observaciones levantadas por el responsable, envía el mensaje observaciones guardadas exitosamente.

	<p>acuerdo al requerimiento del listado. El responsable revisa este listado y digitaliza el documento con sus observaciones al sistema y pulsa guardar observaciones.</p>	
4	<p>El responsable ingresa la fecha de actividad y pulsa guardar.</p>	<p>El sistema muestra el listado resumen del documento. Indica el mensaje actividad guardada.</p>
5	<p>El responsable ingresa las acciones inmediatas por cada ítem con observaciones y pulsa guardar cambios.</p>	<p>El sistema pregunta a que usuario se le asignara las acciones ingresadas por el responsable de tarea.</p>
6	<p>El responsable de tarea selecciona el usuario que le asignara las acciones y adjunta el registro de verificación y pulsa adjuntar y grabar</p>	<p>El sistema enciende un icono en verde que significa que el registro se encuentra en un estado cargado</p>

Caso De Uso Gestionar Investigación de Accidentes



CASO DE USO 2	Gestionar Investigación de Accidentes
Objetivos en Contexto	Permitir al jefe de área, gestionar una investigación de accidente laboral.
Ámbito y Nivel	Gestión una investigación de accidente laboral.
Precondiciones	El Jefe de área, debe estar validado como usuario responsable de investigación de accidentes y habilitado en el sistema.
Condiciones Éxito	El jefe de área logra gestionar la investigación de un accidente
Condición Fallo	El jefe de área no realiza una investigación de un accidente laboral
Actores	Jefe de área, responsable de tarea

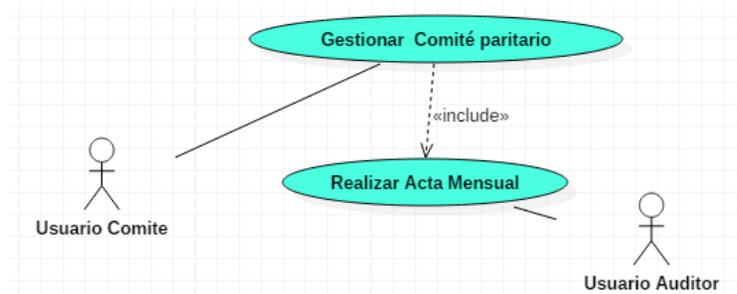
Primarios, Secundarios		
Iniciador	<p>Escenario 1: El jefe de área realiza una investigación de accidente laboral.</p> <p>Escenario 2: El responsable de tarea ejecuta la acción correctiva</p>	
DESCRIPCION	Pasos	Acción
	Escenario 1	
1	El jefe de área ingresa el Rut del trabajador accidentado	El sistema valida los datos ingresados, por el jefe de área contra el módulo SAP, y despliega los datos del accidentado.
2	El jefe de área ingresa la descripción del accidente sufrido por el trabajador, luego va seleccionando opciones predeterminadas de ocurrencias de accidentes y pulsa guardar y adjunta un documento de registro de investigación de accidente y pulsa adjuntar.	El sistema envía un mensaje indicando al usuario las acciones a seleccionar y envía el mensaje que el registro fue guardado exitosamente.

	El usuario selecciona las acciones correctivas y al usuario responsable de tarea para su ejecución y pulsa guardar cambios.	El sistema enciende un icono en verde que significa que el registro se encuentra en un estado cargado y envía el mensaje de éxito.

Escenario 2	Pasos	Acción
1	El responsable de tarea ingresa al sistema con su usuario y clave.	El sistema muestra de acuerdo su perfil la asignación de la acción correctiva del accidente, en este caso por ej. Una compra de herramienta que haya fallado.
2	El usuario adjunta una factura de compra y pulsa guardar cambios.	El sistema enciende un icono en verde que significa que el registro se encuentra en un estado cargado y envía un mensaje de éxito.
3	El usuario adjunta el registro del personal capacitado y pulsa guardar cambios.	El sistema enciende un icono en verde que significa que el registro se encuentra en un estado cargado y envía un mensaje de éxito.
	El usuario adjunta un listado de herramientas y	El sistema enciende un icono en verde que significa que el registro se encuentra

	pulsa guardar	en un estado cargado y
	cambios.	envía un mensaje de éxito.

Caso De Uso Gestionar Comité paritario



CASO DE USO 3	Gestionar Comité paritario	
Objetivos en Contexto	Permitir al usuario de comité gestionar las actas de reunión de comité paritario.	
Ámbito y Nivel	Gestión de Comité paritario.	
Precondiciones	El usuario de comité debe estar validado como usuario responsable de comité y habilitado en el sistema.	
Condiciones Éxito	El usuario de comité logra gestionar el comité paritario	
Condición Fallo	El usuario de comité no logra gestionar el comité paritario	
Actores Primarios, Secundarios	Usuario comité Usuario Auditor	
Iniciador	Escenario 1: El usuario de comité paritario gestiona un acta mensual. Escenario 2: El usuario auditor revisa las actas realizadas por el comité paritario	
DESCRIPCION	Pasos	Acción
	Escenario 1	
1	El usuario de	El sistema muestra el menú de

	comité ingresa su usuario y clave al sistema.	comité paritario
2	El usuario ingresa los acuerdos y selecciona al usuario responsable para su ejecución y pulsa guardar cambios.	El sistema envía un mensaje de si desea ingresar otro acuerdo, en caso de no indica al usuario y adjuntar documento y cerrar la sesión.
3	El usuario adjunta documento de acta y pulsa adjuntar.	El sistema enciende un icono en verde que significa que el registro se encuentra en un estado cargado y envía un mensaje de éxito.
Escenario 2		
1	El usuario auditor ingresa con un usuario genérico y clave al sistema asignado por el administrador de perfiles. Es solo permiso de lectura.	El sistema muestra un listado de las actas ingresadas, muestra el estado de los acuerdos, muestra los responsable de los acuerdos y las fechas acordadas de los comités.
2	El usuario auditor revisa el listado de contenidos y pulsa cerrar sesión.	El sistema indica sesión terminada.

Caso De Uso Gestionar Auditoria de calidad



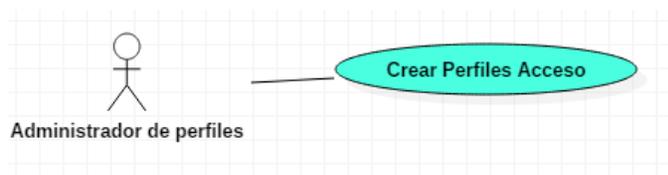
CASO DE USO 4	Gestionar Auditoria de calidad
Objetivos en Contexto	Permitir al auditor gestionar las observaciones de auditoria y asignar responsable de tarea.
Ámbito y Nivel	Gestión de Auditoria de calidad
Precondiciones	El auditor debe estar validado como usuario auditor y habilitado en el sistema.
Condiciones Éxito	El auditor realiza un levantamiento de observaciones de auditoria
Condición Fallo	El auditor no realiza un levantamiento observaciones de auditoria
Actores Primarios, Secundarios	Auditor Responsable de tarea
Iniciador	Escenario 1: El auditor levanta observaciones de auditoria Escenario 2: El responsable de tarea genera

	acción de correctiva de las observaciones	
DESCRIPCION	Pasos	Acción
	Escenario 1	
1	El auditor identifica en el sistema el área a auditar y selecciona la sucursal y selecciona el área y pulsa guardar cambios	El sistema envía el mensaje datos grabados y le envía el mensaje para que el usuario ingrese las observaciones.
2	El auditor ingresa las observaciones y asigna responsables de tarea y pulsa guardar cambios.	El sistema envía un mensaje de si desea ingresar otro observaciones, en caso de no, indica al usuario que adjunte documento y cerrar sesión.
3	El auditor adjunta documento o jpg y pulsa adjuntar.	El sistema notifica vía mail a los responsables de las observaciones para que estos ingresen al sistema y adjunten evidencias de las acciones tomadas
	Escenario 2	
1	El responsable de tarea ingresa	El sistema enciende un icono en verde que

	a la opción de cierre de acciones y adjunta documento de registro y pulsa adjuntar.	significa que el registro se encuentra en un estado cargado y envía mensaje de éxito.
2	El responsable de tarea valida que la información se adjuntó correctamente y pulsa el icono verde para visualizar el documento adjunto.	El sistema despliega el documento adjunto
Escenario 3		
1	El usuario auditor visualiza las acciones tomadas del proceso de auditoría y pulsa el botón aceptar.	El sistema envía un mensaje al responsable de la acción, que su gestión fue aceptada y cambia a verde el icono.
	El usuario auditor visualiza las acciones	El sistema envía un mensaje vía mail al responsable tarea que

	<p>tomadas del proceso de auditoría y pulsa el botón rechazar e ingresa un comentario de mejora de la acción tomada.</p>	<p>ejecuto la acción informando que su gestión fue rechazada y cambia a rojo el icono.</p>
--	--	--

Caso De Uso Crear Perfiles de Acceso



CASO DE USO	Crear Perfiles de Acceso
Objetivos en Contexto	Permitir al Administrador de perfiles crear o modificar un perfil de usuario, dar permiso a los distintos módulos del sistema.
Ámbito y Nivel	El administrador crea perfiles de usuario y

	asigna actividades, agrega permisos de lectura o escritura sobre campos predeterminados por usuario y sube documentos descargables y actualiza la versión los documentos e informa a los usuarios de cambios del sistema.	
Actores Primarios, Secundarios	Administrador de perfiles	
DESCRIPCION	Pasos	Acción
	Escenario 1	
1	El administrador ingresa con su cuenta y clave al sistema	El sistema muestra un menú con todas las pestanas de los módulos disponibles dentro del sistema
2	El administrador selecciona la opción crear nuevo usuario ingresa los datos y pulsa el botón guardar	El sistema valida contra el modulo SAP que contiene la bases de datos del personal e indica un mensaje sobre el estado del usuario.
3	El administrador cierra la sesión y realiza una prueba conectándose con el usuario recién creado	El sistema muestra el acceso a las pestanas de los módulos en los cuales tiene permiso
4	El administrador selecciona el usuario	El sistema guarda la información y envía un

	recién creado y chequea los permisos de los botones y eventos que puede realizar el usuario y pulsa el botón guardar	mensaje de éxito.
	Escenario 2	
1	Se repite paso 1 escenario 1	Se repite paso 1 escenario 1
2	El administrador selecciona la opción de modificar usuario seleccionando el usuario requerido y pulsa editar.	El sistema muestra el registro del usuario editado mostrando las cajas de texto en color gris
3	El administrador modifica un perfil del usuario y pulsa el botón guardar cambios	El sistema envía el mensaje de éxito.
4	El administrador realiza una prueba consultando por el usuario modificado	El sistema despliega la información del usuario modificada
	Escenario 3	
1	Se repite paso 1 escenario 2	Se repite paso 1 escenario 2

2	El administrador selecciona la opción de eliminar usuario seleccionando el usuario requerido y pulsa eliminar.	El sistema envía el mensaje de éxito.
3	El administrador realiza una prueba consultando por el usuario eliminado	El sistema despliega la información usuario eliminado.

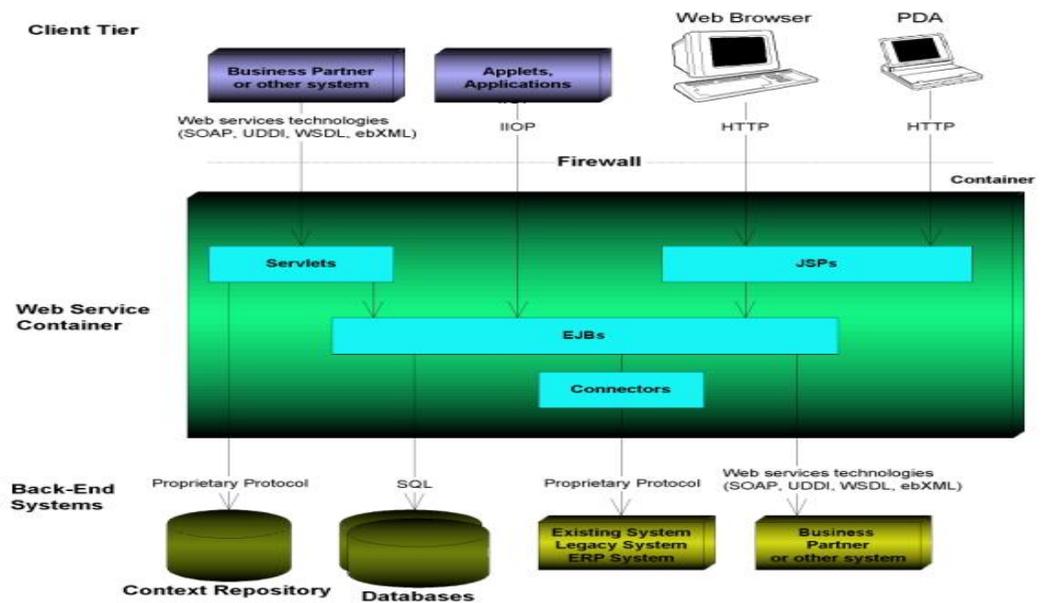
ARQUITECTURA SISTEMA GESTION DOCUMENTAL

Muchas de las ventajas de j2ee ya fueron enunciadas en la etapa factibilidad en la propuesta 1, pero en este capítulo se comenzará a ver en mayor profundidad y su aplicación a la unidad de negocio SISTEMA GESTION DOCUMENTAL. J2EE es una arquitectura de N-capas para implementar aplicaciones empresariales usando java como lenguaje, esta especificación:

- ◆ Provee un modelo de desarrollo de aplicaciones basado en componentes

- ◆ Usa un modelo para el desarrollo de aplicaciones distribuidas basado en múltiples capas o multitiered, la lógica de la Aplicación se divide en componentes que se instalan en equipos diferentes de acuerdo a la capa al que pertenecen.
- ◆ Está constituido por tecnologías estándares: Servlets, JSP, EJB, JDBC, JTA, JMS, JNDI, JCA, etc.
- ◆ Estas soluciones no están atadas a productos ni a interfaces de programación de ningún fabricante.

En la arquitectura j2ee todas las componentes se ejecutan en el entorno de un contenedor



ARQUITECTURA FUNCIONAL

La arquitectura del software en una estructura o estructuras del sistema y una descripción de los subsistemas, las cuales comprenden componentes del software, las propiedades visibles de estos componentes y las relaciones entre ellos, representados mediante una vista que muestran propiedades funcionales y no funcionales más relevantes.

Como ya se ha mencionado inicialmente se hizo un proceso de toma de requisitos (casos de uso) los cuales caen en la categoría de requerimientos funcionales del SISTEMA GESTION DOCUMENTAL y los requerimientos no funcionales, de los cuales se definirá un conjunto de metamodelos expresados en UML y siguiendo el estándar MDA (Domain Specific software Architecture) que se basa en el estudio del dominio para restringir el espacio del problema y de la solución.

Por lo tanto estos metamodelos nos permitirán recoger de forma más completa la aplicación para el SISTEMA GESTION DOCUMENTAL. Al trabajar en un espacio de solución restringido se facilita la tarea de identificar la funcionalidad de los componentes, considerando un nivel de abstracción adecuado que permitirá recuperar elementos de abstracción.

Como ya se ha indicado la arquitectura J2EE¹ se encarga de expresar los componentes lógicos (subsistemas, módulos o componentes de software) que participan en nuestra solución y la relación entre ellos, para conseguir esto se utilizaran patrones definidos en diferentes niveles o capas, además de describir cada capa de un módulo en particular detallando la participación de los componentes de software.

¹ Ver Anexos De Arquitectura J2ee

DIAGRAMA DE CAPAS SISTEMA DE GESTION DOCUMENTAL

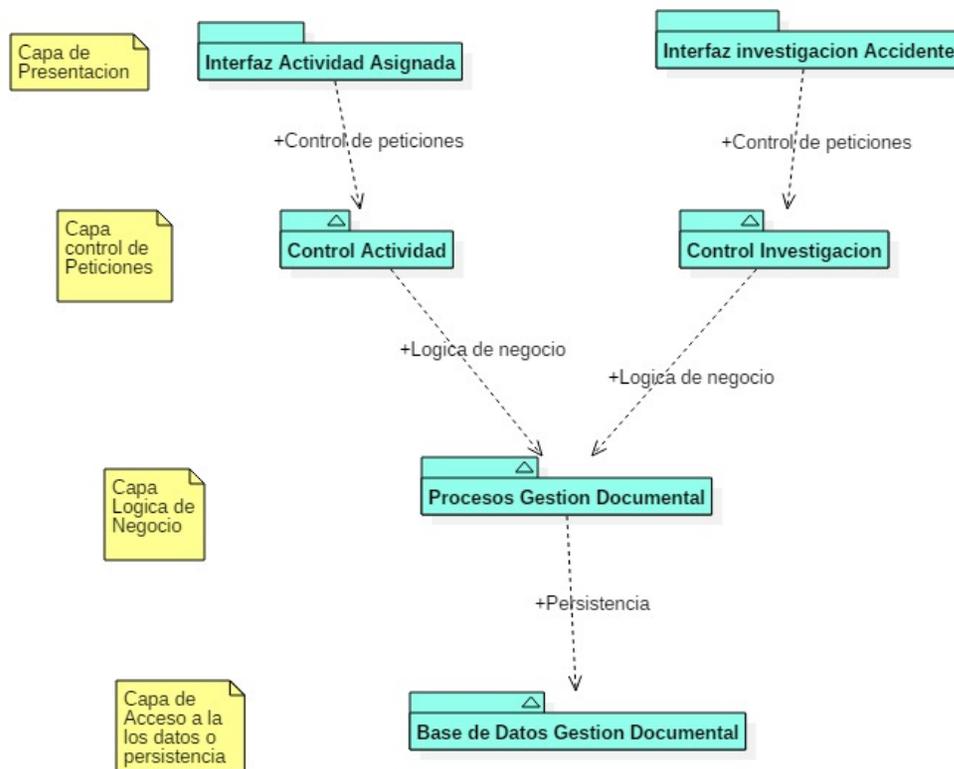


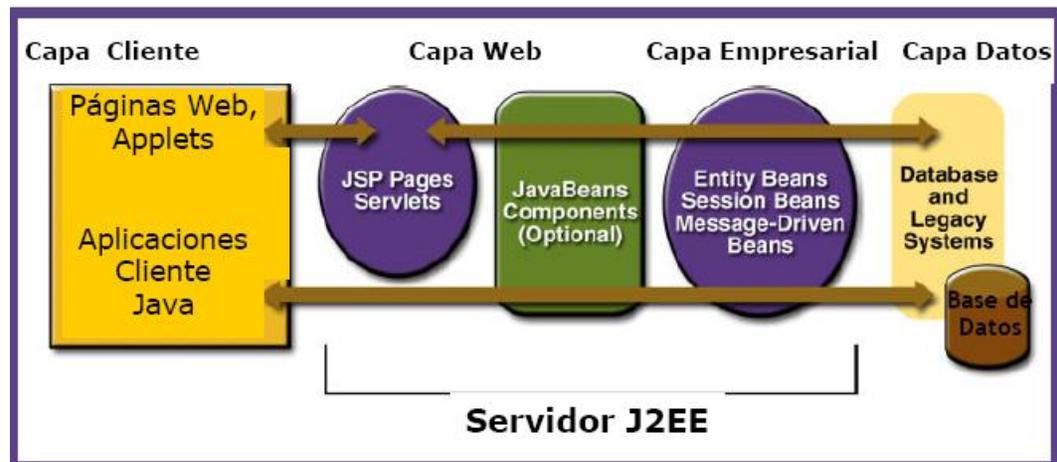
Fig. En este diagrama se aprecia claramente la división por capas del SISTEMA GESTION DOCUMENTAL pudiéndose apreciar los distintos niveles que esta abstracción posee, para la posterior descomposición en distintos componentes que actúan en los distintos niveles y subsistemas.

DIVISION DE LAS CAPAS SISTEMA GESTION DOCUMENTAL

Es importante señalar que para efectos de modularidad se tomó la arquitectura del cliente y administrador de solicitudes, de los cuales la arquitectura J2EE será aplicada y ejemplificada.

J2ee es una arquitectura basada en componentes y con el objetivo de la ocultación de la información, que tenga un bajo acoplamiento externo, una abstracción y alta cohesión interna. Para lograr captar las aplicaciones basadas en componentes es necesario dividir las capas en distintos niveles lógicos, cada nivel o capa está compuesto de uno o varios componentes. Con esta división se logra una abstracción lógica. Y una de las características más importantes es aislar la lógica del negocio. Al hacer una división de capas en el nivel medio aumenta la escalabilidad pudiendo reutilizar los recursos con una mejora sustancial en el rendimiento, además de mejorar la seguridad y la gestión de la aplicación. Para poder aplicar la arquitectura j2ee al Sistema de Gestión Documental la división de capas será la siguiente:

- ◆ Capa de presentación o cliente
- ◆ Capa de control de peticiones o capa Web
- ◆ Capa de la lógica del negocio
- ◆ Capa de persistencia o acceso a los datos



CAPA DE PRESENTACION O CAPA CLIENTE

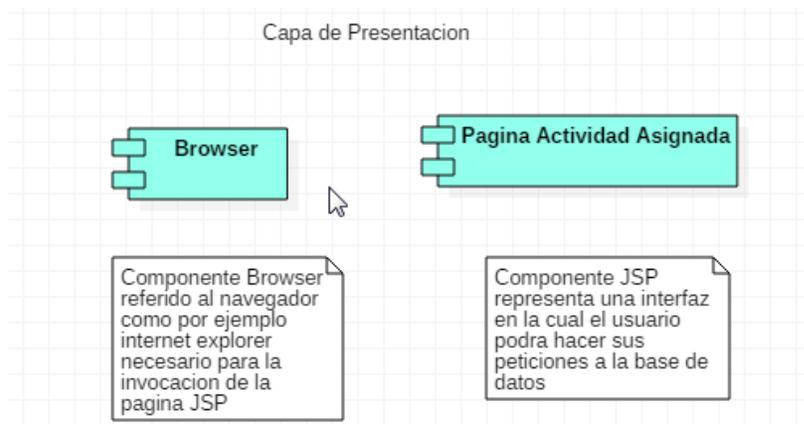
Capa encargada de dar la funcionalidad necesaria para la interacción entre el cliente y la aplicación del SISTEMA GESTION DOCUMENTAL. Esta capa posee una lógica de interacción que está encargada de mostrar la interfaz al usuario final, La capa de presentación está encargada de darle una presentación distribuida al usuario, Construye las pantallas de uso fácil que ayudan al usuario a realizar los trabajos que el usuario tengan previstos para realizar.

La capa de presentación genera las pantallas usando los datos proporcionados por la capa de control. La capa de presentación es responsable de:

- ◆ Diseño de la pantalla y lo amigable que puede ser al usuario.
- ◆ Específica la validación del campo de los Obligatorios, de los límites del tamaño de los campos, indica los mensajes al usuario de las validaciones que se implementen.
- ◆ Hace el trabajo de validar la autenticidad de la petición (no el contenido), identificando el servicio de negocio que satisface las peticiones y las invoca.
- ◆ De la misma manera, recibe el resultado de un servicio de negocio e identifica el componente de que le entrega el resultado y lo recibe a este nivel.

Es importante señalar que la capa de presentación no hace ninguna validación del negocio del SISTEMA GESTION DOCUMENTAL. Las acciones de una capa de presentación se restringen a la dirección compaginando la información, También toma la responsabilidad de autenticar las peticiones encaminadas con ella. Existen componentes que solo se pueden ejecutar en esta capa como:

- ◆ Un Objeto Web es un componente que es ejecutado en el cliente proporcionando la funcionalidad de la interfaz, normalmente se ejecuta en el browser mediante Pluggins como applets java, los componentes Flash, y los Actives.
- ◆ Un Explorador que puede ser cualquier explorador HTML estándar, con soporte o no pluggins flash, java capaz de interpretar las páginas de presentación, estilo y funcional. mostrárselas al usuario, aparte de aceptar y recibir Cookies.



Capa De Presentación Sistema gestión documental:

- ◆ **Página Actividad Asignada:** componente que contiene el contenido necesario para presentar la información de la actividad asignada al usuario, recibir la entrada de los datos del formulario para cotizar algún servicio o consultar el estado de su solicitud de servicio, validarlos si es necesario y además realizar las invocaciones al servidor (Html, Xml, página Flash) que puede contener referencias a otros componentes como estilo y funcional.

- ◆ **Página Investigación de Accidente:** componente que contiene el contenido necesario para presentar la información al jefe de Área, recibir la entrada de los datos de los formularios enviados por los accidentados, además de consultarlos y procesarlos, validarlos si es necesario y además realizar las invocaciones al servidor (HTML, XML, pagina FLASH) puede contener referencias a otros componentes cliente estilo y funcional.

- ◆ Página funcional: componente que mantiene funcionalidad para la interacción del cliente con la interfaz de la aplicación SISTEMA GESTION DOCUMENTAL. Dicha funcionalidad puede ser de validación, presentación, navegación, etc. (por Ej. Javascript, Action script). puede ser referenciado por el componente actividad o investigación de accidentes.
- ◆ Almacén: componente que mantiene un mecanismo de almacenamiento localizado en la capa de presentación, que le va permitir recuperar información entre las sesiones de los usuarios por Ej. : la Cookie (variable que almacena sesión de un usuario).

CAPA CONTROL DE PETICIONES SISTEMA GESTION DOCUMENTAL

En esta capa se implementa un Framework llamado **STRUTS**¹ que implementa el patrón de arquitectura MVC en Java, Un Framework es la extensión de un lenguaje mediante una o más jerarquías de clases que implementan una funcionalidad y que (opcionalmente) pueden ser extendidas.

El patrón de arquitectura MVC (Model-View-Controller) es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la View o vista (interfaz con el usuario u otro sistema) y el Controller (controlador de las peticiones de los usuarios: “si estoy aquí y me piden

¹ Ver Anexos De Capa Control De Peticiones

esto entonces hacer tal cosa, si sale bien mostrar esto y si no lo aquello otro”)

Las acciones que se ejecutarán sobre el modelo de negocio se implementan basándose en clases predefinidas por el Framework y siguiendo el patrón Business Delegate y el patrón Session Facade o Fachada. Que se encargaran de interactuar con los componentes de la capa del modelo de negocio.

A continuación se detalla la separación de capas siguiendo el patrón MVC.

- ◆ View (interfaz usuario o capa de presentación descrita anteriormente)
- ◆ Controller (capa control de las peticiones)
- ◆ Modelo(capa de lógica del negocio)

VISTA (View)

En la interfaz o vista está la Página servidora que es un componente que realiza el procesamiento de la página Web del cliente mediante Scripts ejecutados en el servidor Web o contenedor, que ejecuta el código la página y le devuelve el contenido al cliente. Dicho componente contiene la información referente a la presentación y/o lógica en el servidor y genera uno o más componentes de la interfaz de usuario (por Ej. JSP)

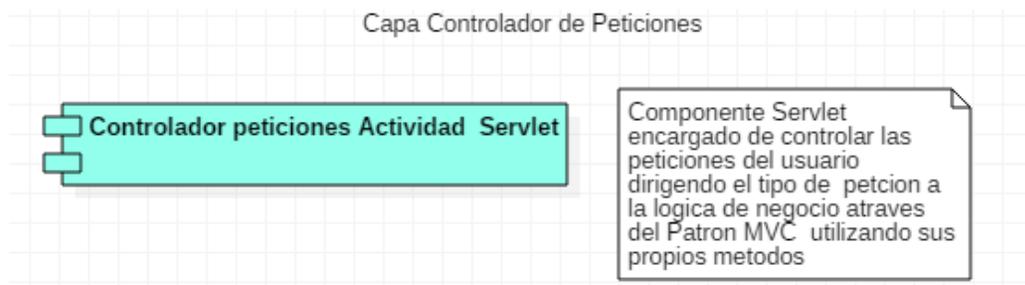
CONTROLADOR (CONTROLLER)

El Controlador es un componente que se encarga de recibir las peticiones realizadas por el usuario y establece una relación de interfaz con este mismo. La funcionalidad que aporta es la de redireccionar las peticiones del usuario al componente del modelo, y la de redireccionar las peticiones de navegación a páginas del Servidor Web.

La capa controladora de la petición es responsable de aceptar las Peticiones de la capa de presentación y de enviar el resultado de nuevo a ellas. Esta Capa actúa encapsulando (ocultando) las peticiones para los diversos objetos del modelo del negocio.

La capa controladora mapea el request o petición a un modelo específico del negocio e indica los pasos de la petición. La capa controladora recibe el resultado que de la capa de negocio y mapea su ruta hasta la capa de presentación por Ej.: SERVLET.

El Controlador comprende la funcionalidad involucrada desde que un usuario genera un estímulo (clic en un link, envío de un formulario, etc.) hasta que se genera la interfaz de respuesta. Entre medio, llamará a los objetos de negocio del Modelo para que resuelvan funcionalidad propia de la lógica de negocio y según el resultado de la misma ejecutará la JSP que deba generar la interfaz resultante.



Sistema de Gestión Documental (Capa Control Petición):

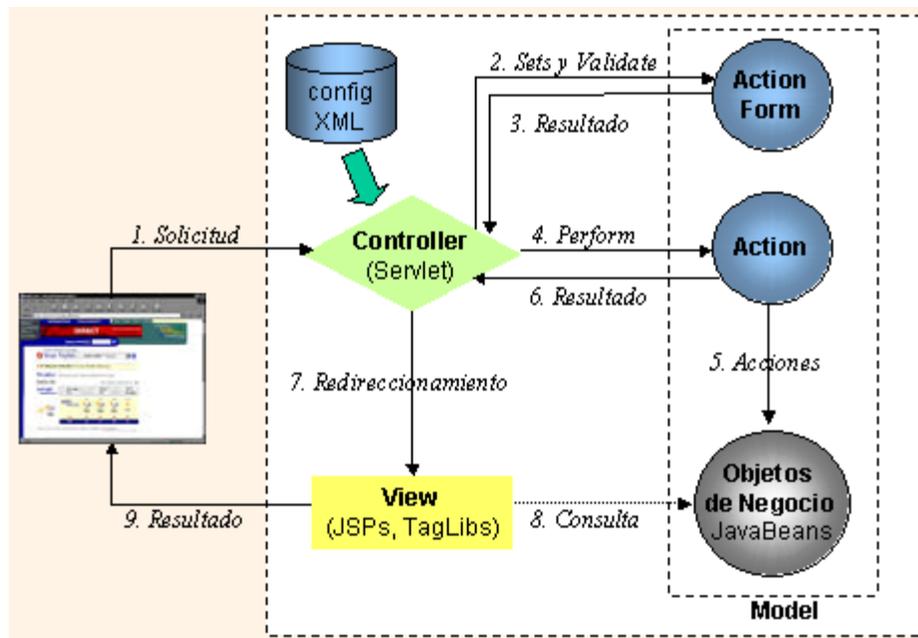
El Jefe de sucursal que utilizara el Sistema de Gestion Documental necesita ver el estado de las actividades asignadas, por lo tanto ingresa al navegador o browser invoca a la pagina de la Actividad Asignada y genera una solicitud que es atendida por el Controller (un Servlet especializado). El mismo se encarga de analizar la solicitud y los elementos o campos del formulario (colecciones), seguir la configuración que se le ha programado en su XML (configuracion interna de struts) y llamar al Action (objeto de la logica de negocio) correspondiente pasándole los parámetros enviados.

Es en este momento cuando el servlet envia los parametros (datos del formulario) a un action form que se encargara de validar el formulario, este devolvera estos parametros nuevamente al servlet el cual delegara a un Action que instanciará y utilizará los objetos de negocio para concretar la tarea de consultar el estado de la solicitud enviada por el cliente.

Según el resultado que retorne el Action, el Controlador derivará la generación de interfaz a una o más JSPs, las cuales podrán consultar

los objetos del Modelo a fines de realizar la petición de consultar sobre el estado de una actividad asignada del responsable de tarea, del SISTEMA GESTION DOCUMENTAL. Evidentemente, como todo Framework, simplifica notablemente la implementación de una arquitectura según el patrón MVC.

El mismo separa muy bien lo que es la gestión del control de la aplicación, del modelo de objetos de negocio y de la generación de interfaz. El controlador ya se encuentra implementado por Struts, aunque si fuera necesario se puede heredar y ampliar o modificar, y el workflow de la aplicación se puede programar desde un archivo XML.



MODELO (CAPA LOGICA DE NEGOCIO)

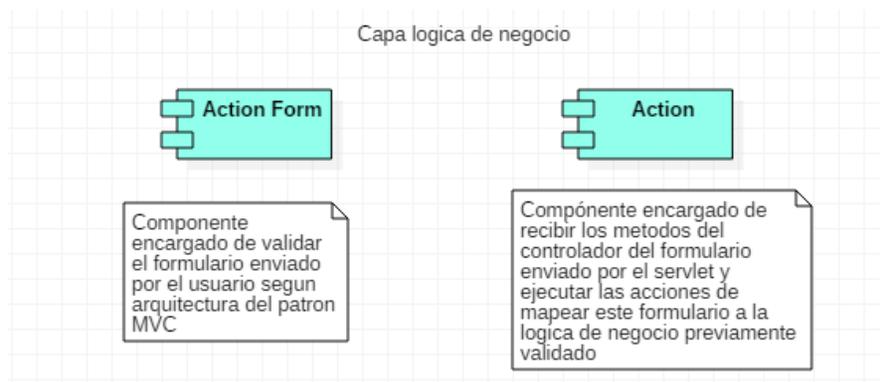


Fig. Muestra los componentes ubicados en la lógica del negocio que interactúan directamente con el Servlet controlador de peticiones.

Es en esta capa donde se resuelven las reglas del negocio especificadas para el Sistema de Gestión Documental para el dominio del problema, también es aquí donde se encuentran los componentes Enterprise Java Beans (EJB)¹.

Esta es una arquitectura que crea componentes para aplicaciones distribuidas y orientadas a transacciones. Las aplicaciones EJB² son escalables, transaccionales y multiusuarios.

Los EJB son una compleja especificación de arquitectura para componentes de servicio que:

- ◆ Permite el desarrollo en java de aplicaciones multinivel apoyadas en servidores o contenedores EJB.

¹ Ver Anexos De La Capa De Lógica Del Negocio

² Ver Anexos De La Capa De Lógica Del Negocio

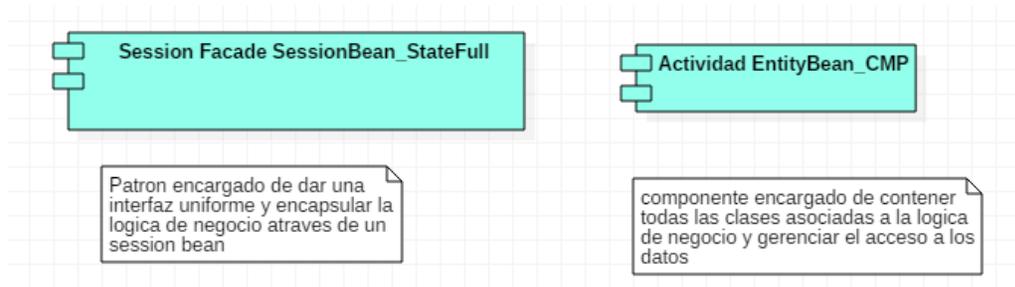
- ◆ Los objetivos de la arquitectura EJB es facilitar el desarrollo de aplicaciones concentrándose en la lógica de negocio: desarrollo, aplicación y aspectos de ejecución.
- ◆ La arquitectura EJB es basada a través de container (Contenedores) que ofrecen un entorno de ejecución para los componentes EJB, proporcionan una vista uniforme de los servicios solicitados en la especificación J2EE.
- ◆ Estos contenedores poseen herramientas adicionales para la instalación y configuración de componentes, la tarea principal de los contenedores es la gestión de recursos de los servicios y el ciclo de vida de los componentes EJB.
- ◆ Los EJB (Enterprise Java Beans) son una arquitectura que define la forma de construir componentes distribuidos del lado del servidor. Esta tecnología garantiza que los componentes programados sean escalables, eficaces y seguros.
- ◆ Los EJB son componentes distribuidos, esto quiere decir que podemos hablar de componentes que se ejecutan en diferentes servidores, contra diferentes bases de datos, de ahí nace su característica de aplicación distribuida.

- ◆ Los EJB¹ se ejecutan dentro de un servidor de aplicaciones, estos servidores deben cumplir el estándar fijado por SUN Microsystem INC., y se encargan de gestionar recursos como red, los pool de conexiones, o la gestión de la seguridad. El desarrollador construye los EJB y los servidores de aplicaciones los gestionan.

Es en la lógica de negocio donde se concentran las reglas del negocio donde actúan patrones de diseño como Business Delegate, Session Facade (serán especificados en la etapa de arquitectura física) es en esta capa donde se reciben las peticiones de los clientes convirtiéndolas en procesos que se realizaran en forma transaccional, en esta capa se identifican los siguientes componentes:

- ◆ Las Sesiones sin estado que son componentes que no mantienen un estado determinado de un determinado cliente, es decir se crea un hilo de ejecución nuevo por cada invocación que se realice. se utiliza cuando el número de usuarios es elevado. Por Ej. Session Stateless.
- ◆ Las sesiones con estado que son componentes que mantienen un estado determinado de un cliente durante toda su sesión. lo que lo obliga a establecer una instancia por cada cliente. Resultando más útil cuando se exigen altas prestaciones, a un número de usuarios no muy elevado. Por Ej. Session Statefull.

¹ Ver Anexos De La Capa De Lógica Del Negocio



Sistema de Gestión Documental (Capa Modelo De Negocio)

- ◆ Cuando los Jefes de Sucursal requieran seleccionar una Actividad asignada para motivos de solo consulta la Sesión seria sin estado ya que no se modificaría el estado de los datos en la lógica del negocio seria solo una rutina de consultar, a su vez esta sesión invocaría a un patrón determinado para lograr conectarse a una base de datos especifica.

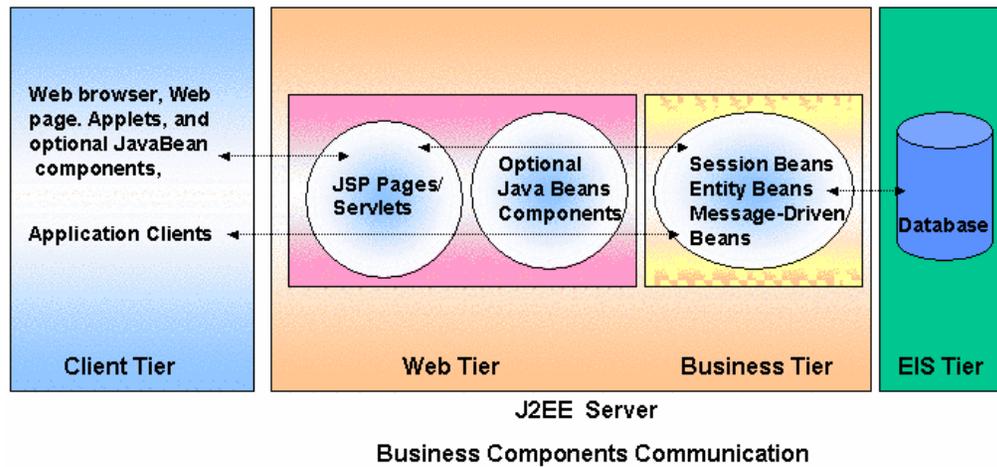
- ◆ Cuando un Jefe de Área decida sobre el estado de una investigación de accidente especifico de un trabajador, el jefe de área revisara la investigación de accidente modificando su estado ya sea para rechazarla o aceptarla en este caso en particular el Session Beans Statefull¹ necesita mantener un estado conversacional con el usuario ya que las modificaciones

¹ Ver Anexos En La Capa De La Lógica Del Negocio

que se hagan serán de carácter persistente, es decir los cambios se registrarán en la base de datos, hasta después que el administrador de solicitudes termine su sesión.

Los otros componentes que se identifican guardan relación con los objetos definidos en el dominio del problema estos componentes representan claramente a las clases definidas en el diagrama de clases en la toma de requerimientos funcionales, en la etapa de análisis del proyecto SISTEMA GESTIÓN DOCUMENTAL.

- ◆ Las Entidades distribuidas son componentes que representan una instancia de una o más clases del modelo. además, mantiene la información del modelo en memoria de manera que previo proceso de carga, se accede a esta información y posteriormente se actualiza dicha información en la capa de persistencia. Esto permite reducir el número de acceso a la base de datos con la ventaja a nivel de rendimiento del sistema. Por Ej. : Entity beans CMP.
- ◆ Las Entidades locales son componentes que representan una instancia de una o más clases del modelo pero que necesitan acceder a la persistencia para realizar cualquier operación, esto significa que necesita de un desarrollador para poder acceder a la base de datos y de un patrón, disminuyendo la portabilidad en la aplicación del contenedor, y siendo una alternativa a la programación orientada al objeto. Por Ej. Entity beans BMP.



Sistema de Gestión Documental (Capa Modelo De Negocio)

Cuando las peticiones ya han pasado por una serie de validaciones tanto en el la capa de control de peticiones. Un componente del Framework Struts mapea o en ruta la petición del cliente pasando a la lógica de negocio, en la cual es recibida por 2 patrones propios de la lógica de negocio (Business Delegate, Patrón Session Facade) los cuales utilizan una serie de métodos para que finalmente puedan llegar a los objetos del negocio como son los Entiy beans, los cuales

representan fielmente a las clases del negocio (con sus atributos, métodos) y pasan a transformarse en estos componentes los cuales en el caso de los Entity CMP¹ se gerencia a través del contenedor EJB lo que significa que todas las conexiones a las bases de datos y lógica de acceso a los datos la realiza el contenedor, es decir es una gran ventaja para el desarrollador y se aplica sobre el negocio , en esta capa se ubicarían todas las clases del Sistema de Gestión Documental, de las cuales el cliente quisiera realizar algún tipo de transacción tanto como simple consulta a la base de datos, el cual no es una lógica persistente o una actualización de cualquier dato que quiera realizar el administrador de solicitudes sobre las cotizaciones que son llegadas por vía Web y recibidas por este mismo, teniendo en este caso una lógica persistente ya que cuando el administrador termine su sesión los cambios que haya realizado sobre un cliente en particular permanecerán en la base de datos.

CAPA DE ACCESO A LOS DATOS O PERSISTENCIA

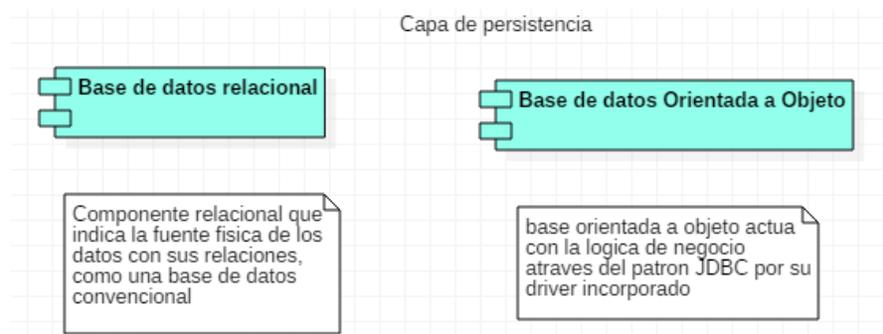
Parte del sistema encargado de dotarle de persistencia al SISTEMA GESTION DOCUMENTAL. Esta se ocupa de contener aquellos elementos que permiten el acceso físico a los datos, desde la capa de la lógica de negocio, además provee de la funcionalidad necesaria para poder otorgar una serie de mejoras en el rendimiento y escalabilidad como son la utilización de un pool de conexiones o disponer de bases de datos remotas.

¹ Ver Anexos De Tipos De Componentes EJB

- ◆ Debe haber una Fuente de datos que es un componente que nos proporciona la conexión lógica con la fuente de datos, Va a mantener una serie de propiedades como son la posibilidad de admitir transacciones, el usuario y password, etc.
- ◆ Tiene que haber un acceso a los datos este componente nos proporciona la posibilidad de realizar una conexión física con la fuente de datos. Dicho puente puede admitir conexiones remotas (TCP/IP), ser estándar (JDBC, ODBC) o legacy.

A su vez aquí es donde se ubican las fuentes físicas de los datos que va a contener al SISTEMA GESTION DOCUMENTAL. Estas fuentes pueden ser de distinta naturaleza y pueden estar localizadas en forma local o remota.

- ◆ Base de datos relacionales, estas bases están compuestas por un conjunto de tablas relacionadas y mediante un lenguaje de consulta SQL que nos facilita el acceso a los datos.
- ◆ Base de datos orientada a objetos, esta es más sofisticada que la relacional puesto que no sólo permite el tratamiento de los datos, sino que también proporciona un lenguaje distinto que proporciona la definición de lógica de negocio a este nivel.



A este nivel actúa el estándar JDBC (Java Data Base Connectivity) es un API que permite lanzar consultas a una base de datos, este estándar contiene un Driver que se conecta con el tipo de base de datos que se esté usando, esto trae una gran ventaja de una independencia de la base de datos que se esté usando. Es la opción más sencilla para la implementación de la persistencia en aplicaciones J2ee.

Se utiliza una persistencia manejada por el propio Bean de Entidad (Entity Bean CMP). La principal ventaja de utilizar JDBC¹ es que con creces es la tecnología más ligera que no añade ningún tipo de funcionalidad adicional. Como se decía anteriormente la persistencia es manejada por el propio contenedor EJB el cual implementa y ejecuta el código necesario para el acceso a la base de datos. Este código es compatible para todas las bases de datos soportadas por el servidor de aplicaciones.

Capa De Persistencia SISTEMA GESTION DOCUMENTAL

Cuando ya se está en la capa de persistencia cuando un Jefe de Sucursal necesita realizar una modificación de una actividad, el Entity Bean CMP² utiliza un lenguaje especial denominado EJB-QL este lenguaje declarativo permite utilizar un conjunto de las instrucciones definidas por el estándar SQL-92.

Este estándar permite ejecutar sentencias SQL o consultas de forma automática sin necesidad que el desarrollador se ocupe de esta implementación. Además posee con mayor rendimiento ya que puede definir diferentes tipos de relaciones entre Bean de entidad, lo cual implica que puede optimizar las consultas jerarquizando o clasificando el tipo de consultas que el administrador o los administradores estén realizando a la base de datos aliviando la carga en el sistema

¹ Ver Anexos En Capa De Persistencia

² Ver Anexos Capa De La Lógica Del Negocio

ahorrando de esta forma la cantidad de accesos a la base de datos y aumentando los tiempos de respuesta de cada usuario que esté realizando peticiones al SISTEMA GESTION DOCUMENTAL, los cambios que se realizan a nivel de esta capa se les llama persistente ya que cuando el administrador termina su sesión los cambios producidos en la base de datos permanecerán, dependiendo del perfil que tenga el usuario.

CAPITULO 5

HALLAZGOS

ARQUITECTURA FISICA

La etapa de arquitectura física define directamente la tecnología y el desarrollo del SISTEMA GESTION DOCUMENTAL basado en componentes. Estos componentes son los que forman parte de los distintos subsistemas o capas que fueron enunciados en la etapa de diseño lógico. En la etapa del diseño lógico fueron explicados los componentes de forma genérica que son parte de cada capa, ejemplificando algunos de estos en una aplicación al SISTEMA GESTION DOCUMENTAL con sus actores.

Por lo tanto en esta etapa corresponde definir cada uno de estos como conceptos y su relación en cada capa de la arquitectura J2EE en directa relación con el SISTEMA GESTION DOCUMENTAL.

Esta descripción se realiza de un modo más técnico especificando los tipos de componentes y aplicándolo al SISTEMA GESTION DOCUMENTAL, esto también implica una descripción de los distintos patrones de diseño que se implementa en cada capa.

En esta etapa se debe hacer un mapping o enrutamiento en cualquier aplicación que un usuario comienza a realizar una petición por la página Web hasta que esta petición es cumplida por la lógica del negocio y llega luego al acceso a los datos, Este enrutamiento será

graficado en un diagrama de componentes el cual representa todo el proceso del SISTEMA GESTION DOCUMENTAL.

Dentro de las capas se encuentran los siguientes componentes y los patrones que actúan:

- ◆ Capa de presentación: componente JSP
- ◆ Capa control de petición: componente SERVLET, patrón MVC
- ◆ Capa lógica de negocio: componente SESSION BEANS, ENTITY BEANS, Patrón Business Delegate, Session Facade, Service Locator.
- ◆ Capa de persistencia: componente base de datos orientada a objeto

DIAGRAMA DE COMPONENTES

Este diagrama se encarga de representar todos los componentes encontrados en la etapa de del diseño lógico o arquitectura funcional, describiendo los patrones de diseño que actúan en la capa controladora como en la capa de lógica del negocio.

En la capa de la lógica del negocio se encarga de atender las peticiones que se reciben del cliente por medio de la capa controladora de peticiones convirtiéndolas en procesos que se realizan en forma transaccional.

Para hacer útil la comunicación con los objetos EJB y así concretar la transacción del usuario sea cliente o administrador, se necesita de

patrones de diseño dependiendo del tipo de negocio y servicio que se necesite para satisfacer la aplicación del Sistema Gestión Documental. El desarrollo de patrones da la posibilidad de organizar las clases construidas en el diagrama de clases y objetos que se comunican y se pueden adaptar dependiendo del contexto del problema que estructuran. El desarrollo de patrones mejora su extensibilidad del SISTEMA GESTION DOCUMENTAL, es decir facilita la reutilización de los distintos componentes e incrementa su rendimiento.

DIAGRAMA DE COMPONENTES SISTEMA GESTION DOCUMENTAL

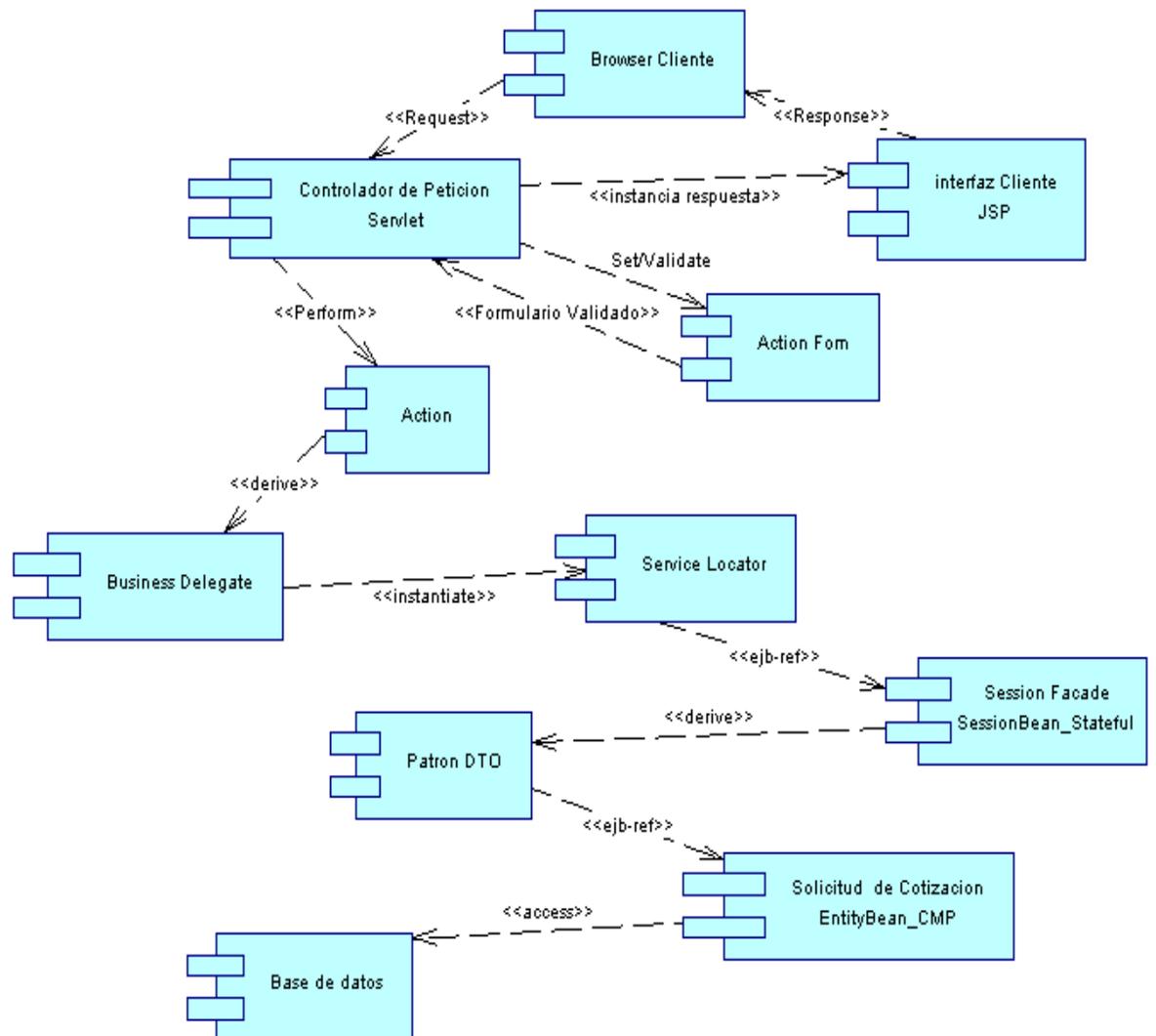


Fig. Muestra los distintos componentes de la arquitectura física del SISTEMA GESTION DOCUMENTAL, los tipos de relación y todo el proceso desde que el cliente envía una solicitud hasta que esta solicitud es procesada es explicado en la aplicación diagrama componente del SISTEMA GESTION DOCUMENTAL.

DIAGRAMA DE COMPONENTES DEL SISTEMA GESTION DOCUMENTAL

El cliente ejecuta el browser del navegador para iniciar la gestion de una actividad asignada, el navegador carga la página Web del Sistema de Gestión Documental que en este caso se trata de una página o componente JSP (JAVA SERVER PAGES)¹ son páginas dinámicas que combinan código HTML, dentro de esta página se encuentra un formulario el cual necesita el llenado de los campos en cada casillero con los datos del cliente, una vez terminado el proceso del ingreso de los datos ,el cliente selecciona un responsable de tarea al cual designar la actividad, cuando envía el formulario dando un clic al botón enviar la petición viaja por el protocolo HTTP vía Web transformándose en una petición la cual es atendida por un controlador en este caso un Servlet, esta página JSP es automáticamente compilada a Servlet por el motor JSP, este contenedor se encarga de procesar este formulario en las cuales viajan peticiones HTTP estos controladores pueden manejar varias peticiones de varios clientes que se estén enviando peticiones (request) al servidor.

Estos componentes actúan encapsulando los datos enviados por el cliente aislando así la capa de presentación con la capa de la lógica del negocio. Los Servlet tienen sus propios métodos para instanciar otros objetos como los Action Form este objeto es un validador del formulario y tiene los métodos para revisar cada casillero del formulario. Si están completos o se encuentran inconsistentes maneja sus propios métodos para avisar al cliente que ingreso mal sus datos al formulario.

¹ Ver Anexos Capa De Presentación

Si éste retornará uno o más errores, el Controller llamaría a la JSP del formulario para que lo volviera a generar (con los valores establecidos por el usuario) e incluyera el o los mensajes de error correspondientes avisándole al usuario por medio de una página JSP con los mensajes correspondientes, por el contrario si el formulario fue bien ingresado lo devuelve al controlador (Servlet)¹ el cual instancia al objeto Action que recibe el formulario del cliente ya validado y utiliza otros métodos para mapear o enrutar al formulario y así ejecutar las acciones sobre los objetos de la lógica del negocio, estos datos son enviados por el Action hasta la clase que implementa el patrón business Delegate el cual la abstracción del lado del cliente.

Este patrón reduce el acoplamiento entre las capas de negocio y la capa controladora. Este patrón encapsula las operaciones que se realizan para la búsqueda y el acceso a los objetos EJB.

También utiliza un componente llamado Service Locator que le proporciona un servicio de búsqueda es responsable de ocultar los detalles de búsqueda del servicio de negocio ya que maneja un directorio de nombres (JNDI), pero en este caso actúa como un localizar inicializando una fachada de sesión (sesión del usuario) o Sesión Facade este patrón utiliza un objetos Sesion Bean para proporcionar una fachada o interfaz uniforme para distintos tipos de casos o transacciones de un mismo proceso. La Sesion Facade mantiene el estado, es decir, tienen una conversación "uno a uno" con cada cliente que los invoca. Creando una sola fachada con sus propios métodos que se encargara de crear nuevas fachadas cambiarlas y ubicar a los Entity beans.

¹ Ver Anexos De La Capa Control De Peticiones

Este patrón reduce la sobrecarga de la red entre el cliente y el servidor porque su uso elimina la interacción directa entre el cliente y los datos y objetos de negocio, además de proporcionar una interfaz uniforme a objetos de negocio o procesos de una similar gestión.

Además enrutan las peticiones del cliente o transacciones a los objetos de negocio o Entitis para poder acceder a la base de datos, las trasferencias de datos y las llamadas a métodos desde la fachada a los Entitis. Se realizan a través del patrón Assembler Tranfer Object que se encarga de transferir utilizando sus propios objetos para transferir el contenido de la petición enviada por el cliente hacia el objeto Entity CMP.

Por su parte este patrón es capaz de acceder a la llave foránea del Entity el cual finalmente realiza la consulta respectiva a la base de datos. Los CMP soportan la persistencia de forma declarativa o implícita gracias al contenedor, por lo que podemos conseguir accesos muy rápidos a esas filas de procesos que le cliente desee realizar, por lo tanto se pueden mantener aquellos registros en memoria que creamos que se van a volver a utilizar guardando una referencia del cliente que utilizo esos objetos en otra sesión.

Por si fuera otro usuario por ejemplo un administrador este quisiera actualizar algún dato serían los mismos pasos pero con las clases correspondientes al jefe de sucursal.

TECNOLOGÍA DEL SISTEMA GESTION DOCUMENTAL

La tecnología J2EE propuesta para SISTEMA GESTION DOCUMENTAL se encuentra en diversas capas de la aplicación se Enumeran abajo:

1.- capa de presentación

- ◆ HTML
- ◆ Java Server Pages(JSP)
- ◆ XML
- ◆ Web Logic 11g

2.-Capa Control de Peticiones

- ◆ Java Servlet(SERVLET)
- ◆ Java clases
- ◆ Servidor Apache TomCat 7.0

- ◆ Weblogic 11g

3.-Capa de Lógica del negocio

- ◆ Enterprise java beans(EJB)
- ◆ Java clases
- ◆ JBOSS
- ◆ JDBC

4.- Capa de Datos o Persistencia

- ◆ PL/SQL
- ◆ MySQL

CONCLUSION

Dentro de las etapas que se realizaron en este informe como, el diagnóstico del problema se logró captar cabalmente las necesidades de los usuarios y los principales problemas que presentaba la entidad ya descrita para el desarrollo del SISTEMA GESTION DOCUMENTAL.

Estas necesidades fueron transformadas en requerimientos funcionales, se identificaron a los actores que interactuaban con los procesos, se logró establecer modularidad, una de las principales ventajas que plantea J2ee, siendo graficado por los casos de uso gracias a la herramienta llamada UML.

Graficando todos los escenarios posibles en los cuales el usuario podía interactuar.

Siguiendo con la etapa del diseño lógico en la cual tuvo un enfoque de la Arquitectura J2ee, la cual establecía una clara separación de capas o niveles del SISTEMA GESTION DOCUMENTAL para poder establecer una clara identificación de los componentes de J2ee, como interactuaban internamente como proceso, ejemplificando claramente dentro de la asignación de una actividad, mostrando así toda la lógica que para el usuario resulta transparente.

Se cumplió por lo tanto con desarrollar así componentes reutilizables, delegando portabilidad y escalabilidad en el tiempo para cualquier proceso o módulo que se quiera integrar en un futuro, satisfaciendo la generación de módulos y solucionando los problemas de gestión de documentación.

Principalmente en el módulo referente a asignación de actividades, y que por lo tanto este servicio pudiera ser gestionado por el jefe de sucursal y así realizar una actividad de charla o capacitación y dando un proceso automatizado, por lo tanto reflejando los índices de cumplimiento. Lo que cumplió con el objetivo de la primera etapa en el desarrollo del SISTEMA GESTION DOCUMENTAL.

Se eligió la tecnología más adecuada para empresas que necesitan procesos confiables, que necesitan contar con

una información óptima y eficiente. Se logró diseñar un sistema que establece claramente la división de todas las capas y principalmente la identificación de la lógica del negocio y la gran cantidad de patrones asociados a este, es por lo tanto un verdadero sistema capaz de brindar tiempos de respuesta eficientes con una cantidad de usuarios elevada y poder aumentar el rendimiento y satisfacción de los usuarios.

Uno de los principales fundamentos fue poder identificar la lógica del negocio ya que esta nos proporcionará la comunicación con la plataforma SAP/R3 de Electroandina la que incluye procesos más complejos de transacción y además por que J2ee posee un conector que es capaz generar una óptima comunicación entre los procesos y transacciones de los usuarios del SISTEMA GESTION DOCUMENTAL y los usuarios de Electroandina.

Por lo tanto una integración de los 2 sistemas. En conclusión montando las bases para la segunda etapa del proyecto SISTEMA GESTION DOCUMENTAL. Como último punto desearía destacar que este informe detalla el enfoque que se hace del punto de vista de la arquitectura del sistema, en este caso de la arquitectura J2ee contribuyendo enormemente al diseño y futura implementación de este sistema.

CAPITULO 6

BIBLIOGRAFÍA

TEXTOS

- ◆ Titulo: Yakarta Struts Live-----Autor: Rick Hightower-----Editorial: Soure Beat, Lcc, Highlands Ranch,Colorado,2004
- ◆ Titulo: Core J2ee Patterns----- Autor: Allur,----- Editorial Softcover,2004
- ◆ Titulo: J2ee Design An Development---Autor: Johnson Rod----- Editorial: Wroxxpress, Inc ,2003

- ◆ Titulo: The J2ee Tutorial Second Edition--- Autor: Stephanie Bodoff, Eric Armstrong, Jennifer Ball, --Editorial Addison Wesley .2004
- ◆ Titulo: Java Programming Language----- Autor: Sun Microsystem----- Editorial Sun Inc,901 San Antonio Road California,2002

SITIOS WEB

www.linti.unlp.edu.ar/catedras/Java%20y%20aplicaciones/2003/Teorias/pdf/clase%207-ejb.pdf

www.ncst.ernet.in/education/

apgdst/adbfac/14_EJB_Design_Patterns.pdf

java.sun.com/blueprints/patterns/SessionFacade.html

java.sun.com/blueprints/corej2eepatterns/Patterns/SessionFacade.html

www-106.ibm.com/developerworks/java/library/j-ejb1008.html

www.javaworld.com/javaworld/jw-01-2002/jw-0111-facade.html

www.tic.udc.es/~fbellas/teaching/is/Tema2Apartado2.3.5-6.pdf
www.tic.udc.es/~fbellas/teaching/is/Tema5Apartado5.6.3.pdf
www.programacion.com/java/tutorial/patrones2/3/ -
java.sun.com/blueprints/corej2eepatterns/Patterns/ServiceLocator.html
java.sun.com/blueprints/patterns/ServiceLocator.html
www.javaworld.com/javaworld/jw-07-2002/jw-0703-service.html
java.sun.com/blueprints/corej2eepatterns/Patterns/BusinessDelegate.html
java.sun.com/blueprints/patterns/BusinessDelegate.html
www-106.ibm.com/developerworks/java/library/j-ejb1022.html
www.tic.udc.es/~fbellas/teaching/is/Tema5Apartado5.6.4.pdf
es.tldp.org/Presentaciones/200103hispalinux/larrosa/html/x18.html
www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=strutsxdoclet
www.cica.es/formacion/JavaTut/Apendice/arg_mvc.html

CAPITULO 7

GLOSARIO

Caso De Uso

Es una descripción de la secuencia de interacciones que se producen entre un actor y un sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representan mediante el diagrama de casos de uso, mediante una elipse con el nombre en su interior, el nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.



Fig. De Un caso de uso

Dentro de los casos de uso se identifican los actores que son los usuarios que interactúan con el sistema y que están involucrados directa o indirectamente con el SISTEMA GESTION DOCUMENTAL.

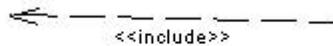
El Actor es un usuario del sistema, que necesita o usa alguno de los casos de uso. Un actor puede jugar más de un rol. Un solo actor puede actuar en muchos casos de uso; recíprocamente, un caso de uso puede tener varios actores. Los actores no necesitan ser humanos pueden ser sistemas externos que necesitan alguna información del sistema actual.



Fig. De Un Actor

Los tipos de relaciones que tienen los casos de uso con los actores son 2 tipos:

Usa (include) Es la relación entre dos casos de uso, denota la inclusión del comportamiento de un escenario en otro. Se utiliza cuando se repite un caso de uso en dos o más casos de uso separados. Frecuentemente no hay actor asociado con el caso de uso común.



Extiende (extends) Esta relación entre dos casos, denota cuando un caso de uso es una especialización de otro. Se usa cuando se describe una variación sobre el normal comportamiento.

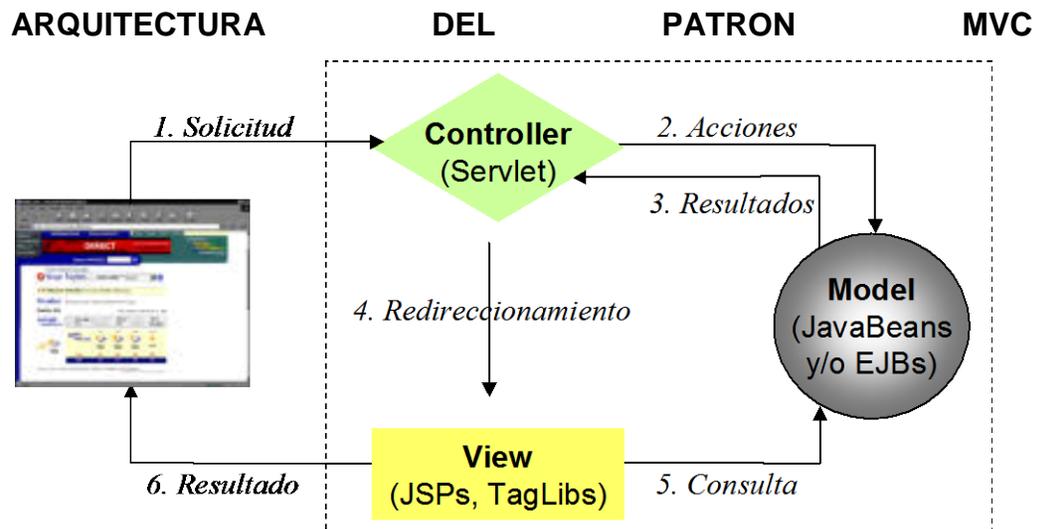


ARQUITECTURA J2EE

El **Modelo** es el objeto que representa la lógica del negocio del sistema. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. En esta capa es donde se ubica la lógica del negocio y sus componentes que actúan con la aplicación, Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus interfaces, y notificar a las interfaces cuando cambia el Modelo.

La **Vista** o interfaz 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 cliente cuando realiza algún tipo de petición. Interactúa con el Modelo a través de una referencia al propio Modelo.

El **Controlador** es el objeto que actúa sobre las órdenes del usuario controlando el tipo de petición y utilizando distintas clases para enrutar los tipos de peticiones del cliente, 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 el tipo de cambio o petición que realiza el cliente, entonces interactúa con el modelo de negocio.

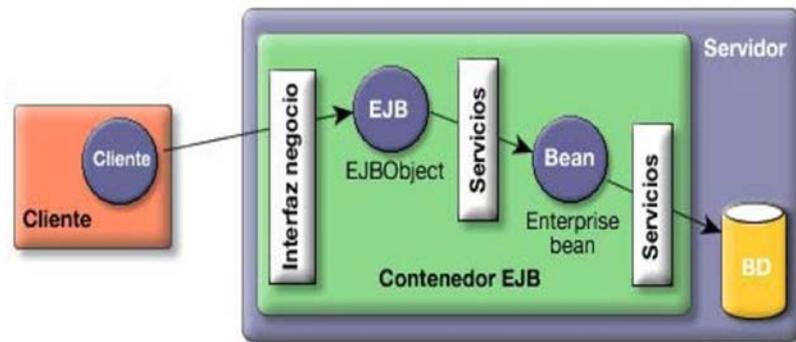


La lógica de negocio es una de las capas más importantes ya que esta va a contener todo el conjunto de entidades, relaciones y reglas que se encargan de la implementación de los procesos de negocio, estas reglas realizan el tratamiento de los datos y los procesos es por ese motivo que se llama lógica del negocio. Además de una serie de servicios como servicios de correo, servicios de seguridad, servicios de nombres y servicios configurables por el desarrollador.

Esta lógica del negocio se representa a través de un modelo de componentes, como son los componentes EJB y que están ubicadas dentro del contenedor EJB. Estos Componentes que se ejecutan en un ambiente controlado con soporte transaccional y de persistencia dentro de (EJB Container). Normalmente ubicados en la capa media de una aplicación, Existen tres tipos de componentes: Session Bean, Entity

Bean, Message Driven Bean, algunos de los patrones de diseño que usa esta capa esta session facade, bussiness delagate, service locator.

ARQUITECTURA DEL CONTENEDOR EJB



En la capa de persistencia o capa de acceso a los datos también actúa un patrón llamado JDBC este patrón puede manejar persistencia de datos consultados a un determinado servicio. Sin embargo, los mecanismos de almacenamiento persistente con frecuencia son diferentes y dependen con frecuencia del tipo de servicio específico que se consulta.

CAPA DE PRESENTACION (View)

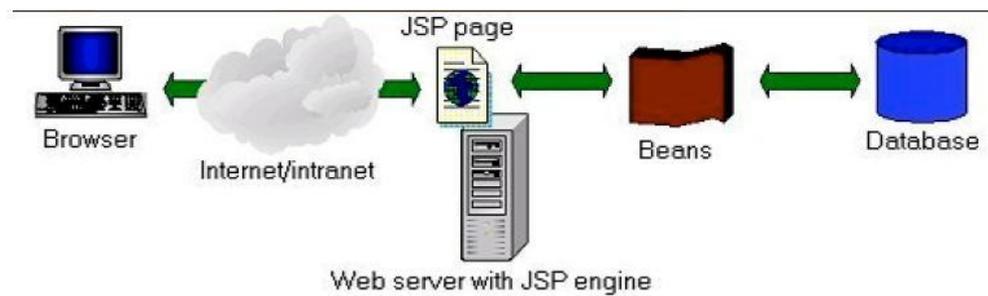
Ya se ha mencionado que esta capa es la generación de interfaz la cual es toda la vista que el usuario posee en cuanto al tipo de transacción que quiera realizar esta capa es soportada mediante un conjunto de Tags predefinidos por Struts, cuyo objetivo es evitar el uso de Scriplets (los trozos de código Java entre “<%” y “%>”), lo cual genera ventajas de mantenibilidad y de performance (desarrollo).

Logísticamente, separa claramente el desarrollo de interfaz del cliente y lógica de negocio permitiendo desarrollar ambas capas en forma de

módulos. En esta capa se ubica el componente JSP (Java Server Pages).

JSP

En la vista o interfaz comprende el componente JSP (JAVA SERVER PAGES) son paginas dinámicas que combinan código HTML, cada página es automáticamente compilada a Servlet por el motor JSP, en primer lugar es recogida y ejecutada, esta JSP es un archivo de texto simple. Cuando un cliente pide una página JSP y no se ha ejecutado antes la página inicialmente pasada al motor de JSP es compilada convirtiéndola en un SERVLET, la ejecuta y devuelve contenido de los resultados del cliente. JSP posee scripts que son bloques de código java residentes entre los TAGS (<% %>) estos TAGS pueden acceder a cualquier variable Beans.



CAPA DE CONTROL DE PETICION

Esta capa es la encargada de controlar y validar a cada usuario en cada sesión de cualquier aplicación de los usuarios del sistema, esta capa está regida por el patrón de diseño MVC, el flujo de la aplicación está dirigido por un Controlador central. El Controlador delega solicitudes - en nuestro caso, solicitudes HTTP a un manejador apropiado. Los manejadores están unidos a un Modelo, y cada controlador actúa como un adaptador entre la solicitud y el Modelo. El Modelo representa, o

encapsula, un estado o lógica de negocio de la aplicación. Luego el control normalmente es devuelto a través del Controlador hacia la Vista apropiada. El reenvío puede determinarse consultando los conjuntos de mapeos, normalmente cargados desde una base de datos o un fichero de configuración. Esto proporciona un acoplamiento cercano entre la Vista y el Modelo, que puede hacer las aplicaciones más fáciles de crear y de mantener para el SISTEMA GESTION DOCUMENTAL.

STRUTS

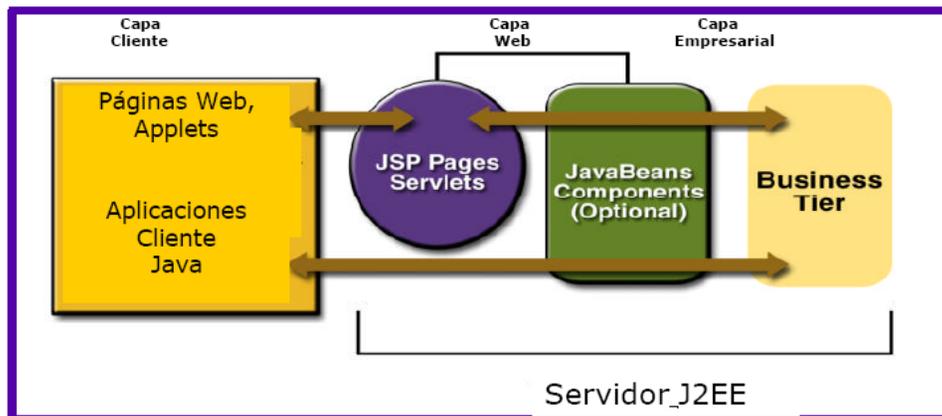
Siguiendo el patrón de diseño Modelo-Vista-Controlador, las aplicaciones Struts tiene tres componentes principales: un servlet controlador, que está proporcionado por el propio Struts, páginas JSP (la "vista"), y la lógica de negocio de la aplicación (o el "modelo").

El servlet controlador Struts une y en ruta solicitudes HTTP a otros objetos del marco de trabajo, incluyendo Java Server Pages y subclases `org.apache.struts.action.Action` proporcionadas por el desarrollador Struts.

Una vez inicializado, el controlador analiza un fichero de configuración de recursos, la configuración de recursos define (entre otras cosas).

Los `org.apache.struts.action.ActionMapping` (enrutamiento) para una aplicación. El controlador usa estos mapeos para convertir las solicitudes HTTP en acciones de aplicación.

Servlets y JSP



Servlet: Es un programa escrito en java que se ejecuta en un contenedor Web y especializado, los servlets más usados son los HTTP Servlet, estos reciben requerimientos peticiones HTTP del Web browser (cliente), la procesan y le devuelven resultados como respuestas HTTP, estos servlets son objetos gerenciados o dirigidos por el contenedor de servlets o contenedor Web, además maneja su ciclo de vida, maneja los errores, provee seguridad.

Los servlet son objetos multithreaded (multihilos), es decir pueden manejar múltiples peticiones de varios clientes que se estén enviando peticiones (request) al servidor.

Los SERVLET son componentes importantes, estos se encargan de encapsular las peticiones de los usuarios y a su vez devuelven esta respuesta encapsulada utilizando sus propios objetos, estos reciben estas peticiones a través de los métodos Get, Set de los formularios de HTML y la instancia sus propios objetos o clases servlet. Cuando un usuario completa un formulario y lo envía, el Controller busca en el

scope especificado el ActionForm Bean correspondiente (todo esto configurado en el **struts-config.xml**) y si no lo encuentra lo crea.

CAPA DEL MODELO DE NEGOCIO (MODEL BUSSINESS)

Model = ActionForms + Actions + Objetos de Negocio

ACTIONFORMS: es un validador de formularios, estos son objetos JavaBeans con las siguientes características:

- ◆ Extienden org.apache.struts.action.ActionForm
- ◆ Implementan SET y GET para cada INPUT del formulario de una página
- ◆ Definen un método VALIDATE que retorna una lista de errores
- ◆ Opcional: método reset para reinicializarse
- ◆ Si éste retornara uno o más errores, el Controller llamaría a la JSP del formulario para que ésta lo volviera a generar (con los valores establecidos por el usuario) e incluyera el o los mensajes de error correspondientes.

- ◆ No debe tener nada que corresponda a la lógica de negocio
- ◆ Finalmente llama al método VALIDATE. Si todo estuviese bien, llamaría al método PERFORM del Action (también configurado en el struts-config.xml) pasándole el ActionForm Bean como parámetro para que sea utilizado para obtener los valores de los datos.
- ◆ Debe ser un Firewall entre el usuario y el Action que detenga todo tipo de errores de incompletitud o inconsistencia
- ◆ No debería tener más que implementaciones de SETS y GETS (obligatoriamente un par por cada Input del FORM; si el Input se llama nombre entonces tendremos GetNombre() y SetNombre (String nombre), y de los métodos reset y Validate.
- ◆ Si el formulario se desarrolla en varias páginas (por ejemplo, en las interfaces de tipo "Wizard"/"Asistentes" el ActionForm y el Action deberán ser los mismos, lo que permitirá, entre otras cosas, que los Input se puedan reorganizar en distintas páginas sin cambiar los ActionForm ni los Action.

ACTION

Ejecutor de acciones sobre los Objetos de Negocio estos Objetos de Negocio son entidades que representan los conceptos del negocio, sus características y comportamiento y los procesos de la lógica del negocio, el Action Obtiene los valores necesarios del Action Form, JavaBean, request, sesión o de donde sea. Y luego llama a los objetos de negocio del Modelo. Los Action son:

- ◆ Son clases que extienden `org.apache.struts.action.ActionForm` (configuración interna de struts).
- ◆ Implementan el método `perform` (`ActionMapping`, `ActionForm`, `HttpServletRequest`, `HttpServletResponse`) que retorna un `ActionForward`.
- ◆ `ActionMapping` indica los posibles valores de retorno del `ActionForward` según el `struts-config.xml`.
- ◆ `ActionForm` es el `ActionForm` que procesó el formulario (es nulo si no se ha definido un `ActionForm` en el `struts-config.xml`).

LA ARQUITECTURA EJB

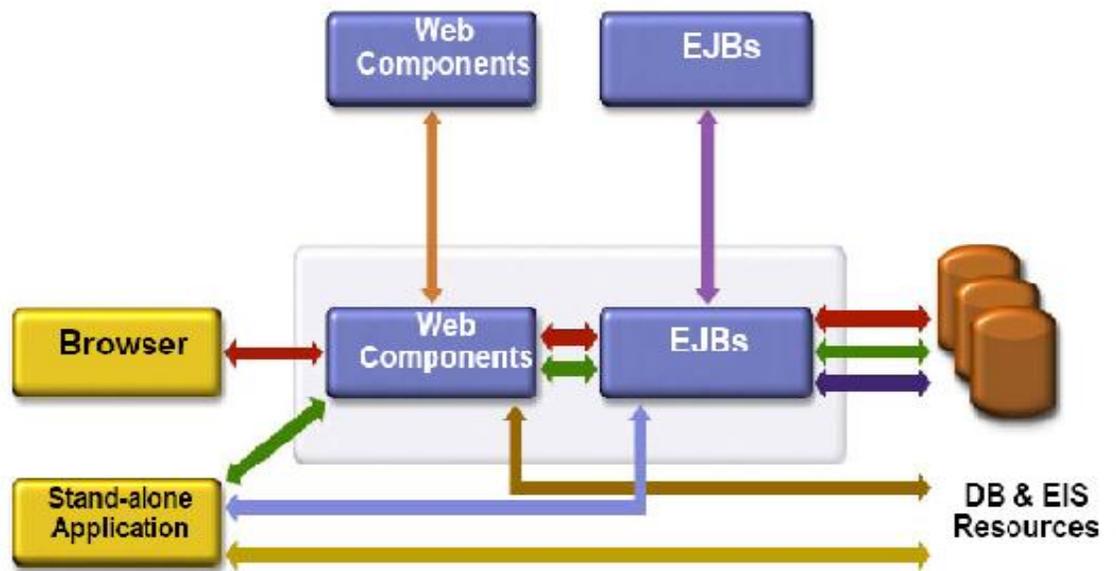
Los EJB son componentes software que se ejecutan en la parte servidora de una aplicación y pueden ser ejecutados en un entorno multi-capa distribuido. Un componente es una pieza de software

accesible a través de un interfaz público, en el caso de los EJB, tanto el software como esta interfaz (en realidad dos) deben cumplir el estándar e implementar obligatoriamente ciertos métodos que dicha especificación define, de esta forma los contenedores de EJB pueden gestionar el ciclo de vida de forma uniforme siendo independiente el contenedor que gerencia al componente.

Los EJB son componentes que pueden estar físicamente alejados del cliente por ello se dice que su interfaz publica es remota (Remote Interface). La especificación EJB hace uso de RMI/IIOP para ofrecer esta característica. En RMI/IIOP todo objeto remoto dispone de un interfaz accesible a los clientes, el cliente puede hacer uso de esta interfaz a través del "Stub" que es un Proxy enviado al cliente.

Cuando el cliente invoca al "Stub" este se comunica con el "Skeleton", que es el Proxy situado en el lado del servidor. Una vez recibido el mensaje por el "Skeleton" este es el encargado de comunicárselo al objeto distribuido, el cual resolverá la operación y dará la respuesta al cliente a través del "Skeleton" primero y el "Stub" después.

Tanto el "Stub" como el "Skeleton" son transparentes para el cliente y este siempre tiene la sensación de estar invocando al objeto distribuido directamente.



Middleware Implícito o Declarativo

La verdadera clave de los componentes distribuidos modernos es que su middleware es implícito, es decir, el código de nuestro software no llama directamente a este software integrado en la parte servidora, sino que define en el descriptor de cada componente, se cómo queremos que sea el contexto en el que se va a ejecutar.

Por ejemplo, un EJB no tiene la necesidad de llamar de forma explícita, en el código, al API de transacciones para comenzar una transacción, sino que podemos definir en un fichero distinto, como queremos que se comporte el EJB.

Para ello aparte de dicho fichero donde se define el comportamiento del EJB, se necesita uno que capture la llamada a los componentes distribuidos y configure la forma en la que estos van a trabajar. En el caso de los EJB el EJB Object. El EJB Object es generado por el propio

contenedor y es el encargado de interactuar con el contenedor para ofrecer sus servicios. Entre otros ofrece la gestión de transacciones distribuidas, seguridad, gestión del ciclo de vida, persistencia, etc.

Las ventajas, componentes más sencillos, el código se centra en resolver la lógica no en la forma de hacerlo. Además se puede cambiar el comportamiento de los EJB sin tocar su código, sólo modificando el fichero que especifica cómo debe funcionar, el descriptor.

Transparencia de situación física

La especificación dice que puede variar el lugar donde se encuentran los EJB, pero de alguna forma se necesita saber cómo podemos acceder a ellos, esta es la labor del Home Object. El Home Object es el encargado de crear y destruir los EJB Object. Es un objeto generado por el contenedor, de tipo "Factory", que implementa el segundo de los interfaces públicos de un EJB, el Home Interface. De esta forma el contenedor conoce cuál es el tipo de EJB, que parámetros debe pasar al EJB Object para crear un nuevo componente o qué tipo de búsquedas ha definido un usuario para obtener la referencia a varios componentes.

Es importante señalar que existe un y sólo un EJBHome implementando su correspondiente Home Interface para cada tipo de EJB en cada contenedor. Esto hace posible cachear la referencia en algunos casos.

Interfaces

En la especificación EJB 1.1 no existía este concepto pero muchos servidores de aplicaciones empezaron a implementar mejoras para llamadas dentro de una misma máquina virtual, de esta forma se eliminaba la necesidad de traducir los datos a RMI para hacer una

llamada no remota. Las personas que llevan a cabo la especificación se dieron cuenta de esta "necesidad" y crearon los interfaces locales en la versión 2.0, que sólo permiten recibir llamadas de clientes de la misma máquina virtual. Estas llamadas son mucho más rápidas y además los parámetros son pasados por referencia no por valor, con lo que disminuye el consumo de memoria.

Tipos de Componentes

Actualmente el estándar EJB define tres tipos de EJB:

Session Beans (SB)

Los EJB de sesión se encargan de resolver la lógica de negocio de la aplicación. Se puede decir que cada método contenido en un SB resuelve un caso de uso de la aplicación, ejemplos pueden ser Control de Usuarios, Gestor de Precios o Control de Procesos.

Dentro de la especificación podemos encontrar dos tipos de Sessions Beans, State Less (SLSB) y State Full (SFSB).

State Less Session Beans

Los SLSB son componentes sin estado, es decir, no guardan ninguna relación entre diferentes llamadas de un mismo cliente. Es más, entre dos llamadas a un mismo tipo de SLSB es posible que el contenedor direcciona al cliente a diferentes instancias del componente. Otro dato esclarecedor de cómo se comportan este tipo de componentes es su relación con el número de usuarios. Para N usuarios hay M instancias

siendo $M < N$ en la mayoría de los casos. El valor máximo y mínimo de M es configurable en la mayoría de los servidores. Con estos dos valores se define el tamaño del "pool".

Los pools de objetos han sido utilizados desde los inicios de Java. Se trata de definir un espacio en memoria para varios objetos, de tal forma que en vez de tener que instanciar un objeto cada vez, podemos reutilizar las instancias existentes en el pool. Esto conlleva un consumo de memoria pero ofrece una mayor velocidad. Además hace trabajar menos al recolector de basura, con lo que la mejora de resultados es bastante notable.

Ese es el objetivo de los pools de EJB que utilizan los contenedores, reutilizar las instancias de los EJB en memoria.

El contenedor garantiza que a una instancia de un EJB sólo puede acceder un hilo, con lo cual nunca dentro de un EJB tendremos un problema de concurrencia, el contenedor lo soluciona por nosotros.

State Full Session Beans

Los SFSB son lo contrario a los SLSB, mantienen el estado, es decir, tienen una conversación "uno a uno" con cada cliente que los invoca. Para mantener esta conversación es necesario hacer uso del objeto Handle.

El objeto Handle es una referencia serializable a un objeto EJB. Guardando esta referencia se puede acceder posteriormente al EJB con el que el cliente había establecido una relación.

Con los SFSB cada cliente tiene asociado un objeto EJB durante una conversación. En caso de que esta referencia se pierda, se puede recuperar gracias al Objeto Handle. Por eso es el cliente quien se tiene que encargar de preservar la instancia de estos objetos si quiere volver a llamar a "su" SFSB.

Este tipo de EJB es tachado como verdadera "maquina pesada" y se desaconseja su uso en la mayoría de los casos. Otros mantienen su utilidad por la posibilidad de mantener una sesión distribuida entre varios servidores. En cualquiera de los casos se debe evitar la pasivación de los mismos.

La pasivación es el proceso de serialización a disco de aquellas instancias menos recientemente usadas (LRU, aunque es posible especificar otra política de liberación dependiendo del Servidor de Aplicaciones) cuando el número de instancias requeridas por los clientes es mayor de las que puede soportar el pool de SFSB.

Este proceso se puede dar, bien porque se acaba la memoria o porque se ha llegado al límite de instancias máximas configurado en el descriptor.

Al proceso contrario se le llama activación, y se produce cuando un cliente llama a un método de su SFSB asociado y este se encuentra pasivado.

Si se prevé que con asiduidad van a ocurrir estos procesos es conveniente intentar buscar otra alternativa, como por ejemplo manteniendo la sesión en el servidor de Servlets-JSP.

Entity Beans

Los EJB de entidad están directamente relacionados con los datos de la aplicación, son objetos que mantienen en memoria los datos que manejan la aplicación, ejemplos, Noticia, Foro, Usuario.

Los Entity Beans normalmente mapean las tablas de una base de datos relacional, aunque también es posible que mantengan la persistencia de los datos en ficheros, por ejemplo un XML, o en LDAP. En cualquiera de los casos el objetivo de un Entity Bean es cachear los datos en memoria desde una fuente persistente y mantener una sincronización total entre el estado de los datos en memoria y la fuente de datos.

Por esta razón se dice que los Entity Bean son los EJB que sobreviven a caídas del sistema, ya que en caso de un fallo del sistema, los datos que hay en memoria están guardados en el dispositivo persistente, con lo cual, cuando se reinicie el servidor se recuperaran sin ningún problema.

Este tipo de EJB abstrae totalmente la capa de persistencia del sistema y funciona como una herramienta de traducción de base de datos relacional a objeto, por ejemplo, se podría cambiar el nombre de una columna de una tabla y el resto de capas no se daría cuenta, ya que a esa columna se accedería a través de un método get/set del Entity Bean.

Quizás las críticas de este tipo de componentes vienen por la lentitud en las funciones de búsqueda, los llamados "finders". Estos métodos devuelven una colección de interfaces remotos que cumplen una serie de condiciones, similares a las instrucciones SQL. Si bien es cierto que es más lento que una simple llamada SQL, no suele ser mucho mayor que estas si se hace un correcto uso de las transacciones y se minimiza el número de llamadas a través de la red. En cualquiera de los casos, no suele ser recomendable hacer este tipo de llamadas si sólo se va a hacer una lectura

Como se ha explicado antes, los EJB se mantienen dentro de un pool en memoria. En caso de los Entity Beans esto significa que tenemos varias filas de la base de datos en memoria, por lo que podemos conseguir accesos muy rápidos a esas filas, por lo tanto es interesante mantener aquellos registros en memoria que creamos que se van a volver a utilizar.

Dentro de los Entity Beans hay dos formas de manejar la persistencia, se puede dejar en manos del programador o en manos del contenedor.

Bean Managed Persistence

Los BMP son los Entity Bean que soportan la persistencia gracias a las instrucciones explícitas del programador, por lo que este deberá introducir las instrucciones necesarias en los métodos definidos por la interfaz EntityBean, ejbLoad, ejbRemove, ejbCreate, etc.

Cada vez se usan menos este tipo de Entity Beans, pero todavía hay algunas operaciones que sólo se pueden realizar gracias a las habilidades de los programadores. En cualquiera de los casos siguen siendo imprescindibles cuando la persistencia no se consigue a través de una base de datos relacional.

Container Managed Persistence

Los CMP soportan la persistencia de forma declarativa o implícita gracias al contenedor, es decir, no es necesario implementar los métodos de la interfaz EntityBean, sólo hay que definir de forma correcta el descriptor para que así el contenedor tenga la información necesaria para gestionar la persistencia contra una base de datos relacional.

A pesar de que en los principios de los EJB, este tipo de Entity Bean no era muy usado, ya que los programadores no se fiaban mucho de los contenedores, actualmente son los más usados e incluso están recomendados por causas de eficiencia con respecto a los BMP.

Message-Driven Beans

Son muy parecidos a los Beans de sesión pero estos reciben mensajes y no ofrecen respuesta al cliente, son asíncronos. Autorización de compra, Chequeo de tarjeta de crédito son ejemplos de ellos. Nuevos en EJB 2.0

Su único método, `onMessage`, recibe el tipo de mensaje y con él realiza una serie de operaciones de las cuales el usuario no obtendrá respuesta directa. Cuando este tipo de operaciones se van a realizar en un tiempo indeterminado y no es necesaria la respuesta inmediata al cliente, es donde toman fuerza este tipo de EJB.

COMPARACIÓN ARQUITECTURA J2EE V/S .NET

CARACTERÍSTICA	J2EE	.NET
TIPO DE TECNOLOGÍA	Estándar	Producto
EMPRESAS QUE LO OFRECEN	Más de 40	Microsoft®
LIBRERÍAS DE DESARROLLO	Java core API	.NET Framework SDK
INTÉRPRETE	JRE	CLR
PÁGINAS DINÁMICAS	Servlets, JSP	ASP.NET
SERVICIOS Web	SOAP, WDSL, UDDI	SOAP, WDSL, UDDI
INTERFACES GRÁFICAS	Java Swing	Win Forms y Web Forms
LENGUAJES UTILIZADOS	Java	C#, Visual Basic, C++, otros
CAPA DE PRESENTATION	JSP	ASP.NET
CAPA LÓGICA DE NEGOCIO	EJB	.NET MANAGED COMPONENTS
CAPA DE ACCESO A BB.DD	JDBC	ADO.NET

CONCEPTO DE SAP / R3

SAP/R3 es una plataforma empresarial que está constituida por módulos aplicables a sus departamentos.

MÓDULOS SAP

Módulo SD (ventas y distribución) venta al cliente que puede ser en el punto de venta o a través de medios de transporte o a través de Internet. etc. en este módulo se hacen las transferencias bancarias de la venta del cliente, la factura y la orden de transporte, Módulo MM (compras) se refiere a compras que la empresa hace con su proveedor o empresa suministradora, también se hacen transferencias bancarias de la compra, factura y orden de transporte.

Módulo FI (finanzas) para la parte contable, es como el Contaplus pero más extenso, se pueden realizar cuentas de mayor y de diario.

Módulo RH (Recursos Humanos), control de salarios de trabajadores, días festivos...etc.

Módulo BW (Business Warehouse) para hacer todo tipo de gráficos estadísticos.