



**UNIVERSIDAD NACIONAL AUTONOMA DE
MEXICO**

FACULTAD DE INGENIERIA

Mensaje Universal SisaXml (SisaXml)

T E S I S

Informe para titulación por experiencia laboral

**PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTA

Juan Plácido Partida

DIRECTOR DE TESIS

Ing. Heriberto Olguín Romo



Contenido

Introducción	5
OBJETIVO	5
ANTECEDENTES.....	5
DEFINICIÓN DEL REQUERIMIENTO Y ASIGNACIÓN.....	7
Capítulo 1 Desarrollo	9
1.1 ANÁLISIS PARA LA PROPUESTA DE SOLUCIÓN.	9
1.2 ALTERNATIVAS DE SOLUCIÓN	9
1.3 LA PROPUESTA DE SOLUCIÓN	10
Capítulo 2 Descripción Mensaje Universal XML (SisaXml)	15
2.1 VERSIÓN INICIAL 1.0.0	15
Capítulo 3 Herramientas de supervisión	25
3.1 AUDITOR DE XML.....	26
3.2 SUPERVISOR DE MSJ XML (WEB SERVICES)	27
3.3 CONSULTAS SISAXML	27
3.4 CONCLUSIÓN SOBRE LAS HERRAMIENTAS DE EXPLOTACIÓN DE LA INFORMACIÓN	28
Capítulo 4 Evoluciones	29
4.1 CONCENTRACIÓN DE HERRAMIENTAS DE SUPERVISIÓN.	29
4.2 EJECUCIÓN PROCESOS POR HILOS	29
4.3 EJECUCIÓN DE WEB SERVICES EXTERNOS Y CONSUMIDORES DE SISAXML	29
4.4 SISTEMA DE CONFIGURACIÓN Y ADMINISTRACIÓN DE MENSAJES.	29
Conclusiones	31
Apéndice	33
APÉNDICE A ORGANIGRAMA DE TELMEX Y SISTEMAS	33
APÉNDICE B DESCRIPCIÓN GENERAL DEL PROCESO DE ATENCIÓN DE UNA QUEJA	37
APÉNDICE C DESCRIPCIÓN GENERAL: CAPTURA DE UNA QUEJA POR MEDIO DEL SISTEMA SISA.	45
APÉNDICE D DEFINICIÓN DE MIDDLEWARE	53
APÉNDICE E BÚSQUEDA DE MIDDLEWARE PARA LA PROPUESTA DE SOLUCIÓN.	61
APÉNDICE F XMLBLASTER PROPUESTA SELECCIONADA	67
APÉNDICE G METODOLOGÍA DE DESARROLLO Y DOCUMENTACIÓN	73
APÉNDICE H ESPECIFICACIÓN GENERAL DEL SISTEMA SISAXML	101
APÉNDICE I HERRAMIENTAS UTILIZADAS	129
APÉNDICE J HISTORIA DEL PROYECTO MENSAJE UNIVERSAL XML.	135
APÉNDICE K GLOSARIO	139
APÉNDICE L DOCUMENTACIÓN GENERADA.....	141
Bibliografía	143
Referencias	143

Introducción

Objetivo

Establecer la comunicaciones entre objetos que se ejecuten en distintos procesos (ya sea en un mismo equipo, Intranet o en Internet) para diferentes aplicaciones con una comunicación sencilla y escalable, considerando que la comunicaciones entre los objetos de las aplicaciones son distintos.

Antecedentes

Telmex

Teléfonos de México S.A.B. de C.V.¹ (*Telmex*) es una empresa Mexicana de Telecomunicaciones con sede en la Ciudad de México. La empresa ofrece una variada gama de productos y servicios relacionados con las telecomunicaciones en México, Latinoamérica (Argentina, Brasil, Colombia, Chile, Perú) y Estados Unidos,

Desde su privatización, TELMEX ha tenido muy clara su misión de ser no sólo una empresa de telecomunicaciones sino uno de los pilares fundamentales en el desarrollo de México, al proveer gran parte de la infraestructura necesaria que nos permite estar comunicados en todos los rincones del país, con productos y servicios de vanguardia tecnológica mundial. Con un enfoque total de servicio a nuestros clientes, para continuar siendo de las empresas más modernas en su ramo.

Por lo que es necesario mantener en correcto funcionamiento todos los servicios telefónicos con un mínimo de fallas. Con estos se logrará impactar positivamente a los clientes de TELMEX y ganar la confianza de los clientes potenciales, para finalmente lograr un crecimiento notable en cuanto a cantidad de clientes satisfechos con los servicios que se proporcionan.

Para satisfacer la demanda de los grandes clientes, TELMEX implementó como un factor prioritario, realizar un análisis completo y certero que proporcionara información y que sirviera de apoyo para efectuar un cambio sustancial a los sistemas y procesos para su atención. Esta investigación dio como resultado factores de cambio muy importantes, se citan algunos puntos:

- Entender y anticiparse a las necesidades de los socios de negocios en telecomunicaciones
- Valor agregado en los servicios
- Servicios de Calidad en menor tiempo y costo
- Reducción de tiempo de solución de problemas

Los requerimientos de los clientes sirvieron de base para la implementación de los sistemas MAC, CAS y CAO, con el siguiente propósito:

Propiciar que TELMEX, ante un ambiente de competencia mundial, proporcione un excelente servicio de telecomunicaciones a sus clientes más importantes.

De esta forma se pretende resolver satisfactoriamente las necesidades de servicios en telecomunicaciones de los clientes, al crearles una atmósfera de confianza y seguridad, a través de la implementación de centros de atención con muy alta tecnología, los cuales podrán interactuar eficazmente con otros centros operativos de TELMEX. Es pertinente aclarar que el MAC, CAS y CAO realizan las mismas funciones y su diferencia estriba en el tipo de clientes.²

La Dirección Corporativa de Mercado Empresarial y Sistemas, es el área encargada de satisfacer las necesidades de los departamentos mencionados, la automatización de los procesos de estas áreas se logra por medio del sistema Sisa (Sistema de Información de Servicios Avanzados)

Sisa

Sisa esta compuesto por tres módulos

Subdirector de sistema de mercado empresarial Dr. Mario Magidin Matluk

Gerente Sistema de información de servicios Avanzados Lic Raul Lopez Ruiz.

¹ Para mayor información ver Apéndice A: Organigrama de Telmex y Sistemas

² Para mayor información ver Apéndice B Descripción General del proceso de atención de una queja.

Modulo	Descripción	Responsable Sistemas
Administración de contratos	Administra la contratación de de los servicios	Act Octavio Hernandez Zetina
Instalación	Administra el seguimiento en la construcción de los servicios	Lic Alfredo Arrieta Duran
Mantenimiento	Administra el seguimiento de las quejas recibidas por la falta de servicio que son reportados a las áreas MAC, CAS, CAO3	Ing Juan Plácido Partida

Tabla I.1 Módulos y responsables de SISA (Sistema de Información de Servicios Avanzados)

Telmex Internacional

TELMEX empresa mexicana, líder mundial en telecomunicaciones, en 2004 expandió sus operaciones a otros países de Latinoamérica, con la finalidad de aprovechar las oportunidades de crecimiento derivadas de cada mercado en el cual se ha incursionado, permitiendo satisfacer las necesidades de los clientes con un servicio integral de alta calidad y diversificando las fuentes de ingresos.

El 21 de diciembre de 2007, el Consejo de Administración de TELMEX aprobó una iniciativa estratégica para reorganizar su estructura corporativa en dos empresas independientes. Acorde a la iniciativa, TELMEX escindió las empresas que operan en Latinoamérica, así como las operaciones de Sección Amarilla y se constituyó TELMEX Internacional.

TELMEX Internacional es una empresa líder en servicios de telecomunicaciones con presencia en Argentina, Brasil, Chile, Colombia, Ecuador, Perú; y Uruguay; cuenta con una completa estructura de soporte regional y local para responder con oportunidad y eficiencia a los requerimientos de los clientes.

Se exporto el mismo esquema de trabajo que se lleva en TELMEX para implantarlo en Telmex Internacional, evolucionado los sistemas que se utilizan en cada país.

De manera individual no existía ningún problema, en la atención de una queja en cada sistema, pero no se contaba con un seguimiento global, cuando se trata de una falla internacional por lo que se requería la administración e integración de los diferentes sistemas cuando se llegase a reportar una queja de tipo internación, México no estaba exento ya que existen servicios internacionales que pasan por nuestro país.

Se requería un sistema que administrara el seguimiento y el intercambio de información entre los diferentes sistemas que fuera general, lo mas sencillo de administrar, robusto en cuanto a seguridad y capas de ser escalable conforme se avanzara en la tecnología y en los servicios de Información.

Los clientes internacionales requerían de una atención inmediata en caso de presentarse una problema con los servicios de comunicación, por ello se desarrolla LATAM (Portal Internacional Telmex).

El sistema pretende llevar la administración de los servicios internacionales, uno de sus módulos, es el control de fallas reportadas por los clientes.

La base de datos de LATAM cuenta con los segmentos y servicios relacionados en cada país por donde cruza el servicio internacional.

Dentro de sus procesos LATAM debería recibir la queja del servicio internacional, analizar la ruta por donde pasa el servicio, detectando los sistemas que llevan la administración de los sub-servicios. Emitiendo un mensaje a cada aplicación del problema reportado.

Los Sistemas involucrados recibirían el mensaje de LATAM, generando un reporte en el sistema local para su atención y seguimiento.

El seguimiento debería, ser retroalimentado a LATAM para informar al cliente el avance en la puesta en servicio de su sistema.

³ Para mayor información ver Apéndice C Descripción general automatización captura de una queja por medio del Sistema SISA.



Fig I.1 Diagrama bosquejando un servicio internacional

Definición del requerimiento y asignación.

El Dr. Mario Magidin Matluk (subdirector de sistemas) no quería que se desarrollara una interfaz punto a punto entre SISA y LATAM, el sistema debería ser lo suficientemente abierto para transportar no solo lo relacionado entre los dos sistemas antes mencionados

Como se trataba de una interfaz adicional a las ya existentes para el intercambio de información con el sistema Sisa, se requería de una aplicación escalable de fácil mantenimiento, liberando de procesos aislados y cada vez más complicado su mantenimiento.

Fui asignado para desarrollar junto con mi equipo, el nuevo proyecto; es cual es bastante ambicioso al momento de escuchar los aspectos que debería considerar, pensé en las siguientes.

1. Comunicar los sistemas de Telmex Internacional (LATAM) al sistema SISA – Mantenimiento, las cual se comunicaría por medio de la red de redes (Internet, aprovechando esta gran plataforma)
2. Agrupar las interfaces existentes con SISA en sus tres módulos, estas interfaces se comunicaban de diferente manera, archivos planos FTP, Sockets
3. Integrar nuevas interfaces de una manera sencilla y trasparente al resto de las que ya estuvieran desarrolladas y trabajando, si importar la manera de comunicarse.

La asignación de este nuevo y ambicioso proyecto es una responsabilidad adicional a las que ya tengo asignadas.

Aplicación		Descripción
Sisa		Modulo de mantenimiento
Supervisor de quejas		Aplicación desarrollada en Web para la consulta en el avance de las quejas levantadas en Sisa-mantenimiento.
Escalador		Notificación por medio de Mail, Celular, al inicio Localizador, en su inicio para quejas que estuvieran fuera del tiempo de atención, en este momento es general y notifica todo lo relacionado a la aplicación SISA
Instalador		Se desarrollo un instalador distribuido para la aplicación SISA para diferentes plataformas, permite integrar otras aplicaciones relacionadas con Sisa, por ejemplo Siatel, Sise.
Seguridad		Relacionado al modulo de instalación se generan las cuentas que tienen permitido el acceso a Sisa, el control se realiza por medio de cuentas de configuración y cuentas de aplicación, control de bajas relacionado con Siatel

Tabla I.2 Proyectos asignados

Capítulo 1 Desarrollo

1.1 Análisis para la propuesta de solución.

El objetivo era buscar la solución que abarcara lo conocido hasta el momento, ya que no todos los departamentos y sistemas podrían migrar a una plataforma muy avanzada (por problemas en el desarrollo, aplicaciones cerradas), así como considerar a los clientes nuevos, estábamos en conocimiento que se debería manejar un estándar para las aplicaciones nuevas, así como las que podrían adaptarse a desarrollar la transición.

También se deberían considerar las aplicaciones que se encuentran en la misma infraestructura de comunicación de TELMEX, como las externas, me tenía que ayudar de recursos como la Internet, por los bajos costos evitando los altos costos por contratar un servicio dedicado en la comunicación, además de identificar quien absorbería los gastos de mantenimiento del servicio.

Inicie agrupando las interfaces existentes y la requerida, por la forma de comunicarse, el volumen de información, interfaz externa o dentro de la misma red de Telmex, el resultado fue el siguiente:

- 1) Por Internet nuevo requerimiento SISA - LATAM
- 2) Sockets
- 3) Archivos FTP
- 4) Macros
Intercambio por medio de archivos con un formato predefinido, generando macros ya sea por medio de un shell de UNIX utilizando herramientas de Oracle como SQL*Loader, leer un archivo plano y por medio de una programa externo almacenar y procesar la información.
Es un procedimiento fijo, considerando la presencia de errores los cuales detienen el proceso hasta que es modificado manualmente el archivo y pueda reiniciarse la carga.
- 5) Carga masiva de archivos planos
Intercambio de archivos planos, procesado de forma manual. Es un proceso lento, se depende de confiabilidad de la captura.

Lo anterior me dio la pauta para buscar alternativas de solución, tenía que buscar el mejor sistema a bajo costo o libre (software con código abierto)

1.2 Alternativas de solución

Finalmente presente tres alternativas de solución, desde aplicaciones con lenguaje abierto (OpenWare) hasta aplicaciones propietarias las cuales requerían un costo por licenciamiento.

Un *Middleware*⁴, era lo que se ajustaba a las necesidades del proyecto, por definición. El middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).

El middleware nos abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware.

Por lo general el middleware del lado cliente está implementado por el Sistema Operativo subyacente, el cual posee las librerías que implementan todas las funcionalidades para la comunicación a través de la red⁵.

Sistema	Descripción	Tipo
XmlBlaster	El xmlBlaster es "Middleware Orientado a Mensajes" ("Message Oriented Middleware", MOM) publicado bajo los términos de la	Sistema de código abierto

⁴ Para mayor información ver Apéndice D: Definición de Middleware

⁵ Para mayor información ver Apéndice E: Búsqueda de Middleware para la propuesta de solución

Sistema	Descripción	Tipo
	GNU Lesser General Public License; el desarrollo está siendo llevado hacia adelante por Michele Laghi, Peter Roth, Konrad Krafft, Manuel Kron, James Birchfield y Marcel Ruff. Dado que muchas personas probablemente no entenderán nada del término "Middleware Orientado a Mensajes", diré algo acerca de él. MOM se necesita cada vez que hay que intercambiar información en una red heterogénea con nodos utilizando protocolos diferentes.	
Websphere IBM	WebSphere es una familia de productos de software propietario de IBM, aunque el término se refiere de manera popular a uno de sus productos específicos: WebSphere Application Server (WAS).	Sistema Propietario
OpenFusion Corba	Alto rendimiento, multi-plataforma, las soluciones multi-idioma Implementaciones escalables desde el chip a la empresa QoS Arquitectura Orientada a Servicios (SOA) de tecnología Simplificación de desarrollo CORBA Apoyo de clase mundial, Servicios Profesionales y Formación	Sistema Propietario

Tabla D.1 Cuadro propuestas de solución y una breve descripción.

La diversidad de los productos middleware y el traslape con sus características hacen difícil darles categorías. La más importante es el servidor de aplicaciones EJB, pero bases de datos de objetos middleware para mensajes, servidores de aplicaciones para redes y muchas otras categorías comprenden un conjunto de elecciones apropiadas para diferentes aplicaciones.

El EJB y estándares que los acompañan proveen un marco para construir bases de estándares abiertos transaccionales y para múltiples aplicaciones Java. Todavía muchas áreas necesitan atención incluyendo áreas que ahora usan soluciones específicas de vendedores, tales como balance de carga y autenticación. El control de acceso necesita ser declarativo y propagado a las bases de datos así como los componentes de software, y esto es necesario para permitir separar dominios administrativos. Mas aún los problemas de modelos de presentación estándar necesitan ser enviados, como lo hace el código de despliegue. Finalmente los drivers de Java puro reemplazables que soportan transacciones distribuidas necesitan ser implementadas.

1.3 La Propuesta de solución

De los middleware y sistemas de comunicación analizados me decidí por el sistema XmlBlaster⁶ por sus diferentes ventajas además que es un sistema de código abierto, el cual puede ser modificado con autorización del autor. La propuesta se podría adaptar a las características requeridas por TELMEX.

XmlBlaster

El xmlBlaster es "Middleware Orientado a Mensajes" ("Message Oriented Middleware", MOM) publicado bajo los términos de la GNU Lesser General Public License; el desarrollo está siendo llevado hacia adelante por Michele Laghi, Peter Roth, Konrad Krafft, Manuel Kron, James Birchfield y Marcel Ruff.

Dado que muchas personas probablemente no entenderán nada del término "Middleware Orientado a Mensajes", diré algo acerca de él. MOM se necesita cada vez que hay que intercambiar información en una red heterogénea con nodos utilizando protocolos diferentes.

⁶ Para mayor información ver Apéndice F: XmlBlaster Propuesta seleccionada

El ejemplo clásico es una red de máquinas expendedoras de tickets distribuida sobre todo un país. Estas máquinas han sido producidas por diferentes vendedores y fueron programadas por sus propios equipos de programadores con diferentes lenguajes de programación. Si, por ejemplo, el precio de un ticket ha de cambiarse, es necesario un traductor para traducir el mensaje "nuevo precio" al formato correcto utilizado por las máquinas reales y distribuirlo entre ellas.

También es importante traducir los resultados (en este caso éxito o error) al lenguaje que entiende el programa de administración. Este trabajo está siendo realizado por el Middleware Orientado a Mensajes como el xmlBlaster que se intercambian mensajes codificados en XML.

XmlBlaster confía también en un modelo publicación / suscripción pero esta vez el servidor está escrito en Java puro y debería ser capaz de ejecutar en cualquier plataforma con JDK >= 1.2.

Los suscriptores cubren casi el espectro completo. Ejemplos pueden utilizar expresiones XPath para filtrar los mensajes que desean recibir, y añadir sus propios MIME de filtrado plugins. C / C ++, Java, Perl, Python, VisualBasic.net, C #, PHP cliente.

La comunicación con el servidor se realiza con CORBA (utilizando JacORB), RMI, XML-RPC, nativo de zócalo, o un plug-in HTTP persistentes y es posible filtrar mensajes con expresiones XPath.

Si CORBA fuera problemático para ciertas aplicaciones, también se puede usar correo electrónico o conexiones http como plug-ins. Como sugiere el nombre los mensajes en sí están codificados en XML, se puede transmitir cualquier cosa (incluyendo datos binarios).

El marco de seguridad plugin permite la autenticación/autorización de muchas maneras. Actualmente hay LDAP y passwd basada en plug-ins disponibles. Licencia: GNU Lesser General Public License (LGPL)

El problema al que me enfrente es que aun siendo código Java es poco accesible, el segundo inconveniente, debido a que es código abierto se tiene poco soporte.

El sistema XmlBlaster fue el que seleccione por que contenía todos los elementos necesarios para la interfaz, el principal problema era que se tenía que instalar gestores a cada cliente de los cuales se deseara la suscripción a la interfaz.

Se deseaba que fuera transparente a cada usuario y estos no tuvieran que casarse con una herramienta.

El cliente debería generar sus propios clientes para suscribirse a la Interfaz, en este momento en análisis de diseño.

Como se trata de una aplicación con código libre, no existía un soporte como tal, solo de los desarrolladores que no era fácil de localizar.

El resto de middleware que seleccione se descartaron por que requería de una inversión permanente por licenciamiento con el producto debido al licenciamiento que tendría absorber en su momento el cliente también me enfrente que los sistemas tienen poca portabilidad

La propuesta aceptada fue el desarrollar un middleware propietario de TELMEX, el cual se podría extender como sistema para el intercambio de información. En un principio y mientras maduraba el sistema corrigiendo los posibles errores se utilizaría para las diferentes interfaces de mercado empresarial.

Contamos con lo necesario para el desarrollo del sistema, expertos en programación, material técnico, comunicaciones y apoyo por parte de nuestro subdirector.

Solución propuesta

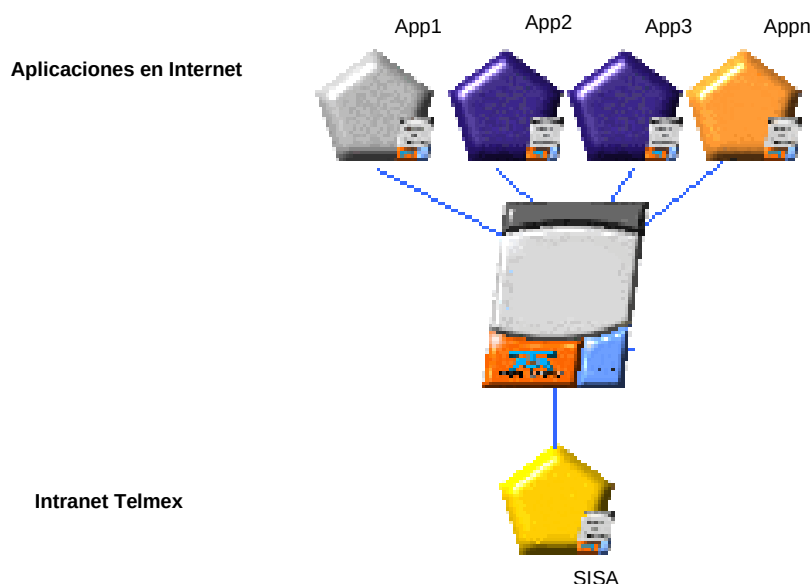


Figura D.1 Interfaz SISA – INTERNET Solución propuesta

Facilitar el intercambio de información entre aplicaciones en Internet y el sistema corporativo de Telmex SISA, proporcionando flexibilidad en la comunicación y agilidad en el servicio de atención a clientes desde Internet.

La herramienta por medio de mensajes XML utiliza un formato estándar que puede integrarse de manera fácil y sin requerir que se instalen componentes externos además de relacionarse de fácil a la aplicación SISA y con poco mantenimiento.

Para ellos se desarrollaran varios programas en la base de datos que nos ayude a integrar la información recibida y enviada (interfaz con la aplicación SISA), esta herramienta debe ser capaz de ejecutar programas externos que nos proporcionen la información requerida, se requiere desarrollar una aplicación para el monitoreo de los mensajes que se transmitan al ciber – espacio.

Interfaz SISA

- Programa Java que identifique si es un mensaje de entrada o salida.
- Si es un mensaje de entrada identifique si se debe almacenar en el modulo de mantenimiento como queja.
- Si es un mensaje de salida o notificación, en esta fase se determina el cliente donde se envía el avance del servicio.

Aplicación Sisa

- Preparación de las pantallas en SISA para que en determinada acción o regla que se ejecute se cree el mensaje xml.
- Al momento de ejecutarse la acción o se cumpla con la regla definida en la base de datos preparada para el mensaje universal se escriba en las tablas de salida.

Monitor de Interfaz SISA

- Pantallas de monitoreo de mensajes. ayuda a identificar rápidamente cual es el error ocurrido al tratar de ingresar las quejas desde el Msj Univ Xml indicarle cual es el problema y como se puede corregir, se propone que el monitor del Mensaje Universal Xml se realice en WEB (por medio servlets, java – script, etc) para tener un interacción desde cualquier parte por medio de un navegador de internet.

La propuesta fue aceptada; se tenía que desarrollar en un máximo de tiempo de 6 meses ya que para ese entonces nuestro primer cliente en la interfaz sería el sistema LATAM.

Que finalmente no fue el primero en inaugurar la comunicación entre dos sistemas, la primera aplicación en utilizar la herramienta fue Clarify de RED UNO aproximadamente por el año del 2003.

Capítulo 2 Descripción Mensaje Universal XML (SisaXml)

2.1 Versión inicial 1.0.0

La propuesta de desarrollo fue terminada el 2 de noviembre del 2002, para desarrollar la propuesta inicial, nos apoyamos de la metodología de desarrollo y documentación UML⁷, la cual nos ayuda a desarrollar un sistema modular, de fácil escalabilidad.

La propuesta la planteo lo mas general posible, manteniéndome desde el principio la idea de tratar a la interfaz como una caja negra es el ejemplo básico de la metodología de desarrollo UML, donde sabemos que hay una entrada y una salida, además que se puede entregar un resultado pero no se sabe a ciencia exacta que es lo que ocurre dentro de esta maquinaria. la cual se debe incrustar en el funcionamiento de otra maquinaria sin afectar el funcionamiento estable de las aplicaciones, si no todo lo contrario, se debe ganar en el proceso de ambas aplicaciones.

Las aplicaciones extras se comportan para el mensaje universal SisaXml como otras cajas negras donde recibiría información se procesa y se entrega una respuesta.

Dentro de la aplicación SisaXml, los subsistemas en si, se deberían comportar de la misma manera como cajas negras, cada modulo debería manejar sus propias reglas de negocio.

Por ejemplo la actualización de la información en la base de datos de SISA, se proyecto para que fueran controlados por medio de STP (Store Procedures) que serian ejecutados desde el SisaXml.

Este mismo procedimiento se utilizaría varias veces, se debería de controlar por medio de configuración almacenada en una instancia de base de datos, la empresa tiene licenciamiento y soporte con Oracle.

Los STP almacenados en la base de datos ORACLE ayuda a que las reglas de negocio se encuentren en sisa y no en el middleware que es un sistema general

Descripción grafica del Sistema SisaXml

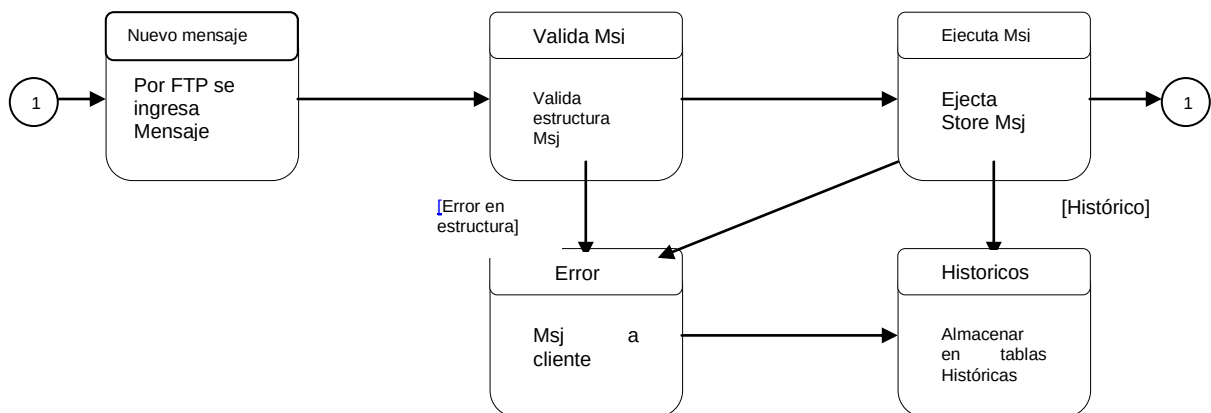


Figura DM.1 Diagrama de transición de estados

⁷ Ver Apéndice G: Metodología de desarrollo y documentación

Diagrama de Flujo (DGF)

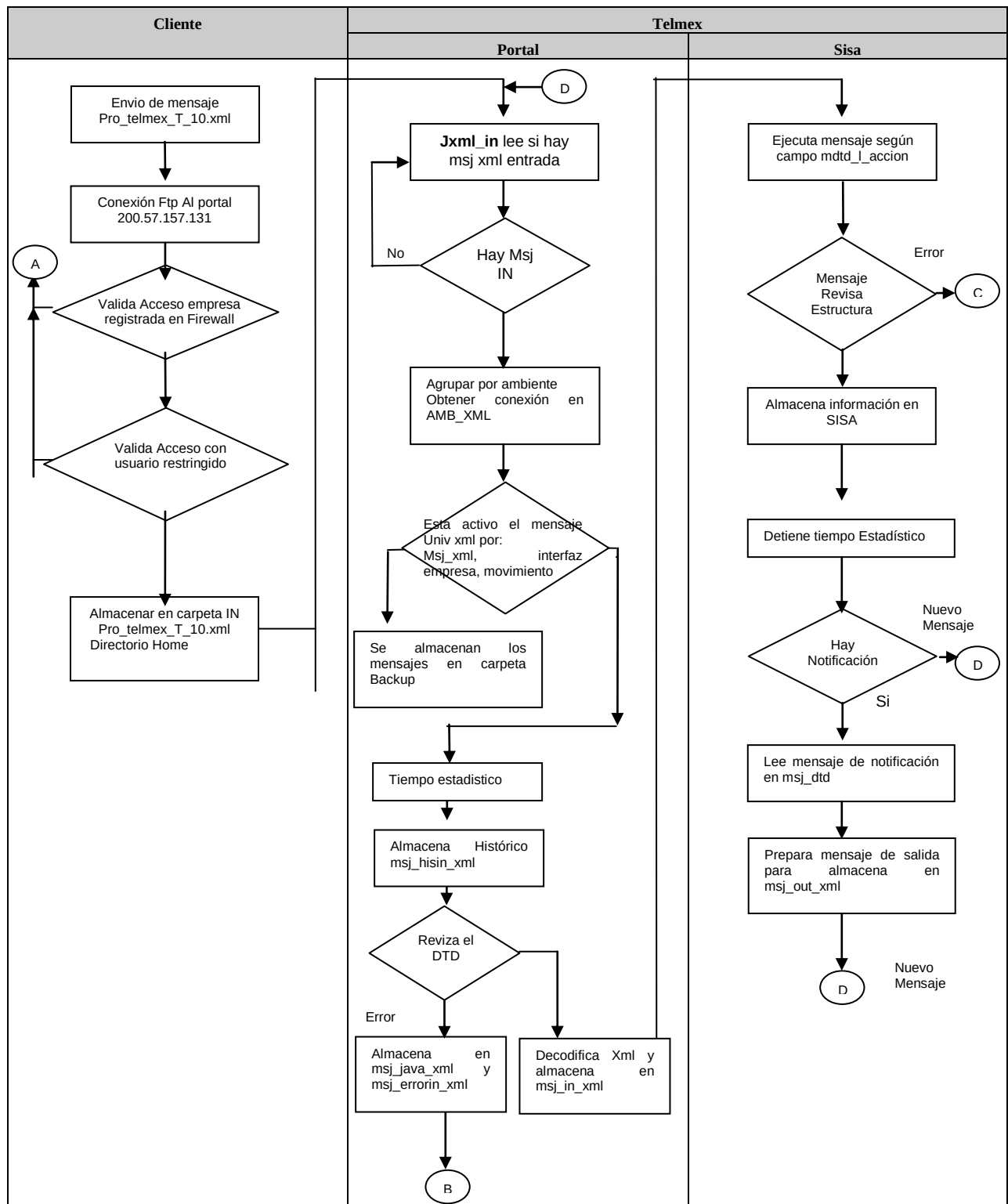


Figura DM.2 Insertar Mensaje Xml y Procesamiento

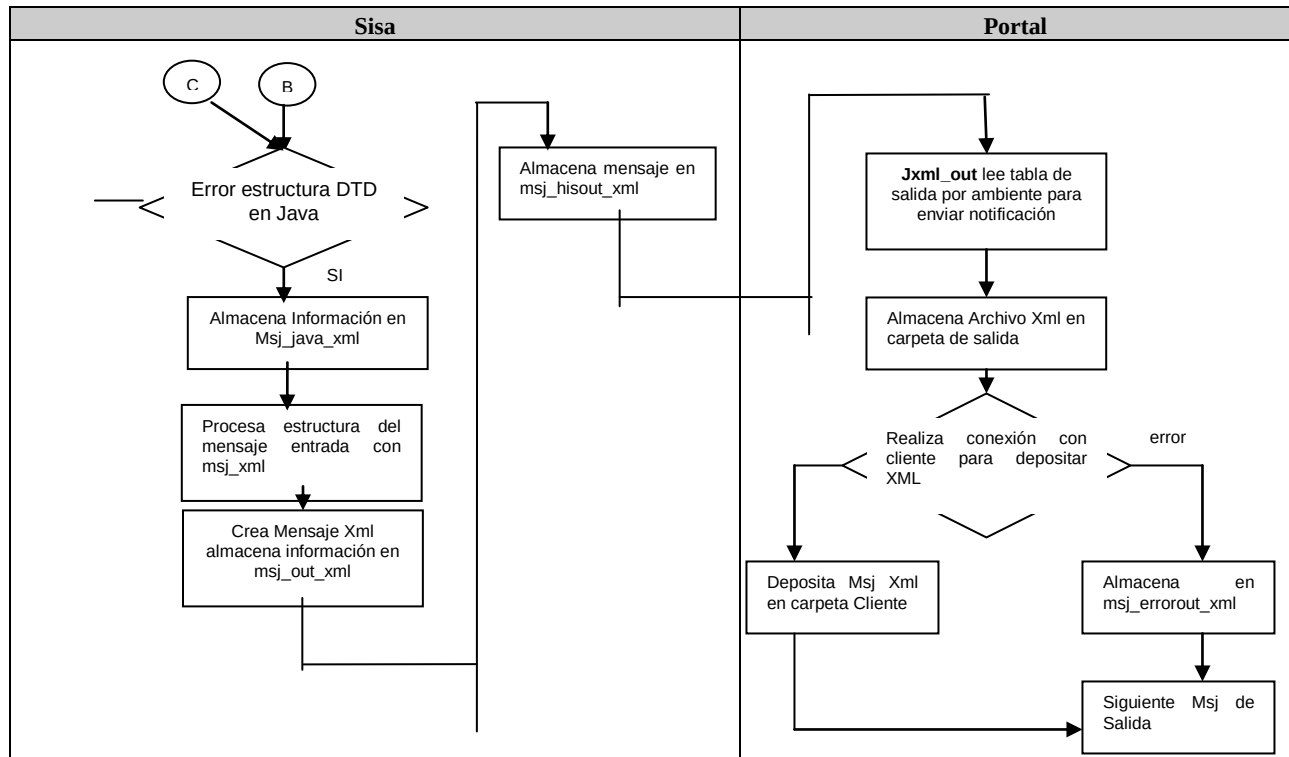


Figura DM.3 Errores en Mensajes Estructura DTD y Mensaje

Portal	Sisa
A Registro del Cliente en FireWare permisos solo FTP Acceso al servidor a clientes con login restringido (in) Sisa restricción de FTP solo salidas (out) Control de accesos a los servidores uno a uno	Del Portal Programa Java acceso a BD por JDBC Base de datos intermedia para acceder a SISA Sinónimos privados entre BD intermedia y SISA Manejo de Mensajes por Stores Procedure Generación de Mensajes de Salida por Store P.

Tabla DM.1 Procesos de seguridad Portal - Sisa

Cuadro de actualización de Entidades (CAE).

Procesos

Comentario	Regla
Las entidades relacionadas a los mensajes de entrada MSJ_IN_XML, MSJ_HISIN_XML, todos los campos son actualizados con la información del mensaje recibido, en caso de existir error en la estructura del DTD o de los campos se almacenan los campos de las tablas MSJ_ERRORIN_XML, MSJ_JAVA_XML así como la de notificación MSJ_OUT_XML.	RTMsjXml-Dol DOL
Para el caso de los mensajes de salida se debe almacenar la información completa de las entidades MSJ_OUT_XML y MSJ_HISOUT_XML, en caso de existir un error en la transferencia del mensaje se almacenaría en las tablas MSJ_ERROROUT_XML.	

Cuadro de actualización de relaciones.

Procesos

Comentario	Regla
Para el caso del Mensaje Universal Xml las relaciones se cuidan en momento de almacenar la información, no existe una relación real en el modelo MSJ_XML ya que la información almacenada y consultada es independiente. La única relación que se puede cuidar es sobre los mensajes de notificación en las tablas ESTADO_NOTIFICA, MSJ_NOTIFICA_XML ya que se busca en estas tablas la forma y movimiento que se debe ejecutar dependiendo de la regla almacenada en la base de datos SISA.	RTMsjXml-Dol DOL

Cuadro de Funciones y/o Librerías afectadas (CFL)

Función	T	Objeto	Comentario	Regla
Eje_MsjXml.sh	A	Sh	Arranca el mensaje universal xml	RTMsjXml-Cfl-01
Gprepara_Xml	AC	Proc	Función global en UNIFACE prepara la forma que se ejecuta para el envío de mensajes de notificación	RTMsjXml-Cfl-02
Gcampos_Xml	AC	Proc	Construye el mensaje xml para enviarlo por medio del Jount_xml	RTMsjXml-Cfl-03
GRecupera_Desc	AC	Proc	Recupera descripcion	RTMsjXml-Cfl-04
Encripta_ambiente	A	Java	Función para encriptar el archivo de ambientes	RTMsjXml-Cfl-05
Jxml_in	A,C	Java	Función Java que permita leer Archivos en formato Xml que son enviados por el usuarios dentro de una carpeta IN del usuario sisaxml	RTMsjXml-Cfl-06
Jxml_out	A,C	Java	Realizar un programa en Java que permita leer Archivos en formato Xml que serán enviados por el usuarios dentro de una carpeta OUT_MsjXml	RTMsjXml-Cfl-07
St_EEXml_In	AC	StrProC	Store Procedure recupera información faltante errores de entrada JAVA	RTMsjXml-Cfl-08
FTP_RCP	C	C	Función para la transferencia de archivos ya sea por FTP o Rcp	RTMsjXml-Cfl-09

Cuadro de archivos afectados (Caf)

Archivo	Contenido	Regla
Ambiente_xml	Se almacena en el servidor un archivo plano donde se encuentran todos los ambientes activos para el Xml	RTMsjXml-Caf-01
Eje_MsjXml.txt	Indica al Mensaje Universal Xml que permanezca activo en memoria	RTMsjXml-Caf-02
TEsp_MsjXml.txt	Tiempo de espera máximo para volver a leer la tabla de entrada y salida	RTMsjXml-Caf-03
DTD_MsjXml	Estructura de los Mensajes Xml definidos por DTD	RTMsjXml-Caf-04

Reglas Técnicas

Regla	Contenido	Reglas y/o Tareas Relacionada
RTMsjXml-Ini-01	Para arrancar el Mensaje Universal Xml se debe ejecutar el shell que se encuentra en el ambiente del Portal Empresarial Eje_MsjXml.sh, hay dos modalidades. Tall Instrucción parecida al Crontab solo que esta ejecuta una vez el comando programado para alguna hora en especial. Eje_MsjXml.sh Directamente desde el prompt del sistema operativo por medio del usuario "sis" El shell simula a un demonio que se duerme cada X tiempo al despertar se conecta a la instancia "xml" para revisar si existen mensajes en algún ambiente configurado en el archivo "ambientes.xml"	Ambientes_Activos.doc
RTMsjXml.Dgcl-01	Para el envío de mensajes Asíncronos no se requiere mas que saber la dirección del usuario destino en este caso la aplicación Mensaje Universal Xml no espera un respuesta por parte del sistema externo Sin embargo para el envío de mensajes sincronos se requiere una respuesta por parte del Mensaje Universal Xml a la aplicación externa, es decir el cliente envía un mensaje xml y espera una respuesta del mismo tipo, el programa JAVA revisa la estructura del mensaje, siendo todo correcto se toma en el primer campo DTD el cual corresponde a la ejecución del programa en la tablas de mensajes MSJ_XML en el campo MXML_L_ACCION se encuentra el nombre del programa a ejecutar. El programa en MXML_L_ACCION una de sus acciones es obtener los datos y almacenar la información en las tablas correspondientes, para el caso de los mensajes sincronos es obtener la información y almacenar el mensaje xml en la tabla de salida MSJ_OUT_XML.	Msj Ems Sisa.doc
RTMsjXml-Dgcl-02	Al recibir un mensaje Asíncrono de la aplicación externa esta no espera respuesta del mensaje universal xml, por lo que se ejecuta el mensaje se almacena la información en la aplicación SISA y se termina. En cambio para los mensajes sincronos la aplicación externa requiere una respuesta por parte del Mensaje Universal Xml posiblemente para terminar un proceso en su aplicación en este momento el Mensaje Universal Xml procesa la información	Msj Ems Sisa.doc
RTMsjXml.Dgcl-01	El nombre del archivo xml esta compuesto de la siguiente manera. PRO_CVEEMP_MULT_FOLIINT PRO Ambiente donde se debe almacenar la información enviada a través del mensaje universal xml cveemp Clave de Internet asignada a la empresa para el alta de las quejas Mult Indica a la aplicación mensaje universal xml que el archivo acepto mas de una queja. FolInt Folio Internet para evitar que se pierdan las quejas enviadas por el mismo cliente el tamaño del folio es de 3 caracteres.	
RTMsjXml-Dgf-02	Para que los clientes puedan ingresar al servidor del Portal y a la inversa, para que desde el Portal se pueda llegar al servidor del cliente es necesario que en los FireWall se registren las direcciones IP de los servidores, Los puertos activos para esta aplicación son el 80 para FTP y el 21 http, el detalle de la configuración se encuentra en el archivo de ambientes	Ambientes_activos.doc

Regla	Contenido	Reglas y/o Tareas Relacionada
RtMsjXml-Dgf-03	El cliente pasa la seguridad de la zona militarizada almacenando el archivo Xml en el servidor Portal en la carpeta de entrada IN (PRO_TELMEX_T_10). El programa JAVA de entrada revisa si en la carpeta de entra (IN) se encuentra almacenado algún mensaje, recolecta los nombre en una pila de mensaje ordenándolos por ambiente. A cada ambiente se conecta con ayuda de la tabla AMBIENTES_XML que se encuentra encriptado. Una ves ordenados los mensaje por ambiente se conecta al primer ambiente dependiendo de la prioridad la prioridad que se manejo es PRODUCCIÓN, PRUEBAS, DESARROLLO. Una ves conectado al primer ambiente el programa JXML_IN se conecta al primer ambiente almacenando el mensaje xml en la tabla de Histórico de entradas MSJ_HISIN_XML, sin importar si el mensaje es correcto o tiene errores. JXML_IN por medio de la definición de campos DTD y de la tabla MSJ_XML revisa la estructura del archivo Xml, el primer campo se refiere al nombre del mensaje que se analizará. Si la estructura de campos es diferente que la almacenada en MSJ_XML se considera como error notificando al cliente de este problema por medio de un Mensaje, ya que en esta fase no es posible detectar de que cliente se trata se almacenan los datos en la tabla MSJ_JAVA_XML y ejecutando un Store Procedure (St_EEXml_In) recuperando la información faltante del Xml, este mensaje se guarda en la tabla MSJ_OUT_XML. El mensaje de error se almacena en la tabla histórica de mensajes MSJ_HISOUT_XML. Si no hubiera error en la estructura del mensaje, es decir en la estructura del DTD que es analizado por el JXML_IN se procede a ejecutar el Store Procedure que lleva el nombre del mensaje solicitado, este Store revisa la estructura física de los campos, en este caso se analizan los campos a almacenar si no conciden con los descritos en la tabla MSJ_XML ocurre un error generando un mensaje al usuario almacenando la información en las tablas MSJ_ERRORIN_XML y en MSJ_OUT_XML notificando cual es el error. En caso de que no se detectara error se insertaría la información en la base de datos como nueva queja y/o realizaría un proceso para retroalimentación al cliente creando un mensaje de salida, esta información se detalla en los mensajes para levantar las quejas desde Internet. El programa Java de salida JXML_OUT revisa la tabla MSJ_OUT_XML de mensajes de salida de cada ambiente almacenado en la tabla AMBIENTES_XML activo ya que en la posición Activo = " T " serían los ambientes que revisaría mensajes para enviarlos al cliente por Internet a través del protocolo FTP Este proceso se realiza hasta que se terminan los mensajes en la carpeta de entrada como en la tabla de salida MSJ_OUT_XML, si no se encuentran mensjes el Mensaje Universal Xml se duerme por un determinado tiempo de espera, almacenado en el archivo Tesp_MsjXml.txt.	Ambientes_activos.doc RTMsjXml-Cfl-04 RTMsjXml-Cfl-05 Msj Ems Sisa.doc

Regla	Contenido	Reglas y/o Tareas Relacionada								
RTMsjXml-Dgf-04	Los mensajes están basados en la estructura de definición de tipo de dato (DTD) la cual no cambia ya que se comporta como una tabla de base de datos, en un archivos DTD pueden almacenarse varios registros. Al momento que se lee un Mensaje Xml Se revisa la estructura por si faltaran campos, la manera de analizar esto es separando los mensajes por Tag indicado por el movimiento por ejemplo. El mensaje PRO_TELMEX_T_10, es un mensaje que se desea ingresar al ambiente de producción de la empresa TELMEX, se aceptan varias quejas en el mismo archivo y por último es el mensaje numero 10 que se transmite al mensaje universal xml. Su estructura es la siguiente: <table><tr><th>PRO_TELMEX_T_10</th><th>DTD Alta_queja</th></tr><tr><td><XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> </XML></td><td><XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML></td></tr></table> En este ejemplo se nota que al comparar el mensaje recibido con el DTD falta un campo, CVEINT en la aplicación se realiza con la tabla MSJ_XML, se detecta que falta un campo, por lo que se dispar un error, ya no se continua con el proceso de alta de queja (ALTA_QUEJA) El mensaje original se almacena en la tabla MSJ_HISIN_XML para realizar conciliaciones con el cliente además de enviar un mensaje de notificación acerca del error detectado. El mensaje de notificación por error detectado se construye con el mensaje recibido Agregando el campo faltante con una nota falta información, el mensaje se construiría con ayuda del la tabla MSJ_XML quedando de la siguiente forma. <table><tr><th>PRO_TELMEX_T_10</th><th>DTD Alta_queja</th></tr><tr><td><XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> Falta información </CveInt> </XML></td><td><XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML></td></tr></table> El nuevo DTD construido con el mensaje recibido se almacena en la tabla MSJ_JAVA_XML ya que el programa JXML_IN ejecuta el Store ST_EEXXML_IN el cual obtiene de la tablas EMP_ETE_INTERNET, DIR_OUTIN_XML los datos restantes para la creación del mensaje de notificación de error en estructura DTD almacenado en la tabla MSJ_OUT_XML, como respaldo se almacena el mensaje de error de entrada en la tabla MSJ_ERRORIN_XML Terminado con este proceso se elimina el mensaje de la carpeta de entrada (IN) y se continua con el siguiente mensaje en el mismo archivo u otro de la carpeta de entrada (IN)	PRO_TELMEX_T_10	DTD Alta_queja	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> </XML>	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML>	PRO_TELMEX_T_10	DTD Alta_queja	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> Falta información </CveInt> </XML>	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML>	RTMsjXml-Caf-04 Msj Ems Sisa.doc
PRO_TELMEX_T_10	DTD Alta_queja									
<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> </XML>	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML>									
PRO_TELMEX_T_10	DTD Alta_queja									
<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> Falta información </CveInt> </XML>	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML>									

Diagrama interconexión entre el Cliente, SisaXml, Sisa

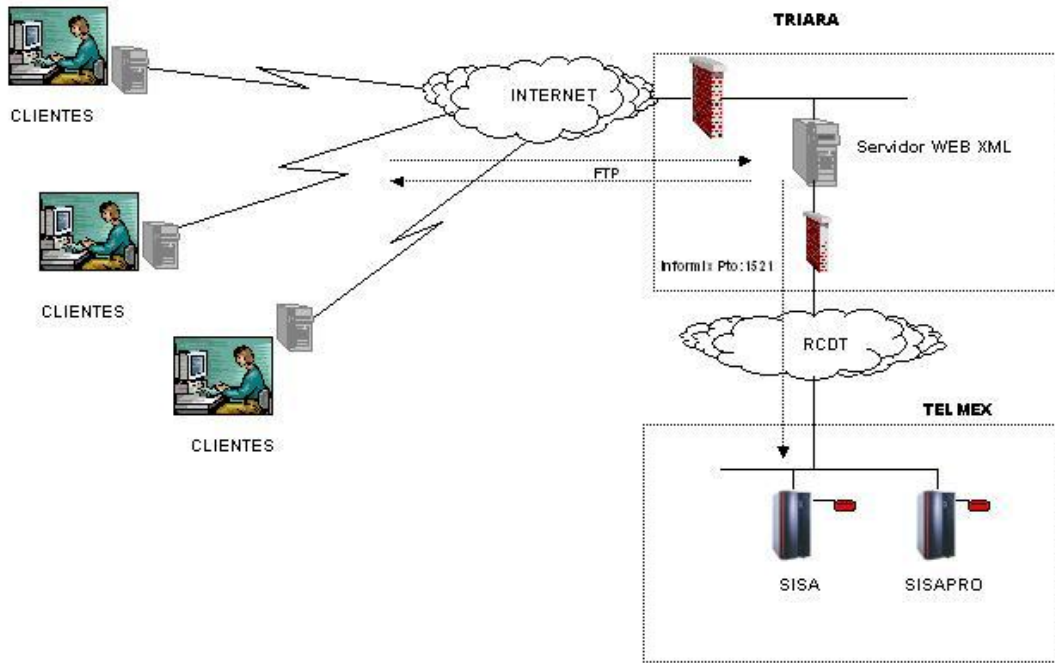


Figura DM 4 Diagrama interconexión entre el Cliente, SisaXml

Servidor Telmex (Sisa)

Sistema Operativo

Herramienta de desarrollo.

Cliente

Servidor

Base de datos

Configuración del Arreglo de Discos VA 7400.

En un esfuerzo de mejorar la velocidad de almacenamiento y la seguridad de su información Telmex esta remplazando el Storage Manager FC60 por uno de mejor tecnología, para lo cual se eligió un arreglo de HP modelo VA 7400.

La siguiente grafica muestra las características del nuevo arreglo de discos donde estará almacenada la Base de Datos SISAP.

Características de la Base de Datos SISA.

A continuación se detalla las características de las estructuras físicas de la base de Datos SISAP, tales como parámetros de configuración de la instancia, segmentos de rollback y las estructuras físicas de la base de datos. Las siguientes tablas tienen como propósito dar una visión global del tamaño y el uso de los espacios reservados para almacenar información de SISA.

Características del Servidor y del Sistema Operativo	
Nombre del Servidor / Dirección IP	Msshba01 IP 13.44.2.52.
Plataforma/Equipo	HP 9000/800 10 CPU's@555Mhz.
Sistema Operativo/versión	HP-UX B.11.00
Memoria física disponible	12 GB

Tabla DM 2 Característica de base de datos Sisa

Servidor Web Xml (SisaXml)

General

En el servidor mwhssn01 con dirección física 10.106.4.129, dirección de Internet 148.223.168.45 esta corriendo el ambiente de desarrollo y pruebas para la aplicación SISA_XML y SISA_WAP

Detalles en el apéndice H, I.⁸

Herramientas Utilizadas⁹

En el apéndice I se describe a detalle las herramientas técnicas, base d datos, sistema operativo configuración de los equipos Sisa, Sisa XML y algunas interfaces desarrolladas

⁸ Ver Apéndice H: Especificación Técnica SisaXml

⁹ Ver Apéndice I: Herramientas Utilizadas

Capítulo 3 Herramientas de supervisión

Pensé desde el inicio del proyecto la forma de recuperar los mensajes tanto de entrada como los de salida los cuales nos ayudarían a conciliar el intercambio de información, con ello detectar de una manera sencilla los posibles errores de comunicación o de los mismo sistemas al generar los mensajes tanto en estructura como en la entrega a cada uno de las herramientas de distribución.

Al inicio solo contaba con la estructura de almacenamiento interno, es decir no se contaba con herramientas digitales para que las áreas de soporte pudieran explotar los mensajes.

Al momento de explotar la información por las áreas de soporte se nos consultaba, extraía el (los) mensaje(s) de manera manual y por medio de mail se entregaba para su explotación, esto se pudo realizar mientras el número de mensajes era mínimo.

Llego el momento que no me daba abasto al recuperar, formatear y entregar los mensajes solicitados por cada interfaz.

Cada vez era más lenta la entrega de los mensajes ya que a parte de la administración mantenimiento y desarrollo de evoluciones relacionadas a la explotación de la comunicación se nos era imposible desarrollar una herramienta de explotación de la información.

Esto fue un arma para que otras áreas de desarrollo tomaran esta iniciativa y propuesta de mejora para el desarrollo de una aplicación que pudiera explotar la información de los mensajes. La cual no tuvo una visión amplia ya que solo se desarrollo la iniciativa de obtener la información de los mensajes interesados de una de las interfaces.

Esto provoco que otras áreas siguieran el mismo ejemplo y desarrollaron su propio modulo de explotación, de la misma manera que la anterior se desarrollo solo la explotación de la intefaz involucrada.

Otros usuarios desarrollaron su propio historial de mensajes, de igual forma solo para explotación de su información que finalmente se trata de conciliar con los historiales del mensaje universal xml (SisaXml)

La herramientas mas utilizadas para explotar la información pero que no contempla en forma general el resto de mensajes son.

Aplicación	Descripción	Desarrollo
Auditor Xml	Muestra los mensajes de la interfaz Pisa Troncales, permite eliminar características para el re-envío del mensaje por parte del usuario Desventajas: - Se instalo en un servidor independiente a la aplicación. - Servidor sin soporte en Windows, si falla el equipo se debe instalar en otro equipo. - Se consulta por medio de autorización al compartir la carpeta. - Se puede instalar en cualquier equipo perdiéndole control de las actualizaciones. - Como es requerida la maquina virtual de java, es complicado para el usuario final instalarlo.	Desarrollado en Java para cliente servidor. Instalado en Windows. Es requerido la maquina virtual de Java para su ejecución.
Consultas SisaXml	Solo es posible consultar mensajes para la aplicación IOM Desventajas - No es posible distribuirlo por ser un Fron End propietario. - La mayor mejora es que se puede consultar desde terminales donde se tenga instalado la aplicación SISA.	Desarrollado en UNIFACE
Supervisor Msj Xml	Una de las mejores aplicaciones pero solo es posible consultar mensajes de la interfaz Pisa LP Desventajas Desarrollado en Web, independiente, tratando de solucionar los problemas de instalación. Se rato de desarrollar como una iniciativa pero no progreso, solos e desarrollo de forma local.	Java Con JDBCMS

Tabla H.1 Herramientas de supervisión consulta, seguimiento y control de mensajes xml

A continuación se muestran los Front End de cada una de las aplicaciones

3.1 Auditor de XML

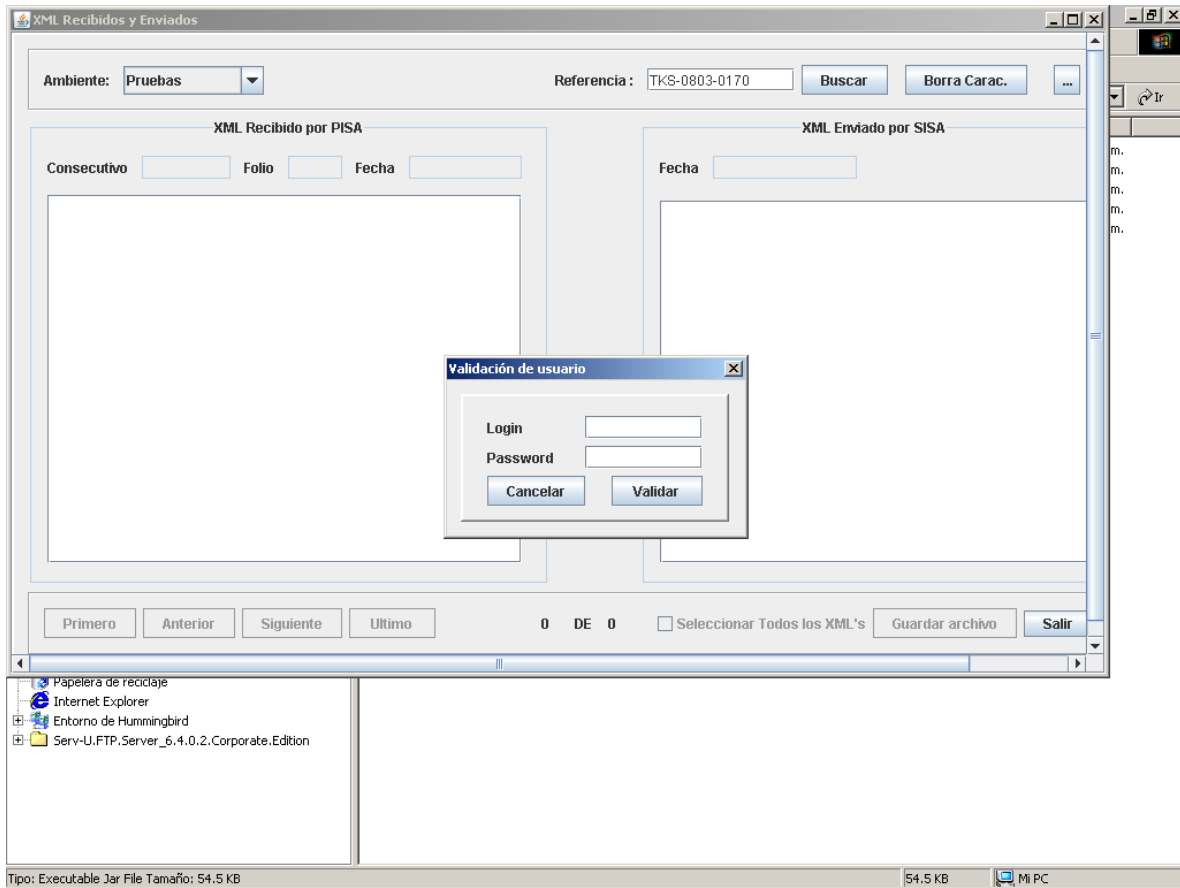


Figura H.1 Auditor Xml

3.2 Supervisor de Msj Xml (Web Services)

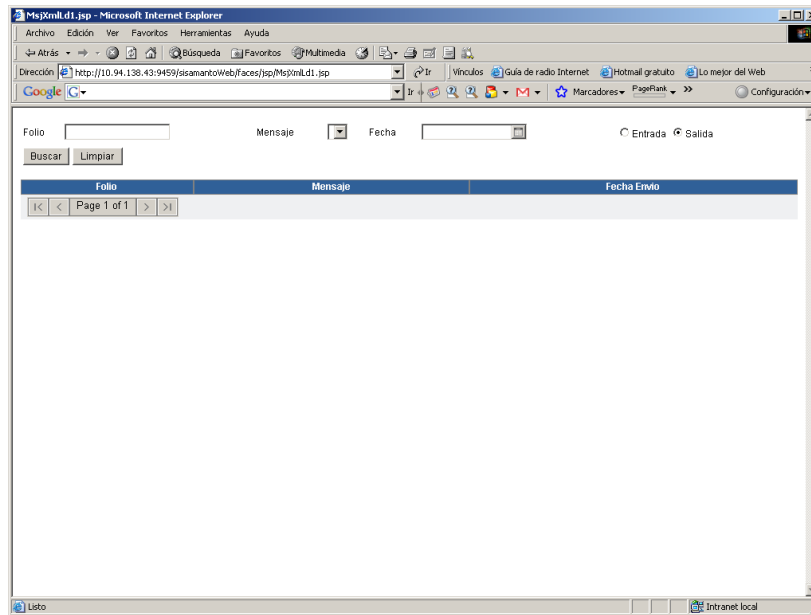


Figura H.2 Supervisor de Msj XML (Web Services)

3.3 Consultas SisaXml

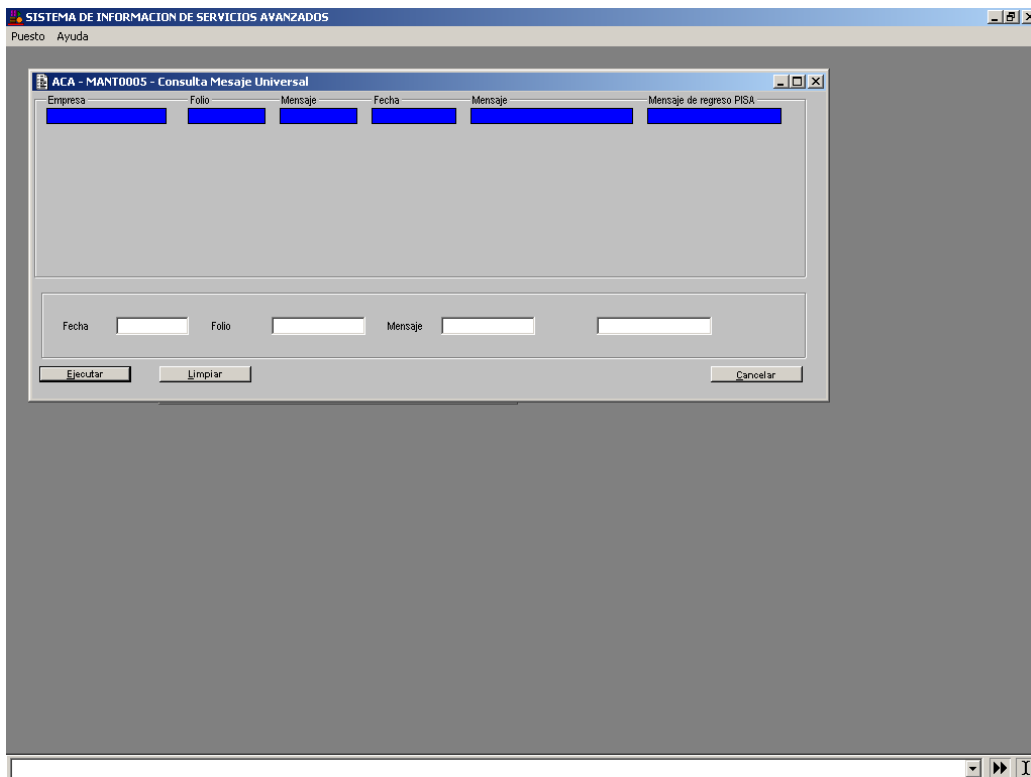


Figura H.3 Consulta de mensajes Sisa Xml

3.4 Conclusión sobre las herramientas de explotación de la información

El poder explotar la información en tiempo y forma es la principal característica de los sistemas de información, esta explotación es indispensable para conciliar, corregir problemas de programación.

Para el caso de una interfaz de comunicación es necesario saber cuales son los mensajes de entrada como de salida; es necesario complementos que ayuden a automatizar y continuar con el flujo del proceso funcional de las aplicaciones relacionadas.

Aunque era posible conciliar los mensajes con el tiempo se salio de control y cada vez es mas lenta la entrega ocasionando un retraso en la solución de los problemas.

También se puede observar que un buen desarrollo trae consigo la posibilidad de mejora que son aprovechados por otros. Pero una falta de visión hace que queden cortos estos proyectos a tal grado que se duplique información.

El duplicar información es uno de los errores que se trata de evitar en el desarrollo de los sistemas de información.

Dentro de las evoluciones y posibilidades de mejora para el Mensaje Universal Xml es el desarrollo de herramientas de explotación el cual sea configurable y lleve el seguimiento de los mensajes. Es un nuevo proyecto que deseo puedas ser aplicado como un Flow.

Un Flow configurable para seguir con el esquema de cajas negras, donde se configure el seguimiento de una mensaje de entrada con la aplicación y con ello tener la posibilidad de corregir errores acarreados por la inexistencia del seguimiento.

Capítulo 4 Evoluciones

4.1 Concentración de herramientas de supervisión.

Como menciono en el punto anterior se desarrollaron algunas herramientas para explotar las bitácoras e histórico de mensajes, las cuales presentan limitaciones.

En esta evolución a SisaXml se pretende agrupar estas herramientas y llevarlas a un ambiente web que pueda ser consultado en la Intranet como en la Internet.

Se selecciono un ambiente web para evitar la instalación de aplicaciones en los clientes de cada usuario, será controlado por medio de accesos (login y password) así como configuración de la cuenta la cual cuente con permisos para mostrar o des-habilitar opciones que el cliente pueda utilizar.

También se selecciono este ambiente para que las mejoras de solución y mejoras al ser implantadas en el ambiente productivo sean transparentes al usuario.

Se contemplaran los mensajes ejecutados por el Web Server por medio de Web Services

4.2 Ejecución procesos por HILOS

Al principio, la trasferencia de mensajes se realizó por medio de archivos planos con un formato administrado XML DTD, el intercambio de mensajes se realizo por medio del protocolo FTP.

Para realizar la actualización de la información en cada sistema se tiene un proceso que esta monitoreando la existencia de mensajes en la bandeja de entrada como en la de salida.

Conforme se han subscrito interfases un solo proceso no fue suficiente, por que todas las interfaces querían tener la mayor prioridad de ejecución. En este caso se tuvieron que dividir los procesos y ejecutar varios.

La evolución es generar un proceso único que tenga la facilidad de analizar y decidir el número de mensajes a procesar con esto generar hilos de ejecución para las diferentes interfaces.

Este proceso se desarrollará para las interfaces que para ellos sea muy caro desarrollar por medio de Web Services, los cuales son utilizados a petición. Finalmente este es un proceso que esta realizando un pooling de una Web Server por ejemplo Tom Cat. También se pretende desarrollar par mantener las interfaces ya existentes

4.3 Ejecución de Web Services externos y consumidores de SisaXml

Ya se cuenta con interfases que SisaXml consume sus Web Services para la actualización de la información de salida, se tiene una primera versión de un Web Services propio de SisaXml donde clientes están consumiendo para actualización de la información en Sisa.

La evolución es generalizar e invitar a interfaces existentes a migrar sus aplicaciones a este tipo de comunicación, las que no sea posible, se mantendrán con el esquema tradicional en la trasferencia de archivos planos administrados.

4.4 Sistema de configuración y administración de mensajes.

Hasta el momento se realiza la configuración de nuevas interfases por medio de una petición y solicitud, se pretende generalizar y automatizar para que usuarios de sistemas por medio de una autenticación (login y password) generen un proyecto o actualicen el mismo.

Ya no sea dependiente de nuestra área para subscribir o manipular los mensajes relacionados a interfaces, esta herramienta deberá estar relacionada con los reportes y herramientas de consulta.

Conclusiones

Se busca una herramienta de comunicación que ofreciera un conjunto de servicios que pudiera ser aplicado a sistemas distribuidos tanto internas como externas.

En un principio se busco herramientas desarrolladas, gracias a sus ventajas y/o desventajas nos dio la pauta para desarrollar un sistema propio.

SisaXml¹⁰ cumple con el objetivo principal de comunicación, se le dio a la aplicación la funcionalidad de ser homogénea; en cuanto al transporte de información y opciones de trabajo de una manera eficiente y clara, el diseño escalable permite relacionarse a las distintas aplicaciones que se crean para mejorar los diferentes medios de interconexión de sistemas

SisaXml tiene una amplia y variada gama de características capaces de hacerlo una herramienta poderosa en el trasporte de información gracias a su manejo de mensajes, continua creciendo y adaptándose a las necesidades de los usuarios

Es claro observar una gran variedad de posibles mejoras. Por ejemplo el reportar resultados en cuanto a la entrega y recepción de mensajes, mejorar y simplificar el ambiente de configuración, en cuanto a los mensajes, los sistemas de intercambio de mensajes como protocolos de comunicación en un nicho que crecerá, ya que siempre se busca una forma mas rápida y fácil de intercambiar información.

Aunque en un futuro no tenga la administración del proyecto, ya que debemos crecer como nuestros diseños, es fácil de adoptar el sistema por las metodologías utilizadas en el diseño y desarrollo genérico utilizadas.

El desarrollo SisaXml, fue muy enriquecedor, aprendí nuevas metodologías de desarrollo y documentación, software orientado a objetos, el beneficio aparte de conocer lo mencionado, fue que me adentre en una pequeña parte de la inmensidad del mundo de las comunicaciones y en el conocimiento de los proceso de la empresa.

Gracias a este reporte sobre el proyecto SisaXml puedo dar ejemplo de los cimientos y excelente directriz obtenida durante mi carrera como INGENIERO EN COMPUTACIÓN egresado de la Universidad Nacional Autónoma de México.

Prueba clara son los proyectos que me han asignado, algunos se encenbran en operación, otros evolucionaron o han sido sustituidos y finalmente los que administro en esta empresa Teléfonos de México SAB de CV

¹⁰ Ver Apéndice J Historia del proyecto Mensaje Universal XML (SisaXml)

Apéndice

Apéndice A Organigrama de Telmex y Sistemas

Al iniciar el proyecto Middleware en el 2003 la estructura organizacional de TELMEX estaba compuesta de la siguiente forma:

Dirección de Planificación
Dirección de Operación y Mantenimiento
Dirección de Ingeniería y Construcción
Dirección de Finanzas
Dirección de Comercialización
Dirección de Sistemas
Dirección de Atención al Cliente

Las anteriores direcciones tienen a su vez divisiones para un mejor control y organización de sus áreas prioritarias.

Las siguientes áreas son las mas afectada al no cumplir con un excelente servicio.

Dirección de Finanzas: debido a que se tiene que pagar compensaciones, recargos y sanciones a los clientes por incumplimiento de su contrato, ello trae por consecuencia que sean menores los ingresos para la empresa. En esa área el contador general puede solicitar que el departamento de Auditoría Interna haga una auditoría para determinar si el retraso es por falta de equipo, capacitación, equipo obsoleto y en qué se ha invertido el presupuesto ya que una parte importante de este se destina precisamente para renovar el equipo y capacitar al personal en las nuevas tecnologías que se tienen en la empresa.

Dirección de Comercialización: entre las consecuencias inmediatas están las de no poder captar más clientes ya que las quejas sobre el mal servicio (deficiencia en la prestación del servicio de comunicación, retraso en la entrega o reparación del problema, reincidencia en las fallas) hace más difícil atraer nuevos clientes y/o retener a los actuales propiciando que se hagan concesiones en precios al momento de renovar o celebrar contratos, cláusulas más rígidas en las penalizaciones con tal de retener a los clientes, y los planes de expansión se retrasan. Una situación importante de recalcar es que los nuevos servicios que se pudieran ofrecer como bloqueo de largas distancias por códigos, descuentos por consumos, etc. No se pueden llevar a cabo por las situaciones anteriores.

Dirección de Ingeniería y Construcción: al querer hacer nuevos proyectos se tiene una restricción en planes de expansión, empleo de nuevas tecnologías ya que no se pueden probar con los clientes actuales ya que la desconfianza es un factor importante

Dirección de Operación y Mantenimiento: Su área de afectación es importante ya que al no contar con los recursos suficientes no se puede dar el mantenimiento oportuno y contar con el stock de partes y tener suficientemente capacitado al personal para la red de comunicaciones. Al reportarles una falla a los diferentes divisionales pueden no contar con más de un recurso (personal ó técnico) y ello puede afectar que un centro operativo o central no pueda trabajar ya que sus equipos de cómputo, comunicaciones, radio, gateway, routers u otro cualquiera no funcionen oportunamente. Además de que las empresas asociadas como UNINET que tiene a su cargo el control, supervisión y arreglo de la Red de TELMEX no podrán cumplir con un ritmo como el exigido, debido a la carencia de recursos.

Dirección de Sistemas: Tiene una afectación importante ya que muchos equipos de comunicaciones trabajan mediante computadoras. La implantación de los sistemas, su mantenimiento, recepción de las fallas en otras áreas como Help Desk sobre sus sistemas de comunicaciones, pruebas de líneas telefónicas, problemas en sus PC's y terminales en todo lo relacionado con los sistemas operativos, software propio de la empresa u homologados como Office, correo electrónico, etc. El área de Desarrollo da solución a los problemas de los sistemas propios de la empresa, hacen las adecuaciones pertinentes a los mismos.

Dirección de Planificación: Una de las afectaciones principales es que los planes de cambio a otras tecnologías, el cubrir una mayor zona con menores costos, la innovación de planes tarifarios, de servicios, series nuevas, expansión hacia otros países, entre otros proyectos, serán frenados o retrasados tanto por la falta de capital, escasez de clientes nuevos con un volumen de facturación importante como por usar una tecnología no adecuada para competir con las nuevas empresas de telefonía de larga distancia como local.

Dirección de atención al cliente: En esta dirección están concentrados los centros de atención y sus diferentes divisiones: CAO, MAC Y CAS, cada una tiene un sector que cubrir y prestar atención a su segmento.

ORGANIZACION ADMINISTRATIVA

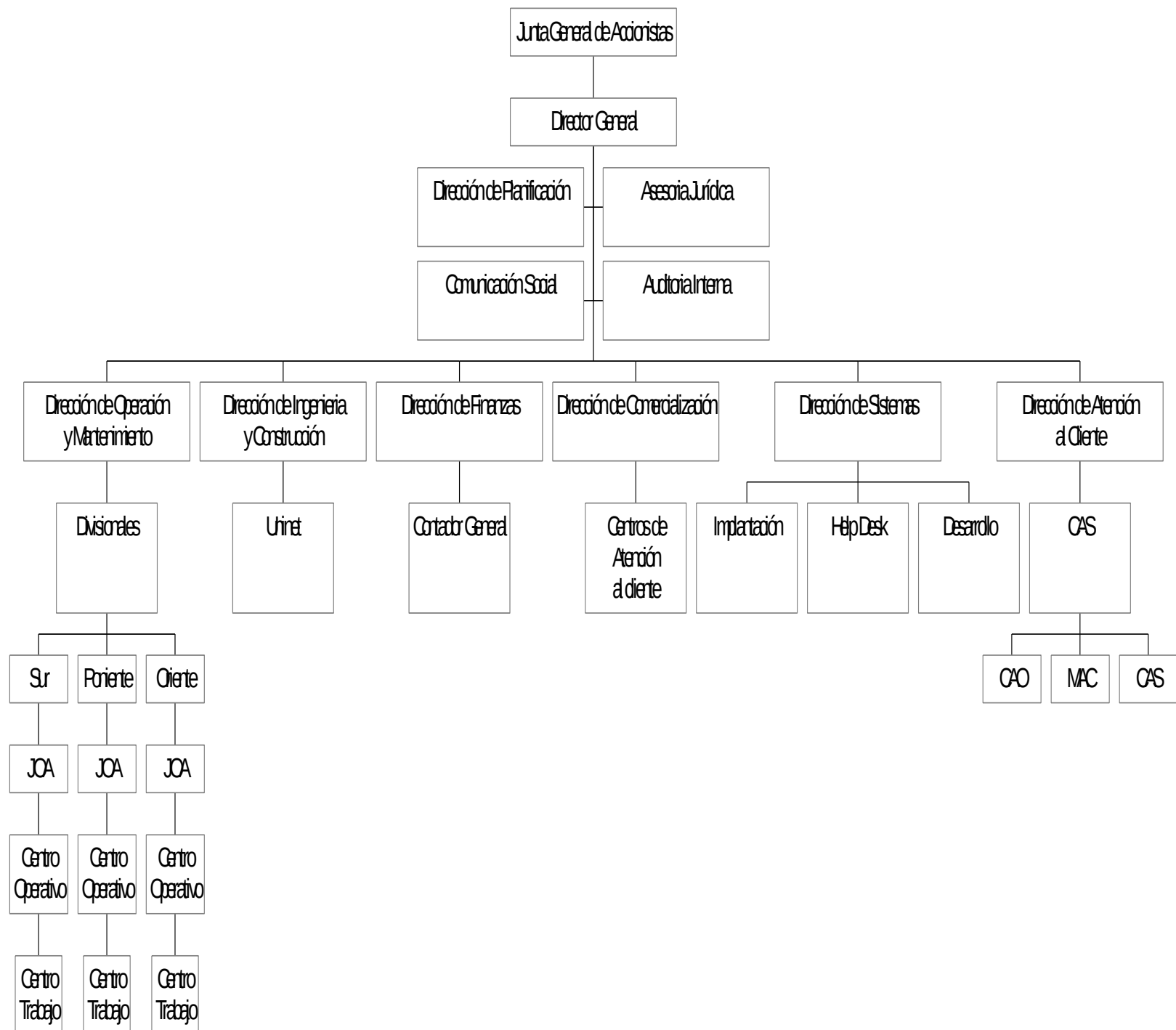


Figura AA1 Diagrama de la organización administrativa.

Apéndice B Descripción General del proceso de atención de una queja

Definición de MAC, CAS y CAO

El MAC (Módulo de Atención a Clientes) es una entidad que cuenta con un sistema automatizado y centralizado para facilitar la comunicación de los grandes clientes con TELMEX, en un solo punto desde cualquier lugar del país. Tiene como prioridad la atención a los “grandes clientes”. Es importante aclarar que el criterio para la selección de los grandes clientes, que deberán ser considerados como tales, no es por la cantidad de llamadas que generen, o por la inversión económica efectuada, ni tampoco por el tipo de servicios solicitados.

Los criterios por considerar son de diferente índole, por ejemplo; si el cliente forma parte de un consorcio industrial, como el Grupo Carso, o si pertenece a un cierto grupo de bancos, o instituciones de jerarquía nacional, internacional o mundial. También son tomadas en cuenta algunas entidades que pertenecen al gobierno y que requieren por sus funciones ser incluidas en el MAC.

La razón de tener registrados a todos los grandes clientes en el MAC, es con la finalidad de ser atendidos en forma eficiente, con rapidez, respeto y amabilidad en un solo punto de contacto y evitarles la inconveniencia de recurrir a diferentes oficinas para solucionarles diferentes tipos de problemas.

Debe quedar muy claro, que todos los clientes son importantes y merecen el mismo respeto, pero se separan para optimar la supervisión, control y una atención personalizada de manera eficiente por el personal asignado a cada área.

El CAS (Centro de Atención y Servicio) tiene como prioridad atender a los que de acuerdo con un minucioso análisis pertenecen a los clientes mayores, los clientes atendidos por el MAC, son muy limitados en comparación con el CAS, que atiende a una cantidad mucho mayor de clientes.

Por está razón serán varios los módulos de CAS, los cuales están programados para implementarse y distribuirlos en las principales ciudades de México, y desde luego cada lugar tendrá uno más y atenderá a poblaciones que estén en su radio de operación. Las ciudades en donde se encuentra un CAS son las siguientes:

México, D.F., Chihuahua, Hermosillo, Monterrey, Guadalajara, Querétaro, Puebla, Mérida

El CAO (Centro de atención a operadores de telecomunicaciones) tiene como prioridad la atención de operadores, estos son todos aquellos que se valen de la infraestructura de TELMEX para brindar un servicio telefónico.

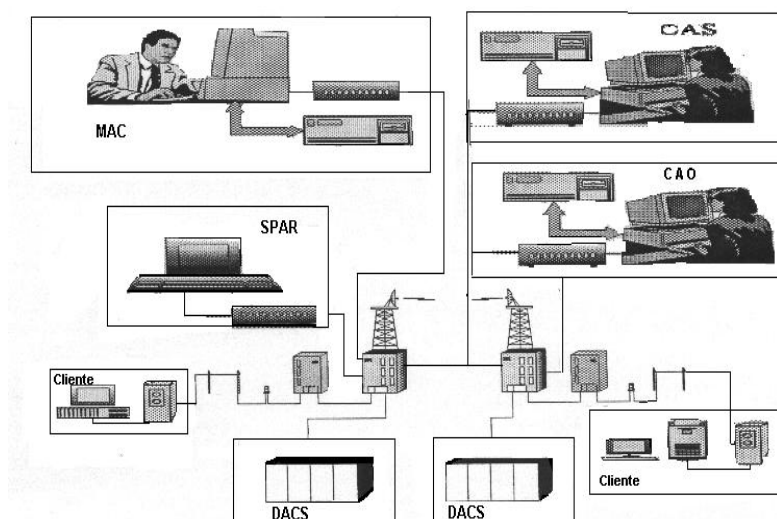


Figura AB1 Centros de atención: MAC, CAS y CAO

Tipos de clientes

Se ha mencionado los tipos de clientes que serán atendidos por los módulos MAC, CAS y CAO, pero los clientes que no sean clasificados para ser atendidos por ellos, deberán ser asignados a otro tipo de módulo que es 050.

La determinación de qué tipos de clientes deberán ser atendidos por alguno de los módulos MAC, CAS, CAO o 050, se basa en estudios aplicados por TELMEX y el cual se conoce como demanda de servicios. Para poder medir la demanda y así satisfacer las necesidades de servicio que requieren los clientes se lleva a cabo una exhaustiva investigación de mercado.

Del resultado de este estudio se determinan las zonas para la planeación las cuales se dividen en:

Residenciales, Comerciales, Industriales

Con base en este criterio, los clientes de casas solas o condominios horizontales se clasifican en:

Residencial Tipo "A"	Más de 250 metros cuadrados construidos
Residencial Tipo "B"	De 160 a 250 metros cuadrados
Residencial Tipo "C"	De 120 a 160 metros cuadrados
Residencial Tipo "D"	De 100 a 120 metros cuadrados
Residencial Tipo "E"	Menos de 100 metros cuadrados

Las zonas consideradas residenciales, exigen pequeños comercios como son las tiendas de abarrotes, misceláneas, pequeños supermercados, tiendas de calzado, tlapalerías, y muchos pequeños talleres de diferentes giros comerciales. Cuando se conjuntan tantos tipos de clientes, se dice que la demanda es heterogénea.

A las zonas comerciales se les ha clasificado según del número de dependientes que tienen los clientes como a continuación se indica:

Comercial de primera	De 20 empleados en adelante
Comercial de segunda	De 5 a 20 empleados
Comercial de tercera	De 1 a 5 empleados

Cuando las líneas solicitadas sean en su mayoría troncales, servicios de Conmutadores, líneas privadas o algunos servicios especiales, se les designa como industrial con la siguiente clasificación:

Industria pesada	Más de 80 empleados
Industria mediana	De 21 a 80 empleados
Industria ligera	De 1 a 21 empleados

Dependiendo de todos estos estudios, los clientes se clasifican en orden de importancia con la finalidad de integrar a cada cliente en el módulo correspondiente.

Funciones del MAC, CAS y CAO

Las funciones del MAC, CAS y CAO son específicas. Ya se ha indicado que la más importante es la atención a los clientes en un solo punto. Pero el saber qué es y qué hace, facilita su comprensión para operarlo con más confianza y así poder brindar con seguridad y firmeza un mejor servicio, lo que dará como resultado que: al beneficiar a un cliente se ganarán más clientes.

El MAC, CAS y CAO atienden los requerimientos de los clientes y la forma de atenderlos es descrita a continuación; desde que se genera un reporte hasta su liquidación.

Recepción de llamadas y tramitación de quejas: Recibir llamadas de los clientes para inmediatamente generar un reporte de mantenimiento y proporcionarle un número de control.

Si el cliente requiere algún tipo de información, ésta deberá ser proporcionada de manera clara y concisa sin dejar dudas, pero si el tipo de información implica datos que sean del área técnica de mantenimiento, o del estado de la instalación de un nuevo servicio o línea, la llamada será transferida hacia alguno de los técnicos a cargo del problema.

Se debe tener presente que según la magnitud del problema y el tiempo que se tome para solucionarlo entra en el proceso de escalación, y si es necesario se le puede llamar al gerente como apoyo para solucionar cualquier tipo de requerimiento del cliente. Se puede presentar el caso en que el cliente reclame bonificación económica por el retardo en solucionar su problema.

El servicio proporcionado por estos centros de atención funciona las veinticuatro horas del día, los siete días de la semana durante todo el año.

Otra más de las funciones son las actividades de prueba, monitoreo y estado de avance de un servicio o instalación. Cada una de ellas es descrita a continuación:

Pruebas: El personal asignado a probar, operará con los reportes de mantenimiento que la recepción de llamadas le transfiera a través del sistema y será el encargado de seccionar y ubicar el lugar y tipo de problema, por medio de equipos de pruebas.

Si el problema está determinado y ubicado en el equipo del usuario, éste tendrá que estar informado y hasta que el cliente esté convencido de que el problema está en su propia área de responsabilidad se cerrará el reporte.

Cuando el probador localice la ubicación del problema en alguna de las áreas de la planta telefónica de TELMEX o no, deberá coordinarse con cada uno de los técnicos responsables de las áreas involucradas y solucionar el problema.

Pero si el problema no es localizado o si ya está focalizado deberá efectuar un seguimiento y coordinar con los técnicos involucrados con el problema, mientras no se encuentre el daño, el probador informará al cliente del estado de avance continuamente hasta que se solucione.

Cuando ya se arregle la falla, aparte de llamar al cliente para informarle; se comprobará junto con él y a su entera satisfacción, la calidad del servicio restablecido, inmediatamente se almacenará los nuevos datos y se cerrará el reporte.

Se tendrá especial cuidado de anotar la fecha y hora, por si el cliente reclama una bonificación económica.

Funciones de mantenimiento: Se efectuarán funciones de monitoreo, para proporcionar un mantenimiento constante, esto quiere decir que en algunos casos no habrá necesidad de esperar hasta que los clientes reporten una anomalía en sus sistemas de comunicaciones telefónicas, ya que con un constante mantenimiento se puede monitorear la calidad del servicio en las líneas de los clientes que están asignados al sistema del MAC, CAS y CAO.

De esta forma se puede detectar un problema antes de que afecte el servicio de telecomunicaciones, con lo que en algunos casos se podría evitar que el cliente genere un reporte de queja.

Con este tipo de mantenimiento se incrementará la confiabilidad de los clientes hacia la calidad del servicio que MEXTEL les proporciona.

Análisis de reportes: Esta función es también muy importante, ya que se podrá efectuar el análisis de los reportes de instalaciones y servicios que se generan al MAC, CAS y CAO.

Se verificará que todas las pruebas de aceptación estén completas, para poder ser recibidas en el área de mantenimiento, con lo que se evitará la duplicación de reportes y la dilación que entorpece el servicio proporcionado a los clientes.

Se compilarán las estadísticas e historial de los reportes, de todos los clientes, que facilitarán la identificación de los problemas que constantemente se repiten, y que con el tiempo se vuelven crónicos.

Estadística de datos: El propósito de realizar el análisis de la incidencia de fallas, es determinar su existencia y ubicación, una vez localizada la falla hay que solucionarla inmediatamente, a fin de proporcionar un servicio de óptima calidad para el cliente.

Este tipo de fallas se pueden detectar cuando se revisan los reportes estadísticos de queja, en donde se presentan continuamente, que tienen poca duración o son intermitentes.

Entre los beneficios generales para la solución de un reporte de queja, se tienen las siguientes características:

- Veinticuatro horas de atención ininterrumpidas
- Atención los siete días de la semana
- Atención todo el año
- Cobertura nacional
- Recursos dedicados
- Respuesta de información ágil y veraz
- Especialistas que entienden sus necesidades
- Economía y eficiencia
- Un solo punto de atención para servicios y reportes
- Informe del estado de avance de la reparación
- Pruebas de nuevas instalaciones
- Informe del estado de avance de las nuevas instalaciones
- Verificación y prueba con el cliente de la reparación efectuada
- Información sobre el estado de avance de la instalación de un nuevo servicio vía telefónica y a nivel nacional
- Pruebas de medios de transmisión y equipos terminales
- Información sobre la facturación a los clientes.

Servicios y enlaces

Los servicios y enlaces de TELMEX utilizan tecnología electrónica digital de punta, transmiten voz y datos con calidad Internacional, tienen inmunidad al ruido y a interferencias electromagnéticas, garantizan la confidencialidad de la información, cuentan con una infraestructura disponible en todo momento para su contratación.

Servicios utilizados en el proceso de atención de fallas

TELMEX cuenta en la actualidad con servicios que se dividen en cuatro categorías:
Servicios digitales:

Troncales digitales.
 Los enlaces (locales, nacionales, internacionales y cruce fronterizo).
 Red satelital, E0's, E1's.
 Líneas de alta calidad (LAC).

Servicios especiales
 300, 500, 700, 800 y 900.
 DID's, DS0's, DS1's.

Servicios analógicos:
 DR, PDR, PDS, PR, PS, DS, X25, Frame relay

Servicios básicos
 Línea directa analógica.
 Línea privada.
 Troncal analógica.
 Servicios de interconexión.
 E1 de interconexión.
 Puerto de interconexión.
 E1 de reventa.

Servicios y enlaces

Enlaces urbanos (local): el servicio básico es el enlace urbano que se establece entre dos clientes de la misma ciudad, los cuales disponen de una línea de dos hilos de cobre de su domicilio a la central local que es la responsable de la interconexión del abonado A (quien origina la llamada) y el abonado B (quien recibe la llamada).

El enlace a la central local se efectúa vía las redes secundaria y primaria, la red principal o primaria generalmente es subterránea y la red secundaria se transporta vía postes hasta el domicilio del abonado.

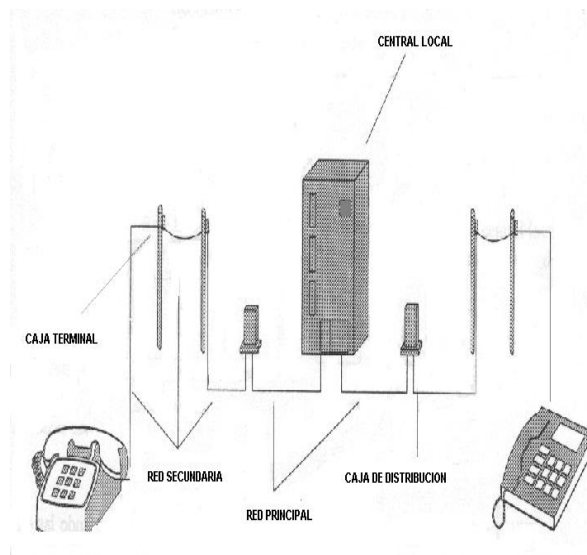


Figura AB2 Enlace urbano

Enlace interurbano L.D.: este enlace es idéntico al urbano se genera de un abonado A, el cual conecta a la central local y está para lograr el enlace se interconecta a una central de L.D., vía la red troncal, la cual tiene comunicación con una central de L.D. distante, ya sea vía microondas, fibra óptica o satélite, a su vez esta central se interconectará con una central local para establecer la comunicación con el abonado B que se ubica en otra ciudad

Estos enlaces son del tipo de voz o analógicos (aún cuando las centrales locales y de L.D. se interconectan con sistemas digitales) y son atendidos por el sistema de quejas 050.

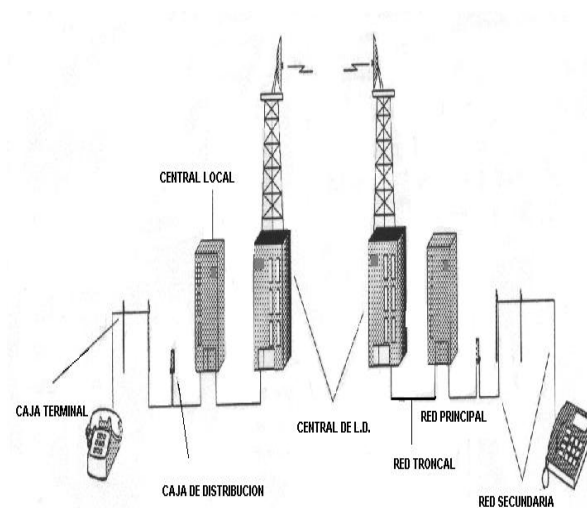


Figura AB3 Enlace interurbano de L.D.

Proceso de Atención

El objetivo de los centros de atención (MAC, CAS y CAO) es proporcionar la mejor atención y servicio a los clientes que a ellos se dirigen, a través de un reporte de queja del cliente al centro de atención. En la actualidad el proceso de atención es como sigue:

- ✚ El cliente se comunica vía telefónica al centro de atención que le corresponde, para realizar un reporte sobre un servicio que está presentando alguna falla o a consideración del cliente para un mantenimiento preventivo del servicio. El cliente es atendido por una operadora.
- ✚ La operadora del centro de atención tiene que validar la empresa asociada al servicio, que el cliente está reportando, esto mediante una lista impresa en papel, donde se encuentran el total de empresas que son atendidas por el centro, si la empresa se encuentra registrada se procede a la recepción del reporte, en caso contrario se transfiere la llamada al posible centro de atención.
- ✚ Se registran información necesaria para el seguimiento, nombre de la persona que está reportando, teléfono donde se le encuentra, nombre de la empresa relacionada al servicio, tipo de servicio que se encuentra dañado, dirección de la punta en A, población de la punta en A, dirección de la punta en B, población de la punta en B y dependiendo del tipo de servicio algunos datos adicionales que facilitan la reparación de la falla, además registra la fecha y hora de inicio del reporte.
- ✚ La boleta se pasa a un supervisor para que asigne a un técnico para la atención, la asignación de éste la registra en la misma boleta.
- ✚ El técnico realiza el diagnóstico con el equipo de supervisión existente en el centro de atención, para detectar dónde se encuentra la posible falla, con ello realiza un reporte al centro operativo correspondiente para la solución del problema. Todo el seguimiento es escrito por el técnico en la boleta.
- ✚ El centro operativo se comunica en cuanto tiene solucionada la falla, con el técnico que lleva el seguimiento.
- ✚ El técnico del centro de atención realiza las pruebas necesarias para verificar que la falla quede solucionada, registra en la boleta cuál fue la falla encontrada y el centro operativo que tuvo la responsabilidad final.
- ✚ El técnico se comunica finalmente con el cliente para liquidar el reporte.
- ✚ Se capturan los datos de liquidación, datos como: falla encontrada, entidad responsable de la falla y fecha/hora de terminación de la queja.

El diagrama siguiente se muestra el flujo actual de la atención de un reporte.

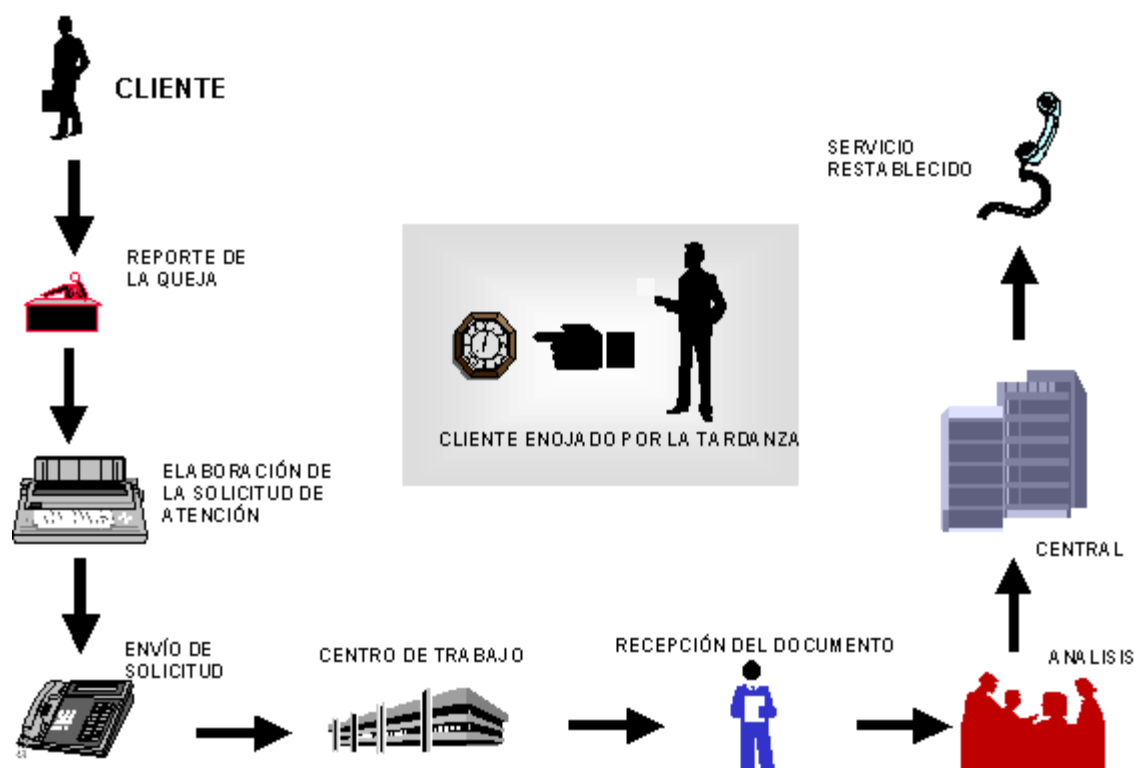


Figura AB4 Flujo de la atención de un reporte.

Apéndice C Descripción general: Captura de una queja por medio del Sistema SISA.

SISA (Sistema de Información de Servicios Avanzados) permite la gestión de los servicios avanzados desde la firma del contrato de servicios por el Cliente hasta su instalación, su puesta en servicio, la facturación y el mantenimiento.

Modulo	Descripción
Administración de contratos	El alcance del ACO (Administrador de Contratos) es para cualquier entidad de TELMEX cuya misión sea la verificación de contratos, generación de órdenes de servicio o emisión de datos a facturación. Las diferentes áreas implicadas en SISA son: El área de administración de contratos. El área de instalación. El área de mantenimiento. El área de facturación. SISA está a cargo de asegurar la supervisión del flujo de las siguientes funciones: Tratamiento de la instalación de servicios contratados. Definir y mandar los datos de facturación de servicios instalados. Procesar los servicios después de su venta.
Instalación	Tiene la finalidad de agilizar el seguimiento de la instalación/desinstalación de un servicio o servicios a través de los tres diferentes papeles que intervienen en el módulo: Responsable de Instalación RIN, Responsable de Servicio RSE y Responsable del Centro de Instalación RCI
Mantenimiento	Consta de una serie de programas diseñados con el fin de facilitar el manejo de la Recepción de Quejas, Creación de los Expedientes de Mantenimiento de Servicio y de los Expedientes de Falla por las Áreas del Módulo de Atención a Clientes (MAC), Centro de Atención y Servicio (CAS) y Centro de Atención a Operadores de Telecomunicaciones (CAO), así como la recepción y liquidación de las Ordenes de Intervención por los Centros Operativos correspondientes. Este dominio será utilizado por todas las Areas de mantenimiento de la empresa tales como las entidades MAC, CAS y CAO distribuidos en toda la República, así como los Centros Operativos de las Direcciones Divisionales y de LD (Larga Distancia)

Tabla AC 1 Módulos del Sistema de Información de Servicios Avanzados

Diagrama de contexto Sisa –Mantenimiento

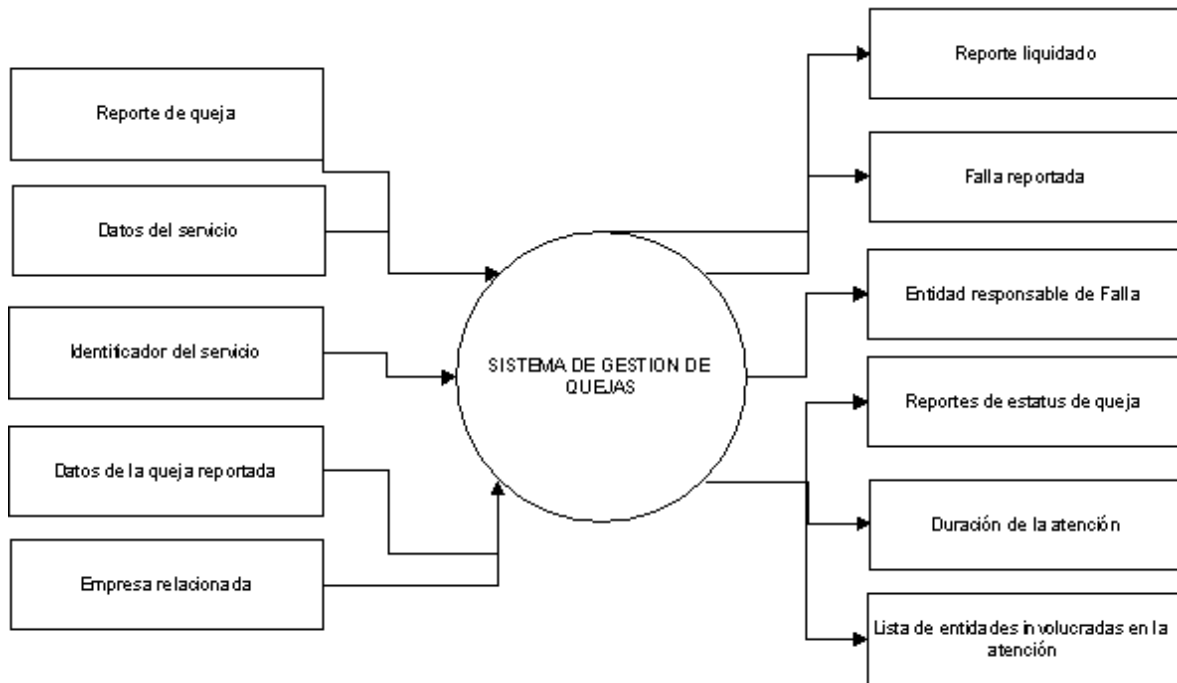


Figura AC 1 Diagrama de Contexto Sisa – Mantenimiento

Alta, Seguimiento y Liquidación de una expediente de mantenimiento y servicio EMS (Queja)

A continuación se muestra un extracto del manual de usuario para los diferentes módulos de la aplicación SISA.

No se pretende explicar el funcionamiento detallado ya que no es el fin de este reporte, el objetivo de dicho reporte es explicar el análisis, desarrollo y puesta en producción del Mensaje Universal SisaXml. Pero SISA es la relación fundamental para la generación del sistema SisaXml.

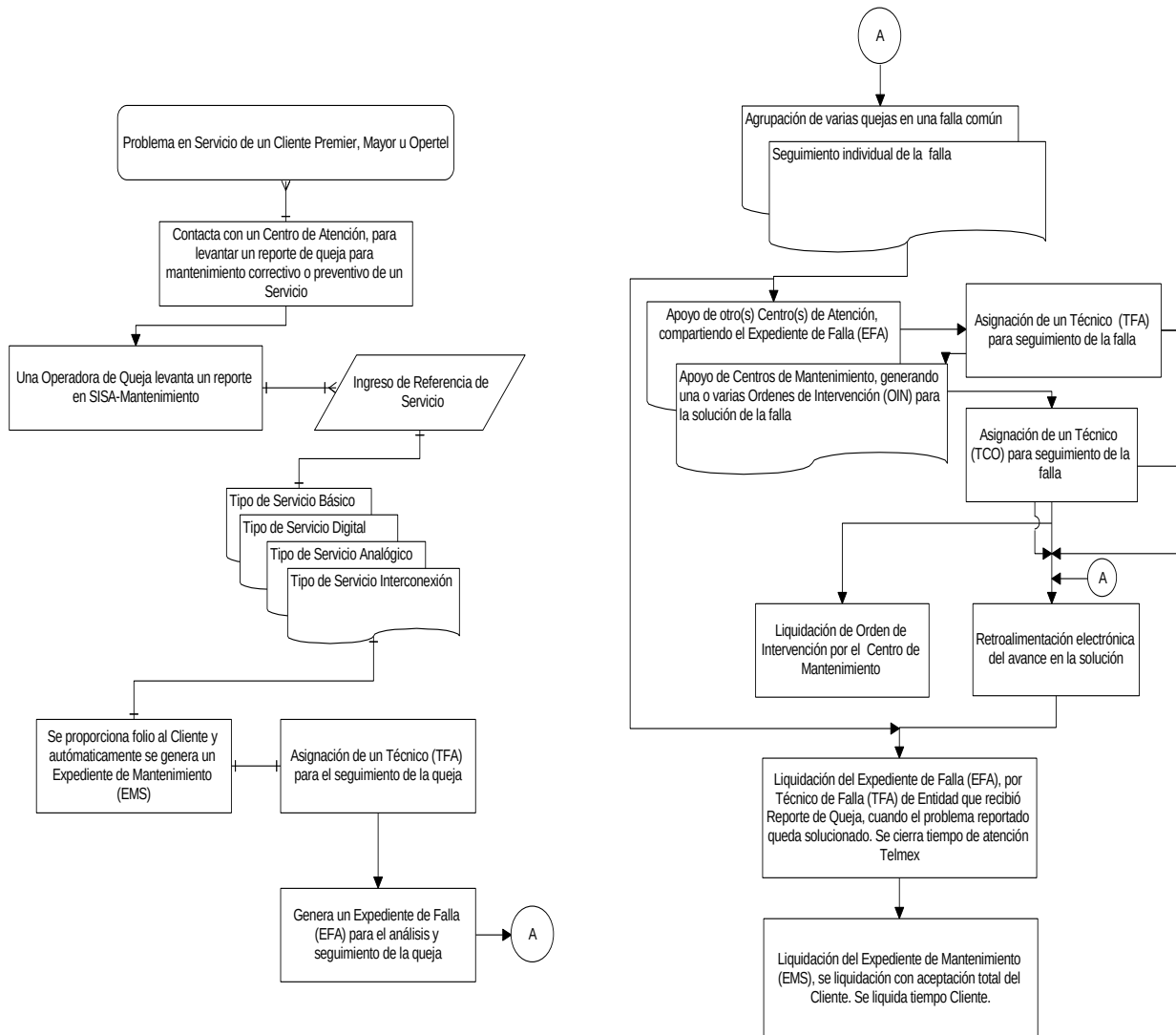


Figura AC 2 Diagrama Proceso alta seguimiento y liquidación de una queja

Inicio de sesión de Trabajo.

Para entrar al sistema SISA deberá acceder primeramente Windows y ubicar el icono del software,



posteriormente sitúe el cursor (la punta de la flecha del mouse) en el icono SISA., y haga dos clicks rápidos para iniciar el Sistema.

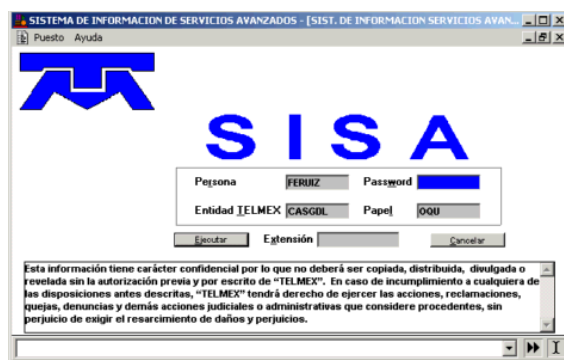


Figura AC 3 Ingreso al Sistema Sisa

Posiciónese con la tecla <TAB> o sitúe el cursor (la punta de la flecha del mouse), en la caja Persona y digite su nombre (el registrado con el Administrador del Sistema), éste se compone de 8 caracteres, la primera letra de su nombre de pila y las restantes del apellido paterno, ejemplo: para Alfredo Guzmán será AGUZMAN.

Lista de contactos en proceso

Muestra las quejas reportadas por entidad

SISTEMA DE INFORMACION DE SERVICIOS AVANZADOS

Expediente Entidades Consultas Ayuda

OQU - SISA3030 - Lista Contactos en Proceso

Total de Quejas: 74

Folio	Fecha y hora	Empresa	Referencia	Desc.Serv.	Interlocutor	Puntaje	Operador(a)	Apoyo	Fase-Est
10001392	19/05/10 11	DEPORTIVO	TKS-0408-064	ALTA DE	123.123	SAN PED	JPLACID		OPE-
10001391	19/05/10 11	RADIO MOV	A01-0408-178	CALVAR	ITAQUE	MEXICO D	CUAUTIT		OPE-
10001390	19/05/10 11	FORD MOTO	ST1-0909-002	TEST	DIAZ FE	MEXICO D	MEXICO D		OPE-
10001389	19/05/10 11	INVE CASA	D04-0409-000	CORPOI	REYES C	MEXICO D	MEXICO D		OPE-
10001388	19/05/10 11	MONEX CAS	D04-0103-009	DEMARQ	Q Q Q	MEXICO D	MEXICO D		ALT-AF
10001387	19/05/10 11	UNINET	A02-0411-024	LIVERPO	ARREOL	MEXICO D	MEXICO D		OPE-
10001386	13/05/10 11	TELEFONOS	ADSL-545454		ESTRAD				SNID
10001385	13/05/10 11	TELEFONOS	ADSL-875211		DEL AN				SNID
10001384	13/05/10 11	TELEFONOS	ADSL-875555		DEL AN				SNID
10001383	13/05/10 11	TELEFONOS	ADSL-487755		DEL AN				SNID
10001382	13/05/10 11	TELEFONOS	ADSL-654545		ESTRAD				SNID
10001379	21/04/10 21	ACCURIDE	LD1-0309-001	10.TRON	ROSAS J	CIENEGA			OPE-
10001379	21/04/10 21	ACCURIDE	LD1-0309-001	10.TRON	ROSAS J	CIENEGA			OPE-

Botones: Abrir, Alta, Baja, Refrescar, Archivo TXT, Cerrar

Figura AC 4 Captura de los expedientes de mantenimiento y servicios por medio de la Operadora de Queja

Descripción de un contacto

Permite el alta de la queja, por medio de la referencia, se obtiene la información del cliente, esta no debe llevar mas de 2 segundos en recuperar la información ya que el cliente esta en la línea telefónica.

DQQU - SISA3010 - Descripción del Contacto

CLIENTE

Origen llamado: ☐ Empresa ☐ Corporativo ☐ Carrier

Ref. Servicio:

SNI:

Tipo Serv.:

Busca Servicios

Detalle Servicio

INTERLOCUTOR

Interlocutores Of.:

Apellidos y Nom. (P): (N): (N): (N):

DATOS GENERALES DE LA FALLA

Prob. reportado:

Folio: 1

☐ De Cliente a Telmex ☐ De Telmex a Cliente

Prioridad:

Horario Aconte:

AMT:

Reportes Similares:

Observaciones:

DATOS DEL SÍMPTOMA

Aceptar Duplicar Limpiar Cerrar

Figura AC5 Alta del expediente de mantenimiento y servicio EMS

Elaboración de solicitud de atención (RMA)

SISTEMA DE INFORMACIÓN DE SERVICIOS AVANZADOS

Expediente Entidades Consultas Ayuda

IRMA - SIS3A3000 - Lista de EMS/EFA's Compartidos en Proceso

EMS: [Filtro] Tipo: [Filtro] Estado: [Filtro] Referencia: [Filtro] Fecha: [Filtro]

EMS	Tipo	Estado	Referencia	Fecha
10001392	TKD	En mantenimiento	10001392	26/06/10

IRMA - SIS3A3020 - Seguimiento a la Queja

FOLIO

Folio	Estado	Referencia	Fecha
10001392	En mantenimiento	10001392	26/06/10

SERVICIO

ALTA DE 10 TRONC

DATOS DEL CLIENTE

REPORTIVO SAN AGUSTIN

OBSERVACIONES

BAJO TRAFICO /

TELEMX

TECNICOS

RESPONSABILIDAD

Tipo Serv

C. Origen

Prioridad

Obs. EFA

Est. EFA

Liq. EMS

Liquidar

Tiempo de penalización

Aceptar **Cancelar** **Exp. de Fallo** **Reincidencias** **Cancelar**

Figura AC6 Seguimiento y detalle del supervisor de Quejas (Responsable de Mantenimiento Servicio)

Asignación de Técnico de seguimiento (TFA).

El técnico de falla realiza pruebas eléctricas, decide si requiere apoyo de planta o de los diferentes sistemas como es PISA, Clarify, Cobo, Interred por medio de una Orden de intervención OIN la cual es atendida por el TCO (Técnico de operación)

SISTEMA DE INFORMACION DE SERVICIOS AVANZADOS

Expediente Entidades Consultas Ayuda

RMA - SIS3160 - Descripción de un Expediente de Falla

Folio: 10001392 F. Cte: F/H In: Queja 26/05/10 10:10 Pri: 0 Obs: RMA: PLACIDO Reincidencia: EFA's Asig: 10001392

Tipo Servicio: Ref. Servicio: SNI: Empresa: Grupo Corp.

Prob. Report: BAJO TRAFICO / Horario Acc: Acus. Rec: Queja

Observación: Interlocutor: 123 123 123 Tel: 21342

Medio Trans. A: Coaxial Medio Trans. B: Empresa Sit. A: Empresa Sit. B:

DATOS DEL SERVICIO

Domicilio: Lazaro Cardenas Pte. Col. Residencial San Agustín No. Ext. 2450 No. Int.

Número de Grupo: (81) 8131900

Ti's afectadas: Ti's asociadas: Número origen asoci: Número destino: Referencia Capburda:

EXPERIENCIA DE FALLA

Folio: 10001392 C/M Orig: MAC F/H Asignac. Téc: 26/05/10 10:03 Observaciones: Actualiza Obs: F/H Creación del EFA: 26/05/10 10:10 Envío: Comentario Seg.

DIAGNOSTICO

Duración: CAO CASCHI CASGOL CASIMMO Detalle Otro: CAS Dur. Rest TELMEX Cons. Falla Encuentra Diagnostico TND

Pendiente por: Estatus: Cons. Serv. nel. Plus:

Tiempo Atención: 3:00 Tiempo Ab. Restante: 147:29 Tiempo Penalización:

Aceptar Contato Orden Int. Servicio Liq/Can. EFA Ref. Asoc. Martis Cancelar

Figura AC7 Seguimiento y detalla de la Falla (Expediente de Falla)

Técnico de Operación y Gestión

Apoyo de planta, se realizan pruebas en campo, centrales, edificios de cliente, acometidas.

SISTEMA DE INFORMACION DE SERVICIOS AVANZADOS

Expediente Entidades Consultas Ayuda

RMA - SIS3210 - Descripción de una Orden de Intervención

Folio: 10001378 F/H In: O: 21/04/10 20:14 Prioridad: 0 Obs: RMA: MARES ANA LILIA -> M

F/H Emisión: 18/05/10 11:21 F/H Tco: Estado: Inicial F.E. Serv.: DPE

T. Serv.: TSKO Servicio: BND-9866-8442 Centro M. Origen: MAC Centro Oper.: CAMEOT

Empresa Sitio A: ABASTECEDORA DE OFICINAS Empresa Sitio B: LIQUIDACION

TRASLADO DE TCO: REPARACION

Envío de personal a pta A: Tco. Centro

Envío de personal a pta B: Acus. Rec.

Envío de personal a central: F.H. Reparación

PROCEDIMIENTO TCO: Duración:

F.H. inicio proced.: Duración:

F.H. fin de proced.: Duración:

RESPONSABILIDAD: Reparación Extensión:

Responsabilidad: LIQUIDACION

Falla encontrada: RMA: MAC, PLACIDO JUAN 18/05/10 Actualiza Obs. Envío personal al Sitio del Cte.

Solución Dada: Observaciones:

Tiempo Atención: 3:00 Tiempo Ab. Restante: 977:23 Tiempo Penalización:

Aceptar Imprimir OIN Servicio Tramos Liq/Can. EFA Cons. serv. nel. Plus Cancelar

Figura AC7 Seguimiento y detalle de una Orden de Intervención.

Requerimientos para el desarrollo

Hardware

PC con Procesador Pentium a 133 Mhz.
 32 MB de Memoria RAM como mínimo.
 Monitor a color.
 Tarjeta adaptadora VGA, o Super VGA para utilizar gráficos.
 Mouse compatible.
 120 MB libres de espacio en disco duro.
 Impresora (Opcional).

Comunicaciones.

Nodo de Red para acceso a la Red Corporativa de Datos Telmex (RCDT).
 Tarjeta de Red para PC (Elite 16T Ultra).

Software

Hostname sisapr01
 IP Adress 10.192.3.65
 Marca HP
 Modelo 9000/800/SD16A
 CPUS 12 de 1.0 Ghz c/u
 Memoria 32 Gb
 Sistema Operativo HP-UX VERSION: B.11.11 64 Bit
 Manejador de Base de datos Oracle Release 9.2.0.4.0

Requerimientos para la operación Cliente

Hardware

PC con Procesador Pentium a 133 Mhz.
 32 MB de Memoria RAM como mínimo.
 Monitor a color.
 Tarjeta adaptadora VGA, o Super VGA para utilizar gráficos.
 Mouse compatible.
 120 MB libres de espacio en disco duro.
 Impresora (Opcional).

Comunicaciones.

Nodo de Red para acceso a la Red Corporativa de Datos Telmex (RCDT).
 Tarjeta de Red para PC (Elite 16T Ultra).

Software

Sistema Operativo Windows 95 o superior.
 Licencia UNIFACE Seven de distribución bajo Windows para desarrollo con drive de ORACLE y Polyserver

Apéndice D Definición de Middleware

El middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).

El middleware nos abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware.

Por lo general el middleware del lado cliente está implementado por el Sistema Operativo subyacente, el cual posee las librerías que implementan todas las funcionalidades para la comunicación a través de la red.

Orígenes

Los middleware han aparecido de manera relativamente reciente en el mundo de la informática, hoy en día tecnologías móviles (celulares) están aplicando esta tecnología para el soporte de llamadas VOIP. En el pasado ganaron popularidad en la década de los 80, ya que eran la solución de cómo integrar las nuevas aplicaciones con los sistemas heredados (legacy systems), en todo caso, el termino ha sido usado desde 1968.[1] También facilitaba la computación distribuida, mediante conexión de múltiples aplicaciones para crear una mucho mayor, sobre una red..

El estado de los Middleware para aplicaciones de Java

El esquema Cliente servidor esta muerto, la mayoría de los grandes vendedores de Cliente/servidor han modernizado sus productos y ahora dirigen su estrategia de mercado hacia tecnologías de tres niveles, una arquitectura en la cual un cliente lejano accesa fuentes de datos remotos vía un servidor intermedio.

Antecedentes

Primero definamos Java Middleware. El termino abarca servidores de aplicación que intentan conectar múltiples aplicaciones a través de usar Java y las tecnologías de Internet, incluyendo aquellos que son primeramente servlet Server, así como aquellos que son basados en Object Request Broker y Object Oriented Database. El Objetivo del middleware es centralizar la infraestructura del software y su desarrollo.

El paquete para usar Java middleware forza y ofrece, entre otras, las siguientes ventajas:

- La habilidad de Internet de economizar oficinas de interconexión y organizaciones
- La necesidad de que las organizaciones cooperen y compartan datos y procesos de negocio.
- El deseo de consolidar servicios genéricos y el manejo de un servicio.
- El deseo de proveer el manejo central de aplicaciones incluyendo el encendido, apagado, recuperación, carga, balanceo y monitoreo.
- El deseo de usar servicios y protocolos abiertos.
- El deseo de re-diseñar lógica de negocio hacia una lógica estructurada, en la cual se usan API's y protocolos.
- La necesidad de soportar aplicaciones de arquitecturas mixtas y cooperativas.
- El deseo de mover las decisiones de infraestructura de redes y servicios fuera del espacio de la aplicación, así los manejadores de sistemas pueden hacer decisiones de la infraestructura sin ser estorbados por aplicaciones que dependen de los protocolos propietarios.
- El deseo de influenciar la experiencia de la programación orientada a objetos extendiendo esta a los dominios del servidor dentro de los nuevos productos para los servidores orientados a objetos y los productos "puente" de objetos-relacional.

El principal componente estándar de Java middleware es el EJB middleware (Enterprise Java Bean) que fue creado con los siguientes objetivos:

- Proveer un modelo de programación fácil de usar mientras mantiene acceso a API's de bajo nivel.
- Dirigir problemas de ciclo de vida incluyendo desarrollo, desplegado y corrida.

- Mantener la compatibilidad con otros API's de Java
- Proveer la interoperabilidad entre EJB y aplicaciones que no son Java.
- La compatibilidad de CORBA.

Objetos, componentes y contenedores de modelo.

Los componentes de la aplicación se deben adherir a algún modelo de despliegue para el runtime el cual especifica como el componente se comunica con su ambiente y como este es instalado, iniciado, detenido y llamado; y como este accesa servicios importantes para este ambiente.

Los mas populares servidores de componentes de runtime para concentrar servidores y contenedores de modelo incluyen RMI (Remote Method Invocation), EJB (Enterprise Java Bean), CORBA (Common Object Request Broker Architecture), DCOM (MicroSoft's Distributed Component Object Model), servlet, JSP (Java Server Page) and Java Store Procedure.

Recursos de Conexión

Una característica de valor agregado para los productos middleware es el tipo de fuente de datos como los cuales se pueden conectar. Implementado confiabilidad, alta grado de lanzamiento, distribución, capacidad de transacciones en los drivers de la fuente de datos, lo cual no es una tarea pequeña, muchos productos viene con adaptadores que permiten a las aplicaciones conectarse a fuentes de datos particulares tal como bases de datos especificas, CICS, productos e message oriented middleware y puentes DCOM así como servicios de protocolos de Internet. La conexión de drivers de propósito general como JDBC o ODBC frecuentemente no es una solución viable si varias fuentes de datos necesitan ser integradas en una simple transacción o si un alto grado de complejidad es requerida en la consulta.

Una parte importante de la especificación EJB (Enterprise Java Beans) se refiere a recursos de pooling (Pooling → mantener una colección de objetos, servidores y conexiones listos para acceso, así no es necesario que estos sean creados cada vez que uno de estos es requerido). El manejo de recursos y transacción de API's están muy relacionados pues los recursos son objetos transaccionales. La mayoría de los productos middleware proveen conexión pooling por que establecen una conexión a los datos en una operación que no es costosa. Los actuales productos que proveen conexión de pool los hace usualmente en una de las dos formas:

- Integrando la conexión de pooling en drivers de fuentes de datos.
- Integrando conexión de pooling en datos de acceso API's

La especificación de la conectividad de la base de datos Java (JDBC) y la especificación del Java Transaction API (JTA) juntas definen un API para el driver para interconectarse con el manejador de transacciones y el manejador de conexión pool.

Manejo de transacciones.

Una de las primeras razones para usar middleware basados en la red es para construir aplicaciones que accedan fuentes de datos diferentes (o quizá distantes geográficamente) partes de una organización.

La alternativa es instalar o rentar costosas líneas y usar una solución cliente/servidor, pero esto ha dado una aproximación mucho mejor enguanto a costo-beneficio usando la Internet y simples accesibles protocolos de red.

La habilidad para que el costo beneficio sea vinculado a múltiples fuentes de datos de todas y cada una de las partes de una organización ha incrementado la importancia del requerimiento de crear transacciones distribuidas usando fuentes de datos de diferentes tipos. El objeto mayor es que se pueda manejar el proceso completo del negocio como una sola transacción de software, aún si este accesa a múltiples bases de diferentes proveedores, posiblemente en varias localizaciones geográficas. Muchos productos hoy permiten accesar como parte de una transacción distribuida, pero solo si todas las fuentes de datos son del mismo tipo y usan el mismo driver. La razón de acuerdo a algunos vendedores de middleware, es que muchas de las librerías de C para transacciones distribuida provistas por la base de datos de los proveedores aparecen para se no-reentrantes o en otro caso inadecuadas en este tiempo. Por eso la mayoría de los productos confirman soporte en las transacciones distribuidas dan esta confiabilidad solo para múltiples fuentes de datos del mismo tipo de base de datos, así el proveedor de middleware puede utilizar la capacidad de distribución e

transacción nativa de la base de datos. Aquellas que de verdad hacen un soporte de transacciones distribuidas a trastes de múltiples bases de datos, tal como BEA WebLogic, típicamente la complementan usando una arquitectura de multiproceso y colas de mensajes de memoria compartida para trabajar en las ya mencionadas librerías no-reentrante de C.

Un aspecto importante de usar transacciones distribuidas en una aplicación es que el mecanismo de sustento debe ser confiable. Primero que todo tu asumes que los datos son confiables y si una fase se va abajo antes que se complemente el commit en las otras dos fases, el sistema caído debe ser capaz de recobrase por si mismo así como a la transacción perdida, trayéndola en sincronía con todos los otros sistemas. También asume que el sistema de transacciones por si mismo es confiable y si este falla tu no tienen que buscar “cola de datos atoradas” para ser restauradas y la base de datos ser arreglada a mano.

Soporte a CORBA

La mayoría de los servidores de aplicación soportan CORBA también proveen su propio ORB (Objet Request Broker) como en el caso de Bea M3, o empaquetando un ORB de tres partes tal como un Visibroker u Orbix. En mucho de los casos el ORB proveído esta derivado de otros ORB’s por ejemplo Oracle ORB es derivado de Visigenic’s

Existen muchos servicios estándares de CORBA, el mas importante de los cuales es el CORBA COS Naming Services. Si un producto provee el COS Naming, este puede publicar objetos persistentes que otros ORB no pueden encontrar de una manera estándar. Objetos publicados con COS Naming pueden ser accedados usando JNDI (Java Naming and Directory Interfase), vía la implementación del JNDI COS Naming.

Una noticia es que un nuevo e importante servicio define un servicio de mensajeo con la calidad de “niveles de servicio”. Los proveedores de esta implementación es probable que lo desarrollen para accesar esto usando el Java Message Services (JMS). El JMS te permite elegir la calidad de los niveles de servicio si lo soporta el servicio.

Es una característica importante de integración el que Java puede ligarse con los sistemas de CORBA. Para middleware con un completo soporte RMI (Remote Method Invocation) Sobre IIOP (*), la herramientas de desarrollo necesitan ser soportadas por los clientes RMI accedando los servidores de CORBA. Por último poca gente se da cuenta de que no hay estándares RMI para propagar transacciones en el contexto de un manejador de una sola transacción a otro, considerando que IIOP (*) provee tal estándar, por lo que al usar CORBA es necesario utilizar un middleware que incluya ORB.

(*) Internet Inter-ORB Protocol, protocolo de CORBA e conexión para transmitir objetos remotos con métodos de invocación.

Directorio de servicios y objetos

Para usar un servicio remoto de cualquier tipo, tú debes saber como encontrarlo. La mayoría de los middleware incorporan un directorio de servicios de algún tipo, permitiendo a algún componente de la aplicación o cliente encontrar otro objeto o servicio. El CORBA COS Naming y el intercambio de servicios son ejemplo del intercambio de servicio, así como el registro y activación de demonios de RMI. Un esquema bien conocido de nombramiento es el URL, y muchos ORB permiten a las aplicaciones registrar objetos usando un URL; de hecho CORBA define dos estándares IIOP y GIOPREF para nombrar objetos usando URL’s

Balance y carga de replicación.

Actualmente el balance de carga es el logro mas común en el punto de localización de objetos o servicios. Esto es verdad aun en los servidores de red, para los cuales los gestores de balance basados en el esquema DNS y los re-direccionadores para balance IP que cargan mapeando un requerimiento entrante para un servidor específico (determinado por el nombre del domino y la dirección IP) a un conjunto de servidores registrados en un concentrado de direcciones IP.

Los servidores de aplicación presentan un requerimiento mas complejo por que los servicios se manifiestan como múltiples objetos sirvientes, potencialmente replicados en varias maquinas, en vez de un simple servidor de red por maquina. Además, mientras las tradicionales conexione http son lógicamente sin estado, la conexión a objetos pueden no serlo. Un localizador de objetos por eso necesita distinguirse entre la conexión

con estado y sin estado cuando se hace una decisión de reconexión. Una vista del objeto es frecuentemente iniciado en un servidor de red basado en componentes tales como el modulo ISAPI o NSAPI, un servlet o un Server-side Script de Java.

Cluster y Liga

El clustering se refiere a la habilidad de definir una colección de servers que proveen un conjunto común de servicios replicados. Un cluster puede ser implementado a varios niveles, incluyendo el nivel de comunicación, el nivel middleware, o el nivel de aplicación. Un cluster puede ser dedicado a servir a un determinado grupo de clientes, permitiendo a los clientes acceder al cluster de mas requerimientos y así minimizar la carga fuera del cluster.

La liga se refiere a la habilidad de ligar clusters o dominios de servicio para formar un gran total, con cada cluster maneja un conjunto de servicios locales. Si un requerimiento es para un servicio mas allá del dominio local, este es automáticamente reenviado al dominio de servicio apropiado.

Detección de fallas y defectos.

Muchos productos proveen la capacidad de detectar fallas en componentes del sistema y dinámicamente substituir con un componente que lo reemplace al defectuoso. El grado en el cual esto es automático y transparente para el cliente varía. Por ejemplo un objeto servidor defectuoso puede ser manejado en el cliente o este puede ser manejado en el servidor. Si la aplicación en el cliente y la aplicación especifica servidora no recibe ninguna excepción debido a una falla y la falla es completamente manejada por los componentes del middleware, la capacidad de recuperación se considera transparente.

Tipos de persistencia soportada

El termino persistencia es muy ambiguo y es usado en muy diferentes maneras. En el contexto de CORBA un objeto persistente es aquel que puede ser reactivado en cualquier momento y sabe como establecer su propio estado interno. La manera en la cual el estado es mantenido varía entre servidores y entre protocolos.

Los servidores de WEB proveen mecanismos de estado que cuentan con un almacenamiento basado en el cliente llamado "cookies". La capacidad de java en los servidores Web que soportan los servlets pueden mantener el estado de la sesión en el servlet por si mismo, si esta diseñado así. Este problema también esta relacionado con la activación de objetos. Existen servidores que pueden soportar el manejador de estados bean-managed o container-managed, este último implementa la persistencia del estado automáticamente utilizando tecnología que es seleccionada por el servidor. En este caso un medio declarativo es usado a tiempo de despliegue para decirle cuales campos en el servicio deben persistentes.

Desplegado de lógica de aplicación y negocio.

Múltiples aplicaciones tienen componentes distribuidos que debe ser actualizado simultáneamente. Los componentes pueden ser instalados en los clientes; estos pueden ser Enterprise JavaBean, SQL Store Procedures y esquemas de base de datos; de igual manera pueden ser cualquier otra forma en la cual las reglas de negocio y datos puedan ser definidos. Indistintamente que forma tome los componentes de la lógica de negocio, un sistema distribuido necesita una estrategia para realizar modificaciones y actualizaciones, así los usuarios en línea no experimentan errores o "tiempo muertos" como resultado de una actualización.

Autenticación y encriptación

La propagación de la autenticación de un usuario desde un componente middleware a otro es un problema importante. Estos productos necesitan una referencia cruzada de relaciones confiable de modo que una base de datos por ejemplo guarda y confía en autenticaciones por otro. Futuras versiones de JDK se dirigirán probablemente a dominios de autenticación y privilegios mas de lleno, mitigando los problemas de autenticación.

Directorio de usuarios y manejo de llaves

Un directorio de usuarios es una base de datos que contiene nombres de usuario e información acerca de ellos. LDAP se esta convirtiendo rápidamente en el directorio universal estándar y es implementado por la mayoría de los directorios de servicio hoy. Este también puede ser usado para almacenar certificados de usuario y verificar firmas digitales y para mandar mensajes encriptadas para certificar a los propietarios. La

mayoría de los middleware de Java usan el JNDI API para almacenar información de directorios de usuarios y servicios en un servidor de directorios que la empresa elija.

En adición, productos que usen certificados digitales deben proveer un medio para:

- Almacenar claves de servidores
- Emisión certificados y llaves
- Recuperar claves de clientes y servidores
- Revocar y validar certificados.

Control de accesos

El acceso debería ser idealmente declarativo y usar un simple mecanismo de autenticación, natural de cada plataforma. Actualmente esto es frecuentemente necesario para encajar nombres de usuario y password en aplicaciones. Algunos productos te permiten declarar esta información en un archivo de configuración, pero esto no es ideal por que esto significa que un administrador tuvo acceso a todos los passwords. Preferentemente una identidad indicada solo por “principal” debe ser propagada desde un componente middleware a otro sin el conocimiento el código de aplicación.

Trabajando con configuración de seguridad en redes.

Los firewall PROXY, las cuales son instaladas en la mayoría de las redes, filtran el tráfico basadas en el protocolo que se pase, el puerto y el posible contenido de los datos mismo. Ahí hay una tendencia para construir nuevos protocolos del http, así el contenido, mientras sea definido por un tipo MIME, aparece para el firewall como meramente tráfico del http sin ninguna reconfiguración especial. Para otros protocolos generalmente es necesario configurar el PROXY para que se permita pasar el protocolo en un puerto específico. Aquí hay al menos dos razones por lo que esto no es práctico. Una es que el protocolo y el software del servidor asociado no sea suficientemente confiable, en términos de su fiabilidad y la falta de vulnerabilidad para ataques externos. Otra es que el protocolo puede requerir múltiples puertos, quizá un numero variable de ellos.

Una solución provista por algunos vendedores de servidores de aplicación para remediar los problemas de los firewall es un PROXY http, el cual envuelve el requerimiento en un extremo y lo desenvuelve en el otro. Esta envoltura puede en algunos casos ser ineficiente, y dado que http es un modelo puramente de un sentido, hace que algunas funciones difíciles de lograr, tales como RMI o CORBA.

Manejo de Herramientas

La convivencia y flexibilidad de la administración en esta área en la cual los productos middleware son muy diferentes. Algunos por una parte más robustos en el flujo principal permiten ver que esta pasando, pero para cambiar cualquier cosa tú tienes que editar los archivos de configuración y recomenzar. Por otro lado, muchos productos middleware mantienen una considerable sofisticación manejo de capacidades desde el principio. Las características de manejo interactivo de varios productos incluye:

- Presentación de estadísticas, límites y contabilidad de proceso o uso de recursos por cliente o por grupo de clientes.
- Estatus de los objetos del negocio y la habilidad de “hacer un ping” a objetos
- Estatus del cliente de conexión y transacciones.
- Habilidad de definir y manejar el despliegue de la configuración de servidores.
- Habilidad de modificar clusters y servidores adicionales de JVM's para manejar carga creciente.
- Habilidad para monitorear y manejar concentraciones de recursos y colas de mensajes.
- Habilidad de determinar las necesidades de recursos y uso de cada componente.
- Habilidad de monitorear y manejar nombres de espacios de recursos y servidores y vista de JNDI por el nombre jerárquico.

Herramientas de desarrollo

La mayoría de los Java middleware viene con un ambiente de desarrollo para crear aplicaciones. Esto no es estrictamente necesario, gracias a que los componentes estándar de Java como JavaBeans, los cuales hacen esto posible para compartir componentes a través de la evolución de ambientes. Dado que los usuarios pueden provenir de diferentes plataformas de clientes incluyendo browsers y JVM's de diferentes marcas y modelos de presentación para un cliente débil es muy diferente para un cliente Java, esto es necesario aislar la lógica del negocio y el estado de la sesión de la dependencia de la presentación y el estado de la presentación.

Los componentes de la presentación tienden a ser altamente ligados al propietario y son frecuentemente empaquetados a otros servicios tales como el estado de una sesión y búsqueda de objetos, y por eso pueden encerrarse en un servidor de aplicación.

Una tendencia reciente para servidores de aplicación es soportar herramientas de diseño de objetos como Rational Rose. Esto hace posible integrar un sistema objeto diseñado con el esquema OODBMS (Objet Oriented Database Management System) y objetos del negocio.

Plataformas

En la selección de Java middleware, debes considerar las plataformas en las que esto esta disponible. Pero con la precaución que Java no corre en cualquier plataforma, no todos los Java middleware son puro Java o algunas pueden serlo. El hecho que los desarrolladores pueden desarrollar objetos de negocio Java no significa que el Server en si mismo esta escrito en Java. Algunos servidores de aplicación son verdaderamente aplicaciones de Java puro, pero la mayoría no los son.

Confiabilidad

Un punto común de falla es el servicio de búsqueda de objetos y algunos productos todavía no soportan directorios de objetos distribuidos y replicados o bases de datos de estado de objetos, haciéndolos vulnerables a una falla. La durabilidad del registro de transacciones usadas por OTM's es un problema, así como el almacenamiento persistente usado por productos basados en intercambio de mensajes. Algunos productos capacitados para Java pero que no son productos Java que tienen severos problemas de confiabilidad en este campo están siendo re-escritos en Java con la confianza de mejorar la confiabilidad.

Conclusión definición del Middleware

La diversidad de los productos middleware y el traslape con sus características hacen difícil darles categorías. La más importante es el servidor de aplicaciones EJB, pero bases de datos de objetos middleware para mensajes, servidores de aplicaciones para redes y muchas otras categorías comprenden un conjunto de elecciones apropiadas para diferentes aplicaciones.

El EJB y estándares que los acompañan proveen un marco para construir bases de estándares abiertos transaccionales y para múltiples aplicaciones Java. Todavía muchas áreas necesitan atención incluyendo áreas que ahora usan soluciones específicas de vendedores, tales como balance de carga y autenticación. El control de acceso necesita ser declarativo y propagado a las bases de datos así como los componentes de software, y esto es necesario para permitir separar dominios administrativos. Mas aún los problemas de modelos de presentación estándar necesitan ser enviados, como lo hace el código de despliegue. Finalmente los drivers de Java puro reemplazables que soportan transacciones distribuidas necesitan ser implementadas.

Tipos de middleware

Se pueden clasificar los diferentes middleware en función de su escalabilidad y su tolerancia a fallos:

Middleware	Descripción
Remote Procedure Call (RPCs)	El cliente realiza una llamada a procedimientos que están corriendo en máquinas remotas. Pueden ser síncronos o asíncronos.
Publish/subscribe	Este tipo de monitores middleware activan y entregan información relevante para los subscriptores.
Message Oriented Middleware (MOM)	Los mensajes enviados al cliente se recogen y se almacenan hasta que son solicitados, mientras el cliente continúa con otros procesos.
Object Request Broker	Este tipo de middleware permite que los clientes envíen objetos y soliciten servicios en un sistema

Middleware		Descripción
(ORB)		orientado a objetos.
SQL-oriented Data Access		middleware entre las aplicaciones y los servidores de base de datos.
Monitores de Transacciones Distribuidas		Provee herramientas y un ambiente de desarrollo y despliegue de aplicaciones distribuidas.
Servidor de aplicaciones		Software instalado en una computadora para facilitar el servicio (ejecución) de otras aplicaciones
Enterprise Service Bus		Una capa abstracta en el tope de un sistema de mensajería de empresa. Otros autores determinan sólo estos dos: Middleware de servicios generales Middleware de servicios específicos
XmlBlaster		Es un Middleware Orientado a Mensajes (Message Oriented Middleware) hecho totalmente en Java, soporta la entrega instantánea de mensajes vía IIOP, TCP, SSL o http, aproximadamente recoge 1800 mensajes/segundo y se distingue por tener la característica de QoS (Quality Of Services) esto indica que una aplicación de negocios provee al servidor cualquier objeto, que esté administrado con servidor QoS. XmlBlaster es un servidor Publish/Subscribe punto a punto el cual intercambia mensajes entre publisher and subscribes. El mensaje es descrito como meta-información XML-Codificado.
Middleware RFID		
SOA		
CORBA		

Tabla AD1 Tipos de Middleware

Middleware comerciales:

-  JBOSS
-  Middleware RFID Data Suite
-  Bea Tuxedo
-  Tibco
-  IBM WebSphere MQ (antes MQSeries)
-  JMS
-  CICS
-  Borland Visibroker

Apéndice E Búsqueda de Middleware para la propuesta de solución.

De los existentes en el mercado tanto abiertos como propietarios seleccione tres, uno abierto y dos propietarios los cuales describo sus principales características.

Finalmente por contener lo necesario y ser un sistema abierto y “gratuito” seleccione el XmlBlaster como propuesta de solución final, describo a detalle su función en el Apéndice F

XmlBlaster

El xmlBlaster es "Middleware Orientado a Mensajes" ("Message Oriented Middleware", MOM) publicado bajo los términos de la GNU Lesser General Public License; el desarrollo está siendo llevado hacia adelante por Michele Laghi, Peter Roth, Konrad Krafft, Manuel Kron, James Birchfield y Marcel Ruff.

Dado que muchas personas probablemente no entenderán nada del término "Middleware Orientado a Mensajes", diré algo acerca de él. MOM se necesita cada vez que hay que intercambiar información en una red heterogénea con nodos utilizando protocolos diferentes.

El ejemplo clásico es una red de máquinas expendedoras de tickets distribuida sobre todo un país. Estas máquinas han sido producidas por diferentes vendedores y fueron programadas por sus propios equipos de programadores con diferentes lenguajes de programación. Si, por ejemplo, el precio de un ticket ha de cambiarse, es necesario un traductor para traducir el mensaje "nuevo precio" al formato correcto utilizado por las máquinas reales y distribuirlo entre ellas.

También es importante traducir los resultados (en este caso éxito o error) al lenguaje que entiende el programa de administración. Este trabajo está siendo realizado por el Middleware Orientado a Mensajes como el xmlBlaster que se intercambian mensajes codificados en XML.

XmlBlaster confía también en un modelo publicación / suscripción pero esta vez el servidor está escrito en Java puro y debería ser capaz de ejecutar en cualquier plataforma con JDK >= 1.2.

Los suscriptores cubren casi el espectro completo. Ejemplos pueden utilizar expresiones XPath para filtrar los mensajes que desean recibir, y añadir sus propios MIME de filtrado plugins. C / C ++, Java, Perl, Python, VisualBasic.net, C #, PHP cliente.

La comunicación con el servidor se realiza con CORBA (utilizando JacORB), RMI, XML-RPC, nativo de zócalo, o un plug-in HTTP persistentes y es posible filtrar mensajes con expresiones Xpath.

Si CORBA fuera problemático para ciertas aplicaciones, también se puede usar correo electrónico o conexiones http como plug-ins. Como sugiere el nombre los mensajes en sí están codificados en XML, se puede transmitir cualquier cosa (incluyendo datos binarios).

El marco de seguridad plugin permite la autenticación/autorización de muchas maneras. Actualmente hay LDAP y passwd basada en plug-ins disponibles. Licencia: GNU Lesser General Public License (LGPL)

Websphere IBM

WebSphere es una familia de productos de software propietario de IBM, aunque el término se refiere de manera popular a uno de sus productos específicos: WebSphere Application Server (WAS).

WebSphere ayudó a definir la categoría de software middleware y está diseñado para configurar, operar e integrar aplicaciones de e-business a través de varias plataformas de red usando las tecnologías del Web. Esto incluye componentes de run-time (como el WAS) y las herramientas para desarrollar aplicaciones que se ejecutarán sobre el WAS.

La familia de productos WebSphere además incluye herramientas para diseñar procesos de negocio (WebSphere Business Modeler), para integrarlos en las aplicaciones existentes (WebSphere Designer) y para ejecutar y monitorizar dichos procesos (WebSphere Process Server, WebSphere Monitor)

Lista de compatibilidades con tecnologías Java

Versión de WebSphere		WebSphere 3.5	WebSphere 4.0	WebSphere 5.0	WebSphere 5.1	WebSphere 6.0	WebSphere 6.1	WebSphere 7.0
	Fecha de lanzamiento	1998 ?	Junio de 2001 ?	Noviembre de 2002?	Noviembre de 2003 ?	Finales de 2004	Mayo de 2006	Septiembre de 2008
	Fin de soporte	30 nov. 2003	30 abr. 2005	30 sep. 2006	30 sep. 2008			
JDK		1.2	1.3	1.3	1.4	1.4	5.0	6.0
Java EE		?	1.2	1.3	1.3	1.4	1.4	5.0
Servlet		2.1 y 2.2	2.2	2.3	2.3	2.4	2.4	2.5
JSP		0.91 1.0 1.1	1.1	1.2	1.2	2.0	2.0	2.1
EJB		1.0	1.1	2.0	2.0	2.1	2.1	3.0

Tabla AE1 Esta tabla está basada en IBM Information Center: Specifications and API documentation

Versión 5

Esta versión se publicó el 19 de noviembre de 2002. Fue un servidor de aplicaciones certificado J2EE 1.3. Tuvo una reescritura principal del codebase V3/V4 y fue la primera vez que WebSphere Application Server fue programado desde un codebase común. Ahora los WAS de todas las plataformas de despliegue, desde Intel x86 a mainframe, son esencialmente el mismo código. El repositorio de configuración basado en bases de datos fue reemplazado con un repositorio de configuración basado en ficheros XML de replicación. Un servicio llamado el Gestor de Despliegue tenía la copia maestra de la configuración de celdas, y los nodos tenían los ficheros que necesitaran copiados desde este servidor maestro cuando quiera que cambiaran. V5 también incluyó una versión miniatura de MQ 5.3 llamado el servidor de Java Message Service (JMS) empotrado.

- ✚ Express Edition reemplaza a la edición Standard. Express ahora se convierte en el término que indica las ofertas orientadas a PYMES desde IBM, a través de todas sus marcas de software.
- ✚ Base
- ✚ Network Deployment (Despliegue de Red). Esta versión soporta el despliegue de una configuración en celdas con cluster y soporte a tolerancia de fallos J2EE. Ahora también incluye Edge Components, previamente conocido como Edge Server. Esto suministra un servidor proxy, balanceado de carga, y enrutamiento basado en contenidos.
- ✚ Enterprise Edition. Esta versión añadió un motor de workflow, llamado el Coreógrafo de Procesos, la primera vez, pero elimina el estándar BPEL. También añadió el primer modelo de threading de aplicaciones completamente soportado llamado WebSphere Asynchronous Beans.
- ✚ WebSphere Application Server para z/OS. Esta versión esencialmente es el mismo que el producto de Network Deployment excepto que está optimizado para tomar ventaja total de las características z/OS, tales como el Gestor de Carga de Trabajo(Workload), para impulsar las tecnologías clave que hacen al mainframe indispensable para cargas de trabajo seguras, importantes y escalables.

Versión 5.1

Publicado el 1 de abril de 2004.

El principal cambio desde V5.0 fue una actualización del JDK a 1.4.2 y el uso del lenguaje Jython para el soporte de scripting wsadmin en lugar de solo soportar Java Tcl.

- ✚ Express
- ✚ Base
- ✚ Network Deployment
- ✚ WebSphere Application Server para z/OS

Versión 5.1 para z/OS es la primera en soportar motores zAAP.

- ✚ WebSphere Business Integration Server Foundation V5.1

Esta es la continuación del producto para WebSphere Application Server Enterprise Edition V5.0. El motor de workflow fue actualizado para soportar BPEL en lugar del formato FDML propietario usado en V5.0. El producto estaba en todas las plataformas IBM desde los entornos Intel al mainframe.

WebSphere eXtended Deployment (XD)

Beneficios del Administrador

Muchos negocios ejecutan varias granjas de servidores pero desean consolidarlas en una única granja de servidores más pequeña. Esto es porque la mayoría de las granjas de servidores son infrautilizadas o sobre-explotadas. Las cajas están normalmente ejecutándose al 10% de carga lo cual es muy costoso y nada flexible. Por ejemplo, una granja de servidores se calienta y funciona al máximo mientras la granja en la siguiente habitación está todavía ociosa al 10%. XD permite a los administradores definir un cluster único (un grupo de nodos) y monitoriza el balanceado de carga y decide dinámicamente en que cajas del grupo de nodos debe hospedar cuales aplicaciones para obtener las metas. Si la aplicación A tiene un tiempo de respuesta actual de 1.5 segundos, XD sacará los recursos de las aplicaciones B y C para incrementar la potencia dedicada a A y decrementar su tiempo de respuesta. XD puede también predecir que A probablemente excederá su tiempo de respuesta en 10 minutos basándose en una tendencia y reaccionará anticipándose al evento. Esto simplifica ampliamente la vida de un administrador y permite a las máquinas ser más eficientes que una granja de varias granjas independientes convencionales. XD también ofrece opciones para generar alertas de email cuando se sobrepasan las condiciones, esto puede reiniciar los servidores cuando parece tener un fallo de memoria o después de X peticiones.

Beneficios del Desarrollador

Facilidad de Partición WebSphere

Las aplicaciones tradicionales J2EE trabajan bien para muchos tipos de aplicaciones. La clase puede ser ampliamente categorizada como aplicaciones que se ejecutan en un cluster simétrico sin estado enfrente de una base de datos:

-  Todos los miembros del cluster pueden realizar cualquiera tarea en cualquier momento.
-  La aplicación es sin estado.
-  La aplicación es modal lo que significa que solo realiza trabajo sincrónicamente en respuesta a la petición del cliente el cual puede ser recibido usando HTTP/IOP o JMS.

Hay otras aplicaciones que no funcionan tan bien en tal entorno, por ejemplo, un sistema de comercio electrónico en un banco. Tales aplicaciones normalmente usan trucos que pueden mejorar el rendimiento como el particionamiento, multi-threading y la escritura a través del cacheado. Estas son aplicaciones que pueden explotar clustering asimétrico. Un cluster asimétrico es prácticamente lo opuesto de un cluster simétrico:

-  Las aplicaciones pueden declarar particiones nombradas en cualquier punto mientras se están ejecutando, las particiones tienen alta disponibilidad, son móviles dentro del cluster y normalmente sólo se ejecutan sobre un miembro de cluster a la vez.
-  El trabajo entrante para una partición es enrutado al miembro de cluster que hospeda la partición.
-  La aplicación es amodal. Las Particiones tienen un ciclo de vida por ellas mismas y pueden iniciar alarmas/hilos en background así como también responder a eventos entrantes si son mensajes IOP/HTTP o JMS/foreign.

WebSphere XD ofrece un nuevo conjunto de APIs llamada WebSphere Partition Facility (WPF). Estas APIs permiten a las aplicaciones que requieren desplegar un cluster asimétrico sobre un servidor J2EE.

ObjectGrid

El ObjectGrid es la primera pieza del framework de cacheado distribuido de IBM. V6 de XD se entrega con soporte para cacheado de Objetos locales Java así como también cacheado peer to peer con invalidación/push de datos asíncronos. Esto está diseñado para instalar framework extensible de cliente para cachear objetos. Todos los aspectos del framework núcleo tienen puntos de plugin (para añadir plugin). Esto permite a los clientes añadir funciones así como también permite a IBM extenderlo en futuros niveles del producto. Esto está diseñado para ejecutarse independientemente del WebSphere. No significa que no se integra con WebSphere 6.0 XD, sí que lo hace. En cambio, esto también funcionará con las versiones viejas del servidor de aplicaciones, servidores tomcat, y servidores de aplicaciones competitivos. Esto es necesario para competir contra otros productos de cache distribuida del mercado. Actualmente solo requiere un nivel de JDK 1.4 de cualquier vendedor. La Version 2 de ObjectGrid está actualmente bajo desarrollo para entrega en XD 6.0.1. Esto añadirá la función significativa al componente así como también un soporte a nivel JDK más

amplio. La intención es permitir a los clientes construir grandes grids de JVMs en las cuales puedan conectar aplicaciones y almacenar objetos de varias calidades de servicio. La grid está siendo diseñada para escalar a 1000s de JVMs y mantener una gran cantidad de datos.

OpenFusion general CORBA

- ✚ Alto rendimiento, multi-plataforma, las soluciones multi-idioma
- ✚ Implementaciones escalables desde el chip a la empresa
- ✚ QoS Arquitectura Orientada a Servicios (SOA) de tecnología
- ✚ Simplificación de desarrollo CORBA
- ✚ Apoyo de clase mundial, Servicios Profesionales y Formación

PrismTech líder en middleware de alto rendimiento

PrismTech es un líder en el mercado de alto rendimiento distribuido middleware para sistemas de misión crítica, en los sectores de la industria, incluyendo aeroespacial y de defensa, las telecomunicaciones comerciales (gestión de red) y servicios financieros. La clave del éxito PrismTech y la posición de liderazgo son las siguientes:

- ✚ Amplitud y calidad de nuestro middleware soluciones
- ✚ clase mundial de apoyo técnico y servicios profesionales
- ✚ Flexibilidad en los modelos de negocio que se pueden adaptar a las necesidades específicas de cada cliente

Basado en estándares, de alto rendimiento Middleware

El Object Management Group es (OMG) Solicitud de objetos común Broker Arquitectura (CORBA ®) estándar de middleware permite que las aplicaciones de software para invocar operaciones en objetos distribuidos sin preocuparse por la localización de objetos, lenguaje de programación, plataforma de sistema operativo, protocolos de comunicaciones, interconexiones o hardware. CORBA sigue siendo el estándar abierto de más éxito en el apoyo a los sistemas heterogéneos distribuidos misión crítica que requieren unos niveles excepcionales de rendimiento y calidad de servicio.

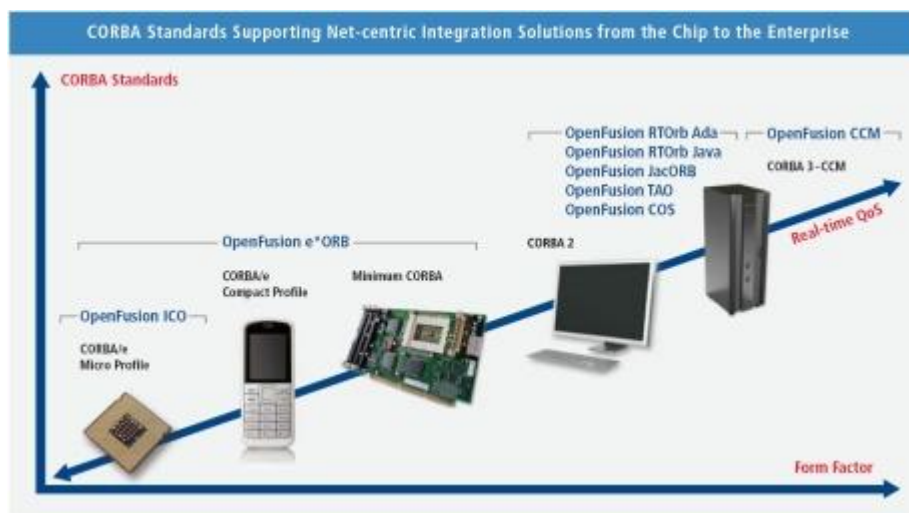
soluciones PrismTech CORBA de abarcar el espectro completo de las necesidades de los sistemas empresariales de gama alta a las aplicaciones más pequeños incorporados en tiempo real. Respaldan todos los principales entornos operativos y proporcionar la más amplia compatibilidad con idiomas de programación disponibles, incluyendo suites de herramientas para C, C + +, Java TM, Ada y VHDL aun.

Probado en el campo

OpenFusion ® es una suite principal PrismTech de productos CORBA y se utiliza en muchos proyectos por los clientes el desarrollo de sistemas de software de misión crítica. Estos sistemas suelen tener requisitos arquitectónicos diversos y complejos, incluyendo la funcionalidad QoS avanzada y necesidades, el lenguaje de programación heterogénea y soporte de plataformas, y factor de forma limitada (huella).

La suite de productos OpenFusion se compone de:

- ✚ Open-Source ORBs Empresa – TAO (Lenguaje C + +) y JacORB (Java).
- ✚ Avanzada Servicios CORBA (COS) – Servicio de nombres, Comercio de servicios, Servicio de Notificación, Registro de mantenimiento, Y Servicio de hora.
- ✚ OpenSource Modelo de componentes CORBA (MCP) - la aplicación de ligero de la OMG CCM (C++ y Java TM) y de implementación y configuración (D & C) las normas, con el apoyo comercial fuera de la plataforma-(COTS) de ingeniería basada en modelos (MDE) herramientas de desarrollo.
- ✚ Especializada en tiempo real ORBs - RTOrb Java y Ada Ediciones.
- ✚ Especializada incrustado middleware para fines generales del procesador (GPP), procesador de señal digital (DSP) y Field Programmable Gate Array (FPGA) ambientes - e * ORB C + + y C Ediciones y ICO, el ORB del circuito integrado.
- ✚ Complementaria avanzada solución de seguridad - Xtradyn I-DBC proporcionando recorrido Secure Firewall CORBA y autenticación completa, autorización, auditoría y mecanismos de Administración.
- ✚ Perfecta convivencia con OpenSplice DDS.



Calidad de Servicio (QoS)-Soluciones SOA habilitada para la Defensa y Telecomunicaciones

Para C2 red centralizada y los sistemas C4ISR, PrismTech ofrece el desarrollo más completo y el entorno operativo para la defensa a gran escala, la industria aeroespacial, y los proyectos federales de integración.

PrismTech de QoS plataformas SOA, Model Driven Engineering Tools, y distribuida en tiempo real y embebidos (DRE) soluciones de sistemas son utilizados por los mundos 'más grande de defensa y de los contratistas aeroespaciales, incluyendo BAE Systems, Boeing, EADS, General Dynamics, Harris, ITT Industrias, Martín Lockheed, Northrop Grumman, Raytheon, Rockwell Collins, Selex, y Thales.

En los proyectos de defensa a gran escala, como Director General Adjunto de la Marina de los EE.UU. 1000 de los programas, la tecnología PrismTech de CORBA es la opción número uno. Por ejemplo, en tiempo real de Java PrismTech ORB (RTOrb Java) fue seleccionado por Raytheon para apoyar su puesta en práctica del Open de la Marina para el Medio Ambiente Arquitectura Informática (OACE). "Estamos entusiasmados de ofrecer en tiempo real de la tecnología de middleware PrismTech al programa DDG 1000", declaró Ed Geisler, Vicepresidente y Director General Adjunto 1000 director del programa para Raytheon Integrated Defense Systems. "OA no se trata sólo de las soluciones técnicas. Se trata también de un modelo de negocio abierto que ofrece la competencia. La selección de PrismTech vino a través de un proceso que utiliza la evaluación formal para asegurar que la Armada consigue la mejor relación. Existen varios productos en esta tecnología espacio y se determinó que la oferta desde PrismTech, una compañía privada, ofrece el mejor valor a la Marina".

En el sector de las telecomunicaciones comerciales PrismTech ofrece una amplia gama de soluciones distribuidas CORBA para los proveedores de telecomunicaciones inalámbricas y de línea fija de infraestructura, los fabricantes de equipos de redes y de terceros fabricantes OSS.

PrismTech gama de productos CORBA se utilizan como un factor clave de Telecom aplicaciones OSS basado en los estándares más recientes del Foro Telegestión (TMF) y de tercera generación Partner Program (3GPP). Para los sistemas basados en NGOSS TMF (Nueva Generación de Sistemas de Operaciones y Software) principios básicos de arquitectura y estándares específicos de la interfaz como el Multi Tecnología OS Interface (MTOSI) sólo rango OpenFusion PrismTech de CORBA QoS basado habilitado plataformas SOA puede soportar la creciente complejidad y la diversidad en equipos, redes y OSS, además de poder hacer frente a un creciente volumen de información que se debe distribuir de manera eficiente a fin de gestionar un extremo a extremo Telecom OSS - la información adecuada en el momento adecuado. PrismTech QoS plataformas SOA son capaces de soportar QoS estrictos requisitos de seguridad de los OSS, que cumplan con los principales estándares de Telecom.

PrismTech listas de líderes como Alcatel-Lucent, Cienna, Ericsson, Nokia Siemens Networks, Samsung, y Telfordia para sus clientes en los sectores de telecomunicaciones y de transmisión de datos.

Compromiso con la innovación

Ningún otro fabricante de middleware otros invierten tanto en la tecnología CORBA nuevas e innovadoras en comparación con PrismTech. Para heterogéneos sistemas de misión crítica con requisitos complejos PrismTech continúa produciendo productos que levantan la barra con respecto a los tipos de entorno que puede ser apoyado por nuestros estándares basados en soluciones de alto rendimiento de integración. Por ejemplo, las innovaciones recientes se cuentan el desarrollo de ORB primero en el mundo del hardware (OpenFusion ICO) Para apoyar la FPGA y ASIC dispositivos, y un En tiempo real de Java ORB (OpenFusion RTOrb Java) que por primera vez ofrece las ventajas de la plataforma Java a la comunidad en tiempo real sistemas para misiones.

PrismTech compromiso de apoyar a la comunidad CORBA significa que estamos constantemente mejorando nuestros productos y tratando de encontrar nuevas maneras de hacer el desarrollo de complejas aplicaciones distribuidas más fácil. Nuestra estrategia de combinar Model Driven Engineering (MDE) con nuestras técnicas de alto rendimiento middleware CORBA ha conducido directamente al desarrollo de la primera comercialmente viable la aplicación de componentes CORBA Modelo. Para aplicaciones complejas OpenFusion CCM ofrece grandes ventajas para la productividad y permite a los usuarios para apoyar de manera eficiente el pleno desarrollo y ciclo de vida del despliegue de un sistema CORBA.

Servicios profesionales de clase mundial

PrismTech ofrece servicios de expertos para ayudar a los arquitectos y desarrolladores de software de los fabricantes de equipos e integradores de sistemas desarrollar e implementar complejas, distribuidas infraestructuras de software. Nuestra tecnología y experiencia en el campo nos permite ofrecer una amplia gama de servicios que se pueden combinar y / o personalizada para satisfacer los requerimientos específicos de un proyecto o línea de productos de inversión, incluyendo:

-  Capacitación y Programas de Educación
-  Talleres
-  personalizaciones
-  Sistema de la Integración

Nuestros expertos han ayudado a muchos clientes aprovechar los beneficios de nuestra tecnología y modelos de negocio flexibles, haciendo que el proceso de migración a nuestras soluciones tan sencilla y rentable como sea posible. PrismTech emplea a uno de los mayores equipos de expertos CORBA en el mundo y para los clientes con sistemas de legado que necesitan ser apoyados por muchos años por venir PrismTech puede aportar los recursos y conocimientos de ingeniería para permitir que los clientes de externalizar completamente el mantenimiento permanente de su base CORBA aplicaciones o productos. A medida que su sistema de edades que ya no tienen que mantener conocimientos CORBA "in-house", que puede ser una tarea costosa. Por otra parte, los valiosos recursos de desarrollo de software puede ser liberado para volver a ser desplegado en tus nuevos proyectos.

Soluciones personalizadas y flexibles de licencias

PrismTech ofrece soluciones a medida para satisfacer las necesidades de grandes proyectos, incluidas las licencias flexibles para cubrir todos los modelos presupuestarios y acceso prioritario a nuestros equipos técnicos de renombre mundial. Al abrazar el modelo de código abierto combinado con los mejores productos comerciales de la industria, PrismTech ofrece las soluciones más innovadoras y rentables y se ha comprometido a su desarrollo en curso

Apéndice F XmlBlaster Propuesta seleccionada

Descripción general.

¿Qué es Xml Blaster?

XmlBlaster es un Middleware Orientado a Mensajes (Message Oriented Middleware) Hecho totalmente en Java y soporta la entrega instantánea de mensajes vía IIOP, TCP, SSL o http, aproximadamente recoge 1800 mensajes/segundo y se distingue por tener la característica de QoS (Quality of Service) esto indica que una aplicación de negocios provee al servidor cualquier objeto, que esté administrado con servidor QoS.

XmlBlaster es un servidor Publish/subscribe punto a punto el cual intercambia mensajes entre Publisher and subscribes. El mensaje es descrito como meta-información XML-Codificado. Los mensajes pueden contener todo, GIF, imágenes, objetos Java, Scripts de Python, datos XML, un documento Word, texto plano. En fin de todo.

El hecho de que sea un servidor Publish/Subscribe indica que el publicador son aquellas aplicaciones que envían mensajes sin que tengan explícitamente o específicamente destinatarios o tenga conocimiento de destinatarios proyectados. De igual manera, receptores de la aplicación (suscriptores) deben recibir sólo esos mensajes que el suscriptor ha registrado un interés en ellos.

Se llevan a cabo este reclamo entre remitente y destinatario usualmente por una entidad intermedia entre el publicador y el suscriptor, que sirve como nivel de direccion. Esta entidad es una cola que representa un objeto o canal.

Un suscriptor suscribe a una cola para expresar interés en los mensajes encolados en aquella cola y por usar un sujeto (o contenido) basado en reglas como un filtro.

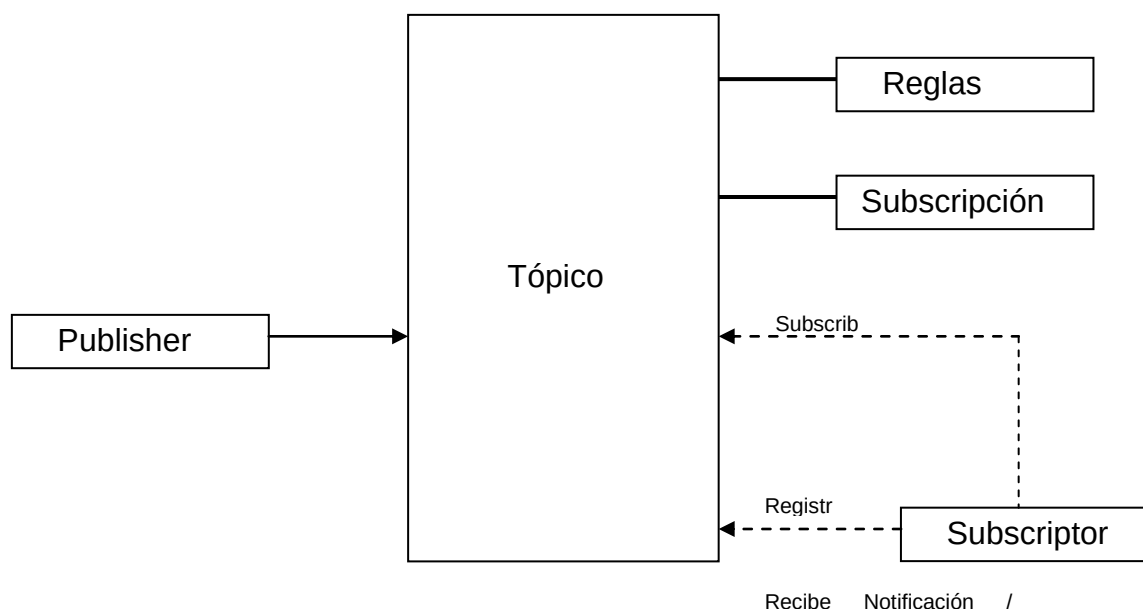


Figura AF1 Diagrama publicador y suscriptor.

La comunicación con el servidor es basado en CORBA (Usando JacORB) o RMI o XML-RPC, los clientes son libres de escoger su protocolo preferido. Otros protocolos como email, socket o SOAP pueden ser enchufados. Dependiendo de la interfase con la que se use XmlBlaster se puede explotar de diferentes usos de acuerdo con las siguientes tabla:

E-mail	CORBA	RMI	http	XML-RCP	SOCKET	SOAP	NATIVE
EN EL CLIENTE							
El recibo de e-mail es soportado, pero para partes múltiple esta aún pendiente	Chat Subclientes	Chat subclientes	Browser with JavaScript/XML y conexión persistente http Browser con applet (http tunneling scheduled)	Chat Subclientes	Presionando a que el cliente regrese la llamada en la misma conexión del Socket permite túneles en las firewalls		Lógica de la aplicación

Tabla AF1 Protocolos utilizados por el Middleware XMLBlaster

Subscriptores pueden usar expresiones Xpath para filtrar los mensajes que ellos deseen recibir.

XmlBlaster usa el parser Sun XML para hacer el análisis sintáctico y XT de James Clark para análisis sintáctico XSL.

Tiene clientes disponibles en la distribución de XmlBlaster para Perl, Python, C/C++ y Java (aplicaciones, applets, servlets)

XLBlaster provee una estructura para llamadas de regreso al browser, permitiendo a los browsers (Netscape, Mozilla, Ms IE) para recibir llamadas de contestación instantánea sobre una conexión persistente de http.

Características

- ✚ Gratis para uso privado, comercial, educación (Licencia LGPL)
- ✚ Middleware orientado a Mensajes (MOM) con soporte publish/subscribe para comunicación punto a punto.
- ✚ Soporte multiplataforma (el servidor es Java puro)
- ✚ Soporte multiprotocolo (CORBA, RMI, XML-RCP, raw socket, email, acceso nativo)
- ✚ Lenguaje neutral (C++/C, Java, Python, Perl)
- ✚ Extensible (XML basado en QoS)
- ✚ Típicos consultables (XPath)
- ✚ MiMe basado e búsqueda completa de texto (actualmente enchufado para expresiones regulares y XPath)
- ✚ Sistema de seguridad independiente (Actualmente enchufado para LDAP y Cryp-password), soporte de autenticación, interceptores de mensajes y autorizaciones.
- ✚ Arquitectura de plugin persistente (actualmente plugin par Xindice – una base de datos XML nativa y manejador de archivos).

Arquitectura del XMLBlaster

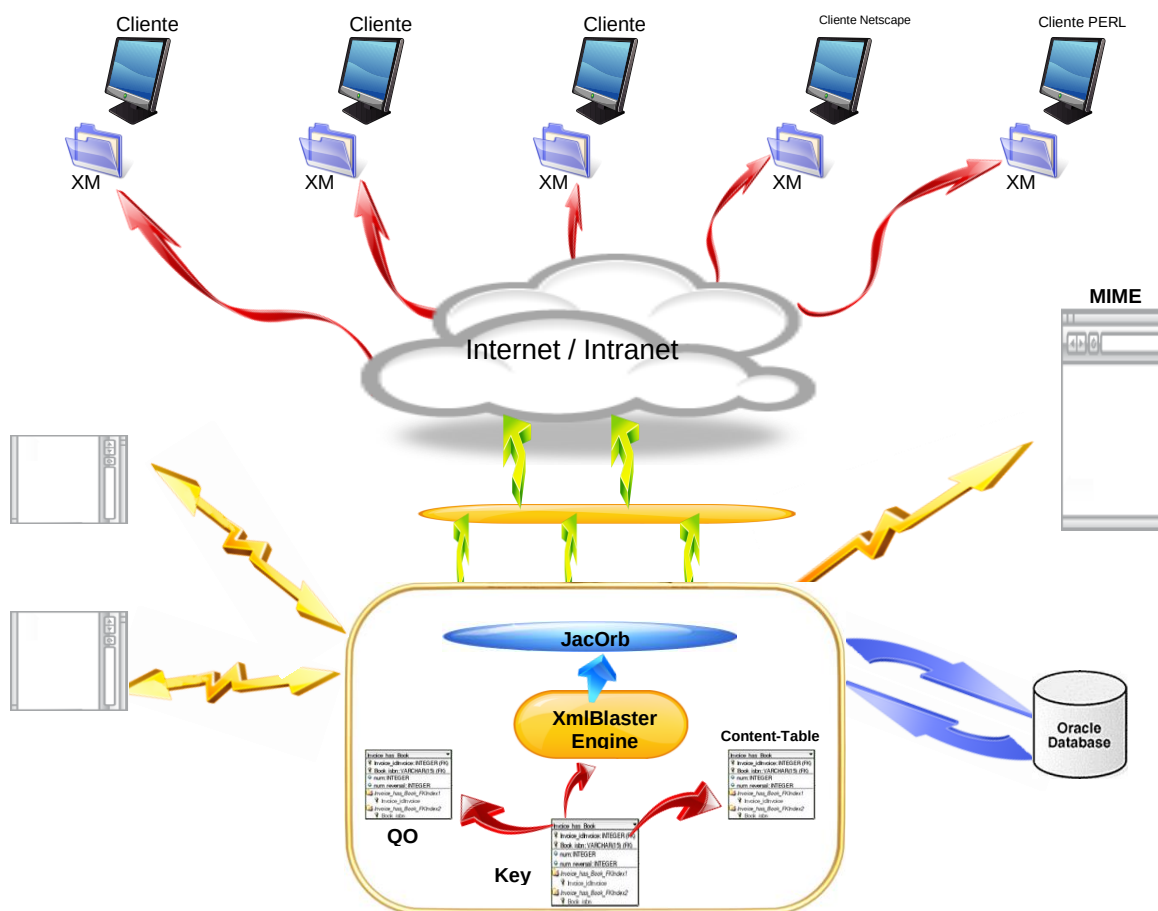


Figura AF2 Diagrama Arquitectura XMLBlaster

Como vemos en el diagrama se puede publicar la información de mas de una aplicación a través de la Internet/intranet y presentarla junta en cualquiera de las terminales de la red sabedores de que no deben ser un problema manejar los requerimientos en distintos idiomas con XMLBlaster.

En la figura se muestra el intercambio de información usando la interfase de CORBA (JacORB), el archivo de configuración para la librería de CORBA JacORB es llamada jacorb.properties. la conexión hacia la Intranet/Internet se hacen por medio de http o bien IOP (Internet Inter-ORB Protocol, que es elservicio de CORBA para transmitir métodos invocación para objetos remotos).

Grafica de estados y configuración de los mensajes y ciclo de vida de los tópicos.

En XMLBlaster se define un tópico como una simple clave de identificador de objeto considerando que una instancia de mensaje son los mensajes publicados (probablemente con un idéntico identificador de objeto). Las instancias del mensaje construyen una cola de histórico del tópico.

Todas las claves de identificación del objeto definen un tópico, tú publicas cualquier número de mensajes con el mismo identificador de objeto, estas unidades de mensajes instanciadas son distinguidas por el momento de emisión que es único (timestamp).

El mensaje es consultable desde la cola del histórico. La cola del histórico solo contiene referencias a la unidad de mensaje real.

Todas las subscripciones ponen una referencia de una unidad de mensaje en la cola de respuesta de su cliente, desde donde el mensaje es enviado al cliente.

La unidad real de mensaje es manejada por la implementación del “Cache”. Los apuntadores sobre las entradas del Cache son referencia con contadores. Si no existe una referencia debilitada la entrada del cache es destruida.

Si la entrada del cache es cambiada al disco duro, la referencia debilitada es puesta a nulo. En este caso acceso forza a revisar y el mensaje es cambiado de nuevo al cache.

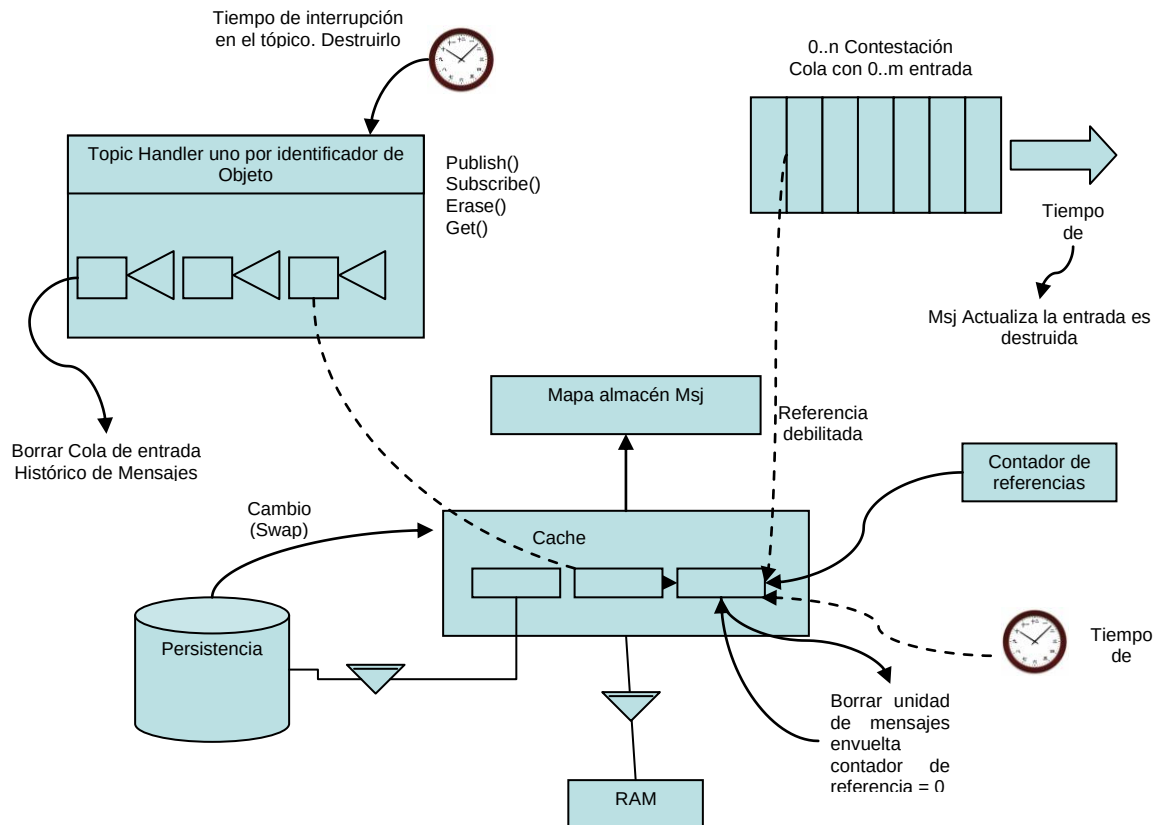


Figura AF2 Ciclo de vida de un Tópico

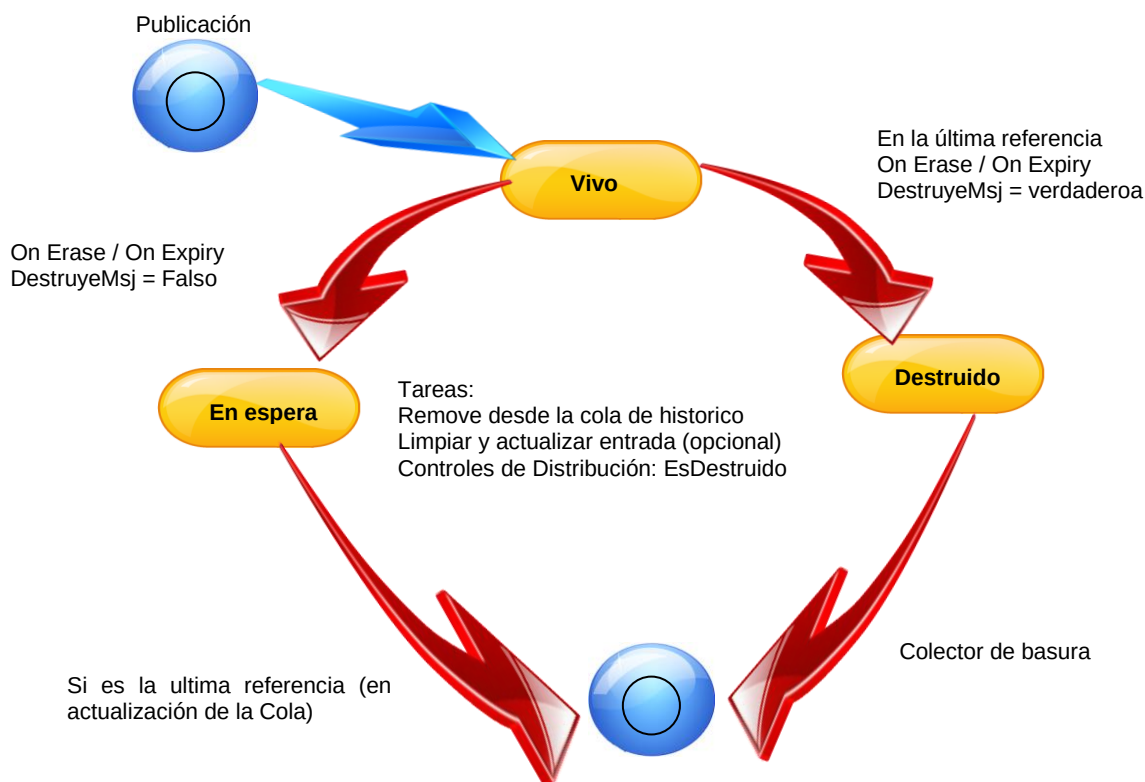


Figura AF3 Ciclo de vida y estados de transacción del mensaje

Propuesta Rechazada.

El sistema XmlBlaster fue el que seleccionó por que contenía todos los elementos necesarios para la interfaz, el principal problema era que se tenía que instalar gestores a cada cliente de los cuales se deseaba la suscripción a la interfaz.

Se deseaba que fuera transparente a cada usuario y estos no tuvieran que casarse con una herramienta.

El cliente debería generar sus propios clientes para suscribirse a la Interfaz, en este momento en análisis de diseño.

Como se trata de una aplicación con código libre, no existía un soporte como tal, solo de los desarrolladores que no era fácil de localizar.

Apéndice G Metodología de desarrollo y documentación

Elementos rotacionales de UML 1.0

A Continuación se revisan en niveles de complejidad los diversos elementos notacionales que presenta el Unified Modeling Language. Estos elementos pretenden ser un lenguaje común para el modelamiento de cualquier sistema.

Esta descripción no pretende ser exhaustiva en términos sintácticos, semánticos y de presentación de los elementos de la notación. Debe entenderse como una guía inicial al tema.

Se agrupan los conceptos básicos por tipo de diagrama:

1. Diagrama de Estructura Estática
2. Diagrama de Casos de Uso
3. Diagrama de Secuencia
4. Diagrama de Colaboración
5. Diagrama de Estados
6. Diagrama de Actividades
7. Diagrama de Implementación

Los conceptos avanzados por tipo de diagrama son:

- I. Diagrama de Estructura Estática
- II. Diagrama de Secuencia
- III. Diagrama de Colaboración
- IV. Diagrama de Estados

1. Elementos básicos en un diagrama de estructura estática

Un diagrama de estructura estática muestra el conjunto de clases y objetos importantes que hacen parte de un sistema, junto con las relaciones existentes entre estas clases y objetos. Muestra de una manera estática la estructura de información del sistema y la visibilidad que tiene cada una de las clases, dada por sus relaciones con las demás en el modelo. Supongamos el modelamiento de una máquina de café. Un diagrama de estructura estática inicial podría ser:

Máquina de Café

Supongamos que se requiere desarrollar el control de una máquina de entrega de café automática. La máquina debe permitir a una persona entregar una cantidad de dinero en monedas de 100, 200 o 500, escoger uno de los productos de acuerdo a su precio (café negro, café claro, caldo), escoger (si es pertinente) un nivel de azúcar y entregar el producto y las vueltas. El dinero que los usuarios introducen se guarda en un recipiente aparte al disponible para vueltas, el cual se encuentra ordenado por denominación.

Existen estados de error de la máquina, cuando detecta un mal funcionamiento, no existencia de vueltas o no existencia de ingredientes. El usuario puede en cualquier momento antes de escoger el azúcar cancelar la operación, mediante un botón existente para este objetivo.

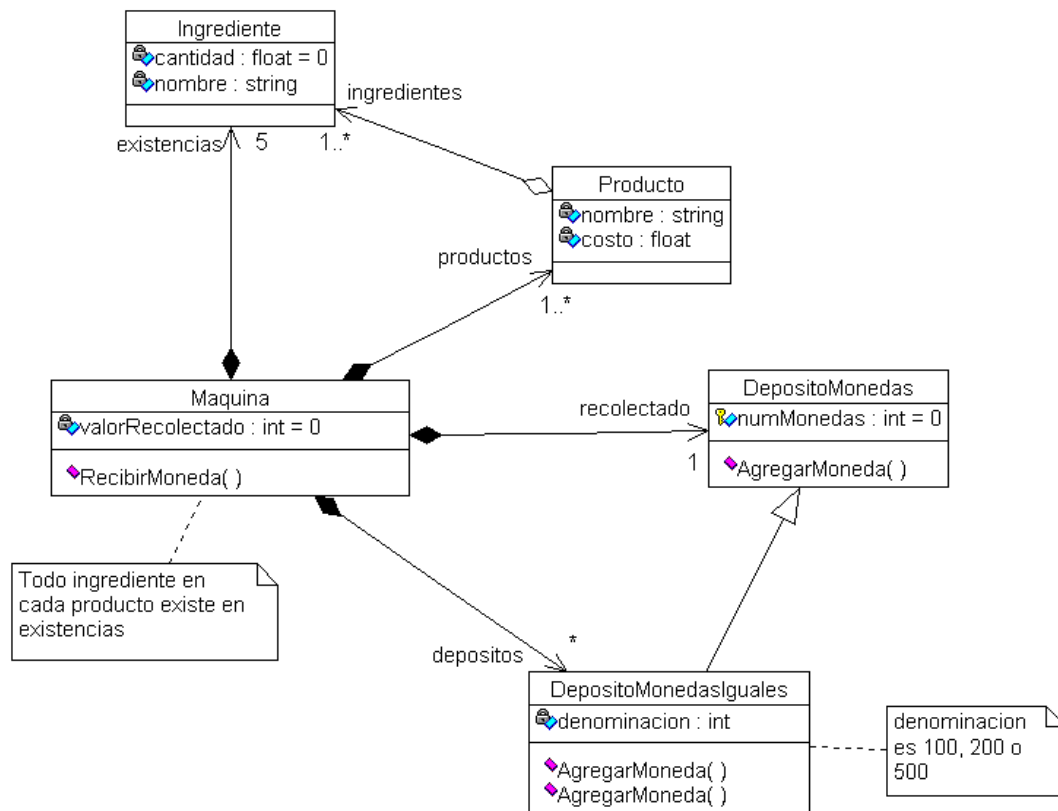


Figura AG 1 Diagrama de Estructura estática (ejemplo maquina de café)

Clase

Representada por un rectángulo con tres divisiones internas, son los elementos fundamentales del diagrama. Una clase describe un conjunto de objetos con características y comportamiento idéntico. En el ejemplo se encuentran las clases Ingrediente, Producto, Maquina, Deposito-Monedas y Deposito-Monedas-Iguales.

Los tres compartimentos estándares alojan el nombre de la clase, sus atributos y sus mensajes, respectivamente.

Atributo

Identifican las características propias de cada clase. Generalmente son de tipos simples, ya que los atributos de tipos compuestos se representan mediante asociaciones de composición con otras clases. La sintaxis de un atributo es

visibility name : type-expression = initial-value { property-string }

Donde visibility es uno de los siguientes:

+ public visibility
protected visibility
- private visibility

type-expression es el tipo del atributo con nombre name. Puede especificarse como se ve un valor inicial y un conjunto de propiedades del atributo.

En el caso del ejemplo, la clase Ingrediente tiene dos atributos: uno denominado cantidad, de tipo float y con valor inicial 0; y el atributo nombre de tipo string sin valor inicial. En este caso, la herramienta utilizada ha cambiado la representación de la visibilidad, utilizando el símbolo  para indicar visibilidad privada, el símbolo  para visibilidad protegida y el símbolo  para indicar visibilidad pública.

Operación

El conjunto de operaciones describen el comportamiento de los objetos de una clase. La sintaxis de una operación en UML es

visibility name (parameter-list) : return-type-expression { property-string }

Cada uno de los parámetros en parameter-list se denota igual que un atributo. Los demás elementos son los mismos encontrados en la notación de un atributo.

Asociación (rol, multiplicidad, cualificador)

Una asociación en general es una línea que une dos o más símbolos. Pueden tener varios tipos de adornos, que definen su semántica y características. Los tipos de asociaciones entre clases presentes en un diagrama estático son :

-  Asociación binaria
-  Asociación n-aria
-  Composición
-  Generalización
-  Refinamiento

Cada asociación puede presentar algunos elementos adicionales que dan detalle a la relación, como son:

Rol : Identificado como un nombre al final de la línea, describe la semántica de la relación en el sentido indicado. Por ejemplo, la asociación de composición entre Maquina e Ingrediente recibe el nombre de existencias, como rol en ese sentido.

Multiplicidad : Describe la cardinalidad de la relación. En el ejemplo anterior se utilizan 1, 1..*, 5, *, como indicadores de multiplicidad.

Asociación binaria

Se identifica como una línea sólida que une dos clases. Representa una relación de algún tipo entre las dos clases, no muy fuerte (es decir, no se exige dependencia existencial ni encapsulamiento). Un posible ejemplo es la relación entre una compañía y sus empleados

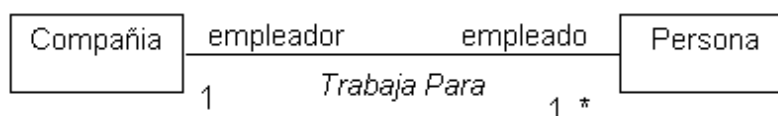


Figura AG 2 Diagrama Asociación Binaria

en este caso la relación recibe el nombre genérico Trabaja Para, la compañía tiene uno o más instancias de la clase Persona denominadas empleado y cada empleado conoce su empleador (en este caso único).

Composición

Es una asociación fuerte, que implica tres cosas :

- ✚ Dependencia existencial. El elemento dependiente desaparece al destruirse el que lo contiene y, si es de cardinalidad 1, es creado al mismo tiempo.
- ✚ Hay una pertenencia fuerte. Se puede decir que el objeto contenido es parte constitutiva y vital del que lo contiene.
- ✚ Los objetos contenidos no son compartidos, esto es, no hacen parte del estado de otro objeto.

Se denota dibujando un rombo relleno del lado de la clase que contiene a la otra en la relación. En el ejemplo inicial de esta hoja se presentan varios ejemplos de relaciones de composición entre Maquina y Producto, Maquina y Deposito-Monedas y Maquina y Deposito-Monedas-Igualas.

Existe también una relación de composición menos fuerte (no se exige dependencia existencial, por ejemplo) que es denotada por una un rombo sin rellenar en uno de los extremos. Un ejemplo puede encontrarse entre Producto e Ingrediente.

Generalización

La relación de generalización denota una relación de herencia entre clases. Se representa dibujando un triángulo sin rellenar en el lado de la superclase. La subclase hereda todos los atributos y mensajes descritos en la superclase. En el ejemplo se encuentra una generalización entre Deposito-Monedas (superclase) y Deposito-Monedas-Igualas (subclase).

Clase paramétrica

Una clase paramétrica representa el concepto de clase genérica en los conceptos básicos OO o de template en C++. Se dibuja como una clase acompañada de un rectángulo en la esquina superior derecha, con los parámetros del caso. Por ejemplo, la clase Lista que utiliza un parámetro formal Tipo se vería de la siguiente manera :

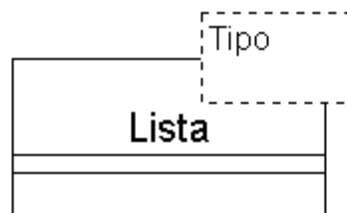


Figura AG3 Diagrama Clase Paramétrica

Paquete

Un paquete es una forma de agrupar clases (u otros elementos en otro tipo de diagramas) en modelos grandes. Pueden tener asociaciones de dependencia o de generalización entre ellos. Un ejemplo puede ser el siguiente:

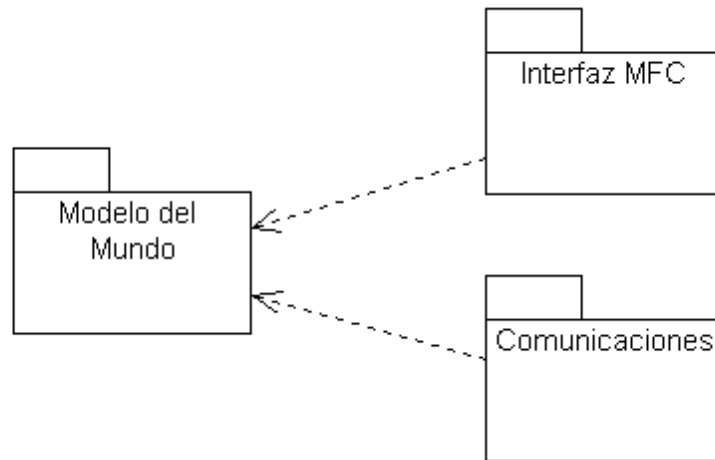


Figura AG 4 Diagrama Paquete

En este caso existen tres paquetes (que se muestran vacíos en este caso, con su contenido encapsulado), con dos de ellos dependiendo del Modelo del Mundo.

Dependencia

Denota una relación semántica entre dos elementos (clases o paquetes, por el momento) del modelo. Indica que cambiar el elemento independiente puede requerir cambios en los dependientes. Se muestra como una línea punteada direccional, indicando el sentido de la dependencia. Puede tener por medio de estereotipos una explicación del tipo de dependencia presentada. En el ejemplo anterior pueden verse dos relaciones de dependencia hacia el paquete Modelo del Mundo.

Nota

Es un comentario dentro de un diagrama. Puede estar relacionado con uno o más elementos en el diagrama mediante líneas punteadas. Pueden representar aclaraciones al diagrama o restricciones sobre los elementos relacionados (cuando el texto se encuentra entre '[y ']'). Se representa mediante un rectángulo con su borde superior derecho doblado. En el ejemplo inicial de esta hoja se encuentran dos notas: Una relacionada con la clase máquina y otra con el depósito de monedas iguales.

2. Conceptos de un diagrama de Casos de Uso

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

Se muestra como ilustración los casos de uso de la máquina de café.

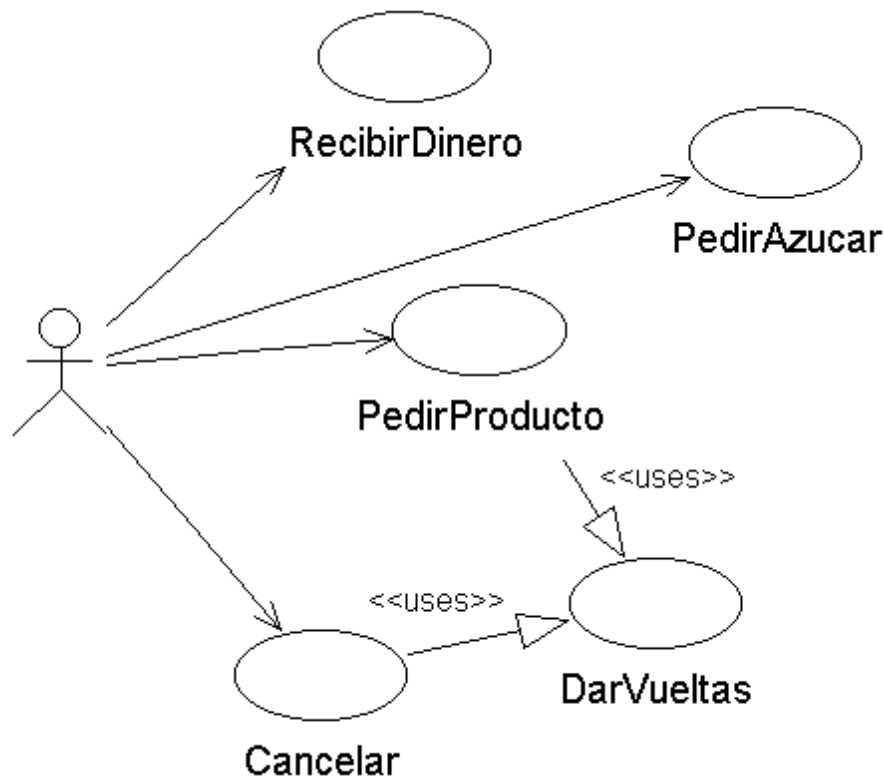


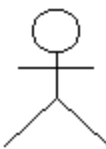
Figura AG 5 Diagrama Caso de Uso

Caso de uso

Se representa en el diagrama por una elipse, denota un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema en un diálogo. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema. Va acompañado de un nombre significativo. En el caso del ejemplo se tienen como casos de uso de la cafetera Recibir-Dinero, Pedir-Azucar, Pedir-Producto, Dar-Vueltas y Cancelar.

Actor

Es un usuario del sistema, que necesita o usa algunos de los casos de uso.

Se representa mediante un , ac de un nombre significativo, si es necesario.

Relaciones en un diagrama de casos de uso

Entre los elementos de un diagrama de Casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas entre ellos (del elemento dependiente al independiente)

Comunica (communicates). Relación entre un actor y un caso de uso, denota la participación del actor en el caso de uso determinado. En el diagrama de ejemplo todas las líneas que salen del actor denotan este tipo de relación.

Usa (uses). Relación entre dos casos de uso, denota la inclusión del comportamiento de un escenario en otro. En el caso del ejemplo el caso de uso Cancelar incluye en su comportamiento Dar-Vueltas; y Pedir-Producto incluye también Dar-Vueltas.

Extiende (extends). Relación entre dos casos de uso, denota cuando un caso de uso es una especialización de otro. Por ejemplo, podría tenerse un caso de uso que extienda la forma de pedir azúcar, para que permita escoger el tipo de azúcar (normal, dietético moreno) y además la cantidad en las unidades adecuadas para cada caso (cucharaditas, bolsitas o cucharaditas, respectivamente). Un posible diagrama se muestra a continuación.



Figura AG 6 Diagrama Caso de Uso (ejemplo Máquina de café)

3. Conceptos básicos en un Diagrama de Secuencia

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

A continuación se muestra un ejemplo de diagrama de secuencia, que da detalle al caso de uso Pedir-Producto del ejemplo de la cafetera.

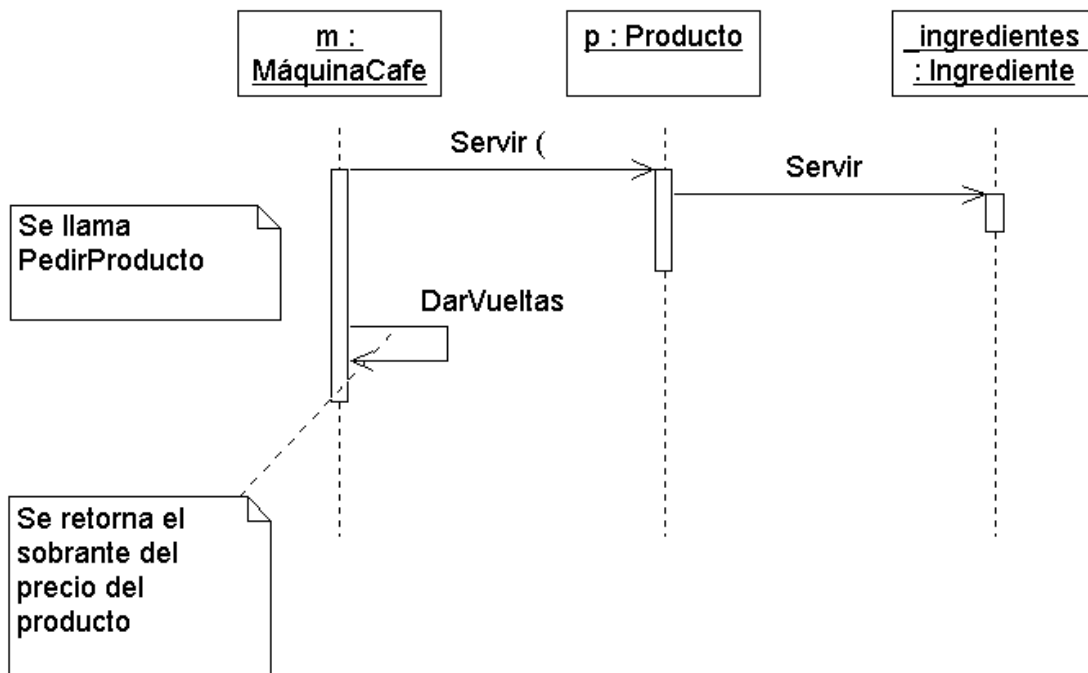


Figura AG 7 Diagrama de Secuencia

Línea de vida de un objeto

Un objeto se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denotan la ejecución de métodos (véase activación). El rectángulo de encabezado contiene el nombre del objeto y el de su clase, en un formato nombreObjeto: nombreClase. Por ejemplo, el objeto m, instancia de la clase MaquinaCafe envía dos mensajes seguidos para dar respuesta a la operación Pedir-Producto : Servir al objeto p de la clase Producto y Dar-Vueltas a sí mismo.

Activación

Muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por sí mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto. En el ejemplo anterior el objeto _ingredientes se encuentra activado mientras ejecuta el método correspondiente al mensaje Servir; el objeto p se encuentra activo mientras se ejecuta su método Servir (que ejecuta _ingredientes.Servir) y el objeto m se encuentra activo mientras se ejecuta p.Servir y DarVueltas.

Mensaje

El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta. En el ejemplo anterior el objeto m envía el mensaje Servir al objeto p y un poco más adelante en el tiempo el objeto m se envía a sí mismo el mensaje DarVueltas.

4. Conceptos básicos en un Diagrama de Colaboración

Un diagrama de colaboración es una forma de representar interacción entre objetos, alterna al diagrama de secuencia. A diferencia de los diagramas de secuencia, pueden mostrar el contexto de la operación (cuáles objetos son atributos, cuáles temporales, ...) y ciclos en la ejecución. Se toma como ejemplo el caso de uso Pedir-Producto ya descrito como diagrama de secuencia.

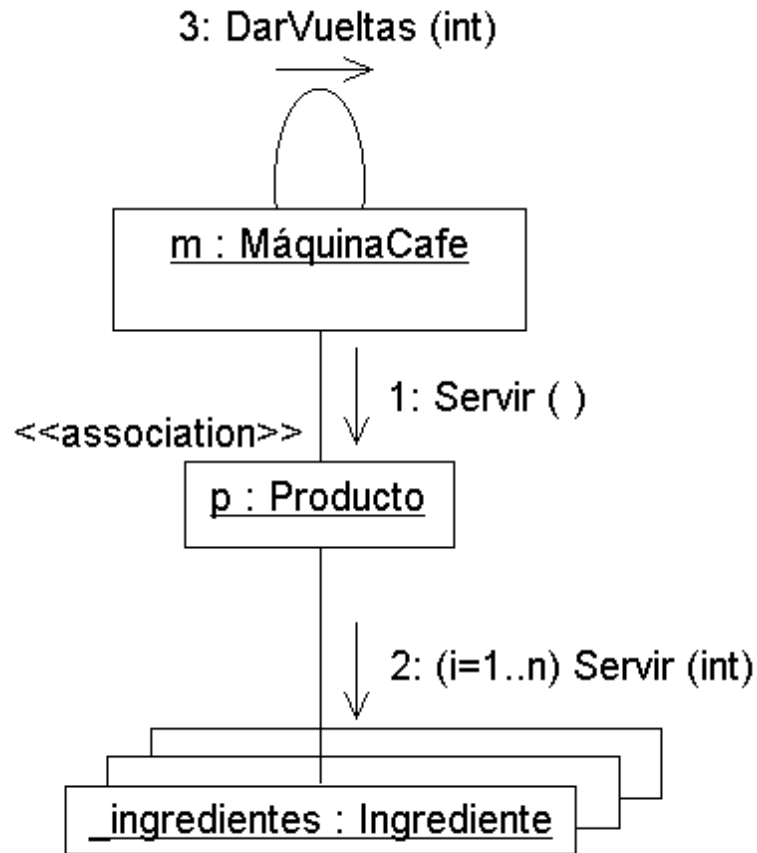


Figura AG 8 Diagrama de colaboración

Objeto

Un objeto se representa con un rectángulo, que contiene el nombre y la clase del objeto en un formato nombreObjeto: nombreClase.

Enlaces

Un enlace es una instancia de una asociación en un diagrama de clases. Se representa como una línea continua que une a dos objetos. Esta acompaña por un número que indica el orden dentro de la interacción y por un estereotipo que indica que tipo de objeto recibe el mensaje. Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones. Los estereotipos indican si el objeto que recibe el mensaje es un atributo (association y se asume por defecto), un parámetro de un mensaje anterior, si es un objeto local o global.

Flujo de mensajes

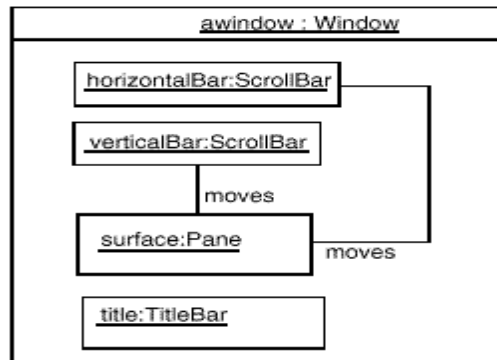
Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cercana a un enlace.

Marcadores de creación y destrucción de objetos

Puede mostrarse en la gráfica cuáles objetos son creados y destruidos, agregando una restricción con la palabra new o delete, respectivamente, cercana al rectángulo del objeto.

Objeto compuesto

Es una representación alternativa de un objeto y sus atributos. En esta representación se muestran los objetos contenidos dentro del rectángulo que representa al objeto que los contiene. Un ejemplo es el siguiente objeto ventana :



5. Conceptos básicos en un Diagrama de Estados

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Un ejemplo en el caso de la cafetera son los estados posibles para la clase MaquinaCafe :

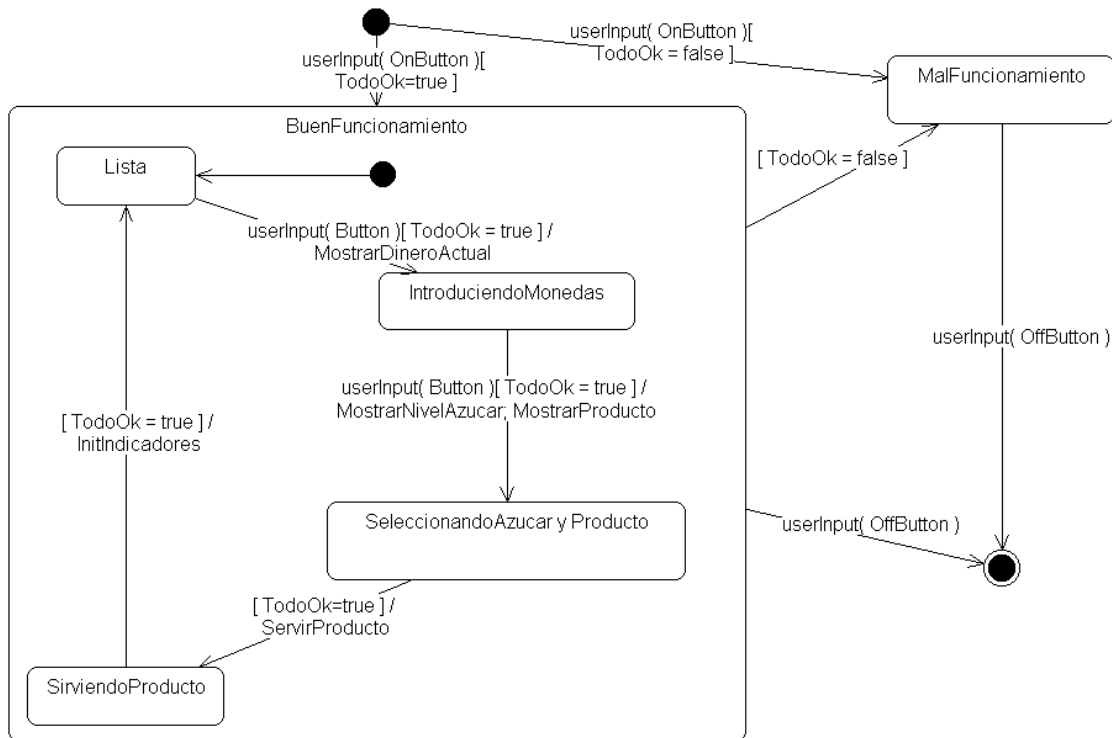


Figura AG 10 Diagrama de Estado

Estado

Identifica un periodo de tiempo del objeto (no instantáneo) en el cual el objeto esta esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos. Se representa mediante un rectángulo con los bordes redondeados, que puede tener tres compartimentos: uno para el nombre, otro para el valor característico de los atributos del objeto en ese estado y otro para las acciones que se realizan al entrar, salir o estar en un estado (entry, exit o do, respectivamente). En el caso del ejemplo anterior, se tienen cuatro estados (EnFuncionamiento, SinCambio, SinIngredientes, MalFuncionamiento), en los cuales se desarrollan ciertas acciones al entrar; por ejemplo, al entrar al estado SinIngredientes se debe realizar la acción "Indicador SinIngredientes en On".

Se marcan también los estados iniciales y finales mediante los símbolos  y , respectivamente.

Eventos

Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas :

- Condición que toma el valor de verdadero o falso.
- Recepción de una señal de otro objeto en el modelo.
- Recepción de un mensaje.

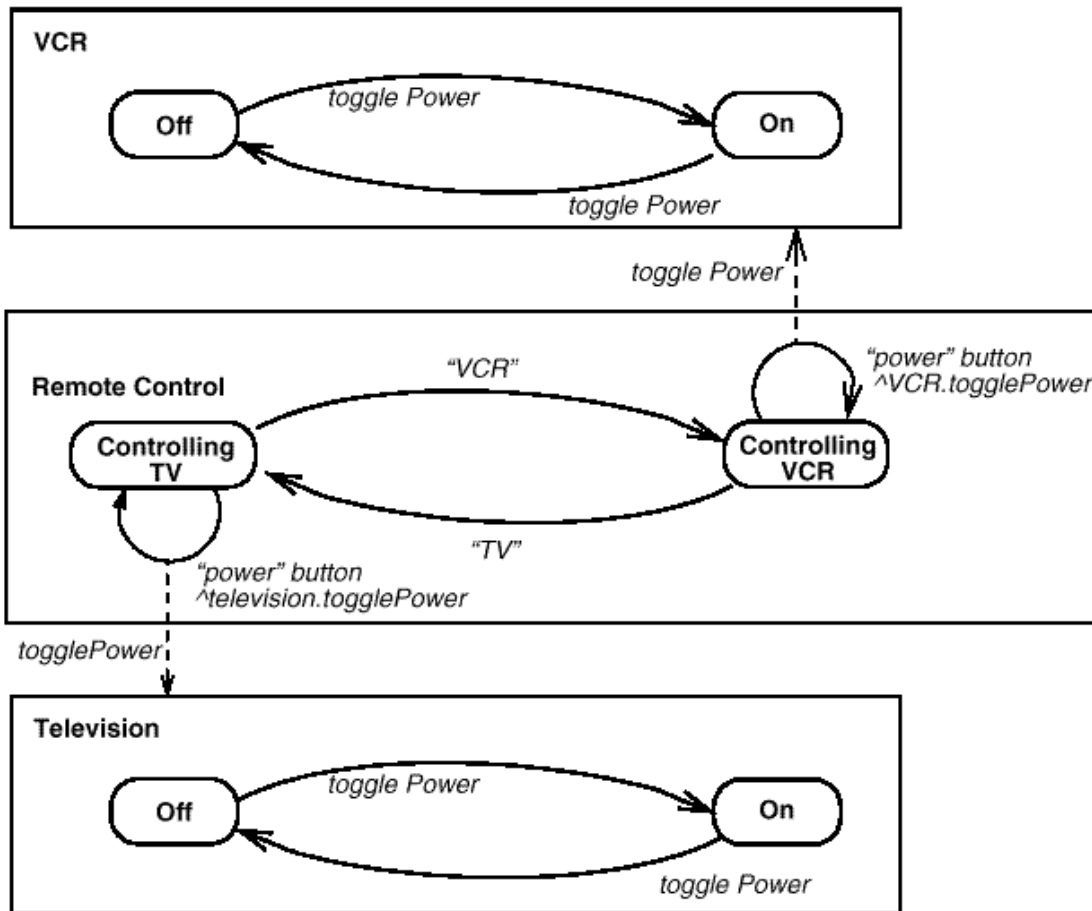
- Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular.

El nombre de un evento tiene alcance dentro del paquete en el cual está definido, no es local a la clase que lo nombre. En el caso del ejemplo anterior se encuentra nombrado en varias transiciones el evento `userInput`, que recibe como parámetro un `Button`, para indicar el botón que ha sido presionado por el usuario de la máquina de café.

Envío de mensajes

Además de mostrar y transición de estados por medio de eventos, puede representarse el momento en el cual se envían mensajes a otros objetos. Esto se realiza mediante una línea punteada dirigida al diagrama de estados del objeto receptor del mensaje. Si tomamos como ejemplo un control remoto que puede enviar órdenes de encender o apagar al televisor o a la videograbadora se puede obtener un diagrama de estados como el de la imagen siguiente.

Los tres aparatos tienen diagramas de estados separados y algunas de las transiciones del control remoto causan el envío de mensajes (`togglePower`) a los otros aparatos.



Transición simple

Una transición simple es una relación entre dos estados que indica que un objeto en el primer estado puede entrar al segundo estado y ejecutar ciertas operaciones, cuando un evento ocurre y si ciertas condiciones son satisfechas. Se representa como una línea sólida entre dos estados, que puede venir acompañada de un texto con el siguiente formato :

event-signature '[' guard-condition] '/' action-expression '^' send-clause,

event-signature es la descripción del evento que da a lugar la transición,

guard-condition son las condiciones adicionales al evento necesarias para que la transición ocurra,

action-expression es un mensaje al objeto o a otro objeto que se ejecuta como resultado de la transición y el cambio de estado, y

send-clause son acciones adicionales que se ejecutan con el cambio de estado, por ejemplo, el envío de eventos a otros paquetes o clases.

En el caso del ejemplo inicial de esta hoja se tiene una transición entre los estados IntroduciendoMoneda y SeleccionadoAzucaryProducto que tiene una transición con el siguiente detalle :

userInput(Button) | [TodoOk=true] / MostrarNivelAzucar, MostrarProducto

El evento que dispara el cambio de estado es userInput(Button) . Se requiere como condición adicional que no se haya detectado ninguna falla (TodoOk = true) y se ejecuta MostrarNivelAzucar y MostrarProducto, que deberían ser ejecutables por el objeto al cual pertenece el diagrama.

Transición interna

Es una transición que permanece en el mismo estado, en vez de involucrar dos estados distintos. Representa un evento que no causa cambio de estado. Se denota como una cadena adicional en el compartimento de acciones del estado.

Supongamos el estado de una interfaz pidiendo password al usuario. En este caso puede tenerse una transición interna que muestre una ayuda al usuario. Esta transición se muestra en el siguiente diagrama con la cadena "help / display help " dentro del cuerpo del estado.

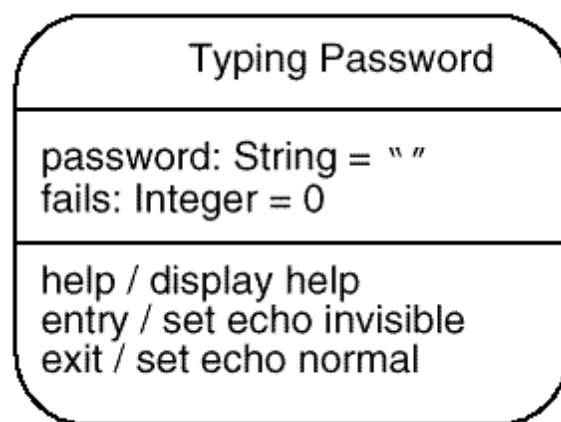


Figura AG 12 Diagrama de transición de estados

6. Conceptos de un Diagrama de Actividades

Un diagrama de actividades es un caso especial de un diagrama de estados en el cual casi todos los estados son estados de acción (identifican que acción se ejecuta al estar en él) y casi todas las transiciones son enviadas al terminar la acción ejecutada en el estado anterior. Puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Sirven para representar transiciones internas, sin hacer mucho énfasis en transiciones o eventos externos. Se presenta a continuación un ejemplo de diagrama de actividades para un mensaje de un objeto. Generalmente modelan los pasos de un algoritmo.

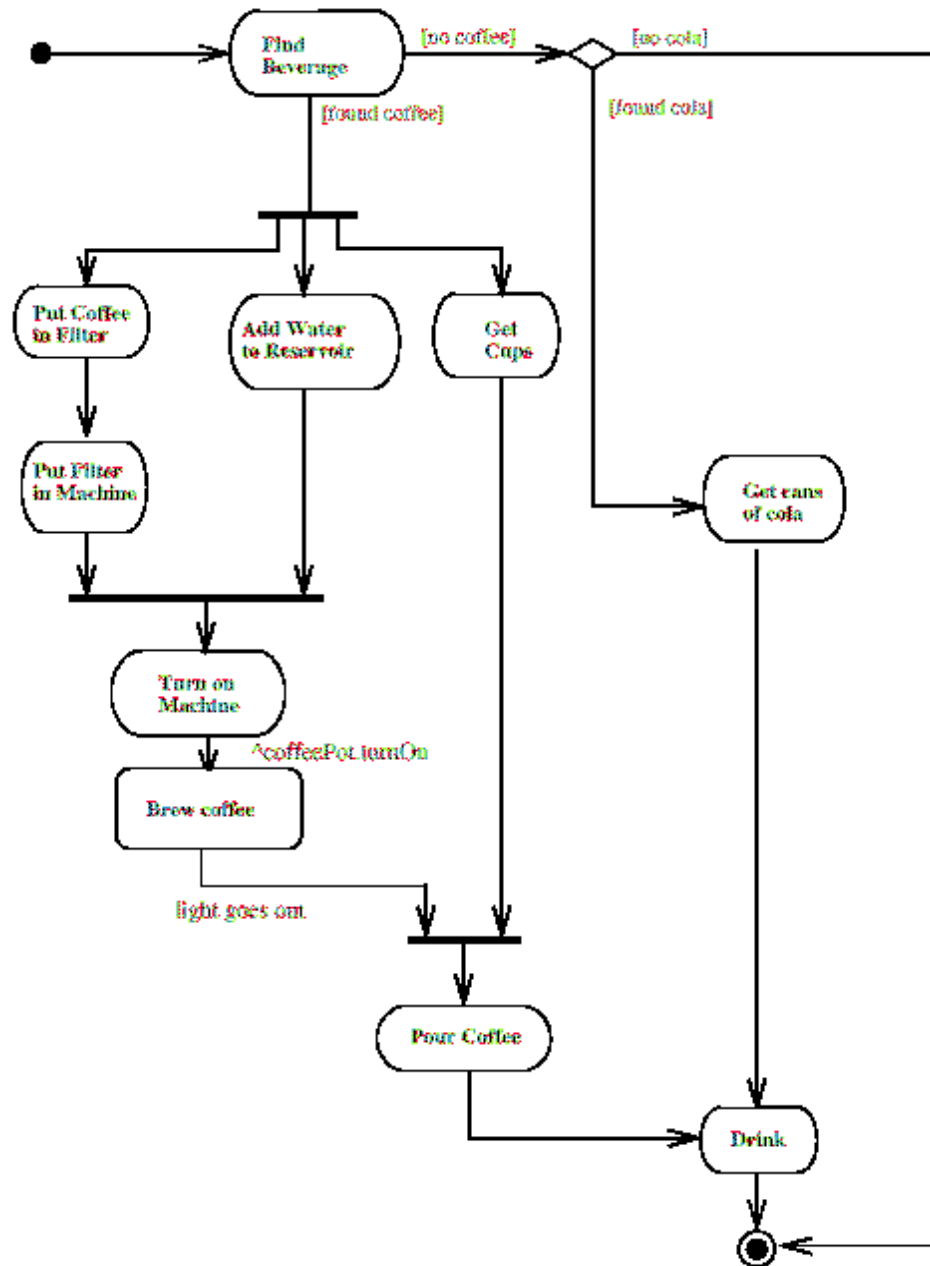


Figura AG 13 diagrama de Transición de estados

Estado de acción

Representa un estado con acción interna, con por lo menos una transición que identifica la culminación de la acción (por medio de un evento implícito). No deben tener transiciones internas ni transiciones basadas en eventos (Si este es el caso, représentelo en un diagrama de estados). Permite modelar un paso dentro del algoritmo.

Se representan por un rectángulo con bordes redondeados.

Transiciones

Las flechas entre estados representan transiciones con evento implícito. Pueden tener una condición en el caso de decisiones.

Decisiones

Se representa mediante una transición múltiple que sale de un estado, donde cada camino tiene un label distinto. Se representa mediante un diamante al cual llega la transición del estado inicial y del cual salen las múltiples transiciones de los estados finales. Un ejemplo se ve en la figura cuando no hay café y se toma una decisión entre hay cola o no hay cola.

7. Conceptos en un Diagrama de Implementación

Un diagrama de implementación muestra la estructura del código (Diagrama de componentes) y la estructura del sistema en ejecución (Diagrama de ejecución).

Diagrama de Componentes

Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables. Los componentes software tienen tipo, que indica si son útiles en tiempo de compilación, enlace o ejecución. Se consideran en este tipo de diagramas solo tipos de componentes. Instancias específicas se encuentran en el diagrama de ejecución.

Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (generalmente de compilación). Puede mostrar también contención de entre componentes software e interfaces soportadas. Un ejemplo es el siguiente :

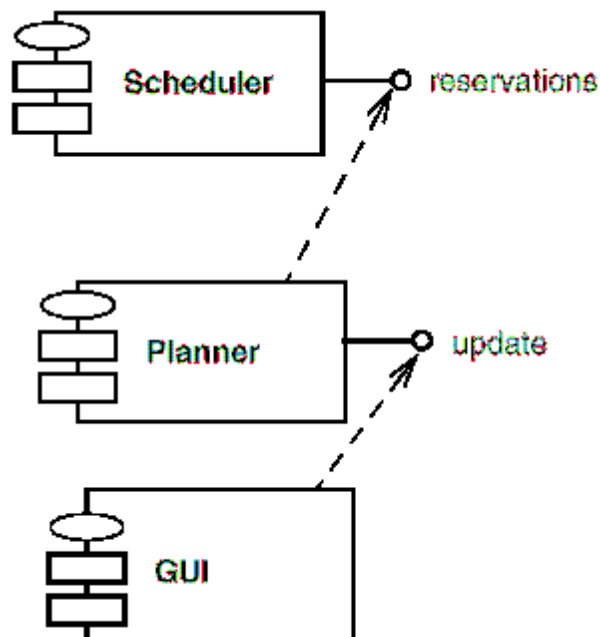


Figura AG 14 Diagrama de Implementación

En este caso tenemos tres componentes, GUI dependiendo de la interfaz update provista por Planner, Planner dependiendo de la interfaz reservations provista por Scheduler.

Diagrama de Ejecución

Un diagrama de ejecución muestra la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software, procesos y objetos que se ejecutan en ellos. Instancias de los componentes software representan manifestaciones en tiempo de ejecución del código. Componentes que solo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes.

Un diagrama de ejecución es un grafo de nodos conectados por asociaciones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (un caso particular de un objeto). Las instancias de componentes software pueden estar unidos por relaciones de dependencia, posiblemente a interfaces.

Un ejemplo de diagrama de ejecución es el siguiente :

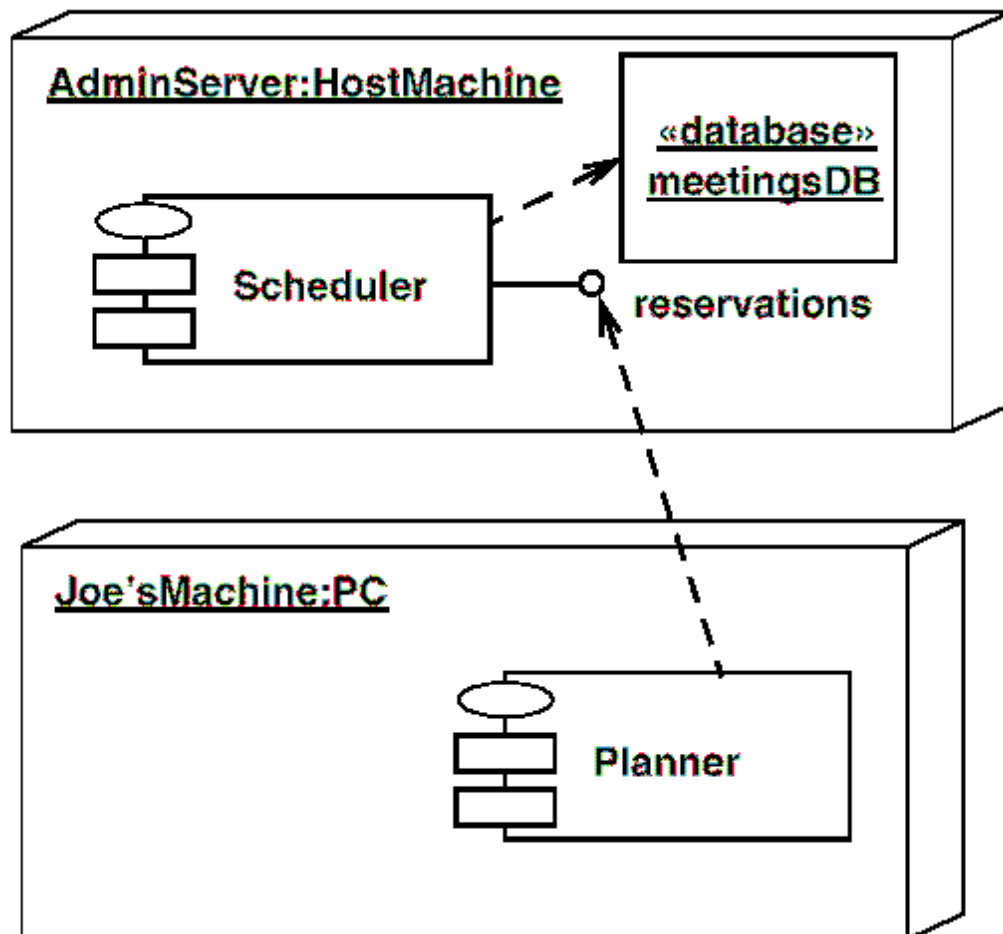


Figura AG 15 Diagrama de Ejecución

En este caso se tienen dos nodos, AdminServer y Joe'sMachine. AdminServer contiene la instancia del componente Scheduler y un objeto activo (proceso) denominado meetingsDB. En Joe'sMachine se encuentra

la instancia del componente software Planner, que depende de la interfaz reservations, definida por Scheduler.

Nodos

Un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. Pueden representarse instancias o tipos de nodos. Se representa como un cubo 3D en los diagramas de implementación.

Componentes

Un componente representa una unidad de código (fuente, binario o ejecutable) que permite mostrar las dependencias en tiempo de compilación y ejecución. Las instancias de componentes de software muestran unidades de software en tiempo de ejecución y generalmente ayudan a identificar sus dependencias y su localización en nodos. Pueden mostrar también que interfaces implementan y qué objetos contienen. Su representación es un rectángulo atravesado por una elipse y dos rectángulos más pequeños. Un ejemplo de componente que implementa dos interfaces es:

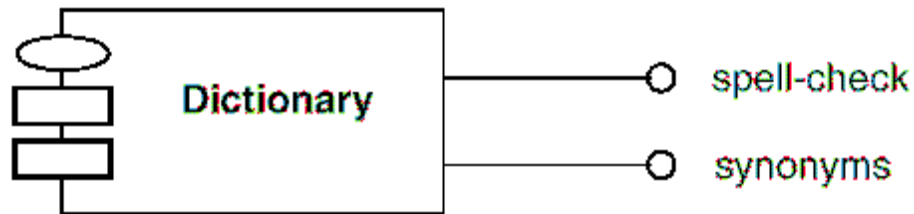


Figura AG 16 Diagrama de componente

I. Conceptos avanzados en un diagrama de estructura estática

Los conceptos que a continuación se presentan muestran posibilidades adicionales de descripción en un modelo orientado por objetos. Aunque los conceptos fundamentales y más frecuentemente usados se describieron en los conceptos básicos, éstos completan el poder de expresión.

Estereotipo

Las clases (y demás elementos notacionales en los diagramas) pueden estar clasificados de acuerdo a varios criterios, como por ejemplo su objetivo dentro de un programa. Esta clasificación adicional se expresa mediante un estereotipo.

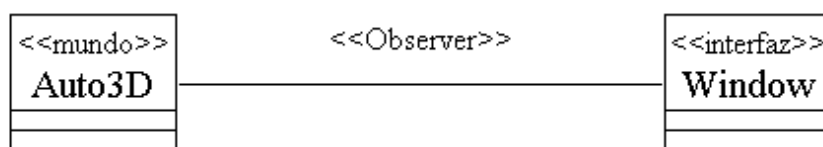


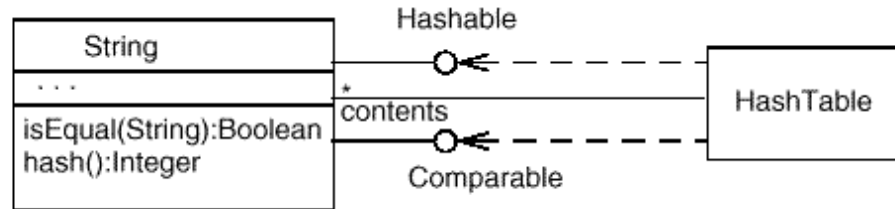
Figura AG 17 Diagrama de componente con notación de Estereotipos

En el ejemplo anterior Auto3D está clasificado con el estereotipo Mundo, y la clase Window con el de interfaz. Nótese que las relaciones pueden tener esta clasificación también. En este caso la relación se identifica como Observer.

Interfaz

Una interfaz es un protocolo exigible a una clase. La representación de este concepto es por medio de una línea terminada en un círculo :

Figura AG 18 Diagrama Interfaz



En el ejemplo anterior, la clase **String** se utiliza dentro de un **hashtable**, gracias a que implementa la interfaz **Hashable** (el método `hash`) y la interfaz **Comparable** (el método `isEqual`).

Asociación or

En algunas ocasiones es necesario describir que una clase esta relacionada con un objeto de una u otra clase. Esto se denota por medio de una relación **or** exclusiva. Su representación es una línea punteada que une dos asociaciones, junto con la aclaración (por medio de una propiedad) del tipo de asociación.

En el ejemplo anterior un automóvil puede tener como dueño una persona natural o una empresa (pero no ambos).

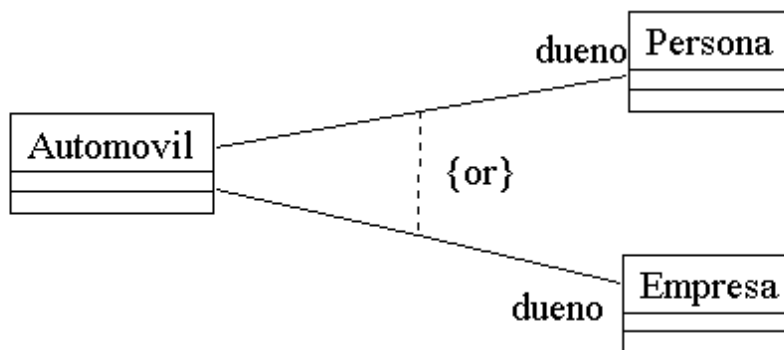


Figura AG 19 Diagrama Interfaz ejemplo excluyendo interfaces por medio del estereotipo OR

Clase de asociación

Una clase de asociación es información de detalle. Se denota como una clase relacionada por una línea punteada a una asociación.

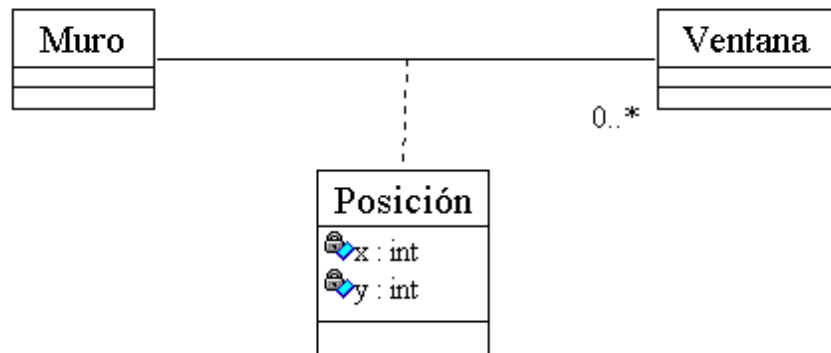


Figura AG 20 Diagrama de clases con información de detalle

En este caso existe una relación entre Muro y Ventana, la cual tiene como detalle un objeto de la clase Posición. Cabe notar que este objeto no podría tomarse como atributo de Muro o Ventana, ya que el contexto de su existencia esta dado precisamente por la relación entre las dos clases.

Asociación n-aria

Es una forma de expresar una relación entre tres o más clases. Se representa como un diamante del cual salen líneas de asociación a las clases.

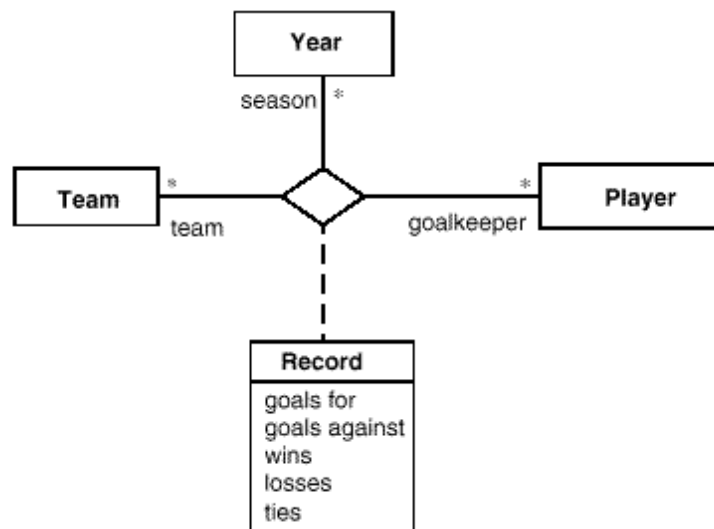


Figura AG 21 Diagrama de clases con mas de una relación

En este caso se tiene una relación ternaria entre las clases Year, Team y Player. A cada terna de objetos <año, equipo, jugador> corresponde un objeto de tipo Record (clase de asociación).

Otros conceptos

Aunque la metodología los incluye, se dejan fuera del alcance de este resumen los siguientes conceptos de UML: Calificador, Compartimiento de Nombre y de Lista, Propiedad, Expresión de Tipo, Elemento acotado, Tipo, Utilidad, Metaclase, Caminos de composición de clases, Relación de refinamiento, Elemento derivado y Expresión de navegación.

II. Conceptos avanzados en un Diagrama de Secuencia

Tiempos de transición

En un ambiente de objetos concurrentes o de demoras en la recepción de mensajes, es útil agregar nombres a los tiempos de salida y llegada de mensajes. Analizando la recepción de una llamada telefónica puede tenerse un diagrama como el siguiente :

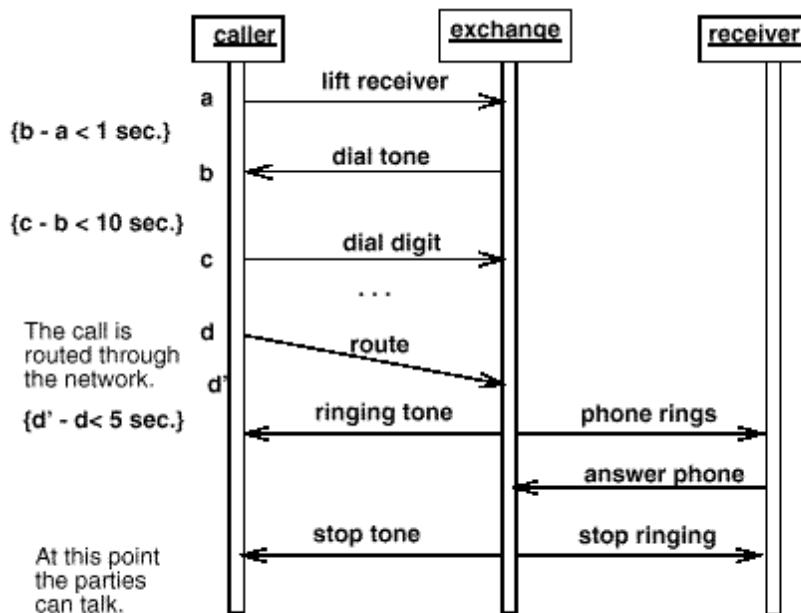


Figura AG 22 Diagrama de clases con tiempos de transición

En este diagrama se tienen tres objetos concurrentes, el que hace la llamada, la central telefónica y el que recibe la llamada. se nombran los tiempos de los mensajes que envía o recibe el caller (a para descolgar, b para el tono de la llamada, c para la marcación, d para el inicio del enrutamiento de la llamada, d' para la finalización del enrutamiento). Estos nombres o tiempos de transición permiten describir restricciones de tiempo (por ejemplo $b - a < 1 \text{ sec.}$) o demoras entre el envío y la recepción (entre d y d').

Condiciones caminos alternativos de ejecución. Concurrencia

En algunos casos sencillos pueden expresarse en un diagrama de secuencia alternativas de ejecución. Estas alternativas pueden representar condiciones en la ejecución o diferentes hilos de ejecución (threads).

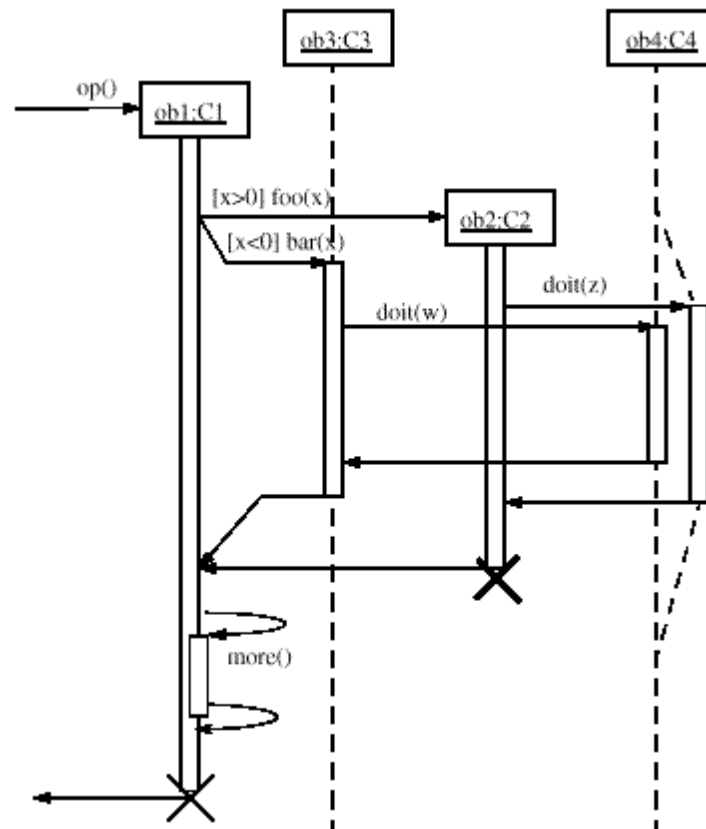


Figura AG 23 Diagrama de secuencia con alternativas de ejecución

En el diagrama anterior se muestran dos casos. ob1 muestra una condición al enviar un mensaje a ob3 o a ob2, dependiendo de si $x > 0$ o $x < 0$. Estas dos líneas de ejecución se vuelven a unir más adelante, indicando el fin del condicional. Por otra parte ob4 muestra dos posibles operaciones dependiendo de si se siguió la condición $x > 0$ o $x < 0$. Ya que se presentan en el mismo instante de tiempo, se requiere dividir la línea del objeto en dos (esta misma representación se utiliza para el caso de dos hilos de ejecución).

Dstrucción de un objeto

Se representa como una X al final de la línea de ejecución del objeto. Por ejemplo, en el diagrama anterior se muestra el final de ob2 y de ob1.

Métodos recursivos

Un ejemplo de un método recursivo es el método more en ob1. Es un rectángulo un poco salido de la activación principal y con líneas de llamado de mensajes, que indican la entrada y salida de la recursión.

III. Conceptos avanzados en un Diagrama de Colaboración

Patrón de diseño

Un diagrama de colaboración puede especificar un contrato entre objetos, parte esencial para la descripción de un patrón de diseño. Este diagrama contiene todos los elementos citados de un diagrama de colaboración, dejando libres posiblemente los tipos exactos de algunos objetos o con nombres genéricos para los mensajes.

Una "instanciación" del patrón se representa como una elipse unida mediante flechas punteadas a los objetos o clases que participan realmente en el patrón. Estas flechas pueden tener roles, indicando cuál es el papel de cada elemento dentro del patrón. Por ejemplo, una instanciación del patrón de observador puede verse como :

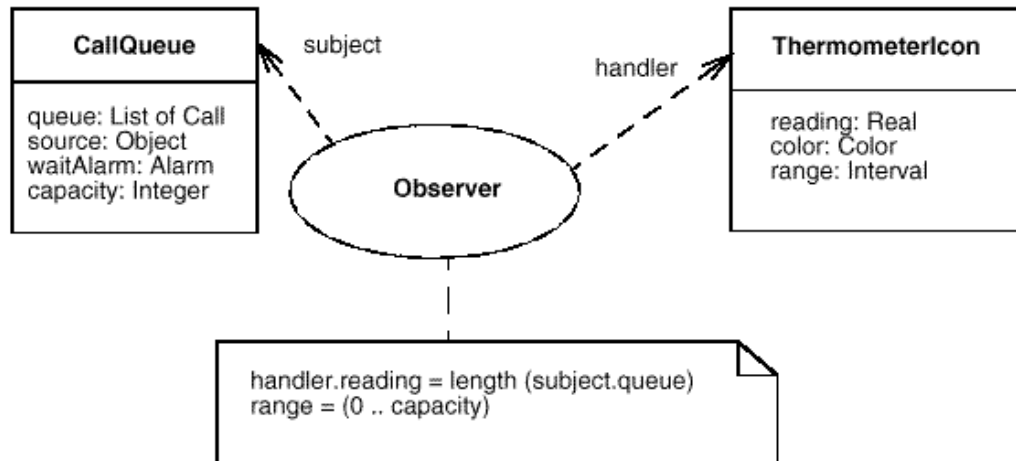


Figura AG 24 Diagrama de colaboración

Contexto

Un contexto es una vista de uno o más elementos dentro del modelo que colaboran en el desarrollo de una acción. Se usa para separar los demás elementos en el modelo de este problema en particular y darle énfasis. Puede mostrar solo los detalles relevantes de las clases u objetos que contiene, para resaltar su utilidad. Un ejemplo es la definición del siguiente tipo :

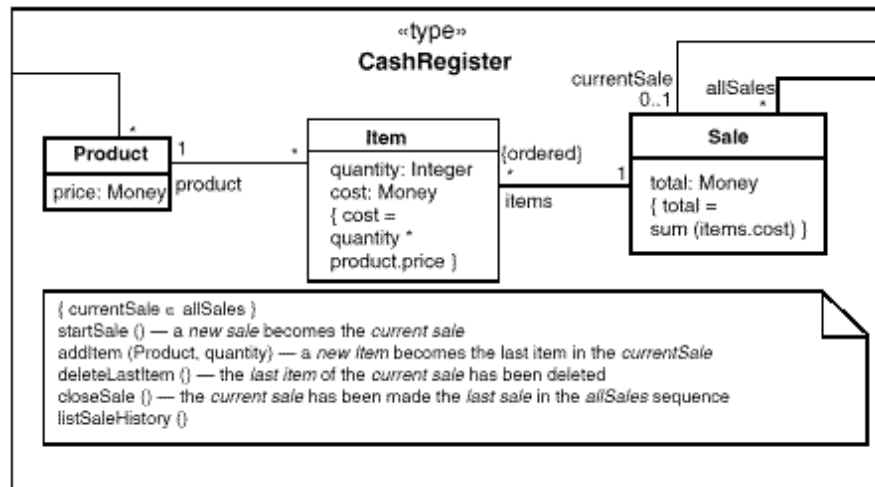


Figura AG 25 Diagrama de contexto en un diagrama de colaboración.

Se representa como un contexto un tipo Registro de Dinero y se muestran los detalles relevantes de Producto, Item y Venta para este tipo. Las relaciones de las clases con otras no visibles dentro del contexto pueden omitirse o conectarse al borde del contexto.

Objeto activo

Un objeto activo es el que contiene su propio flujo de control, a diferencia de un objeto pasivo que encapsula datos y solo reacciona al enviarle mensajes. Un objeto activo se representa con un rectángulo de bordes gruesos. Puede contener otros objetos pasivos o activos. Se presenta a continuación un ejemplo en el contexto de una producción en línea robotizada. Se tiene un ente administrador, un robot y un horno (tres objetos activos) que interactúan para desarrollar su tarea.

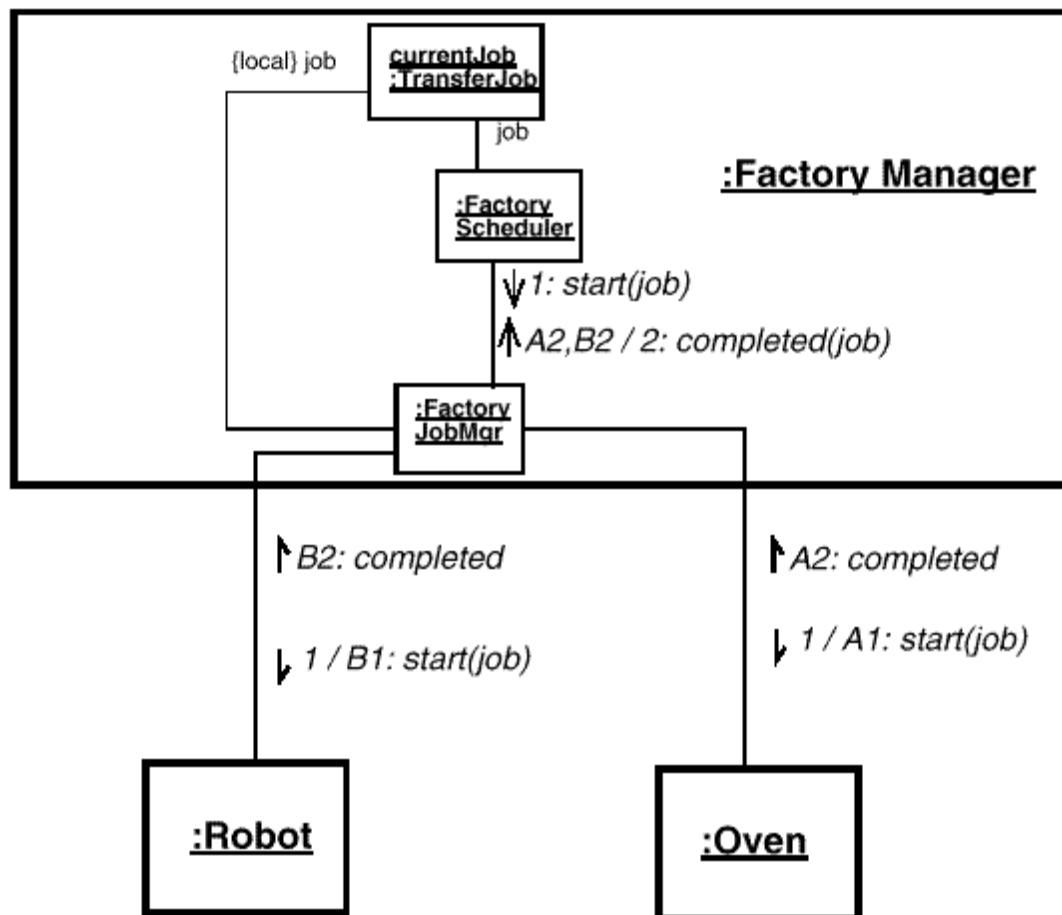


Figura AG 26 Diagrama Objeto activo

Los mensajes entre objetos pasivos se denotan mediante una flecha completa, mientras que los mensajes entre objetos activos se denotan con una media flecha. Los threads de ejecución se denotan con las letras A y B antes del número de orden del mensaje. La sincronización entre threads se muestra mediante un '/' y el nuevo número de orden. Por ejemplo en A2, B2 / 2: completed (job).

Conceptos avanzados en un Diagrama de Estados

Subestados

Un estado puede descomponerse en subestados, con transiciones entre ellos y conexiones al nivel superior. Las conexiones se ven al nivel inferior como estados de inicio o fin, los cuales se suponen conectados a las entradas y salidas del nivel inmediatamente superior. Un ejemplo es el estado marcando de un teléfono puede descomponerse en Inicio y marcado parcial, como lo muestra la siguiente figura :

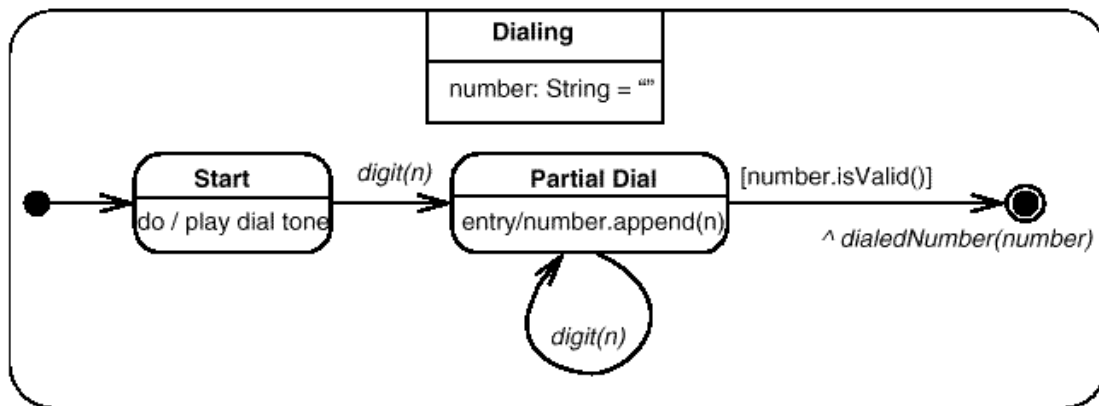
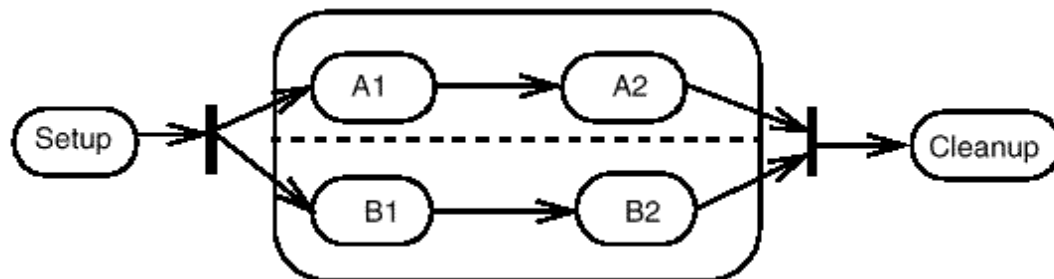


Figura AG 27 Diagrama de Estados (Subestado)

Transición compleja

Una transición compleja relaciona tres o más estados en una transición de múltiples fuentes y/o múltiples destinos. Representa la subdivisión en threads del control del objeto o una sincronización. Se representa como una línea vertical del cual salen o entran varias líneas de transición de estado. En el ejemplo que sigue se muestra una transición a dos threads concurrentes que luego se sincronizan.

Figura AG 28 Diagrama de Estados (Transición compleja)



Transición a estados anidados

Una transición de hacia un estado complejo (descrito mediante estados anidados) significa la entrada al estado inicial del subdiagrama. Las transiciones que salen del estado complejo se entienden como transiciones desde cada uno de los subestados hacia afuera (a cualquier nivel de profundidad).

En la hoja inicial de diagramas de secuencias se encuentran como primer ejemplo los dos casos nombrados: Desde el estado inicial se pasa al estado BuenFuncionamiento (a su estado inicial) y de este estado salen transiciones hacia MalFuncionamiento y hacia el estado final, que deben comprenderse como transiciones de cada uno de los estados internos hacia estos últimos.

SOA Governance: Metodologías de Administración de Proyectos

Parte fundamental del “SOA Governance” es organizar la administración de proyectos informáticos, y el primer paso es definir una metodología acorde con esta nueva Arquitectura.

La dirección de proyectos informáticos es una disciplina compleja, y una forma de manejar esta complejidad es adoptar una estructura de dirección de proyectos (PMS: Project Management Structure), es decir, adoptar una metodología.

Existen varios estándares:

Metodología	Descripción	Enlace
PMI PMBOK	Project Managment Body Of Knowledge) Este es la metodología propuesta por la asociación Project Managment Institute (PMI), es un estándar ampliamente difundido en EEUU	www.pmi.org
PRINCE2	(Projects IN Controlled Enviroments)Es la metodología propuesta por el Gobierno Ingles, y ampliamente difundida en Europa	www.ogc.gov.uk/methods_prince_2.asp
RUP	(Rational Unified Process)	
XP	(eXtreme Programing) es la metodología mas difundida de la asociación Agile, que agrupa varias metodologías de respuesta rápida y altamente flexibles.	
CMMI	(Capability Maturity Model Integration). Es un método de mejoramiento a los procesos	http://www.sei.cmu.edu/cmmi/cmmi.html
P2M	(Project & Program Management for Enterprise Innovation) es el estándar Japonés	http://www.pmaj.or.jp/ENG/P2M_Download.htm
V-Modell	Es el modelo alemán promovido por el Gobierno, y el ministerio de defensa de ese país.	www.v-modell.iabg.de/
HERMES	Adaptación del modelo alemán V-Modell promovido por el Gobierno Suizo.	http://www.hermes.admin.ch/

Tabla AG 1 Metodologías de administración de proyector

Estas metodologías intentan resolver las siguientes preguntas para un Proyecto:

- ✚ Quien? (roles dentro de un proyecto)
- ✚ Que? (procesos, actividades y entregables dentro de un proyecto)
- ✚ Cuando? (plan de un proyecto, oportunidad, reglas de decisión)
- ✚ Como? (como se asignan roles, como se realizan actividades, herramientas)

Las distintas metodología resuelven en mayor o menor grado dichas preguntas, por ejemplo se indica que CMMI resuelve principalmente el “Que”, mientras que otras metodologías como V-Modell resuelven en forma mas completa todas las preguntas, por eso algunos expertos consideran CMMI como solo un estándar mas que una metodología, en otras palabras, se puede alcanzar un nivel de madurez CMMI usando metodologías como RUP o V-Modell, de hecho la metodología PMI PMBOK tiene su propio modelo de madurez OPM3 (Organizational Project Management Maturity Model).

Al momento de seleccionar una metodología hay que tener las siguiente consideraciones:

No hay una única tecnología mejor que todas, ya que algunas se adaptan mejor que otras a una Empresa u organización, y esto depende de la tolerancia al riesgo de la empresa, su nivel de madurez, su tamaño, etc. Todas las metodología manejan las grandes etapas del ciclo de vida de un proyecto, lo que cambia entre ellas es el nivel de disciplina.

Otro aspecto a tomar en cuenta es que toda metodología se debe adaptar a la Organización, e incluso se pueden combinar las metodologías porque generalmente no son excluyentes, logrando una metodología mas ajustada a la realidad de la Organización.

Hay que considerar que las metodologías no lo son todo por si mismas, hay que prepara el ambiente, y preparar a los profesionales, la empresa debe promover la cultura de "Project Managment".

Las metodologías en boga son; PRINCE2 y PMBOK, y sus principales características son:

	PRINCE2	PMI PMBOK
Enfoque	basado en lo que un jefe de proyecto "debe hacer", define los paso a seguir para lograr un proyecto exitoso.	basado en lo que un jefe de proyectos "debe saber", enfocado en lo estandares.y buenas practicas.
Estilo	menos prescriptivo, deja el "Como" mas abierto.	mas prescriptivo, espera que se realicen ciertas tareas, usando tecnicas especificas.
Control de Proyecto	Menos riguroso	Mas riguroso
Necesidad del Negocio (Bussiness Case)	Fuertemente guiado los objetivos del negocio.	Basado más en el "know how" (conocimiento) que en su aplicación, luego esta menos enfocado en las necesidades del negocio.
Estructura del proyecto	Enfasis en la estructura del proyecto.	No hay una estructura definida.
Entregables	Se concentra en lod entregables, y omite aspectos de manejo de acuerdos, y de recursos humanos.	Menos concentrado en los entregables, y cubre los aspectos de manejo de acuerdos, y de recursos humanod
En Resumen	Se enfoca en la forma en que un proyecto se desarrolla.	Se enfoca en las habilidades del Jefe de Proyectos.

Tabla AG 2 Característica de PRINCE2 y PMBOK

Actualmente el tema de Project Management orienta más su investigación en las PMO (Project Management Office) y en las herramientas de apoyo para la administración de proyectos (Project Management Applications). Las PMO son centros de excelencia que apoyan y supervisan a los Jefes de Proyectos, y a la actividad de Administración de Proyectos, para el caso d SOA la PMO es el centro de excelencia de Arquitectura (SOA COE).

Apéndice H Especificación general del sistema SisaXml

Control de versiones

Versión	Fecha	Elaborado	Cve Control de cambio	Modificacion	Observaciones
1.0.0	04/Nov/2002	Jplacido	CveC-MsjXml-01		Intercambio de mensajes por Internet con el formato Xml
8.7.0.0	20/Abril/2004	Jplacido	CveC-MsjXml-02		Descripción de ambientes activos para el Mensaje Universal Xml. Actualizaciones afectaciones con la interfaz Pisa – Sisa Manto. Actualización a mensajes para el alta de quejas por Internet.
9.0.0.0	28/Oct/2004	Jplacido	CveC-MsjXml-03		Recepción de mensajes Xml

Tabla H.1 Control de Versiones

Introducción

Se requiere desarrollar una herramienta con la cual los clientes de TELMEX que tienen contacto con el sistema SISA se puedan comunicar de la manera más sencilla y sin invertir mucho en la comunicación utilizando un medio publico como es Internet.

La herramienta nos debe proporcionar la facilidad para que las entidades internas a TELMEX puedan hacer uso de esta herramienta y se lleve un intercambio de información complementaria entre las aplicaciones.

La aplicación esta basada en el estándar J2EE así como en la codificación de archivos Xml para la Definición de Tipo de Datos (DTD)

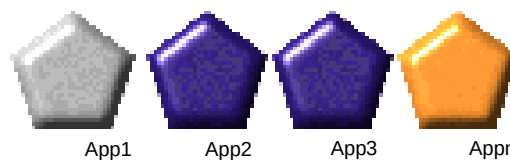
Por lo tanto en este documento se detallaran las reglas para el manejo de los mensajes tanto de entrada como de salida, asi como los mecanismos para detectar los errores de comunicación entre los tres sistemas, es decir de las aplicaciones externas, el mensaje universal xml y la aplicación SISA.

El detalle para el manejo de los mensajes y el tratamiento de la información de entrada como la manipulación y el tratamiento de los mensajes de salida se detallan individualmente dependiendo de la afectación con la aplicación SISA.

Situación actual

Actualmente no se tiene interacción entre Aplicaciones en Internet y el sistema SISA

Aplicaciones en Internet



Intranet Telmex



Figura H.1 Interfaz SISA – INTERNET Situación Actual

Solución propuesta

Aplicaciones en Internet



Intranet Telmex



Figura H.2 Interfaz SISA – INTERNET Solución propuesta

Facilitar el intercambio de información entre aplicaciones en Internet y el sistema corporativo de Telmex SISA, proporcionando flexibilidad en la comunicación y agilidad en el servicio de atención a clientes desde Internet.

Planteamiento

El primer problema que se pretende solucionar con el mensaje universal xml es que algunos usuarios por el numero de servicios contratados tienen sus propios sistemas para controlar los problemas en sus redes y llevar un control con Telmex, a los usuarios les ayudaría tener una herramienta que Interface sus aplicaciones con el modulo de mantenimiento de SISA ya que la queja se capturaría una sola ves y recuperar el avance del mismo el cual se refleje automáticamente desde SISA, sin capturar nuevamente esta información.

La segunda aplicación es poder levantar quejas de servicios básicos que se encuentran en la base de datos de PISA (Proyecto de Integración de Sistemas a Cliente) que en SISA se conocen como servicios básicos, se pretende que la información que proporcione PISA la consulten en SISA como propia

Esta interfaz reduciría el tiempo de atención de las quejas ya que se detectaría en línea que el servicio pertenece a un servicio levantado en PISA, además que se elimina el paso de levantar la queja de manera manual en la aplicación teniendo el riesgo de errores de captura al duplicar el trabajo en el levantamiento de la queja

Objetivo de la tarea

Crear una herramienta por medio de mensajes XML que utilice un formato estándar que puede integrarse de manera fácil y sin requerir que se instalen componentes externos y se pueda relacionar de manera fácil a la aplicación SISA y con poco mantenimiento.

Para ellos se desarrollaran varios programas en la base de datos que nos ayude a integrar la información recibida y enviada (interfaz con la aplicación SISA), esta herramienta debe ser capaz de ejecutar programas externos que nos proporcionen la información requerida, se requiere desarrollar una aplicación para el monitoreo de los mensajes que se transmitan al ciber – espacio.

Interfaz SISA

- Programa Java que identifique si es un mensaje de entrada o salida.
- Si es un mensaje de entrada identifique si se debe almacenar en el modulo de mantenimiento como queja.
- Si es un mensaje de salida o notificación, en esta fase se determina el cliente donde se envía el avance del servicio.

Aplicación Sisa

- Preparación de las pantallas en SISA para que en determinada acción o regla que se ejecute se cree el mensaje xml.
- Al momento de ejecutarse la acción o se cumpla con la regla definida en la base de datos preparada para el mensaje universal se escriba en las tablas de salida.

Monitor de Interfaz SISA

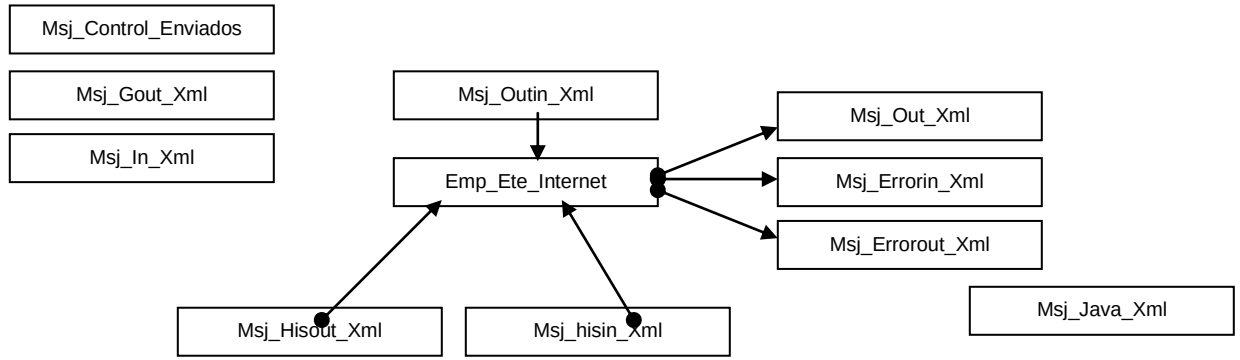
- Pantallas de monitoreo de mensajes. ayuda a identificar rápidamente cual es el error ocurrido al tratar de ingresar las quejas desde el Msj Univ Xml indicarle cual es el problema y como se puede corregir, se propone que el monitor del Mensaje Universal Xml se realice en WEB (por medio servlets, java – script, etc) para tener un interacción desde cualquier parte por medio de un navegador de internet.

Inicial (INI)

Acción	Obligatorio	Regla
Proceso que arranque el Mensaje Universal Xml, con un temporizador variable en el que espere para la recepción y envío de mensaje.	Si	RTMsjXml-Ini-01

Diagrama entidad relación (DGER)

Control de Mensajes Xml



Estructura de Mensajes Xml

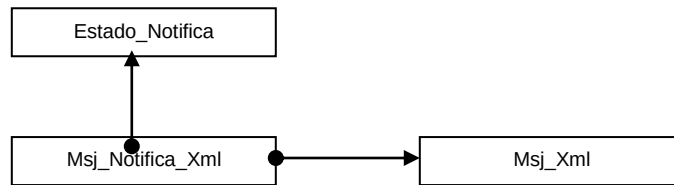


Diagrama Casos de Uso (DCU)

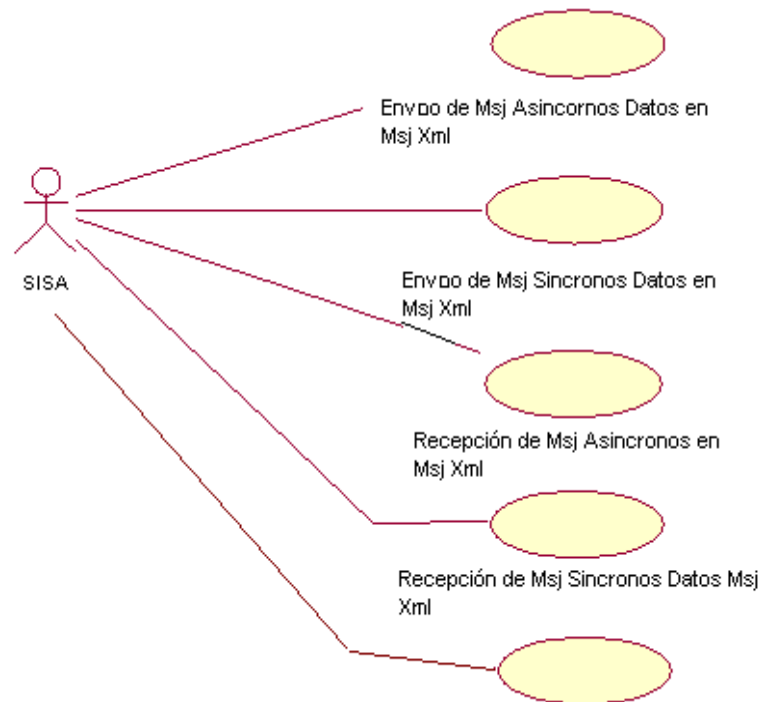


Figura H.3 Diagrama casos de uso

Comentario	Regla
Envío de Mensajes Asíncronos por medio del Mensaje Universal Xml	RTMsjXml.Dgcl-01
Envío de Mensajes Sincronos por medio del Mensaje Universal Xml	RTMsjXml.Dgcl-01
Recepción de Mensajes Asíncronos por medio del Mensaje Universal Xml	RTMsjXml.Dgcl-02
Recepción del Mensaje Sincronos por medio del mensaje universal Xml	RTMsjXml.Dgcl-02

Tabla H.2 Descripción general y reglas de negocio diagrama casos de uso

Diagrama de Transición de Estados (DGTE)

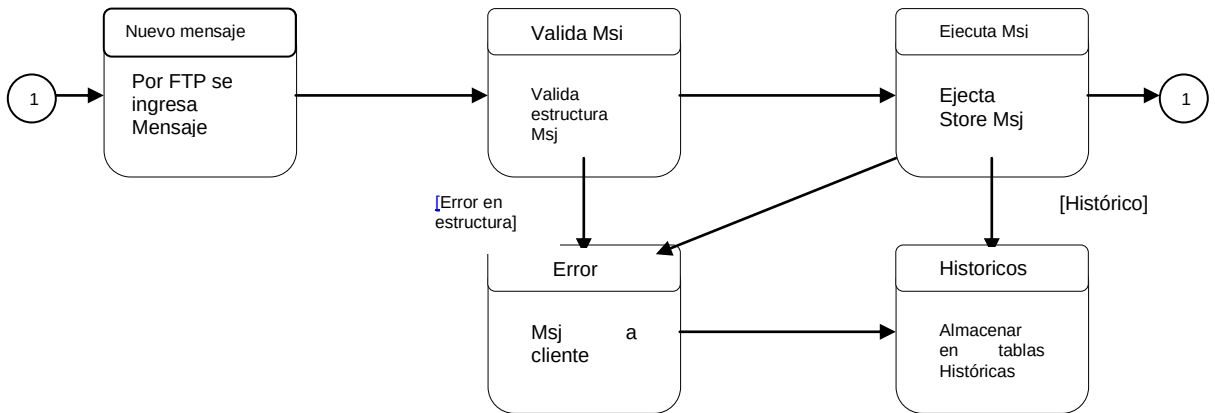
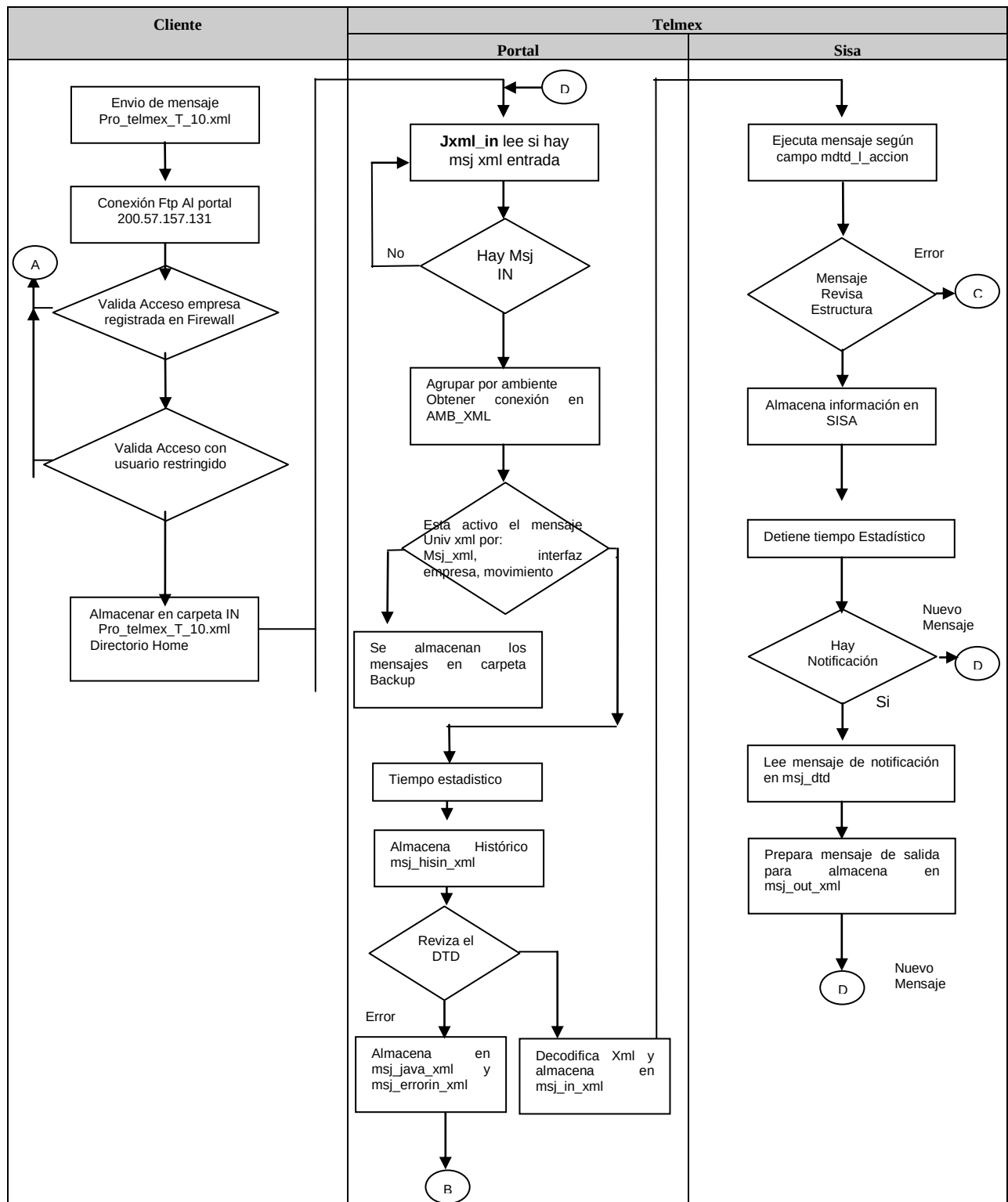


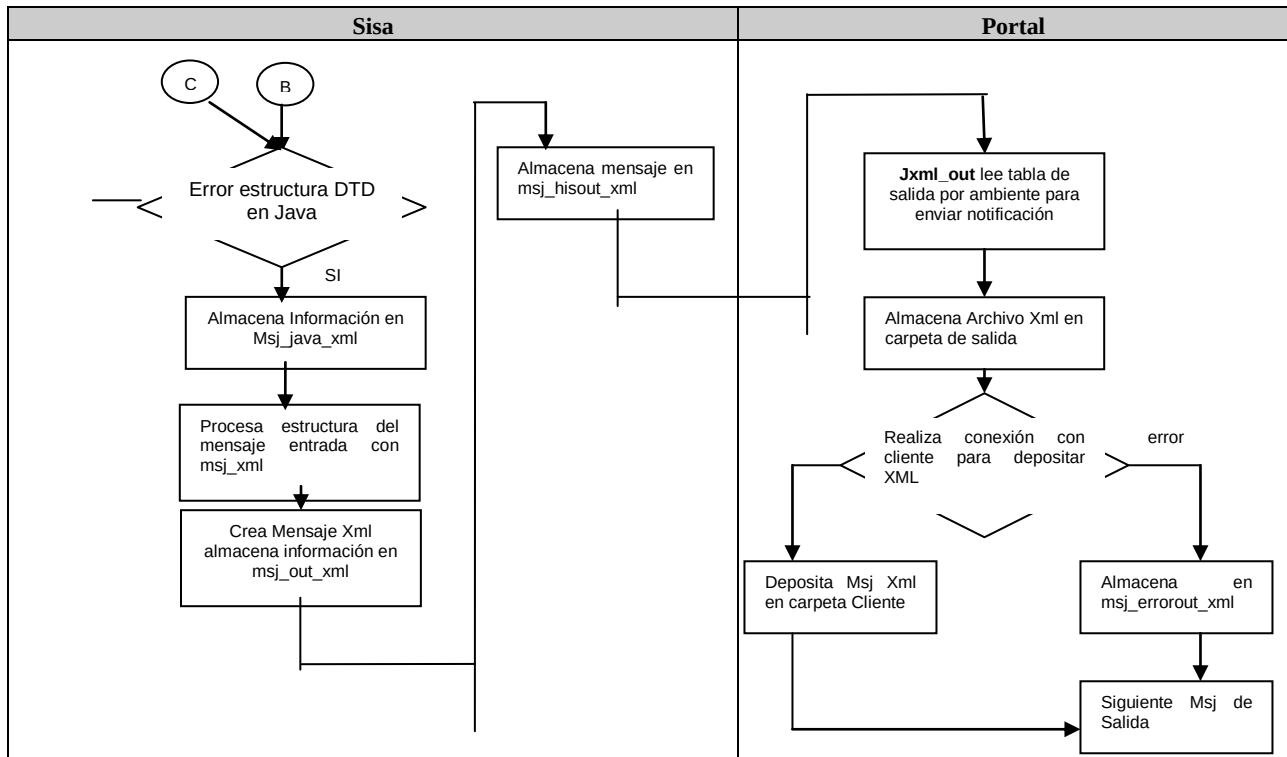
Figura H.4 Diagrama de transición de estados

Diagrama de Flujo (DGF)

Insertar Mensaje Xml y Procesamiento



Errores en Mensajes Estructura DTD y Mensaje



Seguridad Telmex

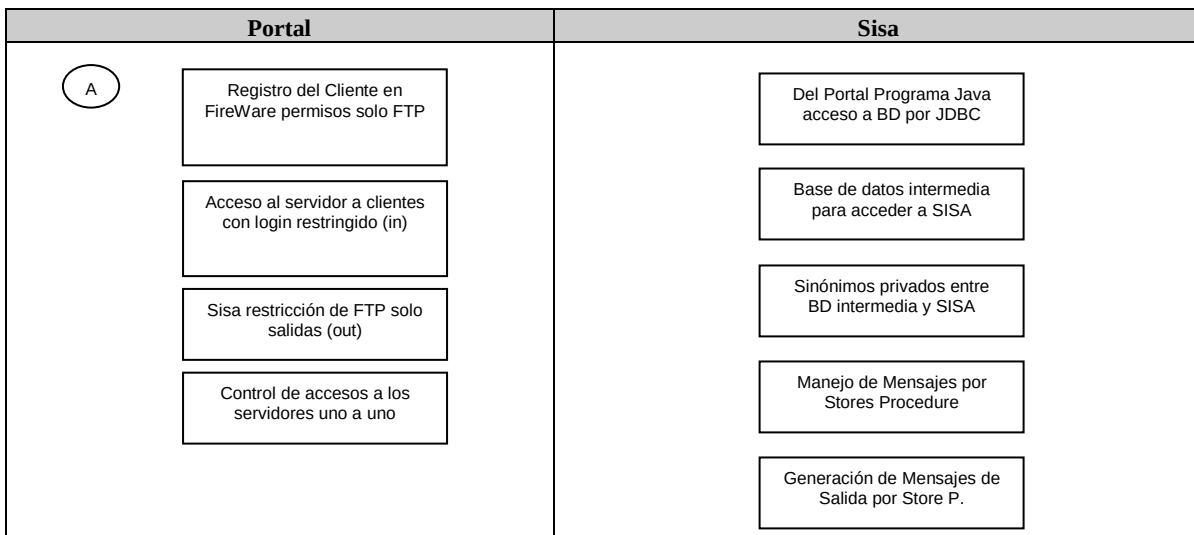


Tabla H.3 Procesos de seguridad Portal - Sisa

Comentario	Regla
Generación nombre del mensaje	RTMsjXml-Dgf-01
Registro de empresas en FireWall	RTMsjXml-Dgf-02
Descripción General Ingreso de Mensajes Xml	RTMsjXml-Dgf-03
Detección de errores al momento de leer el Mensaje Xml (DTD) por el programa JXML_IN de entrada	RTMsjXml-Dgf-04
Desempeño de las tablas del Mensaje Universal Xml para Ingreso de Queja	RTMsjXml-Dgf-05
Detección de errores al momento de ejecutar el Store Procedure indicado en el Mensaje Xml recibido	RTMsjXml-Dgf-06
Preparación de los mensajes de notificación al cliente	RTMsjXml-Dgf-07
Envío de mensajes de notificación al cliente	RTMsjXml-Dgf-08

Tabla H.4 Descripción general Análisis del mensajes xml

Acciones Particulares (Acp)

Procesos

Acción	Comentario	Regla
Detener Msj Univ Xml	Proceso que detenga definitivamente el Mensaje Universal Xml.	RTMsjXml-Acp-01

Campos

Tipo	Widget	Campo	Comentario	Regla

Modelo MSJ_UNIV_XML

Tabla	T	Objeto	Estructura		Descripción	Regla
Dir_outin_xml	A	Entidad	Doix_n	Not null number(8)	Llave Primaria Folio de los datos técnicos para la transmisión de los Mensajes Xml	RTMsjXml-Dol-01
			Eei_l_cve	Varchar2(15)	Clave Asignada de Internet para la transferencia por Empresa o Entidad	
			Doix_l_ip	Varchar2(30)	Dirección IP o DNS del Cliente	
			Doix_l_path	Varchar2(30)	Directoría a depositar los Mensjes Xml	
			Doix_l_login	Varchar2(30)	Login para filmarse y depositar los Msj Xml	
			Doix_l_pass	Varchar2(30)	Passworsd para firmarse y depositar los mensaje Xml	
			Doix_b_mult	Char(1)	Configuración para aceptar en un solo archivo varias quejas de la misma empresa.	

Tabla	T	Objeto	Estructura	Descripción	Regla
Comentarios:		Datos Técnicos del Cliente para la Transmisión de los Mensajes Xml			

Tabla	T	Objeto	Estructura		Descripción	Regla
Eim_dtd	A	Entidad	Mxml_n	varchar2(15)	Llave primaria Movimiento que se debe disparar al leer el msj xml	RTMsjXml-Dol-01
			Eei_l_cve	Varchar2(15)	Llave primaria Clave Asignada de Internet para la transferencia por Empresa o Entidad	
			Mint_n	Number(3)	Llave primaria Folio de la interfaz	
Comentarios: Almacena la relación entre los movimientos, la interfaz y la empresa o entidad telmex						

Tabla	T	Objeto	Estructura		Descripción	Regla
Emp_ete_inter net	A	Entidad	Eei_n	Not null number(8)	Folio consecutivo de las empresas o entidades registradas para el envío de Msj Xml	RTMsjXml-Dol-01
			Emp_n	Number(8)	Folio de la empresa	
			Ete_n	Number(8)	Folio de la Entidad	
			Eei_l_cve	Varchar2(15)	Clave Asignada de Internet para la transferencia por Empresa o Entidad	
			Eei_b	Char(1)	Apaga la Interfaz por empresa	
Comentarios: Identificación de la Empresa o Entidad registrados para la transmisión de los Msj Xml						

Tabla	T	Objeto	Estructura		Descripción	Regla
Estado_Notifica a	A	Entidad	Ent_n	Not null number(8)	Llave primaria Folio Consecutivo para las formas que se envía notificación	RTMsjXml-Dol-01
			Mnx_l_form	Varchar2(15)	Forma que transmitirá un mensaje de notificación	
			Mnx_l_accion	Varchar2(30)	Valida cual es el mensaje que se enviara para la notificación	
			Ent_l_desc	Varchar2(1000)	Descripción del mensaje de notificación Xml	
Comentarios: Mensajes de Notificación de salida, envia mensajes de la bitácora no deseables						

Tabla	T	Objeto	Estructura		Descripción	Regla
Msj_control_en viados	A	Entidad	Mnx_l_form	Not null varchar2(15)	Nombre de la forma donde se enviara la notificación como Xml Llave Primaria	RTMsjXml-Dol-01

Tabla	T	Objeto	Estructura		Descripción	Regla
			Mxml_l	Not null varchar2(15)	Nombre del Msj que se enviara como Msj Xml Llave Primaria	
			Mce_n_evento	Not null varchar2(15)	Folio del EMS, EFA, OIN, OGE´s, estado o campo que ejecutara la acción para el envío del mensaje Llave Primaria	
			Mce_n_enviados	Number(3)		
Comentarios: Mensajes de Notificación de salida						

Tabla	T	Objeto	Estructura		Descripción	Regla
Msj_dtd	A	Entidad	Mdtd_n	Number(8)	Llave primaria Folio consecutivo para los movimientos	RTMsjXml-Dol-01
			Mxml_l	Varchar2(15)	Nombre del movimiento o mensaje	
			Mdtd_l_io	Varchar2(4)	Indica si es una función de entrada o salida	
			Mdtd_b_notifica	Char(1)	Índica si el mensaje de entrada	
			Mdtd_l_accin	Varchar2(30)	Nombre de la función que se debe ejecutar al ingresar un mensaje, mensajes de entrada(IN)	
			Mdtd_l_aplicacion	Varchar2(15)	Indica que tipo de acción se debe realizar en la ejecución de un mensaje de entrada	
			Mdtd_l_accout	Varchar2(30)	Nombre de la función que se debe ejecutar en la notificación de un mensaje de entrada (OUT)	
			Mdtd_b	Char(1)	Indica si la interfaz esta activa o inactiva por movimiento o mensaje	
			Mdtd_l_dtd	Varchar2(4000)	DTD del movimiento	
Comentarios: Almacena los movimientos ya sea de entrada o de salida por empresa						

Tabla	T	Objeto	Estructura		Descripción	Regla
Msj_Dtd_stadistica	A	Entidad	Mds_n	Number(8)	Llave primaria Folio consecutivo de la interfaz	RTMsjXml-Dol-01
			Mxml_l	Varchar2(15)	Nombre del movimiento o mensaje	
			Eei_l_cve	Varchar2(15)	Clave Asignada de Internet para la transferencia por Empresa o Entidad	
			Mint_n	Number(3)	Folio consecutivo de la interfaz	
			Mds_f_ini	Date	Fecha que inicia la ejecución de la inserción del mensaje	
			Mds_f_fin	Date	Fecha que finaliza la ejecución de la inserción del mensaje	
Comentarios: Identificación de la Empresa o Entidad registrados para la transmisión de los Msj Xml						

Tabla	T	Objeto	Estructura		Descripción	Regla
Msj_errorin_xml	A	Entidad	Mex_n	Not null number(8)	Folio Consecutivo para almacenar los errores de los mensajes que entran por medio de Xml Llave primaria	RTMsjXml-Dol-01
			Eei_l_cve	Varchar2(15)	Clave de la empresa asignado para Internet	
			Mex_l_fcte	Varchar2(10)	Folio del sistema del cliente donde el servicio tiene problemas	
			Mex_f	Date	F/H del sistema SISA cuando se ingreso el mensaje y ocasiono un error	
			Msi_l_aplicacion	Varchar2(15)	Aplicación de donde procede el error, este puede ser ocasionado en el programa Java o en el Store procedure	
			Mxml_l	Varchar2(15)	Nombre del movimiento o mensaje	
			Mint_n	Number(3)	Folio de la interfaz	
			Msi_l_nomarch	Varchar2(32)	Nombre del archivo xml, o función que se ejecuto a la entrada	
			Mex_l_xml	Varchar2(4000)	Msj Xml con marcas donde se ocasiono el error, para posibles conciliaciones indicar al Cliente cual fue el error.	
Comentarios: Errores en los Msj de Entrada						

Tabla	T	Objeto	Estructura		Descripción	Regla
msj_errorout_xml	A	Entidad	MEO_N	NOT NULL NUMBER(8)	Folio Consecutivo Llave Primaria	RTMsjXml-Dol-01
			EMP_N	NUMBER(8)	Folio de la empresa donde se obtuvo el error de salida	
			ETE_N	NUMBER(8)	Folio de la entidad donde se obtuvo el error de salida	
			EMS_N	NUMBER(8)	Folio de la queja donde se obtuvo el error en la transferencia del Msj Xml	
			MEO_F	DATE	F/H cuando ocurrió el error en la transferencia del Msj Xml	
			MOX_L_APLICACION	VARCHAR2(15)	Aplicación donde procedió el error de salida, es decir en la aplicación o en el programa de transporte.	
			MEO_L	VARCHAR2(1000)	Descripción del error ocurrido durante la transferencia.	
Comentarios: Errores en los Msj de Salida						

Tabla	T	Objeto	Estructura		Descripción	Regla
msj_gout_xml	A	Entidad	MGOX_L	NOT NULL VARCHAR2(4)	Indica si en alguno de los ambiente se tiene mensajes de salida preparados. Llave primaria	RTMsjXml-Dol-01
Comentarios: Existencia de mensajes de salida para algún ambiente, propuesta y creada, no se esta utilizando ya que el programa Java de salida se conectara a los ambientes por default de manera manual.						

Tabla	T	Objeto	Estructura		Descripción	Regla
msj_hisin_xml	A	Entidad	MHIX_N	NOT NULL NUMBER(8)	Folio consecutivo de los mensajes ingresados por medio del Msj Xml Llave primaria	RTMsjXml-Dol-01
			Eei_l_cve	Varchar2(15)	Clave de la empresa asignado para Internet	
			MHIX_F	DATE	F/H cuando ingreso el mensaje Xml a SISA para dar de alta la queja	
			MSI_L_APLICACION	VARCHAR2(15)	Aplicación donde procede el ingreso del Msj Xml, este es para identificar si es desde Xml o Pisa	
			Mxml_l	Varchar2(15)	Campo Nuevo Nombre del movimiento o mensaje	
			Mint_n	Number(3)	Campo Nuevo Folio de la interfaz	
			MHIX_L_NOMARCH	VARCHAR2(32)	Nombre del archivo donde se encuentran las referencias que se desea dar de alta como queja	
			MHIX_L_XML	VARCHAR2(4000)	Mensaje Xml, enviado por el cliente.	
Comentarios: Histórico Msj de entrada						

Tabla	T	Objeto	Estructura		Descripción	Regla
msj_hisout_xml	A	Entidad	MHOX_N	NOT NULL NUMBER(8)	Folio Consecutivo de los mensaje emitido a un cliente o entidad Llave primaria	RTMsjXml-Dol-01
			EMP_N	NUMBER(8)	Folio de la empresa a donde se emitió el mensaje de notificación	
			ETE_N	NUMBER(8)	Folio de la entidad a donde se emitió el mensaje de notificación	
			EMS_N	NUMBER(8)	Folio de la queja donde que emitio un mensaje de salida Xml	
			MHOX_F	DATE	F/H de salida del mensaje al Cliente	
			MHOX_L_ORIGEN	VARCHAR2(30)	Desde donde se emitio el Msj, programa Java, store procedure aplicación	

Tabla	T	Objeto	Estructura		Descripción	Regla
			DOM_N	NUMBER(2)	Dominio desde donde se emitió el msj	
			MOX_L_APLICACION	VARCHAR2(15)	Aplicación a donde fue emitido el msj, por ejemplo PISA, Xml	
			MHOX_L_XML	VARCHAR2(4000)	Msj Xml de salida	
Comentarios: Histórico Msj de Salida						

Tabla	T	Objeto	Estructura		Descripción	Regla
msj_in_xml	A	Entidad	MSI_N	NOT NULL NUMBER(3)	Folio consecutivo de los msj ingresados Llave primaria	RTMsjXml-Dol-01
			MSI_L_VALOR	VARCHAR2(30)	Valor del campo DTD	
			MXML_L	VARCHAR2(15)	Mensaje que se ejecuta este se encuentra en la tabla msj_xml	
			MXML_L_CAMPD TD	VARCHAR2(30)	Nombre del campo definido para el DTD, estan definidos en la tabla msj_xml	
			MSI_L_APLICACION	VARCHAR2(15)	Que aplicación fue la que ingreso el msj para tomar las acciones correspondientes	
Comentarios: Msj de Entrada después de ser revisada la estructura del Xml por medio del programa JAVA						

Tabla	T	Objeto	Estructura		Descripción	Regla
Msj_interfaz	A	Entidad	Mint_n	Number(3)	Llave primaria Folio consecutivo de la interfaz	RTMsjXml-Dol-01
			Mint_l_abrev	Varchar2(15)	Abreviación de la interfaz	
			Mint_b	Char(1)	Boolean, indica si la interfaz esta activa o inhabilitada	
			Mint_l	Varchar2(500)	Descripción de la interfaz	
Comentarios: Almacena las interfaces que son atendidas por el Msj_xml						

Tabla	T	Objeto	Estructura		Descripción	Regla
Msj_java_xml	A	Entidad	MjX_N	NOT NULL NUMBER(8)	Folio Consecutivo sobre los errores encontrados en los mensajes de entrada en el proceso Java Llave primaria	RTMsjXml-Dol-01
			AXML_L_AMB	VARCHAR2(4)	Ambiente de donde se obtuvo el error al momento de ingresar el Msj Xml	
			EEI_L_CVE	VARCHAR2(15)	Clave de la empresa o entidad que genero el Msj Xml de entrada.	
			MjX_L_XML	VARCHAR2(4000)	Mjs Xml enviado por la empresa o entidad marcando donde se obtuvo el error al revisar el Msj Xml	
Comentarios: Errores detectados por los programas en Java cuando son ingresado por medio de un msj Xml						

Tabla	T	Objeto	Estructura		Descripción	Regla
msj_notifica_xml	A	Entidad	MNX_N	NOT NULL NUMBER(8)	Folio Consecutivo sobre los mensajes de notificación que se disparan de diferentes formas Llave primaria	RTMsjXml-Dol-01
			MNX_L_FORM	VARCHAR2(15)	Forma que dispara la notificación	
			MXML_L	VARCHAR2(15)	Msj o movimiento que dispara la notificación	
			MNX_B_ACT	CHAR(1)	Activa o Desactiva la notificación para los mensajes Xml	
			DOM_N	NUMBER(8)	Dominio de SISA que ejecuta el mensaje de notificación	
			MNX_L_CAMPO	VARCHAR2(15)	Identifica el nombre del campo que dispara la notificación	
			MNX_L_ACCION	VARCHAR2(30)	Acción que lanza la notificación, por ejemplo el estado de un servicio	
			MNX_L_CAMPOACC	VARCHAR2(30)	Nombre del campo que realizara la acción de ejecutar el msj Xml, por ejemplo el botón de aceptar	
			MNX_L_ENTIDAD	VARCHAR2(30)	Entidad donde se realiza la acción de enviar el msj xml	
			MNX_L_QUERY	VARCHAR2(3000)	Por medio de una acción recuperada desde la ejecución de un query y la comparación dispara el Msj Xml.	
Comentarios: Mensajes de Notificación al Cliente, seleccionados por el cambio de estado de la queja						

Tabla	T	Objeto	Estructura		Descripción	Regla
Msj_out_xml	A	Entidad	MOX_N	NOT NULL NUMBER(8)	Folio Msj de Salida Xml Llave primaria	RTMsjXml-Dol-01
			EMP_N	NUMBER(8)	Folio de Empresa	
			ETE_N	NUMBER(8)	Folio de Entidad	
			MOX_L_PATH	VARCHAR2(30)	Directorio a depositar los Msj Xml	
			MOX_L_LOGIN	VARCHAR2(30)	Login Oracle	
			MOX_L_PASS	VARCHAR2(30)	Password Oracle	
			MOX_L_IP	VARCHAR2(30)	Dirección IP del Cliente	
			MOX_L_FCTE	VARCHAR2(10)	Folio de queja del Cliente	
			DOIX_B_MULT	CHAR(1)	Acepta Multiples registros en un solo archivos	
			MOX_L_NOMARCH	VARCHAR2(15)	Nombre del archivo como transmitirá la empresa	
			MOX_B_ERROR	CHAR(1)	Mensaje de error	
			DOM_N	NUMBER(2)	Dominio Sisa	
			EMS_N	NUMBER(8)	Folio de la queja Sisa	

Tabla	T	Objeto	Estructura		Descripción	Regla
			MHOX_N	NUMBER(8)	Folio del Historicode Salida	
			MOX_L_APLICACION	VARCHAR2(15)	Aplicación de donde procede el mensaje	
			MOX_L_ACCION	VARCHAR2(30)	Nombre de la funcion que se debe ejecutar una ves conectado a la base de datos del sistema externo	
			MOX_L_ACCIN	VARCHAR2(30)	Nombre de la función de entrada que se debe ejecutar.	
			MOX_L_XML	VARCHAR2(4000)	Mensaje Xml (DTD)	
Comentarios: Mensajes de salida						

Tabla	T	Objeto	Estructura		Descripción	Regla
Msj_xml	A	Entidad	MXML_L	VARCHAR2(15)	Llave primaria Nombre del movimiento o mensaje.	RTMsjXml-Dol-01
			MXML_N	NUMBER(3)	Llave primaria Folio consecutivo del Msj	
			MXML_L_CAMPD TD	VARCHAR2(30)	Nombre del campo DTD definido para la empresa o entidad	
			MXML_L_CAMPB D	VARCHAR2(30)	Nombre del campo de la Base de datos	
			MXML_L_TIPO	VARCHAR2(5)	Tipo de dato	
			MXML_L_FORM	VARCHAR2(15)	Forma desde donde se envia el Msj	
			ENT_N	NUMBER(8)	Folio de la descripción	
			MXML_N_MAX	NUMBER(3)	Máximo numero de mensajes enviados	
			Mxml_t	Numer(2)	Numero de minutos máximo que la función GcamposXml debe esperar	
Comentarios	Definición de campos para los Mensajes Xml (eventos), que se ejecutan tanto de entrada como de salida, así como la definición de los DTD por cliente o entidad.					

Cuadro de actualización de Entidades (CAE).

Procesos

Comentario	Regla
Las entidades relacionadas a los mensajes de entrada MSJ_IN_XML, MSJ_HISIN_XML, todos los campos son actualizados con la información del mensaje recibido, en cado de existir error en la estructura del DTD o de los campos se almacenan los campos de las tablas MSJ_ERRORIN_XML, MSJ_JAVA_XML así como la de notificación MSJ_OUT_XML.	RTMsjXml-Dol DOL
Para el caso de los mensajes de salida se debe almacenar la información completa de las entidades MSJ_OUT_XML y MSJ_HISOUT_XML, en caso de existir un error en la transferencia del mensaje se almacenaría en las tablas MSJ_ERROROUT_XML.	

Cuadro de actualización de relaciones.

Procesos

Comentario	Regla
Para el caso del Mensaje Universal Xml las relaciones se cuidan en momento de almacenar la información, no existe una relación real en el modelo MSJ_XML ya que la información almacenada y consultada es independiente. La única relación que se puede cuidar es sobre los mensajes de notificación en las tablas ESTADO_NOTIFICA, MSJ_NOTIFICA_XML ya que se busca en estas tablas la forma y movimiento que se debe ejecutar dependiendo de la regla almacenada en la base de datos SISA.	RTMsjXml-Dol DOL

Cuadro de Funciones y/o Librerías afectadas (CFL)

Función	T	Objeto	Comentario	Regla
Eje_MsjXml.sh	A	Sh	Arranca el mensaje universal xml	RTMsjXml-Cfl-01
Gprepara_Xml	AC	Proc	Función global en UNIFACE prepara la forma que se ejecuta para el envío de mensajes de notificación	RTMsjXml-Cfl-02
Gcampos_Xml	AC	Proc	Construye el mensaje xml para enviarlo por medio del Jount_xml	RTMsjXml-Cfl-03
GRecupera_Desc	AC	Proc	Recupera descripcion	RTMsjXml-Cfl-04
Encripta_ambiente	A	Java	Función para encriptar el archivo de ambientes	RTMsjXml-Cfl-05
Jxml_in	A,C	Java	Función Java que permita leer Archivos en formato Xml que son enviados por el usuarios dentro de una carpeta IN del usuario sisaxml	RTMsjXml-Cfl-06
Jxml_out	A,C	Java	Realizar un programa en Java que permita leer Archivos en formato Xml que serán enviados por el usuarios dentro de una carpeta OUT_MsjXml	RTMsjXml-Cfl-07
St_EEXml_In	AC	StrProC	Store Procedure recupera información faltante errores de entrada JAVA	RTMsjXml-Cfl-08
FTP_RCP	C	C	Función para la transferencia de archivos ya sea por FTP o Rcp	RTMsjXml-Cfl-09

Cuadro de archivos afectados (Caf)

Archivo	Contenido	Regla
Ambiente_xml	Se almacena en el servidor un archivo plano donde se encuentran todos los ambientes activos para el Xml	RTMsjXml-Caf-01
Eje_MsjXml.txt	Indica al Mensaje Universal Xml que permanezca activo en memoria	RTMsjXml-Caf-02
TEsp_MsjXml.txt	Tiempo de espera máximo para volver a leer la tabla de entrada y salida	RTMsjXml-Caf-03
DTD_MsjXml	Estructura de los Mensajes Xml definidos por DTD	RTMsjXml-Caf-04

Reglas Técnicas

Regla	Contenido	Reglas y/o Tareas Relacionada
RTMsjXml-Ini-01	Para arrancar el Mensaje Universal Xml se debe ejecutar el shell que se encuentra en el ambiente del Portal Empresarial Eje_MsjXml.sh, hay dos modalidades. Tall Instrucción parecida al Crontab solo que esta ejecuta una vez el comando programado para alguna hora en especial. Eje_MsjXml.sh Directamente desde el prompt del sistema operativo por medio del usuario "sisaxml" El shell simula a un demonio que se duerme cada X tiempo al despertar se conecta a la instancia "xml" para revisar si existen mensajes en algún ambiente configurado en el archivo "ambientes.xml"	Ambientes_Activos.doc

Regla	Contenido	Reglas y/o Tareas Relacionada
RTMsjXml.Dgcl-01	Para el envío de mensajes Asíncronos no se requiere mas que saber la dirección del usuario destino en este caso la aplicación Mensaje Universal Xml no espera un respuesta por parte del sistema externo Sin embargo para el envío de mensajes sincronos se requiere una respuesta por parte del Mensaje Universal Xml a la aplicación externa, es decir el cliente envía un mensaje xml y espera una respuesta del mismo tipo, el programa JAVA revisa la estructura del mensaje, siendo todo correcto se toma en el primer campo DTD el cual corresponde a la ejecución del programa en la tablas de mensajes MSJ_XML en el campo MXML_L_ACCION se encuentra el nombre del programa a ejecutar. El programa en MXML_L_ACCION una de sus acciones es obtener los datos y almacenar la información en las tablas correspondientes, para el caso de los mensajes sincronos es obtener la información y almacenar el mensaje xml en la tabla de salida MSJ_OUT_XML.	Msj Ems Sisa.doc
RTMsjXml-Dgcl-02	Al recibir un mensaje Asíncrono de la aplicación externa esta no espera respuesta del mensaje universal xml, por lo que se ejecuta el mensaje se almacena la información en la aplicación SISA y se termina. En cambio para los mensajes sincronos la aplicación externa requiere una respuesta por parte del Mensaje Universal Xml posiblemente para terminar un proceso en su aplicación en este momento el Mensaje Universal Xml procesa la información	Msj Ems Sisa.doc
RTMsjXml.Dgf-01	El nombre del archivo xml esta compuesto de la siguiente manera. PRO_CVEEMP_MULT_FOLIINT PRO Ambiente donde se debe almacenar la información enviada a través del mensaje universal xml cveemp Clave de Internet asignada a la empresa para el alta de las quejas Mult Indica a la aplicación mensaje universal xml que el archivo acepto mas de una queja. FolInt Folio Internet para evitar que se pierdan las quejas enviadas por el mismo cliente el tamaño del folio es de 3 caracteres.	
RTMsjXml-Dgf-02	Para que los clientes puedan ingresar al servidor del Portal y a la inversa, para que desde el Portal se pueda llegar al servidor del cliente es necesario que en los FireWall se registren las direcciones IP de los servidores, Los puertos activos para esta aplicación son el 80 para FTP y el 21 http, el detalle de la configuración se encuentra en el archivo de ambientes	Ambientes_activos.doc

Regla	Contenido	Reglas y/o Tareas Relacionada
RtMsjXml-Dgf-03	El cliente pasa la seguridad de la zona militarizada almacenando el archivo Xml en el servidor Portal en la carpeta de entrada IN (PRO_TELMEX_T_10). El programa JAVA de entrada revisa si en la carpeta de entra (IN) se encuentra almacenado algún mensaje, recolecta los nombre en una pila de mensaje ordenándolos por ambiente. A cada ambiente se conecta con ayuda de la tabla AMBIENTES_XML que se encuentra encriptado. Una ves ordenados los mensaje por ambiente se conecta al primer ambiente dependiendo de la prioridad la prioridad que se manejo es PRODUCCIÓN, PRUEBAS, DESARROLLO. Una ves conectado al primer ambiente el programa JXML_IN se conecta al primer ambiente almacenando el mensaje xml en la tabla de Histórico de entradas MSJ_HISIN_XML, sin importar si el mensaje es correcto o tiene errores. JXML_IN por medio de la definición de campos DTD y de la tabla MSJ_XML revisa la estructura del archivo Xml, el primer campo se refiere al nombre del mensaje que se analizará. Si la estructura de campos es diferente que la almacenada en MSJ_XML se considera como error notificando al cliente de este problema por medio de un Mensaje, ya que en esta fase no es posible detectar de que cliente se trata se almacenan los datos en la tabla MSJ_JAVA_XML y ejecutando un Store Procedure (St_EEXml_In) recuperando la información faltante del Xml, este mensaje se guarda en la tabla MSJ_OUT_XML El mensaje de error se almacena en la tabla histórica de mensajes MSJ_HISOUT_XML. Si no hubiera error en la estructura del mensaje, es decir en la estructura del DTD que es analizado por el JXML_IN se procede a ejecutar el Store Procedure que lleva el nombre del mensaje solicitado, este Store revisa la estructura física de los campos, en este caso se analizan los campos a almacenar si no conciden con los descritos en la tabla MSJ_XML ocurre un error generando un mensaje al usuario almacenando la información en las tablas MSJ_ERRORIN_XML y en MSJ_OUT_XML notificando cual es el error. En caso de que no se detectara error se insertaría la información en la base de datos como nueva queja y/o realizaría un proceso para retroalimentación al cliente creando un mensaje de salida, esta información se detalla en los mensajes para levantar las quejas desde Internet. El programa Java de salida JXML_OUT revisa la tabla MSJ_OUT_XML de mensajes de salida de cada ambiente almacenado en la tabla AMBIENTES_XML activo ya que en la posición Activo = "T" serian los ambientes que revisaría mensajes para enviarlos al cliente por Internet a través del protocolo FTP Este proceso se realiza hasta que se terminan los mensajes en la carpeta de entrada como en la tabla de salida MSJ_OUT_XML, si no se encuentran mensjes el Mensaje Universal Xml se duerme por un determinado tiempo de espera, almacenado en el archivo Tesp_MsjXml.txt.	Ambientes_activos.doc RTMsjXml-Cfl-04 RTMsjXml-Cfl-05 Msj Ems Sisa.doc

Regla	Contenido	Reglas y/o Tareas Relacionada								
RTMsjXml-Dgf-04	Los mensajes están basados en la estructura de definición de tipo de dato (DTD) la cual no cambia ya que se comporta como una tabla de base de datos, en un archivos DTD pueden almacenarse varios registros. Al momento que se lee un Mensaje Xml Se revisa la estructura por si faltaran campos, la manera de analizar esto es separando los mensajes por Tag indicado por el movimiento por ejemplo. El mensaje PRO_TELMEX_T_10, es un mensaje que se desea ingresar al ambiente de producción de la empresa TELMEX, se aceptan varias quejas en el mismo archivo y por último es el mensaje numero 10 que se transmite al mensaje universal xml. Su estructura es la siguiente: <table><tr><th>PRO_TELMEX_T_10</th><th>DTD Alta_queja</th></tr><tr><td><XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> </XML></td><td><XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML></td></tr></table> En este ejemplo se nota que ala comparar el mensaje recibido con el DTD falta un campo, CVEINT en la aplicación se realiza con la tabla MSJ_XML, se detecta que falta un campo, por lo que se dispar un error, ya no se continua con el proceso de alta de queja (ALTA_QUEJA) El mensaje original se almacena en la tabla MSJ_HISIN_XML para realizar conciliaciones con el cliente además de enviar un mensaje de notificación acerca del error detectado. El mensaje de notificación por error detectado se construye con el mensaje recibido Agregando el campo faltante con una nota falta información, el mensaje se construiría con ayuda del la tabla MSJ_XML quedando de la siguiente forma. <table><tr><th>PRO_TELMEX_T_10</th><th>DTD Alta_queja</th></tr><tr><td><XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> Falta información </CveInt> </XML></td><td><XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML></td></tr></table> El nuevo DTD construido con el mensaje recibido se almacena en la tabla MSJ_JAVA_XML ya que el programa JXML_IN ejecuta el Store ST_EEXML_IN el cual obtiene de la tablas EMP_ETE_INTERNET, DIR_OUTIN_XML los datos restantes para la creación del mensaje de notificación de error en estructura DTD almacenado en la tabla MSJ_OUT_XML, como respaldo se almacena el mensaje de error de entrada en la tabla MSJ_ERRORIN_XML Terminado con este proceso se elimina el mensaje de la carpeta de entrada (IN) y se continua con el siguiente mensaje en el mismo archivo u otro de la carpeta de entrada (IN)	PRO_TELMEX_T_10	DTD Alta_queja	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> </XML>	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML>	PRO_TELMEX_T_10	DTD Alta_queja	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> Falta información </CveInt> </XML>	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML>	RTMsjXml-Caf-04 Msj Ems Sisa.doc
PRO_TELMEX_T_10	DTD Alta_queja									
<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> </XML>	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML>									
PRO_TELMEX_T_10	DTD Alta_queja									
<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> Falta información </CveInt> </XML>	<XML> <Mov> Alta_queja </Mov> <FolServ> <CveEmp> <CveInt> </XML>									

Regla	Contenido	Reglas y/o Tareas Relacionada
RTMsjXml-Dgf-05	En el caso de que no exista error al analizar el DTD el programa de entrada JXML_IN almacena los datos en la entidad MSJ_IN_XML, cada TAG del mensaje sería un registro en la tabla MSJ_IN_XML dependiendo del mensaje ejecutado almacenado en la tabla MSJ_XML. Almacenado el mensaje se ejecuta el Store Procedure del movimiento almacenado en el primer campo del DTD, todos los Store a ejecutarse se marcaran como movimiento en el DTD. De la entidad MSJ_IN_XML se obtiene el campo del DTD MXML_CAMPDTD que se relaciona con el campo MXML_L_CAMPDTD de la tabla MSJ_XML se obtiene el campo real de en el modelo SISA MXML_CAMPBD asignándole el valor del campo MSI_L_VALOR que también se encuentra en la tabla MSJ_IN_XML. Relacionados los campos de entrada (MSJ_IN_XML) con los definidos en el DTD de la tabla MSJ_XML se analiza que los campos obligatorios tengan información en la tabla MSJ_IN_XML además que correspondan con el tipo asignado en el campo MXML_L_TIPO en caso de que exista un error, se suspende la ejecución del Store Procedure ejecutando la notificación de error al cliente en los datos del mensaje.	
RTMsjXml-Dgf-06	El Store del movimiento ejecutado al detectar un error ejecuta un Store llamado Not_Err_Par (Notificación de Error en Parámetros), el cual recupera la información de la empresa almacenada en las tablas DIR_OUTIN_XML y EMP_ETE_INT la cual se almacena con el nuevo DTD almacenado en la tabla MSJ_OUT_XML estos mensajes de notificación de error con verdades (T) en el campo MOX_B_ERROR. El mensaje se almacena en la tabla de errores MSJ_ERRORIN_XML indicando en el campo MSJ_L_APLICACIÓN donde se produjo el error para este caso se almacenaría el nombre del movimiento. El mensaje original se debió almacenar en la tabla MSJ_HISIN_XML por si se requiere hacer alguna conciliación con el cliente, por último se borra el mensaje de la tabla MSJ_IN_XML para continuar con el siguiente mensaje.	
RTMsjXml-Dgf-07	La notificación al cliente esta relacionada a dos funciones generales de UNIFACE GPREPARA_XML, GCAMPOS_XML. GPREPARA_XML Al ingresar a las formas de SISA Mantenimiento se ejecuta la función validando si en la base de datos en la tabla MSJ_NOTIFICA_XML esta dada de alta con alguna acción, si es así, almacena en una variable global los parámetros necesario para construir el mensaje Xml. GCAMPOS_XML dentro de la variable global preparada en la función GPREPARA_XML desde que botón o campo se lanzará el Mensaje Xml, llenando las tablas MSJ_OUT_XML mensaje de salida, obteniendo la información de la empresa DIR_OUTIN_XML, EMP_ETE_INTERNET, se mantiene un histórico de los mensajes enviados MSJ_HISOUT_XML y si hubiera algún error se almacena en MSJ_ERROROUT_XML.	RTMsjXml-Cff-02 RTMsjXml-Cff-03
RTMsjXml-Dgf-08	El programa JXML_OUT es el encargado enviar los mensajes de notificación al exterior, este programa se conecta por medio del protocolo FTP al servidor del cliente, la información la obtiene de la tabla MSJ_OUT_XML en ella se encuentra el archivo XML, la dirección IP del cliente, el usuario y password para conectarse al servidor. Para poder enviar el archivo se almacena en el servidor de Portal, generando el Xml y asignarle un nombre para el cliente, el programa se conecta al servidor del cliente y transfiere por FTP, si por alguna razón hubiera algún error en la transferencia se almacena este en la tabla MSJ_ERROROUT_XML la descripción del error se almacena. También se almacena en la tabla MSJ_HISOUT_XML, en caso de existir algún error es posible reciclar los mensajes de manera manual.	
RTMsjXml-Acp-01	En ocasiones es necesario detener el Mensaje universal para modificar el modelo o mantenimiento del servidor. El Mensaje Universal Xml, se duerme por un determinado numero de segundos cada vez que no encuentra mensajes en la carpeta de entrada o en la tabla MSJ_IN_XML de los ambientes activos, en el momento de despertar busca el archivo Eje_MsjXml.txt lee el contenido si es " T " se mantiene en memoria, en caso de que este almacenado " F " termina su ejecución. Una forma de eliminar el Mensaje Universal Xml es por medio de la instrucción en Unix Kill con el proceso del programa JXML_IN, con esta opción puede existir la posibilidad de que pierda información de un mensaje que se este procesando.	RTMsjXml-Cfa-03

Regla	Contenido	Reglas y/o Tareas Relacionada
RTMsjXml-Dol-01	Las tablas se deben crear en una instancia nueva llamada XML para evitar que la instancia de SISA (SISAP) se fragmente. El usuario de administración de las tablas es el usuario xmlsis con el mismo password para poder consultar las tablas es necesario modificar la variable de ambiente ORACLE_SID=xml, en el usuario sisapro en la carpeta \$HOME/manto/msj_xml se encuentra un shell que permite conectarse a la instancia XML y al usuario xmlsis, ya que existe un usuario UNIX para administrar ambos ambientes (UNIX, ORACLE) En UNIFACE se creo un modelo nuevo para la administración de las tablas del mensaje universal llamado MSJ_XML, el modelo esta apuntando al manejador de base de datos UD0. Para que este ambiente pueda ser utilizado por la aplicación es necesario modificar el archivo de configuración psv.asn como se muestra (el detalle de la configuración se ve en ambientes activos.doc) <pre>[PATHS] \$data ora: ? ? \$sisahis ora: ? ? \$msj_xml ora: ? ? ; ** Make default assignments for SISA ** \$def \$data \$ora \$sisahis \$sud0 \$msj_xml</pre> Como se encuentran en diferentes instancias la configuración establecida para que la instancia Xml vea las entidades de SISAP es crear un usuario en ambas instancias y unirlos por medio de sinónimos privados.	Ambientes_activos.doc
RTMsjXml-Cfl-01	Para iniciar el Mensaje Universal Xml se realiza por medio del shell Eje_MsjXml.sh el cual se encuentra la dirección de las clases y el programa ejecutable en java (JXML_IN) el shell Eje_MsjXml.sh es un ciclo infinito que se “duerme” el tiempo establecido en el archivo TEsp_MsjXml.txt se mantiene vivo mientras el archivo Eje_MsjXml.txt tenga la bandera “T”	
RTMsjXml_Cfl_02	Aunque este activo el Mensaje Universal Xml en memoria, es decir que el programa JXML_IN este corriendo en memoria, debe ser posible deshabilitar de manera general el envío de mensajes de notificación. Para ello se debe crear un contador en la tabla SISA_CONTADORES llamado XML_ACTIVO si el valor es 1 es igual a verdadero, es decir que se encuentra activo. Si la variable es igual a 0 (cero) es falso y el mensaje universal xml esta deshabilitado para el envío de mensajes de notificación es decir no se evalúa la posibilidad de enviar mensaje al cliente. A esta ultima regla, se puede deshabilitar el Mensaje Universal Xml de manera general a los usuarios de SISA pero es posible activarlo por usuario dando privilegios agregando en el campo ACTOR.ACT_L el valor “XML:” Una vez validado la posibilidad de enviar mensajes de notificación a nivel general o particular se debe conectar a la instancia XML esta información se obtiene de la instancia BASE_DATOS.BD_N = 2 En caso de existir un error al conectarse a la base de datos del Mensaje Universal Xml se envía un mensaje al usuario para comunicarse con el administrador y corregir el problema, no es bloqueante se permite proseguir con la operación normal. Conectados a la base de datos se busca en la tabla MSJ_NOTIFICA_XML si existe algún mensaje para la forma activa; en caso de existir mensajes se almacenan la acción en variables globales, existe una restricción en la cual se puede deshabilitar un mensaje, validando el campo MSJ_NOTIFICA_XML.MNX_B_ACT = “F” para activar un mensaje el campo es igual “T” El mensaje o movimiento (MSJ_NOTIFICA_XML.MXML_I) se almacena en la variable \$\$XML_LST_MENSAJE, estos mensajes se obtienen de la tabla MSJ_NOTIFICA_XML eliminando los que ya se hayan enviado o estén por enviarse por estar en la cola de espera (MSJ_OUT_XML), para discriminar los mensajes que ya se hayan enviado al cliente se utiliza la tabla MSJ_GOUT_XML, en donde se almacena el folio de la queja MSJ_GOUT_XML.EMS_N y MSJ_GOUT_XML.MXML_I además de las veces que se haya enviado el mensaje MSJ_GOUT_XML.MNX_N_VECES. Si la forma tuviera mas de un	

Regla	Contenido	Reglas y/o Tareas Relacionada
	mensaje este se separa por comas (“ , “) En donde se ejecutara el mensaje de notificación, es decir desde que botón o campo se ejecutara el mensaje se almacena en el campo MSJ_NOTIFICA_XML.MNX_L_CAMPOACC que se respalda en la variable global \$\$XML_LST_ACCION. Si la forma tuviera mas de un campo de acción estos se separan por comas (“ , “) El último parámetro importante para el disparo de los mensajes de notificación es la regla de gestión, la condición que se debe cumplir para disparar el mensaje de notificación se almacena en el campo MSJ_NOTIFICACIÓN_XML. MNX_L_ACCION se pasa como parámetro por medio de la variable global \$\$XML_LST_ACCDISP, si la forma en los mensajes se utilizaran más de una regla para disparar los mensajes se separan por (“ , “) Las secuencias de mensajes campos y acciones deben corresponder para cada acción su campo de acción y su regla de gestión deben estar en el mismo orden Con esto preparamos a la forma que se enviara un mensaje al exterior.	
RTMsjXml_Cfl_03	Verificar que se encuentre activo el Mensaje Universal Xml para el envío de notificaciones al usuario, la variable \$\$XML_ACTIVADO = 1 indica que esta activo Se valida que la forma activa tenga mensajes activos, aunque la forma sea candidata para el envío de mensajes de notificación puede que ya se hayan enviado todos, es decir que los mensajes o movimientos de MSJ_NOTIFICA_XML se encuentren en la tabla MSJ_GOUT_XML, en este caso ya no se envía un nuevo mensaje de notificación. Si la forma aun tiene que enviar mensaje de notificación por que la variable \$\$XML_LST_MENSAJE contiene información, se obtiene el mensaje unitario, con su campo de acción, desde donde se dispara el mensaje como la regla funcional que debe cumplir Con el mensaje unitario se recuperan los campos del Mensaje Xml de la tabla MSJ_XML ordenando los campos por movimiento y relativo (los campos que conforman el mensaje) MSJ_XML.MXML_N y MSJ_XML.MXML_N_REL. Una vez ordenado el mensaje (DTD) se construye el Xml para ello se realiza un ciclo mientras existan campos del mensaje MSJ_XML.MXML_N_REL, el mensaje se construye como sigue. ✚ Se construye el encabezado del Xml ✚ Nombre campo DTD, MSJ_XML.MXML_L_CAMPDTD ✚ Si existe valor en el campo MSJ_XML.MXML_L_CAMPBD de este campo se obtiene el valor recuperado en la forma por medio de una in dirección. ✚ Si el MSJ_XML.MXML_L_CAMPBD es nulo, revisar que el campo MSJ_XML.ENT_N es diferente de nulo, en este campo se almacena el folio de las descripciones del avance que se desea enviar al cliente, se envía una descripción general del estado que se reporta en la tabla de notificaciones (MSJ_NOTIFICA_XML) ✚ Por ultimo se concatena la información con los TAG del mensaje Xml La descripción del archivo Xml se describe adelante, Un ejemplo del Mensaje Xml obtenido es el siguiente: <pre><?xml version="1.0" ?> <EMS> <EXP_MAN_SERV.SISA> <EMS_N>810</EMS_N> <EST_N>EMA</EST_N> </EXP_MAN_SERV.SISA> </EMS></pre>	RTMsjXml-Caf-04
RTMsjXml-Cfl-04	En el l catalogo de estados de notificación (ESTADO_NOTIFICA) se almacenan las descripciones de los cambios de estado que se enviara en un mensaje de notificación para agregarlo a un mensaje en la tabla MSJ_XML el campo MXML_L_CAMPBD es nulo y el campo ENT_N tiene el folio del estado de notificación que se debe recuperar, la descripción recuperada se concatena con el campo MSJ_XML.MXML_L_CAMPDTD. Esta descripción de cambio de estado es la enviada al cliente para evitar enviar las observaciones del Expediente de falla u otro tipo de estado, la descripción se ve de la siguiente manera. <pre><ASIG_TEC>Se asigna técnico de falla</ASIG_TEC ></pre>	RTMsjXml-Cfl-03

Regla	Contenido	Reglas y/o Tareas Relacionada					
RTMsjXml-Cfl-05	El archivo de ambientes utiliza un algoritmo para encriptar la información de los servidores donde se va almacenar las quejas enviadas por Internet. el algoritmo trabaja de la siguiente manera. Algoritmo de encriptamiento primera parte. - La palabra que se desee encriptar se descompone en caracteres, se obtiene su ASCII - Con el primer carácter se generan dos números aleatorios y se convierten en Hexadecimal. - A cada carácter se le suma la posición - Mas el numero aleatorio que se genero. - Se le suma el numero obtenido - Se convierte en hexadecimal Algoritmo para unir las palabra encriptada. - Se generan dos numeros aleatorios para rellenar, el resultado es la primera posición de la palabra total. El relleno puede tener máximo nueve caracteres de lo contrario la palabra crecería mucho y por lo tanto el archivo encriptado. <table><tr><td>2° numero Aleatorio</td><td>Relleno</td><td>Primer carácter de la palabra en Hexadecimal</td><td>Resto de la palabra en hexadecimal</td><td>1° numero aleatorio</td></tr></table>	2° numero Aleatorio	Relleno	Primer carácter de la palabra en Hexadecimal	Resto de la palabra en hexadecimal	1° numero aleatorio	
2° numero Aleatorio	Relleno	Primer carácter de la palabra en Hexadecimal	Resto de la palabra en hexadecimal	1° numero aleatorio			
RTMsjXml_Cfl-06	Este programa trabajara en Background, de manera repetitiva en un intervalo de tiempo Asignado en le archivo TEsp_MsjXml.txt. El programa lee todos los nombres de archivo almacenados en el directorio de entrada (IN) se almacena en un vector ordenándolos por ambiente, en primer lugar PRODUCCIÓN, PRUEBAS, DESARROLLO, dependiendo también del ambiente activo. Una ves ordenados los mensajes el programa se conecta al ambiente por medio del archivo AMBIENTES_XML el cual se encuentra encriptado. Conectado al ambiente se procede a leer los mensajes almacenados en el vector de mensajes del programa JXML_IN, se lee el primer archivo, este se analiza comparando el numero de campos contra los almacenados en la entidad MSJ_XML (por medio del movimiento que es el primer campo del DTD), si el numero de campos es igual al del mensaje continua con la ejecución del Procedimiento almacenado. El procedimiento almacenado que es ejecutado por el programa JXML_IN analiza los campos del DTD contra los almacenados en la tabla MSJ_IN_XML, que almaceno antes el JXML_IN, el Store Procedure compara los campos de MSJ_IN_XML contra los de MSJ_XML para aplicar las reglas correspondientes al mensaje. En algunos casos es posible enviar mensajes de notificación a partir de los mensajes recibidos. Una ves procesado el mensaje sin errores, se elimina el mensaje de la tabla MSJ_IN_XML y del vector almacenado en el programa JXML_IN y de la carpeta IN que se encuentra en el portal. Se comienza este mismo proceso para el siguiente mensaje. Si no existen mensajes en la cola de entrada carpeta IN, MSJ_IN_XML, vector de mensajes en JXML_IN, antes de disparar la señal para dormirse se espera un minuto activo, después se lanza la señal para esperar el tiempo marcado en TEsp_MsjXml.txt.	RTMsjXml_Cfl-05 RTMsjXml-Cfa-03 Msj Ems Sisa.doc Msj Pisa - Sisa Manto.doc					
RTMsjXml-Cfl-07	Las pantallas de SISA pueden generar mensajes de notificación extrayendo la información de la pantalla consultada, dependiendo de las reglas de gestión almacenadas en la tabla de mensajes MSJ_XML y en la tabla MSJ_NOTIFICA_XML. Los mensajes ya procesados se almacenan en la tabla MSJ_OUT_XML, el programa de salida JXML_OUT. Estos mensajes son enviados por orden primero producción, pruebas y por último Desarrollo, dependiendo de los ambientes activos en la tabla AMBIENTES_XML ya que se conecta a todos los ambientes activos. Para evitar el problema de que se sigan a acumulando los mensajes en la tabla de salida MSJ_OUT_XML para procesar los mensajes de los demás ambientes se crea un vector con los mensajes que en ese momento se encuentren almacenados, al momento de iniciar el envío se inserta un nuevo mensaje este se queda en la cola (MSJ_OUT_XML) hasta la siguiente vuelta. Este programa crea temporalmente el archivo XML en el directorio Out del portal, creando el nombre del archivo, este se construye de la siguiente manera.	RTMsjXml-Cfl-09					

Regla	Contenido	Reglas y/o Tareas Relacionada
	✚ Siglas del ambiente conectado en ese momento por ejemplo PRO (producción) ✚ Clave Internet de la empresa MSJ_OUT_XML.EMP_N o MSJ_OUT_XML.ETE_N DIR_EMP_ETE.DOIX_L_CVEEMP, DIR_EMP_ETE.DOIX_L_CVEETE ✚ Folio consecutivo el cual se obtiene de la tabla SISA_CONTADORES.CONTADOR ='MSJ_XML' incrementado en uno. Se almacena el mensaje a enviar en la tabla histórica MSJ_HISOUT_XML Una vez construido el mensaje se trata de hacer conexión al servidor del cliente por medio del protocolo FTP, los datos para hacer la conexión son MSJ_OUT_XML.MOX_L_PATH, MSJ_OUT_XML.MOX_L_LOGIN, MSJ_OUT_XML.MOX_L_PASS, MSJ_OUT_XML.MOX_L_IP Se realizan 3 intentos de conexión al no poder enviar los mensajes se almacenan en la tabla de errores (MSJ_ERROROUT_XML) con la descripción del por que no se pudo entregar Si al momento de estar transmitiendo el mensaje ocurriera un error, este es reportado por la función FTP, se almacena en la tabla de errores (MSJ_ERROROUT_XML) con la descripción del por que no se pudo entregar Se elimina el mensaje de la tabla MSJ_OUT_XML de la carpeta OUT del Portal y del vector del JXML_OUT. Una vez procesado el mensaje se elimina de la tabla de salida MSJ_OUT_XML, así como de la carpeta OUT del Portal. Una vez que se terminó de leer todas las colas de salida el programa JXML_OUT se debe esperar un minuto antes de lanzar la señal para dormirse el tiempo establecido en el archivo de espera TEsp_MsjXml.txt	
RTMsjXml-Cñ-08	Al analizar el Mensaje Xml puede que se encuentre un error en el número de parámetros del mensaje, este error se puede presentar más cuando se inicie la interfaz con un nuevo cliente. El programa JXML_IN compara el número de parámetros entre el Mensaje Xml y el DTD almacenado en la tabla MSJ_XML. Si hubiera algún error en el número de parámetros se envía un mensaje de notificación de error. Como en este momento el programa JXML_IN no sabe de quien es el mensaje se almacena el Xml en la tabla MSJ_JAVA_XML al Mensaje se agrega el campo faltante indicando en la descripción "Error de número de parámetros" También se almacena el mensaje en la tabla MSJ_HISIN_XML y en la tabla de errores MSJ_ERRORIN_XML para posibles conciliaciones Se ejecuta el procedimiento almacenado (St_EEXml_In) para recuperar la información faltante como es la dirección IP, login, password, Path, etc. Esto por medio del campo EEI_L_CVE, recuperada la información faltante se almacena en la tabla de salida MSJ_OUT_XML, listo el mensaje en la cola de salida se elimina de la tabla MSJ_JAVA_XML	
RTMsjXml-Cñ-09	La función FTP_RCP es la encargada de realizar la conexión con el servidor del cliente, además de obtener los errores que se presente al momento de transferir el mensaje XML, Alguno de ellos son: ✚ No es posible establecer la conexión con el servidor ✚ Se consigue la conexión con el servidor pero no es posible transferir el archivo por falta de derechos ✚ Se consigue la conexión con el servidor pero no es posible transferir el archivo por falta espacio en disco.	

Regla	Contenido	Reglas y/o Tareas Relacionada																																	
RTMsjXml-Caf-01	Este archivo es leído por el programa JAVA de entrada, en el se encuentran almacenados los ambientes activos para el Mensaje Univ Xml, la información esta encriptada. Esta archivo contiene la información almacenada de la siguiente manera Axml_l_amb Inicales del ambiente, estas son por ejemplo Prb pruebas, Des Desarrollo, Pro producción Axml_l_dns Dirección IP o DNS del equipo donde se almacenarala queja enviada por medio del Xml mas el puerto y la instancia donde se conectara a la base de datos Por ejemplo: sisades:1521:sisau7 Axml_log_ora Login de Oracle Axml_pas_ora Password de Oracle																																		
RTMsjXml-Caf-02	Indica por medio de valor boléano si el mensaje debe permanecer activo en memoria o eliminarse, si en el archivo se encuentra almacenado "I" se mantiene activo en memoria recibiendo o enviando mensajes, si se almacena "F", se baja de memoria, este archivo se lee después de despertar, es decir cuando se venció el tiempo de espera e iniciar nuevamente su actividad se lee el archivo.	RTMsjXml-Ini-01																																	
RTMsjXml-Cfa-03	Se almacena en tiempo máximo de espera antes de comenzar a leer la cola de entrada o la de salida, el valor se almacena en minutos, por el momento el tiempo por default para la espera de mensajes es de 5 minutos. Los programas JXML_IN y JXML_OUT no se duermen mientras se encuentren mensajes tanto en la cola de entrada como de salida. El tiempo como esta almacenado en el archivo TEsp_MsjXml.txt es variable																																		
RTMsjXml-Caf-04	El Mensaje Xml se puede enviar toda clase de datos, imágenes video, etc. Para no perder la consistencia de la información se basa en la definición de tipo de dato propuesto por la J2EE. En la definición del tipo de dato (DTD) en nuestro caso DTD se encuentra en la tabla MSJ_XML de cada mensaje. <table border="1"> <thead> <tr> <th>DTD</th><th>XML</th><th>Descripción</th></tr> </thead> <tbody> <tr> <td></td><td><?xml version="1.0"?></td><td>Encabezado indica que es un archivo tipo XML</td></tr> <tr> <td></td><td><EMS></td><td>Movimiento</td></tr> <tr> <td><!ELEMENT EMS (EXP_MAN_SERV.SISA*)></td><td><EXP_MAN_SERV.SISA></td><td>Inicia estructura de campos del XML</td></tr> <tr> <td><!ELEMENT EXP_MAN_SERV.SISA (EMS_N, CAR_N, EST_N, EFA_N)></td><td></td><td>Elementos definidos en el DTD, no se presentan en el XML</td></tr> <tr> <td><!ELEMENT EMS_N (#PCDATA)></td><td><EMS_N>810</EMS_N></td><td>1° elemento del XML</td></tr> <tr> <td><!ELEMENT CAR_N (#PCDATA)></td><td><CAR_N></CAR_N></td><td>2° elemento del XML</td></tr> <tr> <td><!ELEMENT EST_N (#PCDATA)></td><td><EST_N>EMA</EST_N></td><td>3° elemento del XML</td></tr> <tr> <td><!ELEMENT EFA_N (#PCDATA)></td><td><EFA_N>810</EFA_N></td><td>4° elemento del XML</td></tr> <tr> <td></td><td></EXP_MAN_SERV.SISA></td><td>Cierra estructura de campos Xml</td></tr> <tr> <td></td><td></EMS></td><td>Cierra estructura en nuestro caso es el movimiento</td></tr> </tbody> </table>	DTD	XML	Descripción		<?xml version="1.0"?>	Encabezado indica que es un archivo tipo XML		<EMS>	Movimiento	<!ELEMENT EMS (EXP_MAN_SERV.SISA*)>	<EXP_MAN_SERV.SISA>	Inicia estructura de campos del XML	<!ELEMENT EXP_MAN_SERV.SISA (EMS_N, CAR_N, EST_N, EFA_N)>		Elementos definidos en el DTD, no se presentan en el XML	<!ELEMENT EMS_N (#PCDATA)>	<EMS_N>810</EMS_N>	1° elemento del XML	<!ELEMENT CAR_N (#PCDATA)>	<CAR_N></CAR_N>	2° elemento del XML	<!ELEMENT EST_N (#PCDATA)>	<EST_N>EMA</EST_N>	3° elemento del XML	<!ELEMENT EFA_N (#PCDATA)>	<EFA_N>810</EFA_N>	4° elemento del XML		</EXP_MAN_SERV.SISA>	Cierra estructura de campos Xml		</EMS>	Cierra estructura en nuestro caso es el movimiento	
DTD	XML	Descripción																																	
	<?xml version="1.0"?>	Encabezado indica que es un archivo tipo XML																																	
	<EMS>	Movimiento																																	
<!ELEMENT EMS (EXP_MAN_SERV.SISA*)>	<EXP_MAN_SERV.SISA>	Inicia estructura de campos del XML																																	
<!ELEMENT EXP_MAN_SERV.SISA (EMS_N, CAR_N, EST_N, EFA_N)>		Elementos definidos en el DTD, no se presentan en el XML																																	
<!ELEMENT EMS_N (#PCDATA)>	<EMS_N>810</EMS_N>	1° elemento del XML																																	
<!ELEMENT CAR_N (#PCDATA)>	<CAR_N></CAR_N>	2° elemento del XML																																	
<!ELEMENT EST_N (#PCDATA)>	<EST_N>EMA</EST_N>	3° elemento del XML																																	
<!ELEMENT EFA_N (#PCDATA)>	<EFA_N>810</EFA_N>	4° elemento del XML																																	
	</EXP_MAN_SERV.SISA>	Cierra estructura de campos Xml																																	
	</EMS>	Cierra estructura en nuestro caso es el movimiento																																	

Sincronización de actualización de Tareas.

Tareas	Contenido	Reglas relacionadas
AQI	Alta de quejas vía Internet	
IPSM	Alta de quejas a Pisa para servicios básicos Interfaz Pisa – Sisa Manto	
EFMX	Esquemas del funcionamiento del Msj Xml	

Apéndice I Herramientas Utilizadas

Diagrama interconexión entre el Cliente, SisaXml, Sisa

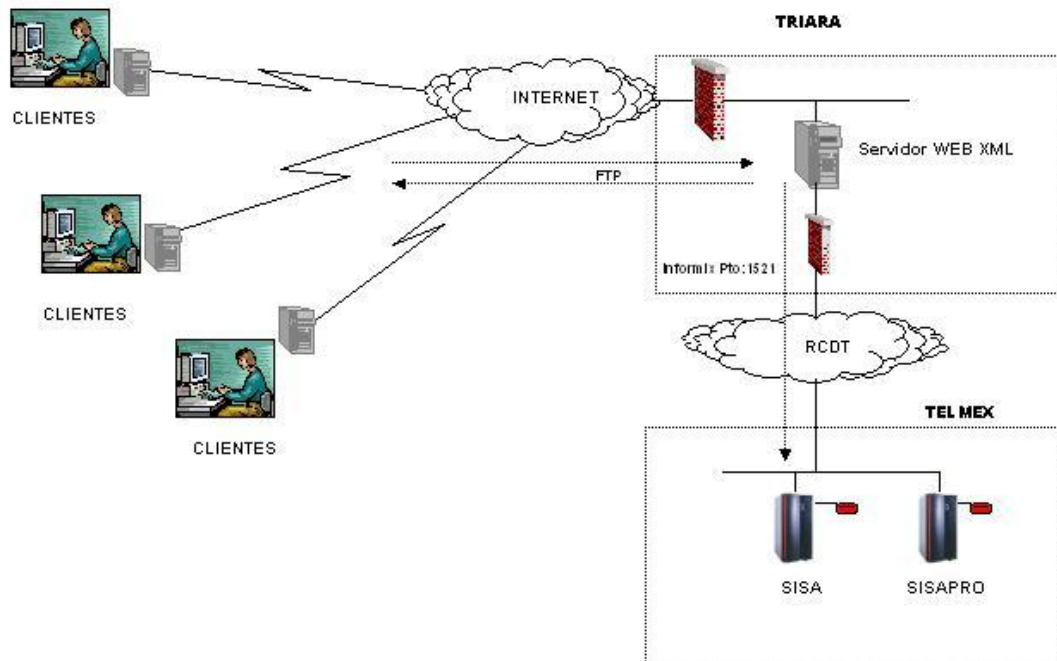


Figura AI 1 Diagrama interconexión entre el Cliente, SisaXml

Servidor Telmex (Sisa)

Sistema Operativo

Herramienta de desarrollo.

Cliente

Servidor

Base de datos

Configuración del Arreglo de Discos VA 7400.

En un esfuerzo de mejorar la velocidad de almacenamiento y la seguridad de su información Telmex esta reemplazando el Storage Manager FC60 por uno de mejor tecnología, para lo cual se eligió un arreglo de HP modelo VA 7400.

La siguiente grafica muestra las características del nuevo arreglo de discos donde estará almacenada la Base de Datos SISAP.

VA 7400

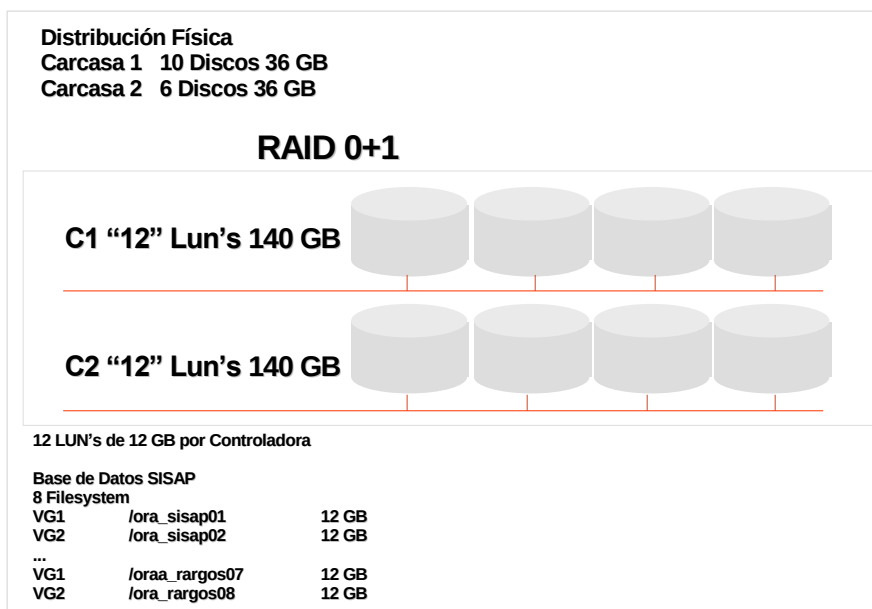


Figura AI 2 Distribución de Discos y configuración del servidor Sisa

En resumen, el nuevo arreglo de discos contiene las siguientes características:

- ✚ 12 LUN's de 12 GB por controladora.
- ✚ Espacio real disponible para almacenar datos de 280GB.
- ✚ Dos volume groups de 60GB cada uno, donde cada volume group esta formado por 5 disco y se tiene acceso a estos por ambas controladoras.
- ✚ El espacio inicial en disco para almacenar la BD SISAP es de 96GB, distribuidos en 8 filesystem de 12GB cada uno.
- ✚ El nombre del filesystem incluirá el nombre de la Base de Datos y un numero consecutivo de dos dígitos. El numero consecutivo determina el Volume Group al que pertenece el filesystem.
- ✚ Distribución de los filesystem en ambos volume groups, quedando por ahora cuatro filesystem por volume group, con la finalidad de balancear operaciones de E/S.

Distribución de filesystem Base de Datos SISAP.

La Base de Datos SISAP cuenta con 8 filesystem, el estándar para nombrar los filesystem es el siguiente: El nombre inicia con el prefijo "ora_", seguido por el nombre de la BD y como subfijo un numero consecutivo de dos dígitos. Donde el número impar será usado para filesystem en el primer volume group y el numero par para filesystem del volume group dos. El propósito de identificar a que volume group al que pertenece cada filesystem es porque el espacio reservado para los tablespace con mayor actividad se esta tomando de filesystem de diferente volume group.

Se utilizo el prefijo "arc_" para identificar el espacio reservado para los redo log files archivados antes de ser almacenados a cinta. El estándar para nombrar filesystem para almacenar los redo log files es: prefijo "arc_" seguido por en nombre de la BD y un numero consecutivo. Se crearon dos y consideramos que no será necesario crear más.

Recomendamos utilizar el estándar ya se para agregar mas espacio a la BD o para crear nuevas BD's.

Los filesystem de la BD SISAP son:

Filesystem's del volume group /dev/oradata_sisap01

✚ /ora_sisap01

✚ /ora_sisap03
✚ /ora_sisap05
✚ /ora_sisap07
✚ /arc_sisap01

Filesystem's del volume group /dev/oradata_sisap02

✚ /ora_sisap02
✚ /ora_sisap04
✚ /ora_sisap06
✚ /ora_sisap08
✚ /arc_sisap02

Características de la Base de Datos SISA.

A continuación se detalla las características de las estructuras físicas de la base de Datos SISAP, tales como parámetros de configuración de la instancia, segmentos de rollback y las estructuras físicas de la base de datos. Las siguientes tablas tienen como propósito dar una visión global del tamaño y el uso de los espacios reservados para almacenar información de SISA.

Características del Servidor y del Sistema Operativo	
Nombre del Servidor / Dirección IP	Msshba01 IP 13.44.2.52.
Plataforma/Equipo	HP 9000/800 10 CPU's@555Mhz.
Sistema Operativo/versión	HP-UX B.11.00
Memoria física disponible	12 GB

Tabla AI 1 Característica de base de datos Sisa

Servidor Web Xml (SisaXml)

General

En el servidor mwhssn01 con dirección física 10.106.4.129, dirección de Internet 148.223.168.45 esta corriendo el ambiente de desarrollo y pruebas para la aplicación SISA_XML y SISA_WAP

Se cuentan con tres cuentas a nivel sistema operativo

Ambiente de desarrollo

Cuenta

Login	Password
Jplacido	456789

Directorio de Archivos					
drwxr-xr-x	2	jplacido	cteiedf	3584	Jun 5 13:40 backup
drwxrwxrwx	2	jplacido	cteiedf	512	Oct 13 2003 docs
drwxr-xr-x	8	jplacido	cteiedf	1024	Aug 14 17:32 fuentes
drwxr-xr-x	2	jplacido	cteiedf	2560	Jul 18 10:07 in
drwxr-xr-x	2	jplacido	cteiedf	512	Jan 7 2005 jar
drwxr-xr-x	2	jplacido	cteiedf	512	Aug 6 21:42 log
drwxr-xr-x	6	jplacido	cteiedf	512	Mar 15 18:46 mrsalvad
drwxr-xr-x	2	jplacido	cteiedf	512	Aug 14 23:16 out
drwxr-xr-x	8	jplacido	cteiedf	512	Oct 17 2006 pruebas
drwxr-xr-x	2	jplacido	cteiedf	512	Jun 28 17:39 tmp

Nota: La estructura de archivos con sus subdirectorios

Tabla AI 2 Distribución de carpetas Servidor SisaXml

Ambiente de pruebas

Cuenta

Login	Password	Path	Privilegios
Sisa	741369	/export/home/sisa Nota: con privilegis para modificar los archivos deL usuario sisaxml	Posibilidad de modificar los archivos de sisaxml posibilidad de ejecutar crontab
Sisaxml	852456 Nota: actualmente esta cuenta tiene como password 26593, favor de crear con la sugerida.	/export/home/sisaxml/in Nota: Con Shell restringido	posibilidad de ejecutar crontab

Directorio de Archivos				
-rw-r--r--	1	sisa	cteiedf	106 Mar 26 2005 @LongLink
drwxr-xr-x	2	sisa	cteiedf	1024 Apr 26 12:04 backup
drwxr-xr-x	2	sisa	cteiedf	512 Jul 28 2005 docs
drwxr-xr-x	6	sisa	cteiedf	1024 Nov 7 2005 fuentes
drwxr-xr-x	7	sisa	cteiedf	512 Jul 11 19:15 htdocs
drwxrwxr-x	3	sisa	cteiedf	1024 Aug 6 17:08 in
drwxr-xr-x	11	sisa	cteiedf	512 Feb 8 2007 jakarta-tomcat-5.5.9
drwxr-xr-x	2	sisa	cteiedf	512 Nov 7 2005 jar
drwxr-xr-x	2	sisa	cteiedf	1024 Jul 13 10:23 librerias
drwxr-xr-x	2	sisa	cteiedf	512 Nov 3 2006 log
drwxr-xr-x	2	sisa	cteiedf	512 Jan 10 2007 out
drwxr-xr-x	9	sisa	cteiedf	512 Apr 27 2006 pruebas
drwxr-xr-x	2	sisa	cteiedf	512 Nov 14 2003 tmp

Nota:La estructura de archivos con sus subdirectorios

Tabla AI 3 Distribución del servidor Sisa

Software instalado

Apache 2.2.4
jakarta-tomcat 5.5.9
java version 1.4.2_06
Utilización de JDBC de JAVA comunicación con Base de datos Oracle

Configuración de caracteres en español

/etc/TIMEZONE

TZ=US/Central
CMASK=022
LC_COLLATE=en_US.ISO8859-15
LC_CTYPE=en_US.ISO8859-15
LC_MESSAGES=en_US.ISO8859-15
LC_MONETARY=en_US.ISO8859-15
LC_NUMERIC=en_US.ISO8859-15
LC_TIME=en_US.ISO8859-15
LANG=en_US.ISO8859-15

Puertos activos

Puerto	Descripción
21	Comunicación FTP
1521	Comunicación con servidor de pruebas SISA y Base de Datos
9083	Ejecución Bean con PISA

80	Comunicación Apache, ejecución de HTTP
8021	Relacion entre Apache y TomCat
8080	TomCat
514	automatización de porcesos para la aplicación de SISA XML
443	Comunicación con servidor Tralix
25	Configuración de mail, relación con el servidor de correo tmxmail22.intranet.telmex.com
2025	Comunicación LATAM

Tabla AI 4 Puertos activos en el servidor SisaXml

Comunicación entre servidores

Comunicación bi-direccional		Puerto
10.106.4.129	10.192.3.77	1521
10.192.2.136	10.106.4.129	9083
192.100.183.168	10.106.4.129	21
10.184.129.106	10.106.4.129	21
10.192.2.137	10.106.4.129	9083
10.106.4.129	www.servicios-ie.telcel.com	Mensajes escritos a celular
10.106.4.129	www.telbip.com.mx	Ejecución de Servicio HTTP
200.57.159.161 200.57.159.162 200.57.159.163 200.57.159.164	10.106.4.129	80,443 Proyecto TRALIX, nuevas direcciones
10.192.3.77	10.106.4.129	514 Ejecución de Rsh del servidor de SISA a portal desarrollo
10.106.4.129	10.106.4.129	Para Proyecto Latam se esta utilizando puerto 2025
10.106.4.129	SRVSAR4	

Tabla AI Comunicación entre servidores por puerto

Configuración de Apache y TomCat para aplicación SISA_WAP

Configuración .profile

```
#
# @(#)local.profile 1.4 93/09/15 SMI
#
PATH=/usr/j2se/bin:/usr/bin:/usr/sbin:.
export PATH
SAFT_CP=.:$HOME/ClienteWS1.4/Librerias/xerces.jar:$HOME/ClienteWS1.4/Librerias/b
ootstrap.jar:$HOME/ClienteWS1.4/Librerias/commons-discovery.jar:$HOME/ClienteWS1
.4/Librerias/commons-logging-api.jar:$HOME/ClienteWS1.4/Librerias/ffdc.jar:$HOME
/ClienteWS1.4/Librerias/j2ee.jar:$HOME/ClienteWS1.4/Librerias/ras.jar:$HOME/Clie
nteWS1.4/Librerias/webservices.jar:$HOME/ClienteWS1.4/Librerias/wsdl4j.jar:$HOME
/ClienteWS1.4/Librerias/wsexception.jar:$HOME/ClienteWS1.4/Librerias/xml.jar
export SAFT_CP
#
# @(#)local.profile 1.4 93/09/15 SMI
#
PATH=/usr/j2se/bin:/usr/bin:/usr/sbin:.
export PATH
SAFT_CP=.:$HOME/ClienteWS1.4/Librerias/xerces.jar:$HOME/ClienteWS1.4/Librerias/b
ootstrap.jar:$HOME/ClienteWS1.4/Librerias/commons-discovery.jar:$HOME/ClienteWS1
.4/Librerias/commons-logging-api.jar:$HOME/ClienteWS1.4/Librerias/ffdc.jar:$HOME
/ClienteWS1.4/Librerias/j2ee.jar:$HOME/ClienteWS1.4/Librerias/ras.jar:$HOME/Clie
```

```
nteWS1.4/Librerias/webservices.jar:$HOME/ClienteWS1.4/Librerias/wsdl4j.jar:$HOME
/ClienteWS1.4/Librerias/wsexception.jar:$HOME/ClienteWS1.4/Librerias/xml.jar
export SAFT_CP
export SISA
CLASS=/export/home/sisaxml/jakarta-tomcat-5.5.9/webapps/manto/WEB-INF/classes
export CLASS
LOGS=/export/home/sisaxml/jakarta-tomcat-5.5.9/logs
export LOGS
BIN=/export/home/sisaxml/jakarta-tomcat-5.5.9/bin
export BIN

stty erase "^H" kill "^U" intr "^C" eof "^D"
```

Apéndice J Historia del proyecto Mensaje Universal Xml.

Fecha	Observaciones
03/Septiembre/2002	Presentación propuesta inicial del Mensaje Univesal Xml
04/Noviembre/2002	Propuesta de desarrollo Intercambio de mensajes por Internet con el formato Xml
21/Septiembre/2003	Primera versión, Sistema piloto Pisa Quejas, no se implanto
18/Noviembre/2003	Desarrollo y puesta en producción SisaXml
30/Diciembre/2003	Entrega de aplicación a soporte usuarios, proceso ejecutados en servidor intermedio entre el Firewall y SisaXml
20/Abril/2004	Descripción de ambientes activos para el Mensaje Universal Xml. Actualizaciones afectaciones con la interfaz Pisa – Sisa Manto. Actualización a mensajes para el alta de quejas por Internet.
12/Agosto/2004	Intercambio de mensajes por Intranet, es trasparente ya que se maneja en la configuración de SisaXml la IP, URL del destinatario. Actualización de Históricos y seguimiento de mensajes Primera versión interfaz Sisa – Pisa Quejas
30/Agosto/2004	Análisis de requerimientos Clarify
28/Octubre/2004	Recepción de mensajes Xml para la interfaz Clarify
08/Noviembre/2004	Proceso de inserción de un mensaje Xml, liberación del Sistema SiasXml al área de procesos para su supervisión las 7 por 24
12/Noviembre/2004	Subscripción sistema Clarify, desarrollo final.
14/Diciembre/2004	Subscripción Pisa Troncales (Pisa Tk)
25/Enero/2005	Actualización a la Interfaz Sisa – Pisa Quejas
24/Febrero/2005	Puesta en producción Clarify
30/Mayo/2005	Análisis de requerimiento IOM del gestor de aprovisionamiento integrado (IOM, Integrated Order Manager)
27/Julio/2005	Subscripción y puesta en producción Sisa – IOM
08/Agosto/2005	Desarrollo para notificación por celular, mail desde SisaXml
09/Septiembre/2005	Subscripción sistema PROSA
07/Noviembre/2005	Reactivación de proyecto LATAM

Fecha	Observaciones
19/Enero/2006	Evolución al sistema Pisa Tk
24/Abril/2006	Evolución al proyecto Sisa -IOM
24/Abril/2006	Tercera versión Sisa – Pisa Quejas
07/Julio/2006	Actualización Mensaje Universal XML con un nuevo analizador sintético
02/Agosto/2006	Subscripción sistema EMMA envío de mensajes a celular masivos (alertas)
20/Septiembre/2006	Pruebas entre mensaje Sisa – LATAM
25/Enero/2007	Permisos a servidor de correo Share
21/Mayo/2007	Subscripción sistema Siatel
07/Julio/2007	Evolución Pisa – Tk
17/Julio/2007	Subscripción sistema TRALIX notificación formateado de clientes
17/Agosto/2007	Migración de servidor
31/Agosto/2007	Automatización de procesos SisaXml procesión, procesoout
10/Abril/2008	Siatel
16/Mayo/2008	Subscripción a Argos Portabilidad
17/Julio/2008	Subscripción sistema Pisa 900
24/Febrero/2009	Subscripción sistema Pisa -800
18/Marzo/2009	Subscripción del escalador de quejas a SisaXml
14/Julio/2009	Evolución al sistema Pisa Tk
20/Agosto/2009	Subscripción Pisa LP
21/Noviembre/2009	Cuarta versión Sisa – Pisa- Pisa Plex (Pisa Quejas)

Fecha	Observaciones
21/Enero/2010	Evolución Pisa Tk
23/Marzo/2010	Evolución Proceso de bajas Pisa Lp

Apéndice K Glosario

Termino	Descripción
Came	Centro de Atención Metropolitano
Cao	Centro de Atención a Operadores.- Entidad responsable de notificar a las compañías de telecomunicaciones autorizadas en el país cuando el personal operativo de Telmex efectúe labores de operación y mantenimiento en la red.
Car	Centro de Administración de la Red Telmex.- Es la entidad responsable en el ámbito regional y nacional de supervisar la red de telecomunicaciones de las direcciones divisionales.
Care	Centro de Atención de Redes Empresariales
Cas	Centro de Atención de Servicios
Cliente	a la que se le proporciona un servicio de Telecomunicaciones Entidad
Clli	Identificación de Localidades en Lenguaje común. / Código de lenguaje común que identifica las ubicaciones individuales de una planta física. Un código CLLI está compuesto por 8 u 11 caracteres alfanuméricos. Los primeros 4 caracteres son un número de lugar abreviado. Los caracteres 5 y 6 son abreviaturas de estado. Las posiciones 7 y 8 identifican una construcción específica y 9,10 y 11 representan una pieza de equipo en particular.
Cm	Centro de Mantenimiento.- Es la entidad responsable de asegurar la eficiente operación y mantenimiento de los elementos de red de telecomunicaciones de las direcciones divisionales y su infraestructura.
Cns	Centro Nacional de Supervisión.- Es la entidad responsable a nivel nacional de supervisar la red de telecomunicaciones de Larga Distancia.
Com	Centro Operativo de Mantenimiento
Ld	Larga Distancia.- Es el sistema de telecomunicaciones utilizado para establecer los servicios de larga distancia proporcionada por Teléfonos de México.
Mac	Módulo de Atención a Clientes.- Centro de atención y servicio al público clasificado por Telmex como Grandes Clientes y Clientes Premier.
Restauración	Recuperación o restablecimiento del servicio del cliente aunque persista la falla. • REPARACION: Compostura de la falla reestableciendo el servicio en el estado en que se encontraba inicialmente
Servicio	Cualquier categoría única y funcional de la red de Telmex. El código del sistema cuando se integra a su localización, estado y edificio proporcionará una identificación única en el mundo
Sisa	Sistema de Información de Servicios Avanzados
Triara	El Centro de Datos Triara que surge como resultado de una co-inversión entre Telmex y Hewlett Packard provee servicios de infraestructura vitales para mantener a su empresa lista para su crecimiento y expansión. El Servicio Administración de sistemas del Centro de Datos Triara incluye una amplia gama de servicios, que abarcan desde la administración del sistema operativo de sus equipos, hasta la administración completa de sus aplicaciones ERP, dependiente de las necesidades de su empresa. ✚ Configuración del sistema - Coordinación e implementación de cambios lógicos o físicos en el ambiente de cómputo. ✚ Soporte de desempeño - Corrección de fallas localizadas a nivel sistema operativo. ✚ Configuración del usuario - Asegurar que los recursos del ambiente de cómputo estén disponibles para los usuarios finales, proporcionando a cada uno de ellos, los niveles y atributos de acceso apropiados. ✚ Soporte de sistemas - Soporte y Mantenimiento del Hardware y Software. ✚ Administración de bases de datos - Recuperación de los ambientes de bases de datos utilizando las herramientas y procesos proporcionados por el proveedor o fabricante. ✚ Administración de Sistemas ERP - Apoyo en la implementación de actualizaciones de sistemas ERP; instalación de parches o actualizaciones al sistema ERP; instalación de agentes de monitoreo pro-activo; y elaboración de copias dentro de una misma instancia y realizar exportación / importación de información entre instancias.
Sistema, aplicación	En informática, una aplicación o sistema es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajo.
Siatel	En este sistema se lleva la base de datos del personal de Telmex, la relación con sis es que no se permite ingresar a usuarios que no sean de Telmex o de alguna filial. Fue una de las interfaces que se migraron al Mensaje Universal Sisa Xml (SisaXml)

Termino	Descripción
Sise	Lleva la administración y el inventario de la planta.
Sockets	Los sockets son un sistema de comunicación entre procesos de diferentes máquinas de una red. Más exactamente, un socket es un punto de comunicación por el cual un proceso puede emitir o recibir información. Los cuales son muy complejos en su programación y mantenimiento, además que se requiere del conocimiento interno de las dos aplicaciones para realizar la distribución de la información. En caso de involucrar base de datos se debe realizar a un nivel bajo como ProC para poder realizar el intercambio de información esto siempre y cuando se trate de una misma base de datos, de lo contrario se requiere de ODBC que puedan realizar la interpretación de la misma.
Diagrama de contexto	El diseño del sistema es la primera etapa donde se selecciona la propuesta básica para resolver el problema. Durante el diseño del sistema, toda la estructura y estilo se deciden. La arquitectura del sistema de toda la organización se divide en componentes llamados subsistemas. La arquitectura proporciona el contexto para tomar las decisiones más detalladas. Al tomar las decisiones de alto nivel que se apliquen al sistema entero, el diseñador del problema la divide en subsistemas para que el trabajo futuro pueda ser realizado por varios diseñadores que trabajen independientemente en subsistemas diferentes, estos se representan en un diagrama llamado "Diagrama de contexto del sistema"

Apéndice L Documentación generada.

Anexo

Bibliografía

Titulo	Autor	Editorial
Programación avanzada con XML	Fabio Arciniegas	McGraw-Hill
Servlets y JavaServer Pages	Marty Hall	Pearson Educación
Programación UML	Metodología de desarrollo	
Manuales Oracle		V9.2.2
Sql y java	Jim Melton Andrew Eisenberg	Alfaomega
Aprendiendo UML en 24 Horas	Joseph Schmuller	Prentice Hall
Origin of the term middleware	Gall, Nick	(30 de julio de 2005)
Definition of middleware	Carnegie Mellon	Software Engineering Institute

Referencias

Descripción	Liga
Comunicación entre aplicaciones	http://msdn.microsoft.com/es-es/library/aa292464%28VS.71%29.aspx
Comunicación con sockets	http://zarza.usal.es/~fgarcia/doc/tuto2/V_2.htm
Información de la empresa Telefonos de mexico	http://www.telmex.com/mx/corporativo/index.html
Información de telefonos de mexico Internacional	http://www.telmexinternacional.com/index.html
Telmex	http://es.wikipedia.org/wiki/Telmex
Manuel de Java	http://www.freebsd.org/doc/en/books/developers-handbook/sockets.html
BSD Sockets	http://www.ecst.csuchico.edu/~chafey/prog/sockets/sinfo1.html
Glosario XML	http://msdn.microsoft.com/en-us/library/ms256452%28v=vs.80%29.aspx
Rápida Introducción a XML	http://www.adobe.com/es/devnet/dreamweaver/articles/xml_overview_03.html
Diseñar una estructura de datos XML	http://www.XML.org
Desarrollo Soluciones XML con la Tecnología JavaServer Pages	http://www.programacion.com/articulo/desarrollo_de_aplicaciones_web_con_jsp_y_xml_120
Mostrar un fichero XML con el analizador SAX	http://www.programacion.com/articulo/el_api_jaxp_113/21
ObjectWeb Open Source Middleware	http://middleware.objectweb.org/
Software Engineering Institute	http://www.sei.cmu.edu/str/descriptions/middleware.html
Digital Rights Management Middleware	http://www.wireless.ucla.edu/techreports2/sun05_final.pdf
Internet2 Middleware Initiative	http://middleware.internet2.edu/
Middleware	http://www.opensubsystems.org/core/index.html

Descripción	Liga
Abstraction Layer	
Gerencia de Documentación y estandarización	Intranet documentación sobre la estandarización en el desarrollo de interfaces en la empresa http://estandarizacion/
PMI	El PMI ha publicado una guía para la gestión de proyectos denominada PMBOK, esta contiene todas las pautas (que son generales) para la gestión de cualquier tipo de proyecto sea de TI, de construcción, etc. La ultima edición del PMBOK es la 2000, pero en Internet he encontrado la versión del año 1996, http://www.monografias.com/trabajos12/pmbok/pmbok.shtml
Los modelos de CMMI	Implementación de CMMI (En castellano) en el ITBA (Instituto Tecnológico de Buenos Aires): http://www.itba.edu.ar/capis/webcapi...demagister.pdf http://www.sei.cmu.edu/cmmi/models/
Facilidades para educación	https://seir.sei.cmu.edu/seir/
La presentación del anteproyecto	http://www.uaq.mx/matematicas/ptesis/chispas.pdf
Como conducir un proceso de mejora de la construcción de software (SPI	http://gridtcs.frm.utn.edu.ar/docs/ENIDI%2008%20Herram%20eval%20mejora.pdf