

Especialización en Sistemas y Servicios Distribuidos

Trabajo Final Especialización

Tema:

Plataforma distribuida de venta de productos virtuales

Tutor

Cilia, Mariano
Blanco, Javier

Alumnos

Ferreyra, Emanuel
García Mattío, Mariano

Fa.M.A.F.

Índice

Agradecimientos	3
Abstract	4
1. Introducción	5
2. El requerimiento	8
3. Análisis de la solución	14
4. Tecnologías para la solución	17
5. Arquitectura	21
6. Conclusión	22

[Apendice A. Comparación de tecnologías.](#)

Agradecimientos y dedicatorias

*Le dedico este trabajo a mi familia que me apoyó incondicionalmente.
A Catalina, por las horas que no estuviste con papá, Andrea por las horas
de apoyo y contención. A Emanuel, sin palabras, un gran amigo y compañero.*

Mariano

*Quiero agradecer y dedicar este logro a Claudia, por su paciencia y apoyo
constante y también a mi familia. A Mariano porque una vez confirmo el tipo
persona y amigo que sos.
Y por supuesto a Dios, que me acompaña en todo momento.*

Emanuel

Abstract

En este trabajo se plantea un problema o requerimiento de un negocio real, presentado por un grupo de empresarios relacionados a la industria de la telefonía celular.

El requerimiento se traduce en la necesidad de un sistema distribuido, y en este trabajo desarrollamos el proceso de análisis de dicho requerimiento hacia los componentes de un sistema.

Se analizarán los componentes, proponiendo los diseños e implementaciones más convenientes, para finalizar visualizando una arquitectura posible del sistema distribuido que satisfaga los requerimientos iniciales.

1. Introducción

Presentaremos inicialmente algunos conceptos, necesarios para la adecuada comprensión de este trabajo.

Telcos. Con este nombre se identifican a las grandes empresas que brindan soluciones de telecomunicaciones, generalmente empresas internacionales. Ejemplos de estas empresas en nuestro país son Telecom, Telmex y Telefónica.

Particularmente para la orientación de este trabajo, nos interesan las Telcos que participan en la industria de la telefonía celular: Personal (Telecom), Claro (Telmex) y Movistar (Telefónica).



Pospago vs Prepago

Las Telcos ofrecen un abanico de productos para los consumidores finales. En el caso de la telefonía celular, una de las divisiones primarias que se concibe, es la referida a los productos prepagos y pospagos.

Los productos pospagos son aquellos en los que el consumidor tiene un abono cíclico fijo, normalmente mensual, y dispone de un crédito que puede ir consumiendo. Al final de cada ciclo el consumidor recibe una factura por el abono correspondiente, y se habilita nuevamente su crédito. Esta modalidad no es la que nos interesa.

En el caso de los prepagos, el consumidor no cuenta con una facturación cíclica, no con un crédito y abono mensual, sino que debe pre-comprar su crédito antes de poder utilizarlo. Cuando se compra crédito y se acredita a la línea celular, entonces el consumidor está en condiciones de utilizar el servicio de la Telco.

El consumidor de una línea prepaga tiene actualmente dos opciones principales para acreditar saldo a su línea.

La primera y tradicional es comprar una tarjeta física en un punto de venta, que está asociada a un monto, y que posee un pin numérico. El consumidor entonces una vez adquirida la tarjeta llama a un IVR (Interactive Voice Responsing), que lo guiará en un proceso que solicita el pin, para finalizar acreditando el monto en la línea. También es posible actualmente en algunas telcos enviar un sms en lugar de interactuar con un IVR.

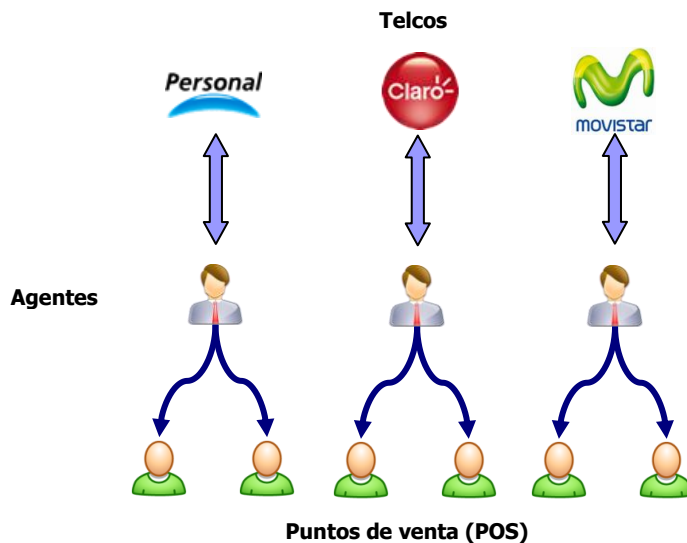
La segunda opción es utilizar un sistema que esté interconectado a los sistemas de las telcos, y acreditar en forma electrónica a la línea celular el saldo. La posibilidad de hacer esto es uno de los pilares de este trabajo. Este producto en particular, se conoce como saldo virtual, o carga virtual.

Pospago vs Prepago

Las telcos no comercializan sus productos en forma directa a los consumidores finales. En cambio, lo hacen a través de una red exclusiva, encabezada por Agentes oficiales. Estos Agentes son los únicos que pueden comprar directamente los productos, desde equipos, abonos y tarjetas, hasta canales virtuales para saldo virtual.

Trabajar de esta forma permite a las telcos, entre otras cosas, mantener cierta uniformidad respecto a precios y modalidades de venta de sus productos, ya que los Agentes deben cumplir varias reglamentaciones para mantener su representación oficial.

Luego, los Agentes redistribuyen de la misma forma los productos, hacia distribuidores, subdistribuidores y puntos de venta.



Dos puntos que no deben pasar desapercibidos son los siguientes: Un agente oficial de una telco, tiene beneficios muy marcados en términos de costos de compra de productos, justamente por ser agente oficial. Y con su carácter exclusivo, tiene prohibido comercializar directamente productos de otras telcos, situación que puede poner en peligro su representación. Los dos puntos mencionados, son también fundamentales para la concepción del sistema que presentaremos en este trabajo.

Finalmente en esta red comercial, el punto de venta es el último eslabón, y quien realiza la venta al consumidor final.

Las ventajas de la carga virtual

En el último tiempo ha tomado tracción la modalidad de carga virtual frente a la otra opción, las tarjetas físicas, por ciertas diferencias, más allá de las obvias, entre ambos productos.

Entre los factores que influyen a favor de la carga virtual, está el problema del stock, posteriormente se comprenderá mejor este concepto dado el funcionamiento del sistema. Por ahora, basta con decir que en el caso de tarjetas físicas hay que mantener stocks por cada telco y por cada diferente monto de tarjetas (\$5, \$10, \$20, para Claro, Movistar y Personal). Esto se

vuelve tedioso y permite que los puntos de venta pierdan la posibilidad de vender si tienen faltantes de una determinada tarjeta en particular. Por otro lado, si poseen tarjetas que no se venden, pasan a tener un problema de stock inmovilizado. La modalidad de cargas virtuales soluciona ambos problemas.

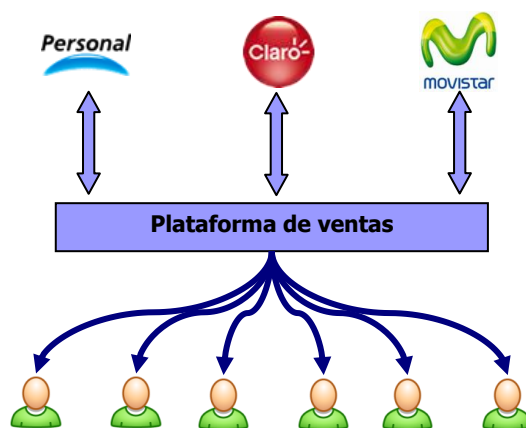
Otro factor interesante es que en la carga virtual el producto físico desaparece, y con él todos los problemas derivados de esa característica. No es necesario transportar las tarjetas, el saldo virtual no se puede robar ni extraviar, no se deteriora y no se vence.

2. El requerimiento

En esta industria, el punto de venta requiere de forma excluyente comercializar el saldo para prepagos de todas las telcos (Claro, Personal y Movistar). Tácitamente no se admite la venta para una única telco.

Un grupo de agentes que en conjunto representan a las tres Telcos, y que poseen cuentas para la venta de saldo virtual, pretende formar un consorcio, para unir esos recursos y comercializarlos en forma conjunta. Así aprovechan los beneficios que tienen como representantes oficiales y evitan comprar productos de la competencia, respetando su exclusividad.

Para trabajar con este producto virtual, y trasladarlo desde las telcos a través de la red de ventas hasta los puntos de venta, requieren una **plataforma de software distribuida**, permitiendo entre otras cosas la venta al consumidor final y la administración del saldo virtual.



En esta plataforma se deberá ofrecer la capacidad de realizar cargas virtuales a líneas prepagas de todas las telcos, utilizando en cada caso el canal virtual exclusivo que el agente tiene habilitado. Quien consuma este servicio estará accediendo a una misma interfaz, sin importar las complejidades y características independientes de cada telco o tecnología de conexión, por lo que estaremos aplicando **patrones de diseño (fachada)** para resolver este punto.

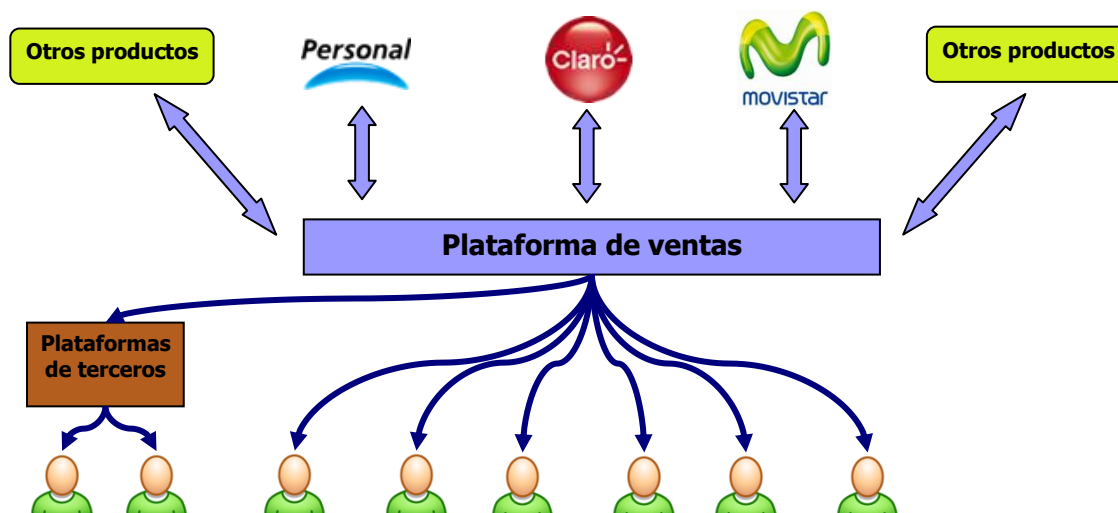
El punto de venta obtiene entonces una aplicación de software que le permite vender a cualquier línea prepaga del mercado, accediendo de esa forma a los beneficios que hemos detallado anteriormente.

El consorcio también presenta una modalidad diferente al momento de adquirir el stock. Recordemos que previamente era necesario especificar qué producto se deseaba comprar: tarjeta Claro de \$10, tarjeta Personal de \$20, etc. Con la plataforma de ventas, cada usuario (distribuidor, punto de venta) simplemente opera comprando un monto, que llamamos bolsa de dinero. Luego utiliza su bolsa de dinero para vender el producto que desee, carga virtual de Claro de \$10, o carga virtual Personal de \$20. Claramente su stock

necesario para ese producto disminuye drásticamente, lo que es un beneficio muy valorado.

Bajo esta modalidad de la bolsa de dinero, el consorcio requiere además, que la plataforma tenga modularidad suficiente para permitir incorporar nuevos productos, que funcionen de la misma manera. Ejemplos de posibles productos son: entradas para espectáculos, boletos de transporte, pines Veraz, etc.

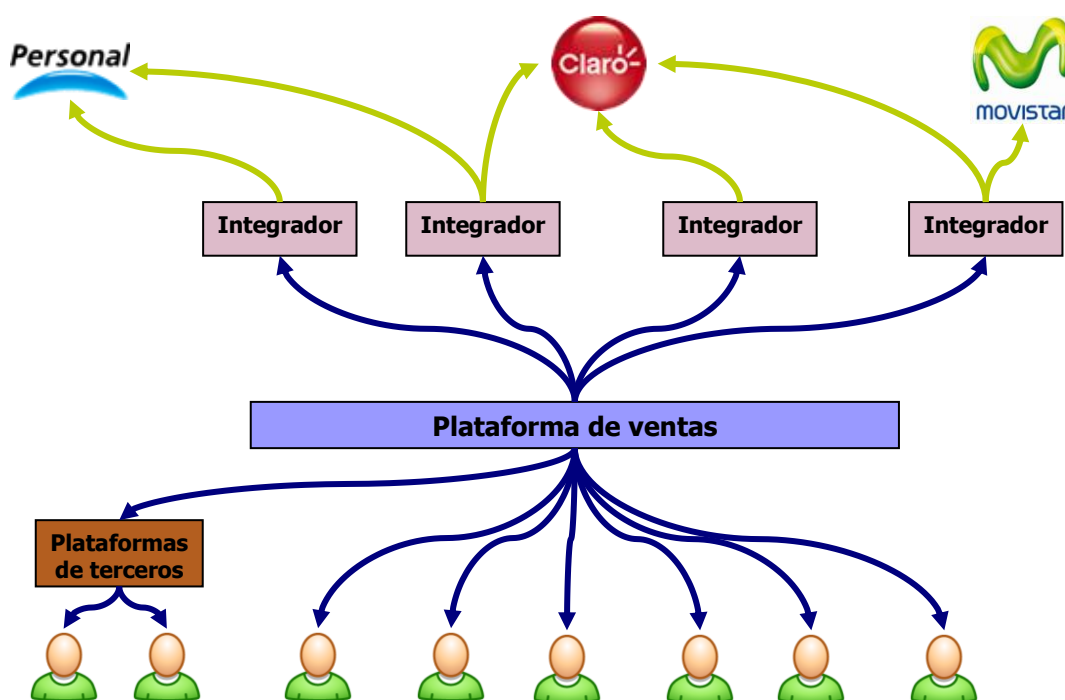
Existen otras redes de puntos de venta (por ejemplo RapiPago), que son candidatas a vender el producto de carga virtual, pero ya disponen de su propia plataforma de ventas. Para éstos debe preverse que nuestra plataforma ofrezca un canal para consumir el servicio, sin pasar por la aplicación de ventas. Aquí es necesario una interfaz para integraciones **B2B**.



Otra característica del negocio, es que las telcos no ofrecen su servicio de recargas electrónicas directamente a los agentes. En cambio existen empresas u organizaciones, llamadas integradores, que hacen de intermediarios para el servicio.

Esta realidad implica por un lado que un agente puede acceder a los canales virtuales de las telcos a través de varios integradores, y por otro lado, implica mayor complejidad a nivel sistemas, ya que cada integrador ofrece el servicio de la manera (y con la tecnología) que él decide, esto se debe a que las telcos no lo regulan.

Vemos a continuación un esquema que representa lo explicado:



Aquí puede apreciarse por ejemplo que para Claro existen tres integradores que permiten el acceso al canal virtual de la telco, y que un mismo integrador, el de la derecha, permite el acceso de Claro y a Movistar.

Los agentes seleccionarán los integradores a utilizar, basándose fundamentalmente, en las condiciones del negocio, como comisiones y crédito, pero también tendrán en cuenta la calidad del servicio que ofrecen.

El consorcio conoce el ambiente en que se utilizarán los clientes de la plataforma, como así también, ciertas características que es importante considerar.

El punto de venta por excelencia, es una pc en un telecentro o kiosko. Es normal que la PC posea pobres recursos y que la misma funcione en forma "lenta".

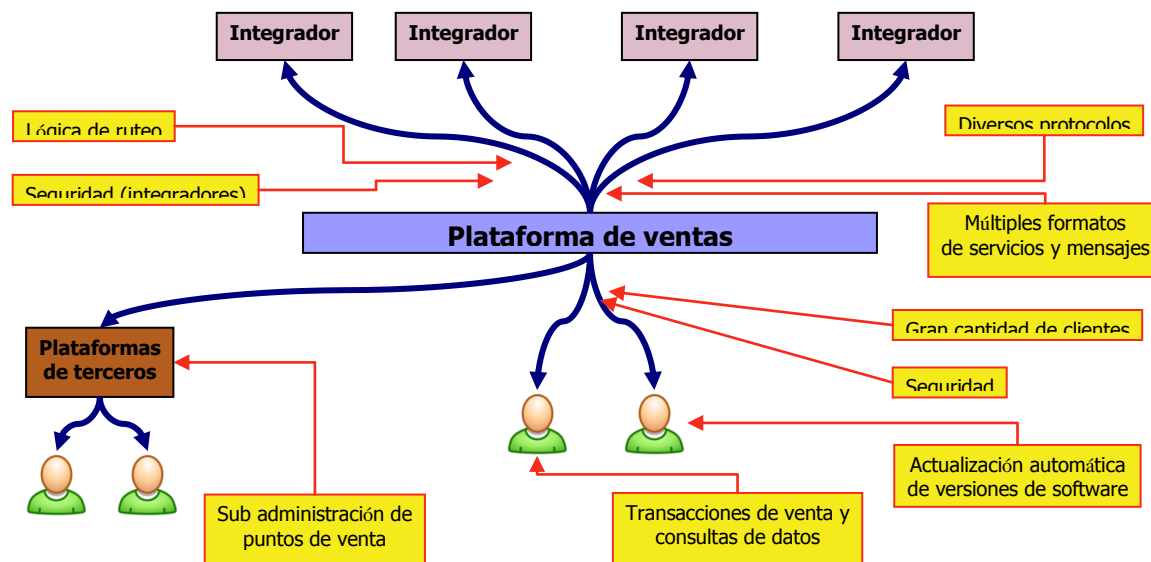
El operador del punto de venta necesita una interfaz de usuario extremadamente simple, al menos para la venta de saldo virtual (aquí estamos hablando de una pantalla con 2 cajas de texto, para número de línea y monto y un botón para hacer la carga).

Es ideal que en una pantalla de la aplicación, puedan visualizarse al mismo tiempo las opciones de carga para todas las telcos, dado que le permite al operador mantenerlas abiertas a la espera de los consumidores, y realizar la venta rápidamente.

También sucede que los operadores intentan varias cargas en forma consecutiva, y como la solicitud de carga puede demorar entre 3 y 15 segundos, requieren que el sistema permita realizar esto sin tener que esperar a que finalice la primera antes de continuar. Aquí es necesario disponer de una tecnología de **comunicación asíncrona**, para permitir múltiples cargas desde un punto de venta sin necesidad de esperar la respuesta de una para iniciar la siguiente.

Análisis de la arquitectura

Siguiendo nuestro esquema inicial, analizaremos ahora los aspectos que deben considerarse en la arquitectura para tomar las decisiones correctas respecto a tecnologías y diseño.



Comenzaremos describiendo los aspectos de la plataforma "hacia arriba", luego, continuaremos con los relacionados al servicio que la plataforma brindará "hacia abajo".

Diversos protocolos

Como se explicó, cada integrador define la tecnología sobre la cual ofrecerá el servicio de cargas virtuales. Tendremos entonces diversos protocolos, dependiendo del integrador. Por ejemplo, algunos utilizan WebServices y otros sockets. Es necesario que la plataforma oculte esta diversidad, trabajando en forma indistinta, con uno u otro integrador, también es deseable que quien consume el servicio de ventas de la plataforma sea abstraído de ese detalle.

Múltiples formatos

Así como el protocolo varía entre integradores, pasa lo mismo con el mensaje que se envía y recibe sobre ese protocolo.

Ciertos integradores aceptan mensajes SOAP, otros mensajes XML y otros directamente tramas de datos definidas por ellos mismos, *formatos propietarios*.

También varían los datos a ser enviados, aunque en ese caso, algunos son estándar, ya que los requiere directamente la telco. Básicamente siempre es necesario enviar el número de línea celular y el monto a cargar.

Esto es importante, porque como veremos nos permitirá definir una interfaz estándar para todos los servicios.

Seguridad (integradores)

El integrador también define la seguridad del canal de comunicaciones. En este caso pueden requerir la utilización de VPNs, el acceso seguro mediante certificados SSL, o bien, requerir datos para autenticación y autorización dentro del mismo mensaje de request del servicio.

Lógica de ruteo

El servicio de los integradores, es en esencia el mismo, conectarse a una telco e ingresar una carga virtual. Sin embargo, estos integradores tienen características comerciales diferentes, dados por los acuerdos que ellos mismos realizan con las telcos. Esta diferencia de condiciones comerciales se traslada a los agentes, y es normal que en determinados momentos les convenga utilizar un integrador, y en otro, uno diferente. Es necesario entonces, que sea posible establecer una lógica para decidir que integrador utilizar al momento de realizar una carga, ésta lógica, puede ser tan simple como establecer manualmente el integrador para cada telco, o bien, tan complejo como elegir el integrador adecuado basándose en días u horas de la semana, estado del servicio del integrador, comisiones que se están pagando, etc.

Las modificaciones deben poder realizarse en forma dinámica, sin detener el funcionamiento de la plataforma.

Gran cantidad de clientes

La red de distribuidores y puntos de ventas será numerosa en términos de cantidad de clientes de software para la plataforma.

Esto indica que el sistema debe ser escalable, y los recursos, especialmente los de servidor, deben ser utilizados cuidadosamente, evitando fallas causadas por la cantidad de clientes.

Estamos hablando en este punto de **escalabilidad y disponibilidad**, los cuales si bien deben ser considerados no son objetivos en este trabajo.

Transacciones de venta y consultas de datos

En esta plataforma, identificamos dos tipos de actividades que un cliente realizará, y que deben ser atendidas por servicios de la plataforma. La primera actividad está formada por las transacciones de venta. Para éstas, el servicio debe ser seguro y requerir autenticación, ya que accesos no autorizados ponen en riesgo la integridad de los datos de la plataforma, que trabaja directamente con dinero.

Por otro lado, la plataforma debe poner a disposición de los clientes los datos e información transaccional, en este caso no es crítica la seguridad, por otro lado, sí lo es la velocidad de respuesta a las consultas de datos. Con esto en mente, se analizará la solución adecuada para no poner en riesgo la seguridad en las transacciones de venta, y lograr alta performance en las consultas de datos.

Seguridad

Como describíamos en el punto anterior, para las transacciones de venta deberemos hacer uso de una tecnología y protocolo que permita proteger el

servicio, y para las consultas, evitar utilizar aquellas que sobrecarguen los datos o el canal de comunicaciones, disminuyendo la velocidad de respuesta.

Seguridad tampoco es un objetivo en nuestro trabajo, aunque consideraremos algunos aspectos básicos en ciertos puntos.

Actualización automática de versiones de software

Dada la posible extensión de la red de ventas, los agentes han requerido que el cliente de software, que utilizan distribuidores y puntos de venta, tenga la capacidad de actualizarse automáticamente, esto quiere decir, que el usuario no tenga que realizar operaciones complicadas para tener siempre la última versión del software. Una de las principales razones para solicitar esto, es que se pueden incorporar cambios a la plataforma en forma continua, e impactar en los clientes sin necesidad de trabajo extra.

Se ha tenido en cuenta que el formato más adecuado para este objetivo seguramente es uno **SaaS** (Software as a Service), con tecnología web. Sin embargo, al momento de iniciar la primera versión de la plataforma, fue un requerimiento explícito que el software para el usuario fuera una aplicación instalable en PCs.

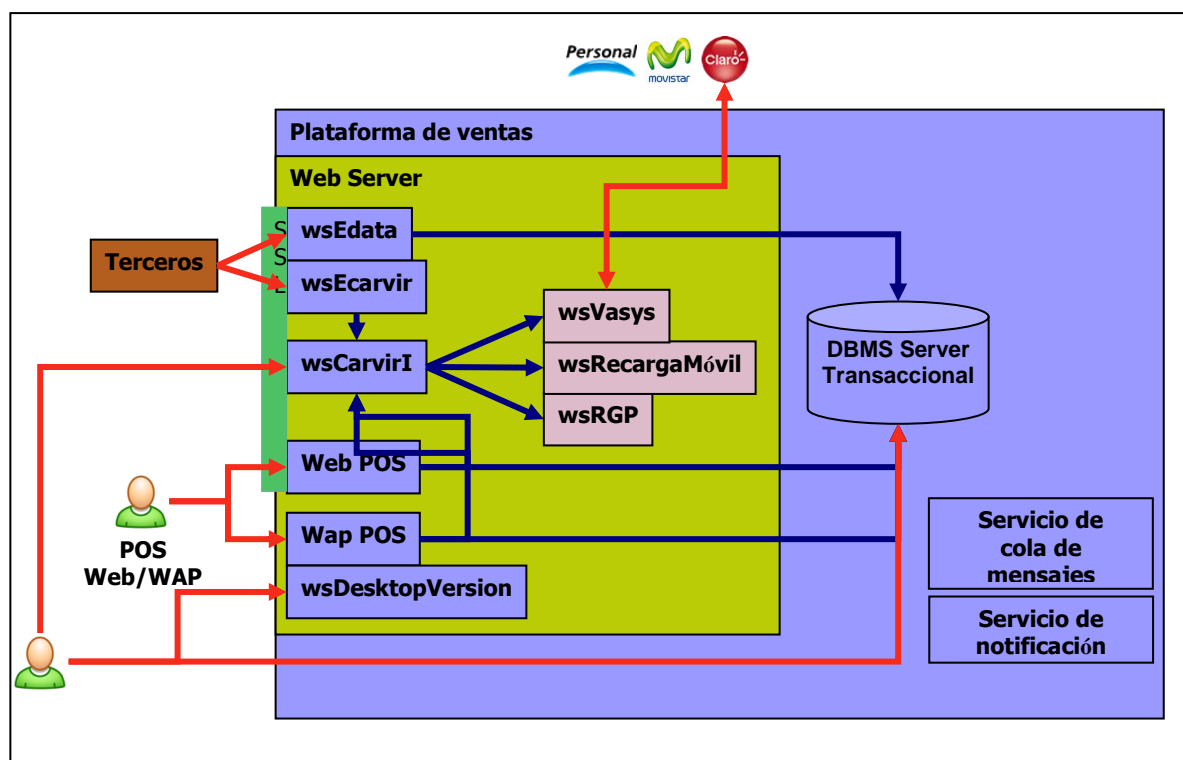
Actualmente se está realizando la evaluación para decidir el diseño e implementación de una **RIA** (Rich Internet Application) utilizando Adobe Flex, y migrar luego la plataforma a esa tecnología.

3. Análisis de la solución

Hemos presentado el problema detallando sus requerimientos que debemos resolver, como se habrá observado, se trata de una oportunidad de negocios del mundo real, que requiere como solución, una aplicación de software distribuida.

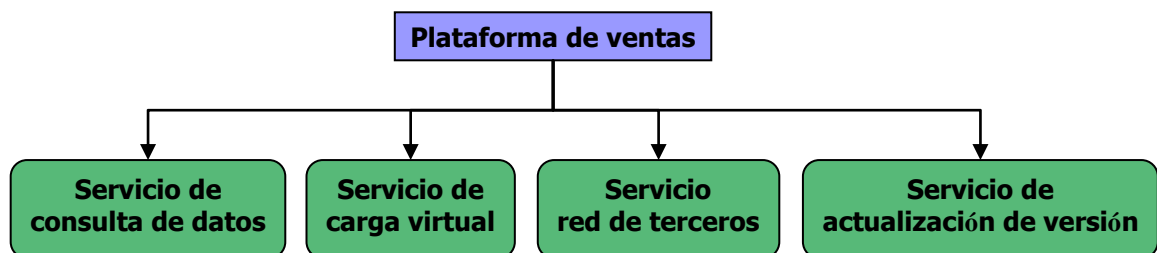
A continuación describiremos la solución, primero en forma lógica, luego analizando la tecnología más conveniente para implementarla.

Para contribuir y facilitar la comprensión de la arquitectura que proponemos utilizar la presentamos a continuación, e iremos analizando cada parte que la conforma.



Teniendo en cuenta que este es nuestro objetivo final, comenzamos a describir cómo está compuesta.

Hemos identificado los siguientes servicios lógicos, que deben estar presentes en la plataforma, estableciendo objetivos que deben cumplir, y aclaraciones a tener en cuenta.



Servicio de consulta de datos

Objetivo

- Ofrecer datos primando la velocidad de respuesta.

Los datos que este servicio pondrá a disposición de los clientes no son críticos en su gran mayoría, por lo que no es necesario asegurar el canal de comunicación por el que se moverán.

Los datos sensibles ofrecidos por el servicio, deberán ser encriptados o protegidos de otra manera (por ejemplo, autenticación).

Para que la velocidad de respuesta sea mayor, evitaremos encriptar el canal de comunicaciones, disminuyendo el tamaño de datos a transferir.

En el caso de datos de login, se almacena en el servidor un hash de los mismos, por lo que nunca viajan en texto claro.

La gran mayoría de los datos que se transfieren por este servicio son transaccionales: listados de cargas realizadas, mensajes, resumen de comisiones, etc.

El costo de transferir datos sensibles hasheados en un canal sin seguridad versus un canal con seguridad es mucho menor para el primer caso, dado que encriptar el canal produce que el dato a transferir incremente su tamaño, además de aumentar considerablemente el procesamiento de los mismos.

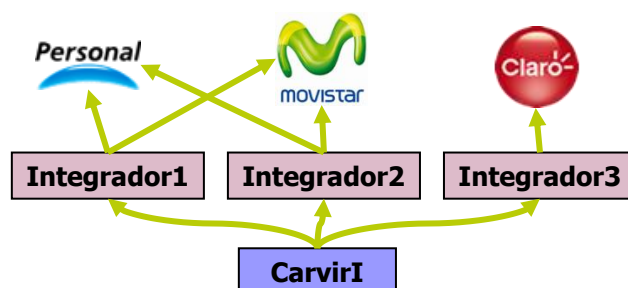
Servicio de carga virtual

Objetivos

- Proveer el servicio para la venta de saldo virtual.
- Ofrecer una interfaz unificada ocultando las diferencias y complejidades de los servicios específicos de cada Telco o integrador.
- Permitir la gestión y ejecución de lógicas de ruteo.

Este servicio debe ser utilizado a través de un canal seguro. Será consumido desde diversas tecnologías: web, wap, aplicación de escritorio, otras plataformas, y lo deseable, es que el acceso sea transparente, independientemente de quien desee consumirlo y con qué. El servicio, indiscutiblemente, deberá ofrecer una interfaz estándar.

Para este servicio lo adecuado sería la utilización de un patrón de fachada, se puede apreciar en la figura siguiente una vista de su posible implementación:



En la imagen vemos que la fachada del servicio CarvirI, estandariza la interfaz de la recarga virtual, y se encarga de invocar el servicio de cada integrador, en donde se implementa.

Servicio de actualización de versión

Objetivo

- Proveer un método de actualización automática del software para los clientes de la plataforma.

Los agentes que utilizan la plataforma, necesitan que los clientes distribuidos tengan la posibilidad de ser actualizados, incorporando nuevos servicios y funcionalidades, evitando que los usuarios finales tengan que atender las actualizaciones.

Este aspecto es dependiente de la tecnología en que se implementarán los clientes, en caso de ser web y wap el problema está solucionado naturalmente por esos modelos, y en el caso de una aplicación instalable en la pc, si requerimos un servicio de actualización.

Servicio red de terceros

Objetivo

- Ofrecer un canal exclusivo para la consulta de datos y uno para la venta de productos.

Este servicio debe implementarse teniendo en cuenta que los terceros pueden disponer de tecnologías de cualquier tipo, por lo que debe utilizarse nuevamente una tecnología que sea estándar.

Las comunicaciones deben ser seguras para estos datos, por pedido explícito del consorcio, para mantener los datos de terceros en forma segura.

4. Tecnologías para la solución

Luego de haber analizado en una forma lógica las características y aspectos que la solución debe alcanzar y cubrir, continuaremos ahora analizando las implementaciones reales que permiten esas funcionalidades.

Debemos considerar que existen diferentes tecnologías que tienen funcionalidades similares, y también fortalezas y debilidades, que permiten elegir la opción adecuada dependiendo de las necesidades reales de nuestro problema.

Por ello justificaremos la elección de una tecnología ante otra en cada servicio, al prevalecer una mayor cantidad de fortalezas o necesidades de negocios.

Dado que ciertas tecnologías aparecerán recurrentemente en ciertos servicios o funcionalidades, describiremos las características de cada tecnología luego de presentar para cada servicio la que consideramos más adecuada.

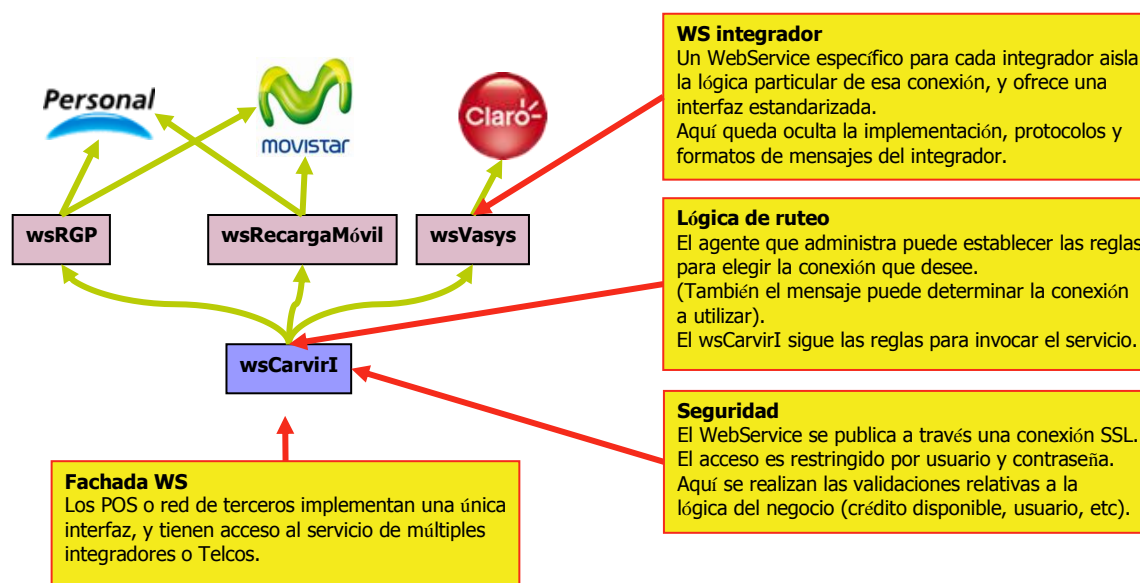
Servicio de carga virtual

La tecnología elegida para implementar este servicio es WebServices.

Consideramos también como opciones a RMI, DCOM y CORBA, pero prevaleció WebServices.

Entre las ventajas conocidas de los WebServices, y muy relevantes para las características de este servicio de venta, tenemos que posibilita un bajo acoplamiento entre la funcionalidad de la venta de cargas y la aplicación que hará de interfaz con el usuario. De esta forma el servicio puede ser utilizado fácilmente por diferentes aplicaciones, incluso de tecnologías diferentes, sin necesidad de customizaciones adicionales.

La otra ventaja importante, es la disponibilidad sobre protocolos y puertos estándares, http sobre puerto 80, lo que evita posibles trabajos de configuración sobre firewalls y dispositivos de red para el correcto acceso al servicio. Utilizando WebServices una configuración estándar es suficiente.



Como habíamos analizado anteriormente, y decidido que era adecuado utilizar un patrón de fachada, mostramos aquí la implementación correspondiente, con un WebService wsCarvirI que es la única interfaz que debe implementar el software cliente que desee utilizar el servicio.

Este WebService debe ser ofrecido a través de una conexión SSL (Secure Sockets Layer) para mayor seguridad en los datos (aspectos más detallados acerca de SSL y seguridad, no serán cubiertos en este trabajo, ya que no se encuentran en los alcances previstos), ya que entre ellos están incluidas las credenciales de acceso del usuario de la plataforma.

En este WebService también se ejecutan las validaciones necesarias según las reglas del negocio del consorcio, como el crédito disponible de un usuario, recargas a números repetidos en cortos lapsos de tiempo, etc.

En este punto, la plataforma tendrá también la capacidad de enrutar la petición de venta hacia el integrador adecuado, dependiendo de una serie de reglas que pueden ser estáticas o dinámicas, dependiendo las necesidades de los integradores. El requerimiento inicial para esto, es que el administrador de la plataforma (miembro del consorcio), crea a través de la aplicación la regla estática que indica que integrador utilizar, que es almacenada en BD. El servicio web de carga virtual evalúa las reglas al procesar las peticiones, decidiendo de esa manera que integrador utilizar.

Hacia “arriba”, wsCarvirI invocará a otros WebSevices, que también deben ofrecer una interfaz en común, como es el caso en el gráfico de wsRGP, wsRecargaMovil o wsVasys. Al implementar la misma interfaz estándar cada uno de estos WebServices específicos de cada integrador, es posible desde wsCarvirI utilizar las mismas invocaciones, y también intercambiar o agregar nuevos servicios de integradores, sin afectar a wsCarvirI.

Por último, y como hemos propuesto, cada WebService específico se encarga de implementar y ocultar la complejidad y lógica propia de una conexión en particular, lo que incluye protocolos, formatos de mensajes y lógicas de negocio.

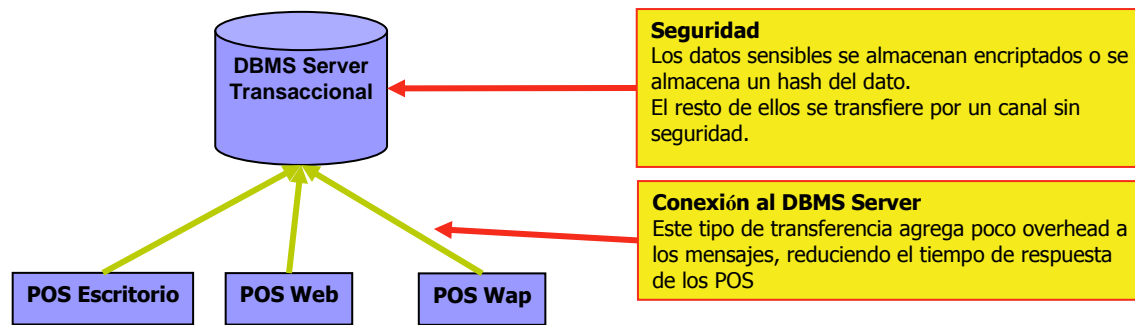
Servicio de consulta de datos

Para este servicio la tecnología a utilizar es una conexión al DBMS (DataBase Management Sytem), dado que el requerimiento principal es la velocidad de respuesta (**Acceso directo a la BD**).

Comparando ante las demás alternativas: EJBs o WebServices, consideramos más adecuada la opción de consulta directa al DBMS.

La desventaja principal de estas otras dos tecnologías, es el overhead que incorporan, afectando la velocidad de respuesta.

Una aclaración que vale la pena hacer, es la de recordar que este servicio será utilizado únicamente por las implementaciones de puntos de venta propietarios del consorcio. Dada la aclaración, se tiene en cuenta que la tecnología elegida produce un alto nivel de acoplamiento entre el software cliente y el DBMS, pero prevalece el requerimiento de velocidad de respuesta. De todas formas, se implementará también una opción con menor acoplamiento para clientes que no tengan posibilidad de acceder por este servicio.



Vemos en el gráfico como los clientes propios de la plataforma se conectarán directamente.

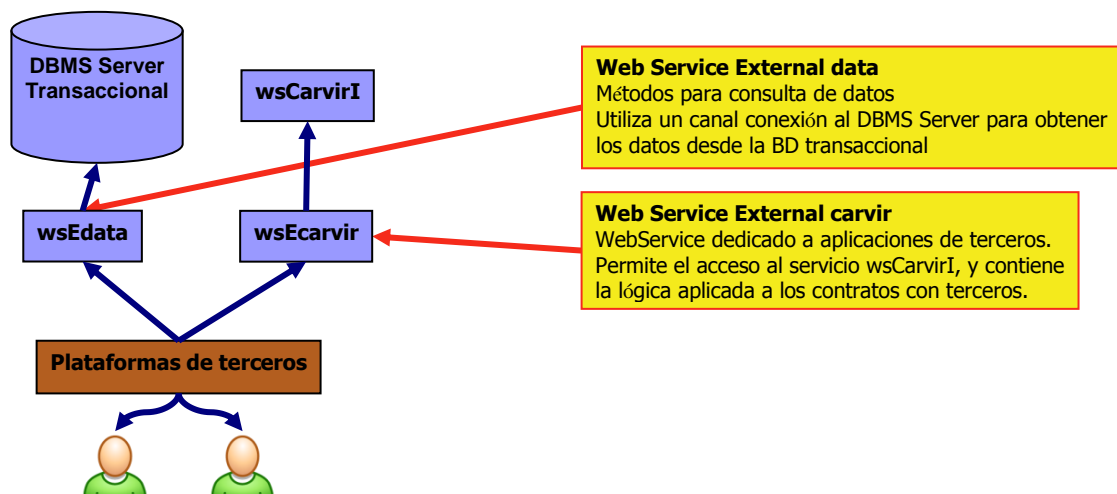
Servicio red de terceros

Este caso es similar al servicio de la venta de carga virtual. Las tecnologías que brindan la funcionalidad necesaria son nuevamente RMI, DCOM, CORBA y WebServices.

Nuevamente la elegida es WebServices, principalmente porque permite estandarizar para los terceros el acceso, sea cual sea la tecnología con la que desee consumir servicios de la plataforma.

Como se había analizado, se implementarán dos WebServices, uno para permitir la venta del producto saldo virtual y otro para permitir la consulta de datos.

En este caso, ofrecemos el servicio de consulta a través de una tecnología que tiene un bajo acoplamiento, evitando así la dependencia generada con el servicio de consulta de datos en los clientes propios de la plataforma del consorcio.

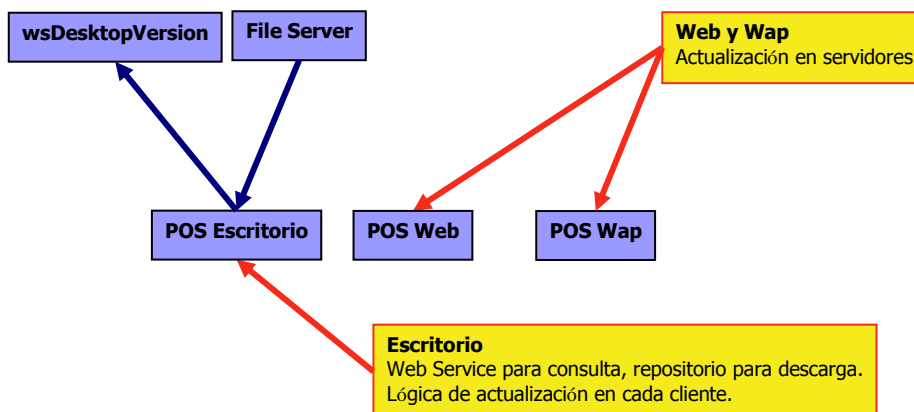


Servicio de actualización de versión

Otro de los requerimientos para la plataforma, es el poder realizar actualizaciones en el software en los puntos de venta, y que la misma sea desatendida y automática.

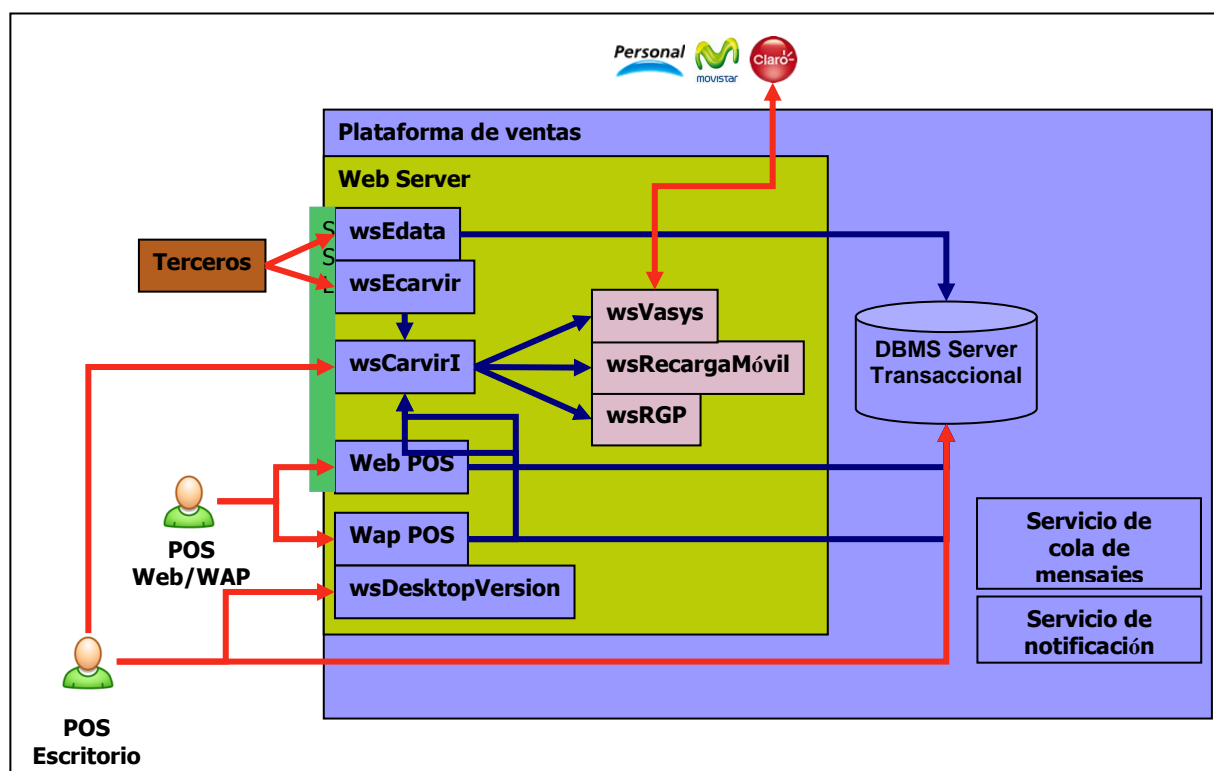
En las versiones web y wap esta necesidad está resuelta por la misma tecnología, ya que el cliente siempre accede a la última versión, que reside en el backend.

Para la versión de escritorio se utiliza un servicio web para publicar la versión actual, y en caso que cada POS cliente detecte que hay una nueva versión disponible, procederá a descargarla desde un repositorio.



5. Arquitectura

Como parte final de nuestro trabajo, presentaremos a continuación una visión general de la arquitectura de la plataforma, poniendo en contexto todos los servicios.



Pueden verse, además de todos los componentes que hemos analizado como necesarios, un par de servicios que pueden considerarse como servicios "to do", planificados como necesarios a futuro previendo que el sistema escale.

Servicio de cola de mensajes

Este servicio permite implementar el servicio de cargas en forma asíncrona. Hay períodos en los que el volumen de recargas es alto, y los sistemas de las Telcos no responden rápidamente. La cola de mensajes permite acumular los request y un proceso dedicado se encarga del procesamiento asíncrono.

Servicio de notificación

Permite la notificación de mensajes hacia los puntos de venta, a través de mails o mediante un servicio de mensajes interno a la plataforma de ventas. En el caso de mails, estamos hablando de un servicio de push desde el servidor, no periódico. El cliente, a través del POS, debe suscribirse a los eventos que desea (carga finalizada, asignación realizada, cierre diario, etc). El sistema de mensajería interna se implementa con un polling desde los clientes.

6. Conclusión

Se presentó inicialmente un requerimiento real de un grupo comercial, que fue interpretado para analizar una solución posible utilizando un sistema distribuido. Sumando a nuestra experiencia y conocimientos previos los adquiridos en las diferentes materias de la especialización, se identificaron los aspectos y condiciones que el sistema requiere para convertirse en una solución viable tanto desde el punto de vista tecnológico como el comercial y funcional.

Respetando esto, se prosiguió identificando en una primera instancia las partes o componentes lógicos del sistema que conforman el sistema distribuido, y sus interrelaciones, para culminar escogiendo la tecnología real más adecuada para la implementación de cada servicio, realizando la justificación correspondiente.

Finalmente, y poniendo todo en contexto, se esbozó una visión genérica de la arquitectura del sistema distribuido, que rápidamente permite captar el diseño propuesto.

Personalmente consideramos que este trabajo ha permitido consolidar, confirmar o poner en práctica, una gran parte de los temas tratados en las materias de la especialización, logrando de esta forma, el objetivo esperado.

Apéndice A. Comparación de tecnologías.

RPC. Remote Procedure Call

RMI (Remote Method Invocation)

Es la solución que propone SUN para implementar el paradigma de objetos distribuidos en Java, es muy eficiente, pero si se utiliza el protocolo propietario de JRMP (Java Remote Method Protocol) el desarrollo queda acoplado a la implementación, además de obligar al sistema operativo subyacente a poseer una interfaz de transporte del tipo TCP/IP. Con una implementación un tanto menos eficiente, pero con la ventaja de desacoplar tecnologías, RMI puede sustituir su protocolo de transporte por IIOP (Internet InterORB Protocol), que es el protocolo abierto propuesto por CORBA.

- + Mucha documentación y recursos.
- + Trabaja en diversas plataformas, tantas como posean JVM.
- + Varias herramientas de monitoreo y posibilidad de usar las interfaces de monitoreo de la JVM.
- + Lenguaje de definición de interfaces estándar Java-IDL compatible con IDL de CORBA.
- Altamente eficiente, si se utiliza JRMP.
- Propietario.

Web Services (WS): están destinados, principalmente, a resolver problemas de interconexión entre sistemas heterogéneos¹, creando repositorios, y promoviendo el reuso del código. XML es la forma en que se representan los datos, SOAP el protocolo que normaliza la representación de los mensajes transmitidos, y, a nuestro criterio, uno de los aspectos más importantes de los WS, WSDL es la interfaz pública, permite que los servicios sean identificados e invocados con el solo hecho de conocer el contrato², como último integrante del trío, no por ello menos relevante, se encuentra UDDI, permite la creación de registros estándares de servicios existentes, que podrán ser accedidos, tanto desde dentro, como desde fuera de una organización. Estos registros sirven para ubicar a los servicios disponibles y determinar el papel que cumplen. SOAP, WSDL y UDDI, comprenden la primera generación de los WS, la segunda generación, añade funcionalidades como: seguridad, direccionamiento, definición del proceso de negocios, etc.

- + Escaso acoplamiento
- + Independencia de la plataforma
- + Independencia de la tecnología de implementación
- + Independencia del modo de transporte (HTTP, HTTPS, HTTP-R, BEEP, JABBER, IIOP, SMTP o FTP, etc)
- + Múltiples modos de invocación. (estática o dinámica).

¹ En cuanto a la plataforma (Windows, Unix, Linux, etc.) y a las herramientas de desarrollo (Java, .NET, PHP, etc.)

² A los descriptores de servicio WSDL, también se los denomina contratos.

- + Múltiples estilos de comunicación. (comunicación síncrona (RPC) o asíncrona)
- + Extensibilidad. (Al estar basados en XML, son fáciles de adaptar, extender y personalizar).
- + Actualmente, en cuanto a su uso, son un estándar de facto.
- Seguridad (no desde el punto de vista de la viabilidad, es totalmente factible montar WS seguros, pero en muchos casos es muy complejo, sobre todo cuando existen workflows de WS)
- Performance: en muchos casos, el hecho de envolver mensajes en XML, luego en SOAP, para más tarde ser transportados y una vez en destino tener que realizar la tarea inversa, aumenta notablemente el overhead.

Distributed objects

DCOM (Distributed Component Object Model).

Solución propuesta por Microsoft para implementar una solución al paradigma de objetos distribuidos, está muy acoplado a tecnologías MS, el protocolo de transporte es propietario, aunque, mediante adaptadores puede compatibilizarse con IIOP. Los lenguajes de definición de interfaces son propietarios, pero compatibles con IDL CORBA.

- + Gran cantidad de recursos aportados a esta tecnología por parte de MS.
- + En sistemas MS es el más performante.
- + Herramientas de monitoreo muy potentes.
- Gran cantidad de "adaptadores" que permiten la comunicación e interoperación con sistemas parecidos, aunque la mayoría son cerrados.
- Solo para sistemas MS.
- Propietario.

CORBA (Common Object Request Broker Architecture)

Es una arquitectura abierta propuesta por la OMG (Object Management Group) creada con la finalidad de dar solución a los diversos problemas que acarrea el desarrollo de aplicaciones distribuídas, particularmente objetos distribuidos. Existen diversas implementaciones de CORBA, tanto comerciales como libres, con una variación sustancial en lo referente a precio, implementación de servicios y rendimiento. Un gran aporte de CORBA es la posibilidad de que diversas tecnologías de programación puedan interoperar y la implementación de objetos distribuidos permanecer agnóstica al resto de las tecnologías componentes. Mediante el lenguaje IDL (Interfaz Definition Language) pueden describirse las interfaces de los objetos a ser distribuidos.

- + Estandar abierto
- + Interoperabilidad de tecnologías
- Las implementaciones más performantes y con mejor soporte de implementación de servicios son propietarias y costosas.
- La implementación final es en muchos casos engorrosa.

- No se definen estándares de monitoreo.
- Incompatibilidades entre productos.

Database access and Object Relational Mapping (ORM)

Hibernate.

Es un framework ORM (Object Relational Mapper), permite mapear datos relacionales con objetos, lo realiza de manera transparente y performante. Hibernate permite realizar los mapeos de dos maneras, utilizando marcas en el código (anotaciones) que implementan el estándar JPA o bien creando descriptores XML de mapeo externos al código fuente. Este ORM no solo independiza al desarrollador del RDBMS, también lo hace respecto del lenguaje de programación, ya que está disponible para los principales lenguajes utilizados hoy en día. Actualmente da soporte al tipo XML. Ofrece un lenguaje de consulta llamado HQL, el interprete HQL genera código nativo SQL de muy alto rendimiento.

MS LINQ (Language Integrated Query)

Framework propietario de MS que introduce capacidades de consulta a diversas fuentes de datos en los lenguajes .NET. Permite realizar mapeos objeto-relacionales, objeto-XML y acceder a diversas fuentes de datos, entre otras funcionalidades.

- + Simple de usar
- + Acompañado de gran cantidad de herramientas gráficas
- + Lenguaje muy expresivo
- Produce alto overhead
- Solo plataforma MS
- No estándar
- Propietario