Universidad Católica Boliviana “San Pablo”

Departamento de Ingenierías



**Agro-Market Bolivia**

Nombres:

Garvizu Michell

Gerard Sebastian

Lara Sandra

Docente:

Samuel Lujan

*Fecha:*

21/01/2013

Cochabamba-Bolivia

INTRODUCCIÓN

La [computación](http://www.monografias.com/Computacion/index.shtml) desde sus inicios ha sufrido muchos cambios, desde los grandes ordenadores que permiten realizar tareas en forma limitada y de uso un tanto exclusivo de [organizaciones](http://www.monografias.com/trabajos6/napro/napro.shtml) muy selectas, hasta los actuales ordenadores ya sean personales o portátiles que tienen las mismas e incluso mayores capacidades que los primeros y que están cada vez más introducidos en el quehacer cotidiano de una [persona](http://www.monografias.com/trabajos7/perde/perde.shtml).

La computación distribuida o informática en malla, es un nuevo modelo para resolver problemas de computación masiva utilizando un gran número de [ordenadores](http://es.wikipedia.org/wiki/Ordenador) organizadas en [racimos](http://es.wikipedia.org/wiki/Cluster_(inform%C3%A1tica)) incrustados en una infraestructura de telecomunicaciones distribuida.

Los sistemas distribuidos deben ser muy confiables, ya que si un componente del sistema se descompone otro componente debe ser capaz de reemplazarlo, esto se denomina *tolerancia a fallos*.

El tamaño de un [sistema](http://es.wikipedia.org/wiki/Sistema) distribuido puede ser muy variado, ya sean decenas de hosts ([red de área local](http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_local)), centenas de hosts ([red de área metropolitana](http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_metropolitana)), y miles o millones de hosts ([Internet](http://es.wikipedia.org/wiki/Internet)); esto se denomina escalabilidad.

Los sistemas distribuidos son la solución a los problemas de muchas empresas, ya que se puede interconectar varios equipos y poder manejar una sola información.

Las empresas actuales dependen del buen manejo de la información y requieren una gran capacidad competitiva a través de la renovación, interacción y cumplimiento con el cliente, entre ellas se encuentra AGRO-MARKET Bolivia, ONG encargada de buscar mercados para los productos agropecuarios en Bolivia. Las empresas interesadas en los productos presentan ofertas de las cuales AGRO-MARKET selecciona las más convenientes. Para obtener las ofertas de productos de sus afiliados, AGRO-MARKET consulta en servicios implementados por cada uno de sus afiliados y a su vez los oferta al mercado. Las empresas interesadas obtienen la información de los servicios publicados ARKET y realizan también sus ofertas a través de dichos servicios. AGRO-MARKET los evaluará e informará a las empresas ganadoras.

1. OBJETIVOS
   * Recopilar información tecnológica respecto a las diferentes plataformas para el desarrollo de tecnologías Web.
   * Evaluar las características tecnológicas y presupuestarias de cada plataforma.
   * Escoger una plataforma de desarrollo de tecnologías Web que sea adecuada a los requerimientos del cliente “Agro-Market” que cumpla con las características necesarias para una buena implementación y que sea de poco presupuesto.
   * Implementar los servicios requeridos.
2. MARCO TEÓRICO

La computación distribuida ha sido diseñada para resolver problemas demasiado grandes para cualquier [supercomputadora](http://es.wikipedia.org/wiki/Supercomputadora) y mainframe, mientras se mantiene la flexibilidad de trabajar en múltiples problemas más pequeños.

Un sistema distribuido se define como: una colección de computadoras separadas físicamente y conectadas entre sí por una red de comunicaciones distribuida; cada máquina posee sus componentes de hardware y software que el usuario percibe como un solo sistema (no necesita saber qué cosas están en qué máquinas). El usuario accede a los recursos remotos de la misma manera en que accede a recursos locales, o un grupo de computadores que usan un software para conseguir un objetivo en común.

**SOAP**

SOAP fue diseñado por David Winer, Don Box, Bob Atkinson y Mohsen Al-Ghosein en 1998 con respaldo de [Microsoft.](http://www.alegsa.com.ar/Dic/microsoft.php) Actualmente, la especificación SOAP es mantenida por el XML Protocol Working Group de la W3C

SOAP es un [protocolo](http://www.alegsa.com.ar/Dic/protocolo.php) para el intercambio de mensajes sobre [redes de computadoras,](http://www.alegsa.com.ar/Dic/red%20de%20computadoras.php) generalmente usando [HTTP.](http://www.alegsa.com.ar/Dic/http.php) Está basado en [XML,](http://www.alegsa.com.ar/Dic/xml.php) a diferencia de DCOM y CORBA que son binarios; esto facilita la lectura por parte de los humanos, pero también los mensajes resultan más largos y, por lo tanto, considerablemente más lentos de [transferir.](http://www.alegsa.com.ar/Dic/transferencia.php) Existen múltiples tipos de modelos de mensajes en SOAP pero, por lejos, el más común es el [RPC,](http://www.alegsa.com.ar/Dic/rpc.php) en donde un [nodo](http://www.alegsa.com.ar/Dic/nodo.php) de red (el cliente) envía un mensaje de solicitud a otro nodo (el [servidor)](http://www.alegsa.com.ar/Dic/servidor.php)y el servidor inmediatamente responde el mensaje al cliente.Los mensajes SOAP, son independientes del [sistema operativo,](http://www.alegsa.com.ar/Dic/sistema%20operativo.php) y pueden transportarse en varios protocolos de [internet](http://www.alegsa.com.ar/Dic/internet.php) como [SMTP,](http://www.alegsa.com.ar/Dic/smtp.php) [MIME](http://www.alegsa.com.ar/Dic/mime.php) y [HTTP.](http://www.alegsa.com.ar/Dic/http.php) SOAP al principio significaba Simple Object Access Protocol, luego fue Service Oriented Architecture Protocol, pero actualmente es simplemente SOAP.

**RESTFUL**

[RESTful](http://es.wikipedia.org/wiki/Representational_State_Transfer) es una propuesta muy interesante de [Roy Fielding](http://es.wikipedia.org/wiki/Roy_Fielding) que permite manejar los servicios web con métodos definidos, manteniendo la simpleza del protocolo como XML, pero que cada servicio sea identificado únicamente con un solo [URI](http://es.wikipedia.org/wiki/Uniform_Resource_Identifier).

Una implementación concreta de un servicio web REST sigue cuatro principios de diseño fundamentales:

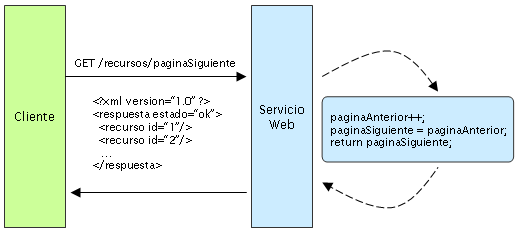
* Utiliza los métodos HTTP de manera explícita

REST hace que los desarrolladores usen los métodos HTTP explícitamente de manera que resulte consistente con la definición del protocolo. Este principio de diseño básico establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP. De acuerdo a esta asociación:

* Se usa POST para crear un recurso en el servidor
* Se usa GET para obtener un recurso
* Se usa PUT para cambiar el estado de un recurso o actualizarlo
* Se usa DELETE para eliminar un recurso
* No mantiene estado

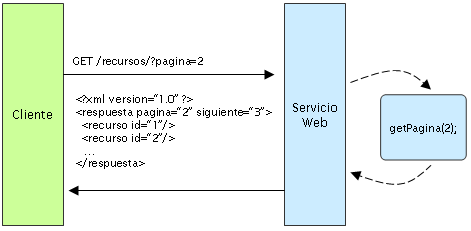
#### **Servicios con estado vs. sin estado**

La siguiente ilustración nos muestra un servicio con estado, del cual una aplicación realiza peticiones para la página siguiente en un conjunto de resultados múlti-página, asumiendo que el servicio mantiene información sobre la última página que pidió el cliente. En un diseño con estado, el servicio incrementa y almacena en algún lugar una variable pagina anterior para poder responder a las peticiones siguientes.



Los servicios con estado tienden a volverse complicados. En la plataforma Java Enterprise Edition (Java EE), un entorno de servicios con estado necesita bastante análisis y diseño desde el inicio para poder almacenar los datos eficientemente y poder sincronizar la sesión del cliente dentro de un cluster de servidores. En este tipo de ambientes, ocurre un problema que le resulta familiar a los desarrolladores de servlets/JSP y EJB, quienes a menudo tienen que revolver buscando la causa de una java.io.NotSerializableException cuando ocurre la replicación de una sesión. Puede ocurrir tanto sea en el contenedor de Servlets al intentar replicar la HttpSession o por el contenedor de EJB al replicar un EJB con estado; en todos los casos, es un problema que puede costar mucho esfuerzo resolver, buscando el objeto que no implementa Serializable dentro de un grafo complejo de objetos que constituyen el estado del servidor. Además, la sincronización de sesiones es costosa en procesamiento, lo que impacta negativamente en el rendimiento general del servidor.

Por otro lado, los servicios sin estado son mucho más simples de diseñar, escribir y distribuir a través de múltiples servidores. Un servicio sin estado no sólo funciona mejor, sino que además mueve la responsabilidad de mantener el estado al cliente de la aplicación. En un servicio web REST, el servidor es responsable de generar las respuestas y proveer una interfaz que le permita al cliente mantener el estado de la aplicación por su cuenta. Por ejemplo, en el mismo ejemplo de una petición de datos en múltiples páginas, el cliente debería incluir el número de página a recuperar en vez de pedir "la siguiente", tal como se muestra en la siguiente figura:



Un servicio web sin estado genera una respuesta que se enlaza a la siguiente página del conjunto y le permite al cliente hacer todo lo que necesita para almacenar la página actual. Este aspecto del diseño de un servicio web REST puede descomponerse en dos conjuntos de responsabilidades, como una separación de alto nivel que clarifica cómo puede mantenerse un servicio sin estado.

* Expone URL’s con forma de directorios

Podemos también enumerar algunas guías generales más al momento de crear URIs para un servicio web REST:

* Ocultar la tecnología usada en el servidor que aparecería como extensión de archivos (.jsp, .php, .asp), de manera de poder portar la solución a otra tecnología sin cambiar las URI.
* Mantener todo en minúsculas.
* Sustituir los espacios con guiones o guiones bajos (uno u otro).
* Evitar el uso de strings de consulta.
* En vez de usar un 404 Not Found si la petición es una URI parcial, devolver una página o un recurso predeterminado como respuesta.

Las URI deberían ser estáticas de manera que cuando cambie el recurso o cambie la implementación del servicio, el enlace se mantenga igual. Esto permite que el cliente pueda generar "favoritos" o bookmarks. También es importante que la relación entre los recursos que está explícita en las URI se mantenga independiente de las relaciones que existen en el medio de almacenamiento del recurso.

* Transfiere XML, JavaScript Object Notation (JSON) o ambos

La última restricción al momento de diseñar un servicio web REST tiene que ver con el formato de los datos que la aplicación y el servicio intercambian en las peticiones/respuestas. Acá es donde realmente vale la pena mantener las cosas simples, legibles por humanos, y conectadas.

**SOA**

La arquitectura orientada a servicios de cliente (en inglés **S**ervice **O**riented **A**rchitecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

Permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios (comúnmente pero no exclusivamente [servicios web](http://es.wikipedia.org/wiki/Servicio_Web)), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

SOA define las siguientes capas de software:

* **Aplicaciones básicas** - Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad;
* **De exposición de funcionalidades** - Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (generalmente como servicios web);
* **De integración de servicios** - Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración;
* **De composición de procesos** - Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio;
* **De entrega** - donde los servicios son desplegados a los usuarios finales.

|  |  |
| --- | --- |
| Término | Definición / Comentario |
| Servicio | Una función sin estado, auto-contenida, que acepta una(s) llamada(s) y devuelve una(s) respuesta(s) mediante una interfaz bien definida. Los servicios pueden también ejecutar unidades discretas de trabajo como serían editar y procesar una transacción. Los servicios no dependen del estado de otras funciones o procesos. |
| Orquestación | Secuenciar los servicios y proveer la lógica adicional para procesar datos. No incluye la presentación de los datos. Coordinación. |
| Sin estado | En una SOA los servicios no son dependientes de la condición de ningún otro servicio. Reciben en la llamada toda la información que necesitan para dar una respuesta. |
| Proveedor | La función que brinda un servicio en respuesta a una llamada o petición desde un consumidor. |
| Consumidor | La función que consume el resultado del servicio provisto por un proveedor |

1. PLATAFORMA TECNOLÓGICA

Existen muchas plataformas para el desarrollo de servicios web de las cuales se destacan las siguientes:

* [**JBoss**](http://es.wikipedia.org/wiki/JBoss)

Servidor de aplicaciones J2EE Open Source de [Red Hat](http://es.wikipedia.org/wiki/Red_Hat), está orientada a servicios con licencia de código abierto que permite desarrollar, desplegar, organizar, presentar aplicaciones y servicios web en una arquitectura orientada a los servicios mediante un servidor de aplicaciones Open Source implementado en java puro.

Las características destacadas de JBoss incluye :

* Producto de licencia de código abierto sin costo.
* Cumple los estándares.
* Confiable a nivel de empresa
* Incrustable, orientado a arquitectura de servicios.
* Soporte completo para [JMX](http://es.wikipedia.org/wiki/JMX)

Ventajas:

* + Middleware es sencillo, abierta y accesible de implementar arquitecturas Orientadas a los servicios.
  + JBoss Hibernate es para mapeo y persistencia de datos relacionales.
  + JBoss Seam simplifica el desarrollo de aplicaciones web 2.0
  + JBoss jBPM permite crear y automatizar procesos de negocio.
  + JBoss Rules para fácil acceso, modificación y gestión de políticas
  + Amplio servicio técnico.
  + Acceso al producto: Código fuente y binario.
  + Actualizaciones regulares que ofrecen mejoras generales, nuevas características, entre otros.
  + Revisiones y acceso a las últimas correcciones de fallas y errores de seguridad.
  + Mejoras en los derechos a las nuevas versiones de JBoss Enterprise Middleware lanzadas durante el plazo de la suscripción.
  + Flexibilidad: Las suscripciones no son específicas para una versión, por lo tanto incluyen el acceso a versiones anteriores y el soporte correspondiente.
* [**Oracle Fusion Middleware**](http://es.wikipedia.org/w/index.php?title=Oracle_Fusion_Middleware&action=edit&redlink=1)

Es una plataforma tecnológica que puede simplificar todas las capas de TI con el fin de la innovar.

Ventajas:

* Optimizar los procesos críticos y la administración de TI en tiempos de incertidumbre económica.
* Mejorar los conocimientos de negocio y administrar el desempeño.
* Mitigar el riesgo y asegurar la información corporativa.
* Impulsar la productividad y brindar mayor capacidad a los usuarios.
* [**IBM Lotus Domin**](http://es.wikipedia.org/w/index.php?title=IBM_Lotus_Domino&action=edit&redlink=1)**o versión 7.0**

Lotus Notes es un sistema cliente/servidor de colaboración y [correo electrónico](http://es.wikipedia.org/wiki/Correo_electr%C3%B3nico), desarrollado por [Lotus Software](http://es.wikipedia.org/wiki/Lotus_Software), de [IBM](http://es.wikipedia.org/wiki/IBM), la parte del [servidor](http://es.wikipedia.org/wiki/Servidor) recibe el nombre *Lotus Domino*, mientras que el [cliente](http://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)) se llama *Lotus Notes*.

Lotus Domino/ Notes es un sistema de comunicación el cual permite enviar correo electrónico y manejo de Calendarios y Agendas, también es una plataforma de [colaboración](http://es.wikipedia.org/wiki/Colaboraci%C3%B3n) que permite compartir bases de datos con información, como sería bases documentales, de procedimientos, manuales o foros de discusión. Y finalmente es una plataforma de [Coordinación](http://es.wikipedia.org/wiki/Coordinaci%C3%B3n) - utilizando aplicaciones Notes con [flujo de trabajo](http://es.wikipedia.org/wiki/Flujo_de_trabajo). En cualquier proceso de una empresa que requiere que un documento fluya entre varias personas o departamentos para su autorización.

La plataforma IBM Lotus Notes/Domino está integrada:

1. Lotus Domino - servidor
2. Lotus Notes - cliente de correo y colaboración
3. Domino Administrator - cliente para administrar el servidor Domino
4. Domino Designer - ambiente integrado de desarrollo (IDE) para crear aplicaciones

Ventajas:

* Es muy potente, completo, versátil, fácil de usar.
* Rapidez para crear complejos sistemas de comunicación y es muy potente para la gestión de base de datos y [gestión del conocimiento](http://es.wikipedia.org/wiki/Gesti%C3%B3n_del_conocimiento).
* Soporta [Servicios Web](http://es.wikipedia.org/wiki/Servicios_Web) y el repositorio de datos pueden ser bases de datos NSF o [DB2](http://es.wikipedia.org/wiki/DB2)..
* [**Axis**](http://es.wikipedia.org/w/index.php?title=Axis_(computer_program)&action=edit&redlink=1) **y el servidor** [**Jakarta Tomcat**](http://es.wikipedia.org/wiki/Jakarta_Tomcat) **(de Apache)**

Es un [servidor web](http://es.wikipedia.org/wiki/Servidor_web) con soporte de [servlets](http://es.wikipedia.org/wiki/Servlets) y JSPs. Tomcat no es un [servidor de aplicaciones](http://es.wikipedia.org/wiki/Servidor_de_aplicaciones), como [JBoss](http://es.wikipedia.org/wiki/JBoss) o [JOnAS](http://es.wikipedia.org/wiki/JOnAS). Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el [servidor web Apache](http://es.wikipedia.org/wiki/Servidor_web_Apache).

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

Ventajas:

* Producto de licencia gratuito
* Es muy sencillo de configurar y dispone de gran documentación.
* [**ColdFusion**](http://es.wikipedia.org/wiki/ColdFusion) **MX**

Coldfusion (Adobe ColdFusion) es un [servidor de aplicaciones](http://es.wikipedia.org/wiki/Servidor_de_aplicaciones) y un [lenguaje de programación](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) usado para desarrollar aplicaciones de i[nternet](http://es.wikipedia.org/wiki/Internet), generalmente sitios web generados [dinámicamente](http://es.wikipedia.org/wiki/Contenido_din%C3%A1mico). En este aspecto, es un producto similar a [ASP](http://es.wikipedia.org/wiki/Active_Server_Pages), [JSP](http://es.wikipedia.org/wiki/JavaServer_Pages) o [PHP](http://es.wikipedia.org/wiki/PHP).

ColdFusion es una herramienta que corre en forma [concurrente](http://es.wikipedia.org/w/index.php?title=Concurrencia&action=edit&redlink=1) con la mayoría de los [servidores web](http://es.wikipedia.org/wiki/Servidores_web) de [Windows](http://es.wikipedia.org/wiki/Windows), [Mac OS X](http://es.wikipedia.org/wiki/Mac_OS_X), [Linux](http://es.wikipedia.org/wiki/Linux) y [Solaris](http://es.wikipedia.org/wiki/Solaris_(sistema_operativo)). El servidor de aplicaciones web de ColdFusion trabaja con el servidor [HTTP](http://es.wikipedia.org/wiki/HTTP) para procesar peticiones de páginas web, una vez que se solicita una página de ColdFusion, el servidor de aplicaciones ColdFusion ejecuta el [guión](http://www.google.com/url?q=http%3A%2F%2Fes.wikipedia.org%2Fwiki%2FScript&sa=D&sntz=1&usg=AFQjCNGX8Nu0tNhjt-dZy1d3eJ0cM7O1Fg) o programa contenido en la página.

Ventajas:

* + Mejora la productividad por el lenguaje de scripting del servidor basado en tags, aplicado solo en aplicaciones web.
  + Permite a los equipos de desarrollo colaborar compartiendo el mismo servidor y trabajando local o remoto.
  + Acelera el desarrollo con un conjunto de herramientas de diseño, programación, implantación.
  + Puede ser usado en un sitio cada vez que se necesita interactuar con el usuario. Se puede usar para construir diarios murales, clientes POP, salas de chat.
  + Permite escribir scripts para rastrear estadísticas.
  + Permite usar menos líneas y son intuitivas para los resultados, y deja al usuario migrar a otros servidores web y motores de base de datos con pocos cambios sin plug-ins externos.
  + Genera y envía javascript transparentemente on the fly cuando ciertos tags de input son utilizados, esto facilita por el lado del cliente ver los inputs sin forzar al programador a escribir, revisar y modificar javascript para hacer eso.
  + Viene con un motor para indexar sitios web. Se puede realizar balanceo de carga dinamico, tambien pueden mantener porciones de código como propietarios.
  + ColdFusion ahorra dinero en mantenimiento.
* [**Java Web Services Development Pack**](http://es.wikipedia.org/wiki/Java_Web_Services_Development_Pack) **(JWSDP)**

Es el paquete de desarrollo de servicios web de java basado en jakarta Tomcat, es un kit libre de desarrollo de software para desarrollar servicios web, aplicaciones web y aplicaciones java con las nuevas tecnologías java. Actualmente JWSDP fue sustituido por GlassFish y su código abierto compatible con aplicaciones Java EE 5. El código fuente está disponible bajo la licencia CDDL, la cual está aprobada por la Open Source Initiative.

Los componentes y APIs disponibles en el JWSDP 1.6 son:

* Java API for XML Processing (JAXP)
* Java Architecture for XML Binding (JAXB)
* JAX-RPC
* JAX-WS
* SAAj
* Java API for XML Registries
* Web Services Registry
* [**JOnAS**](http://es.wikipedia.org/wiki/JOnAS)(parte de *ObjectWeb* una iniciativa de código abierto)

Es un servidor de aplicaciones J2EE de código abierto implementado en Java. Jonas forma parte de la iniciativa de código abierto de ObjectWeb, la cual fue lanzada en colaboración con varios socios, entre los que se encuentran Bull, France Télécom e INRIA.

JOnAS se libera bajo licencia de código abierto LGPL.

Establecido por Sun para organizaciones sin fines de lucro. JOnAS es considerado tecnológicamente superior debido a la performance y escalabilidad de las aplicaciones de EJB. La nueva arquitectura de JOnAS 5 se basa en gran parte en un marco de OSGi que hace mucho más fácil que antes el desarrollo del componente dinámico.

JOnAs tiene una consola de administración remota relativamente clara, proporcionando herramientas fácilmente accesibles para carga e implementación remota de aplicaciones web, configuración de puertos de acceso web seguros y ordinarios, y gestionar los conjuntos de datos usuario nombre - contraseña -rol. Esta consola se basa en el código JavaScript.

JOnAS es un servidor de aplicaciones Java EE 5 de código abierto. Está disponible con un contenedor web incrustado Tomcat o Jetty, además que proporciona un contenedor EJB totalmente compatible mediante EasyBeans. JOnAS se puede ejecutar en numerosos sistemas operativos como Linux, Windows, AIX, muchas plataformas POSIX y otros, siempre que esté disponible una máquina virtual Java.

* [**Microsoft .NET**](http://es.wikipedia.org/wiki/Microsoft_.NET)

WSE 3.0 simplifica el desarrollo e implementación de servicios Web seguros. Permite a desarrolladores y administradores aplicar políticas de seguridad a los servicios Web en Framework 2.0 usando WSE 3.0, la comunicación de servicios Web puede ser suscrita y encriptada usando Kerberos, certificados X.509 y otra seguridad personalizada como ser XML-based. Algunas características son:

* Seguridad simplificada: intercambio de mensajes SOAP entre cliente y servicio.
* Envío de una gran cantidad de datos binarios: provee la capacidad de enviar gran cantidad de datos binarios eficientemente a través de la especificación W3C SOAP MTOM (Message Transmission Optimization Mechanism).
* Características de seguridad adicionales: confirmación de la firma, la capacidad de crear explícitamente clave encriptada.
* Integrado con Visual Studio 2005: WSE 3.0 provee soporte para ASP.NET Web Services. Los protocolos TCP, UDP y SMTP son soportados.
* WSE 3.0 soporta plataformas Windows de 64 bits.
* Interoperabilidad con Windows Communication Foundation(WCF): WSE 3.0 está diseñado con el mismo set de especificaciones de seguridad que WCF por lo que son compatibles.
* Cumplimiento de normas: WSE provee una fundación para la construcción de aplicaciones basadas en especificaciones de servicios Web publicada por Microsoft y compañeros de la industria incluyendo WS-Security 1.0 y 1.1 WS-Trust, WS-SecureConversation, Ws-Addressing y MTOM.
* Soporte de plataformas: WSE 3.0 fue construido para desarrolladores usando Visual Studio 2005 y .NET Framework 2.0 y tiene una política de soporte de por vida
* [**Novell exteNd**](http://es.wikipedia.org/w/index.php?title=Novell_exteNd&action=edit&redlink=1)(basado en la plataforma [J2EE](http://es.wikipedia.org/wiki/J2EE))

Es una suite completa para el rápido desarrollo y despliegue de aplicaciones Web orientadas a servicios. Con Novell exteNd 5 desarrolladores de aplicaciones corporativas, pueden combinar los servicios de identidad, integración y proporcionar información de manera segura. ExteNd 5 incluye:

1. Novell exteNd Composer es un servidor de integración excepcional que se conecta a los sistemas existentes y presenta la información como servicios web que son fácilmente consumidos en otros sistemas y aplicación. Las barreras de aplicación a la información se eliminan y los datos son de fácil acceso.
2. Novell exteNd Director es un servidor de interacción intuitiva que proporciona herramientas visuales para consumir servicios rápidamente y presentar información. Con un portal hacia fuera de la caja, motor de reglas, transcodificación, administración de contenido, flujo de trabajo y la información puede ser fácilmente capturada, administrada y presentada.
3. Novell exteNd Application Server es un servidor de aplicaciones J2EE certificadas que es la base de alto rendimiento, fiable y escalable para exteNd 5.

* [**WebLogic**](http://es.wikipedia.org/w/index.php?title=BEA_Weblogic&action=edit&redlink=1)

Es un servidor de aplicaciones para la nube y entornos convencionales. Provee una tal performance y escalabilidad, un desarrollo y administración dramáticamente simplificado y acelera el tiempo de comercialización con una moderna y ligera plataforma de desarrollo.

Proporciona flexibilidad entre nube públic a y nube privada, y se integra con Oracle Exalofic Elastic Cloud para un rendimiento, confiabilidad y la escalabilidad. A medida que la piedra angular de la nube de Oracle Aplication Fundation, Oracle WebLogicServer 12 aprovecha al máximo las últimas arquitecturas de hardware y proporciona soporte completo para Java EE 6. Con Oracle WebLogic, puede superar a sus competidores, tendrá plena autoridad y flexibilidad, y reducir el coste total de propiedad

* [**WebSphere**](http://es.wikipedia.org/wiki/WebSphere)

WebSphere es una familia de productos de [software privado](http://es.wikipedia.org/wiki/Software_privado) de [IBM](http://es.wikipedia.org/wiki/IBM), aunque el término se refiere de manera popular a uno de sus productos específicos: [WebSphere Application Server](http://es.wikipedia.org/wiki/WebSphere_Application_Server) (WAS). WebSphere ayudó a definir la categoría de software [middleware](http://es.wikipedia.org/wiki/Middleware) y es una plataforma modular basada en estándares abiertos soportados por la industria. Puede conectar sus activos ya existentes a WebSphere a través de interfaces confiables y perdurables; también puede seguir ampliando su entorno a medida que las necesidades aumentan. WebSphere funciona en varias plataformas, como AIX, HP Unix, i5/OS, Linux, Sun Solaris, Windows y z/OS.

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 monitorear dichos procesos (WebSphere Process Server, WebSphere Monitor)

[CICS Transaction Server](http://www.ibm.com/software/htp/cics/tserver/index.html) actualmente es la base de la mayoría de las aplicaciones de sistema principal y soporta el desarrollo de aplicaciones en lenguajes muy utilizados, como COBOL, PL/I, C/C++ y Java.

* [**JAX-WS**](http://es.wikipedia.org/wiki/JAX-WS)(Java API for XML Web Services)

Es una [API](http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones) de [Java](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)) con GlassFish para la creación de [servicios web](http://es.wikipedia.org/wiki/Servicio_web). Es parte de la plataforma [Java EE](http://es.wikipedia.org/wiki/Java_EE) de [Sun Microsystems](http://es.wikipedia.org/wiki/Sun_Microsystems). Al igual que las otras APIs de Java EE, JAX-WS utiliza [anotaciones](http://es.wikipedia.org/wiki/Anotaci%C3%B3n_Java), introducidas en [Java SE](http://es.wikipedia.org/wiki/Java_SE) 5, para simplificar el desarrollo y despliegue de los clientes y puntos finales de servicios web. Es parte del [Java Web Services Development Pack](http://es.wikipedia.org/wiki/Java_Web_Services_Development_Pack).

La [implementación de referencia](http://es.wikipedia.org/w/index.php?title=Implementaci%C3%B3n_de_referencia&action=edit&redlink=1) de JAX-WS se desarrolla como un proyecto de [código abierto](http://es.wikipedia.org/wiki/C%C3%B3digo_abierto) y forma parte del proyecto [GlassFish](http://es.wikipedia.org/wiki/GlassFish), un servidor de aplicaciones Java EE de código abierto. Se llama JAX-WS y se dice que es la implementación de calidad de producción (contrariamente a la implementación de referencia antigua que era una prueba de concepto)

GlassFish es un [servidor de aplicaciones](http://es.wikipedia.org/wiki/Servidor_de_aplicaciones) de software libre desarrollado por [Sun Microsystems](http://es.wikipedia.org/wiki/Sun_Microsystems), compañía adquirida por [Oracle Corporation](http://es.wikipedia.org/wiki/Oracle_Corporation), que implementa las tecnologías definidas en la plataforma [Java EE](http://es.wikipedia.org/wiki/Java_EE) y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada Oracle GlassFish Enterprise Server. Es gratuito y de [código libre](http://es.wikipedia.org/wiki/C%C3%B3digo_libre), se distribuye bajo un licenciamiento dual a través de la licencia [CDDL](http://es.wikipedia.org/wiki/Common_Development_and_Distribution_License) y la [GNU GPL](http://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_de_GNU).

GlassFish está basado en el código fuente donado por Sun y [Oracle Corporation](http://es.wikipedia.org/wiki/Oracle_Corporation), éste último proporcionó el módulo de persistencia [*TopLink*](http://www.oracle.com/technology/products/ias/toplink/index.html). GlassFish tiene como base al servidor *Sun Java System Application Server* de [Oracle Corporation](http://es.wikipedia.org/wiki/Oracle_Corporation), un derivado de [Apache Tomcat](http://es.wikipedia.org/wiki/Apache_Tomcat), y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.

* [**Zope**](http://es.wikipedia.org/wiki/Zope)

Zope es un [servidor de aplicaciones](http://es.wikipedia.org/wiki/Servidor_de_aplicaciones) web de código abierto escrito en el [lenguaje de programación](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [Python](http://es.wikipedia.org/wiki/Python). Para las funciones de edición de contenidos, así como personalizaciones básicas, puede ser usado mediante un navegador web. La programación avanzada así como el desarrollo de nuevas funcionalidades requiere la edición de componentes en «file system».

Un [sitio web](http://es.wikipedia.org/wiki/Sitio_web) de Zope está compuesto de objetos en lugar de archivos, como es usual con otros sistemas de [servidores web](http://es.wikipedia.org/wiki/Servidor_web).

Las ventajas de usar objetos en lugar de archivos son:

* + Combinan el comportamiento y los datos en una forma más natural que los archivos de texto plano.
  + Alientan el uso de componentes estándares que se ocupan de una parte particular de las que forman una [aplicación Web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_Web), permitiendo flexibilidad y buena descomposición.
  + Posibilitan procesos automáticos de gestión de información.

Lo más característico de Zope es su [base de datos](http://es.wikipedia.org/wiki/Base_de_datos) orientada a objetos, llamada [ZODB](http://es.wikipedia.org/wiki/ZODB) o [Zope Object Database](http://es.wikipedia.org/wiki/Zope_Object_Database). Esta base de datos almacena objetos ordenados en un sistema similar a un [sistema de ficheros](http://es.wikipedia.org/wiki/Sistema_de_ficheros), pero cada objeto tiene propiedades, [métodos](http://es.wikipedia.org/wiki/M%C3%A9todo_(programaci%C3%B3n_orientada_a_objetos)) u otros [objetos](http://es.wikipedia.org/wiki/Objetos_(programaci%C3%B3n_orientada_a_objetos)). Esta aproximación es muy diferente de las base de datos relacionales habituales. Sin embargo, Zope dispone de múltiples conectores para las diferentes bases de datos relacionales y ofrece sistemas básicos de conexión y consulta abstrayéndose como objetos.

Actualmente existen dos ramas principales, zope2 y zope3, este último es una re- implementación del servidor zope, donde se ha tratado de volcar toda la experiencia adquirida en zope2. Zope3 no trae compatibilidad hacia atrás, por lo que los componentes hechos para zope2 no funcionan. Aún se está en un proceso de adaptación hacia este nuevo zope, para lo cual está usando un componente llamado five, con el cual desde zope2 pueden tener la facilidad de zope3.

* [**VERASTREAM**](http://es.wikipedia.org/w/index.php?title=VERASTREAM&action=edit&redlink=1)

De AttachmateWRQ para modernizar o integrar aplicaciones host IBM y VT, encapsula los datos del mainframe y la lógica a través de la interfaz de la aplicación mediante la exposición de los procesos de negocio como servicios web, XML, Java o componentes. NET. Se integra con [IBM System z](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/IBM_System_z&usg=ALkJrhivWuVoBIMoFxzIuLJVAnvbVMUWow) ( [S/390](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/S/390&usg=ALkJrhizAcIbIobP26PZqrqxfGCh3Xn8Ag) ), [IBM System i](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/IBM_System_i&usg=ALkJrhjiItxtsc8Yy-FUMmj2ICx5rbqBSA) ( [AS/400](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/AS/400&usg=ALkJrhiwuJu_tF9mrEgx_oRvKPkUtylqwA) ), 420-VT100, [OpenVMS](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/OpenVMS&usg=ALkJrhj4V6Vt3NXwgmQ8_KVLPB_TMXP_KA) y [HP 3000](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/HP_3000&usg=ALkJrhiPJ_Bs_9_qkUacOBW4fiMJ2bpo-A) sin molestar aplicaciones mainframe aplicación de código o de las operaciones diarias del negocio.

Lista de las características técnicas de Verastream Host Integrator:

* Extender la funcionalidad de host mainframe y empresa para su uso en web, portales, SOA, CRM, mobile, centros de contacto, soluciones web o de autoservicio.
* Point-and-click proceso a la funcionalidad legado resumen en forma de componentes
* Crea automáticamente interfaces de componentes como [JavaBeans](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/JavaBeans&usg=ALkJrhiIxP4_sopvffyxL3A52YY327FbsA) , [.NET](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/.NET_Framework&usg=ALkJrhiJOlsTJidheIsVYuFssw5FpF2LMw) bibliotecas de clases, o [servicios web](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/Web_services&usg=ALkJrhifV0qUXZicHfQna-0TEJQ-E5pghA) ( [WS-I](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/WS-I&usg=ALkJrhjQmeA9rw9kEQygqODIDsvtFzfvLg) compatible)
* Genera [ASP.NET](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/ASP.NET&usg=ALkJrhjwPB8sdLEAlaPdfD6WvEE2DXu-JQ) o [JSP](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/JavaServer_Pages&usg=ALkJrhh8_HiPBHPZgRrk78mdPhT8CJeCKg) aplicaciones web
* Escalas de tiempo de ejecución a través de servidores múltiples en el equilibrio de carga y la configuración de conmutación por error
* Soporte de seguridad - [SSL](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/Transport_Layer_Security&usg=ALkJrhjIK60AhKIrloixVFJJgE_iiiIXpA) cifrado [FIPS 140-2](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/FIPS_140-2&usg=ALkJrhiOB6WOvPXcwTITUMb17PlLYIQ9lg) validación, [RACF](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/RACF&usg=ALkJrhgZayWkIngySaLetRXvRHZADOjbDA) , [ACF2](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/ACF2&usg=ALkJrhifEeanZNClAtAQtvfMKVk0AZpFUQ) , [LDAP.](http://translate.googleusercontent.com/translate_c?depth=1&ei=IHLsUI7NLLOq0AGO0IGQDQ&hl=es&prev=/search%3Fq%3DVERASTREAM%2B%2Bservidor%26hl%3Des%26tbo%3Dd%26biw%3D1366%26bih%3D587&rurl=translate.google.com.bo&sl=en&twu=1&u=http://en.wikipedia.org/wiki/LDAP&usg=ALkJrhjtRHJOx04WW3IirsaWvYX6AMLcug)
* La agrupación de sesiones - Sesiones de mainframe están preestablecidos en pantallas específicas, lo que elimina la navegación innecesaria y reduce la carga de trabajo de mainframe.
* [**PHP**](http://es.wikipedia.org/wiki/PHP)

PHP es un lenguaje de programación [de uso general](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general) de [script del lado del servidor](http://es.wikipedia.org/wiki/Script_del_lado_del_servidor) originalmente diseñado para el [desarrollo web](http://es.wikipedia.org/wiki/Desarrollo_web) de [contenido dinámico](http://es.wikipedia.org/wiki/Contenido_din%C3%A1mico). Fue uno de los primeros lenguajes de programación del lado del servidor que se podrán incorporar directamente en el documento [HTML](http://es.wikipedia.org/wiki/HTML) en lugar de llamar a un archivo externo que procese los datos.

El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de [línea de comandos](http://es.wikipedia.org/wiki/L%C3%ADnea_de_comandos) que puede ser usada en [aplicaciones gráficas](http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario) independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

1. **DESCRIPCIÓN DEL SISTEMA**

Nombre: Agro-Market Bolivia

Acrónimo: AMB

Sentencia de misión: Dar servicio de subasta de para productos agropecuarios

Responsabilidades:

* Registrar usuario
* Registrar producto
* Registrar oferta
* Editar usuario
* Ver usuario
* Ver producto
* Permite introducir precio al producto de dicha oferta
* El administrador elige la mejor oferta
* Comunica ganadores por correo electrónico

Exclusiones:

* Eliminar usuario
* Eliminar producto
* Historial de compras
* No permite registrar más de un administrador

4.1 Árbol de Refinamiento (ARF)

* + Soportar la administración de usuarios

- Crear perfil de Cliente

- Modificar perfil de Cliente/Administrador

- Buscar Cliente

* + Soportar el control de acceso de usuarios

- Iniciar Sesión

- Cerrar Sesión

* + Soportar la administración de las ofertas

- Hacer oferta

- Ver ofertas(Para administrador)

- Evaluar oferta(Para administrador)

* + Soporta la administración de los productos

- Registrar Producto

- Mostrar lista de Productos

- Mostrar lista de Productos pendientes

4.2 Descripción de servicios

* Crear perfil de Cliente

Evento: El cliente desea crear su perfil

Servicio: Registrar un nuevo perfil de cliente en el sistema

Suposiciones: El sistema no tiene un perfil registrado el usuario

* Modificar perfil de Cliente/Administrador

Evento: El cliente/administrador desea modificar su perfil

Servicio: Modifica los datos de perfil del cliente/administrador

Suposiciones: El sistema tiene registrado el perfil del cliente/administrador

* Buscar Cliente

Evento: Búsqueda del cliente por el nombre

Servicio: Muestra descripción del cliente buscado o resultados similares

Suposiciones: El cliente buscado esta registrado

* Iniciar sesión

Evento: El cliente solicita iniciar sesión

Servicio: Ingresar al perfil del cliente

Suposiciones: El sistema tiene registrado al cliente

* Cerrar sesión

Evento: El cliente solicita cerrar sesión

Servicio: Salir del perfil del cliente

Suposiciones: El sistema tiene activo el perfil del cliente

* Registrar Producto

Evento: El Cliente Registra un Producto

Servicio: Se registra y se valida el producto

Suposiciones: La fecha de oferta esta vigente

* Mostrar lista de Productos

Evento: El cliente solicita ver la lista productos

Servicio: Muestra la lista de productos que están abiertos a la subasta

Suposiciones:

- Que existan productos

- El cliente está conectado

- La fecha de expiración de los productos no han pasado

* Mostrar lista de Productos pendientes

Evento: El administrador solicita ver la lista productos

Servicio: Muestra la lista de todos los productos que no fueron evaluados por el administrador

Suposiciones:

- Que existan productos

- El administrador está conectado

* Hacer oferta

Evento: El cliente solicita hacer una oferta por un producto

Servicio: registra la oferta del cliente hacia el producto

Suposiciones:

- Cliente debe estar conectado

- La oferta esta con fecha de caducidad válida

- La oferta aún no tiene dueño

* Ver ofertas

Evento: El administrador solicita ver ofertas de un producto

Servicio: Mostrar la lista de ofertas del producto solicitado

Suposiciones:

- Un producto ha sido seleccionado

- El administradordebe estar logueado

- Existen ofertas registradas en el sistema

* Evaluar oferta

Evento: El administrador solicita evaluar las ofertas de un producto

Servicio: Se escoge un ganador(es) para el producto evaluado, enviando un correo electrónico al ganador(es) del producto por la mejor oferta.

Suposiciones:

- El administradordebe estar logueado

- Existe al menos una oferta para este producto.

1. SOLUCIÓN PROPUESTA

Para encontrar la mejor solución al problema es necesario descartar todas aquellas plataformas tecnológicas que no son gratuitas o impliquen un costo considerable para la empresa porque no cuenta con grandes recursos económicos, entre ellas se encuentran las siguientes:

- ColdFusion MX de Macromedia

- Microsoft .NET

- Novell exteNd

- WebLogic (Oracle)

- WebSphere

- VERASTREAM de AttachmateWRQ

De las plataformas que quedan, la que se escogió es JOnAS ya que es de código abierto, soporta múltiples plataformas (Microsoft, Linux, Unix, etc), pensado para organizaciones sin fines de lucro (ONG’s), provee un stack de ejecución completo Java EE 5, soporta EJB3 con OW2 EasyBeans container, Servlet 3.0 con Tomcat 7, JAX-WS 2.1 con CXF 2.2, JAX-RS 1.1 con Jersey, CDI CON Weld, etc.

Los usuarios y sus funciones del sistema se describen a continuación:

* Cliente (Empresa)
  + Sube / Registra su producto en sistema para que sea subastado
  + Hace un pedido de un producto (subasta)
  + Maneja cuentas (Loggin, Registro, Edición datos)
* Administrador:
  + Escoge las mejores ofertas(Ganadores de la subasta de un producto)

Por otra parte el sistema se encarga del envío de mensajes de correo electrónico comunicando que se vendió su producto / parte de él y a qué precio, que ganó la oferta y que la oferta caducó.

**Manejo y organización de código**

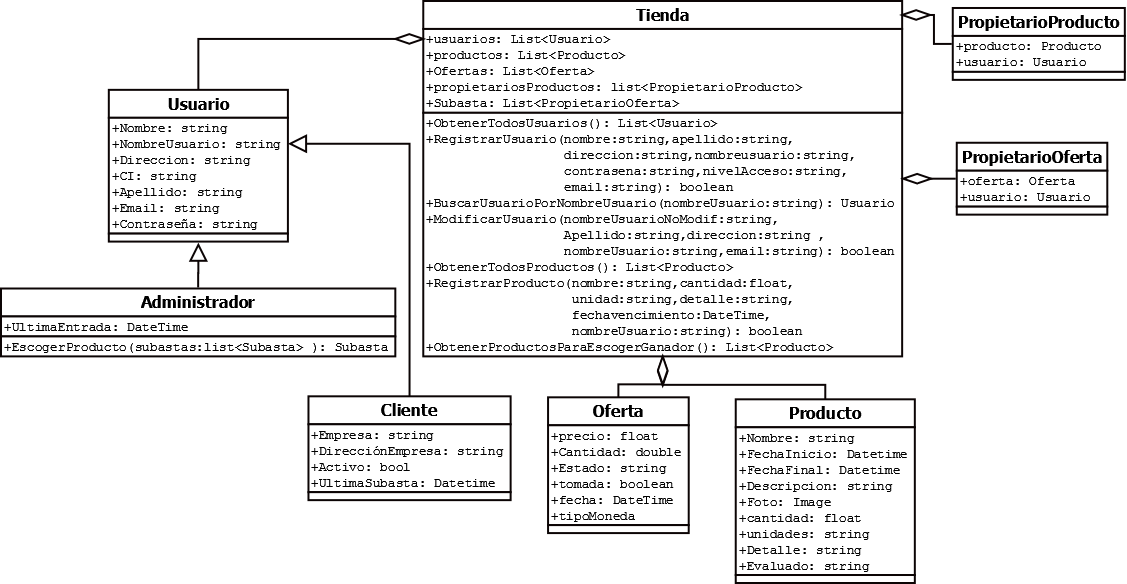
Se optó por utilizar 4 capas en la parte del cliente:

* Base de datos: Contiene todas las clases estáticas y sus diferentes métodas para el acceso a la base de datos.
* ModeloWCF: Contiene todos los objetos que se manejaran para la llamada remota e interna, a través de DataContract.
* Utilidades: Contiene elementos extra, para una mejor organización.
* ServicioWCF: Contiene todos los métodos que se manejarán para la llamada remota, a través de las Interfaces, donde se definen los ServiceContract y las clases que contienen la implementación de las interfaces. Ésta capa se comunica con las tres anteriores.

1. IMPLEMENTACIÓN

* Para la implementación se utilizará el IDE de Visual Studio 2010
* Windows Comunication Fundation (WCF)
* Para un cliente utilizaremos ASP.NET
* Para el manejador de base de datos utilizaremos Postgress
* Para prueba de servicios se utilizará WCF Test Client.

Diagrama de Clases



Base de Datos

