MUNICIPALIDAD DE CORONEL OVIEDO

UNIDAD OPERATIVA DE CONTRATACIONES (UOC).

ADDENDA Nº 01/2012

CONTRATACION DIRECTA No.22/2012 EXP.No.321285.ID No.241977 IMPLEMENTACIÓN DE SISTEMA INFORMATICO DE GESTIÓN MUNICIPAL.

"ESPECIFICACIONES TECNICAS DE SISTEMA DE GESTION MUNICIPAL"

PLATAFORMA DE DESARROLLO A APLICAR

1- **LOS OFERENTES DEBERAN CONSIDERAR LAS ESPECIFICACIONES DE LA PRESENTE ADENDA PARA PREPARAR Y PRESENTAR CON LAS OFERTAS.**
2- **SE DEBERA PRESENTAR CURRICULO DE ESPECIALISTA EN JAVA J2EE JUNTO CON LOS CURRICULOS DE LOS RESPONSABLES.**

Se describe a continuación la plataforma en la que deberá estar desarrollada la plataforma del SIA:

PLATAFORMA POSTGRE

Servidor de Aplicaciones:

POSTGRE Application Server Edition 3.5g for Linux Red Hat o Windows Server 2008

Herramientas de desarrollo:

POSTGRE Developer Suite 10 g

Base de Datos:

POSTGRE 3.5g Release 2 for Linux Red Hat

Herramientas de anális y reportes

POSTGRE Intelligence Discoverer

PLATAFORMA DE SOFTWARE LIBRE PARA EL DESARROLLO

En la actualidad, existen también algunos módulos desarrollados en plataforma J2EE con un framework que ya requiere actualización. Por lo tanto, para el nuevo SIA deberá optarse por entornos de desarrollo más actualizados basados en plataforma de software libre bajo las características que se enuncian a continuación:

CARACTERÍSTICAS

Escalabilidad y alta disponibilidad: El software debe estar diseñado y desarrollado para funcionar en un ambiente de servicios en *cluster,* fácilmente escalable y que ofrezca alta disponibilidad.

Capacidad de actualización componentes de desarrollo (framework): La plataforma de desarrollo debe proveer mecanismos de actualización de sus componentes, teniendo en cuenta dependencias y posibles conflictos entre ellos.

Portabilidad: El ambiente de desarrollo y el ambiente de producción deberá poder ser montado en distintos sistemas operativos (Windows, GNU-Linux/Unix, etc).

Reusabilidad: El desarrollo de módulos del sistema deberá basarse en componentes comunes y reutilizables.

Interoperabilidad: El software desarrollado deberá permitir el intercambio de datos con sistemas externos de manera segura vía Servicios Web u otra interfaz.

Interfaz rica de usuario: El software debe proveer al usuario interfaces intuitivas y fáciles de utilizar.

Productos libres: El conjunto de herramientas y servicios para los ambientes de producción y desarrollo deberán ser de licencia libre.

Partiendo de las características mencionadas la arquitectura propuesta se basa sobre un entorno web **Java EE** (arquitectura Java *Enterprise Edition*) versión 5 o superiores con posibilidad de actualización, con productos totalmente libres (*open source*), utilizando Java como lenguaje de programación y respetando las características antes mencionadas.

A continuación se presentan los diagramas de la arquitectura Java EE, en sus últimas especificaciones (JEE 5 y JEE 6).

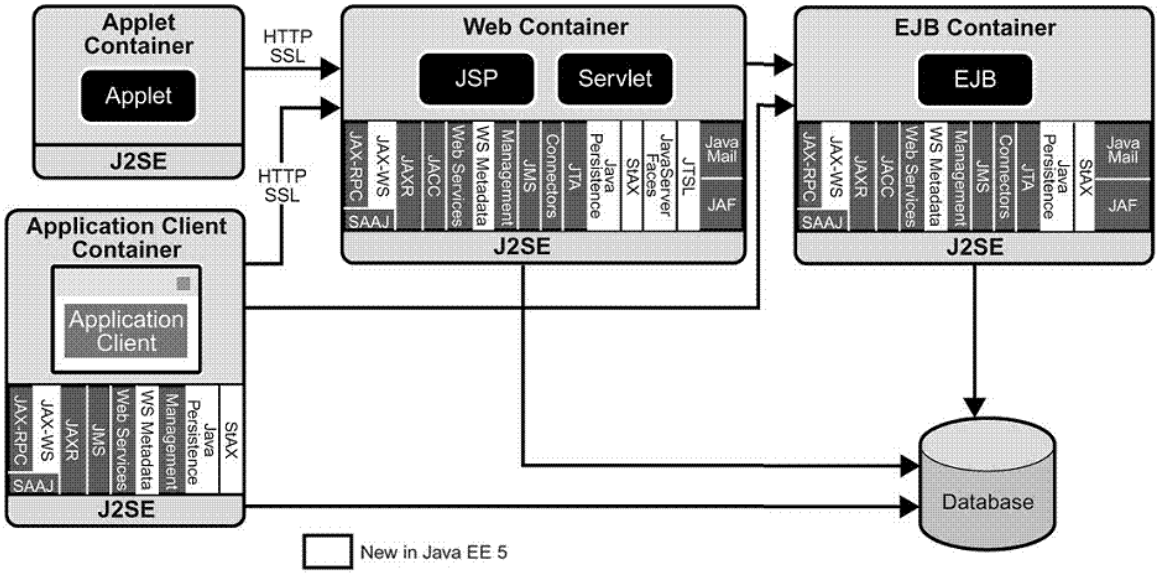

Fig. 1. Arquitectura de Java EE 5

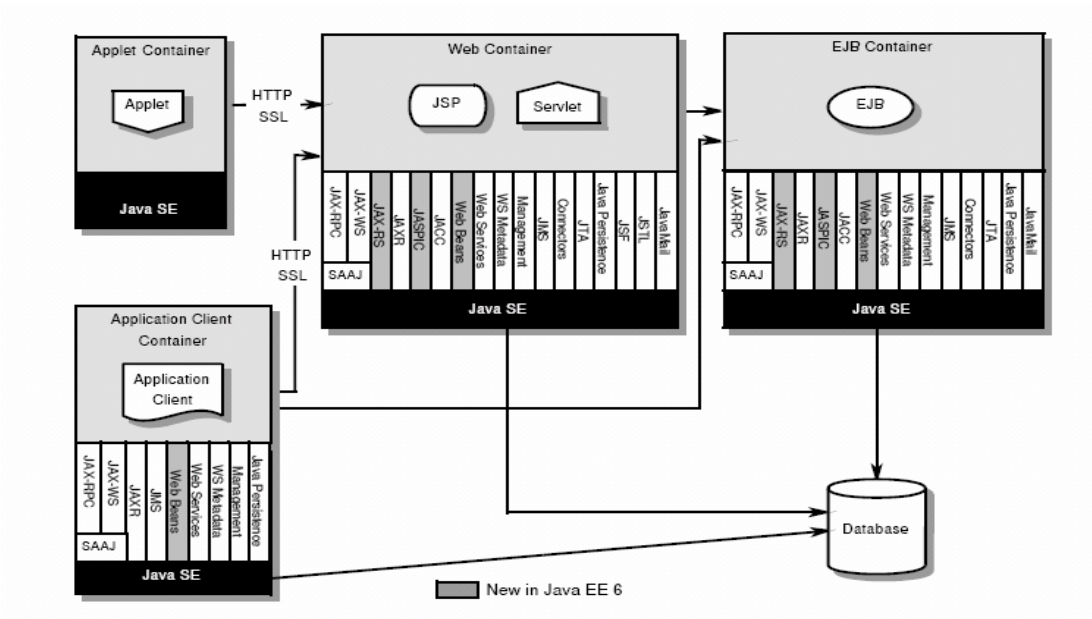

Fig. 2. Arquitectura de Java EE 6

JBOSS APPLICATION SERVER

El Servidor de Aplicaciones que provee:

Contenedor para aplicaciones Web/JEE estándares,
Soporte de configuración en cluster,
Servicios de seguridad (*Security Domains*, JAAS), servicio de tareas programadas (*JBoss Scheduler*), servicio de mensajería (JBossMQ), conectores a fuentes de datos (*datasources*), etc.

JBOSS SEAM

JBoss Seam provee una plataforma open source que integra diversas tecnologías para el desarrollo de aplicaciones WEB JEE.

Principales tecnologías que integra JBoss Seam:

JSF – JavaServer Faces: Tecnología que simplifica el desarrollo de interfaces ricas de usuarios.

JBoss Seam incluye **RichFace**s, librería que provee componentes altamente reutilizables y configurables de interfaz de usuario, útiles para construir esquemas de pantallas que presentan datos complejos.

EJB 3 – Enterprise Java Beans: Tecnología robusta, flexible y altamente escalable que permite desarrollar componentes reutilizables, implementando de manera más sencilla la lógica de negocio.

JPA - Java Persistence API: Tecnología para acceder, persistir y manejar datos entre componentes java y bases de datos relacionales.

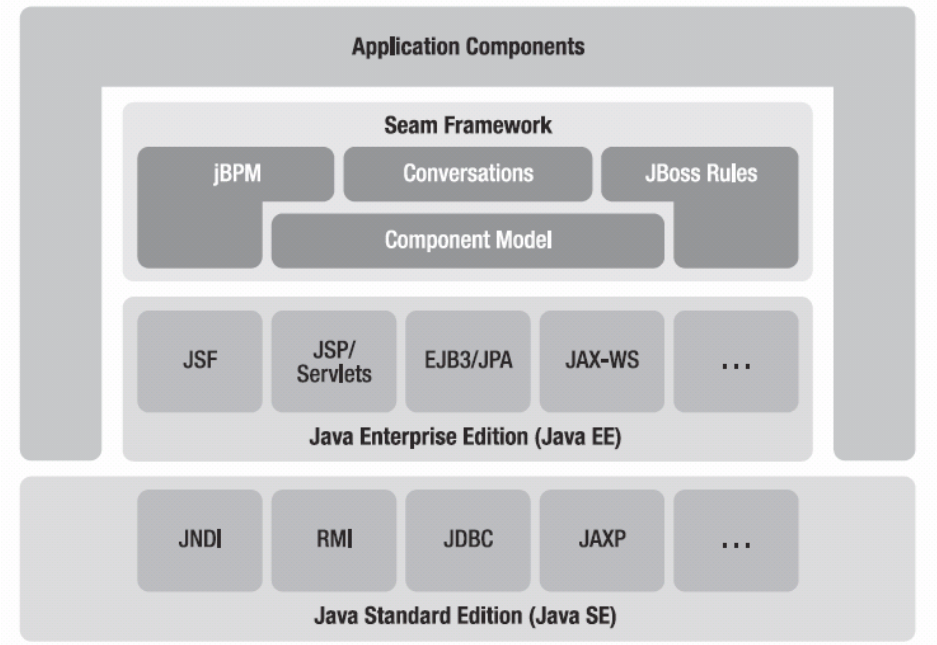

Fig. 3. Arquitectura de JBoss Seam

JBossWS

El JBossWs implementa especificaciones JAX-WS para desarrollo de Servicios Web. Este producto viene incluido en el Servidor de Aplicaciones JBoss versión 5 en adelante.

Eclipse IDE – JBoss Tools

Eclipse *Integrated Development Environment* es un entorno para desarrollo de aplicaciones Java. El JBoss-Tools es un agregado o *plug-in* para Eclipse que provee un conjunto de herramientas para simplificar el desarrollo de aplicaciones Web utilizando el framework Seam.

Jasper Reports - BIRT

Herramientas para diseño y generación de reportes que soportan distintos formatos.

Programación de Tareas

El JBoss Application Server ofrece servicios de tareas programadas (JBoss Scheduler), sin embargo podría considerarse el uso de otras alternativas como Quartz.

Unidades de Testeo

En cuanto al testeo de componentes Java, JBoss Seam provee al desarrollador librerías que facilitan la generación de unidades de testeo.

Estas librerías funcionan con los complementos o plug ins de testeo para eclipse TestNG y JUnit.

Herramientas de apoyo al desarrollo

Administración de Incidencias

Existen varias alternativas de licencia libre para administración de incidencias. Las recomendadas son Mantis, o Bugzilla.

Gestión de Configuración de Software (SCM)

El control y registro de cambios en el código fuente y documentación del proyecto es indispensable.

Existen soluciones libres para esto, la recomendada es Subversion.

Herramientas de Diseño

Modelado UML y diseño de bases de datos

El *Enterprise Architect* es una herramienta propietaria que, entre otras características, permite diseño y modelado UML de software.

ESQUEMA DE LA PLATAFORMA PROPUESTA

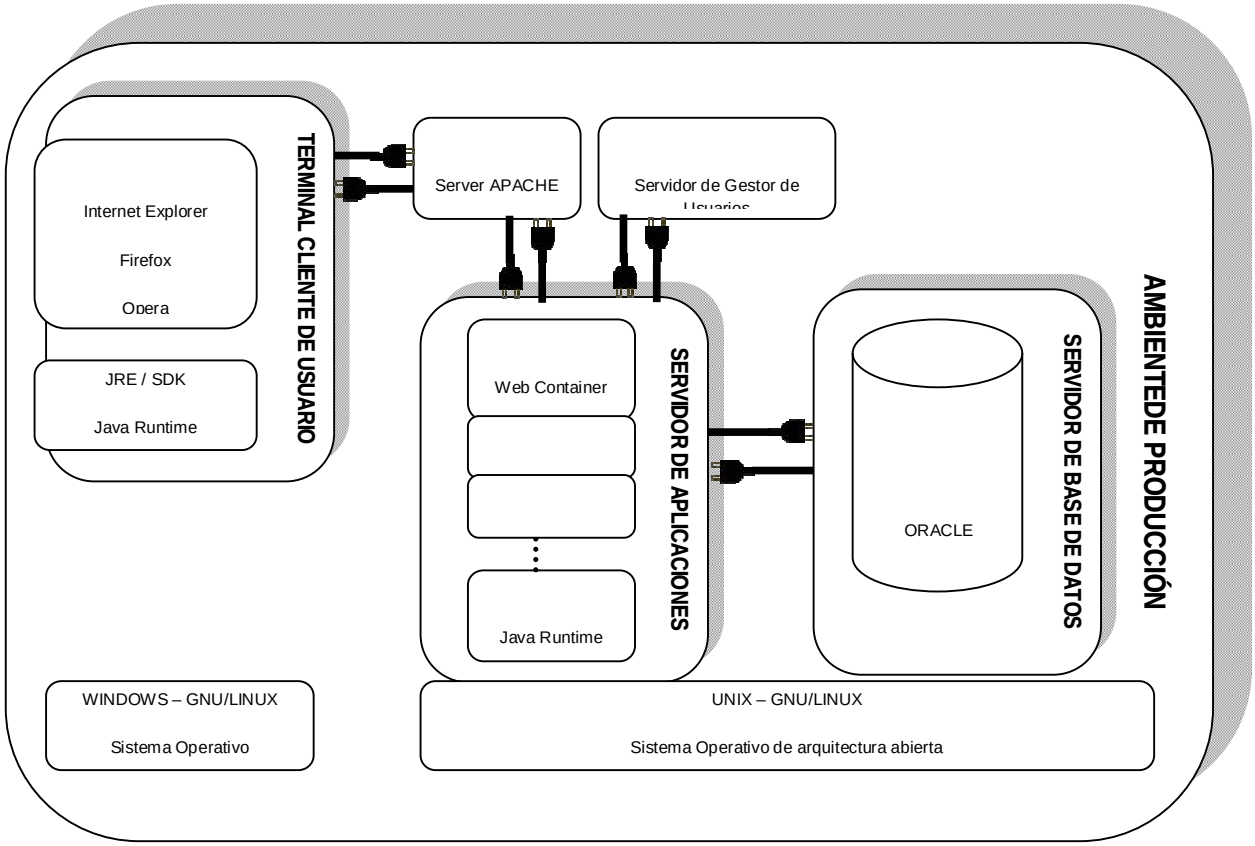

Fig. 4. Ambiente de producción

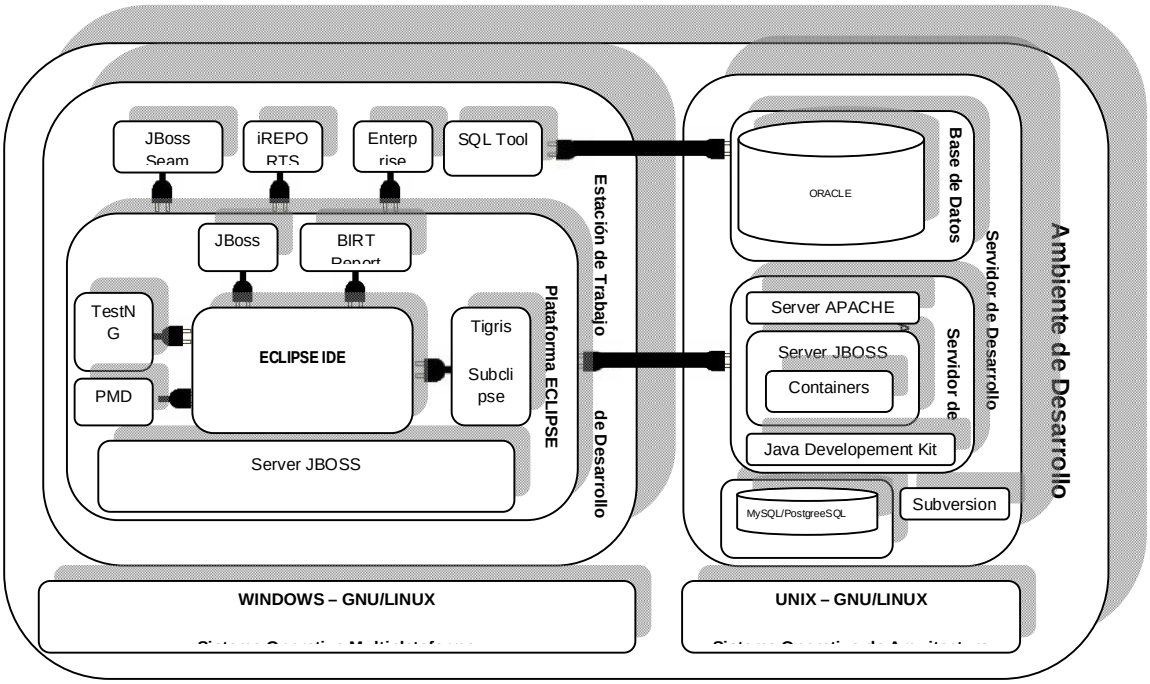

Fig. 5. Ambiente de desarrollo

CONSIDERACIONES ADICIONALES

- El software deberá ser probado en ambiente de *cluster* y funcionar satisfactoriamente antes de ser aprobado para su puesta en producción. Los servidores de aplicaciones como JBoss soportan distintas configuraciones de *cluster*.

- Los componentes relacionados a la interfaz de usuario deben adecuarse a los estándares y ser en lo posible conocidos y probados por la comunidad de desarrollo. Hay que considerar que las necesidades de los usuarios son pantallas con características de *rich client* que involucran a elementos web de alto desempeño basados en la comunicación asíncrona entre el cliente y el servidor (AJAX). La flexibilidad, interoperabilidad y reusabilidad de los componentes web son factores importantes a considerar para el desarrollo de las interfaces de usuario.

- El sistema deberá hacer uso de los servicios del servidor de aplicaciones, evitando recurrir a otros *frameworks* o API's que implementen servicios similares. Todo software adicional a ser incorporado al sistema (librerías de clases, artefactos, recursos, etc.) y sobre el cual se construya el resto de la aplicación deberá estar sujeta al estándar Java EE y tener el propósito de facilitar la integración de los componentes de la aplicación mejorando u optimizando alguna funcionalidad.

- Al momento de la elaboración de este documento, la última versión estable del *JBoss Application Server* es la 5.1, el cual satisface la especificación 5 de la arquitectura Java EE. No obstante, debe considerarse que está próximo el lanzamiento de la versión 6.0 del *JBoss AS*, el cual va a implementar la especificación *6* de Java EE. Durante el transcurso de tiempo del proyecto "Reingeniería y Mejora del SIAF" debe tenerse la posibilidad de adoptar la versión más nueva del servidor de aplicaciones, para evitar una migración de la plataforma a posteriori del desarrollo del sistema proyectado. Esto tendría la ventaja de adoptar un servidor robusto con la arquitectura más nueva y las mejoras introducidas en la especificación Java EE 6. Entre las nuevas características de Java EE 6 se puede mencionar la actualización de JavaServer Faces a la versión

2.0, lo cual facilitaría aun más el desarrollo de la capa de presentación, y por ende, al desarrollo de las interfaces de usuario.

ESQUEMA DE SEGURIDAD PARA SERVICIOS WEB

El sistema a ser desarrollado deberá estar preparado para proveer servicios web, actuando como proveedor de servicios, como también ser cliente de otros servicios web externos, siendo en este último caso cliente de aquellos servicios.
El marco de seguridad para los servicios web debe ser robusto y flexible, permitiendo elegir el grado de seguridad a ser aplicado a cada servicio web según el alcance que tiene sobre operaciones críticas del sistema. Debe tenerse en cuenta que los datos intercambiados por esta vía por lo general son datos sensibles o confidenciales, por lo que suelen estar protegidos para mantener la integridad y la confiabilidad.

Debe considerarse que cualquiera sea la arquitectura Java EE a utilizarse (Java EE 5 o Java EE6), y su correspondiente servidor de aplicaciones que cumple con la arquitectura, debe poder obtenerse el grado máximo de seguridad para los web services.
La autorización de acceso a servicios web, encriptación de mensajes y la firma de mensajes debe disponerse y poder implementarse con los web services usando los frameworks estándares de la arquitectura Java EE elegida.
Actualmente el máximo grado de seguridad utiliza la infraestructura de claves x.509v3, algoritmo de encriptación AES-256 con tamaño de clave 2048 bits.

Puede también complementarse con SSL (*Secure Sockets Layer*), el cual es el protocolo estándar para transacciones seguras por Internet mediante conexiones HTTPS (túnel seguro).

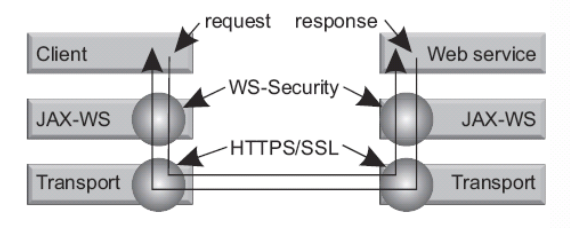

Fig. 6. Protocolos de seguridad involucrados en distintas capas

De habilitarse SSL, se requieren certificados digitales autofirmados, ó la compra de los mismos de algún *Certificate Authority* - entidad emisora de certificados digitales.