



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

# **Arquitecturas de Software Empresariales Aplicadas en Campo Laboral**

**INFORME DE ACTIVIDADES PROFESIONALES**

Que para obtener el título de

**Ingeniero en Computación**

**P R E S E N T A**

Javier Alberto Estrada García

**ASESOR DE INFORME**

Stalin Muñoz Gutiérrez



**Ciudad Universitaria, Cd. Mx., 2016**

A la Universidad Nacional Autónoma de México.

A la Facultad de Ingeniería

A mis padres Silvia y Javier, que siempre me apoyaron y han sido un ejemplo a seguir en toda mi carrera, sin ellos no sería la persona que soy en estos momentos, esto que me brindaron no tiene precio y para mi es algo invaluable que conservare toda mi vida, los amo a ambos.

A mi abuelo Alberto que empezó a enseñarme muchas cosas desde temprana edad que han ayudado a mi desarrollo profesional, con el aprendí las bases con las que he crecido, y sigo aprendiendo, muchas gracias por todo abuelo.

A mi familia, mis abuelas Irene, Martha y mi hermano Mauricio, que me apoyaron siempre.

Al profesor Stalin Muñoz que me dio mi primer trabajo y ha sido una persona ejemplar en mi carrera y a mi amigo Renato que ya no está con nosotros, pero del cual aprendí muchas cosas en la vida y con el cual compartí mi primer empleo junto con el profesor Stalin.

## Índice

1	Introducción .....	5
2	Objetivo .....	7
3	Descripción de la empresa o medio en que labora.....	8
3.1	Oracle. ....	8
3.1.1	¿Qué es Oracle?.....	8
3.1.2	Posición Consultor Asociado .....	8
3.2	Negocios Digitales. ....	9
3.2.1	¿Qué es Negocios Digitales? .....	9
3.2.2	Visión .....	9
3.2.3	Posiciones desarrollador Java y líder técnico del proyecto.....	9
3.3	Softtek.....	10
3.3.1	¿Qué es Softtek? .....	10
3.3.2	Misión y Visión .....	10
3.3.3	Posiciones desarrollador Java e Integrador Tecnológico. ....	11
4	Antecedentes del tema .....	12
4.1	Estilo Modelo N Capas.....	13
4.2	Estilo Manejador de eventos o mensajes asíncronos.....	14
4.3	Estilo SOA (Service Oriented Architecture).....	15
4.4	Arquitecturas heterogéneas .....	17
4.5	Arquitecturas no consideradas estilos de arquitectura .....	17
4.5.1	Tuberías y filtros .....	17
4.5.2	Base de eventos, invocación implícita. ....	18
4.5.3	Repositorios.....	18
5	Definición del problema o contexto de la participación profesional.....	20
5.1	Proyecto ASAP .....	20
5.2	Proyecto Metasolv Solution .....	21
5.3	Proyecto OSM .....	21
5.4	Proyecto SNIIM .....	22
5.5	Proyecto Identidad por Huella Digital .....	23
6	Metodologías utilizadas .....	24
6.1	OUM Oracle Unified Method .....	24
6.1.1	Comienzo (Inception).....	24
6.1.2	Elaboración (Elaboration).....	24

6.1.3	Construcción (Construction)	25
6.1.4	Transición (Transition)	25
6.1.5	Producción (Production)	25
6.2	MAAGTICSI	26
6.2.1	Planeación	26
6.2.2	Análisis y Diseño	27
6.2.3	Construcción	27
6.2.4	Integración	27
6.2.5	Aceptación del sistema	27
6.2.6	Control y seguimiento	27
6.3	Softtek Software Development Process (SSDP)	28
6.3.1	Origen (Inception)	28
6.3.2	Elaboración (Elaboration)	28
6.3.3	Construcción (Construction)	29
6.3.4	Transición (Transition)	29
7	Resultados	30
7.1	Resultados de mi participación en el proyecto ASAP	30
7.2	Resultados de mi participación en el proyecto Metasolv	30
7.3	Resultados de mi participación en el proyecto OSM	31
7.4	Resultados de mi participación en el proyecto SNIIM	31
7.5	Resultados de mi participación en el proyecto Identidad por Huella Digital	32
8	Conclusiones	33
9	Bibliografía	35
10	Anexos	37
10.1	Detalle técnico de la participación profesional	37
10.1.1	Proyecto plataforma ASAP	37
10.1.2	Proyecto plataforma MetaSolv Solution	39
10.1.3	Proyecto plataforma OSM	42
10.1.4	Proyecto SNIIM (Sistema Nacional de Información e Integración de Mercados)	46
10.1.5	Proyecto Identidad por Huella Digital	49
10.2	Glosario	53

## 1 Introducción

Este informe contiene los cambios realizados a las arquitecturas de 5 proyectos en los cuales estuve involucrado en mi carrera profesional. El rubro de arquitectura de software es muy amplio, por lo que este trabajo está limitado al área de estudio de arquitecturas empresariales. El enfoque que presento en el informe es el de un desarrollador y consultor. Cabe mencionar que mi rol no fue el de arquitecto, pero trabajé directamente con ellos en las actividades que se detallan más adelante.

El arquitecto de software la mayoría de las veces tienes que adaptarte a sistemas que previamente han sido construidos e integrados, sin embargo, él puede innovar en la solución tecnológica que esté planteando, para que ésta soporte tanto los sistemas legados con los cuales tiene que interactuar, como también nuevas funcionalidades que esta solución puede ofrecer al negocio. También tenemos el caso de proyectos donde se necesita que se pase de una tecnología a otra, aquí vemos que los requerimientos funcionales son los mismo y tal vez los requerimientos no funcionales tengan cambios para hacer la solución más robusta, eficiente e interoperable.

El arquitecto de software es el rol de la persona que se encarga de definir el sistema o la solución. Existen varios tipos de arquitectos con los que tuve interacción, algunos se encargaban de hacer soluciones de sistemas (organizar los componentes de software para resolver un conjunto de casos de uso orientados a los actores del sistema), otros de hacer la arquitectura de un producto. Un punto en común entre ellos es que tienen la responsabilidad de crear la especificación del sistema que satisface los requerimientos y toma en cuenta la visión y posibilidades de expansión, al mismo tiempo que aprovecha al máximo las capacidades de los recursos que tienen asignados y proyectados.

Los arquitectos de software realizan ciertas tareas que son indispensables para el desarrollo de un proyecto, una de ellas es generar la propuesta arquitectónica. Esta se realiza en las primeras etapas de casi todas las metodologías de desarrollo de software.

Esto permite dar un panorama del sistema o la solución al equipo de desarrollo, al cliente y a las partes interesadas. Los arquitectos de soluciones, además acompañan a las personas que venden el producto a juntas de inicio con los clientes, para poder analizar el negocio y ver si es posible integrar las plataformas que están vendiendo con la actual arquitectura del negocio. El proceso de análisis consiste en evaluar la factibilidad de que la propuesta cubra con los requerimientos. Cuando la integración del sistema legado no es posible se debe hacer una reingeniería del mismo.

Los arquitectos que realizan productos tienen la tarea de generar ciertos artefactos que son clave para la elaboración del proyecto, entre ellos se encuentran los diagramas de clases, diagramas de componentes, diagramas de secuencia y los diagramas de despliegue, con esto el equipo tiene claro a donde se quiere llegar con el sistema. Los arquitectos satisfacen tanto los requerimientos funcionales como los no funcionales, para que el sistema pueda tener el mejor rendimiento y cumpla con los requerimientos que el cliente este solicitando. En lo requerimientos funcionales tenemos los casos de uso a cubrir, mientras que en los requerimientos no funcionales se tienen la alta disponibilidad, resguardo de datos, escalabilidad, etc.

Las arquitecturas empresariales han tenido éxito ya que actualmente soportan la operación de grandes negocios, por bondades que este tipo de arquitecturas brinda a la operación de los clientes. Podemos hablar de una arquitectura empresarial cuando se enfoca a los procesos negocio, mientras las plataformas sean flexibles para adaptarse a cambios en estos procesos, será más rentable para el negocio. Entre algunos de los términos que hacen realidad este tipo de flexibilidad tenemos la escalabilidad, asegura que el sistema va a soportar mayor carga a largo plazo, alta disponibilidad asegura que el sistema va a tener disponibilidad las 24 horas del día, los 7 días de la semana, concurrencia y persistencia de datos, asegura que la información no se perderá, será consistente y estará resguardada, el acceso controlado a recursos, asegura que el sistema se pueda integrar a través de medios agnósticos.

Entre las características de este tipo de arquitecturas están los dominios, estos existen a diferentes niveles, pueden ser a nivel de red, nivel de sistema operativos, o nivel de servidor de aplicaciones. Los dominios existen para resguardar sistemas, ya que no todos necesitan interactuar entre sí, estas prácticas ayudan a evitar vulnerabilidades en la seguridad de las plataformas.

En este reporte detallo las actividades en las cuales estuve trabajando en 5 diferentes proyectos. La definición de cada uno de estos problemas y los alcances de mi participación se encuentran ubicados en la sección 5. En la sección 7 doy los resultados que tuve en cada uno de los proyectos comparándolos con su problemática correspondiente. En la sección 8 discuto las conclusiones de mi reporte con respecto a los resultados obtenidos contrastados con los objetivos planteados en la sección 2. Para mayor precisión, la descripción técnica de cada una de las soluciones propuestas junto con algunos diagramas de arquitectura, se encuentran en el anexo de la sección 10.

## **2 Objetivo**

Describir, analizar y comparar arquitecturas de software empresariales que son usadas actualmente en el campo laboral en las industrias de telecomunicaciones, gobierno y sector bancario en proyectos de desarrollo de software.

### **3 Descripción de la empresa o medio en que labora**

#### **3.1 Oracle.**

##### **3.1.1 ¿Qué es Oracle?**

Es una empresa de TI que tiene presencia a nivel mundial, entre sus objetivos está el vender hardware y software que ellos mismos desarrollan.

Esta empresa tuvo éxito gracias a su motor de base de datos Oracle, esto represento que pudieran expandir su negocio a otras áreas de TI y no solo concentrarse en las bases de datos.

Oracle es conocido por comprar tecnología y absorber empresas, esto le genera una cartera de clientes más grande y plataformas que ya tienen un mercado en la industria. Ver la referencia [8].

##### **3.1.2 Posición Consultor Asociado**

Es una posición introductoria en el ámbito laboral, las funciones que se realizan van desde aprender acerca de las plataformas que Oracle implementa en el negocio de las telecomunicaciones, como también realizar instalaciones de los productos, levantar requerimientos junto con consultores de más experiencia, hacer soporte de los productos (ver las bitácoras de las aplicaciones que se instalan o dar capacitación a las personas que estarán usando los sistemas), además de hacer desarrollos a la medida.



## **3.2 Negocios Digitales.**

### **3.2.1 ¿Qué es Negocios Digitales?**

Es una empresa mexicana de consultoría, servicios tecnológicos y Outsourcing. Ver referencia [10].

### **3.2.2 Visión**

La visión de la empresa tiene como objetivo atender proyectos en el sector privado y el público.

Sus principales servicios son:

- Planeación estratégica
- Desarrollo organizacional
- Reingeniería y mejora de procesos
- Aplicación de metodologías de administración de proyectos y oficina de proyectos
- Seguridad de la información
- Auditoria de procesos, sistemas, seguridad y servicios
- Capacitación especializada en Tecnologías de la Información y Comunicación (TIC)
- Mesa de servicio institucional
- Implementación de infraestructura de TIC

### **3.2.3 Posiciones desarrollador Java y líder técnico del proyecto**

El desarrollador Java se enfoca al desarrollo de software en tecnología Java, las actividades relacionadas son el entendimiento del requerimiento, el diseño del

componente que se quiere construir, codificar el componente con tecnología Java. El desarrollador tiene que ser capaz de probar el componente tanto en pruebas unitarias como integrales.

La posición de líder técnico tiene la visión de la solución completa y no de algún módulo en específico, gran parte del trabajo es analizar como la solución puede ser construida, que tecnologías pueden ser usadas y reportar avances al líder del proyecto. Es importante que el líder técnico este en juntas de requerimientos con el cliente, con esto el líder del proyecto tiene confianza de aceptar requerimientos reales en los tiempos establecidos.

### **3.3 Softtek.**

#### **3.3.1 ¿Qué es Softtek?**

Es una empresa que fue fundada en 1982, Softtek es un proveedor global de servicios orientados a procesos de TI. Una de las premisas de Softtek es mejorar el tiempo de entrega de soluciones de negocio, para reducir costos de las aplicaciones existentes, y entregar aplicaciones mejor diseñadas y probadas. Ver referencia [9].

#### **3.3.2 Misión y Visión**

En Softtek, existe el termino Responsabilidad Social Corporativa (CSR) que es un proceso de creación de valor en el cual se esfuerzan por lograr el equilibrio entre su rentabilidad, el entorno y las personas.

Sus principios se apegan, entre otros, a los estándares que estipula la norma ISO 26000 y la OIT (organización internacional del trabajo). El intercambio de conocimiento y la colaboración con otras empresas y organizaciones sociales, son de gran importancia.

### **3.3.3 Posiciones desarrollador Java e Integrador Tecnológico.**

La posición de desarrollador Java que desempeñé tuvo como objetivo diseñar nuevos componentes, desarrollarlos, probarlos, llevarlos a producción a través de un sistema de control de cambios y hacer mejoras a componentes ya existentes.

Por otro lado, como Integrador Tecnológico se tiene que dar seguimiento al trabajo de los equipos de otras plataformas involucradas en el proyecto, estar en juntas técnicas y de negocio, tener comunicación continua con el dueño de la plataforma para informar cambios o problemas. Se necesitan conocimientos técnicos para saber que las interfaces son capaces de comunicarse y negociar fechas de entrega de la implementación de los requerimientos.

## 4 Antecedentes del tema

Una arquitectura de desarrollo se puede definir como la forma en que se organizan los componentes para que puedan interactuar entre ellos y poder dar solución a un problema.

Existen diversos tipos de arquitecturas a lo largo de la historia, que tienen propósitos diferentes, como construir un sistema operativo, hacer una aplicación web 2.0, etc. En las arquitecturas empresariales uno de los términos más importantes es el de aplicaciones robustas, esto quiere decir que son ampliamente configurables para poder tener un mayor rendimiento a la hora de la operación diaria. Entre sus principales características están la alta disponibilidad, quiere decir que la aplicación tendrá servicio 24x7, para este fin muchas veces se recurre a clusterizar los ambientes para tener mayor potencia en los servidores. El termino balanceo de cargas es también parte de una aplicación robusta, quiere decir que existen componentes que tienen la tarea de evaluar el servidor que tiene menor carga, con la finalidad de que la respuesta sea más rápida.

Existen diferentes tipos de arquitecturas de sistemas, como:

- Arquitecturas de Sistemas Operativos: La mayoría de las veces son construidas en lenguajes de programación de bajo nivel, los componentes se modelan principalmente en función del hardware.
- Arquitecturas de Aplicaciones Web: La mayoría de las arquitecturas de aplicaciones Web 2.0 se diseñan en 3 capas, se les denomina como MVC (Modelo, Vista y Controlador).
- Arquitecturas de Aplicaciones Empresariales: Estas arquitecturas están basadas en recursos que los servidores de aplicación proveen para una fácil integración. Además, son arquitecturas muy utilizadas en varias industrias ya que tienen la ventaja de poder ser escalables.

Mencionare la especificación de tecnologías que son usadas en la versión Java EE 6, como jsr224 que es la especificación que se da a la tecnología de servicios

web 2.0 SOAP, jsr-311 es la especificación de los servicios web RESTful 1.1, jsr-000318 es la EJB 3.1, JSR315 que es Servlets 3.0, JSR314 de JavaServer Faces 2.0, JSR317 para JPA 2.0. Estas especificaciones son las más usadas al momento de realizar una aplicación empresarial en tecnología Java.

- Arquitecturas tecnologías móviles: En este tipo de arquitectura lo más importante es cuidar los recursos, ya que una aplicación al correr en un dispositivo móvil se necesita que sea ligera y no consuma mucho en tiempo de procesamiento.
- Arquitecturas en la nube: Estas arquitecturas tienen la innovación de no necesitar infraestructura para poder realizar tu aplicación, todo se ve como un servicio, donde el proveedor del servicio en la nube da lo necesario para que se despliegue la aplicación y tenga un servicio de 24x7.

En las arquitecturas de software podemos encontrar muchos estilos. Los estilos son una manera de implementar una solución a un problema, esto marca una tendencia a la hora de hacer la programación del sistema, a continuación, presento algunos de los estilos que son usados en aplicaciones empresariales y con los cuales estuve involucrado en los proyectos que reporto.

#### **4.1 Estilo Modelo N Capas**

Las arquitecturas en capas son muy comunes en las aplicaciones empresariales. Este modelo se puede definir como la separación de problemas, cada capa se encarga de resolver cierto problema sin delegar a la capa siguiente un problema que no le corresponde. Este ha sido un modelo que ha funcionado bastante tiempo y por lo tanto es de los más implementados en la época actual.

Entre los modelos más famosos tenemos el MVC (Modelo, Vista y Controlador), los componentes principales son la vista (es lo que el usuario ve), el modelo de datos (es donde se aloja la información) y el controlador (es el disparador cuando un evento sucede en la vista). Ver figura 1.

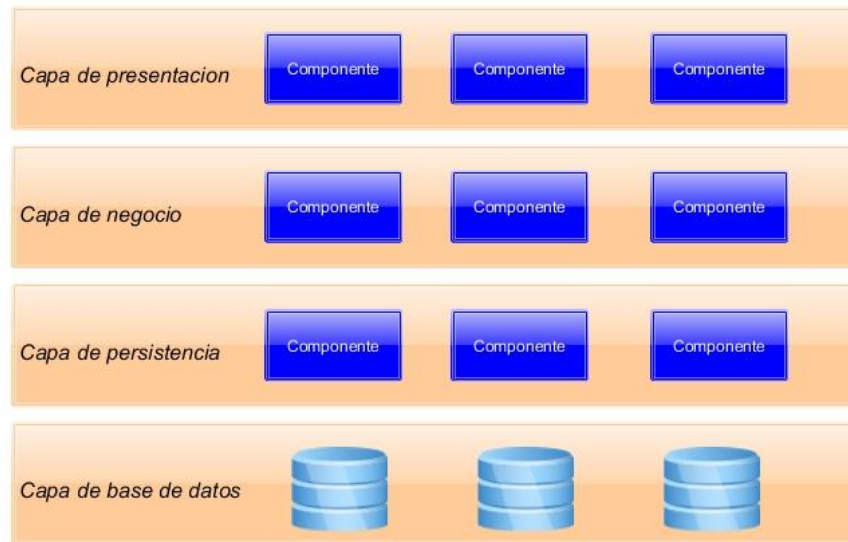


Figura 1: Patrón de arquitectura por capas. Diagrama basado en referencia [2].

## 4.2 Estilo Manejador de eventos o mensajes asíncronos

Este tipo de arquitectura es usado para un modelo de comunicación asíncrona en la mayoría de los casos. La tecnología utilizada para este propósito son las colas y las suscripciones por lo que el lenguaje de programación en el que se leen o escriben los mensajes es agnóstico. Este tipo de tecnología es llamado "Message Broker", que también es conocido por ser un "Middleware", una capa que se usa para integrar sistemas. El traslado de los mensajes se puede hacer de dos maneras, la primera mediante el API que el lenguaje de programación provee, y la segunda mediante funciones que el servidor de aplicaciones provee, en este último caso tenemos métodos como "puente" o "almacenamiento y reenvío", en donde los mensajes pueden ser enviados a colas o suscripciones remotas. Ver figura 2 y figura 3.

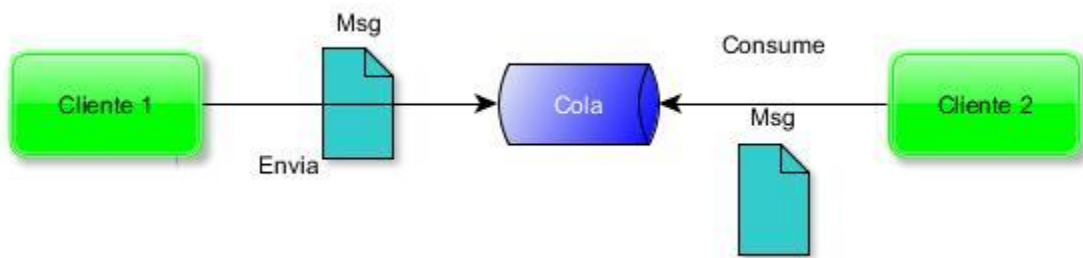


Figura 2: En el diagrama se observa el funcionamiento de una cola, el Cliente1 se encarga de escribir el mensaje en la cola, y Cliente2 se encarga de leerlo y procesarlo. Diagrama basado en referencia [19].

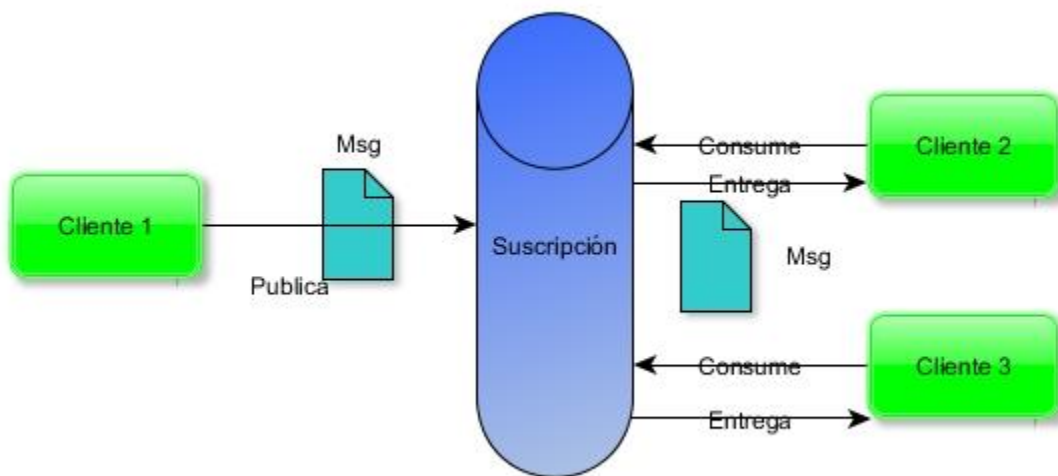


Figura 3: En este diagrama observamos el comportamiento de una suscripción, donde todos los clientes que están suscritos recibirán el mensaje. Diagrama basado en referencia [19].

### 4.3 Estilo SOA (Service Oriented Architecture)

Las arquitecturas orientadas a servicios son comunes en el mundo de las aplicaciones empresariales.

Se utilizan protocolos de comunicación para que el mensaje sea transferido. Los protocolos más comunes son HTTP y HTTPS. Las tecnologías que son más usadas para este tipo de tareas son SOAP (Simple Object Access Protocol) y RESTful (Representational State Transfer).

En el caso de servicios SOAP, tienen su especificación a través del archivo WSDL (Web Services Descriptor Language), a este archivo se le denomina contrato, esto ayuda al desarrollo del servicio ya que es agnóstico al lenguaje de programación en el cual se implementará la solución. La tecnología con la que trabaja este tipo de servicios es XML, donde la información viaja de forma segura a través del protocolo.

Para los servicios RESTful tenemos una especificación que está definida en el archivo WADL (Web Application Descriptor Language) que tiene formato XML. A diferencia de SOAP en esta tecnología los mensajes tienen formato JSON (JavaScript Object Notation). Ver figura 4.

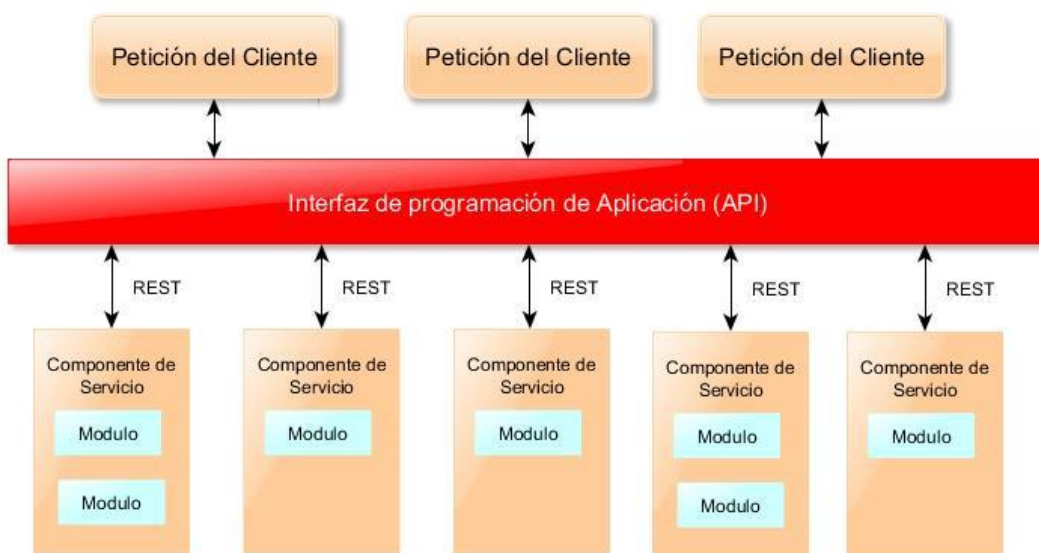


Figura 4: Topología basada en RESTful API. Diagrama basado en referencia [2].

Otro concepto importante en las arquitecturas SOA, es el canal de servicios empresariales (ESB por sus siglas en inglés de Enterprise Service Bus), este middleware sirve para poder controlar el intercambio de mensajes entre servicios, como también de manejar eventos y transformar datos.



## **4.4 Arquitecturas heterogéneas**

Existen muchas maneras en que se pueden combinar los estilos de arquitecturas. Una forma es a través de jerarquías. Un componente de un sistema organizado en un estilo de arquitectura puede tener una estructura interna que es desarrollada de manera diferente. Por ejemplo, en una tubería en UNIX los componentes individuales pueden representar uso interno virtual en cualquier estilo, incluido por supuesto otra tubería y filtro del sistema.

Una segunda forma de estilos para ser combinados es permitir a un solo componente use conectores de arquitectura mixtos. Por ejemplo, un componente puede acceder a un repositorio a través de su interfaz, pero interactúa a través de tuberías con otros componentes en un sistema y acepta el control de la información a través de otra parte de la interfaz.

Otro ejemplo es en base de datos activas. Este es un repositorio que activa componentes externos a través de invocación implícita. En esta organización componentes externos registran interés en partes de la base de datos. La base de datos automáticamente invoca la herramienta apropiada en base de su asociación.

La tercera forma para estilos combinados es elaborar completamente un nivel de descripción de arquitectura en un estilo completamente diferente de arquitectura.

## **4.5 Arquitecturas no consideradas estilos de arquitectura**

### **4.5.1 Tuberías y filtros**

En el estilo de tuberías y filtros cada componente tiene sus entradas y sus salidas. El componente lee el flujo de datos de entrada y produce un flujo de datos de salida, entregando una instancia completa y secuencial de resultados. El término de filtros es por la forma síncrona en que se procesan los datos. Los conectores de este tipo sirven

como conductos para los flujos, trasmitiendo salidas de un filtro a la entrada de otro. Por lo tanto, los conectores tienen el termino de tuberías. Ver figura 5.

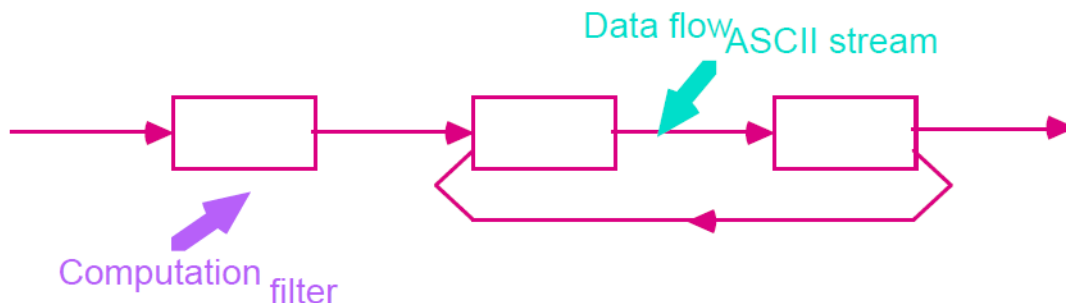


Figura 5: Ejemplo de arquitectura en tuberías

Los mejores ejemplos de esta arquitectura son programas escritos en UNIX.

#### 4.5.2 Base de eventos, invocación implícita.

Tradicionalmente en un sistema se tienen componentes con interfaces que proveen una colección de procedimientos y funciones, los componentes interactúan entre ellos con invocación explícita a través de esas rutinas.

La invocación implícita, integración reactiva y emisión selectiva están basados en los actores, satisfacción de restricciones, demonios y redes de conmutación de paquetes. La idea detrás de invocación implícita es que en vez de invocar a un procedimiento directamente un componente puede anunciar uno o más eventos. Otros componentes en el sistema pueden registrar un interés en el evento con la asociación al procedimiento. Cuando el evento es anunciado, el sistema mismo invoca todos los procedimientos que están registrados para el evento.

#### 4.5.3 Repositorios

En el estilo de repositorio hay dos tipos de componentes, una estructura de datos central que representa el estado actual y la colección de componentes independientes

operando en el almacenamiento de datos central. La interacción entre el repositorio y sus componentes externos pueden variar significativamente entre sistemas. Ver figura 6.

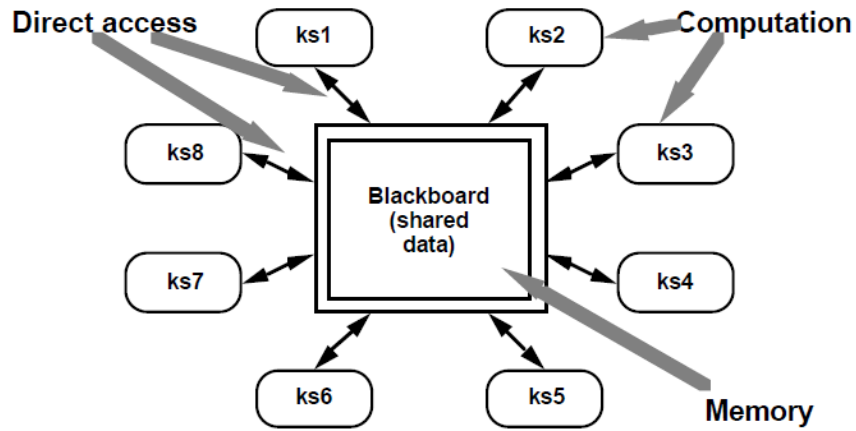


Figura 6: Ejemplo de arquitectura con uso de repositorios

## **5 Definición del problema o contexto de la participación profesional**

### **5.1 Proyecto ASAP**

El proyecto tuvo como propósito implementar la plataforma de activación de servicio automático (ASAP por sus siglas en inglés) en una arquitectura por capas. Este sistema es la capa intermedia entre la orden de servicio y las operaciones que se mandan a el sistema de administración de servicios (OTL) que maneja el cliente, esto fue un cambio a su operación, ya que ellos enviaban las ordenes directo al OTL, si alguna llegaba a fallar el sistema que generaba la orden hacía el reintento por completar el proceso. Al ser ASAP un sistema de aprovisionamiento, en el momento en que una orden es puesta en la cola, ASAP se encarga de completarla en un proceso asíncrono, la orden puede resultar exitosa o fallida dependiendo de los reintentos necesarios configurados, por default se dan 5 reintentos. Las ventajas adquiridas en este tipo de arquitecturas es que cada capa tiene una responsabilidad específica, y con eso se liberan recursos en la capa donde se genera la orden ya que ASAP es responsable de abrir los hilos necesarios para que cada orden pueda ser procesada lo antes posible.

ASAP es una plataforma que se encarga de aprovisionamiento y administración de mensajes que funciona con cartuchos para poder dar lógica de negocio al contenido, para este propósito es necesario utilizar Java donde se usan ciertas clases propietarias de Oracle para poder comunicar el mensaje, esta plataforma es desplegada en Weblogic, usa librerías hechas en C y C++ para hacer el proceso de entrega de mensajes más eficiente.

Las actividades realizadas en este proyecto fueron el desarrollo del cartucho (con tecnología Java y un Plug-In de Eclipse) que ASAP necesita para especificar la lógica del negocio, hacer pruebas unitarias e integrales con los miembros de las plataformas que viven en las otras capas, además del monitoreo de órdenes para ayudar en los casos de prueba.

El alcance del proyecto fue delimitado a hacer 5 operaciones del OTL ya que fue una arquitectura piloto, cuyo objetivo fue evaluar su desempeño en el negocio del cliente.

## **5.2 Proyecto Metasolv Solution**

El propósito del proyecto Metasolv fue configurar y desplegar el sistema de inventario lógico y físico de las redes que el cliente provee. El cliente requería una plataforma para administrar su inventario y dar seguimiento a sus procesos mediante ordenes de trabajo. La flexibilidad que les ofreció Metasolv para integrarse con otras plataformas fue un factor relevante para que se decidiera implementar esta solución.

Para esta asignación realice actividades de levantamiento de requerimientos, instalación de ambientes, capacitación de personal, desarrollo de scripts de migración de datos y pruebas unitarias. La actividad que más me agrado y fue la más productiva fue la carga masiva de los datos.

El volumen de información administrado por la plataforma fue un factor importante en la decisión arquitectónica ya que la plataforma administraba los datos de inventario físico y lógico, además de flujos de trabajo y ciudades de operación. Esta plataforma es desplegada en Weblogic, y la base de datos que utiliza para operar es Oracle.

El alcance del proyecto consistió en la elaboración de casos de uso, el proceso de despliegue, incluyendo las configuraciones pertinentes para la operación del sistema de inventario.

## **5.3 Proyecto OSM**

En esta asignación el objetivo fue hacer la primera implementación de administrador de órdenes y servicios (OSM por sus siglas en ingles), a nivel América Latina. OSM se encuentra en una capa intermedia entre el CRM y los sistemas de aprovisionamiento e inventario. El cliente quería unificar sus plataformas para tener mayor control de los

procesos. Con esto las plataformas que alimentan OSM se encargan de mandar una orden de servicio, OSM está a cargo de procesarlas, transformarlas, retroalimentarlas y mandarlas a las plataformas respectivas.

Las actividades que realice en esta asignación fueron el levantamiento de requerimientos, la construcción de algunos módulos, instalación de todos los ambientes para administrador central de ordenes (COM por sus siglas en inglés), las pruebas unitarias e integrales del sistema y el monitoreo de órdenes.

El alcance de la asignación fue la implementación de más de 120 requerimientos en el cliente, la instalación de todos los ambientes, extensiva documentación del sistema y capacitación de personal para poder mantener la operación diaria.

Al implementar esta solución se quitó carga al sistema que genera las ordenes de servicio, esto fue bueno ya que el sistema que genera la orden, ahora solo se dedica a hacer una orden de servicio con las plataformas que necesita impactar, OSM se encarga de estar haciendo reintentos o verificar si la orden fue completada exitosamente en el sistema involucrado.

#### **5.4 Proyecto SNIIM**

Es una aplicación web hecha en capas basado en tecnología Java. En este proyecto el objetivo principal fue hacer cambios mayores al sistema legado y agregar funcionalidad. El cambio propuesto, fue hacer una mejora en la capa de presentación actualizando la versión del framework PrimeFaces, esta actividad se realizó para que la mayoría de los navegadores fueran compatibles con la aplicación, también cambio la capa de negocio para soportar nuevos servicios de consulta de datos.

Las actividades que realice en este proyecto, fueron el desarrollo de vistas, el desarrollo de nuevos componentes en la capa de negocio, levantamiento de requerimientos con el

cliente, la elaboración de planes de trabajo con el equipo, y consultoría de tecnologías para poder lograr tareas y actividades.

El alcance del proyecto fue tener una versión de SNIIM que fuera compatible con los navegadores más recientes, además de agregar algunos módulos que el cliente estaba requiriendo para nuevas vistas de datos.

Se hizo la construcción de un framework que construye vista de tablas dinámicas, esto fue necesario ya que el cliente necesitaba vistas más complejas de los datos que tienen en sus bases.

## **5.5 Proyecto Identidad por Huella Digital**

Este proyecto es una aplicación empresarial construida con tecnología Java. Su principal función es verificar la identidad de una persona en tiempo real. La arquitectura de la aplicación es en capas.

En esta asignación el objetivo fue dar soporte al sistema en producción y construir nueva funcionalidad a la aplicación.

Las actividades que realice fueron la construcción de mejoras en la capa de negocio, construcción de nuevos módulos, el diseño de esos módulos, las propuestas de mejora a la aplicación, la integración del sistema con las otras plataformas (descritas en la sección 10.5.3), dar seguimiento a actividades puntuales de los equipos de las otras plataformas, desarrollo de scripts para el despliegue automático de la pieza en producción, hacer pruebas unitarias e integrales y dar soporte en producción.

## **6 Metodologías utilizadas**

### **6.1 OUM Oracle Unified Method**

La metodología que Oracle usa en sus proyectos es OUM (Oracle Unified Method), esta metodología está diseñada para ser utilizada por todas las implementaciones de productos de Oracle. Ver referencia [14].

Esta metodología cuenta con las siguientes 5 etapas. Ver figura 7:

- Comienzo
- Elaboración
- Construcción
- Transición
- Producción

#### **6.1.1 Comienzo (Inception)**

En esta fase se definen los objetivos del proyecto, se contacta a los interesados para ver los alcances del proyecto. Esta fase es crítica para saber el esfuerzo que se va a realizar. Se empiezan a observar los requerimientos a alto nivel y los riesgos involucrados se deben entender antes de empezar el proyecto.

#### **6.1.2 Elaboración (Elaboration)**

La meta de esta fase es mover el desarrollo de la solución, el alcance y los requerimientos, de la fase Comienzo a desarrollar requerimientos detallados, particionar la solución, crear prototipos necesarios y hacer la base de la arquitectura del sistema, esto provee fundamentos para el diseño y esfuerzo de implementación en la fase de Construcción.



### **6.1.3 Construcción (Construction)**

El objetivo de esta fase es llevar el modelo detallado de los requerimientos, a través de configuraciones de funcionalidad, desarrollo y pruebas de componentes, para realizar la integración a los sistemas que ya han sido lanzados a producción. Completar el desarrollo del sistema aplicativo, validar que todos los componentes trabajen juntos y preparar el sistema para las pruebas de aceptación y el despliegue.

### **6.1.4 Transición (Transition)**

El objetivo de esta fase validar que el sistema es probado sistemáticamente y esté disponible para usuarios finales. Durante esta fase, el nuevo sistema es aceptado por la organización del cliente, esta se prepara para el nuevo sistema, y el sistema es puesto en producción, si el nuevo sistema reemplaza alguno viejo, se capacita a la gente para poder utilizar este nuevo sistema.

### **6.1.5 Producción (Production)**

El objetivo de esta fase es operar el nuevo sistema desarrollado, evaluar el éxito y monitorear problemas. Esto incluye que se actué apropiadamente para mantener la operación continua, medir el rendimiento del sistema, operar y mantener el soporte del sistema, responder a peticiones de ayuda, reportes de errores y peticiones de mejora del usuario, manejar el proceso de control de cambios para defectos, y nuevas funcionalidades pueden ser priorizadas y asignadas a futuras liberaciones.

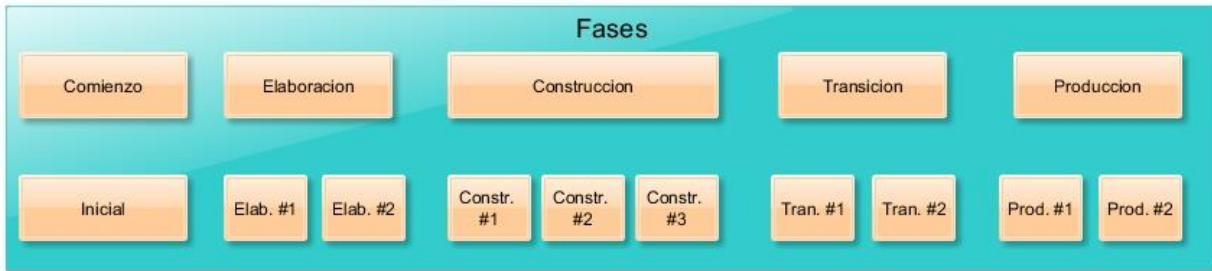


Figura 7: En el diagrama podemos observar las fases de la metodología con las iteraciones que se sugieren en la parte inferior. Basado en diagrama de referencia [18].

## 6.2 MAAGTICSI

MAAGTICSI es un manual administrativo de aplicación en materia de tecnologías de la información y de comunicaciones, así como de seguridad en los datos. Está basado en MAAGTIC, y sus procesos se basan en prácticas internacionales como Six Sigma, normas ISO/IEC 27,000, CMMI, entre otros.

Su objetivo es definir los procesos con los que en materia de TIC y de Seguridad de la Información, las instituciones deberán regular su operación, independientemente de su estructura organizacional y las metodologías de operación con las que cuenten. Ver referencia [13].

Sus etapas son las siguientes:

### 6.2.1 Planeación

Esta etapa es introductoria, sirve para ver de qué se trata el proyecto e ir definiendo alcances y requerimientos de alto nivel.

### **6.2.2 Análisis y Diseño**

La etapa de análisis y diseño es de las más importantes en el proyecto, se plasma lo que el cliente necesita. Estos documentos son usados como avales del trabajo cuando el cliente los firma. Al final del proyecto lo que está escrito en esos documentos, es lo que se tiene que entregar como producto final.

### **6.2.3 Construcción**

La etapa de construcción es donde se hace el desarrollo del sistema se hacen pruebas unitarias del componente.

### **6.2.4 Integración**

En esta etapa se hacen pruebas integrales con sistemas involucrados, y se realizan las pruebas de aceptación de usuario.

### **6.2.5 Aceptación del sistema**

Es cuando el cliente valida los requerimientos y da el visto bueno que el sistema hace lo que necesita. El proyecto es llevado a producción.

### **6.2.6 Control y seguimiento**

Esta etapa se hace durante todo el ciclo de vida del proyecto.

En la figura 8 podemos observar cómo está constituido el proceso de desarrollo de software según MAAGTICSI en sus 5 fases:

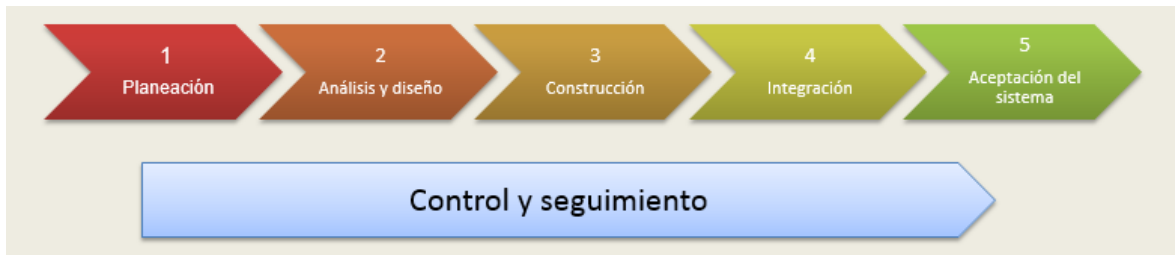


Figura 8: Diagrama de procesos en MAAGTICSI, como se observa en todas las fases debe seguirse la etapa de control y seguimiento

### 6.3 Softtek Software Development Process (SSDP)

El SSDP es usado en proyectos largos, legados o de mantenimiento. Esta metodología satisface los estándares de calidad de CMMI y Six Sigma. Ver referencia [9].

La metodología de Softtek consta de 4 etapas:

#### 6.3.1 Origen (Inception)

Es la etapa donde se hace la venta del proyecto se dan a conocer los primeros acercamientos de lo que se necesita y se ven posibles problemas para ver si es viable poder realizar la solución con el presupuesto y riesgos involucrados.

#### 6.3.2 Elaboración (Elaboration)

Esta etapa es para poder levantar los requerimientos del cliente, además se realizan los documentos de análisis y requerimientos de la solución y el documento de diseño funcional y técnico, así como también de la propuesta arquitectónica.

### **6.3.3 Construcción (Construction)**

Se hace el desarrollo del proyecto, se hacen las pruebas unitarias del código, se resuelven los problemas encontrados. En esta etapa se lleva el control de la versión de software que se va a liberar.

### **6.3.4 Transición (Transition)**

Esta etapa es para la liberación de las piezas de software en producción, también se encarga de capacitar personal para que pueda operar la nueva versión del sistema.

## **7 Resultados**

### **7.1 Resultados de mi participación en el proyecto ASAP**

El cartucho soportó los casos de uso que el cliente requería. Por lo que sustituyó en producción la implementación previa.

Al ser ASAP una plataforma para el aprovisionamiento, se mejoró el rendimiento de la plataforma en producción, quitando carga en el sistema que genera la orden. Con esto el soporte al sistema se hace de una forma más transparente y rápida.

El cliente quedó satisfecho con la solución lograda en el piloto, en consecuencia, se abrió el camino para posteriores mejoras, agregar funcionalidad al cartucho construido, y expandir el negocio.

### **7.2 Resultados de mi participación en el proyecto Metasolv**

En la asignación pude completar la migración de información que el cliente necesitaba en Metasolv, estos scripts son muy genéricos por lo que pueden ser reutilizados para migrar datos en otros proyectos, se logró hacer la instalación de los ambientes DEV, UAT y PROD, la plataforma sirvió para alojar los datos de inventario lógico y físico de redes.

Capacité personal para el uso de algunos módulos de la operación, el número de personal fue aproximadamente de 25 integrantes, la mayoría de ellos del área de sistemas. De esta experiencia pude aprender a tratar con las personas para poder lograr objetivos del proyecto.

### **7.3 Resultados de mi participación en el proyecto OSM**

Los resultados de mi participación en este proyecto fue la instalación exitosa de los ambientes de Sandbox, DEV, UAT, COB y PROD, de OSM en servidores del cliente en alta disponibilidad. El levantamiento de requerimientos con lo cual construimos más de 120 casos de usos, que fueron la base para la entrega del proyecto, la construcción de más de 150 casos de prueba para que la solución fuera probada.

### **7.4 Resultados de mi participación en el proyecto SNIM**

En este proyecto realice más de 200 cambios mayores y menores en el sistema legado, esto con el objetivo que la aplicación pudiera ser visualizada en navegadores más recientes y agregar nueva funcionalidad.

El cliente quedó satisfecho con las mejoras, entre ellas se incluye la elaboración de un componente dinámico para creación de tablas, este fue un componente genérico que funcionó para varias vistas y facilitó la presentación de los datos.

Como líder técnico de la solución, pude tomar los mejores caminos para llegar a las fechas que necesitábamos, así como las tecnologías que usamos para que el trabajo fuera más rápido y de calidad, además de dirigir al equipo para poder tener orden en las entregas de código que estábamos realizando.

El framework que genera tablas dinámicas fue completado con éxito en su totalidad, este componente fue utilizado para visualizar aproximadamente 75% de las vistas que el sistema ofrecía al usuario.

## **7.5 Resultados de mi participación en el proyecto Identidad por Huella Digital**

Los cambios aplicados al sistema en producción redujeron significativamente el tiempo de respuesta de la aplicación de 4 minutos a 40 segundos en el peor de los casos, además que en las siguientes versiones los cambios que propuse sirvieron para que el sistema soportara mayor volumen de peticiones, ampliándose de 5000 (en un principio) a 40000 tras las modificaciones a la arquitectura.

Como integrador tecnológico se llegó a tener una versión 2.1 del sistema, pude integrar la solución con las 6 plataformas (descritas en la sección 10.1.5.4) que estaban involucradas en el flujo de la información y cumplir con las fechas de entrega que se tenían contempladas. Esto no siempre se cumplía, por lo que se tenían que hacer negociaciones con el cliente para poder acordar nuevas fechas respecto a los problemas que se iban presentando, todo esto era posible ya que en el plan de trabajo se contemplaban tiempos para este tipo de situaciones.

Se completó la migración de la plataforma de Weblogic 10.3 a la versión 12, para esto se tuvo que hacer algunas adecuaciones en piezas de infraestructura que el cliente usa como interfaces entre sus productos.



## 8 Conclusiones

En este reporte profesional describí y analicé las arquitecturas de los sistemas con los cuales estuve trabajando en mi estancia por las industrias de telecomunicaciones, gobierno y sector bancario.

Lo que puedo concluir con respecto a la industria de las telecomunicaciones es que tienen muy bien definido los siguientes subsistemas, tienen a sus clientes en el CRM, su inventario (lógico y físico), su aprovisionamiento ASAP, su facturación BRM y algunas veces su orquestación OSM, lo que hace que la arquitectura también este muy bien dividida.

En la industria de las telecomunicaciones muchos proveedores de servicios ofrecen a los clientes sus productos, con la finalidad de vender un sistema que tiene un fin común, este se adecua a los subsistemas que menciono en el párrafo anterior. Estos productos se adaptan a las necesidades específicas de cada cliente, en la medida que el producto ofrezca mayor flexibilidad para integrarse con otras plataformas, este será más atractivo para el cliente.

En cuanto a las arquitecturas tecnológicas de este sector vemos una separación de responsabilidades muy bien establecida, lo que hace sencillo poder integrar estos productos. Pude observar que para este tipo de plataformas los clientes compran servidores dedicados a cada subsistema para que el rendimiento y la disponibilidad sean los mejores posibles en la operación.

Mientras tanto en sector gobierno no tienen una definición de negocio tan marcada, ya que cada dependencia de gobierno tiene un fin diferente y no hay convergencia entre los requerimientos de una dependencia a otra, por lo que los sistemas que son usados, tienden a ser un producto que es hecho por consultoras, desarrollados a la medida de cada dependencia, lo que lo convierte en un sector con muchos tipos de sistemas y abre la posibilidad a nuevos desarrollos y proyectos de integración.

En el sector bancario observe que tienen negocios muy específicos y aunque un nuevo proyecto se proponga, tratan de adecuarlo a la arquitectura ya planteada del banco, esto se debe a que hay proveedores que le dan ciertos servicios y lo único que hacen es reutilizar esos servicios para poder crear nuevos proyectos. Los proyectos en donde el cambio de la arquitectura rompe con los estándares del banco, son experimentales y están desplegados en ambientes aislados para ir haciendo pruebas sobre la tecnología. Aquí me di cuenta que los recursos internos del banco son compartidos entre las plataformas que tengan interés por cierto servicio.

En las arquitecturas del sector de las telecomunicaciones y sector bancarias observe que son muy parecidas en cuestión de su operación diaria como un CRM, un sistema de inventarios, pero observo que en los bancos se tienen muchos sistemas base que son muy difíciles de ser reemplazados. Estos sistemas llevan años operando de forma correcta y son muy difíciles de migrar a plataformas modernas, pero tienen un buen rendimiento en producción, por lo tanto, son conservados y son parte fundamental del negocio, mientras que en la industria de telecomunicaciones observo que se tiene más iniciativa a hacer cambios para tener tecnología de punta en sus servicios y soportar la operación diaria.

En arquitecturas del sector gobierno veo que al momento de integrar es significativamente más complicado, ya que esas interfaces de integración deben ser construidas y definidas para que la comunicación se establezca. Mientras que en las otras dos industrias existen productos y estándares que establecen esas interfaces de comunicación, haciendo que el tiempo de integración sea menor.

## 9 Bibliografía

1. Sommerville, Ian. (2011). Ingeniería de Software. México: PEARSON EDUCACIÓN.
2. Mark Richards. (February 2015). Software Architecture Patterns. United States of America: O'Reilly.
3. Richard N. Taylor, Nenad Medvidovic, Eric M. Dashofy. (January 2009). Software Architecture: Foundations, Theory, and Practice. United States of America: Wiley.
4. Kai Qian, Xiang Fu, Lixin Tao, Chong-Wei Xu, Jorge I. Diaz-Herrera. (February 2009). Software Architecture and Design Illuminated. Canada: Jones & Bartlett Learning; 1 edition.
5. Len Bass, Paul Clements, Rick Kazman. (September 2012). Software Architecture in Practice (SEI Series in Software Engineering). United States of America: Addison-Wesley Professional; 3 edition.
6. Shari Lawrence Pfleeger, Joanne M. Atlee. (2010). Software Engineering Theory and Practice. United States of America: Prentice Hall College Div; Fourth Edition.
7. Roger S. Pressman. (2010). Software Engineering A practitioner's Approach. United States of America: Prentice Hall College Div; Seventh Edition
8. Oracle. (2016). Visión. 1/03/2016, de Oracle Sitio web: <http://www.oracle.com/us/corporate/careers/diversity/overview/index.html>
9. Softtek. (2016). Visión. 1/03/2016, de Softtek Sitio web: <http://www.softtek.com/es/acerca-de-softtek/historia>
10. Negocios Digitales. (2016). Visión. 1/03/2016, de Negocios Digitales Sitio web: <http://www.negocios-digitales.com.mx>
11. Wikipedia. (2014). Applications architecture. 30/01/2016, de Wikipedia Sitio web: [https://en.wikipedia.org/wiki/Applications\\_architecture](https://en.wikipedia.org/wiki/Applications_architecture)
12. Wikipedia. (13/02/2016). Architectural pattern. 25/02/2016, de Wikipedia Sitio web: [https://en.wikipedia.org/wiki/Architectural\\_pattern](https://en.wikipedia.org/wiki/Architectural_pattern)
13. Secretaria de Función Pública. (2014). MAAGTICSI. 15/01/2016, de Secretaria de Función Pública Sitio web: <http://cidge.gob.mx/cursos/curso042014-1/manual.html#gobernanza>

14. Oracle. (2016). OUM. 1/03/2016, de Oracle Sitio web:  [<http://www.oracle.com/partners/en/products/applications/oracle-unified-method/get-started/index.html>](http://www.oracle.com/partners/en/products/applications/oracle-unified-method/get-started/index.html)
15. Oracle. (2012). ORACLE COMMUNICATIONS ASAP. 10/03/2016, de Oracle Sitio web:  [<http://www.oracle.com/us/industries/communications/oracle-communications-asap-ds-1596315.pdf>](http://www.oracle.com/us/industries/communications/oracle-communications-asap-ds-1596315.pdf)
16. Oracle. (2012). ORACLE COMMUNICATIONS METASOLV SOLUTION. 10/03/2016, de Oracle Sitio web:  [<http://www.oracle.com/us/industries/communications/communications-metasolv-ds-398152.pdf>](http://www.oracle.com/us/industries/communications/communications-metasolv-ds-398152.pdf)
17. Oracle. (2012). ORACLE COMMUNICATIONS ORDER AND SERVICE MANAGEMENT. 10/03/2016, de Oracle Sitio web:  [<http://www.oracle.com/us/industries/communications/050733.pdf>](http://www.oracle.com/us/industries/communications/050733.pdf)
18. Oracle. (2014). Oracle's Full Lifecycle Method for Deploying Oracle-Based Business Solutions. 10/03/2016, de Oracle Sitio web:  [<http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method-069204.pdf>](http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method-069204.pdf)
19. Oracle. (2010). The Java EE 5 Tutorial. 18/02/2016, de Oracle Sitio web:  [<http://docs.oracle.com/javaee/5/tutorial/doc/bncdx.html>](http://docs.oracle.com/javaee/5/tutorial/doc/bncdx.html)

## 10 Anexos

### 10.1 Detalle técnico de la participación profesional

#### 10.1.1 Proyecto plataforma ASAP

Mi primera asignación fue ASAP, esta plataforma se encarga del aprovisionamiento de servicios. El funcionamiento de ASAP es como el de una cola básica donde se mandan las peticiones y estas son liberadas hasta que el servicio es exitosamente aprovisionado, esta plataforma forma parte de la suite de plataformas que CGBU tiene para vender a sus clientes. Oracle tiene una suite de programación donde a través de eclipse y un Plug-in llamado “Design Studio for ASAP”, se realiza la programación de Cartuchos (“Cartridges”), que incluyen la lógica de negocio del cliente, estos cartuchos son programados con el lenguaje de programación Java.

El objetivo del proyecto era liberar carga del generador de órdenes, ya que antes se aprovisionaba directamente en la OTL, si se llegaba a tener fallos, el mismo sistema que generaba la orden se encargaba de hacer la lógica necesaria para seguir el proceso. El diagrama de arquitectura se puede observar en la figura 9.

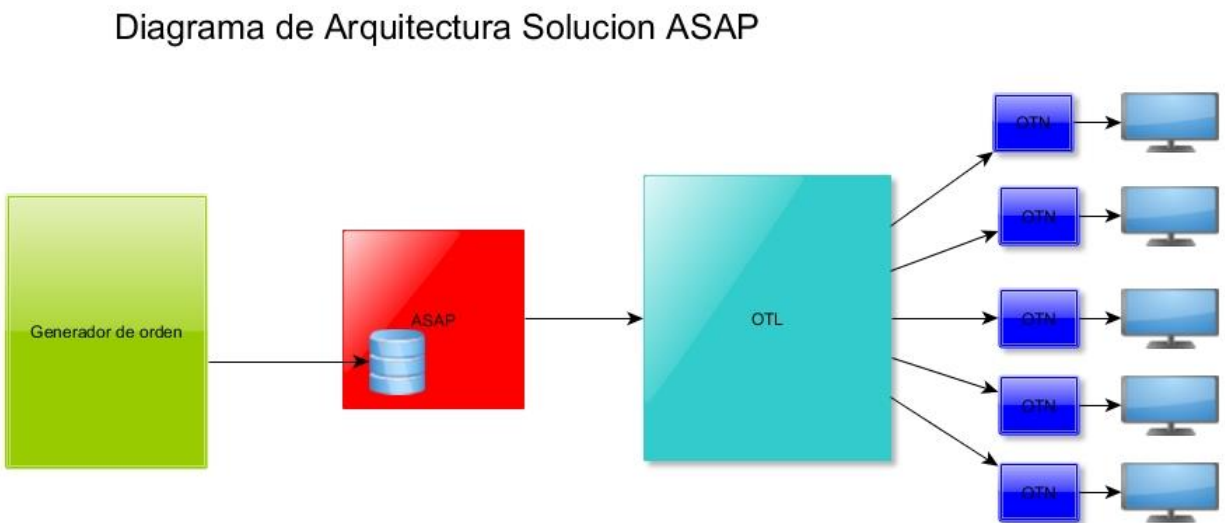


Figura 9: Diagrama de arquitectura de la implementación de ASAP.

### 10.1.1.1 ¿Qué es ASAP?

Es una plataforma que provee un servicio convergente de activación que automáticamente activa servicios de usuario en redes heterogéneas y ambientes de IT. Soporta la activación de servicios de consumidor, de negocio y de dominios de móviles a través de redes de aplicación de IT.

ASAP funciona como una aplicación J2EE desplegada en el servidor de aplicaciones Weblogic, y a través de los cartuchos cobra funcionalidad de negocio. Como es desplegado en un servidor de aplicaciones este puede estar en alta disponibilidad y proporcionar un mejor servicio gracias a las configuraciones que este adquiere por medio del servidor de aplicaciones, en este caso Weblogic.

ASAP se conforma de las siguientes características:

- **Diseño de Activación Rápido (Rapid activation design en inglés):** quiere decir que es altamente configurable, ASAP puede ser configurado por Design Studio, para ensamblar, configurar y desplegar la solución. El tiempo de diseño es muy sencillo y rápido en el IDE Design Studio para ASAP.
- **Entrega de orden automática (Automated order delivery en inglés):** ASAP provee un rendimiento alto de activación de servicios en la plataforma, recibe órdenes de activación de entrada, se descomponen en la activación de los paquetes de acciones de servicio y acciones de redes atómicas, y luego se ejecutan las acciones de red y aplicación de IT requeridas, automáticamente activan todos los servicios de la orden recibida. El contexto de ejecución de ASAP con otras plataformas de Oracle es como se muestra en la figura 10.

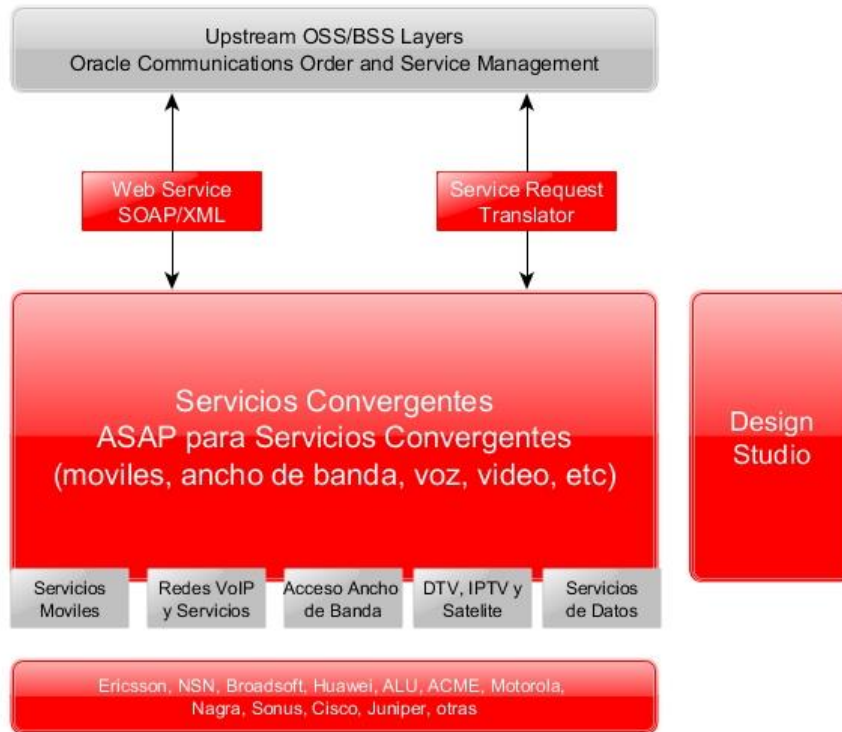


Figura 10: Diagrama de arquitectura plataforma ASAP, se puede observar algunas interfaces que tiene para interactuar con otras plataformas. Basado en diagrama de referencia [15].

### 10.1.2 Proyecto plataforma MetaSolv Solution

Mi segunda asignación fue con el equipo de MetaSolv Solution, esta es otra aplicación J2EE que se despliega sobre un servidor de aplicaciones Weblogic, su propósito es mantener el inventario físico y lógico de las redes, además como proporcionar flujos de trabajo a los usuarios de la plataforma. Este sistema mantiene el mapeo de troncales digitales y otras conexiones del cliente.

### **10.1.2.1 ¿Qué es MetaSolv Solution?**

Es una plataforma que proporciona a los proveedores de servicios de red una integración de la solución que incluye ordenamiento, flujo de trabajo, gestión de inventario y capacidad de gestión de problemas. El foco del aplicativo es proporcionar funcionalidad y capacidades fuera de la caja con soporte para regular estándares clave, esto para habilitar la rápida introducción al servicio, y entrega eficiente a un bajo costo.

MetaSolv Solution tiene 6 módulos:

**Customer Care:** Este módulo provee ordenes básicas de entrada basadas en el catálogo de MSS, como una vista unificada de los detalles del cliente incluyendo el estatus de órdenes, servicios y problemas. Como modulo opcional, los proveedores de servicios pueden elegir usarlo con un CRM como Siebel, para administrar la información de usuario dependiendo el requerimiento. Ver figura 11.

**Order Management:** Este módulo provee las siguientes capacidades:

- Provisionar el catálogo de productos, con su clasificación flexible de definición, ordenando requerimientos de datos.
- El cliente puede ordenar vía Product Service Request (PSR), para habilitar el ordenamiento de una gran variedad de servicios actuales y nuevas tecnologías.

**Work Management:** Este módulo define trayectos y tendencias para todas las tareas en el servicio o el proceso de aprovisionamiento de red, provee un proceso visible y comprensible que provisiona métricas. Estas actividades habilitan el diseño de planes de provisión personalizados, que organiza y administra los flujos de tareas, ambos automáticos y manuales, esto acorde al proceso de negocio.

**Data Management:** Este componente incluye un repositorio de datos común de toda la información que está administrada por las aplicaciones, junto con un set comprensible de administración y reportes de operación.



Network Design: Este módulo soporta diferentes niveles de información de inventario, geográfica, física y lógica. MSS soporta una gran variedad de tecnologías de red y provee una serie de módulos de tecnología para habilitar el rápido, y bajo riesgo de despliegues.

Service Provisioning: Este módulo soporta el diseño gráfico y proceso de asignación, del diseño de redes complejas arbitrarias. Esto provee el análisis de manera inteligente para determinar el camino disponible a través de la red con mínimas configuraciones.

Trouble Management: Este módulo es un componente opcional, en el que los tickets con problemas pueden ser creados y darles el seguimiento hasta ser cerrados.

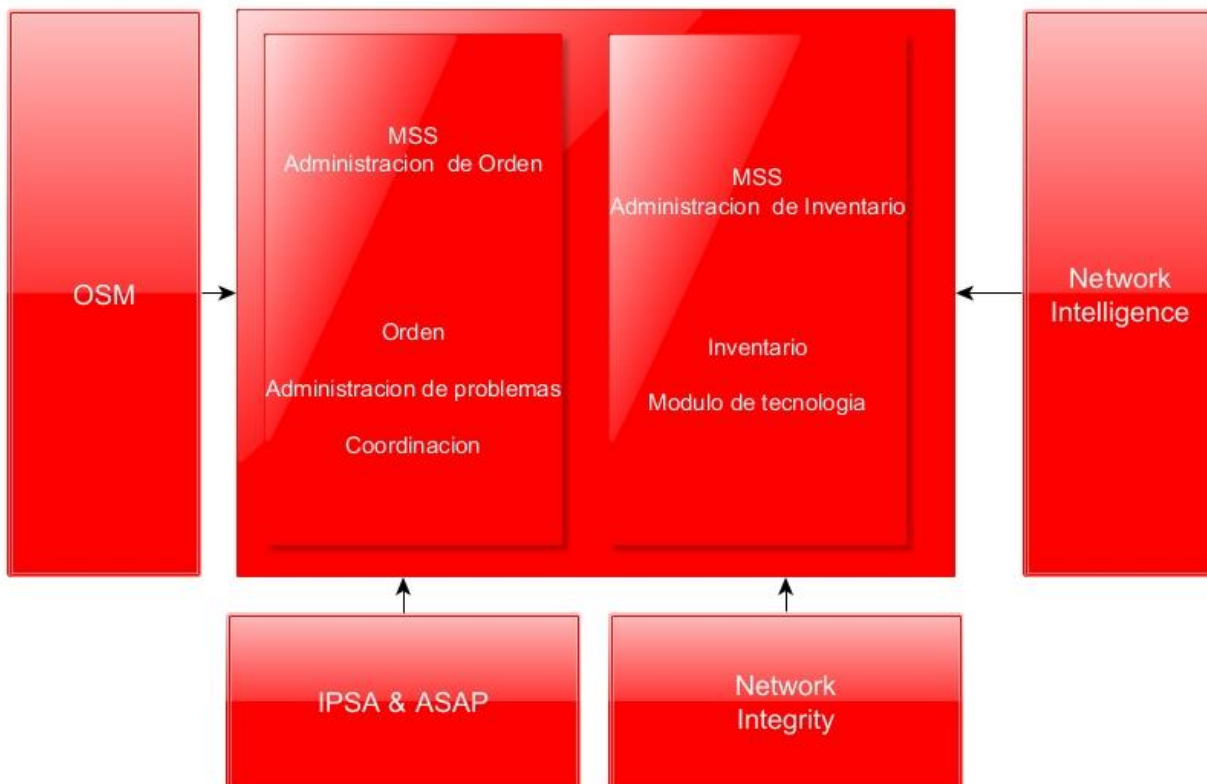


Figura 11: Aquí podemos observar como otras plataformas que pertenecen a Oracle pueden integrarse con Metasolv, esto mediante las interfaces que posee. Basado en diagrama de referencia [16].

### 10.1.3 Proyecto plataforma OSM

Mi tercera asignación fue con el equipo de OSM (Order and Service Management), esta es una aplicación J2EE que es desplegada en el servidor de aplicaciones Weblogic, tiene como función principal la distribución de las ordenes a los diferentes sistemas involucrados, esto quiere decir que es el orquestador de las peticiones. En este caso se necesitaba una forma de centralizar ordenes con las plataformas que dan servicio, por lo tanto, se decidió utilizar el sistema OSM. Ver figura 12.

Diagrama de arquitectura proyecto OSM

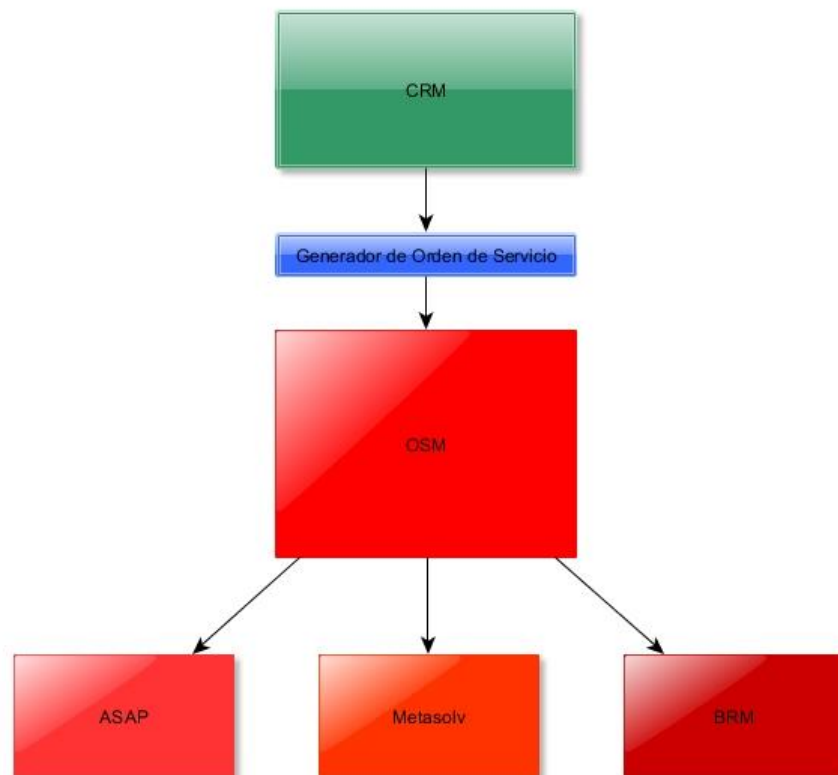


Figura 12: Diagrama de arquitectura de la solución OSM, se puede observar que interactúa con varios sistemas, la orden es generada desde el Generador de Orden de Servicio, OSM se encarga de impactar los sistemas que estén involucrados en la orden.

### **10.1.3.1 ¿Qué es OSM?**

Order and Service Management es construido para ayudar a los clientes en el largo camino de retos del ciclo de vida end-to-end del manejo de órdenes, empezando del diseño de la oferta, el lanzamiento, todo el camino a través de las ordenes y activación de la facturación. Sus capacidades para la sincronización de la oferta y el diseño de servicios, automatiza el manejo de órdenes fallidas y en tiempo real el cambio de orden de servicio.

OSM es uno de los componentes principales de Oracle Communications Solutions, como sigue:

- Diseño de Oferta Rápida y Entrega de Orden (Rapid Offer Design and Order Delivery en inglés y sus siglas RODOD): Es una solución de Oracle con capacidades para proveer funciones al Manejador de Orden Central (Central Order Management), ver figura 13, este soporta las siguientes características:
  - Habilidad para separar productos comerciales de los flujos de manera rápida.
  - Habilidad para separar la topología de los flujos para una gran flexibilidad en IT.
  - Un bajo de tiempo de diseño, integración con Siebel para sincronizar definiciones de clases de productos para minimizar el fallo de órdenes.
  - En tiempo de entrega, tiene la habilidad de descomponer dinámicamente y orquestar ordenes de clientes, disparar tickets de problemas para resolución de ordenes fallidas y proveer en tiempo real, visibilidad de estatus de órdenes.

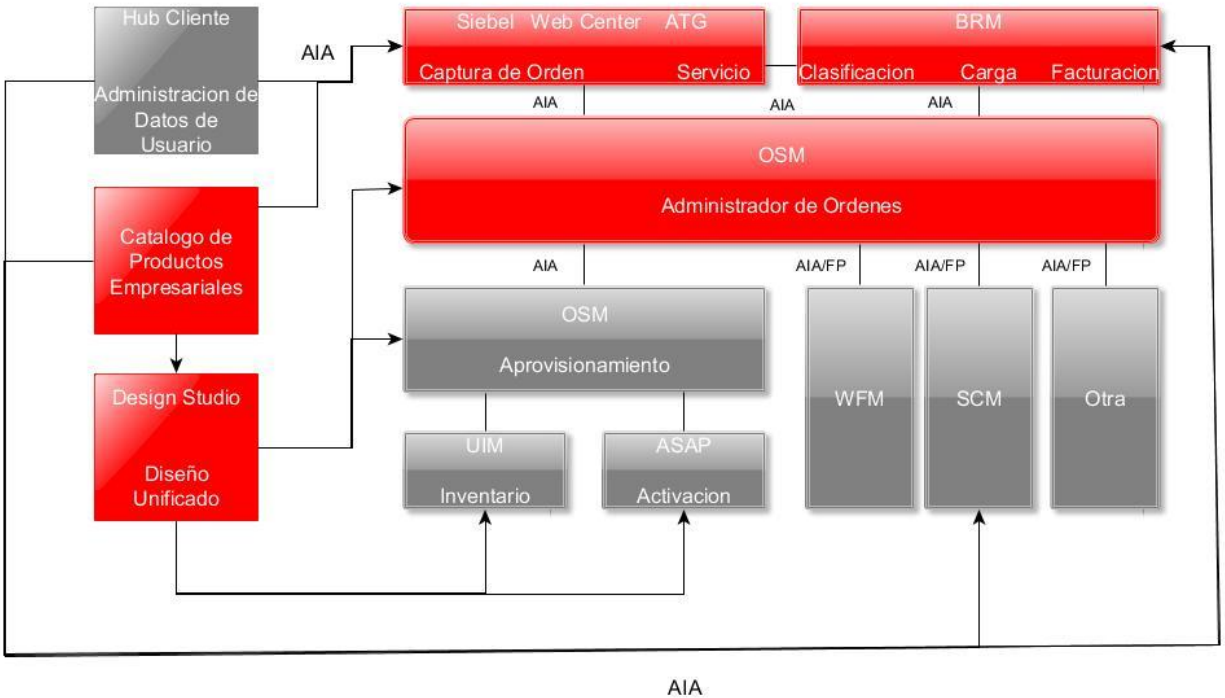


Figura 13: Diagrama que muestra la solución RODOD, donde los componentes en rojo son los involucrados. Basado en diagrama [17].

- Diseño de Servicio Rápido y Entrega de Orden (Rapid Service Design and Order Delivery en inglés y sus siglas RSDOD): Es una solución de Oracle, en conjunto con ASAP, ver figura 14, soporta las siguientes capacidades:
  - En tiempo de diseño de servicios, hace definición flexible de los patrones de soporte catálogos.
  - En tiempo de captura de orden, hace soporte para calificación de servicio técnico, reservaciones, etc.
  - En tiempo de entrega de orden, la habilidad de descomponer dinámicamente y orquestar órdenes de servicio a través de uno o más dominós, coordina el diseño y la asignación y activación de actividades requeridas para levantar los servicios y dar soporte a los requerimientos de COM.

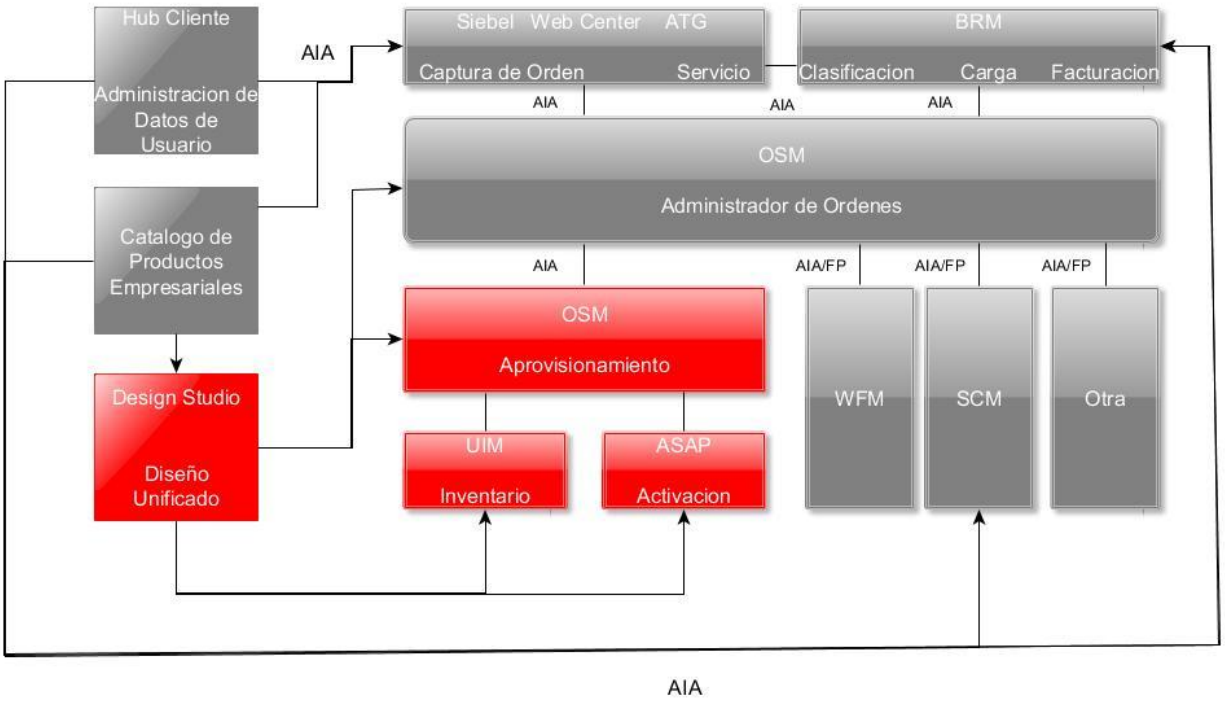


Figura 14: Diagrama que muestra la solución RSDOD, los componentes en rojo son los involucrados. Basado en diagrama [17].

#### **10.1.4 Proyecto SNIIM (Sistema Nacional de Información e Integración de Mercados)**

Esta asignación fue para hacer una migración de un sistema legado, además de agregar nueva funcionalidad en la capa de negocio. Fue mi única asignación con el equipo de Negocios Digitales.

##### **10.1.4.1 *Acerca de la Secretaría de Economía***

La secretaria de economía es una dependencia del gobierno, que se encarga de regularizar los precios de los productos, las importaciones y exportaciones del país en varios de los sectores principales, ya sea comercio minero, agropecuario, pesquero, etc.

##### **10.1.4.2 *¿Qué es SNIIM?***

Es el Sistema Nacional de Información e Integración de Mercados, es un servicio de la Secretaría de Economía que tiene el propósito de ofrecer información sobre el comportamiento de los precios al por mayor de los productos agrícolas, pecuarios y pesqueros que se comercializan en los mercados nacionales e internacionales.

##### **10.1.4.3 *Tecnologías involucradas en proyecto SNIIM***

El proyecto estaba en una fase avanzada, lo que se necesitó fue agregar funcionalidad y hacer una actualización de la versión del framework que se estaba utilizando en el Front-end, esto para que la aplicación fuera visualizada en navegadores en sus últimas versiones.

A continuación, listare las tecnologías asociadas a esta aplicación:

- J2EE 5
- Java SDK 7

- JavaServer Faces 2.0
- PrimeFaces 3.5
- MyBatis 3
- EJB 3.0
- Spring 3.2

El servidor de aplicación donde se desplegó la aplicación es Glassfish en su versión 3.2.2.

La instancia de base de datos que se designó a este proyecto fue Oracle en su versión 11g.

#### **10.1.4.4 Descripción de arquitectura de la aplicación**

Esta aplicación tiene una arquitectura en capas y es orientada a objetos, está basada en tecnología Java, el servidor de aplicaciones que se utiliza es Glassfish, se basa en la especificación J2EE en su versión 5 y el Front-end estaba construido con JavaServer Faces y PrimeFaces.

La parte Back-end tenía servicios de negocios EJB en su versión 3, utilizaba Spring para hacer inyección de dependencias además del módulo de seguridad para resguardar la aplicación, en la capa de persistencia de datos y consultas se utilizaba el framework MyBatis. Ver figura 15.

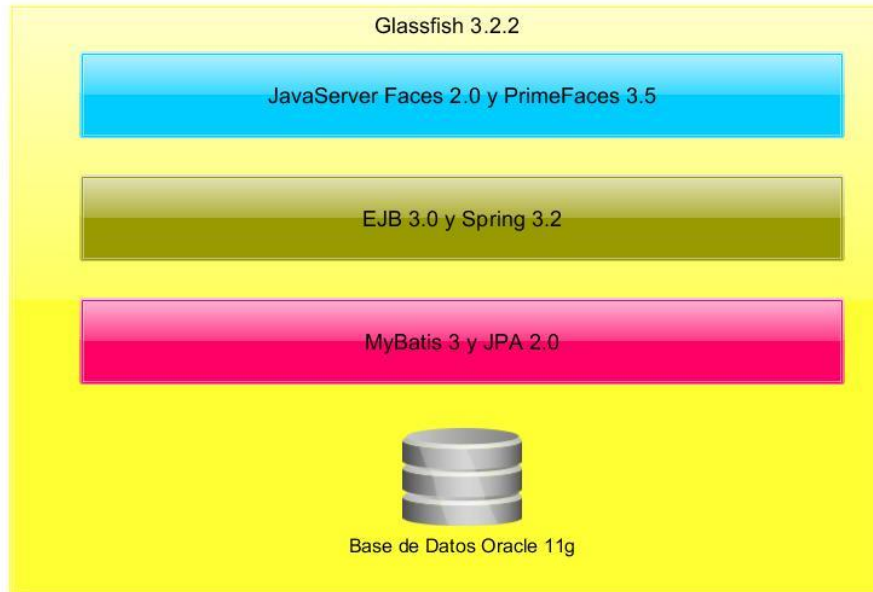


Figura 15: Diagrama de arquitectura en el proyecto SNIIM. Es una aplicación MVC, donde tenemos la capa de presentación en color azul, la capa de servicios en color café y la capa de persistencia de datos en color rosa.



### **10.1.5 Proyecto Identidad por Huella Digital**

Este fue el primer proyecto con el que estuve trabajando cuando me incorporé a Softtek, era un sistema que ya estaba en su fase piloto y estaba siendo probado en ciertas sucursales, el objetivo era ver que tan eficiente era en producción. La aplicación consistía en un Front-end que está hecho en Java, donde se insertan los datos de la credencial de elector, como nombre, apellido, año de nacimiento, el OCR (es un número que da el INE a cada uno de los mexicanos), las huellas digitales del cliente, y al final esta información es consultada en los servicios del INE, se regresa si los datos enviados son correctos o hay alguna anomalía de la información enviada. Con esto se valida la identidad de la persona.

#### **10.1.5.1 *¿Qué es Identidad por Huella Digital?***

Identidad por Huella Digital es una plataforma que sirve para la identificación de identidad de los clientes del banco, esta idea surgió como un medio más para poder detectar fraudes en tiempo real. Mi participación fue para hacer la segunda fase y dar soporte a ese piloto. Esta aplicación está alojada en Weblogic.

Uno de los inconvenientes es que, en algún momento del proceso de consulta, se manda una petición hacia el consumo de Web Services del INE, esto con la finalidad de consultar si el cliente es legítimo o hay alguna incongruencia con la información que se está proporcionando, al ser un servicio proporcionado por un tercero se tiene una fuerte dependencia en cuanto a la calidad de respuesta que el servicio ofrece, como el consumo de los servicios web era lento hacia que la respuesta en el Front-end no fuera eficiente.

#### **10.1.5.2 *Identidad por Huella Digital Versión 1***

La primera versión del sistema tenía un consumo de servicios web ineficiente, la respuesta al Front-end tardaba hasta cuatro minutos en el peor de los casos.

La arquitectura de la aplicación la describo en la figura 16.

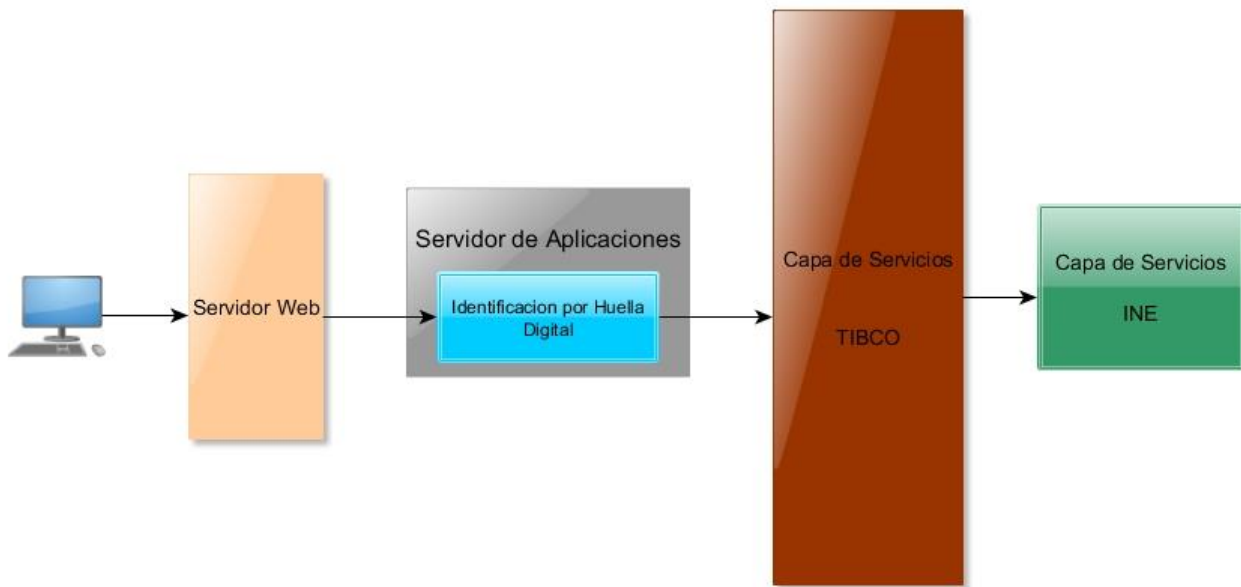


Figura 16: Diagrama de arquitectura del sistema Identificación por Huella Digital. El llamado al servicio web era directo en el módulo de la aplicación.

### 10.1.5.3 *Identidad por Huella Digital Versión 2*

En la segunda versión del sistema el objetivo fue guardar la información que era consultada por los ejecutivos en las sucursales, por lo que se tuvieron que hacer cambios técnicos y a nivel de arquitectura de los componentes.

Uno de los primeros cambios fue dividir el componente que realizaba las consultas, este se encontraba embebido en el sistema, pero paso a ser un Stateless EJB de negocio que podía ser consumido desde un cliente o desde el mismo servidor de aplicaciones. La parte de TIBCO cambio para estar en alta disponibilidad. Para guardar los datos que se consultaban, se utilizó un Data Warehouse llamado Teradata, este proceso se realizaba mediante carga masiva de datos con scripts en las madrugadas, por lo que no se consideró para que guardara los datos en tiempo real. Para este propósito se utilizó una plataforma llamada CM. Con este sistema es posible guardar los datos en tiempo

real en una base de datos con tecnología Oracle, en las madrugadas se corre un script que extrae los datos de las tablas en Oracle y las pone en un archivo con un formato especial, este archivo es transferido a un repositorio central, que es un punto compartido en toda la infraestructura del cliente, donde luego es tomado por Teradata que hace la carga masiva de los datos mediante un ETL. Ver figura 17.

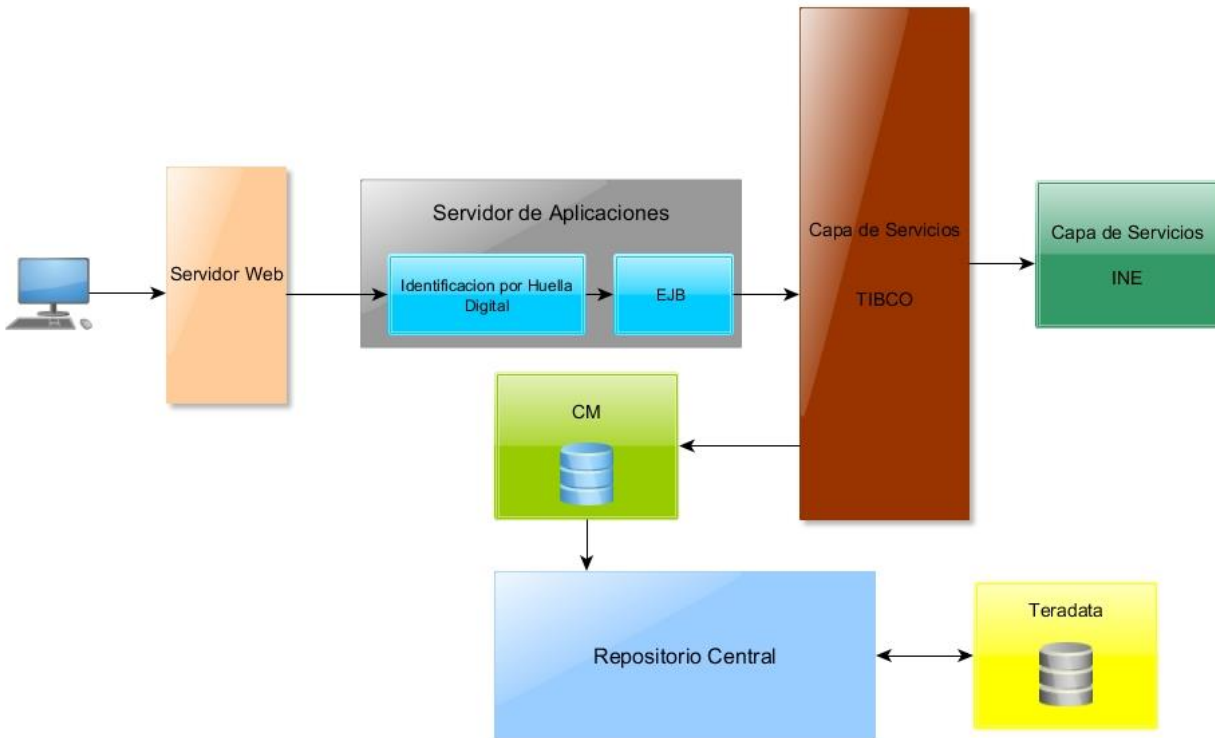


Figura 17: Diagrama de arquitectura de Identidad por Huella Digital en su versión 2. Aquí se implementó un EJB para hacer más robusto el servicio, se controló el timeout hacia el llamado del servicio web del INE, y se implementó la solución para guardar los datos de consulta en la base de datos.

#### 10.1.5.4 Identidad por Huella Digital Versión 2.1

En la última versión de Identidad por Huella Digital se reemplazó el componente EJB por un Web Services, esta decisión fue tomada porque el cliente que consumía el EJB utilizaba RMI, el proceso de llamado al servicio de negocio era lento, al hacer el cambio el llamado fue más rápido. Ver figura 18.

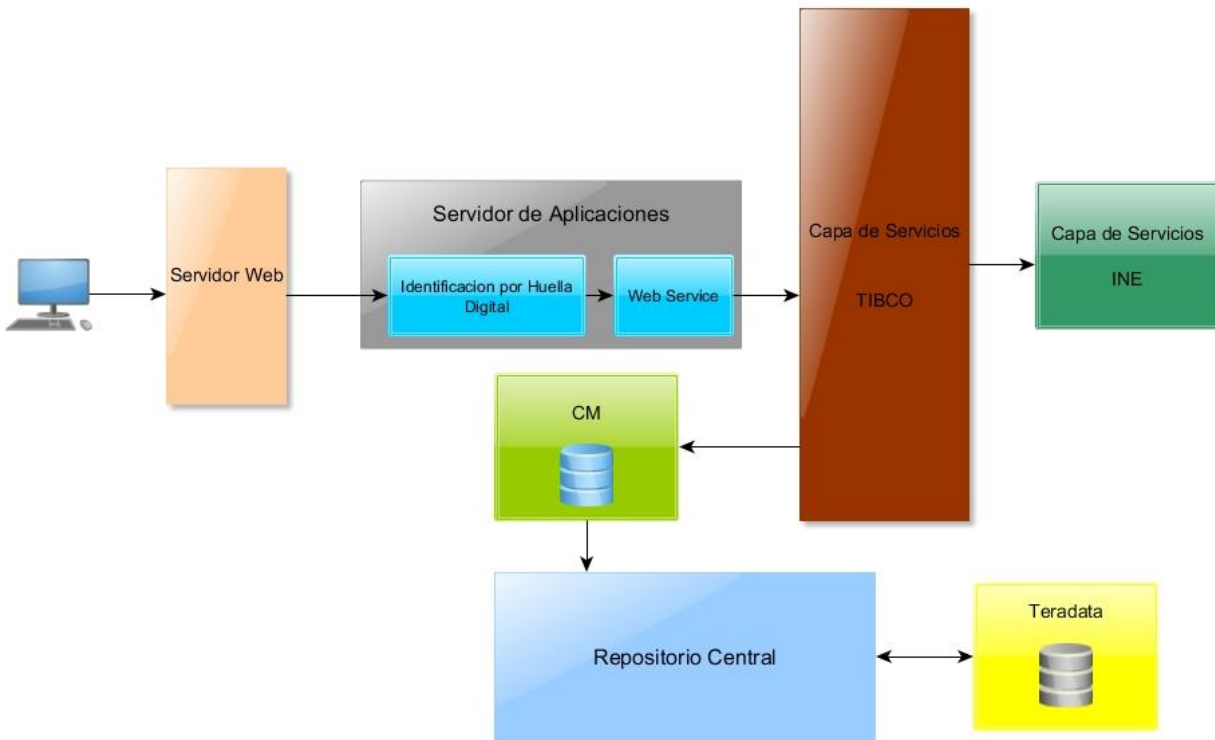


Figura 18: Se reemplazó el componente EJB por un Web Service.

#### 10.1.5.5 *Identidad por Huella Digital Migración Weblogic 10 a Weblogic 12c*

La migración del sistema se hizo mediante el cambio de compilación de SDK de la versión 6 a la versión 7 y la especificación de J2EE 6, con esto se pudieron utilizar nuevos APIs, que sirvieron para hacer cambios menores a la pieza y se pudieron desarrollar nuevas funcionalidades.

## 10.2 Glosario

- ASAP: Automated Service Activation Platform
- Back-end: Se le denomina de esta forma a las aplicaciones que están corriendo en el servidor y hacen la mayor parte de la lógica del negocio, estas viven comúnmente en un Servidor de Aplicación que provee ciertos servicios que facilitan la operación de estos servicios de negocio.
- BIRT: Plataforma que se encarga de crear reportes que pueden ser incluidos en clientes y aplicaciones web.
- BRM: Billing Revenue Manager.
- BSS: Business Support System.
- CGBU: Communications Global Business Unit.
- COM: Central Order Management.
- EJB: Enterprise Java Beans.
- ETL: Extract, transform and load.
- Front-end: Se le dice así a la mayoría de las aplicaciones donde interactúa el usuario final, estas pueden ser basadas en WEB, o pueden ser interfaces Stand-alone que son en su mayoría programas que corren en la PC del usuario final.
- Glassfish: Servidor de aplicaciones.
- HTML: HiperText Markup Language.
- INE: Instituto Nacional Electoral.
- JavaServer Faces: Es una tecnología basada en eventos para realizar interfaces web dinámicas, su característica es que necesita un bean para poder hacer el control de las pantallas, este objeto se encarga de guardar los estados y puede ser completamente dinámico.
- JDK: Java Development Kit.
- JEE: Java Enterprise Edition.
- MSS: MetaSolv Solution.
- OSM: Order and Service Management.
- OSS: Operational Support System.
- OUM: Oracle Unified Method.

- PrimeFaces: Es una extensión de JavaServer Faces que provee componentes que tienen funcionalidades muy vistosas, hacen que el proceso de captura de datos sea más dinámico y ordenado.
- RMI: Remote Method Invocation.
- RODOD: Rapid Offer Design and Order Delivery.
- RSDOD: Rapid Service Design and Order Delivery.
- Sandbox: Ambiente de pruebas, se le dice caja de arena, haciendo referencia a que puedes jugar en él y no afecta en nada a los ambientes que se utilizan para el desarrollo del proyecto.
- Servidor de Aplicaciones: El servidor de aplicaciones tiene como función principal alojar las piezas que son desarrolladas con la especificación J2EE, este tiene todo lo necesario para poder comunicar el código, desplegarlo, ponerlo en alta disponibilidad, clusterizar la aplicación y prácticamente se asegura de proveer el servicio.
- SOM: Service Order Management.
- SSDP: Softtek Software Development Process.
- Struts: Tecnología usada para creación de páginas web dinámicas, tiene como característica principal estar basado en acciones, estas son configurables mediante un archivo XML, esto permite el fácil mantenimiento de código.
- UAT: User Acceptation Test.
- Weblogic: Servidor de Aplicaciones que provee Oracle.
- WSDL: Web Services Descriptor Language.