



**SMS Chat**  
**Documento de Línea base de Arquitectura de Software**

**Versión 1.0**

### Historial de Revisiones

<u>Fecha</u>	<u>Versión</u>	<u>Descripción</u>	<u>Autor</u>
12/07/2012	1.0	Versión inicial	Dany E. Vásquez
16/08/2012	1.1	Normalizaciones realizadas a los diagramas de secuencia.	Dany E. Vásquez

### Historial de Observaciones

<u>Fecha</u>	<u>Versión</u>	<u>Observaciones</u>

## **Tabla de Contenido**

<b>1. INTRODUCCIÓN</b>	<b>4</b>
1.1 PROPÓSITO	4
1.2 ALCANCE	4
1.3 DEFINICIONES, ACRÓNIMOS Y ABREVIACIONES	4
1.4 REFERENCIAS	4
<b>2. OBJETIVOS Y RESTRICCIONES DE LA ARQUITECTURA</b>	<b>5</b>
2.1 OBJETIVOS:	5
2.2 REQUERIMIENTOS ESPECIALES	5
<b>3. MECANISMOS DE ARQUITECTURA</b>	<b>7</b>
<b>4. VISTA DE CASOS DE USO</b>	<b>8</b>
<b>5. VISTA DINÁMICA</b>	<b>10</b>
<b>6. VISTA LÓGICA</b>	<b>25</b>
6.1 INTRODUCCIÓN	25
6.2 ENTIDADES	25
6.3 PAQUETES DE DISEÑO EN LA ARQUITECTURA	26
<b>7. VISTA DE DESPLIEGUE</b>	<b>27</b>
7.1 INTRODUCCIÓN	27
7.2 DISTRIBUCIÓN PARA APLICACIÓN DE ADMINISTRACIÓN DE PARÁMETROS	27
7.3 DISTRIBUCIÓN PARA APLICACIÓN DE SMS CHAT	27
7.4 DISTRIBUCIÓN PARA SERVICIO UTILITARIO LOGGER	28
<b>8. VISTA DE COMPONENTES</b>	<b>28</b>

---

## **Documento de Arquitectura de Software**

### **1. Introducción**

La definición de una Arquitectura de Software aporta en particular una visión abstracta de alto nivel al realizarse el diseño, que pospone cada uno de los módulos definidos a los pasos posteriores del mismo.

Por lo que al esbozar el diseño, se debe tomar mucha preponderancia en el momento de fijar una arquitectura y así tomarla como algo clave a la hora de diseñar una solución.

#### **1.1 Propósito**

El *Documento de Arquitectura de Software* presenta la arquitectura del proyecto *SMS CHAT* a través de diferentes vistas, las que mostrarán cada una de las características particulares y propiamente dichas de la aplicación a desarrollar. Con esto, lo que se propone es que por medio de este documento, se tenga una clara visión del diseño de esta aplicación, lo cual podría ser de mucha utilidad en el mantenimiento posterior.

#### **1.2 Alcance**

Este documento se concreta, de manera puntual, en el desarrollo de la vista lógica, vista dinámica y de implementación correspondientes a este primer Sprint, que comprenderán de cierta manera el resto de vistas a elaborarse en el proceso. Así como también a la identificación de los diferentes componentes pertenecientes a cada vista. Lo que concierne a los diferentes componentes externos se anexa una descripción centralizada de los mismos, indicándose referencias para su consulta.

#### **1.3 Definiciones, Acrónimos y Abreviaciones**

Ver Glosario

#### **1.4 Referencias**

Ver Anexo

## **2. Objetivos y restricciones de la arquitectura**

El proyecto *SMS Chat* será desarrollado en base a los diferentes modelos principalmente dados por la Vista Lógica y de Casos de Uso.

### **2.1 Objetivos:**

- Diseñar una arquitectura que sea necesaria y suficiente, que cuente con un funcionamiento efectivo, es decir, que provea las funciones y propiedades especificadas.
- Elaborar un diseño conciso, claro, con alto grado de cohesión y bajo acoplamiento entre cada uno de los componentes, para así, permitir la manipulación y reemplazo de los mismos con facilidad que anticipe mudanzas posibles y permita el trabajo a futuro.
- Desarrollar un bosquejo inicial de la aplicación, que integre los distintos servicios existentes.

### **2.2 Requerimientos Especiales**

#### *2.2.1 Usabilidad*

Front End para Administración de Parámetros.

- La aplicación en donde se dan mantenimiento a las los parámetros, mensajes, tarifas, promociones, debe ser una página web.
- La aplicación web generada debe seguir el estándar gráfico definido por Technology Factory. El estándar gráfico se muestra en la sección de anexos.
- La aplicación debe soportar ser ejecutada en al menos los siguientes navegadores de Internet
  - Microsoft Internet Explorer 8 o superior
  - Mozilla Firefox 9 o superior
  - Google Chrome 3 o superior
  - Apple Safari 5 o superior

#### **2.1.2 Escalabilidad**

La aplicación Web tiene que ser completamente escalable sin que un aumento de los recursos dedicados a la misma suponga modificación alguna en su comportamiento o capacidades.

#### **2.1.3 Seguridad**

El acceso al aplicativo web debe ser controlado con nombres de usuario y contraseñas. Solo los usuarios con derechos de administrador podrán acceder las funciones administrativas mientras los usuarios normales no. Se deberá utilizar un esquema de autenticación por LDAP.

Para los servicios WEB basados en SOAP se deberá emplear el estándar WS-Security.

#### **2.1.4 Interoperabilidad**

La aplicación debe soportar la capacidad de interoperar con sistemas externos a nivel de datos y procesos.

#### **2.1.5 Importación y exportación de datos**

El sistema deberá almacenar todos los datos en una base de datos Oracle 11g donde pueda ser accesado por otros programas.


#### **2.1.6 Restricciones de Diseño**

Los ambientes de pruebas y de producción para el aplicativo SMS Chat poseen las siguientes características

- Sistema Operativo Linux
- Apache 2
- Tomcat 6.0.3
- Base de datos Oracle 11g
- Desarrollo se requiere en Java.
- Aplicación EAR se debe desarrollar en JDeveloper

### 3. Mecanismos de Arquitectura

A continuación se listan los mecanismos de arquitectura identificados para este proyecto:

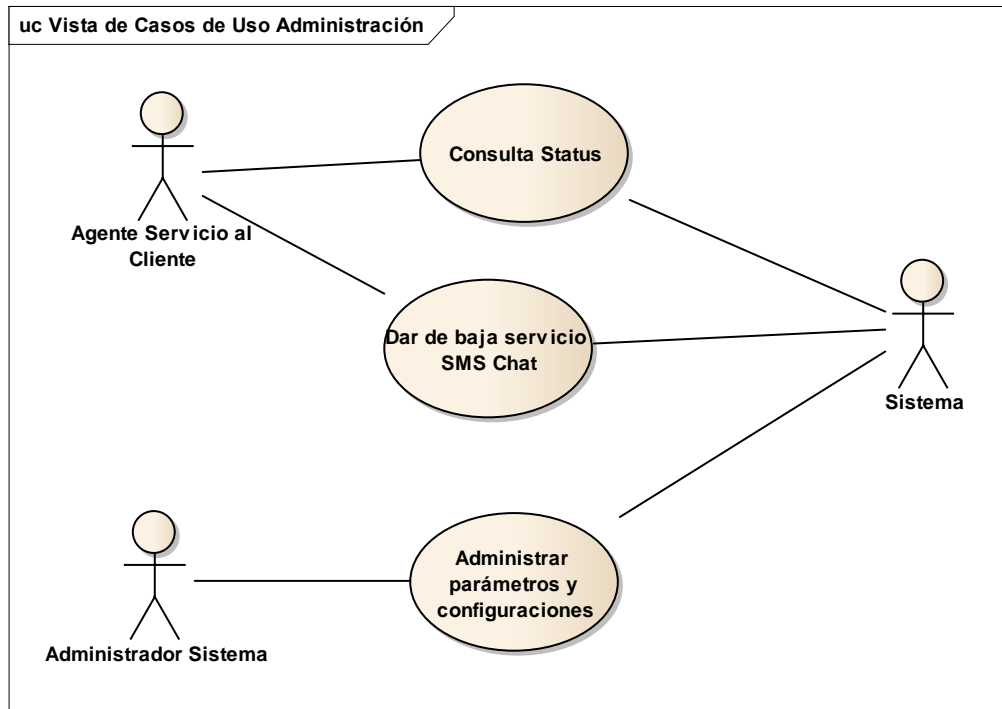
Nombre (mecanismo de análisis)	Descripción	Solución Propuesta (Mecanismo de diseño e implementación)
Integración con servicio SMSService	Se debe invocar al servicio SMSService   SMSService.wsdl	A nivel de capa de lógica de negocio se deben implementar adaptadores para consumir éste servicio.
Interacción con bases de datos a través de servicios web.	Este mecanismo define la manera en que se va a interactuar con las siguientes bases de datos: 1. WSDB (que se encuentra en 172.30.4.30/ORACLE 10g) 2. SMSCHATDBPIL-SRV (que se encuentra en 172.30.5.162/ORACLE 10g)	Se utilizará el adaptador que trae el Oracle Middleware para realizar conexiones a bases de datos.
Escalabilidad		La infraestructura con la que se cuenta puede crecer en base a análisis de transacciones y aumento de capacidad. Esto no afecta la funcionalidad del desarrollo.
Interoperabilidad		La interoperabilidad entre los sistemas se deberá lograr aplicando correctamente los estándares SOAP a nivel de contrato del servicio, siguiendo los estándares de nombramiento de servicios de Tigo.
Seguridad		La seguridad requerida para la interfaz web deberá ser implementada vía LDAP y con el estándar WS-Security para los Web Services basados en SOAP.

#### 4. Vista de Casos de Uso

Véase a detalle el documento de Casos de Uso de SMS Chat



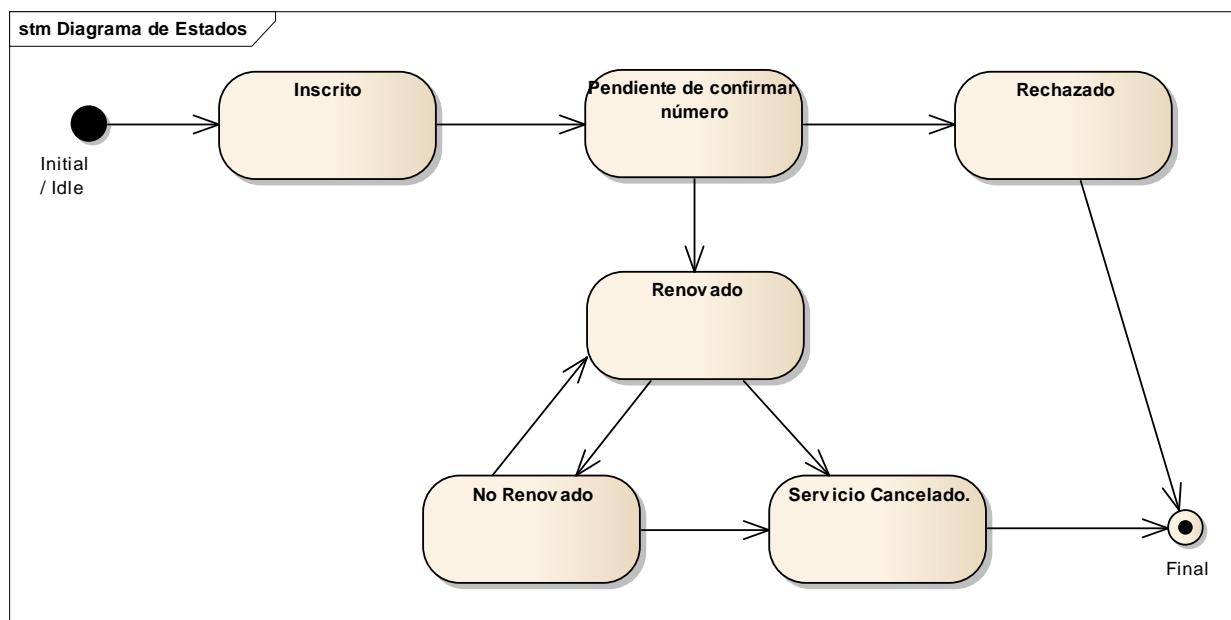




## 5. Vista Dinámica

### 5.1 Diagramas de Estado

A continuación se muestran los distintos estatus que pueden tener las suscripciones con otros números.



Estado	Descripción
Inscrito	Cuando el usuario se suscribe al servicio de SMS Chat
Pendiente de Confirmar Número	Cuando usuario A invita a otro número a formar parte de su chat y espera respuesta
Renovado	Servicio renovado cuando se ha cobrado exitosamente la cuota diaria del servicio.
Rechazado	Refleja la acción de que un número de B no aceptó formar parte del chat de A.
No Renovado	El servicio no está renovado debido a que no se pudo hacer el cobro recurrente del servicio.
Servicio Cancelado	El servicio se encuentra cancelado a solicitud de uno de los dos usuarios.

---

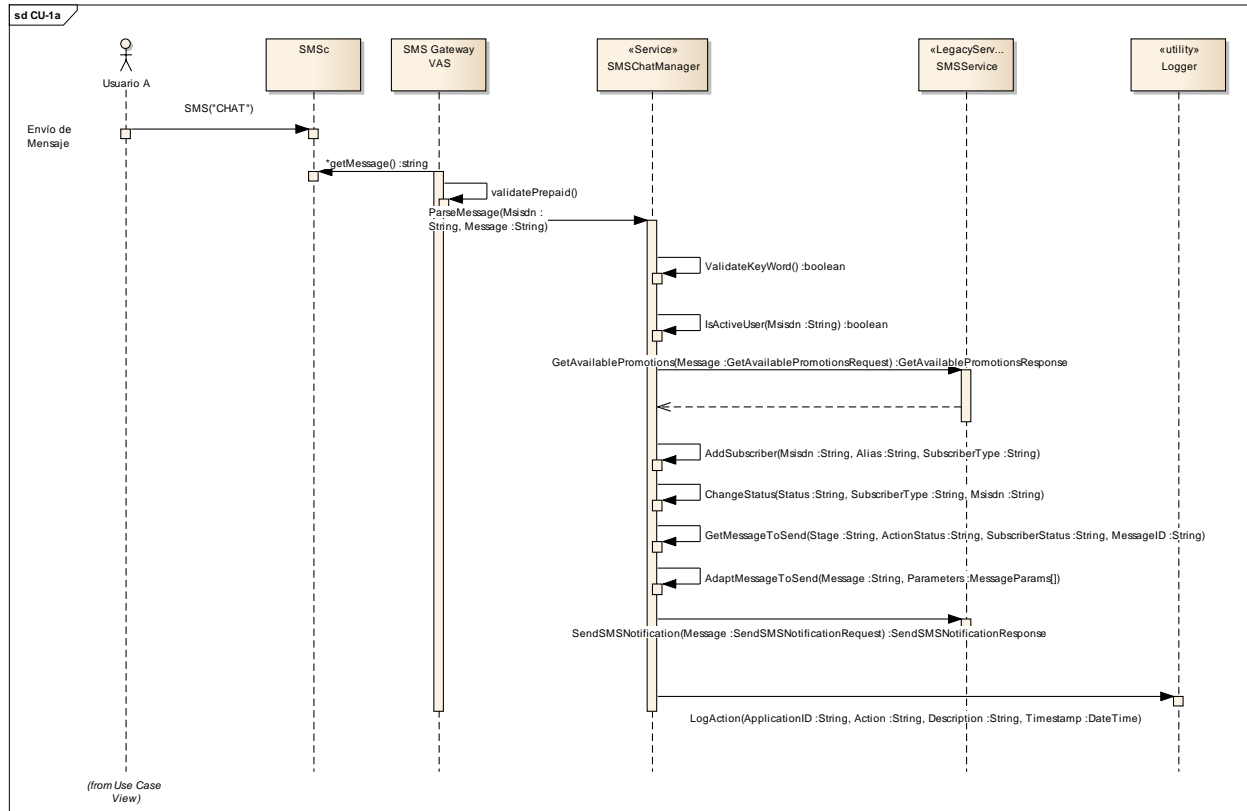
## **5.2 Diagramas de Secuencia**

Los diagramas de secuencia ilustran la interacción entre objetos y el orden secuencial en el que ocurren dichas interacciones, es decir como se comunican los objetos entre sí.

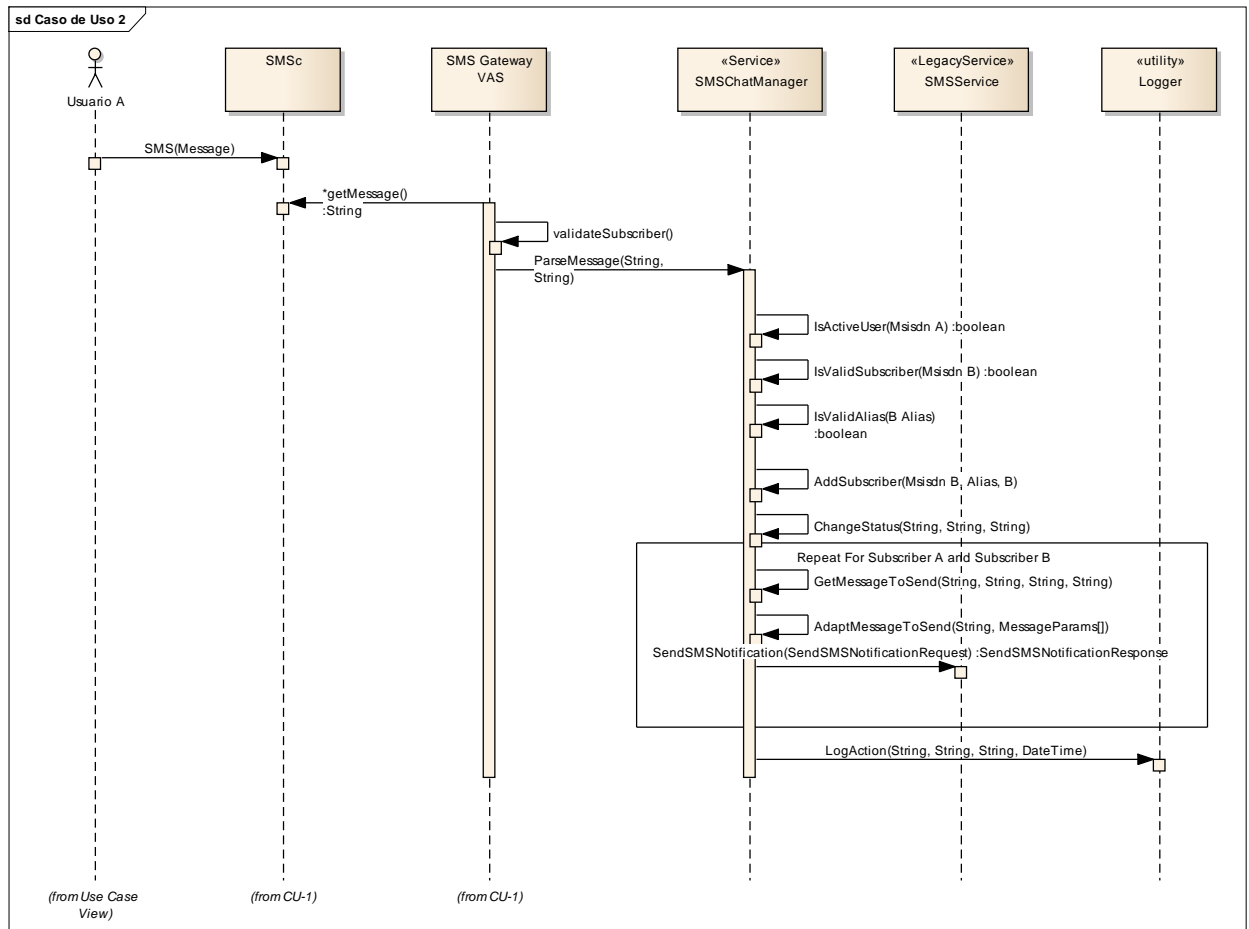
En los Casos de Uso se modelan las características del sistema y se desarrollan escenarios, los diagramas de secuencias proporcionan un camino a partir de los escenarios para describir las operaciones en una forma más detallada. A continuación se detallan los diagramas de secuencias considerados para los Casos de Uso de la sección anterior:

Nota: Los modelos se adjuntan en formato web.

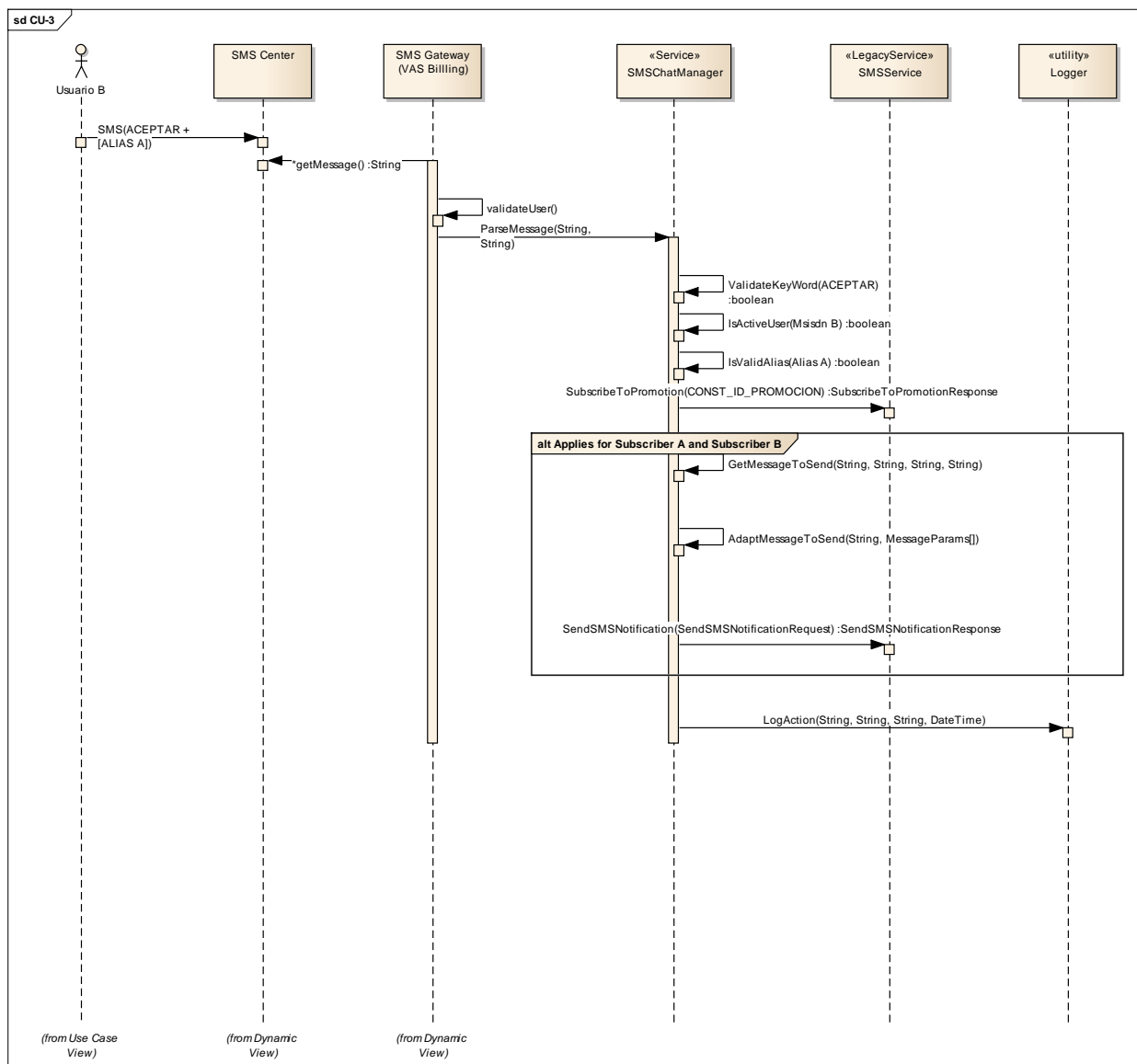
■ **DS01. Suscribir a Usuario a SMS Chat**



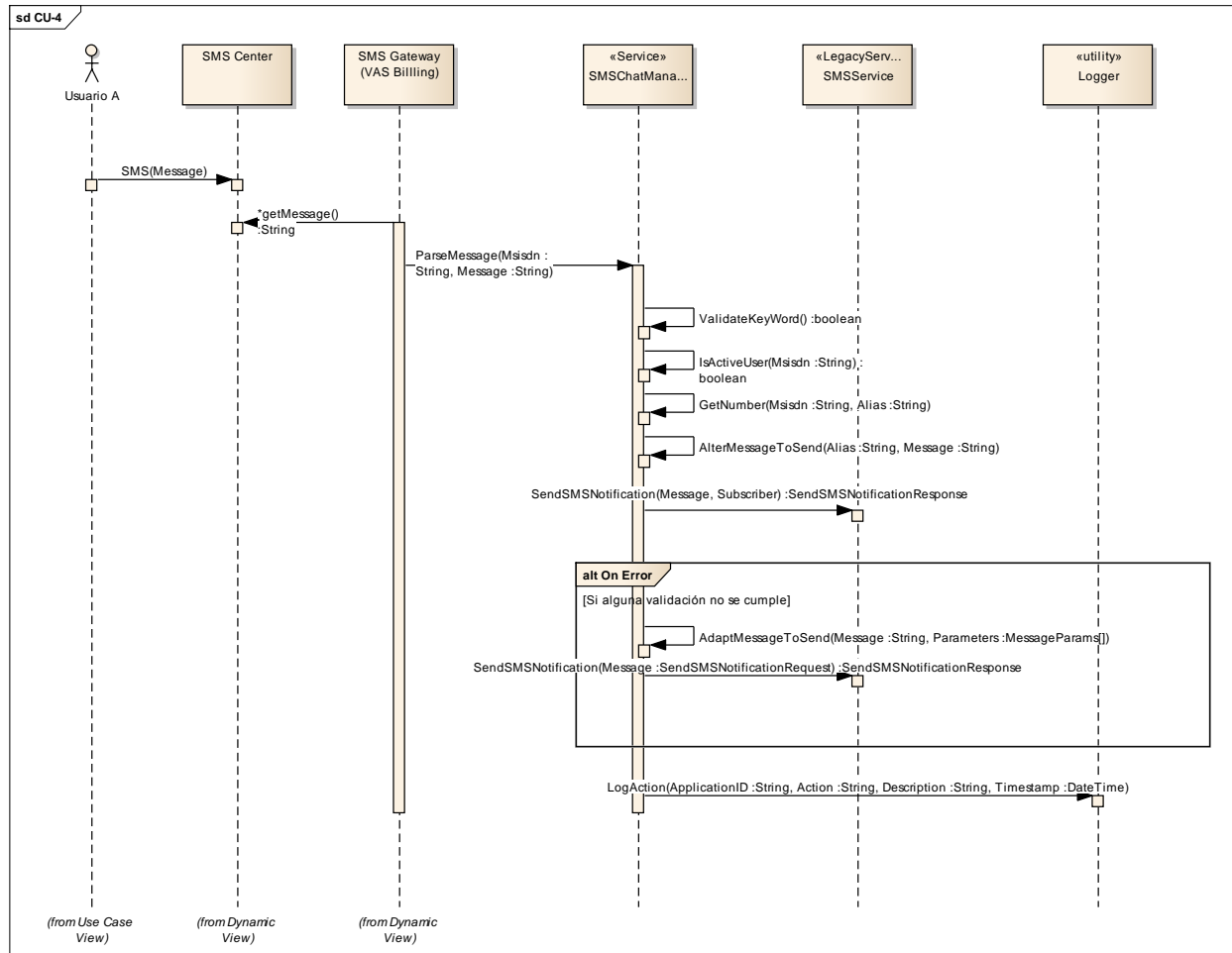
▪ **DS02. Usuario A registra a Usuario B**



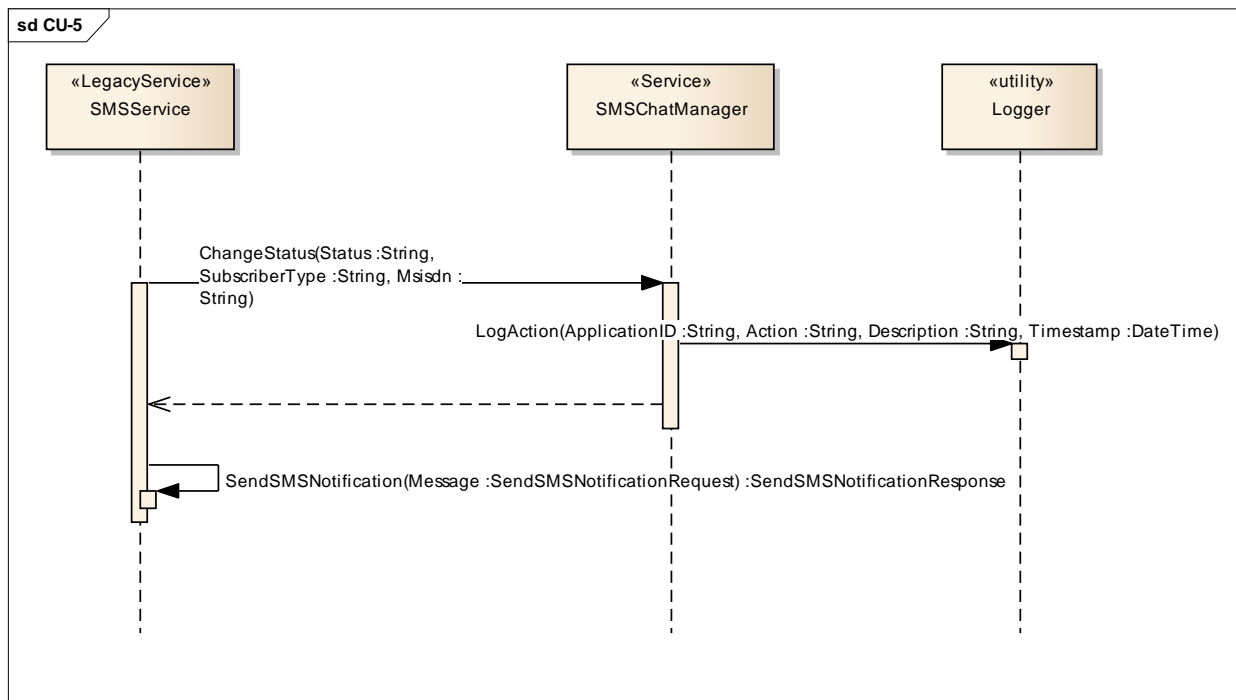
▪ **DS03. Usuario B confirma Chat con Usuario A.**



▪ **DS04a: A y B se Comunican sin costo.**



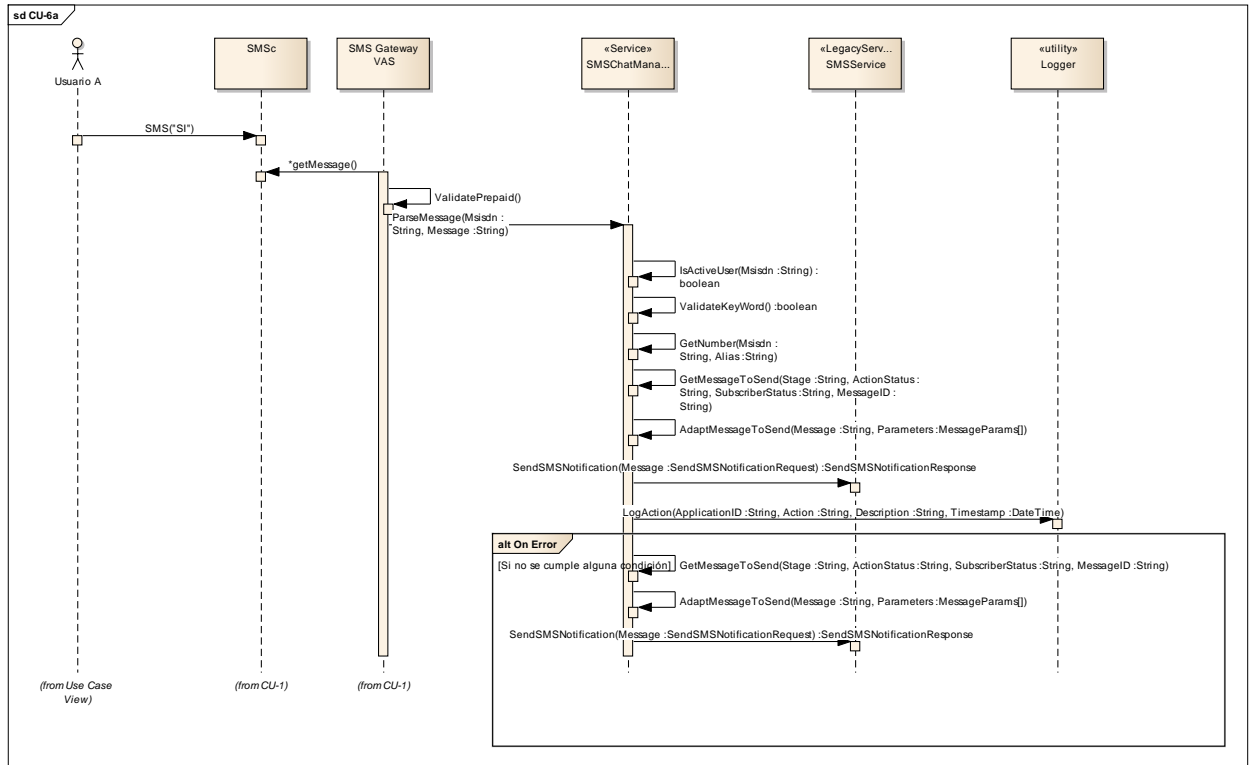
### DS05. Renovación diaria del Servicio SMS Chat



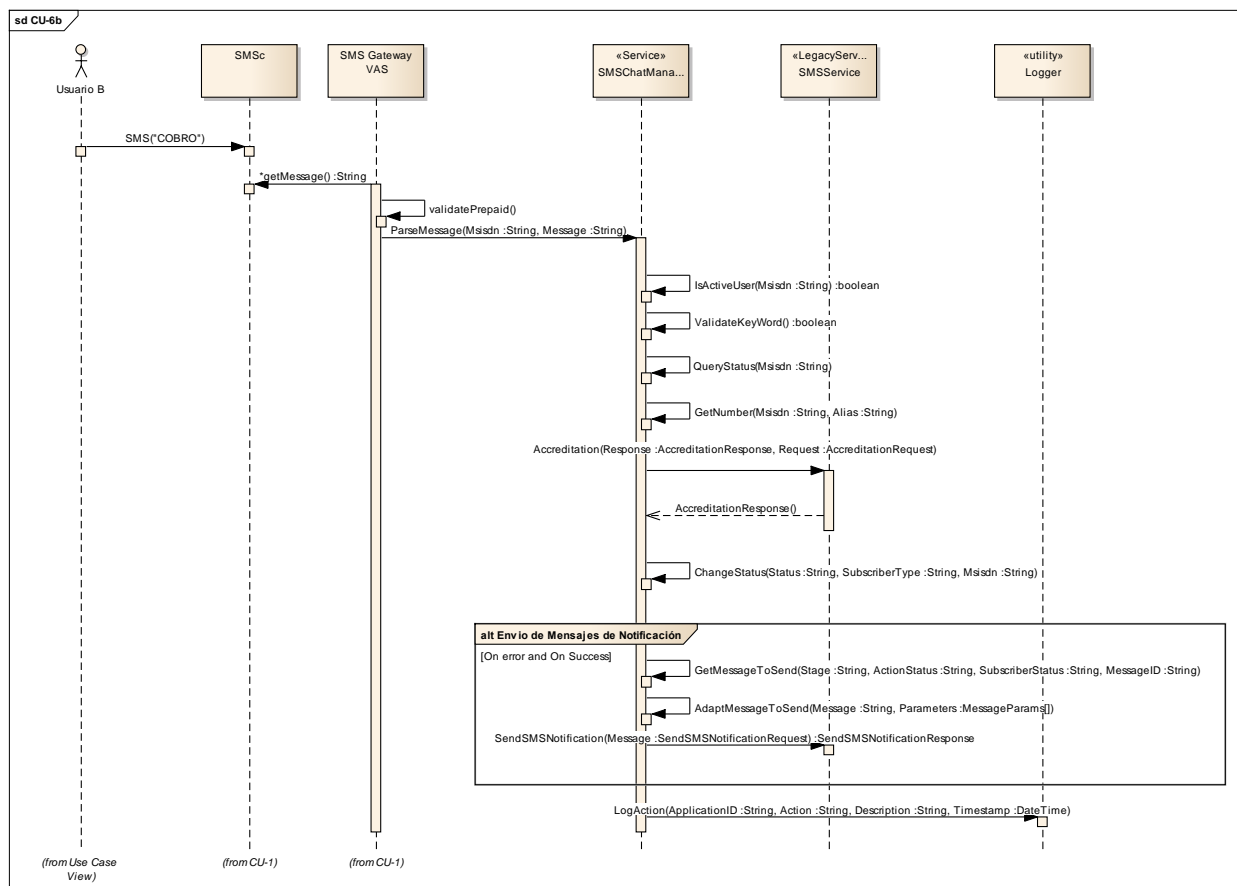
\*Esta operación se hará por medio del cobro de suscripción que ya realiza VAS Billing (se configurará como una suscripción de cobro recurrente)



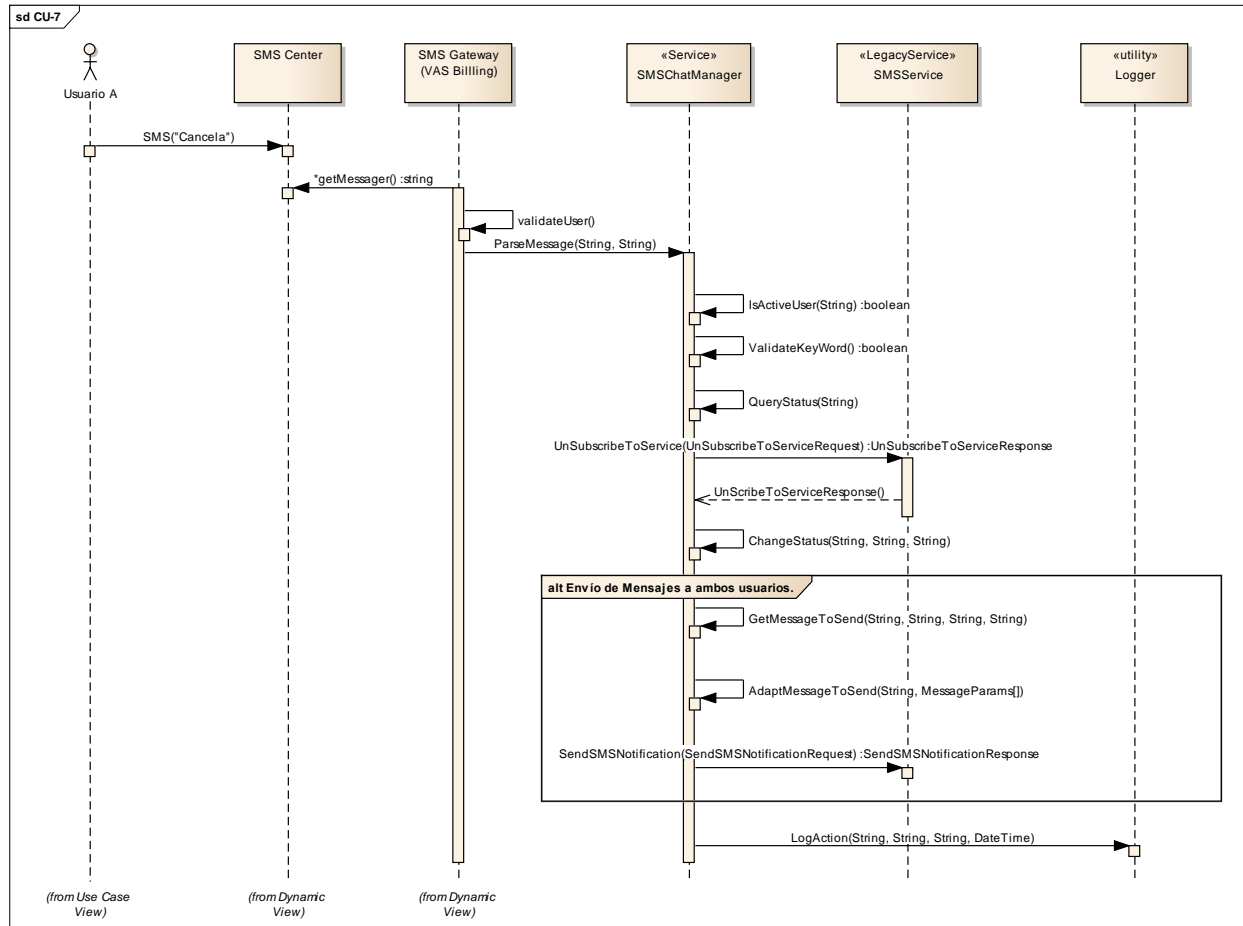
- **DS06a. Usuario A solicita a Usuario B que haga el pago del Servicio.**



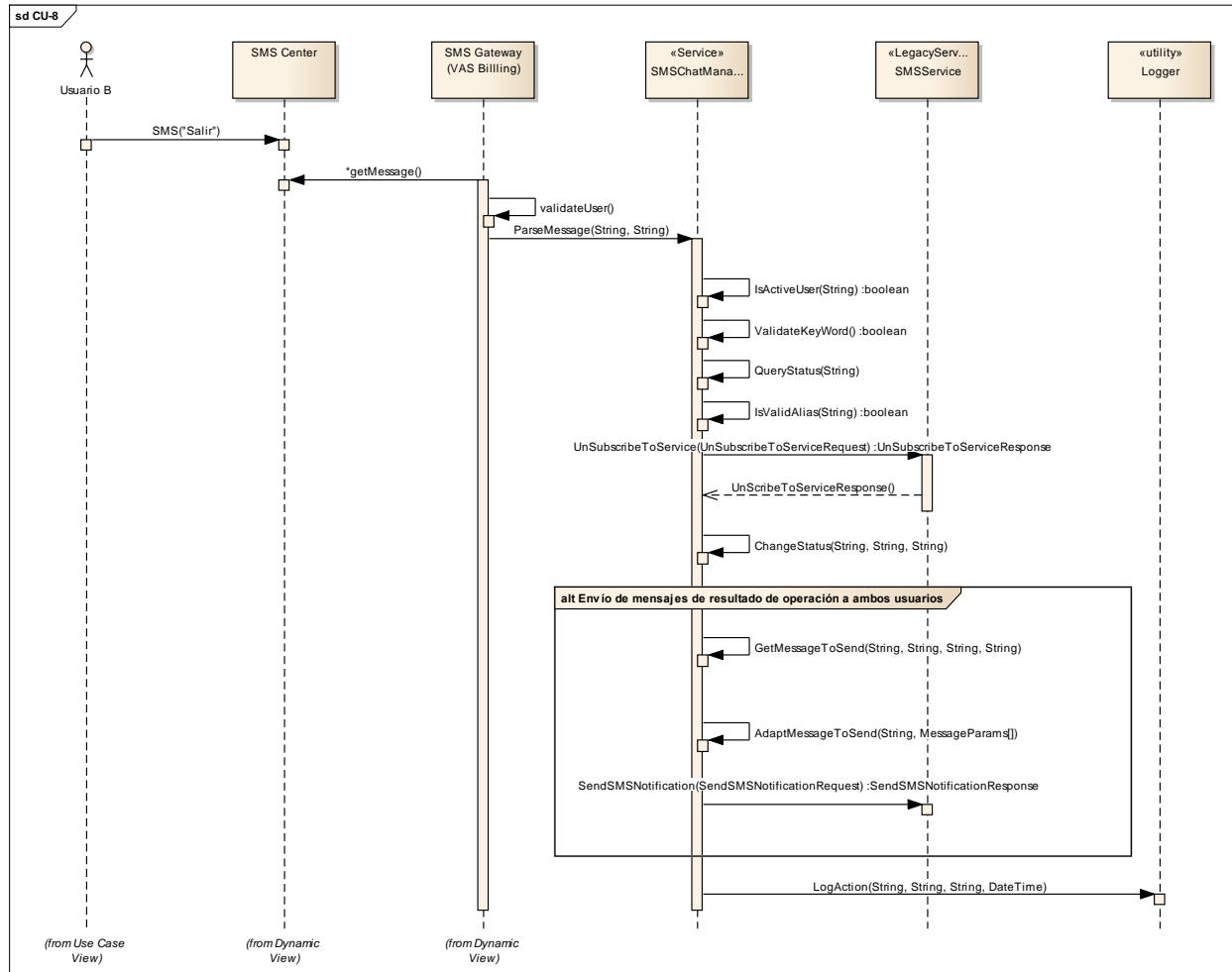
▪ **DS06b. Usuario B responde a solicitud de cobro enviada por Usuario A**



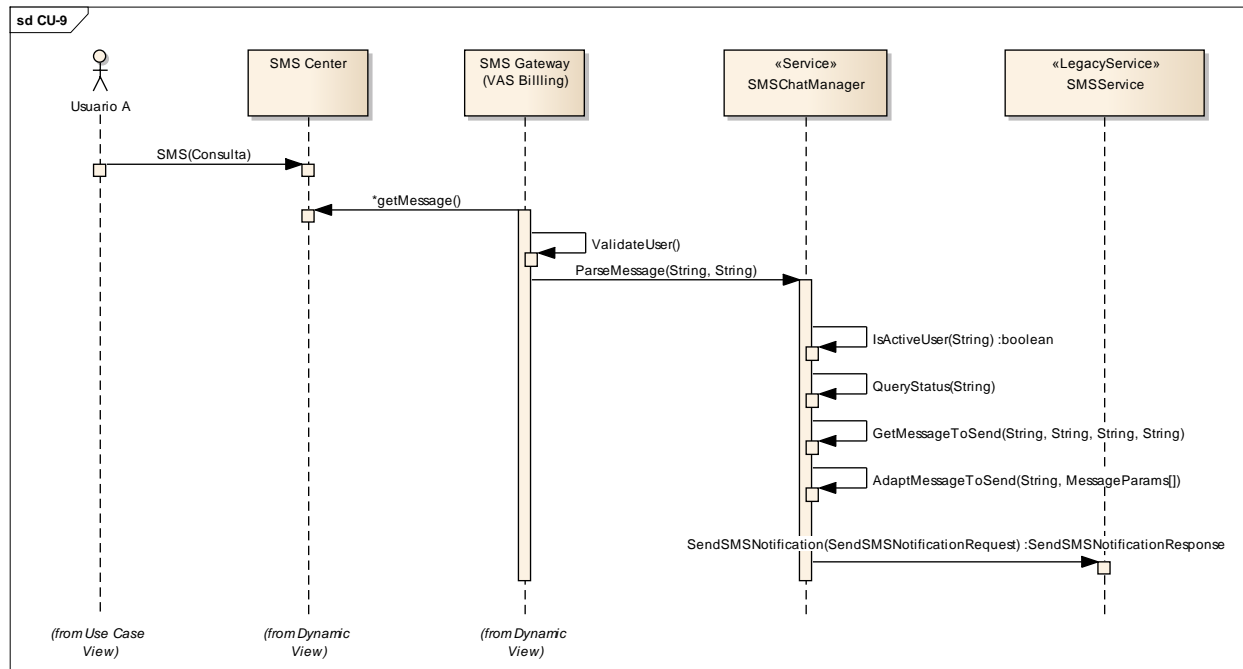
▪ **DS07. Usuario A solicita Cancelación de Chat**



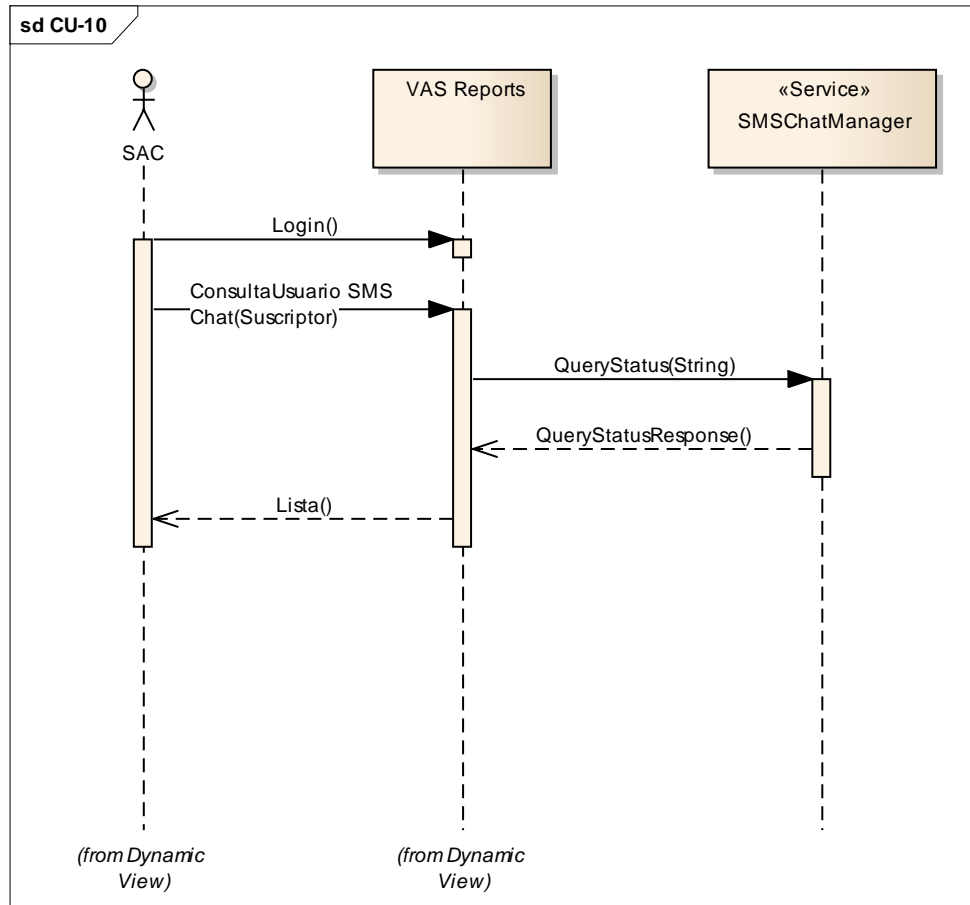
▪ **DS08. Usuario B solicita Cancelación de Chat**



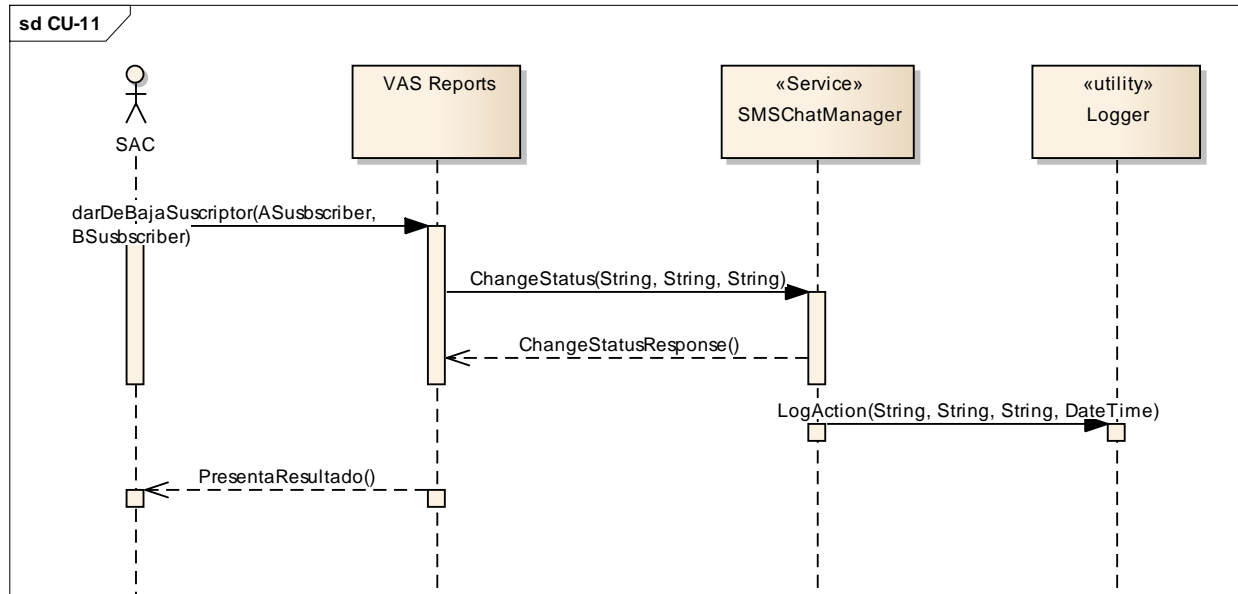
▪ **DS09. Usuario Consulta Status Suscripción**



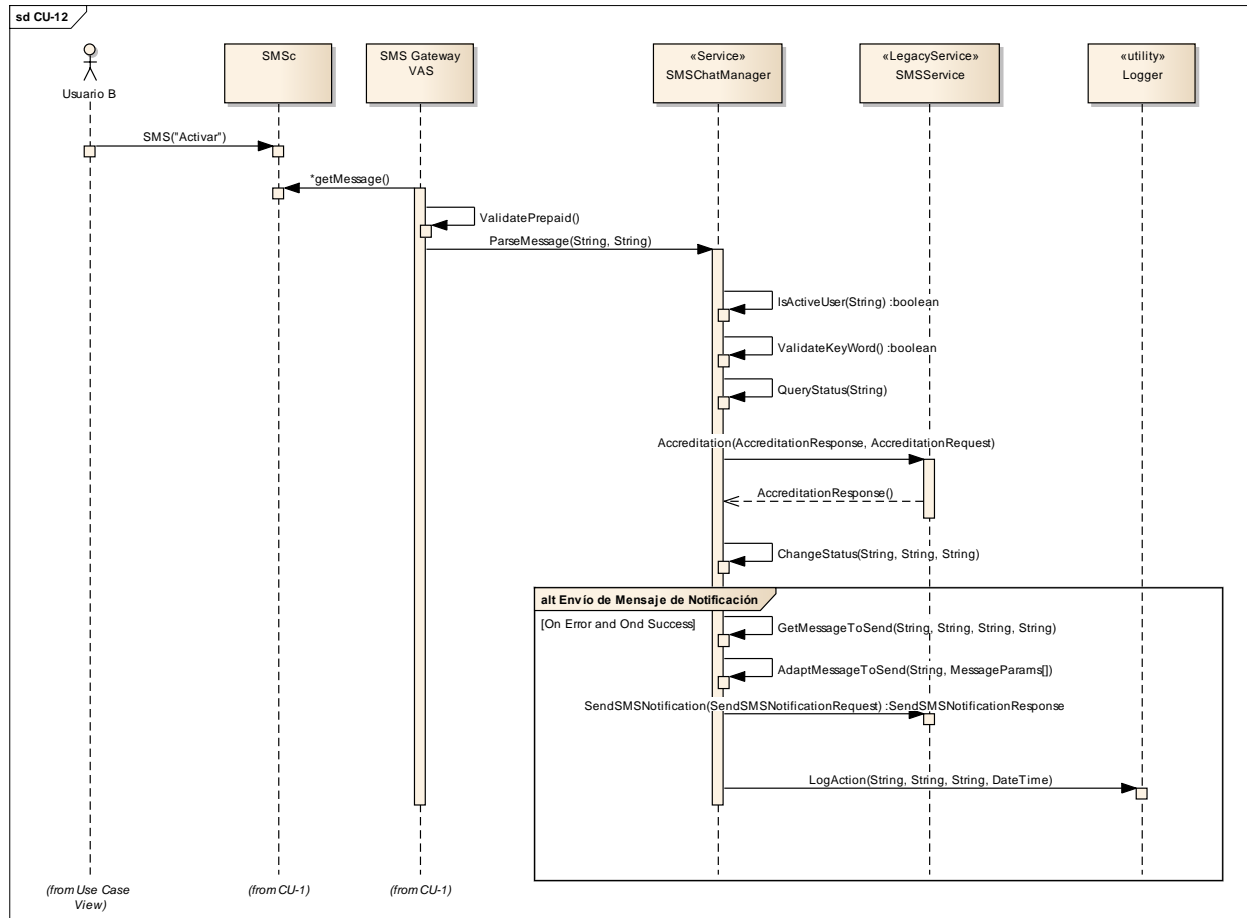
▪ **DS10. Agente SAC consulta Status.**



▪ **DS11. Agente SAC da de baja servicio SMS Chat**



▪ **DS12. Reactivación de SMS Chat**





## 6. Vista Lógica

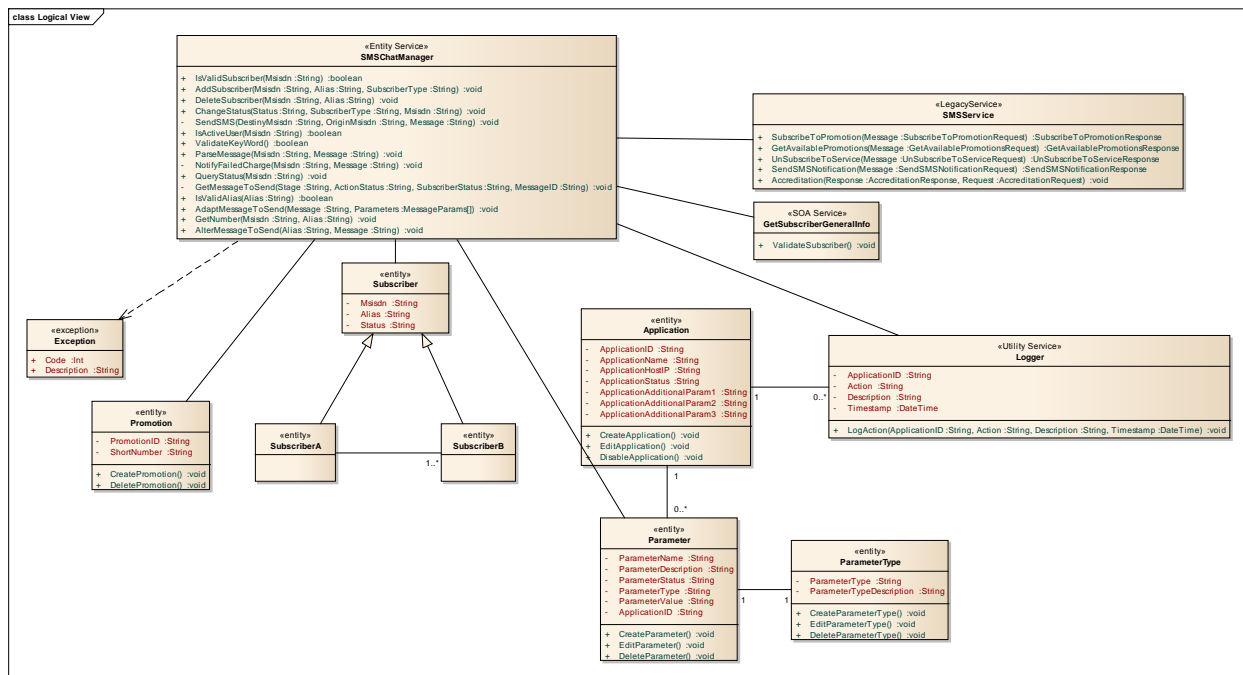
### 6.1 Introducción

En esta vista se presentan las unidades lógicas que componen la arquitectura del proyecto SMS Chat, así como también los refinamientos en cada etapa.

Los refinamientos descritos involucran:

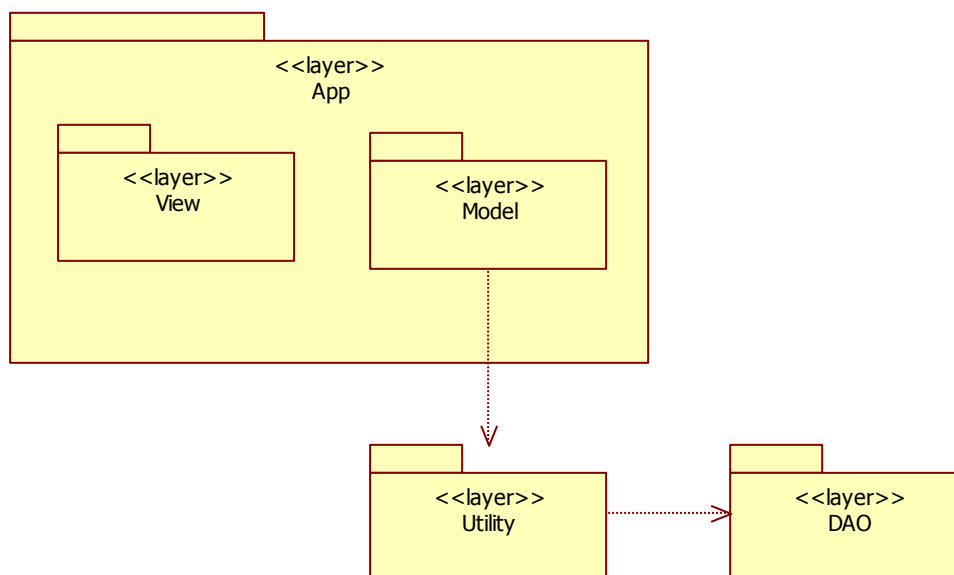
- Identificación de principales entidades con atributos
- Descomposición en subsistemas
- Determinación de los componentes de cada uno de los subsistemas.
- Realización de casos de uso, en base a la vista anterior, mediante los componentes arquitectónicos ya definidos.

### 6.2 Entidades



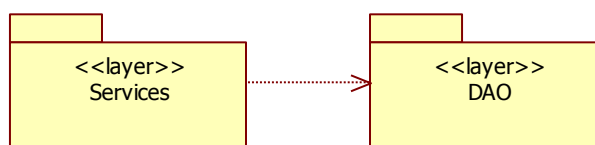
### 6.3 Paquetes de Diseño en la Arquitectura

#### Paquete para Aplicación de Mantenimiento de Parámetros



- **Capa App:** es la capa de presentación, que invocará los componente utilitarios definidos para la parametrización. Esta capa es un proyecto web generado por la herramienta JDeveloper y se compone de las siguientes dos capas:
  - **Capa View:** Esta capa contiene las páginas JSF del proyecto web.
  - **Capa Model:** Esta capa contiene los controles hacia la capa de negocio.
- **Capa Utility:** Es la capa de objetos del modelo de parametrización.
- **Capa DAO:** Capa de acceso a base de datos donde se persiste la información de los parámetros

#### Paquete para Motor de SMS Chat



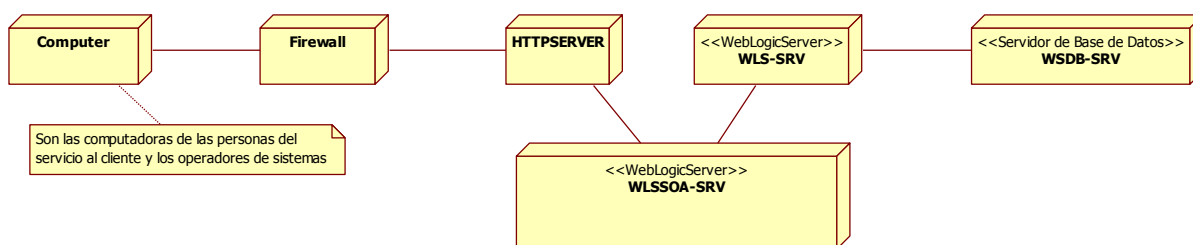
- **Capa Services:** Es la capa de Servicio Web de encapsula la lógica de negocio de SMS Chat.
- **Capa DAO:** Capa de acceso a base de datos donde se persiste la información del producto SMS Chat.

## 7. Vista de Despliegue

### 7.1 Introducción

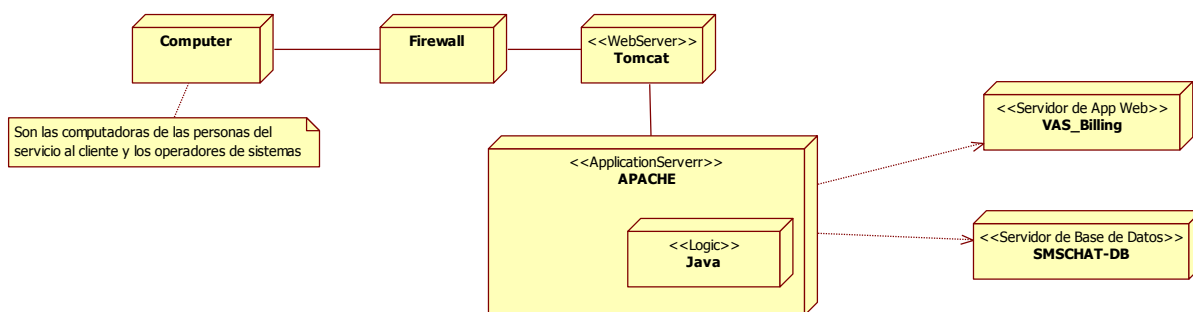
En esta vista debemos de mostrar la configuración de los nodos que participan en la ejecución y de los componentes que residen en los nodos.

### 7.2 Distribución para Aplicación de Administración de Parámetros



- **Computer:** es la estación de trabajo donde se va a desplegar la página web.
- **Firewall:** es el punto de filtración para la estaciones de trabajo que ingresan a la plataforma de TIGO.
- **WLSOA-SRV:** servidor que contiene la instalación del Oracle SOA Suite y el Oracle Service Bus.
- **WLS-SRV:** servidor que contiene la instalación de Web Logic Server.
- **WSDB-SRV:** Servidor de base de datos en donde se almacenará el esquema de parametrizaciones generales.

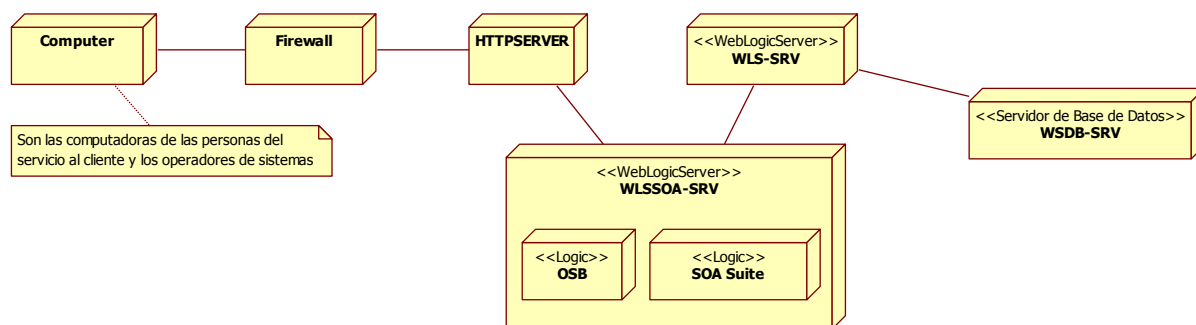
### 7.3 Distribución para Aplicación de SMS Chat



- **Computer:** es la estación de trabajo donde se va a consultar la información del servicio web.
- **Firewall:** es el punto de filtración para la estaciones de trabajo que ingresan a la plataforma de TIGO.
- **Apache/Tomcat:** Servidor de aplicaciones web.
- **Java:** Instancia de la máquina virtual de Java para la ejecución del web service.
- **VAS\_Billing:** Servidor que contiene el servicio web SMSService.

- **SMSCHAT\_DB:** Base de datos que contendrá la información de los abonados del procto.

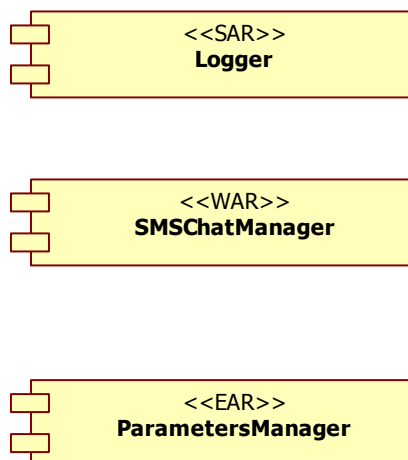
## 7.4 Distribución para servicio utilitario Logger



- **Computer:** es la estación de trabajo donde se va a consultar la información del servicio web.
- **Firewall:** es el punto de filtración para la estaciones de trabajo que ingresan a la plataforma de TIGO.
- **WSSOA-SRV:** servidor que contiene la instalación del Oracle SOA Suite y el Oracle Service Bus.
- **OSB:** servidor lógico que contiene toda la infraestructura del bus empresarial de Oracle
- **SOA Suite:** servidor lógico que contiene toda la infraestructura para manejar servicios SOA, incluyendo: BPEL, Reglas de Negocio, Mediador, Tareas Humanas, entre otros
- **WLS-SRV:** servidor que contiene la instalación de Web Logic Server.
- **WSDB-SRV:** Servidor de base de datos en donde se persistirá la información de Log

## 8. Vista de Componentes

El siguiente diagrama muestra los componentes de la estructura global del modelo de implementación



- 
- **Logger:** Servicio Logger que se debe instalar en Oracle Service Bus
  - **SMSChatManager:** Servicio Web que se debe instalar en el servidor SMSCHATPIL-SRV que contendrá la lógica de manejo del producto SMS Chat
  - **ParametersManager:** Aplicación desarrollada en JDeveloper que será la interfaz para el mantenimiento de los parámetros de las aplicaciones. Esta se instalará en la capa Web de WebLogicServer.