

PLIEGO DE PRESCRIPCIONES TÉCNICAS

Servicio de Mensajería SMS



INDICE

1. OBJETO	2
2. DISPOSICIONES Y NORMAS DE APLICACIÓN	2
3. TÉRMINOS Y DEFINICIONES.....	2
4. REQUISITOS TÉCNICOS	3
5. DESCRIPCIÓN DE APLICATIVOS “BRÓKER SMS” Y “METRO SMS”	4
6. ALCANCE.....	4
7. REQUISITOS DE SEGURIDAD, SALUD Y MEDIO AMBIENTE	4
8. DISPONIBILIDAD DEL SERVICIO	6
9. CORTES PROGRAMADOS	6
10. ATENCIÓN AL CLIENTE.....	6
11. GARANTÍA	6
12. ANEXO I: APLICACIÓN BROKERSMS.....	6
13. ANEXO II: APLICACIÓN METROSMS.....	30

Control del documento:

Versión	Fecha	Código
1.0	22/12/2022	PL-CTI-COM-22-00-0018

1. OBJETO

El presente Pliego de Prescripciones Técnicas tiene como objeto definir los alcances de los trabajos a realizar para la contratación del SERVICIO DE MENSAJERÍA SMS a través de una plataforma con un bróker SMS, garantizando la continuidad, disponibilidad y calidad actuales del servicio de mensajería.

2. DISPOSICIONES Y NORMAS DE APLICACIÓN

Los trabajos objeto del contrato se llevarán a efecto mediante la plena observancia y cumplimiento de todas las disposiciones legales vigentes, actuales y futuras, que afecten a dichos trabajos, ya se trate de leyes, reglamentos, ordenanzas, instrucciones o normas de cualquier otro rango que resulten obligatorias, ya sean de ámbito comunitario, nacional, autonómico o local.

Entre tales disposiciones, y a título de relación no exhaustiva, se destaca la necesidad de dar cumplimiento a todas las normas jurídicas vigentes relativas a las siguientes actividades:

- REBT. Real Decreto 842/2002.
- Seguridad y Salud en los Lugares de Trabajo. Real Decreto 486/1997.
- Señalización de Seguridad y Salud. Real Decreto 485/1997.
- UNE 21302-91/2M. Vocabulario electrotécnico o equivalente.
- UNE 200001-3-2. Gestión de la confiabilidad o equivalente.
- Norma ISO 9001. Sistemas de Gestión de la Calidad o equivalente.
- ISO 14001. Gestión Medioambiental o equivalente.
- UNE 13460. Mantenimiento o equivalente.
- EN 60950. Safety of Information Technology equipment o equivalente.
- ISO 20000. IT Service Management o equivalente.
- ISO 27000. Sistemas de Gestión de Seguridad de la Información o equivalente.

Especialmente, el contratista estará obligado a cumplir los procedimientos que Metro de Madrid, S.A. tiene establecidos, o pueda establecer en el futuro, para los trabajos que se realicen en sus instalaciones, de los que será cumplidamente informado antes del inicio de los mismos, con objeto de que pueda trasladar dicha información a sus trabajadores, quienes deberán cumplirla debidamente.

Todas las prescripciones y especificaciones técnicas que se formulen en el presente pliego por referencia a cualesquiera de las tipologías normativas recogidas por el artículo 42.3 b) de la Directiva 2014/24/UE, de 26 de febrero, sobre Contratación Pública, habrán de entenderse hechas también a sus equivalentes, correspondiendo al licitador acreditar dicha equivalencia en la forma establecida en el artículo 42.5 de la mencionada Directiva

3. TÉRMINOS Y DEFINICIONES

A efectos del presente documento se entenderá por:

“Ofertante”: empresa que presenta una Oferta Técnica y Económica para la prestación del servicio de mantenimiento objeto de este Pliego.

“Contratista”: empresa adjudicataria del servicio de mantenimiento objeto de este Pliego.

“Metro”: Metro de Madrid, S.A.

4. REQUISITOS TÉCNICOS

Los ofertantes tomarán en cuenta que las plataformas BRÓKER SMS y METRO SMS son las aplicaciones propietarias de Metro de Madrid que se encargan de hacer de pasarelas con el resto de sistemas finales o de usuario, igualmente propiedad de Metro, sirviendo de interfaces entre el proveedor de SMS y las aplicaciones corporativas de la compañía. Los fundamentos técnicos de estas aplicaciones quedan descritos en los Anexos I y II del presente documento.

El Contratista deberá, obligatoriamente, integrar su servicio de mensajería SMS en dichas plataformas, por lo que, si fuese necesario, asumirá los costes de todas las tareas relacionadas con la modificación del hardware, software, sistema operativo, código de la aplicación, etc., que permitan la compatibilidad de las plataformas del Contratistas con las plataforma BRÓKER SMS y METRO SMS de Metro de Madrid.

Durante la prestación del servicio, el Contratista aceptará y cumplirá los requisitos que se detallan a continuación:

- Asignación de numeración fija para el envío y recepción de los SMS.
- Envío de SMS certificado con validez legal en España.
- Posibilidad de envío de mensajes vía correo electrónico certificado.
- Acceso vía Web a la plataforma del operador del bróker para la gestión y control total de éste, incluyendo la consulta del salto de mensajes disponibles y el resumen de los consumidos.
- Posibilidad de recepción de mensajes o MO (Mensajes Originado en móvil con destino Metro) procedentes de cualquier operador. (1)
- Envío de SMS de más de 160 caracteres.
- Envío de SMS diferidos (con entrega en día/hora establecidos).
- Especificación del remitente mediante un texto ASCII.
- Notificación de entrega.
- Recepción de los mensajes originados con hora española (GMT +1).
- Envío/recepción a todos los operadores, incluidos los OMV y a todos los rangos de numeración móvil actuales, incluidos los que responden al patrón “+347XXXXXXX”.
- Posibilidad de obtener distintos usuarios o números públicos como remitente o destinatario de los SMS.
- Posibilidad de intercambiar SMS normales por certificados o viceversa (sin llegar a sobrepasar el montante económico global).
- Disponibilidad de la documentación de las “APIs” de programación del bróker para que un tercero pueda desarrollar aplicaciones.
- Notificaciones de estado de los mensajes (1).

- Consulta del estado de los mensajes enviados o programados.
- Porcentaje típico de mensajes entregados: $\geq 95 \%$.
- Tiempo de entrega inferior a 5 segundos para mensajes con destino España.
- Caudal ofertado mínimo de 150 SMS/s.
- El bróker deberá poder desplegar servicios REST con HTTPS. Como se ha comentado, en el apartado 5 se desarrolla el detalle de los aplicativos empleados por Metro de Madrid para la comunicación con el bróker.

(1): es obligatorio que estos servicios sean gestionados por el Contratista sin perjuicio de que puedan ser accesibles y consultados desde Metro de manera síncrona.

5. DESCRIPCIÓN DE APLICATIVOS “BRÓKER SMS” Y “METRO SMS”

Los Anexos I y II del Pliego describen detalladamente la arquitectura y funcionamiento de los aplicativos corporativos BRÓKER SMS y METRO SMS. En caso de ser necesario, los Licitantes deberán indicar explícitamente en su oferta la asunción de los costes de integración de su servicio en las plataformas BRÓKER SMS y METRO SMS, siendo el tiempo máximo de adaptación de 1 mes.

6. ALCANCE

El alcance de los trabajos es la contratación de un servicio de mensajería que contemple todos los requisitos técnicos indicados en el punto 4.

El volumen estimado de consumo total se presenta a continuación, calculado en función de los envíos realizados en años anteriores, si bien la petición de paquetes de SMS se hará con carácter mensual, ajustando Metro de Madrid la cantidad y tipología que solicitará en función de su necesidad real.

- 570.000 SMS de hasta 160 caracteres.
- 40.000 SMS MO entrantes a Metro de Madrid.
- 6.000 SMS certificados de hasta 160 caracteres.

Se destaca que, además de los SMS convencionales, la plataforma debe ser capaz de ofrecer los siguientes servicios:

- SMS certificado, que son básicos para determinadas aplicaciones que necesita disponer de legalidad y acuse de recibo del SMS enviado.
- MO o SMS de entrada a Metro de Madrid, servicio esencial para aplicaciones que, ante determinadas incidencias críticas, deben utilizar este canal como vía mediante la que se le comunique a Metro la resolución de las mismas.

7. REQUISITOS DE SEGURIDAD, SALUD Y MEDIO AMBIENTE

7.1. Requisitos de seguridad y salud

El Contratista, como responsable de las condiciones de trabajo de su personal, estará directamente obligado a cumplir cuantas disposiciones, presentes o futuras, estuvieren vigentes en materia laboral, de seguridad social, de prevención de riesgos laborales y de medio ambiente, debiendo adoptar las medidas necesarias para asegurar la indemnidad, integridad y salubridad de la personas, así como prevenir cualquier tipo de accidentes que pudieran producirse con

ocasión del cumplimiento del servicio, sean cuales fueren las causas de los mismos. Esto se entenderá, referido, en su caso también, a todo el personal subcontratado a través de otras empresas, así como a los trabajadores autónomos y procedentes de ETT que se contraten para determinados servicios asociados a este Pliego.

El Contratista dispondrá de los técnicos titulados cualificados en prevención que al respecto fueran precisos, arbitrando a pie de obra todas las medidas obligatorias al respecto.

El Contratista queda obligado a observar y hacer cumplir a todo su personal las normas de seguridad y salud en el trabajo que establezca la legislación, normativas vigentes, así como los procedimientos que en materia de prevención laboral establezca Metro para sus propios trabajadores, conforme a la correspondiente documentación, preexistente o sobrevenida, que, a tal efecto, reciba de Metro. En caso de observar alguna discrepancia o incompatibilidad entre las normas y procedimientos de prevención de riesgos de aplicación y elaboración propia, respecto a los indicados por Metro como referencia en actividades de idéntica naturaleza, habrá de ser puesto de manifiesto para su análisis, discusión y resolución necesaria.

Especialmente:

- A) En materia de prevención laboral establecerá las medidas pertinentes relacionadas con la seguridad de las personas (usuarios y trabajadores).
- B) Muy especialmente, todos los trabajos que se realicen en locales, armarios de maniobra y otros cofres con componentes con riesgo eléctrico, se efectuarán dejando sin tensión los elementos próximos que pudieran ser objeto de riesgo y cuyo contacto fortuito pudiera dar lugar a accidentes.
- C) Se utilizarán equipos de protección personal (EPI) adecuados y herramientas especiales.
- D) Se llevarán a cabo las diferentes reuniones que con carácter obligatorio y como coordinación de actividades empresariales marca la Ley de Prevención de Riesgos Laborales.

7.2. Condiciones generales exigidas para el cumplimiento en materia de Medio Ambiente

Con el fin de minimizar el impacto medioambiental, no sólo se tendrá en cuenta la explotación y mantenimiento de los equipos, sino también su diseño, fabricación, selección y manipulaciones de materiales. Se considerará la afección al medio ambiente desde el origen del proyecto y toda solución técnica o estética será precedida de un riguroso análisis para la integración de los siguientes aspectos:

- Siempre que sea viable, se presentará la alternativa de diseño que genere menos emisiones, ruidos, vibraciones y/o radiaciones electromagnéticas; así como el menor consumo de agua y energético posible.
- Se proyectarán las instalaciones y metodologías necesarias para la correcta gestión de los residuos que se vayan a generar.
- Se proyectarán las medidas oportunas para evitar cualquier vertido de sustancias peligrosas.
- Se tendrá en cuenta que el horario de trabajo minimice las molestias que se pudieran ocasionar por ruido emitido al exterior.
- Se tendrá en cuenta el impacto visual negativo que pudiera tener la instalación/obra, tomando las medidas necesarias para disminuirlo.

7.3. Condiciones exigidas en materia de gestión de residuos

La propiedad de los residuos generados en la obra será de Metro de Madrid. No obstante, será responsabilidad de la empresa contratada la disposición de los mismos, de acuerdo a la legislación vigente, en las instalaciones y condiciones que la Dirección de Servicio prescriba.

8. DISPONIBILIDAD DEL SERVICIO

Se ofertará la prestación del servicio de mensajería SMS en la modalidad de 24 horas los 365 días del año.

En cuanto a la disponibilidad del servicio, se establece un porcentaje mínimo del **99,5 %** del tiempo de la prestación del servicio para cada año.

9. CORTES PROGRAMADOS

Los cortes programados deberán ser notificados a Metro con una antelación de, al menos, 3 días laborables. Las notificaciones de trabajos programados deben contener, como mínimo, los siguientes datos:

- Comienzo y duración de los trabajos.
- Descripción de las tareas a realizar.
- Persona de contacto.

El tiempo total abarcado por los cortes programados respetará el porcentaje de disponibilidad indicado en el apartado anterior.

10. ATENCIÓN AL CLIENTE

Para la notificación de incidencias, el Contratista debe poner a disposición de Metro un número de teléfono 24 horas y un correo electrónico.

Cualquier cambio en los canales de comunicación se deberá comunicar por escrito y con la antelación suficiente para la configuración de los sistemas de gestión de mantenimiento de Metro de Madrid.

11. GARANTÍA

Por tratarse de un servicio de telecomunicaciones que se presta en tiempo real, este servicio no tiene asociado ninguna garantía.

12. ANEXO I: APLICACIÓN BROKERSMS

12.1. Glosario de términos

- SMS: Short Message Service
- MODEM: (GPRS o GSM) Dispositivo capaz de enviar/recibir SMS
- BRÓKER: Intermediario entre operadora y clientes para el envío de mensajes
- SOAP: Simple Object Access Protocol.
- HTTP: Hypertext Transfer Protocol
- HTTPS: Hypertext Transfer Protocol – Secure
- JDK: Java Development Kit
- WSDL: Web Services Description Language
- UML: Unified Modeling Language
- AE: Aplicación Externa

12.2. Arquitectura técnica

La aplicación módulo de mensajería se implementa sobre la base de un BEA Weblogic 10 y como base de datos un ORACLE 10g.

La aplicación se divide en las siguientes partes: la del envío de SMS a los usuarios (envío MT que incluye los distintos métodos de recepción de notificaciones solicitadas de forma síncrona), envío de mensajes MO y recepción de notificaciones de forma asíncrona, por último, estaría la parte de administración de la herramienta.

Para la parte del envío de SMS, el punto de entrada de la aplicación es un WebService con los siguientes métodos implementados:

- envioSMSSimple: permite el envío de SMS simple, el envío será realizado por el bróker del Contratista.
- envioSMSCertificado: permite el envío de SMS certificado.
- envioSMSCertificadoDiferido: permite el envío de SMS certificado diferido.
- envioSMSDiferido: permite el envío de SMS diferido, el envío será realizado el bróker.
- obtenerEstado: comprobar el estado de un mensaje previamente enviado.

Para la parte de administración, la aplicación se construye utilizando clases Java para la implementación de la lógica del negocio y páginas JSP para la implementación del 'control' y las 'vistas'. Dependiendo del perfil con que el usuario se logue, se tiene acceso a ciertas funcionalidades. En el caso de usuario administrador, las funcionalidades son las siguientes:

- Administración de aplicaciones externas: permitirá crear, eliminar, modificar y listar las aplicaciones que tendrán acceso al sistema de envío de mensajes MT y recepción de notificaciones y mensajes MO.
- Consola de Estadísticas: presenta por consola un conjunto de mensajes seleccionados mediante un filtro.
- Consola de Actividad: presenta por consola las distintas incidencias y su nivel seleccionados mediante un filtro.

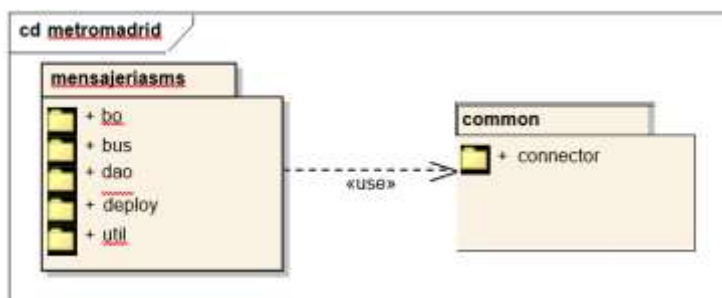
En el caso del perfil consulta las funcionalidades a las que tiene acceso son las siguientes: Consola de Estadísticas y Consola de Actividad.

Las diferentes clases utilizadas para la realización de los requisitos se desarrollarán de acuerdo a un 'modelo de capas' intentando desacoplar lo más posible la dependencia entre componentes. Con esto se conseguirá una aplicación flexible, escalable y fácilmente modificable.

Ver "ANEXO II. Arquitectura de la solución", para una información más detallada.

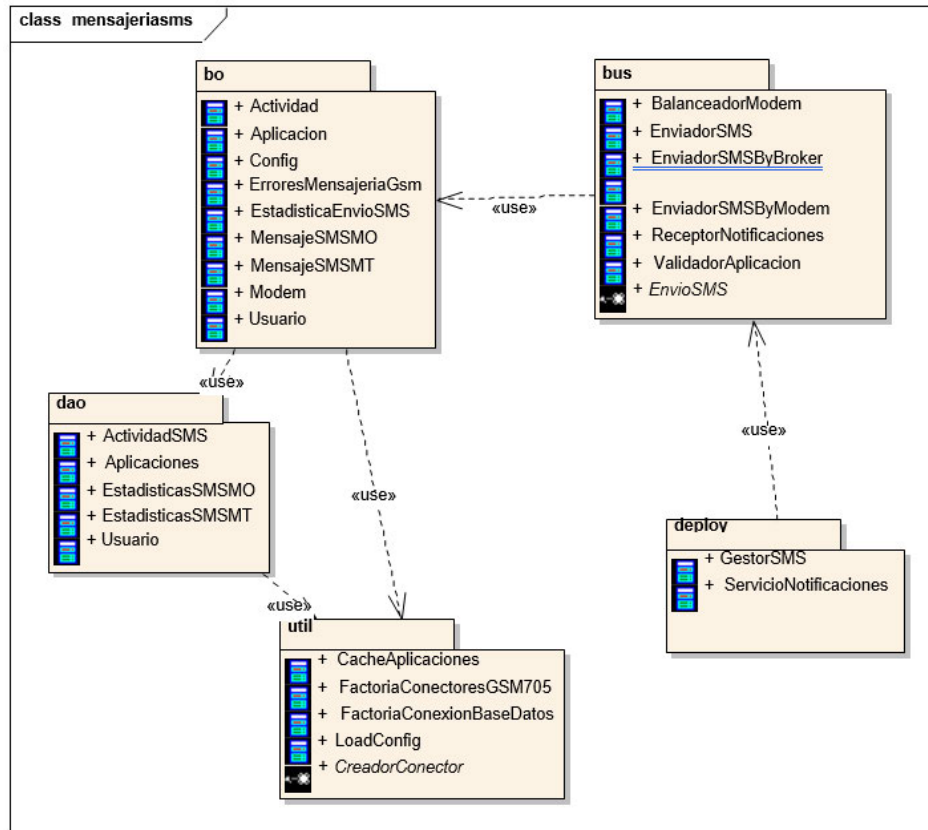
12.3. Diseño de clases

12.3.1. Diagrama de paquetes



El paquete mensajería SMS utiliza el paquete common, en éste, se implementan todas las funcionalidades específicas del módem (conectarse, enviar, leer respuesta, desconectarse...), así como de los conectores con el bróker de Contratista.

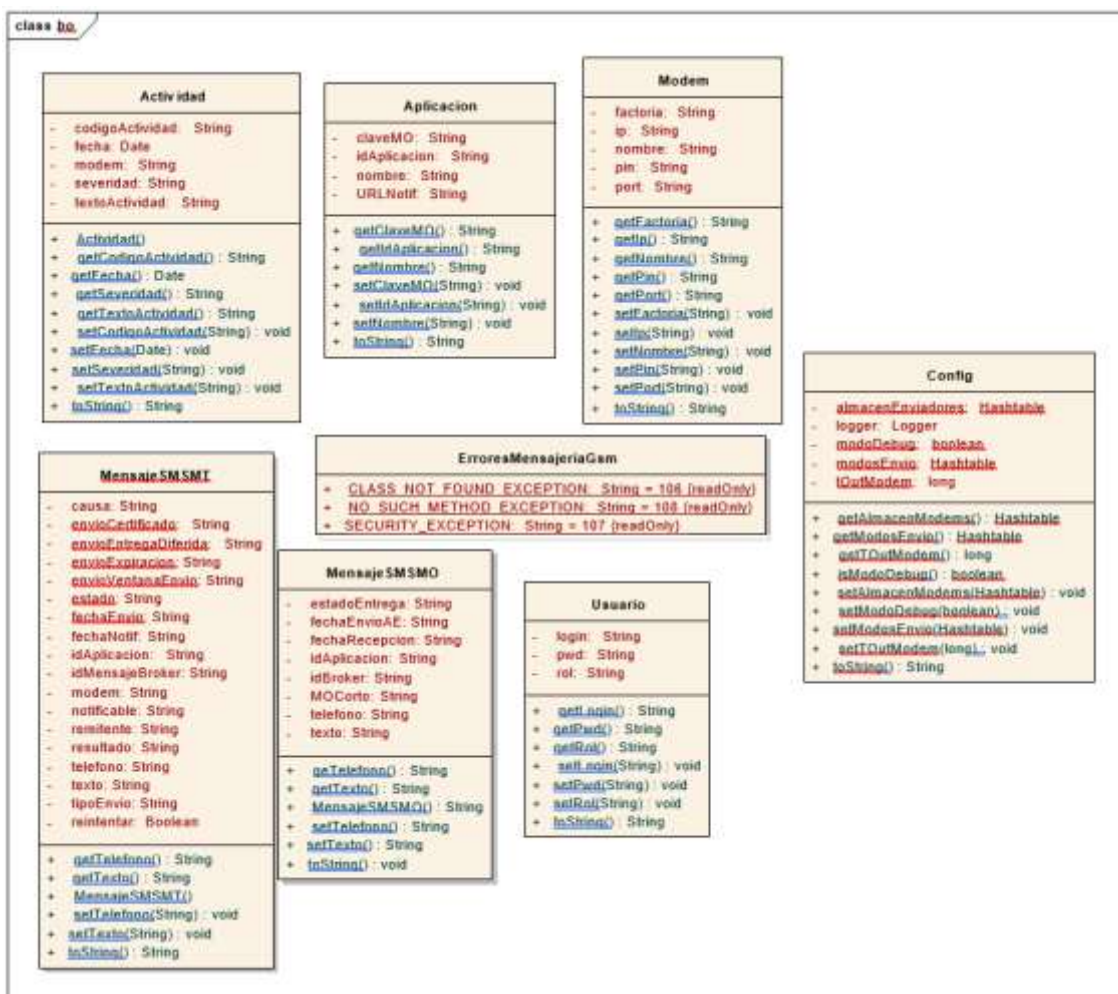
12.3.1.1. Paquete es.metromadrid.mensajeriasms



Este paquete contiene las clases de la capa de presentación y lógica de la aplicación:

- Bo: contiene las clases de negocio de la aplicación.
- Dao: contiene las clases de acceso a base de datos.
- Bus: contiene las clases que implementan la lógica de negocio de la aplicación.
- Deploy: contiene las clases interfaz de la aplicación.
- Util: contiene las clases de ayuda.

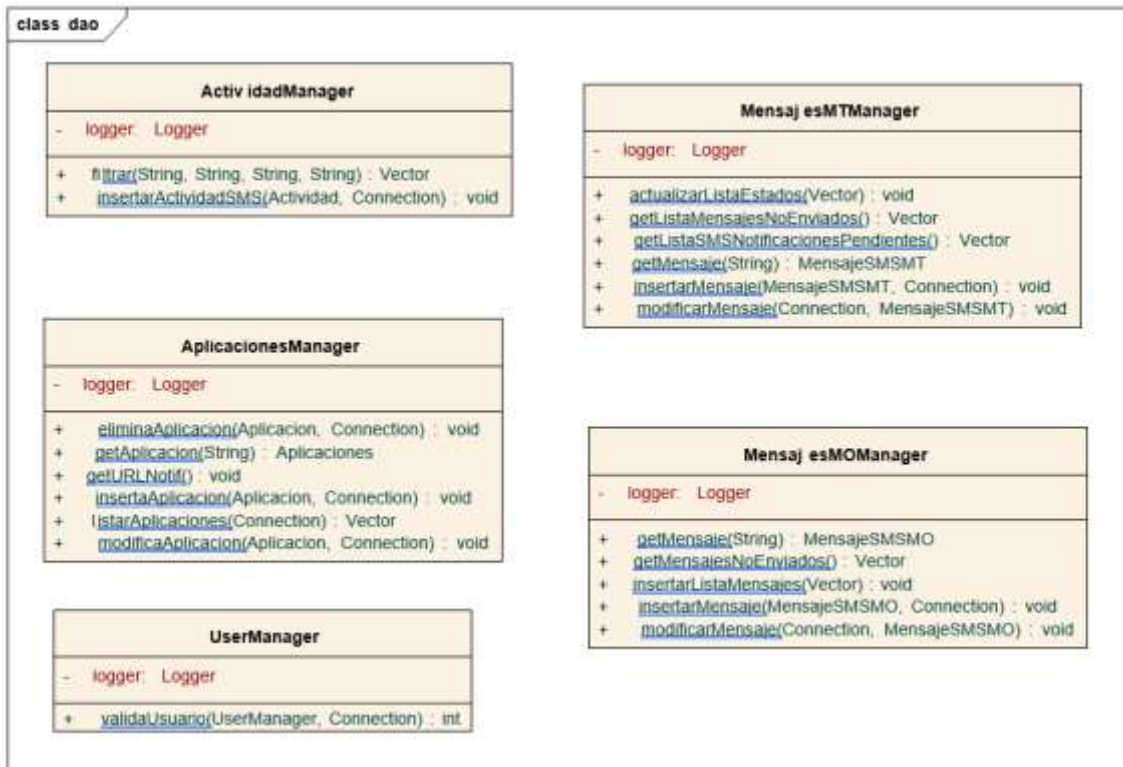
Paquete es.metromadrid.mensajeriasms.bo



Las clases de negocio son las siguientes:

- Actividad: almacena en memoria la información relativa a una actividad (incidencia) producida.
- Módem: datos relativos del módem
- Config: datos de la configuración general de la aplicación.
- MensajeSMSMT: datos de los SMS MT
- MensajeSMSMO: datos de los SMS MO
- Aplicación: datos relativos con la información de una aplicación dada de alta en nuestro sistema.
- Usuario: datos de un usuario (administrador o consulta) de nuestro sistema.
- ErroresMensajeríaGsm: conjunto de constantes que almacenan los diferentes errores que se pueden producir en nuestro sistema.

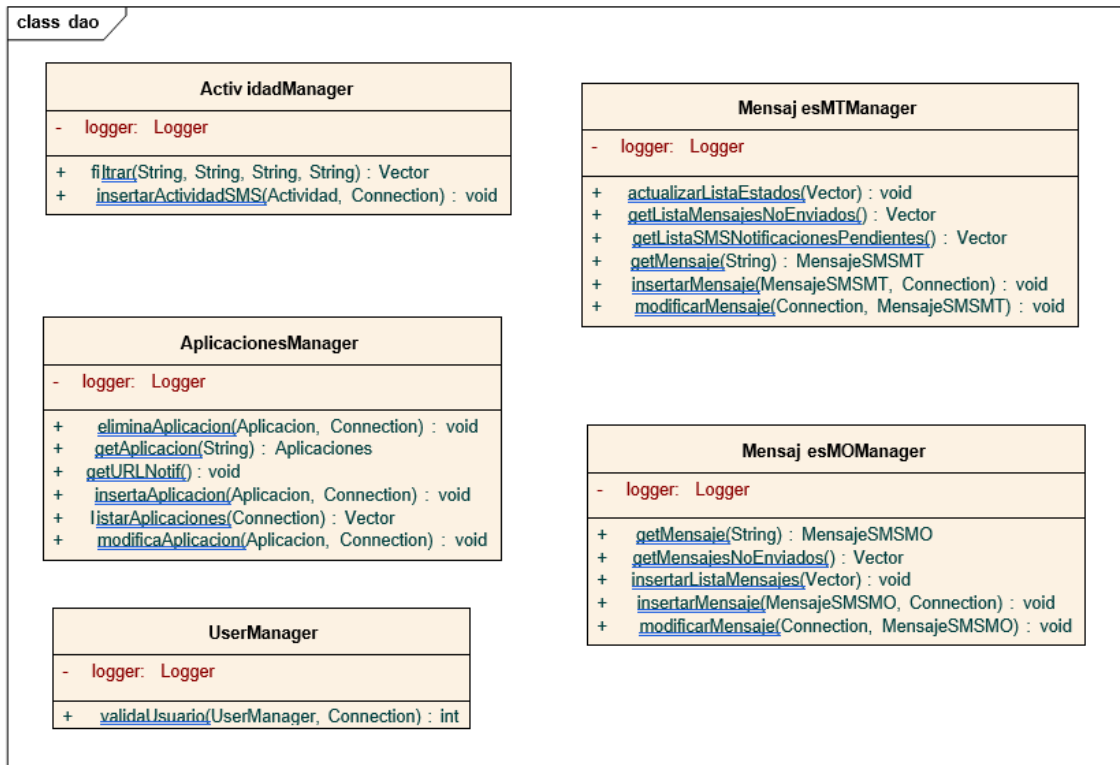
Paquete es.metromadrid.mensajeriasms.dao



Las clases que realizan las operaciones con la base de datos. Cada clase realiza las operaciones necesarias sobre una tabla, dichas clases son:

- **ActividadManager:** gestiona las operaciones sobre la tabla de Actividades (insertar, filtrar).
- **AplicacionesManager:** gestiona las operaciones sobre la tabla de Aplicaciones (insertar, eliminar, modificar, listar y validar).
- **UserManager:** gestiona las operaciones sobre la tabla de Aplicaciones (validar)
- **MensajesMTManager:** gestiona las operaciones sobre los mensajes MT
- **MensajesMOManager:** gestiona las operaciones sobre los mensajes MO.

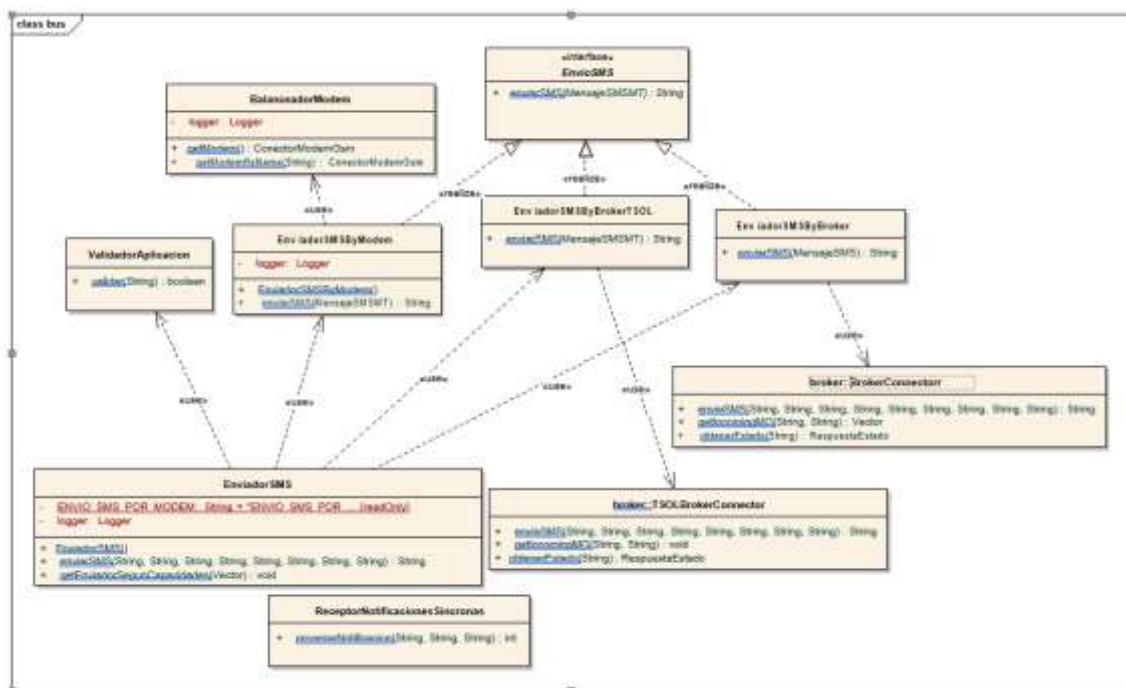
Paquete es.metromadrid.mensajeriasms.dao



Clases que realizan las operaciones con la base de datos. Cada clase realiza las operaciones necesarias sobre una tabla, dichas clases son:

- ActividadManager: gestiona las operaciones sobre la tabla de Actividades (insertar, filtrar).
- AplicacionesManager: gestiona las operaciones sobre la tabla de Aplicaciones (insertar, eliminar, modificar, listar y validar).
- UserManager: gestiona las operaciones sobre la tabla de Aplicaciones (validar)
- MensajesMTManager: gestiona las operaciones sobre los mensajes MT
- MensajesMOManager: gestiona las operaciones sobre los mensajes MO.

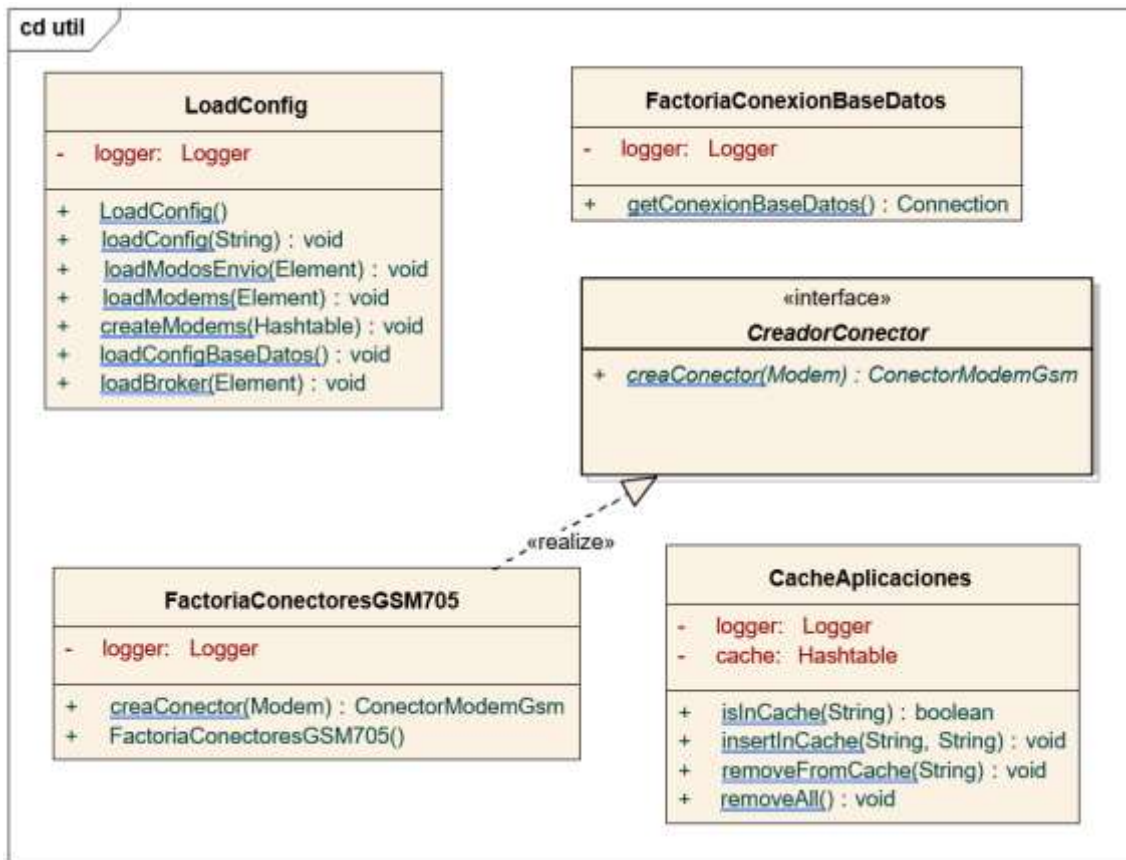
Paquete es.metromadrid.mensajeriasms.bus



Este conjunto de clases implementan la lógica de negocio de la aplicación, que son:

- **BalanceadorModem**: se encarga de seleccionar el primer módem disponible.
- **EnvioSMSByModem**: realiza las tareas necesarias para el envío de un SMS a través del módem.
- **EnvioSMSByBroker**: realiza las tareas necesarias para el envío de un SMS a través del bróker.
- **ValidadorAplicacion**: valida si una aplicación que solicita el envío de un SMS está dada de alta en el sistema.
- **EnvioSMS**: Es la interfaz que deben implementar todas las clases que realicen el envío. Dicha clase además de realizar el envío del mensaje deberá implementar el método `getEnvioSegunCapacidades` que permitirá seleccionar el tipo de envío a utilizar (modem, bróker), el funcionamiento de dicha operación será el siguiente:
 1. El aplicativo tendrá cargados en memoria todos los tipos de envíos disponibles en el fichero de configuración, los cuales están ordenados por las prioridades especificadas en dicho fichero.
 2. Dependiendo de las capacidades seleccionadas por la AE en la petición (Ventana de envío, SMS Certificado..., etc.) el aplicativo comenzará a comprobar uno a uno los tipos de envío de los que dispone hasta encontrar uno que disponga de las capacidades seleccionadas (en caso de que varios lo cumplan se cogerá el de mayor prioridad).
 3. A continuación, se enviará el mensaje utilizando dicho tipo de envío (modem, bróker).
- **ReceptorNotificacionesSincronas**: Encargada de atender las solicitudes de cambio de estado de mensajes solicitadas por la AE de forma síncrona.

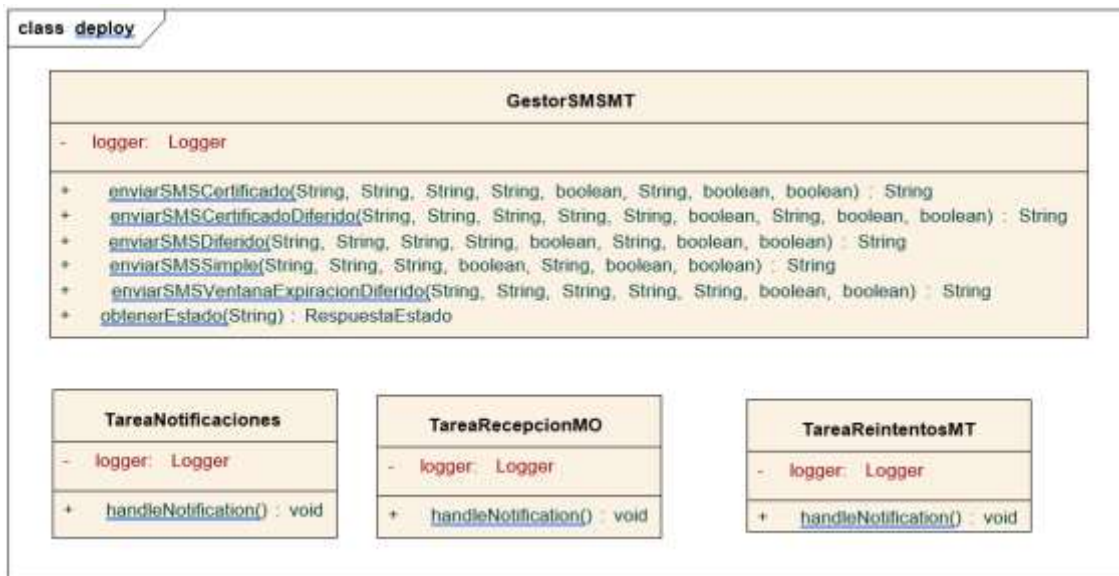
Paquete es.metromadrid.mensajeriasms.util



Diversas clases que sirven de apoyo para poder llevar a cabo varias funcionalidades:

- LoadConfig: se encarga de cargar en memoria la información contenida en los distintos ficheros de configuración del sistema.
- FactoríaConexionBaseDatos: encargada de toda la gestión con la base de datos.
- FactoriaConectoresGSM705: su tarea es la de gestión de los módems de un cierto tipo.
- CacheAplicaciones: gestiona el “cacheado” de la información de las aplicaciones en memoria.
- CreadorConector: interfaz que especifica el método a implementar a la hora de crear un conector a un módem.

Paquete es.metromadrid.mensajeriasms.deploy



Contiene 4 clases:

- GestorSMSMT: Las AE invocarán a este servicio para el envío de mensajes MT y la consulta de estado de mensajes de forma síncrona. Los métodos de los que dispone dicho servicio son los siguientes:
 1. envíoSMSSimple: Permite el envío de SMS simple, el envío será realizado por el bróker. Los parámetros que recibe dicha operación son los siguientes:
 - String idApp: Identificador de la aplicación que realiza el envío.
 - String teléfono: Teléfono al que se envía el mensaje.
 - String texto: Texto del mensaje
 - boolean notificar: Indica si se quieren recibir notificaciones de cambio de estado.
 - String remitente: Indica si se quiere especificar un remitente para el mensaje.
 - boolean partirSMS: Si el mensaje ocupa más de 160 caracteres determina si se partirá o no.
 - boolean reintentar: Permite activar el reintento del mensaje en el caso de que no se pueda realizar el envío.

Como parámetro de retorno devolverá el identificador asociado al mensaje o el error producido.

2. envíoSMSCertificado: Permite el envío de SMS certificado. Los parámetros que recibe dicha operación son los siguientes:
 - String idApp: Identificador de la aplicación que realiza el envío.
 - String teléfono: Teléfono al que se envía el mensaje.
 - String texto: Texto del mensaje
 - String mail: Correo electrónico al que se enviará el certificado de la entrega.
 - boolean notificar: Indica si se quieren recibir notificaciones de cambio de estado.
 - String remitente: Indica si se quiere especificar un remitente para el mensaje.

- boolean partirSMS: Si el mensaje ocupa más de 160 caracteres determina si se partirá o no.
- boolean reintentar: Permite activar el reintento del mensaje en el caso de que no se pueda realizar el envío.

Como parámetro de retorno devolverá el identificador asociado al mensaje o el error producido.

3. envioSMSCertificadoDiferido: Permite el envío de SMS certificado diferido. Los parámetros que recibe dicha operación son los siguientes:
 - String idApp: Identificador de la aplicación que realiza el envío.
 - String teléfono: Teléfono al que se envía el mensaje.
 - String texto: Texto del mensaje
 - String mail: Correo electrónico al que se enviará el certificado de la entrega.
 - String entregaDiferida: Indica cuando se realizará la entrega diferida.
 - boolean notificar: Indica si se quieren recibir notificaciones de cambio de estado.
 - String remitente: Indica si se quiere especificar un remitente para el mensaje.
 - boolean partirSMS: Si el mensaje ocupa más de 160 caracteres determina si se partirá o no.
 - boolean reintentar: Permite activar el reintento del mensaje en el caso de que no se pueda realizar el envío.

Como parámetro de retorno devolverá el identificador asociado al mensaje o el error producido.

4. envioSMSDiferido: Permite el envío de SMS diferido. Los parámetros que recibe dicha operación son los siguientes:
 - String idApp: Identificador de la aplicación que realiza el envío.
 - String teléfono: Teléfono al que se envía el mensaje.
 - String texto: Texto del mensaje
 - String entregaDiferida: Indica cuando se realizará la entrega diferida.
 - boolean notificar: Indica si se quieren recibir notificaciones de cambio de estado.
 - String remitente: Indica si se quiere especificar un remitente para el mensaje.
 - boolean partirSMS: Si el mensaje ocupa más de 160 caracteres determina si se partirá o no.
 - boolean reintentar: Permite activar el reintento del mensaje en el caso de que no se pueda realizar el envío.

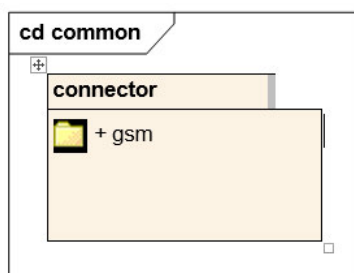
Como parámetro de retorno devolverá el identificador asociado al mensaje o el error producido.

5. obtenerEstado: comprobar el estado de un mensaje previamente enviado. Los parámetros que recibe dicha operación son los siguientes:
 - String idMensaje: Identificador del mensaje del cual se quiere conocer el estado.

Como parámetro de retorno devolverá el estado del mensaje.

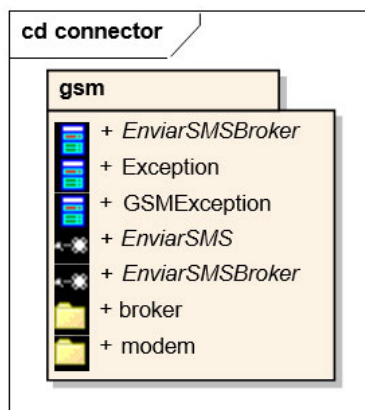
- TareaNotificaciones: Tarea Asíncrona que cada cierto tiempo configurable consulta al bróker el estado de los mensajes notificables que haya en BBDD.
- TareaRecepcionMO: Tarea Asíncrona que cada cierto tiempo configurable consulta al bróker si dispone de nuevos mensajes MO.

12.3.1.2. Paquete es.metromadrid.common

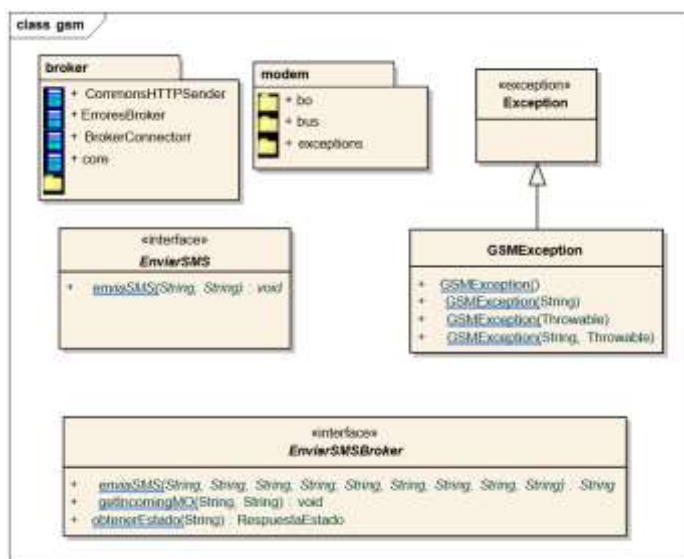


Contiene las clases específicas del conector para el módem gsm y el conector de un bróker específico, si se quisiese ampliar el espectro de agentes de mensajería.

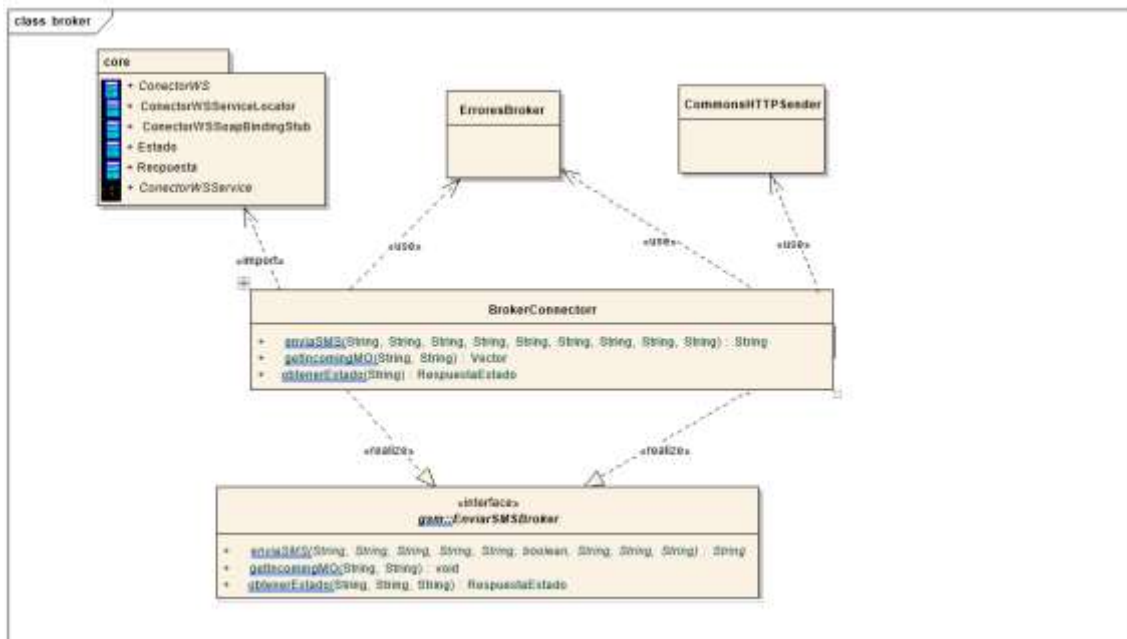
Este paquete contiene a su vez el paquete conector, cuyo contenido es:



A su vez contiene el paquete gsm, que contiene las clases y paquetes que implementan la lógica del módem GSM:



Paquete es.metromadrid.common.connector.gsm.broker

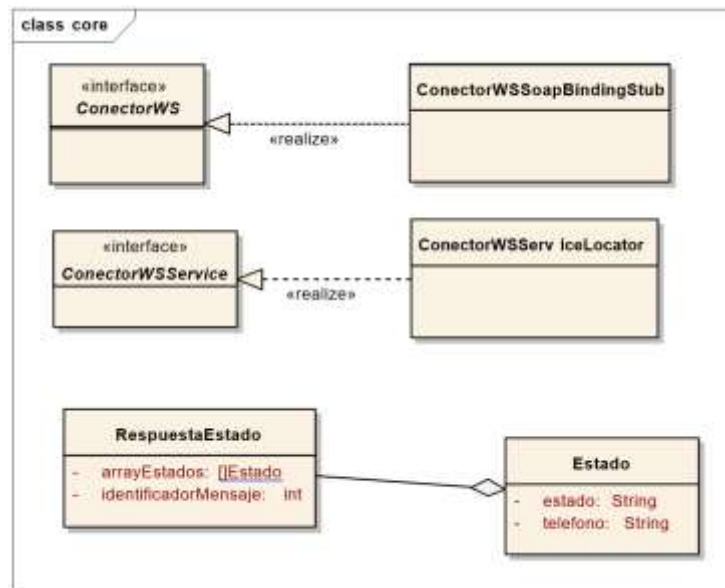


Son las clases de negocio del bróker:

- ErroresBroker: los errores tipificados dentro de la aplicación para el bróker.
- BrokerConnector: Clases que implementa la conexión con el bróker, para facilitar futuras ampliaciones con nuevos conectores implementan una interfaz común con las siguientes operaciones:
 1. enviaSMS: Permite realizar el envío del mensaje, recibe como parámetros todas las posibilidades de envío:
 - String idApp: Identificador de la aplicación que realiza el envío.
 - String teléfono: Teléfono al que se envía el mensaje.
 - String texto: Texto del mensaje
 - String mail: indica que el mensaje es por envío certificado y dicho envío ha de hacerse al correo electrónico indicado en este parámetro. String entregaDiferida: Indica cuando se realizará la entrega diferida.
 - String expiración: Indica la expiración del mensaje.
 - boolean notificar: Indica si se quieren recibir notificaciones de cambio de estado.
 - String remitente: indica si se quiere especificar un remitente para el mensaje.
 - boolean reintentar: Permite activar el reintento del mensaje en el caso de que no se pueda realizar el envío.
 2. getIncomingMO: permite conectarse al bróker para recuperar los MO pendientes de ser recibidos, los parámetros que admite son:
 - String usuario: Usuario asignado a metro de Madrid
 - String password: Password asignado a metro de Madrid.
 3. obtenerEstado: Permite obtener el estado de un mensaje enviado, los parámetros que recibe son:
 - String usuario: Usuario asignado a metro de Madrid
 - String password: Password asignado a metro de Madrid.
 - String idMensaje: Identificador del mensaje del que se solicita conocer el estado.

- CommonsHTTPSender: Cliente http usado por el conector del bróker.
- Paquete core: implementa el servicio especificado por el WSDL.

Paquete es.metromadrid.common.connector.gsm.broker.core



Son las clases usadas para la conexión con el bróker. Dicho paquete consta de las siguientes clases:

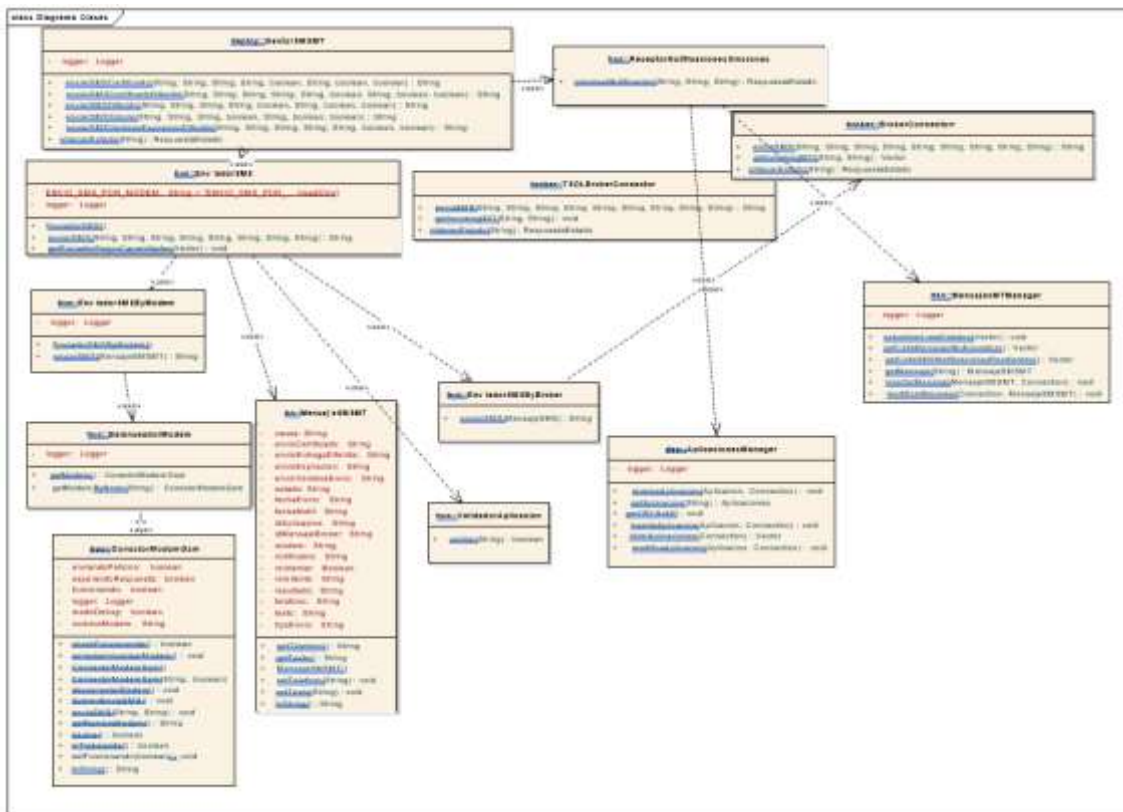
- RespuestaEstado: Es el objeto complejo utilizado por los conectores a los bróker como respuesta a la operación de notificación, contiene el identificador del mensaje que ha sido enviado y la lista de estados del mismo para el caso en el que haya sido dividido en partes.
- Estado: Define el estado de cada envío.
- ConectorWS: Interfaz a implementar por el cliente para la invocación remota de métodos.
- ConectorWSService: Interfaz a implementar por el cliente del servicio del bróker para permitir el establecimiento de la comunicación con el mismo.
- ConectorWSServiceLocator: Clase que permite el establecimiento de la comunicación con el servicio, implementa la interfaz ConectorWSService
- ConectorWSSoapBindingStub: Clase conectora final con el servicio, implementa la interfaz ConectorWS.

12.3.1.3. Librerías externas

Se han usado las siguientes librerías externas:

- El paquete 'Log4Java' es un paquete estándar y 'opensource' que implementa la funcionalidad de 'logging' de la aplicación. Ver <http://logging.apache.org/log4j/docs/index.html> para más información.
- 'Commons DBCP' es un paquete estándar y 'opensource' que implementa la funcionalidad de pool de base de datos. Ver <http://jakarta.apache.org/commons/dbcp/> para más información.
- OracleDriver, contiene las clases necesarias para acceder a una base de datos oracle. Ver <http://www.oracle.com> para más información.
- 'Axis', conjunto de clases que implementan el acceso web services. Ver <http://ws.apache.org/axis/> para más información.

12.3.2. Diagrama de clases



12.3.3. Trazabilidad clases de análisis y diseño

12.3.3.1. Clases de Análisis de Control vs Clases de Diseño

DISEÑO /ANALISIS	<<control>>			
	RecepcionPeticones	ConectorBroker	TareaNotificaciones	EnvioSMS
mensajeriasms:bus:EnvioSMS	X			X
mensajeriasms:bus:EnvioSMSByBROKER		X		
mensajeriasms:bus:ReceptorNotificaciones			X	

12.3.3.2. Clases de Análisis de Entidad vs Clases de Diseño

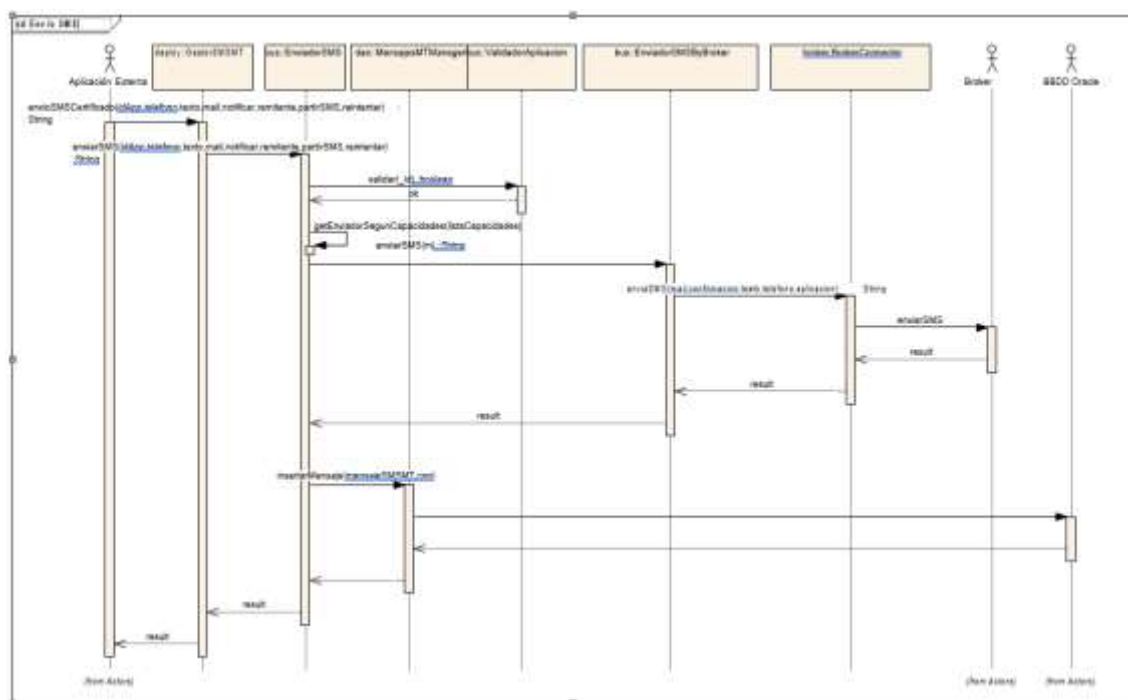
DISEÑO /ANALISIS	MensajeSMS<<entity>>	Aplicación<<entity>>
mensajeria::dao::MensajesMTManager	X	
mensajeria::dao::MensajesMOManager	X	
mensajeria::dao::Aplicaciones		X

12.3.3.3. Clases de Análisis de Interfaz vs Clases de Diseño

DISEÑO /ANALISIS	EnvioSMS<<interface>>	ReceptorNotificaciones<<interface>>
deploy::GestorSMSMT	X	X

12.3.4. Diagramas de secuencia

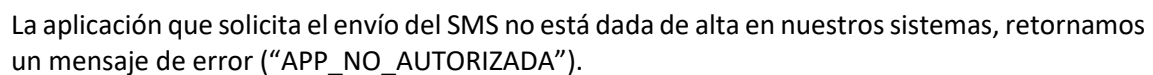
12.3.4.1. Enviar MT por bróker



En este diagrama se representa la situación de un envío que se realiza correctamente. Los pasos a seguir son los siguientes:

- La AE invoca al Web Service para mandar un SMS, dependiendo de las opciones de envío que quiera utilizar la AE invocará a un método u otro del servicio. En el diagrama anterior se puede ver la invocación al método `envioSMSCertificado` que permite el envío certificado del mensaje a través del bróker. Los parámetros que se pasan en esta llamada son los siguientes:
 - String `idApp`: Identificador de la aplicación que realiza el envío.
 - String `teléfono`: Teléfono al que se envía el mensaje.
 - String `texto`: Texto del mensaje
 - String `mail`: Correo electrónico al que se enviará el certificado de la entrega.
 - boolean `notificar`: Indica si se quieren recibir notificaciones de cambio de estado.
 - String `remitente`: Indica si se quiere especificar un remitente para el mensaje.
 - boolean `partirSMS`: Si el mensaje ocupa más de 160 caracteres determina si se partirá o no.
 - boolean `reintentar`: Permite activar el reintento del mensaje en el caso de que no se pueda realizar el envío.
- El Web Service delega toda la operativa a la clase `EnviadorSMS`. Con esto conseguimos total independencia entre la interfaz con el usuario y la lógica de la aplicación. El `EnviadorSMS` se encarga principalmente de procesar el mensaje, ver si la aplicación está dada de alta.
- El `enviadorSMS` deberá de comprobar por que bróker se va a realizar el envío, para ello invoca al método `getEnviadorSegúnCapacidades` pasándole la lista de capacidades de envío que ha solicitado la AE, a continuación, empieza a recorrer la lista de brókers configurados y selecciona aquel bróker configurado con mayor prioridad que cumpla con las capacidades de envío que ha solicitado la AE
- `EnviadorSMS` se encarga de insertar en base de datos el envío de este SMS.

- #### 12.3.4.2. Error validación aplicación



```

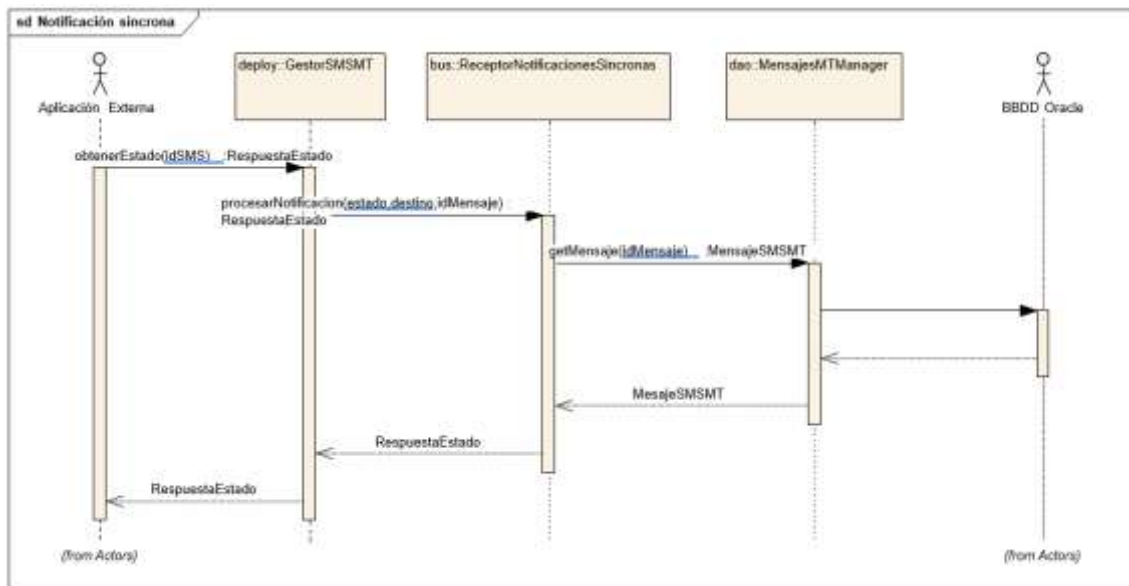
sequenceDiagram
    participant Deploy as Deploy-TanahReinventorMT
    participant EnvManager as Env-MessageMTManager
    participant EnvBroker as Env-EnviroSMSByBrokerTSQL
    participant BrokerTSQL as Broker-TSQLBrokerConnector
    participant EnvBroker2 as Env-EnviroSMSByBroker
    participant Broker2 as Broker-BrokerConnector
    participant Broker as Broker
    participant Broker as Broker

    Deploy->>Deploy: handReinventor
    Deploy->>EnvManager: getSMSDataFromDatabase...
    EnvManager->>Deploy: dataMessage
    Deploy->>EnvBroker: [For code message] enviroSMSmessage(SMSMT)...
    EnvBroker->>BrokerTSQL: enviroSMS(message(SMSMT)..., string)
    BrokerTSQL->>EnvBroker: result
    EnvBroker->>Deploy: result
    Deploy->>EnvManager: rootReinventorMessage(SMSMT)
    EnvManager->>Deploy: Ok
    Deploy->>EnvBroker2: [For Code message] enviroSMSmessage(SMSMT)...
    EnvBroker2->>BrokerTSQL: enviroSMS(message(SMSMT)..., string)
    BrokerTSQL->>EnvBroker2: result
    EnvBroker2->>Deploy: result
    Deploy->>EnvManager: rootReinventorMessage(SMSMT)
    EnvManager->>Deploy: Ok
    
```


Secuencia:

- Cada cierto tiempo configurable vence la tarea de reenvío de MT.
- Se recuperan los mensajes de BBDD que no hayan podido ser enviados, se tomarán los de las últimas X horas siendo X configurable.
- Por cada mensaje que hubiera sido intentado enviar se intenta realizar el envío nuevamente, se utilizarán las mismas opciones de envío que en el intento anterior (Envío SMS certificado, ventana de envío etc..). Se modificará el estado de los mensajes enviados correctamente en la BBDD.

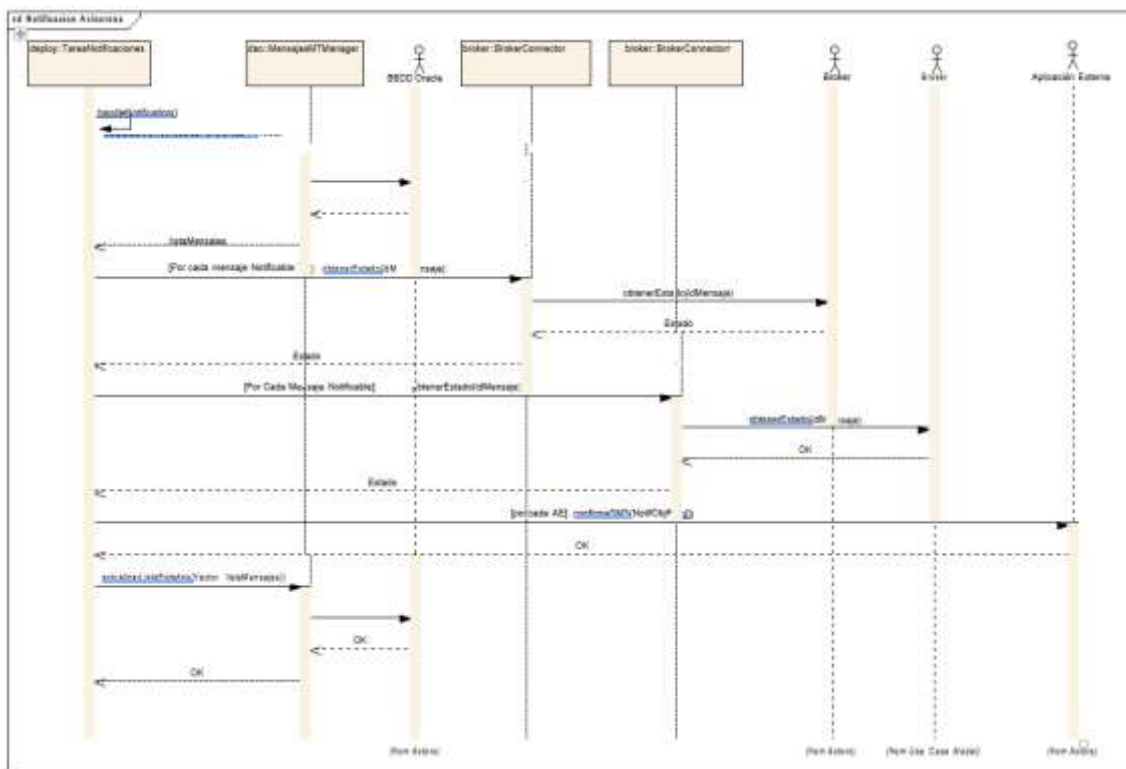
12.3.4.4. Notificación síncrona



Secuencia:

- Solicita el estado de SMS a partir del id devuelto por la llamada de envío SMS.
- Como respuesta recibe un entero que indica cómo ha ido la petición (<0 error) y un array de teléfono-estado.

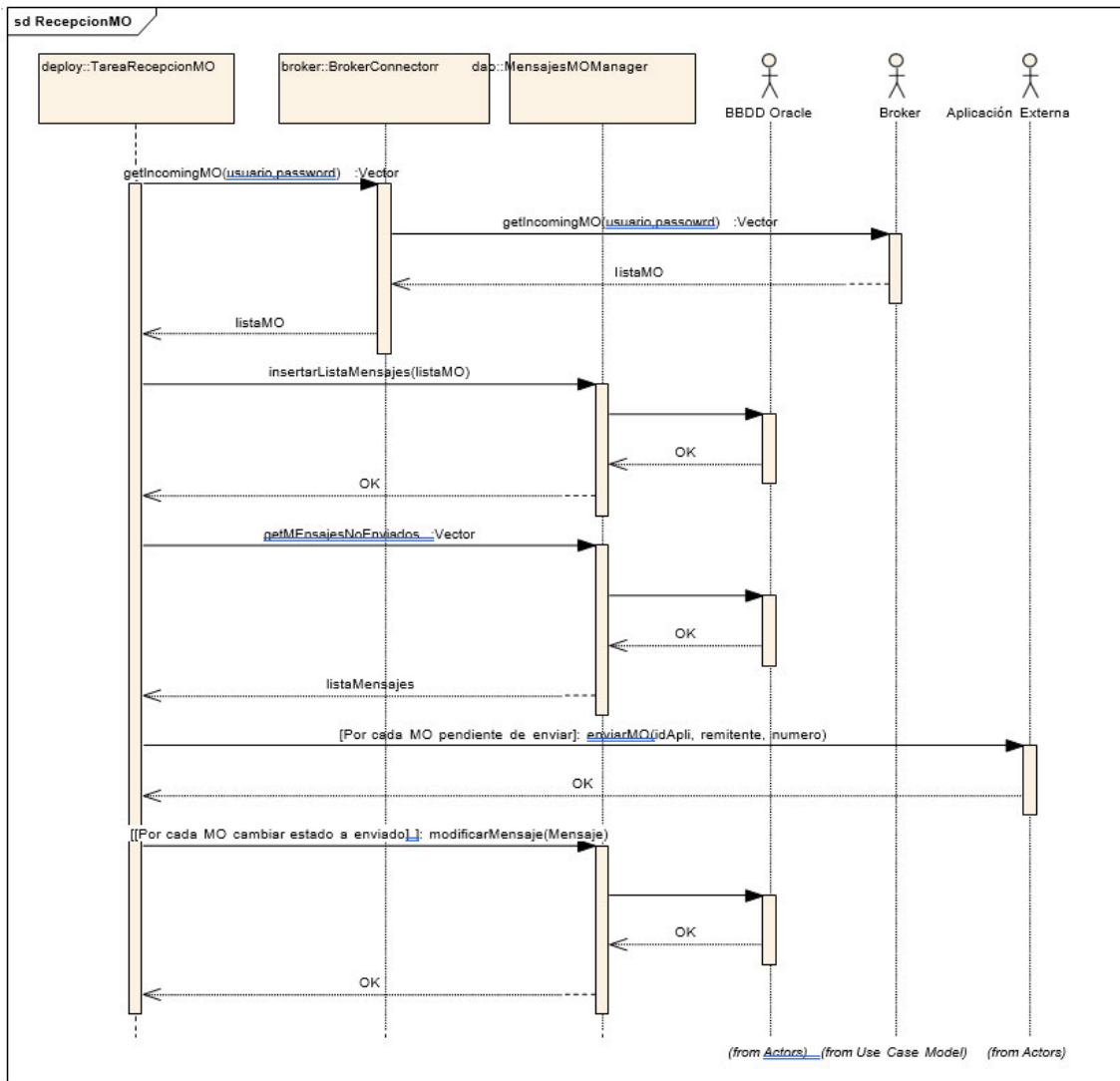
12.3.4.5. Notificación asíncrona



Cada cierto tiempo configurable se ejecuta la tarea de notificaciones, dicha tarea realiza las siguientes acciones:

- Obtiene de BBDD la lista de mensajes pendientes de notificar
- Agrupa dicha lista de mensajes por bróker
- Por cada mensaje se realiza una petición al bróker para consultar su estado.
- Cuando se ha consultado el estado de todos los mensajes se procede a comunicarlo a las AE, por cada AE se invoca a su servicio de notificaciones pasándole el listado de los mensajes con sus estados
- Por último, se actualizan los mensajes en BBDD con su nuevo estado.

12.3.5. Recepción de MO



Periódicamente se lanzará una tarea para el tratamiento y posterior envío de los MO a las distintas AE. Los pasos a realizar por la tarea serán los siguientes:

- Invocar al bróker para obtener los nuevos MO entrantes.
- Dichos MO se guardarán como pendientes de enviar en la tabla correspondiente de base de datos. Para ello se procesa el contenido del mensaje, separando la palabra clave (que identificará a la aplicación externa destino) del texto.
- Se obtienen todos los MO pendientes de enviar de la base de datos (tanto los obtenidos en el primer paso como los que no han podido ser enviados en iteraciones anteriores)
- Por cada MO pendiente de enviar:
 1. Se obtiene la URL de la AE asociada a la palabra clave.
 2. Se invoca al servicio de la AE pasando como parámetro el MO recibido.
 3. Si la AE procesa correctamente el mensaje, se modifica su estado como enviado.

12.4. Diseño de interfaz de usuario

12.4.1. Definición de páginas

- Consulta Estadísticas MT:
 1. Datos de Entrada (Filtro):

- Aplicación
 - Fecha de inicio
 - Fecha de Fin
 - Teléfono
 - Resultado de envío (TODOS, OK, NOK)
 - Tipo Envío (TODOS, BROKER, MODEM)
- 2. Datos de salida: se mostrarán todos los registros de la tabla Mensajes que cumplan el filtro introducido. Los campos son:
 - idAplicacion
 - Teléfono
 - Texto
 - Resultado
 - Causa
 - Tipo Envío
 - Estado (si envío mediante bróker)
 - Id Mensaje Bróker
 - Nombre del Módem (si procede)
 - Fecha y hora del envío
 - Notificable (Si se ha solicitado notificación de cambio de estado del mensaje)
- Consulta Estadísticas MO:
 - 1. Datos de Entrada (Filtro):
 - Aplicación
 - Fecha de inicio
 - Fecha de Fin
 - Teléfono Origen
 - Resultado de envío (ENTREGADO, NOENTREGADO)
 - 2. Datos de salida: se mostrarán todos los registros de la tabla MensajesMO que cumplan el filtro introducido. Los campos son:
 - idAplicacion
 - Teléfono
 - Texto
 - Resultado
 - Id Mensaje Bróker
 - Fecha Recepción
 - Fecha Envío AE
 - Estado Entrega
- Consulta Actividad:
 - 1. Datos de Entrada (Filtro)
 - Fecha Inicio
 - Fecha Fin
 - Código error
 - Tipo de envío
 - Módem (sólo si el código error está vinculado a error de módem, será responsabilidad del usuario el saber relación códigos error-módem)
 - 2. Datos de salida: se mostrarán todos los registros de la tabla Actividad que cumplan el filtro introducido. Los campos son:
 - Fecha y Hora
 - Severidad
 - Código Error
 - Texto Error

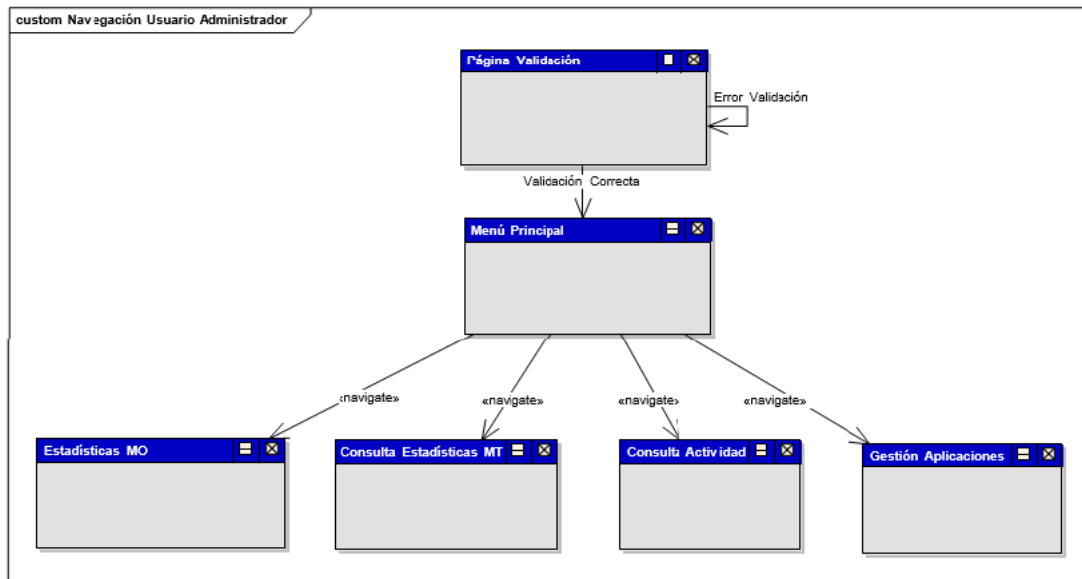
▪ Módem

- Gestión de aplicaciones, se permitirán las siguientes acciones:

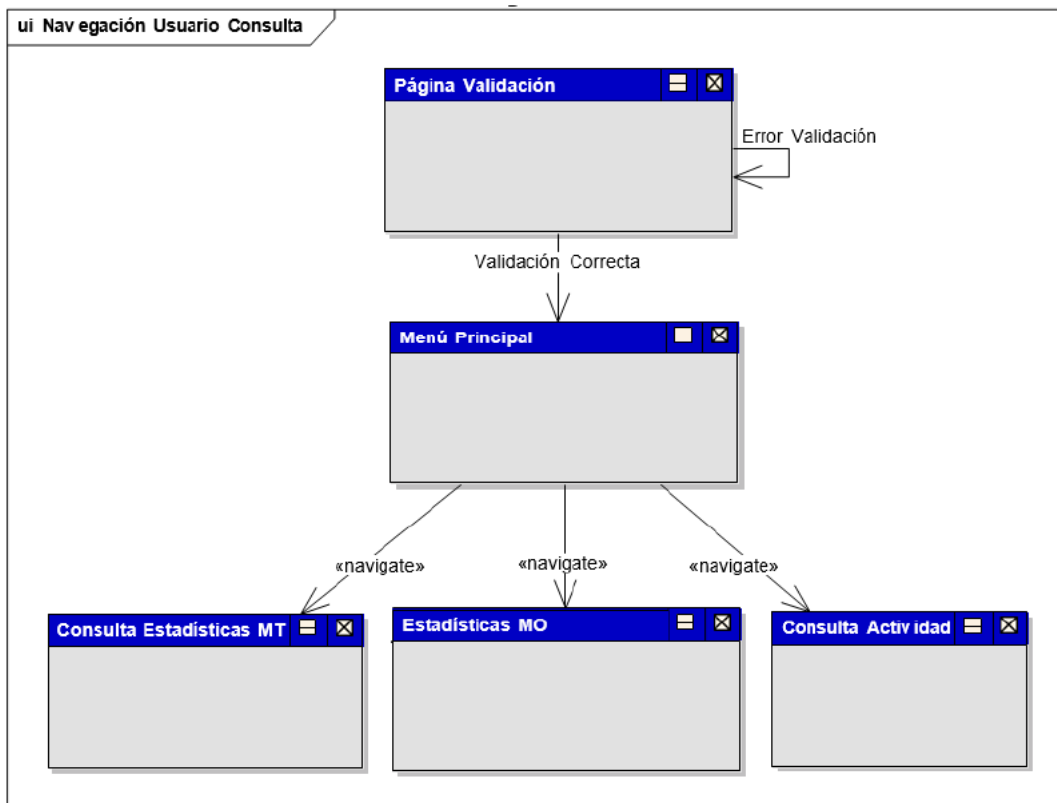
3. Alta de aplicación
4. Baja de aplicación
5. Modificación de aplicación
6. Listado de aplicaciones dadas de alta en el sistema.

12.4.2. Flujo de navegación de aplicación

El usuario administrador tiene estas posibilidades de navegación:



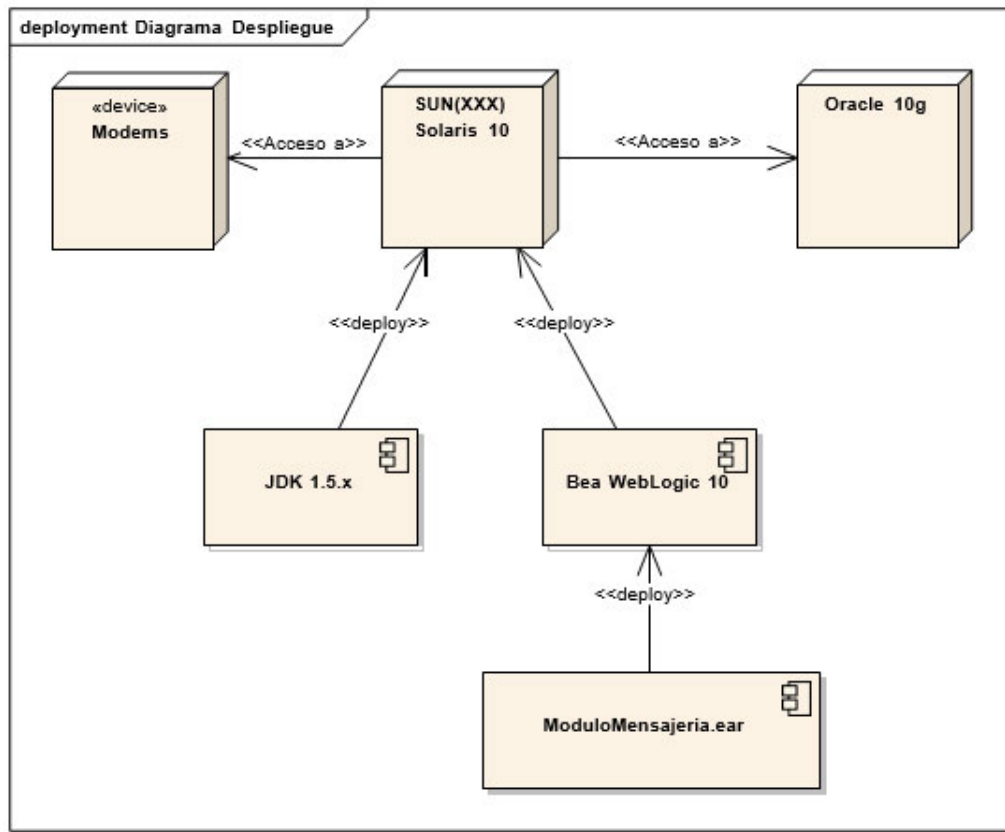
El usuario Consulta tiene este árbol de navegación:



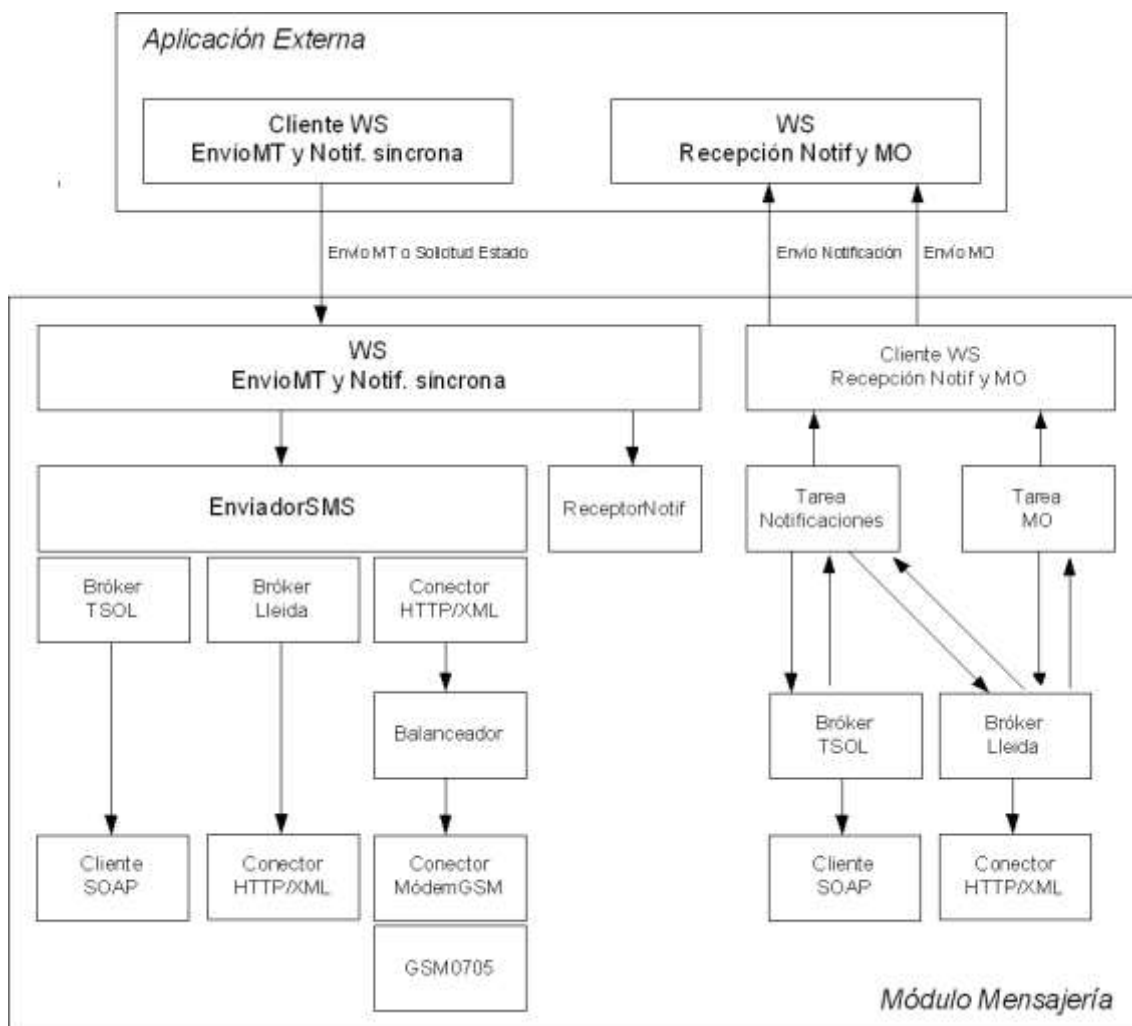
12.5. Diseño físico de datos

TABLA	CAMPO	CONCEPTO	TIPO	PK	FK
TMSWA01AP	NN APIDAPLI	Identificador de la aplicación	VARCHAR2(50)	SI	
	NN APNOMBRE	Nombre de la aplicación	VARCHAR2(50)		
	NN APMOCLAV	Palabra Clave identificativa de MO	VARCHAR2(10)		
	N APURLMO	Url de MO	VARCHAR(1000)		
	N APURLNOT	Url de Notif	VARCHAR(1000)		
TMSWA02AC	NN ACIDACTI	Identificador de la actividad	NUMBER(6)	SI	
	NN ACFECHA	Fecha de inserción de la Actividad	DATE		
	NN ACSEVERI	Nivel de severidad de la Actividad	VARCHAR2(10)		
	NN ACCODIGO	Código del error	VARCHAR2(4)		
	N ACTEXTO	Texto descriptivo del error	VARCHAR2(1000)		
	N ACMODEM	Nombre del módem	VARCHAR2(50)		
TMSWA03US	NN USIDUSUA	Identificador del usuario	NUMBER(2)	SI	
	NN USLOGIN	Login del usuario	VARCHAR2(25)		
	NN USPASSWO	Password del Usuario	VARCHAR2(25)		
	NN USROL	Perfil del usuario	VARCHAR2(10)		
TMSWA04ME	NN MEIDMENS	Identificador del mensaje	NUMBER(6)	SI	
	NN MEIDAPLI	Identificador de la aplicación	VARCHAR2(50)		SI*
	NN METELEFO	Teléfono destino	VARCHAR2(12)		
	NN METEXTO	Texto del SMS	VARCHAR2(1000)		
	NN MERESULT	Resultado del envío	VARCHAR2(4)		
	NN FECHANOTIF	Fecha de última notificación	DATE		
	NN MEFECHA	Fecha del envío	DATE		
	NN METENVIO	Tipo de envío	VARCHAR2(15)		
	N MECAUSA	Texto del error	VARCHAR2(1000)		
	N MEMODEM	Nombre del módem	VARCHAR2(50)		
	N MEESTAME	Estado de envío	VARCHAR(4)		
	NN MENOTIF	Si el mensaje es notificable	BOOLEAN		
	N MEIDBROK	Id sms del broker	VARCHAR(10)		
	NN MEEXP	Opción de envío con expiración	VARCHAR2(50)		
	NN MEVENTENVIO	Opción de envío ventana de envío	VARCHAR2(50)		
	NN MEENTDIFERIDA	Opción de envío entrega diferida	VARCHAR2(50)		
	NN MEENCERT	Opción de envío Certificado	VARCHAR2(50)		
	NN MERINTMT	Determina si se reintenta en los MT	BOOLEAN		
	NN MEREMIT	Remitente del mensaje	VARCHAR(20)		
	NN MEOWNER	Maquina que ha realizado la inserción	VARCHAR(60)		
TMSWA05MO	NN MOIDMENS	Identificador del mensaje	NUMBER(6)	SI	
	NN MOIDAPLI	Identificador de la aplicación	VARCHAR2(50)		SI*
	NN MOTELEFO	Teléfono origen	VARCHAR2(12)		
	NN MOTEXTO	Texto del SMS	VARCHAR2(1000)		
	NN MOCORTO	Número al que se ha enviado el MO	VARCHAR2(12)		
	NN MOFECHAENV	Fecha de envío a la AE	DATE		
	NN MOFECHAREC	Fecha de Recepción	DATE		
	NN MOESTAENT	Estado de la Entrega	VARCHAR(20)		

12.6. Diagrama de despliegue



12.7. Arquitectura de la solución



13. ANEXO II: APLICACIÓN METROSMS

13.1. Descripción del sistema

13.1.1. Introducción

El objeto del presente apartado es la elaboración del análisis y diseño del módulo de recepción de mensajes originados, desde un móvil a Metro de Madrid, notificaciones de estado de los envíos de los mensajes creados desde Metro y envío de mensajes desde cualquier operadora. Dentro de esta orden de cambio el módulo debe contemplar las siguientes funcionalidades:

- Recepción mediante el uso de los servicios web de las notificaciones de los estados de los mensajes enviados por Metro de Madrid al exterior.
- Recepción, mediante el uso de los servicios web de los mensajes originados desde el exterior a Metro de Madrid.
- Una interfaz WebService que facilita a distintas aplicaciones un acceso común y sencillo para enviar SMS a los potenciales usuarios de dichas aplicaciones.
- El intercambio de información se realiza mediante un web services.
- Homogeniza el comportamiento de los distintos canales de envío integrados en la plataforma.
- Sencilla integración de canales de envío nuevos.

- Implementa un mecanismo de autenticación de las aplicaciones cliente, restringiendo el acceso sólo a las aplicaciones autorizadas en la plataforma.
- Funcionalidades de envío de SMS por parte de aplicaciones:
 1. Entrega diferida.
 2. Envío de SMS Certificados.
 3. Notificación a las aplicaciones cliente del estado de un mensaje.
 4. Envío de mensajes.
- Transmisión a las aplicaciones clientes de SMS originados por los usuarios.

13.1.2. Especificaciones técnicas

Requerimientos técnicos para la correcta instalación de la aplicación:

- Requerimientos HW:
 1. RAM: 2 Gb.
 2. Espacio en disco: 1GB
- Requerimientos SW:
 1. BEA WebLogic Server 10.1

13.1.3. Diagrama de bloques

Diagrama de bloques funcional de MetroSMS:

Diagrama de entrada (cuando sea de aplicación, para bróker que inicien la comunicación):

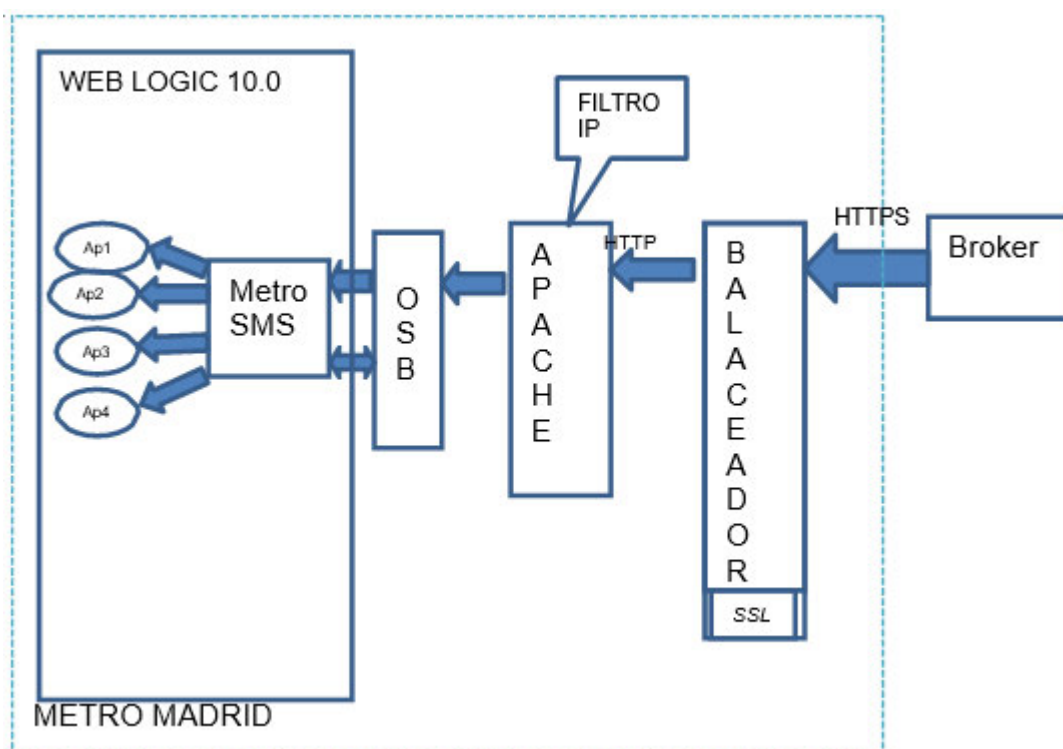
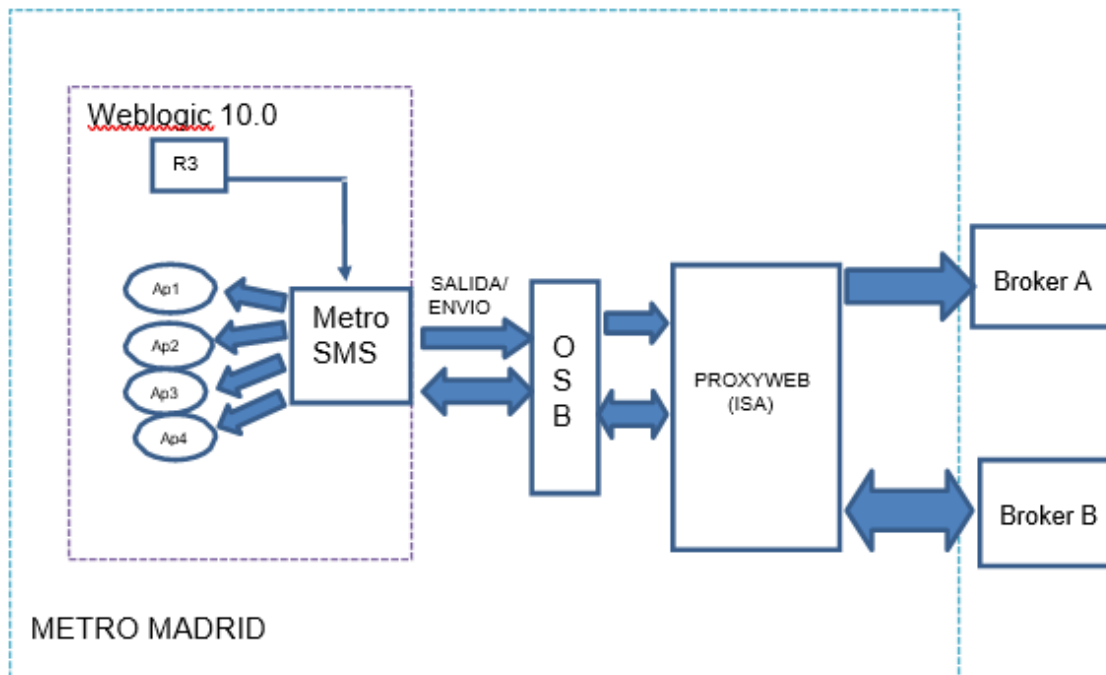


Diagrama de salida para Brókers que inicien la comunicación (tipo A) y de entrada/salida para otros Brókers (tipo B):



Consultar el diagrama detallado al final de este documento.

13.1.4. Interfaces

Interfaz con BBDD

Los datos de conexión con la base de datos que almacena los datos de la aplicación son los siguientes:

- Usuario:
- Nombre del DataSource: <BEA>_METROSMS_JDBC
- Nombre JNDI del DataSource: jdbc/metrosmsdatasource

En la siguiente tabla se muestra los valores que puede tomar <BEA> en el nombre del DataSource:

Entorno	Valor/es
Desarrollo	BEAD
Pre-producción	BEAT
Producción	BEAP

Interfaz con OSB

La comunicación entre el OSB y la aplicación se realiza mediante servicios web.

Para el envío de mensajes, la aplicación MetroSMS llama al servicio MetroSMS_EnvioSMS_Generico_PS definido en el OSB y para la recepción de mensajes y notificaciones MetroSMS expone el WS metrosms-webservices/RecepcionNotificacionImplService.

En el punto 11.2.3 se inventarian las URLs donde MetroSMS y el BUS deben efectuar sus llamadas.

13.2. Implantación

En este apartado se detallan las tareas necesarias para la implantación del sistema, desde la instalación, cuando proceda, configuraciones específicas para adaptar el sistema al comportamiento requerido, detalle de las tareas necesarias para la implantación y acciones necesarias para la verificación del correcto funcionamiento.

13.2.1. Instalación

Pre-condiciones:

- Instalado y configurado un BEA WLS 10.X.
- Instalado y configurado OSB 10.3.X.
- Instalado y configurado servidor Apache.
- Instalado y configurado Balanceadores SSL.
- Haber creado una instancia o server sobre el WLS.
- Tener creada la instancia de base de datos.

Componentes a instalar y/o configurar:

- Configuración DataSource de acceso a Base de Datos (Importante no haber marcado la opción de “Transacciones distribuidas”).
- El instalable del componente principal es el archivo “metroSMS.ear” que se despliega automáticamente(en preproducción) mediante la ejecución del job de Hudson METROSMS-DEPLOY-PRE
- Configuración propia de la aplicación (variables de configuración definidas en BBDD).
- Configuración de mensajería JMS. Se deben crear y configurar correctamente 2 colas de mensajería JMS.
- Configuración OSB.
 1. Despliegue del webservices que va a permitir que MetroSMS solicite el envío de mensajes.
 2. Despliegue de los webservices definidos por el bróker y la transformación de la información recibida desde el bróker a través de esos webservices hacia la aplicación METROSMS.
 3. Despliegue del módulo php que disparan la solicitud al bróker de los mensajes entrantes y las notificaciones pendientes.
- Configuración del servidor apache: la llamada del bróker a Metro para la entrega de mensajes MO entrantes y notificaciones se realiza a través de los apaches de la DMZ en los que se ha configurado una regla para filtrar la IP del bróker de tal forma que sólo se puedan atravesar los apaches viniendo de dicha dirección. Para ello es necesario modificar el fichero httpd.conf:

```
#### METROSMS ####  
Include servicioexternos/conf/metrosmsdes.conf  
#### FIN METROSMS ####
```

y definir el fichero metrosms.conf.

13.2.2. Shared Libraries

Wls- commonslogging- bridge- war (1.0,1.0)	Librería WL	Librerías necesarias para el correcto funcionamiento de la aplicación
---	-------------	--

13.2.3. URL de la aplicación

Herramienta	Entorno	Valor/es
WebService METROSMS	Desarrollo	
	Pre-producción	
	Producción	
WebService OSB	Desarrollo	
	Pre-producción	
	Producción	
WebService OSB	Desarrollo	
	Pre-producción	
	Producción	
WebService METROSMS	Desarrollo	
	Pre-producción	
	Producción	
WebService OSB	Desarrollo	
	Pre-producción	
	Producción	

13.2.4. Verificación de la instalación/configuración

- Comprobar que la aplicación está correctamente levantada en la consola de administración del BEA
- Comprobación DataSource. Comprobar, mediante la consola del WLS que existe comunicación entre el DataSource y la bbdd
- Comprobar que el OSB está correctamente configurado.
- Comprobar que las colas JMS están correctamente configuradas.
- Verificar que está habilitada la URL hacia el exterior y presenta correctamente el certificado además de permitir el acceso exclusivamente a las direcciones IP que se hayan definido.

13.3. ADMINISTRACIÓN

En este apartado se recogen las labores de administración necesarias para garantizar el correcto funcionamiento del sistema/aplicación, desde el arranque y la parada, cuando proceda, a los procedimientos de monitorización del sistema/aplicación, así como las tareas periódicas que sea necesario ejecutar.

13.3.1. Arranque y parada

Se arranca desde la consola de weblogic como cualquier otra aplicación.

13.3.2. Logs y monitorización

La localización de dichos logs es la siguiente:

- Logs del componente JEE:
 1. /bea/logs/aplicaciones/metroSMS/metroSMS_ServerN.log (Hay un log por cada servidor en el que esté desplegada la aplicación)
- Monitorización. Para el correcto funcionamiento del módulo de mensajería es necesario que el OSB funcione correctamente (y su correspondiente comunicación desde el exterior a metro). Por tanto, la monitorización que se esté llevando a cabo actualmente sobre OSB es necesaria. También se debe monitorizar los logs de las aplicaciones y los códigos de error según se describe en el apartado 11.4.

13.4. RESOLUCIÓN DE PROBLEMAS

En este punto se pretende recoger información sobre posibles errores o problemas que puedan surgir durante el funcionamiento normal del sistema/aplicación que sean predecibles, así como las comprobaciones que se puedan hacer y su posible solución.

Inventario de errores, chequeos y soluciones

- Comprobar conectividades con Sistemas Externos:
 1. Base de datos de aplicación.
 2. Apache:
 - Verificar URL. Ver apartado 11.2.3 “¡Error! No se encuentra el origen de la referencia.”
 - Comprobación de la definición de las reglas de entrada al proxy desde el exterior.
- Errores devueltos por el componente core JEE:
 1. [METROSMS-001] Problemas al intentar acceder a la base de datos. Es necesario revisar que la base de datos está accesible.
`"[METROSMS-001] Problemas de acceso a BBDD"`
 2. [METROSMS-002] El operador por el que hemos recibido el mensaje no está registrado en nuestra base de datos.
`"[METROSMS-002] No se ha encontrado el nombre del operador (findByNombre) "`
 3. [METROSMS -003] Problemas al intentar recuperar valores de configuración de la BBDD (tabla T_CONFIGURACION_ADMINISTRACION). Es necesario revisar el contenido de dicha tabla e incluir la clave que falta.
`"[METROSMS-003] No se ha encontrado la clave de la aplicación <clave>"`
 4. [METROSMS -004] El operador desde el que se nos ha mandado el mensaje no está registrado en nuestra bbdd

- "[METROSMS-004] Operador no existe en el sistema"
5. [METROSMS -005] Error al intentar enviar los datos a aplicación, bien porque esté caída o por cualquier otro motivo.
- "[METROSMS-005] Error en la conexión con APLICACION."
6. [METROSMS -006] Problemas cuando intentamos crear la conexión JMS. Revisar si la cola JMS está creada y configurada correctamente.
- "[METROSMS-006] Problemas al crear la conexión JMS (<nombre Factoria>/<nombre cola>)"
7. [METROSMS -007] Problemas al intentar colocar el mensaje o la notificación en su respectiva JMS (cola de mensajes o cola de notificaciones)
- "[METROSMS-007] Problemas al introducir el mensaje JMS en la cola de..."
8. [METROSMS -008] Problemas al intentar cerrar la sesión JMS. Revisar el estado de las colas JMS en el administrador.
- "[METROSMS-008] Error cerrando la sesión JMS (<nombre Factoria>/<nombre cola>)"
9. [METROSMS -009] Problemas al intentar cerrar la conexión JMS. Revisar el estado de las colas JMS en el administrador.
- "[METROSMS-009] Error al cerrar la conexión JMS (<nombre Factoria>/<nombre cola>)"
10. [METROSMS -010] La composición de la query que hay que lanzar a la base de datos en base a las capacidades de envío solicitadas por el usuario tiene errores.
- "[METROSMS-010] La consulta SQL para encontrar un operador con dichas capacidades ha fallado.
(findOperadorPorCapacidad)"
11. [METROSMS -011] Cuando la aplicación intenta establecer la conexión con el OSB se produce algún tipo de error. Revisar la que la URL del OSB sea la correcta y que funcione de manera adecuada.
- "[METROSMS-011] ERROR: El mensaje NO ha sido enviado."
12. [METROSMS -013] Cuando la aplicación intenta establecer la conexión con el OSB se produce algún tipo de error. Revisar la que la URL del OSB sea la correcta y que funcione de manera adecuada.
- "[METROSMS-013] Error en la conexión con el BUS. "
13. [METROSMS -014] Algunos de los parámetros del mensaje tales como el texto, el remitente, la aplicación.... son incorrectos.
- "[METROSMS-014] Los datos introducidos no son correctos, por favor reviselos"
14. [METROSMS -015] No existe ninguna aplicación a la que vaya dirigido ese mensaje.
- "[METROSMS-015] No se ha encontrado el nombre de la aplicación"
15. [METROSMS -016] La URL que se nos ha proporcionado del bus o no funciona o no tiene ningún valor en la bbdd.
- "[METROSMS-016] No se ha encontrado la URL en BD. Revise

configuración"

16. [METROSMS -017] Error en la llamada al método del webService del bus. MetroSMS no está pudiendo mandar un mensaje saliente al bus.

"[METROSMS-017] Error a llamar al método enviarSMS del bus"

17. [METROSMS -018] Error en la llamada al método de recibir SMS del webService de la aplicación. MetroSMS no está pudiendo mandar un mensaje entrante a la aplicación.

"[METROSMS-018] Error a llamar al método recibirSMS de la aplicación"

18. [METROSMS -019] Error en la llamada al método de recibir notificaciones del webService de la aplicación. MetroSMS no está pudiendo enviar una notificación de cambio de estado del mensaje previamente enviado a la aplicación.

"[METROSMS-019] Error a llamar al método recibirNotificacionSMS de la aplicación"

19. [METROSMS -020] Ocurrió algún tipo de error inesperado durante el procesamiento del mensaje que no está contemplado en la aplicación.

"[METROSMS-020] Excepción general durante el procesamiento"

20. [METROSMS -021] Se produce algún error al intentar conectar con la aplicación y enviarla bien una notificación y o bien un mensaje entrante.

"[METROSMS-021] Error enviando mensaje entrante/notificación a la aplicación."

- **Errores devueltos por el componente php:**

1. [METROSMS -023] El servicio del BUS ha tenido algún problema durante la ejecución.

"[METROSMS-023] ERROR: Ejecución con errores"(Llamando servicio de notificaciones)

2. [METROSMS-024] La llamada al BUS no se ha podido realizar o ha saltado el tiempo máximo de espera del cliente.

"[METROSMS-024] ERROR:"