

Integración de telefonía móvil (sms) y servicios http para la implementación del sistema de información pesquera, oleajes y mareas del Instituto del Mar del Perú (IMARPE)

Integration of mobile (sms) and http services for the implementation of fisheries information system, waves and tides of the Instituto del Mar del Peru (IMARPE)

Carlos Carreño Villarreyes¹

Resumen

El presente, tiene como propósito explicar de manera detallada los componentes del sistema de información pesquera implementando en IMARPE (Instituto del Mar del Perú) para dispositivos móviles utilizando el servicio de mensajes cortos (SMS).

Palabras claves

Sms, Smpp, Sistemas de información pesquera, Mensajes cortos

Abstract

This aims to explain in detail the components of fisheries information system implemented in IMARPE (Instituto del Mar del Perú) for mobile devices using short message service (SMS).

Key words

Process, process thread, process thread manager, learning, meaningful learning.

Introducción

El Instituto del mar del Perú (Imarpe) es una entidad de gobierno de carácter técnico y científico, esta adscrito al Ministerio de la Producción. El Imarpe tiene como principal función el estudio científico del mar y sus recursos para asesorar en el uso racional de los recursos naturales del mar del Perú [1].

El Imarpe investiga la relación entre los recursos pesqueros, el ambiente y la explotación de los recursos pesqueros por la actividad de la pesca artesanal y pesca industrial y asesora en la explotación de manera responsable de los recursos pesqueros [1].

La pesca artesanal es una actividad extractiva que aprovecha el conocimiento de la bio diversidad y los ecosistemas existentes para su sustento. El Imarpe para el desarrollo de la pesca artesanal provee de información clave relativa a las mejores zonas de pesca, situación del ambiente y mejores métodos y tecnologías de pesca.

La información veraz y oportuna que Imarpe brinde a los pescadores, permitirá a estos establecerse formalmente como micro empresarios y desarrollar el rubro. En este sentido Imarpe implementa soluciones creativas que aplican tecnologías de información y comunicaciones para aprovechar la infraestructura de telefonía móvil existente y en creciente adopción para difundir la información relativa a precios de especies, oleajes y mareas en los distintos puertos del país, denominado Sistema de precios, oleajes y mareas (SPOM).

En la implementación del SPOM se ha utilizado el protocolo SMPP y tecnologías Java EE para la programación de componentes del sistema.

Protocolo de comunicaciones Smpp

SMPP, que corresponde las siglas en inglés de short message peer-to-peer protocol, es un protocolo estándar de telecomunicaciones pensado para el intercambio de mensajes SMS entre equipos que gestionan los mensajes como pueden ser los SMSC (Short message service center) o los GSM USSD (Unstructured Supplementary Services Data Server), y un sistema de solicitud de SMS como puede ser un servidor WAP o cualquier gateway de mensajería, incluso implementados por software.

SMPP fue desarrollado por Aldiscon, una pequeña firma irlandesa comprada posteriormente por Logica. En 1999, SMPP pasó formalmente a manos del 'SMPP Developers Forum', posteriormente rebautizado como el SMS Forum, el cual es un consorcio [3].

SMPP define básicamente:

- El conjunto de operaciones para el intercambio de mensajes SMS entre los ESME y el SMSC.
- Los datos que los ESME deben intercambiar con el SMSC durante la conexión.

El protocolo se basa en el intercambio, petición/respuesta, de pares de PDUs (protocol data units), estos se intercambian sobre la capa 4 OSI (sesiones TCP/IP o X.25). El intercambio de datos puede realizarse de manera síncrona, esperando cada parte la respuesta/petición del otro para enviar la correspondiente petición/respuesta, o asíncrona, donde cada el envío y la recepción van a través de distintos hilos.

Actualmente las versiones más utilizadas, pues son las más comúnmente soportadas por los operadores, son por orden, SMPP v3.3 y v3.4. Esta última soporta el modo tranceiver (una misma conexión puede enviar y recibir al 'mismo' tiempo), en el sistema SPOM se ha utilizado la versión 3.3.

SMPP, es un protocolo abierto, diseñado para proporcionar un interfaz de comunicación para la transferencia de mensajes cortos (SM) entre External Short Message Entities (ESMES), Routing Entities (RE), y Message Centres (MC).

- *Message Centre (MC)* o Centro de Mensajes es un término genérico utilizado para describir sistemas tales como un Short Message Service Centre (SMSC), GSM Unstructured Supplementary Services Data (USSD) o Cell Broadcast Centre (CBC).
- *External Short Message Entity (ESME)* típicamente representa un cliente SMS de una red fija/una entidad de red, tal como un Server Proxy WAP, un Email Gateway, un Servidor de Mensajería de voz (Voice Mail Server). También puede representar un Cell Broadcast Entity (CBE).
- *Routing Entity (RE)* o Entidad de ruteo es un término genérico para un elemento de red que es utilizado por un MC a otro MC y un ESME a un MC para ruteo de mensajes. Un SMPP RE tiene la capacidad de emular la funcionalidad de asociación tanto de un MC como de un ESME o de ambos a la vez. Para un ESME, un RE aparece como un MC y para un MC, un RE aparece como un ESME. Un carrier o proveedor de acceso a la red celular o Internet puede utilizar RE's para ocultar una red de MC's, presentando únicamente las RE's como el punto de interfaz externo para las ESME's.

Aplicaciones típicas de Smpp

Las aplicaciones son variadas del protocolo Smpp sin embargo destacan las siguientes (ver Figura 1 Red de telefonía celular y el protocolo Smpp) [7]:

- Alertas de voicemail originadas desde un VPS (Voice Processing System) indicando mensajes de voz al mailbox del cliente, son una de las primeras aplicaciones SMS basadas en ESME que todavía son utilizadas en la industria.
- Servicios de información. Por ejemplo, permitir a los suscriptores móviles consultar tasas de cambio o información de precios de acciones de una base de datos o en la WWW, y que estos reciban esa información desplegada en su móvil como un SM (short message).
- Se utiliza para llamadas directamente marcadas o desviadas hacia un operador, quien a su vez direcciona o remite los mensajes hacia una MC para entregarlas al terminal o celular del suscriptor

- Servicios de Directorio
- Servicios basados en Posicionamiento o localización. Incluyen rastreo de vehículos, envío de taxis, controle logístico.
- Aplicaciones de seguridad tales como sistemas de alarmas que pueden utilizar los servicios SMS para accesos remotos y propósitos de alertas. Por ejemplo, un padre recibe un SMS de su compañía de seguridad para informarles que su hija ha llegado a casa y que ingreso el código de seguridad para acceso.
- Banco en línea, E-Commerce.
- SMS – Chat, juegos. Los usuarios de móviles pueden interactuar unos con otros utilizando un Server ESME y utilizar esta interacción para jugar inalámbricamente.
- Instant Messaging.
- Cell Broadcasting Services (Servicios de Difusión). Apoyo de mensajería geográfica como alarmas de tráfico y servicios de emergencia.

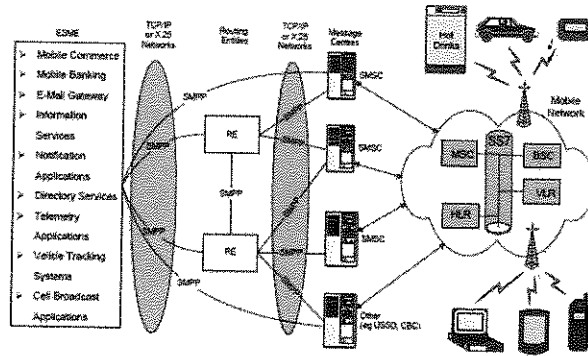


Figura 1. Red de telefonía celular y el protocolo Smpp

Java Edición Empresarial (Java EE)

Java es un lenguaje de programación y una plataforma, el lenguaje de programación es un lenguaje orientado al objeto que tiene su sintaxis y estilo particular. La plataforma Java es un entorno conformado por servicios base, APIs, librerías en el cual las aplicaciones escritas en lenguaje Java pueden ejecutarse [2]. Existen varias plataformas Java sin embargo la plataforma Java EE es la de mas alto nivel esta diseñada para soportar aplicaciones escritas en Java que requieren se escalables, seguras, confiables, usar multihilos y servicios de red.

IMPLEMENTACION

Software de base y de capa media

Como software base y capa media se utilizaron productos open source que están disponibles de manera gratuita a través de internet, esto permitió que la solución no tuviera un costo para Imarpe, en cuanto al software base utilizado. (Ver Figura 2.- Software base y de capa media)

Sistema Operativo CentOS 5.8 64 bits

CentOS (Community ENTERprise Operating System) es una bifurcación a nivel binario de la distribución Linux Red Hat Enterprise Linux RHEL, compilado por voluntarios a partir del código fuente liberado por Red Hat.

Red Hat Enterprise Linux se compone de software libre y código abierto, pero se publica en formato binario usable (CD-ROM o DVD-ROM) solamente a suscriptores pagados. Como es requerido, Red Hat libera todo el código fuente del producto de forma pública bajo los términos de la Licencia pública general de GNU y otras licencias. Los desarrolladores de CentOS usan ese código fuente para crear un producto final que es muy similar al Red Hat Enterprise Linux y está libremente disponible para ser bajado y usado por el público, pero no es mantenido ni asistido por Red Hat. Existen otras distribuciones también derivadas de las fuentes de Red Hat [4].

JVM HotSpot 64 Bits

Una Máquina virtual Java (en inglés Java Virtual Machine, JVM) es un máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java [2].

El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable incluso como lenguaje de entrada para un microprocesador físico. Como todas las piezas del rompecabezas Java, fue desarrollado originalmente por Sun Microsystems.

La gran ventaja de la máquina virtual java es aportar portabilidad al lenguaje de manera que desde Sun Microsystems se han creado diferentes máquinas virtuales java para diferentes arquitecturas y así un programa .class escrito en un Windows puede ser interpretado en un entorno Linux. Tan solo es necesario disponer de dicha máquina virtual para dichos entornos. De ahí el famoso axioma que sigue a Java, "escribelo una vez, ejecútalo en cualquier parte", o "Write once, run anywhere".

Jboss Application Server 6.0 GA

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible Java. Los principales desarrolladores trabajan para una empresa de servicios, JBoss Inc., adquirida por Red Hat en abril del 2006, fundada por Marc Fleury, el creador de la primera versión de JBoss. El proyecto está apoyado por una red mundial de colaboradores. Los ingresos de la empresa están basados en un modelo de negocio de servicios [5].

JBoss implementa todo el paquete de servicios de J2EE.

JBoss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción de JEE, disponible en el mercado en versión Enterprise y comunitaria (open source), ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business. Combinando una arquitectura

orientada a servicios revolucionaria con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, embebido y distribuido sin restricciones por licenciamiento. Por este motivo es la plataforma más popular de middleware para desarrolladores, vendedores independientes de software y, también, para grandes empresas.

Las características destacadas de JBoss incluyen:

- Producto de licencia de código abierto sin costo adicional.
- Cumple los estándares.
- Confiable a nivel de empresa
- Imbebible, orientado a arquitectura de servicios.
- Flexibilidad consistente
- Servicios del middleware para cualquier objeto de Java
- Ayuda profesional 24x7 de la fuente, solo en versiones Enterprise
- Soporte completo para JMX

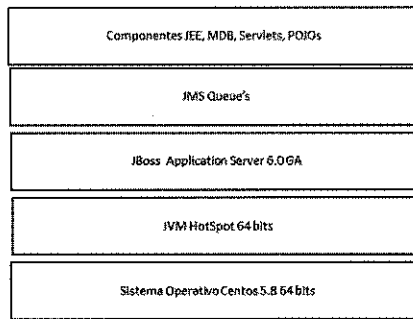


Figura 2. Software base y de capa media

Arquitectura de la solución

El proveedor del servicio de acceso a internet y de telefonía celular es movistar. La implementación de SPOM se ha realizado con la arquitectura conformada por los siguientes componentes:

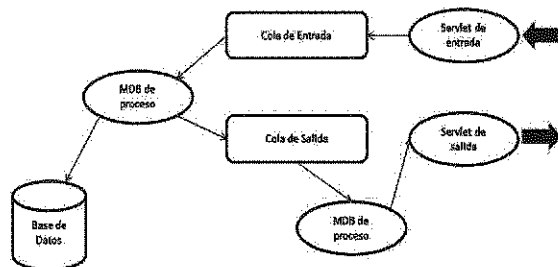


Figura 3. Arquitectura de aplicación

Cola de entrada y salida

Una cola de mensajes es un lugar común donde se reciben y se publican mensajes, que son consumidos por otros programas. Existen así cuatro elementos principales en un sistema de mensajería [6]:

- **Publicador**, objeto que publica un mensaje en una cola
- **Consumidor**, objeto que consume mensajes de una cola
- **Mensaje**, es un objeto que representa un mensaje entre objetos, este tiene algún formato que tanto publicador como consumidor conocen.
- **Cola**, es el mecanismo de intermediación de publicadores y consumidores permite que se conecten y comuniquen a través de mensajes.

Es conveniente destacar que la solución implementa un sistema asíncrono: el publicador y el consumidor no necesitan estar disponibles a la vez, ya que la cola actúa de intermediario, guardando y distribuyendo los mensajes a medida que el consumidor pueda atenderlos.

MDB de proceso

Los message-driven beans o MDBs son componentes en EJB 3.0 que pueden recibir y consumir mensajes asíncronamente. Un message-driven bean es invocado por el container como resultado de la recepción de un mensaje enviado por un cliente utilizando Java Message Service (JMS) [6].

Un cliente no ejecuta directamente un message-driven bean, si no que sólo debe utilizar la API de JMS para enviar mensajes. Por esto, un message-driven bean no tiene una clase home ni interfaz local o remota ni retorna valores o excepciones al cliente. El cliente no espera que su mensaje sea respondido si no que continúan su ejecución una vez enviado.

Los message-driven beans sólo reciben mensajes JMS, sin conocer de antemano la información sobre contenido del mensaje recibido. Por esta razón sólo tienen un método con lógica de negocio llamado `onMessage()`, que recibe un Message JMS que puede representar todos los tipos de mensajes existentes en JMS como mensajes de bytes, de texto y de objetos serializables. Luego hay que discriminar el tipo de mensaje recibido utilizando el operador `instanceOf`.

Los message-driven beans son stateless ya que no mantienen estados de conversación entre cada procesamiento de mensajes recibidos, por lo cual las instancias de la misma clase son equivalentes entre sí.

Servlet de entrada y salida

Un servlet es un objeto que se ejecuta en un servidor o contenedor JEE, especialmente diseñado para ofrecer contenido dinámico desde un servidor web, generalmente HTML. Un servlet implementa la interfaz `javax.servlet.Servlet` o hereda alguna de las clases más convenientes para un protocolo específico (ejemplo: `javax.servlet.HttpServlet`). Al implementar esta interfaz el servlet es capaz de

interpretar los objetos de tipo `HttpServletRequest` y `HttpServletResponse` quienes contienen la información de la página que invocó al servlet. En la aplicación se utilizan dos servlets uno para Recepcionar los mensajes desde movistar y el otro para enviar mensajes hacia movistar [8].

RESULTADOS Y DISCUSIÓN

La solución como resultado implementa tres tipos de servicios:

- Alertas de oleajes
- Alerta de mareas
- Alerta de precios de especies

En distintos lugares.

Alerta de oleajes

La alerta de oleajes es un servicio informativo que permite enviar un mensaje al 9009 con dos datos de entrada, la palabra "oleaj" y el nemónico del lugar. La lista de nemónicos de los lugares esta disponible en el siguiente URL <http://190.81.184.115:7001/ReportesArtesanal/> esta se puede descargar e imprimir. Ejemplo: oleajlima.

Se puede solicitar de momento el estado del oleaje en los siguientes lugares:

Nemónico	Nombre del lugar
TUMBES	TUMBES
PAITA	PAITA
BAYOVA	BAYOVAR
CHICLA	CHICLAYO
TRUJIL	TRUJILLO
CHIMBO	CHIMBOTE
HUARME	HUARMEY
HUACHO	HUACHO
LIMA	LIMA
CHINCH	CHINCHA
PISCO	PISCO
ICA	ICA
AREQUI	AREQUIPA
MOLLEN	MOLLENDO
ILO	ILO

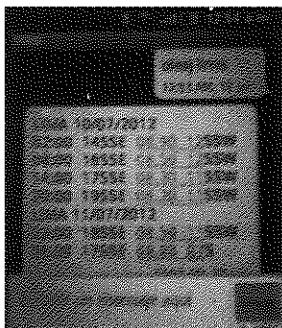


Figura 5. Respuesta de servicio de oleajes

El servicio de oleaje responderá con un mensaje en formato tabular en una estructura formada por los siguientes campos:

1. Hora
2. Velocidad del viento(expresado en nudos)
3. Dirección del viento
4. Altura máxima del oleaje (expresado en pies)
5. Intensidad de la ola
6. Dirección de la ola

Alerta de mareas

El servicio de oleajes proporciona los datos del estado de la marea en un lugar en particular, enviamos un Sms al 9009 con la palabra "marea" acompañada del nemónico del lugar.

La respuesta del servicio de mareas esta compuesta por una estructura con los siguientes campos:

1. Fecha
2. Hora
3. Altura de la marea (pies)

Alerta de precios

Este servicio permite conocer el precio de una especie en algún lugar donde se ha pescado la especie. Para usar el servicio se debe enviar al 9009 el nemónico de la especie acompañado del nemónico del lugar. Ejemplo: merl1 talara. El servicio enviara como respuesta la fecha y los precios de la especie y tamaño.

CONCLUSIONES

Como principales conclusiones tenemos:

- La experiencia ganada durante al finalizar la implementación de este proyecto es muy enriquecedora, toda vez que se interactuó con varios roles para la configuración de cada componente clave de la solución, así se coordinó con el proveedor de comunicaciones, para este caso Movistar (Telefonica) para habilitar la conectividad entre el MC (Movistar) y el ESME (Imarpe) de esta manera ambas entidades lograron intercambiar mensajes SMPP
- Se introdujo una solución basada en componentes empresariales que permitirá lograr niveles de escalabilidad necesarios para atender el crecimiento de la demanda en la medida que los servicios de información que ofrecerá la plataforma..
- El carácter asincrónico de la solución garantiza que la recepción de las consultas y respuesta se realicen sin intervención humana, es decir la solución tiene un carácter autónomo en cuanto a su funcionamiento.

- Las entidades de gobierno están siempre en la búsqueda de soluciones eficientes con el presupuesto mas bajo posible, pues al tratarse de una entidad dedicada a la aplicación de métodos científicos para la investigación, el presupuesto asignado a tales tareas es bastante reducido, la solución esta basada en todos los aspectos en software open source, esto a permitido que el costo del software base y de capa media sea cero, ahorrándose Imarpe de esta manera un costo importante. El open source permite entonces acceder a tecnología de punta a muy bajo costo, sin embargo conviene aclarar que si bien por un lado el costo es cero del software de tipo open source es necesario invertir en servicios pues se requiere de personal profesional que conozca las herramientas de software necesarias para una segura y exitosa implementación.
- El trabajo en equipo es importante, en todo momento se conto con el apoyo del área de sistemas y de la persona funcional responsable sobre todo para desatar los nudos administrativos que comúnmente se generan en este tipo de proyectos. La colaboración inmediata de los expertos del dominio y el personal de planta del área de sistema fue clave esto permitió acortar el tiempo de implementación.
- El ciclo de vida de la solución aun no esta determinado, queda una tarea pendiente. En este momento no podemos asegurar que la tecnología SMS prevalezca a otras, quizá en un futuro cercano sea remplazada por otra, para entonces será necesario alguna migración sin embargo en este momento SMS y SMPP resultan tecnologías muy conocidas, baratas y de fácil acceso a la población dedicada a la pesca artesanal que no cuenta en principio con abundantes recursos económicos.
- Como mejora de la solución reflexionamos sobre la necesidad de contar con una herramienta que genere una vista con indicadores de uso del servicio actualmente se cuenta con reportes básicos pero tal vez un cuadro de mando del servicio sea la mejor solución para mostrar los indicadores a los tomadores de decisiones.

REFERENCIAS BIBLIOGRÁFICAS

Sitio Web de Imarpe. <http://www.imarpe.gob.pe>

ORACLE CORPORATION. "Java First Cup", 2012, Oracle Press

SMPP DEVELOPERS FORUM. "Short Message Peer to Peer Protocol Specification v3.4", 1999

CENTOS. <http://www.centos.org/>

JBOSS COMMUNITY. <http://www.jboss.org/>

ANDY TAYLOR. "JBoss Enterprise Application Platform 5 HornetQ User Guide", 2010, Red Hat Inc.

SMPP FORUM. <http://http://smsforum.net/>

CHRIS PELTZ. "ten best practices for J2EE development", 2001, Hewlett Packard, Co.