

## DESARROLLO DE UNA PLATAFORMA MULTIDISPOSITIVO DE GESTIÓN DE EVENTOS

*Ing. Arturo Cervera García*

*Investigador Instituto de Robótica (Universidad de Valencia, España)*

*Arturo. Cervera@robotica.uv.es*

*Dr. José Javier Samper Zapater*

*Miembro de la ETSE e Investigador en el Instituto de Robótica*

*Profesor de la Universidad de Valencia*

*jose.j.samper@uv.es*

*<http://robotica.uv.es/~jsamper/>*

*Dr. Eduardo Carrillo Zambrano*

*Profesor Titular Facultad de Ingeniería de Sistemas, Escuela de Ciencias Naturales e Ingeniería, Universidad Autónoma de Bucaramanga*

*[eduleidy@eudoramail.com](mailto:eduleidy@eudoramail.com)*

## DESARROLLO DE UNA PLATAFORMA MULTIDISPOSITIVO DE GESTIÓN DE EVENTOS

Resumen en Español

El presente artículo trata de abordar la problemática existente en la gestión del envío de información a dispositivos con características y necesidades de diversa naturaleza. Inicialmente se plantea una propuesta general de diseño para una plataforma de información mediante eventos donde la flexibilidad y la escalabilidad son los requisitos principales. En este punto se propone una arquitectura que es independiente de la fuente de información utilizada, así como también de los tipos de dispositivos de usuario a los que va dirigido el sistema. Posteriormente se describe la implementación realizada de esta plataforma para crear un servicio de alertas de tráfico capaz de enviar notificaciones mediante correo electrónico y llamadas telefónicas.

**Palabras clave:** Sistemas multimedia, integración de voz y datos, sistema de alertas, gestión de eventos

## DEVELOPMENT OF A MULTIDEVICE PLATFORM FOR EVENTS MANAGEMENT

Abstract

The present article deals with the problems existing in managing traffic delivery of information to devices that have different characteristics and needs. Initially, there is a general proposal for the design of an information platform with events. Flexibility and scalability are the main requirements of the platform. An architecture that is independent from the information source being used has been first suggested as well as an architecture of the user's devices types the system is being addressed at. Next, the implementation of the platform has been described that is intended to create a traffic alert service to send email notifications and phone calls.

**Keywords:** Integrated voice-data communication, Management information systems, Multimedia systems, Events management, Alert System

## INTRODUCCIÓN

### *Comunicación inalámbrica*

La diversidad de características es una de las principales particularidades presentes en los dispositivos de comunicación en la actualidad. La información que ofrecen al usuario puede acomodarse en múltiples formatos y puede ser enviada a través de sistemas de comunicación heterogéneos. Todo ello, hace que el diseño de un sistema deba plantear una arquitectura lo suficientemente abierta que le permita llegar al mayor número de usuarios posible y ser escalable para asegurar un largo periodo de vida.

En los últimos años se han producido importantes avances en el campo de las tecnologías de comunicación que han hecho posible la aparición de nuevas formas de acceso a la información. El objetivo principal que se intenta alcanzar es hacer que la información sea accesible a cualquier persona, en cualquier sitio y con total disponibilidad. Esta necesidad de poder acceder a los contenidos Web bajo esta independencia ha dado origen a áreas como la computación ubicua (Pervasive Computing). A pesar de la alta tasa de crecimiento de Internet, la cual está estimada en torno al 100%, la mayor parte de los usuarios acceden mediante un ordenador personal conectado a la red mediante un cable, lo que implica una restricción en el acceso, que debe realizarse desde un lugar determinado. Debido a esta limitación, tras Internet, han aparecido una serie de tecnologías como los teléfonos móviles y agendas personales basados en la tecnología WAP (Wireless Application Protocol) e I-mode que permiten el acceso inalámbrico a la información, lo que hace posible la obtención de información desde cualquier lugar que disponga de cobertura, a costa de verse limitada la capacidad de representación gráfica y de procesamiento. El espectacular crecimiento del número de ventas de terminales móviles ha propiciado la expansión de servicios de información que utilizan los mensajes cortos SMS (Short Message Service) como medio de comunicación. El éxito de esta tecnología en todo el mundo ha sido posible gracias a dos factores. Por una parte, el bajo precio de los dispositivos que son capaces de soportar SMS y por otro lado, el bajo coste que tiene este tipo de mensajes, que es muy inferior al coste de una comunicación WAP a través de las redes GSM (Global System for Mobile communication, o también Group Special Mobile). Estos dos factores han hecho posible la popularización de servicios orientados a SMS como por ejemplo servicios de información bancaria, información de tráfico, votaciones en concursos y servicios de alertas por SMS.

### *El Mercado de la voz*

El mercado de servicios de voz ha experimentado un considerable aumento en los últimos años, apoyado por el crecimiento en las ventas de dispositivos móviles. La interacción mediante la voz con los sistemas informáticos permite conectar a los usuarios con bases de datos en cualquier lugar y en cualquier instante utilizando algo tan natural para el ser humano como el habla. Además, el teléfono es el dispositivo más extendido y utilizado, lo que asegura un mercado potencial de usuarios para las aplicaciones de voz mucho mayor que el de otros tipos de aplicaciones.

## DESARROLLO

### *Descripción general de la plataforma*

El propósito marcado con este trabajo es la creación de un sistema gestor de información mediante eventos, el cual utilice en una fase inicial la voz y el correo electrónico como medio de difusión, y además funcione sobre una plataforma escalable que permita la fácil ampliación a nuevas formas de envío, como SMS o WAP Push. Esto permitirá disponer de un sistema flexible, capaz de enviar información a los usuarios mediante diferentes medios, adaptándose a las preferencias de los mismos. También hace posible disponer de un sistema de información ampliable, el cual es capaz de acomodarse a nuevos dispositivos que puedan surgir.

Dado que el uso general del sistema es el envío de cualquier tipo de información de interés para el usuario en tiempo real, la primera de las condiciones es el acceso en todo instante a los eventos acaecidos.

El paso previo a la gestión de los eventos es la creación de los perfiles de usuario. Generalmente, el usuario debe indicar cuales son sus preferencias atendiendo a diferentes criterios establecidos por el tipo de servicio. Por lo tanto, se hace indispensable un repositorio donde se almacene esta información, así como un conjunto de herramientas lo suficientemente generales para permitir la creación de perfiles por parte de usuarios desde diferentes tipos de dispositivos, como navegadores Web, PDAs (Personal Digital Assistant), teléfonos móviles o teléfonos convencionales. Mediante las preferencias de los usuarios y la información en tiempo real disponible, el sistema debe discriminar aquellos eventos que sean de interés, realizando la oportuna clasificación para su posterior envío.

La notificación de los eventos a los usuarios interesados también puede realizarse atendiendo a múltiples esquemas. La política de envío de un sistema de mensajes cortos no tiene porque coincidir con la política de envío de un sistema de alertas por voz o por correo electrónico. Por lo tanto, esta tarea debe realizarse por módulos específicos para cada sistema de envío. Existirán diferentes repositorios de salida, a los cuales accederán los distintos tipos de sistemas de envío que gestionarán la información atendiendo a sus políticas concretas.

### **Arquitectura general**

Basándonos en las necesidades anteriores, se ha diseñado una plataforma donde existe dos subsistemas con funcionalidades diferentes:

1. Interfaz de suscripción de usuarios. La arquitectura de capas propuesta está basada en la gestión de los datos mediante servicios Web, fundamentados en el intercambio de mensajes SOAP (Simple Object Access Protocol). Esta base permite que sea posible definir aplicaciones cliente basadas en entornos diferentes, como Web, entornos Java, teléfonos móviles, PDAs o incluso teléfonos convencionales (utilizando la voz para la creación de perfiles). La tecnología SOAP permite que la implementación del cliente pueda realizarse en lenguajes como JSP (Java Server Pages), PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) u otros, según las necesidades, y que esta implementación sea independiente del servidor.

2. Un sistema gestor de eventos (en adelante SGE), cuya finalidad es detectar en tiempo real los eventos de interés, relacionándolos con los perfiles de usuario. Una vez filtrada la información, el módulo gestor clasifica el envío de eventos en diferentes repositorios, dependiendo del tipo de medio por el cual se desea enviar. Estos repositorios actúan como buffers intermedios a los que acceden los diferentes sistemas que se encargan de la gestión de la entrega de los eventos a los usuarios.

En la figura 1 podemos ver los diferentes elementos que componen el sistema. Vemos que el centro del sistema es el SGE, que accede a la información en tiempo real y a las preferencias de los usuarios, almacenadas en una base de datos. Después del procesado de la información, los eventos que deban ser entregados (enviados por el medio que corresponda) serán depositados en los buffers intermedios, donde encontramos un buffer diferente para cada tipo de envío. A estos buffers accederán diferentes subsistemas, encargados del envío por cada canal concreto. Respecto a la definición de perfiles de usuario, tal y como se ha comentado anteriormente, la arquitectura propone la utilización de servicios Web y SOAP por su flexibilidad y escalabilidad.

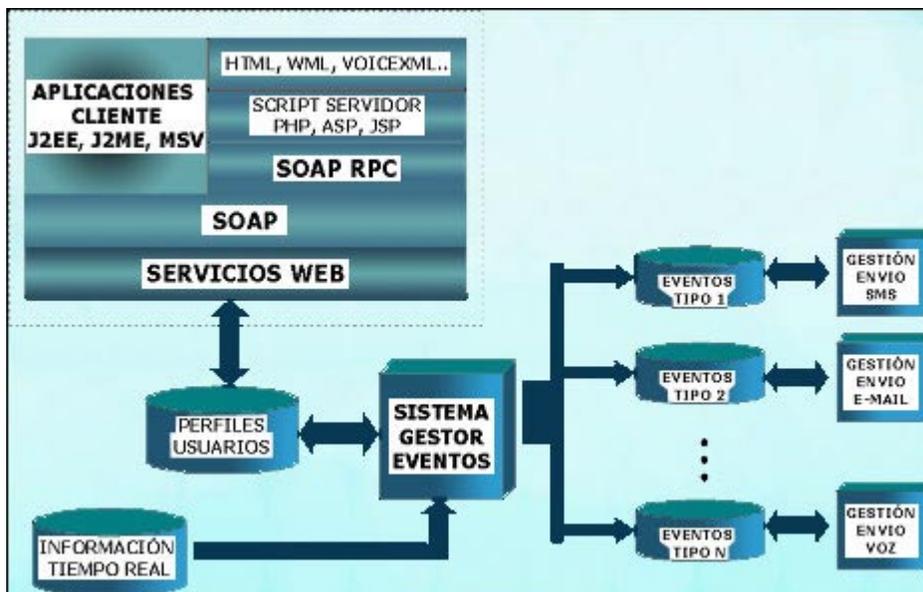


Fig. 1. Arquitectura general de la plataforma

## INFORMACIÓN DE INCIDENCIAS DE TRÁFICO VIAL

A continuación vamos a proceder a explicar cuales han sido las decisiones que se han tomado para implementar la parte del sistema gestor de eventos para el caso concreto de un servicio de alertas de incidencias de tráfico. El objetivo de este sistema es informar a los usuarios suscritos acerca de las incidencias ocurridas en las carreteras, como accidentes o retenciones de tráfico.

El origen de datos de la información en tiempo real es en este caso una página Web de información de incidencias de tráfico. El acceso se realiza mediante el protocolo HTTP (Hyper Text Transfer Protocol), lo que permite la independencia de localización del sistema. El único requisito es una conexión a Internet.

Internamente, el SGE se implementa como dos hilos de procesamiento que cooperan entre sí:

1. El hilo de volcado, el cual accede a un fichero XML del servidor Web de incidencias de tráfico, realiza una interpretación o parser del mismo para extraer los datos significativos y almacena esa información en una tabla temporal de su base de datos propia, actualizándola con una frecuencia configurable desde la aplicación. De esta forma, el sistema supone una mínima carga para el servidor Web de tráfico. El sistema debe tener la capacidad de informar no sólo de los nuevos eventos aparecidos, sino también de la desaparición de eventos. La forma de realizar esta gestión es utilizando un campo de marcas en la tabla, de forma que siempre que se lean incidencias y se actualice la tabla temporal, se marcará aquellas filas leídas. Las filas no marcadas en un ciclo de volcado serán filas eliminadas, por lo que se generará eventos de este tipo.

2. El hilo de consultas, cuya finalidad es relacionar los usuarios suscritos con la información de incidencias de la base de datos local. La salida de este proceso será la modificación de los diferentes repositorios de envío de eventos. Esta modificación puede implicar la inserción de nuevas alertas o la eliminación de alertas pendientes de envío que hayan quedado desfasadas. También es posible actualizar la información si esta ha cambiado y la alerta todavía no ha sido servida.

La figura 2 muestra la estructura utilizada en la implementación del SGE. Vemos que existen dos hilos, cuya comunicación se realiza mediante una base de datos temporal. El hilo de volcado se encarga de actualizar estos datos locales con las incidencias en tiempo real y de detectar la desaparición de incidencias. El hilo de consultas hace un emparejamiento de la información local con los perfiles de usuario, detectando qué usuarios están interesados en recibir información de los eventos activos. Como resultado, se modifican los datos de los diferentes buffers de salida.

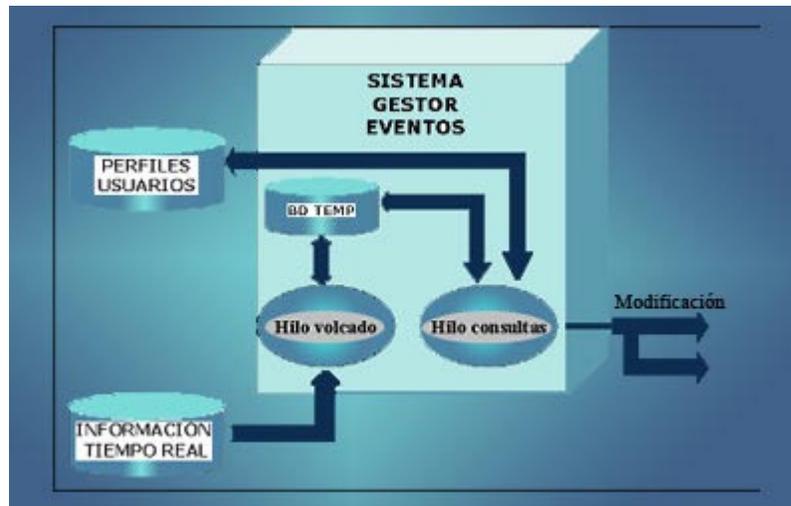


Figura 2: Estructura del sistema gestor de eventos

La figura 3 describe el algoritmo utilizado para implementar el hilo de volcado. Este algoritmo se basa en la existencia de una base de datos temporal donde se guarda información local de las incidencias de tráfico. Esta tabla sirve para, en cada iteración, comprobar la información que se acaba de leer con la que se leyó en iteraciones anteriores. De esta forma se comprueba qué eventos son nuevos, cuales ya estaban antes (y por lo tanto ya han sido procesados) y cuales han desaparecido, y por lo tanto puede ser necesario informar al usuario de su desaparición. El algoritmo consta de un bucle que se repite indefinidamente, donde tras un tiempo dormido, accede al origen de datos de incidencias en tiempo real y carga estas incidencias en una tabla auxiliar. Tras esto, bloquea el acceso para que ningún otro hilo opere con la listas del sistema y se puedan producir inconsistencias. Una vez ha conseguido el acceso, itera en un bucle donde, para cada incidencia de tráfico de la tabla auxiliar, busca en una tabla temporal si la incidencia ya había aparecido en instantes anteriores. Si ya la teníamos, le ponemos una marca. Si no la teníamos, la insertamos en nuestra base de datos temporal. Una vez finalizado el bucle, tendremos una serie de incidencias con marcas NUEVA y ACTIVA y otras con marca BORRADA. Esto nos permite posteriormente procesar las incidencias desaparecidas, si es necesario. Por ejemplo, podría ser necesario informar al usuario que una retención de tráfico que anteriormente le habíamos comunicado ya ha desaparecido y el tráfico es fluido.

## GESTIÓN DEL ENVÍO POR VOZ

La implementación realizada de la arquitectura propuesta incluye el envío de información mediante llamadas telefónicas y correo electrónico. Dado que la gestión del envío mediante correo electrónico es mucho más sencilla que la gestión telefónica, vamos a proceder a detallar la implementación de esta última. El módulo de gestión de llamadas está integrado por los siguientes elementos:

1. Un repositorio en el que el SGE deposita los eventos a enviar. Para cada evento, se almacena la información del número de teléfono destino y el cuerpo del mensaje a comunicar. Dado que estamos ante un sistema de eventos de incidencias de tráfico, la prioridad de cada evento se asigna clasificando las carreteras dependiendo de su importancia en base a la densidad de tráfico que soportan.

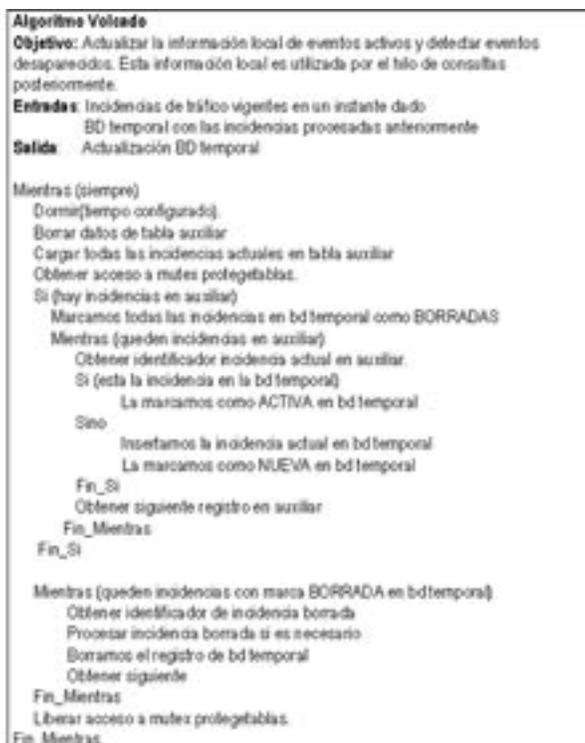


Fig. 3. Algoritmo del hilo de volcado

2. Un hilo de ejecución (hilo clasificador) que se encarga de acceder periódicamente al repositorio, leer la información y cargarla en una lista de nodos de usuario (ver Figura 4). Cada nodo de este tipo tiene un enlace a una lista de eventos pendientes de envío. Esta estructura de datos pretende minimizar el número de llamadas al usuario comunicándole toda la información disponible realizando una sola llamada. Los nodos 'usuario' tienen información de control que indica si el nodo está siendo servido por otro hilo, o si ya se han realizado varias llamadas y el usuario no ha descolgado. Los nodos de información siguen una ordenación por prioridades, ya que de esta forma, en momentos de sobrecarga del sistema, se servirá antes la información más importante, dejando el resto para instantes posteriores de menor carga del sistema.

3. Una serie de hilos de ejecución (hilos llamadores), uno por cada canal o puerto de envío que son los encargados de realizar las llamadas telefónicas.

4. Un módulo de síntesis de voz, que se encarga de convertir el texto a formato de audio, que será lo que se transmita por el canal telefónico al usuario

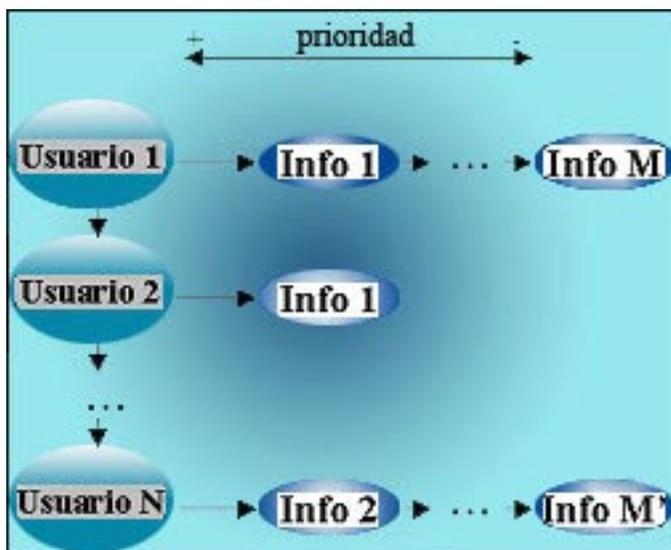


Figura 4: Estructura de datos de llamadas pendientes de envío

Esta estructura de tipo productor-consumidor, en la cual existe un proceso que inserta datos en la lista (hilo clasificador) y existen una serie de procesos que acceden a la lista para procesarlos (hilos llamadores) asegura que el reparto de carga entre los diferentes canales va a ser óptimo. Dado que la estructura de datos es compartida por varios hilos, es necesario acceder a la misma de forma exclusiva al realizar una escritura, bloqueando el acceso a través de mutex. Un mutex es un objeto software que permite que un determinado recurso pueda ser compartido por varios hilos de ejecución, ya que gestiona la concurrencia en el acceso. Las escrituras se realizarán cuando sea necesario añadir o eliminar nuevos usuarios, o cuando se desee cambiar el estado de un nodo de 'PendienteDeEnvío' a estado 'Sirviendo'. En el caso en que el usuario no descuelgue su terminal, se iniciará un temporizador en el nodo que mantendrá su estado a 'Dormido', después del cual volverá a estar 'PendienteDeEnvío' y podrá volver a ser atendido por el siguiente hilo llamador que acceda. Pasado un cierto número de intentos sin que el usuario descuelgue, se descartará la llamada, eliminándose el nodo correspondiente.

### AJUSTANDO PARÁMETROS DEL SISTEMA

Un sistema de voz supone un coste temporal elevado ocasionado por el tiempo empleado en realizar las comunicaciones telefónicas. El sistema debe establecer la conexión telefónica, comunicar el mensaje mediante síntesis de voz y desconectar la llamada. Se debe ejercer un control sobre el número de usuarios registrados, de forma que la calidad del servicio quede garantizada. Un número desproporcionado de usuarios puede originar que, ante un determinado evento, existan usuarios que tarden un tiempo excesivo en recibir la llamada informativa.

Para asegurar la calidad del servicio, definiremos varios conceptos. Llamaremos paso (P) a la unidad mínima de comunicación de nuestro sistema. Un paso corresponderá a un nodo de información asociado con un usuario. Definiremos  $T_p$  como la cota superior del tiempo que va a durar la comunicación de un paso,  $T_{es}$  como el tiempo máximo que va a esperar el sistema hasta el descolgado del terminal por parte del usuario y  $T_{max}$  como el tiempo máximo de desfase temporal entre el instante en que se tiene constancia de un evento y la comunicación al último de los usuarios suscritos a dicho evento.

Con estas definiciones, podemos deducir que  $T_{max}$  será la suma del tiempo dedicado a atender a cada usuario en cola ( $T_i$ ), dividido entre el número de canales de salida disponibles (N).

$$T_{max} = \frac{\sum_{i=1}^{U-1} T_i}{N} \quad (1)$$

Donde U es el número de usuarios en cola en un instante determinado.

Dado que el tiempo máximo dedicado a cada usuario  $T_i$  es el tiempo de espera hasta el descolgado,  $T_{es}$ , sumado al número de pasos realizados P por el tiempo máximo de comunicación de un paso  $T_p$ , podemos obtener (2).

$$\begin{aligned}
 T_{\max} &= \frac{\sum_{i=1}^{U-1} T_i}{N} \\
 &= \frac{\sum_{i=1}^{U-1} (T_{es} + P * T_p)}{N} = \frac{(U-1)(T_{es} + P * T_p)}{N} \quad (2)
 \end{aligned}$$

De la anterior fórmula, se puede concluir que el número máximo de usuarios en cola que garantice los plazos de entrega establecidos responderá a la ecuación (3).

$$U = \frac{T_{\max} * N + T_{es} + P * T_p}{T_{es} + P * T_p} \quad (3)$$

Aplicando la fórmula anterior con un valor de P igual a 1 obtendríamos el número máximo de usuarios que se pueden tener en la cola de espera. Hay que tener en cuenta que con P=1, sólo se dispone de un paso, por lo que solo se comunica una unidad de información, que generalmente corresponderá con 1 evento. En momentos de menor carga, es interesante aumentar P para comunicar más información en cada llamada.

El valor máximo de P en un instante dado lo podemos obtener mediante (4).

$$P = \left\lceil \frac{T_{\max} * N + T_{es} - U * T_{es}}{T_p (U - 1)} \right\rceil \quad (4)$$

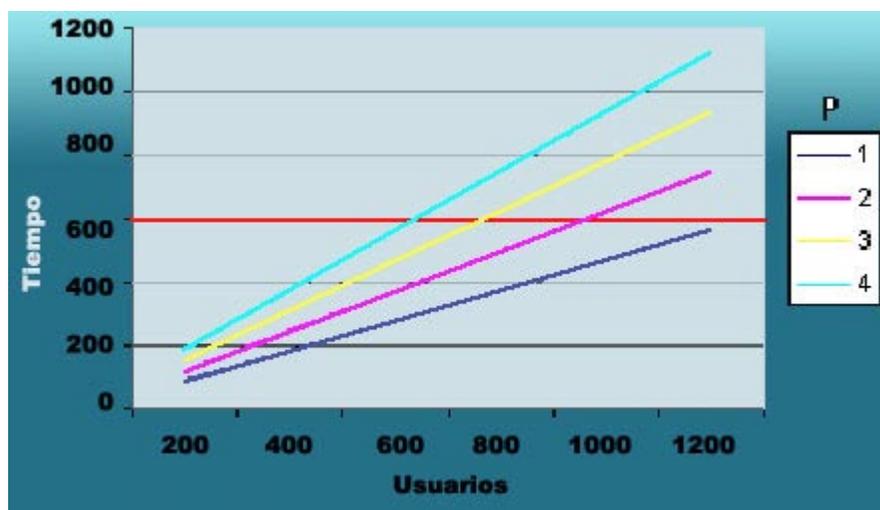
Hay que tener en cuenta que P tiene que ser un número entero de pasos y por lo tanto hay que aplicar la función suelo.

Considerando un canal o puerto de salida como cada uno de los medio físicos de envío de información (por ejemplo una línea telefónica en el caso de la voz) y teniendo en cuenta un sistema con 32 puertos de salida, establecemos T<sub>max</sub>= 600 seg, T<sub>p</sub>=5 seg y T<sub>es</sub>=10 seg. Aplicando la fórmula anterior, obtenemos un valor máximo de 1281 usuarios. Ese es el número máximo de usuarios que podemos tener en cola de espera. El número máximo de usuarios que nuestra aplicación aceptará podemos ajustarlo utilizando estadísticas de la distribución de los eventos en el tiempo. Sistemas con una distribución a priori homogénea de los eventos, aceptarán mayor número de usuarios suscritos que sistemas donde la función de distribución tenga un número alto de picos y valles.

En la figura 5 podemos ver, teniendo en cuenta las restricciones temporales del ejemplo anterior, cuales serían los tiempos máximo de desfase  $T_{max}$  dependiendo del número de usuarios. Tal y como se ha comentado anteriormente, vemos que si queremos mantener la cota máxima de espera de 10 minutos (600 seg.), podemos soportar hasta unos 1281 usuarios en la cola de espera, comunicando sólo una unidad de información o mensaje ( $P = 1$ ) cada vez (línea azul oscuro). Si quisiéramos comunicar 2 mensajes cada vez (línea rosa en el gráfico), podríamos tener un máximo de 961 usuarios en cola. El número de usuarios no es muy elevado, pero hay que tener en cuenta que la naturaleza del medio (comunicación de un mensaje de voz) hace que la transmisión de cada uno de los mensajes implique un consumo de un tiempo elevado.

Figura 5: Tiempo máximo de envío del total de eventos con 32 canales de salida

Dado un número de usuarios en cierto instante menor que la cota máxima admitida, el diseñador de la aplicación debe llegar a una solución de compromiso entre comunicar más información por llamada, aumentando el tiempo consumido en informar a todos los usuarios suscritos en el sistema (aunque siempre por debajo del umbral  $T_{max}$ ) o bien informar a todos los usuarios lo más rápidamente posible, a costa



de enviar primero la información importante y realizar posteriormente nuevas llamadas para comunicar el resto de importancia menor.

## GESTIÓN DEL ENVÍO POR CORREO ELECTRÓNICO

El sistema desarrollado también es capaz de enviar incidencias mediante correo electrónico, si el usuario así lo ha deseado en su perfil. Esta parte es menos interesante desde el punto de vista educativo ya que la complejidad es mucho menor y por lo tanto no se va a entrar en detalles. Simplemente indicar que este módulo se limita a acceder periódicamente a su buffer de eventos y los envía accediendo a un servidor de correo mediante el protocolo SMTP (Simple Mail Transfer Protocol). Dado que el envío de correos es mucho más rápido que la comunicación por voz, el número de usuarios se encuentra mucho menos limitado, ya que, por ejemplo, en un segundo, se pueden enviar varios emails. En cualquier caso, las fórmulas anteriores son perfectamente válidas.

## CONCLUSIONES

El presente artículo ha mostrado una arquitectura para la gestión de eventos orientada al envío de notificaciones a través de diferentes medios de transporte. El sistema es escalable y adaptable, permitiendo fácilmente tanto la ampliación a nuevos tipos de dispositivos y nuevas políticas de gestión de envíos, como la implantación en otros sistemas de información distintos al utilizado en nuestro estudio. La estructura de

capas permite la instalación de las distintas partes en máquinas diferentes, permitiendo realizar balance de carga. Así mismo se ha mostrado un ejemplo de implementación del envío de alertas mediante voz, planteando los problemas propios de la naturaleza del medio y exponiendo las soluciones tomadas. Se han definido parámetros que garantizan la calidad del servicio ofrecido, permitiendo cumplir con los plazos de entrega establecidos.

### **Reconocimiento**

Esta investigación ha sido financiada por el proyecto CICYT del Ministerio de Ciencia y Tecnología, de España, cuya referencia es TRA2004-06276 / MODAL ("Desarrollo de un sistema de intercambio de información entre camiones y dispositivos externos para control de mercancías mediante el uso de una infraestructura conceptual basada en ontologías. ")

## BIBLIOGRAFÍA

BOB, Edgar. *"The VoiceXML Handbook. Understanding and Building the Phone Enabled Web"*. Dialogic Corp, 2001.

GUTHERY, Scott b., Cronin, Mary J. *"Mobile Applications Development with SMS and the SIM Toolkit"*. McGraw-Hill TELECOM, 2002.

HUIXIANG, Gu, Jianming Li, Walter, Ben, Chang, Eric. *Spoken Query for Web Search and Navigation*. Tenth International World Wide Web Conference, Hong Kong, May 1-5, 2001.  
<http://www10.org/cdrom/posters/p1010/>

SAMPER J.Javier , Cervera Arturo , Sánchez Ignacio. *"Development of voice services to provide traffic information"* (application in the SCT), presentado en el 10th World Congress and Exhibition on Intelligent Transport Systems and Services, Madrid, Spain, 16-20 Nov. 2003.

SAMPER J. Javier, Cervera Arturo, Carrillo Eduardo & Sánchez Ignacio. *"Speakeasy. Talking congestion. Traffic Technology International."* June/July 2003. ISSN 1356-9252.

SAMPER J Javier , Carrillo Eduardo, Cervera Arturo. *"Dynamic access to web contents trough voice with PHP and VoiceXML"*. ICTE 2002 (International Conference on Information Technologies in Education). Badajoz, 20-23 Noviembre 2002. ISBN: COLECCIÓN-84-95251-76-0

SHARNA , Chetan, Kunis, Jeff. *"Strategies and Techniques for Effective Voice Application Development with VoiceXML 2.0"*. Willey Computer Publishing, 2002.

World Wide Web Consortium (W3C). <http://www.w3.org>