

# Conclusiones

## 9.1 Conclusiones

Este capítulo ofrece un resumen del trabajo realizado, incluyendo un apartado con algunas ideas para proyectos futuros.

El cliente J2ME, pese a no ser la plataforma más adecuada para el desarrollo de servicios web, se ha comportado de forma óptima, no superando en ningún momento el tiempo de espera máximo en simulación de 10 segundos, que es el tiempo estimado que una persona puede centrar su atención mientras espera los resultados.

La aproximación JSR 172 es más eficiente que kSOAP en muchos aspectos, principalmente en el tamaño de las librerías y en los requisitos de memoria. Respecto a las librerías, JSR 172 está diseñado para ser incluido en el propio dispositivo, por lo que el tamaño de la aplicación se reduciría drásticamente. Otra gran ventaja que presenta JSR 172 frente a kSOAP es la absoluta transparencia al realizar la llamada al procedimiento remoto.

Por el contrario, kSOAP consigue una mejor compatibilidad con servicios anteriormente desplegados, mientras que JSR 172 por lo general requiere servicios web XML diseñados expresamente, o modificados para adaptarse a sus requisitos. Puede llevar un tiempo considerable encontrar la configuración adecuada.

Los obstáculos surgidos durante el desarrollo del proyecto se han debido principalmente a los requisitos especiales que impone JSR 172, problema potenciado al no existir mucha documentación al respecto. Como curiosidad, de los 23 servicios web de ejemplo que incluye Axis, tan sólo uno funcionó a la primera en JSR 172, mientras que kSOAP permitió la comunicación sin problemas ni modificaciones de ningún tipo en la mayoría de ellos.

El servidor Apache Tomcat en conjunción con Axis para dotarle de compatibilidad SOAP resulta ser una herramienta muy potente. Por un lado, posee facilidades que permiten desplegar y probar servicios web simples sin complicaciones mediante el despliegue automático, y por otro lado permite mucha libertad en la configuración de servicios más complejos o con restricciones gracias al lenguaje WSDD.

Por lo anteriormente comentado, a pesar de que en Apache Axis existe la posibilidad de desplegar automáticamente los servicios, la necesidad de controlar los parámetros hace necesario un despliegue del servicio mediante un documento WSDD. En el caso de kSOAP el despliegue por defecto es suficiente para asegurar la compatibilidad con el cliente.

Apache Axis también ha sido, en algunas ocasiones, fuente de problemas, ya que al intentar desplegar un servicio erróneo puede bloquearse, y en algunas ocasiones ha sido necesaria su reinstalación.

## 9.2 Líneas futuras de desarrollo

Se ha comprobado que el rendimiento en memoria del móvil es aceptable. Sin embargo los dispositivos inalámbricos tienen un segundo inconveniente: el ancho de banda no sólo es limitado, sino que comunmente la facturación se realiza por volumen de datos. Al ser los mensajes SOAP documentos XML, que contienen muchos datos redundantes y cabeceras demasiado largas, esta puede ser la mayor desventaja de los servicios web en móviles.

Como ya se ha comentado, JSR 172 no obliga a usar el protocolo SOAP directamente en el dispositivo. Una posible ampliación muy interesante sería la investigación de posibles protocolos binarios ligeros entre el móvil y la pasarela inalámbrica. Así, ésta última podría generar el mensaje SOAP para el servidor a partir de los datos que envíe el dispositivo y adaptar la respuesta de nuestra petición al nuevo protocolo para enviarlo al móvil.

Otra implementación interesante es la de dotar a nuestro servicio de seguridad. Podría existir, por ejemplo, un nombre de usuario y una contraseña, de forma que sólo los clientes abonados puedan acceder al servidor, aparte de encriptar los datos transmitidos.