

9. Conclusiones

Como es recomendable en todo proceso de *software testing*, el diseño de las pruebas ha de comenzar a realizarse nada más que se defina y planee el *software* a desarrollar, es decir, nada más que existan las primeras especificaciones. De este modo, y como resultado de este proyecto, hemos obtenido una aplicación que nos permitirá llevar a cabo un conjunto de pruebas a medida que el *software* se vaya desarrollando. Esto servirá por un lado para ayudar a los programadores en la depuración, y por otro, para verificar el funcionamiento del *software*.

En el capítulo anterior hemos comentado como la aplicación diseñada se ha ejecutado, aunque no sea para realizar su objetivo de poner a prueba un dispositivo. Sin embargo, sería interesante llevar a cabo los tests cuando el *software* lo permita, y ver cómo y cuáles son sus resultados, para poder plasmarlos como continuación de este proyecto.

Además de esto como objetivo último, este proyecto nos ha permitido ver algunos temas de gran interés. Por un lado se ha introducido y utilizado una de las tecnologías inalámbricas principales actualmente, como es Bluetooth, en un mundo en el que las comunicaciones de este tipo van adquiriendo cada vez más importancia.

Otro aspecto fundamental ha sido la programación de dispositivos móviles. Hemos visto como J2ME proporciona un soporte bastante adecuado para el desarrollo de aplicaciones de este tipo, y se ha realizado una introducción a la programación con el perfil MIDP, que nos permite crear aplicaciones de forma bastante sencilla.

Combinando los conceptos de Bluetooth y J2ME tratados, se nos ha permitido describir la API de Java para Bluetooth. Hemos podido concluir que esta API proporciona una manera bastante sencilla de utilizar esta forma de comunicación del dispositivo. Sin embargo cabe destacar que si bien los protocolos básicos de Bluetooth pueden ser cubiertos con JABWT, no todos los perfiles y posibilidades de la tecnología podrán ser implementados.

Como concepto final, y quizás el más particular, es el de la característica *Push Registry*. El inicio automático de aplicaciones que ésta nos permite, además de hacer más fácil para los usuarios el proceso de conexión, hemos visto como nos brinda una gran cantidad de nuevas posibilidades. Lo más importante que nos aporta es que podemos tener cualquier aplicación de comunicaciones siempre disponible y esperando a conexiones remotas, pero sin tener por que estar iniciada.

Esto me permite proponer el uso de *PushRegistry* para aplicaciones típicas de comunicaciones que se han ido utilizando hasta el momento. Desde sencillas herramientas de mensajería o sincronización, hasta juegos o aplicaciones empresariales.