

Tema 3

Gestión de la comunicación Bluetooth

3.1. Introducción al protocolo Bluetooth

En 1994, Ericsson se interesó en conectar sus teléfonos móviles y otros dispositivos (tales como PDAs) sin necesidad de cables. En conjunto con otras cuatro empresas formó un SIG con el propósito de desarrollar un estándar inalámbrico para interconectar ordenadores, dispositivos de comunicaciones y accesorios a través de radios inalámbricos de bajo consumo de energía, corto alcance y económicos. A dicho proyecto se le llamó Bluetooth.

En el apéndice B se muestra una descripción detallada del protocolo Bluetooth, su arquitectura, sus perfiles, su pila de protocolos asociada, la estructura de sus tramas, etc.

3.2. JSR-82. Gestión de la comunicación Bluetooth

En J2ME, la API encargada de gestionar las comunicaciones Bluetooth se llama JSR-82 [16], también conocida como “Java APIs for Bluetooth”. Esta API ha sido diseñada para depender de la configuración CLDC, aunque también existen implementaciones para poder hacer uso de este API en J2SE.

La JSR-82 consta tan sólo de dos paquetes totalmente independientes, que son: el paquete `javax.bluetooth` y el paquete `javax.obex`. En el primero se definen las clases e interfaces básicas para el descubrimiento de dispositivos Bluetooth, descubrimiento de servicios, conexión y comunicación. La comunicación a través de este paquete es a bajo nivel, es decir, mediante flujos de datos o mediante transmisión de arrays de bytes. Éste será el paquete que usaremos en este curso, principalmente.

El paquete `javax.obex`, por su parte, permite manejar el protocolo de alto nivel OBEX, muy similar a HTTP y utilizado sobre todo para el intercambio de archivos.

3.2.1. Paquete `javax.obex`

El protocolo OBEX (como ya hemos dicho, muy parecido a HTTP) consiste en el intercambio de mensajes entre un cliente y un servidor, consistentes en un conjunto de cabeceras y, opcionalmente, un cuerpo de mensaje. En este protocolo el cliente envía comandos al servidor (CONNECT, PUT, GET, DELETE, SETPATH, DISCONNECT) junto algunas cabeceras de mensaje y en ocasiones (comando PUT) un cuerpo de mensaje. Las cabeceras de mensaje están encapsuladas en un objeto `HeaderSet`¹ y el cuerpo del mensaje se lee/escrbe mediante un objeto `Input/OutputStream`.

¹<http://ants.dif.um.es/~felixgm/docencia/j2me/javadoc/jsr82/javax/obex/HeaderSet.html>