



Software de Comunicaciones  
I.T.T. Especialidad Telemática  
Escuela Politécnica Superior  
Universidad Carlos III de Madrid

## Problemas Java ME (2 puntos)

A continuación se visualizan fragmentos de código de una aplicación MIDP, `ExamenMidlet`. Esta aplicación consiste en reproducir un sonido de tipo midi almacenado previamente de forma persistente, a partir de una URL introducida por el usuario. El comportamiento de la aplicación es el siguiente: Si el usuario introduce por primera vez la URL, se descarga el contenido de ésta y se almacena en un `RecordStore music`. Asimismo, la URL se almacena en otro `RecordStore urls`. Si no es la primera vez, se reproduce directamente el contenido del recurso. Para saber si es la primera vez o no que se introduce una URL se hace una búsqueda en el objeto `RecordStore` correspondiente.

A partir del siguiente código, **responda razonadamente a las preguntas que se le realizan**, indicando el número de línea cuando sea necesario. Notése que aunque no se muestran en el código, los atributos globales del midlet son dos objetos de tipo `RecordStore`: `rs` y `rsurl`, un objeto de tipo `TextField turl`, y otro de tipo `Form main`.

```
01 public class ExamenMid extends MIDlet implements CommandListener {
02     public ExamenMid() {
03         turl = new TextField("Introduzca URL: ", "", 80, TextField.URL);
04         Command conectar = new Command("Conectar", Command.SCREEN, 0);
05         Command salir = new Command("Salir", Command.EXIT, 0);
06         main = new Form("Examen SWC 2007");
07         main.append(turl);
08         main.addCommand(conectar);
09         main.addCommand(salir);
10         main.setCommandListener(this);
11     }

10     public void startApp() throws MIDletStateChangeException {
11         Display.getDisplay(this).setCurrent(main);
12     }

18     public void commandAction(Command c, Displayable d) {
19     }

20     private void download (String url) throws IOException {
21         HttpURLConnection c = null;
22         InputStream is = null;
23         rs = RecordStore.openRecordStore("music",true);
24         rsurl = RecordStore.openRecordStore("urls",true);
25         try {
26             c = (HttpURLConnection)Connector.open(url);
27             is = c.openInputStream();
28             int len = (int)c.getLength();
29             if (len > 0) {
30                 byte[] data = readMidi(is, len);
31                 rs.addRecord(data, 0, len);
32                 byte[] b = getBytesURL(url);
33                 rsurl.addRecord(b, 0, b.length);
34             }

```

```

35     rs.closeRecordStore();
36     rsurl.closeRecordStore();
37 } finally {
38     if (is != null) is.close();
39     if (c != null) c.close();
40 }
41 }

42 private void reproducirMidi(int id) {
43     Player p;
44     VolumeControl vc;
45     try {
46         rs = RecordStore.openRecordStore("music", true);
47         InputStream bais = new ByteArrayInputStream(rs.getRecord(id));
48         rs.closeRecordStore();
49     } catch (Exception e) { e.printStackTrace(); }
50 }

51 public boolean matches(byte[] candidate) {
52     ByteArrayInputStream bais = new ByteArrayInputStream(candidate);
53     DataInputStream inputStream = new DataInputStream(bais);
54     String nurl = null;
55     try {
56         nurl = inputStream.readUTF();
57     } catch (Exception ex) { ex.printStackTrace(); }
58     return (nurl.equals(turl.getString()));
59 }

60 }

```

1. (0.8 punto) Implemente el mecanismo de búsqueda de la URL en el objeto `RecordStore rsurl`. Especifique las modificaciones que tendría que hacer en la clase para esto y utilice el método proporcionado para ello.

**Pista:** el método `enumerateRecords(RecordFilter, RecordComparator, Boolean)` del objeto `RecordStore` devuelve un objeto `RecordEnumeration` para recorrer los registros.

2. (0.7 punto) Implemente el código necesario de las acciones a realizar cuando el usuario selecciona la opción ‘‘Conectar’’. Tenga en cuenta que el método `download(String)` sólo es invocado si la URL es nueva. En caso contrario (si la ya está almacenada), se invoca el método `reproducirMidi(int)`, asumiendo que el identificador del `record` de la URL es igual al identificador del `record` que almacena el contenido del recurso en el `RecordStore music`.

**Pista:** el método `getLabel()` devuelve el texto del comando.

3. (0.5 punto) Añada el código necesario en el método `reproducirMidi(int id)` para que el contenido del recurso obtenido en el objeto `InputStream bais` pueda ser reproducido con un volumen de nivel 50.

**Pista:** el método `getControl(String)` permite obtener el control especificado y el método `setVolume(int)` permite establecer el nivel de volumen indicado.

Otras posibles preguntas sobre el código podrían ser:

1. Añada e implemente los métodos que hacen falta para completar el ciclo de la vida del `MIDlet`.

2. Escriba el fichero descriptor (.JAD) para el MIDlet Suite en el que estaría empaquetado el MIDlet anterior. Considere que sólo existe este MIDlet y que el manifiesto no contiene los atributos que describen la configuración y el perfil.
3. Añada las líneas de código necesarias para terminar la ejecución del MIDlet cuando se selecciona la opción “Salir”.
4. Explique ¿Por qué es necesario cerrar las conexiones de red?