

CAPÍTULO 8. Internet: *sockets*, HTTP y servicios web

8.1. Comunicaciones en Internet mediante *sockets*

8.1.1. La arquitectura cliente/servidor

8.1.2. ¿Qué es un *socket*?

8.1.3. Un ejemplo de un cliente / servidor de ECHO

```
public class ClienteECHO extends Activity {

    private extView output;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        output = (TextView) findViewById(R.id.TextView01);
        ejecutaCliente();
    }

    private void ejecutaCliente() {
        String ip="15.42.146.75";
        int puerto=7;
        log(" socket " + ip + " " + puerto);
        try {
            Socket sk = new Socket(ip,puerto);
            BufferedReader entrada = new BufferedReader(
                new InputStreamReader(sk.getInputStream()));
```

```

        PrintWriter salida = new PrintWriter(
            new OutputStreamWriter(sk.getOutputStream()), true);
        log("enviando...");
        salida.println("Hola Mundo");
        log("recibiendo ... " + entrada.readLine());
        sk.close();
    } catch (Exception e) {
        log("error: " + e.toString());
    }
}

private void log(String string) {
    output.append(string + "\n");
}

}

public class ServidorECHO {

    public static void main(String args[]) {
        try {
            ServerSocket sk = new ServerSocket(7);
            while (true) {
                Socket cliente = sk.accept();
                BufferedReader entrada = new BufferedReader(
                    new InputStreamReader(cliente.getInputStream()));
                PrintWriter salida = new PrintWriter(
                    new OutputStreamWriter(cliente.getOutputStream()), true);
                String datos = entrada.readLine();
                salida.println(datos);
                cliente.close();
            }
        } catch (IOException e) {
            System.out.println(e);
        }
    }
}

```

8.1.4. Almacenando las puntuaciones en un servidor por sockets

```
public class ServidorPuntuaciones {

    public static void main(String args[]) {

        Vector<String> puntuaciones = new Vector<String>();

        try {
            ServerSocket s = new ServerSocket(1234);
            while (true) {
                Socket cliente = s.accept();
                BufferedReader entrada = new BufferedReader(
                    new InputStreamReader(cliente.getInputStream()));
                PrintWriter salida = new PrintWriter(
                    new OutputStreamWriter(cliente.getOutputStream()), true);
                String datos = entrada.readLine();
                if (datos.equals("PUNTUACIONES")) {
                    for (int n=0; n<puntuaciones.size(); n++) {
                        salida.println(puntuaciones.get(n));
                    }
                } else {
                    puntuaciones.add(0, datos);
                    salida.println("OK");
                }
                cliente.close();
            }
        } catch (IOException e) {
            System.out.println(e);
        }
    }
}
```

En Asteroides utiliza la siguiente clase:

```
public class AlmacenPuntuacionesSocket implements AlmacenPuntuaciones{

    public void guardarPuntuacion(int puntos, String nombre, long fecha){
        try {
            Socket sk = new Socket("X.X.X.X", 1234);
            BufferedReader entrada = new BufferedReader(
                new InputStreamReader(sk.getInputStream()));
        }
    }
}
```

```

        PrintWriter salida = new PrintWriter(
            new OutputStreamWriter(sk.getOutputStream()), true);
        salida.println(puntos + " " + nombre);
        String respuesta = entrada.readLine();
        if (!respuesta.equals("OK")) {
            Log.e("Asterooides", "Error: respuesta de servidor incorrecta");
        }
        sk.close();
    } catch (Exception e) {
        Log.e("Asterooides", e.toString(), e);
    }
}

public Vector<String> listaPuntuaciones(int cantidad) {
    Vector<String> result = new Vector<String>();
    try {
        Socket sk = new Socket("X.X.X.X", 1234);
        BufferedReader entrada = new BufferedReader(
            new InputStreamReader(sk.getInputStream()));
        PrintWriter salida = new PrintWriter(
            new OutputStreamWriter(sk.getOutputStream()), true);
        salida.println("PUNTUACIONES");
        int n = 0;
        String respuesta;
        do {
            respuesta = entrada.readLine();
            if (respuesta != null) {
                result.add(respuesta);
                n++;
            }
        } while (n < cantidad && respuesta != null);
        sk.close();
    } catch (Exception e) {
        Log.e("Asterooides", e.toString(), e);
    }
    return result;
}
}

```

8.2. La web y el protocolo HTTP

8.2.1. El protocolo HTTP

8.2.2. Utilizando HTTP desde Android

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <EditText android:id="@+id/EditText01"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:text="palabra a buscar"/>
    <Button android:id="@+id/Button01"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="buscar en Google"/>
    <TextView android:id="@+id/TextView01"
        android:layout_height="fill_parent"
        android:layout_width="fill_parent"
        android:textSize="8pt"/>
</LinearLayout>
```

Reemplaza el código de `HTTP.java` por:

```
package org.example.http;

import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import ...;

public class HTTP extends Activity {

    private EditText entrada;
    private Button boton;
    private TextView salida;
```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    entrada = (EditText)findViewById(R.id.EditText01);
    boton = (Button)findViewById(R.id.Button01);
    salida = (TextView)findViewById(R.id.TextView01);
    boton.setOnClickListener(new OnClickListener() {
        public void onClick(View view) {
            try {
                String palabras = entrada.getText().toString();
                String resultado = resultadosGoogle(palabras);
                salida.append(palabras+"--"+resultado+"\n");
            } catch (Exception e) {
                salida.append("Error al conectar\n");
                e.printStackTrace();
            }
        }
    });
}

String resultadosGoogle(String palabras) throws Exception {
    String pagina = "", devuelve = "";
    URL url = new URL("http://www.google.es/search?hl=es&q=\""+
        URLEncoder.encode(palabras, "UTF-8")+"\"");
    HttpURLConnection conexion =
        (HttpURLConnection) url.openConnection();
    conexion.setRequestProperty("User-Agent", "Mozilla/5.0" +
        " (Linux; Android 1.5; es-ES) Ejemplo HTTP");
    if (conexion.getResponseCode() == HttpURLConnection.HTTP_OK) {
        BufferedReader reader = new BufferedReader(
            new InputStreamReader(conexion.getInputStream()));
        String linea = reader.readLine();
        while (linea != null) {
            pagina+=linea;
            linea = reader.readLine();
        }
        reader.close();
        int ini=pagina.indexOf("Aproximadamente");
        if (ini!=-1){
            int fin=pagina.indexOf(" ", ini+16);
            devuelve = pagina.substring(ini+16, fin);
        }else{
            devuelve ="no encontrado";
        }
    } else {

```

```

        salida.append("ERROR :"+conexion.getResponseMessage()+"\n");
    }
    conexion.disconnect();
    return devuelve;
}
}

```

8.3. Servicios web

8.3.1. Alternativas en los servicios web

8.3.1.1. Servicios web basados en SOAP

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  <soapenv:Header>
    <wsa:MessageID>
      uuid:920C5190-0B8F-11D9-8CED-F22EDEEBF7E5
    </wsa:MessageID>
    <wsa:To>
      http://localhost:8081/axis/services/BankPort
    </wsa:To>
  </soapenv:Header>
  <soapenv:Body>
    <axis2:echo xmlns:axis2="http://ws.apache.org/axis2">
      Hello World
    </axis2:echo>
  </soapenv:Body>
</soapenv:Envelope>

```

8.3.1.2. Servicios web basados en REST

8.3.2. Acceso a servicios web de terceros

<http://search.yahooapis.com/WebSearchService/V1/webSearch?appid=YahooDemo&query=Antonio+Banderas>

<http://developer.yahoo.com/search/web/V1/webSearch.html>

Nota: Google también dispone de un servicio similar basado en REST.
<http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=Antonio+Banderas>

```

String resultadosYahoo(String palabras) throws Exception {
    URL url = new URL("http://search.yahooapis.com/WebSearchService"
        + "/V1/webSearch?appid=YahooDemo&query=\""
        + URLEncoder.encode(palabras, "UTF-8") + "\"");
    SAXParserFactory fabrica = SAXParserFactory.newInstance();
    SAXParser parser = fabrica.newSAXParser();
    XMLReader lector = parser.getXMLReader();
    ManejadorXML manejadorXML = new ManejadorXML();
    lector.setContentHandler(manejadorXML);
    lector.parse(new InputSource(url.openStream()));
    return manejadorXML.getTotalResults();
}

public class ManejadorXML extends DefaultHandler {

    private String totalResults;

    public String getTotalResults() {
        return totalResults;
    }

    @Override
    public void startElement(String uri, String nombreLocal, String
        nombreCualif, Attributes atributos) throws SAXException {
        if (nombreLocal.equals("ResultSet")) {
            totalResults = atributos.getValue("totalResultsAvailable");
        }
    }
}

```

8.3.3. Diseño e implantación de nuestro servicio web

8.3.3.1. Instalación del servidor de servicios web

8.3.3.2. Creación un servicio web en Eclipse


```
public class PuntuacionesServicioWeb {

    static Vector<String> puntuaciones = new Vector<String>();

    public String nueva(int puntos, String nombre, long fecha) {
        puntuaciones.add(0, puntos + " " + nombre);
        return "OK";
    }

    public List<String> lista(int maximo) {
        return puntuaciones.subList(0,
            Math.min(puntuaciones.size(), maximo));
    }
}
```

8.3.3.3. Explorando el servicio web desde Eclipse

<http://localhost:8080/PuntuacionesServicioWeb/services/PuntuacionesServicioWeb?wsdl>

8.3.3.4. Explorando el servicio web desde HTML

```
<html>
<body>
    <h2>Introducir puntuación</h2>
    <form method="POST"
        action="http://localhost:8080/PuntuacionesServicioWeb/services/
PuntuacionesServicioWeb.PuntuacionesServicioWebHttpEndpoint/nueva">
        puntos: <input name="puntos" type="text" value="1234">
        nombre: <input name="nombre" type="text" value="tu nombre">
        fecha: <input name="fecha" type="text" value="0">
        <input type="submit" value="Invocar el Servicio Web">
    </form>
    <h2>Listar puntuaciones</h2>
    <form method="POST"
        action="http://localhost:8080/PuntuacionesServicioWeb/services/
PuntuacionesServicioWeb.PuntuacionesServicioWebHttpEndpoint/lista">
        maximo: <input name="maximo" type="text" value="6">
        <input type="submit" value="Invocar el Servicio Web">
    </form>
</body>
</html>
```

Nota: La URL de los parámetros `action` ha de encontrarse en una misma línea.

8.3.3.5. Utilizando el servicio web desde Android

```
public class AlmacenPuntuacionesSerWeb implements AlmacenPuntuaciones {

    public Vector<String> listaPuntuaciones(int cantidad) {
        try {
            URL url = new URL("http:// X.X.X.X:8080/"
                + "PuntuacionesServicioWeb/services/PuntuacionesServicioWeb."
                + "PuntuacionesServicioWebHttpEndpoint/lista");
            HttpURLConnection conexion = (HttpURLConnection) url
                .openConnection();

            conexion.setRequestMethod("POST");
            conexion.setDoOutput(true);
            OutputStreamWriter sal = new OutputStreamWriter(
                conexion.getOutputStream());

            sal.write("maximo=");
            sal.write(URLEncoder.encode(String.valueOf(cantidad), "UTF-8"));
            sal.flush();

            if (conexion.getResponseCode() == HttpURLConnection.HTTP_OK) {
                SAXParserFactory fabrica = SAXParserFactory.newInstance();
                SAXParser parser = fabrica.newSAXParser();
                XMLReader lector = parser.getXMLReader();
                ManejadorSerWeb manejadorXML = new ManejadorSerWeb();
                lector.setContentHandler(manejadorXML);
                lector.parse(new InputSource(conexion.getInputStream()));
                return manejadorXML.getList();
            } else {
                Log.e("Asteroides", conexion.getResponseMessage());
                return null;
            }
        } catch (Exception e) {
            Log.e("Asteroides", e.getMessage(), e);
            return null;
        }
    }

    class ManejadorSerWeb extends DefaultHandler {
        private Vector<String> lista;
        private StringBuilder cadena;

        public Vector<String> getList() {
            return lista;
        }
    }
}
```

```

@Override
public void startDocument() throws SAXException {
    cadena = new StringBuilder();
    lista = new Vector<String>();
}

@Override
public void characters(char ch[], int comienzo, int longitud){
    cadena.append(ch, comienzo, longitud);
}

@Override
public void endElement(String uri, String nombreLocal,
    String nombreCualif) throws SAXException {
    if (nombreLocal.equals("return")) {
        try {
            lista.add(URLDecoder.decode(cadena.toString(), "UTF8"));
        } catch (UnsupportedEncodingException e) {
            Log.e("Asteroides", e.getMessage(), e);
        }
    }
    cadena.setLength(0);
}
}

public void guardarPuntuacion(int puntos, String nombre, long fecha) {
    try {
        URL url = new URL("http://158.42.38.166:8080/"
            + "PuntuacionesServicioWeb/services/PuntuacionesServicioWeb."
            + "PuntuacionesServicioWebHttpEndpoint/nueva");
        HttpURLConnection conexion = (HttpURLConnection) url
            .openConnection();

        conexion.setRequestMethod("POST");
        conexion.setDoOutput(true);
        OutputStreamWriter sal = new OutputStreamWriter(
            conexion.getOutputStream());

        sal.write("puntos=");
        sal.write(URLEncoder.encode(String.valueOf(puntos), "UTF-8"));
        sal.write("&nombre=");
        sal.write(URLEncoder.encode(nombre, "UTF-8"));
        sal.write("&fecha=");
        sal.write(URLEncoder.encode(String.valueOf(fecha), "UTF-8"));
        sal.flush();
        if (conexion.getResponseCode() == HttpURLConnection.HTTP_OK) {
            SAXParserFactory fabrica = SAXParserFactory.newInstance();

```

```
SAXParser parser = fabrica.newSAXParser();
XMLReader lector = parser.getXMLReader();
ManejadorSerWeb manejadorXML = new ManejadorSerWeb();
lector.setContentHandler(manejadorXML);
lector.parse(new InputSource(conexion.getInputStream()));
if (manejadorXML.getLista().size() != 1
    || !manejadorXML.getLista().get(0).equals("OK")) {
    Log.e("Asteroides", "Error en respuesta servicio Web nueva");
}
} else {
    Log.e("Asteroides", conexion.getResponseMessage());
}
conexion.disconnect();
} catch (Exception e) {
    Log.e("Asteroides", e.getMessage(), e);
}
}

}
```