

Comunicaciones



Rogelio Ferreira Escutia

Tipos de Comunicaciones en Android

Comunicaciones en Android

- **Sockets**
- **HTTP**
- **Web Services**
- **JSON**



Sockets

Concepto de Socket

- **Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.**

Arquitectura Cliente Servidor

Cliente

1. Se conecta al servidor.
2. Solicita alguna información al servidor.
3. Recibe la respuesta.
4. Ir al punto 2, si hay más solicitudes.
5. Cierra la conexión.

Servidor

1. A la espera de que algún cliente se conecte.
2. Recibe solicitud.
3. Envía respuesta.
4. Ir al punto 2, si hay más solicitudes.
5. Cierra la conexión.
- 6- Ir al punto 1

Requisitos para conectar Sockets

- **Un par de direcciones del protocolo de red (dirección IP, si se utiliza el protocolo TCP/IP), que identifican la computadora de origen y la remota.**
- **Un par de números de puerto, que identifican a un programa dentro de cada computadora.**



Tipos de Sockets

- **Orientado a Conexión (TCP)**
- **Orientado a No Conexión (UDP)**



Sockets TCP

- **Los sockets stream ofrecen un servicio orientado a conexión, donde los datos se transfieren como un flujo continuo, sin encuadrarlos en registros o bloques.**
- **Este tipo de socket se basa en el protocolo TCP, que es un protocolo orientado a conexión.**
- **Esto implica que antes de transmitir información hay que establecer una conexión entre los dos sockets.**

Sockets UDP

- **Los sockets datagram se basan en el protocolo UDP y ofrecen un servicio de transporte sin conexión.**
- **Es decir, podemos mandar información a un destino sin necesidad de realizar una conexión previa. El protocolo UDP es más eficiente que el TCP, pero tiene el inconveniente de que no se garantiza la fiabilidad. Además, los datos se envían y reciben en datagramas (paquetes de información) de tamaño limitado.**
- **La entrega de un datagrama no está garantizada: estos pueden duplicarse, perderse o llegar en un orden diferente del que se envió.**

Código Cliente (haciendo ping)

```
private void ejecutaCliente() {
    String ip = "158.42.146.127";
    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... Hola Mundo ");
        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");
}
```

Código Servidor (en Java)

```
public class ServidorECHO {
    public static void main(String args[]) {
        try {
            System.out.println("Servidor en marcha...");
            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();
                System.out.println(datos);
            }
        } catch (IOException e) {
            System.out.println(e);
        }
    }
}
```

HTTP

La Web y el HTTP

- Dentro del mundo de Internet destaca una aplicación que es, con mucho, la más utilizada: la World Wide Web (WWW), a la que nos referiremos coloquialmente como la web.
- Para la comunicación entre los clientes y los servidores de esta aplicación, se emplea el protocolo HTTP (Hypertext Transfer Protocol) que es la utilizada por la Web.

HTTP

- **HTTP es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los navegadores web y los servidores web.**
- **Fué propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como la World Wide Web.**
- **En la web los servidores han de escuchar en el puerto 80, esperando la conexión de algún cliente web.**

Internet (1): Preparar Conexión

- Inicializamos variables

```
String pagina="";  
    String devuelve="";  
    URL url= new URL("https://www.google.com.mx/search?h1=es&q=\""+  
URLLEncoder.encode(palabras, "UTF-8")+ "\"");
```



Internet (2): Hacer Conexión

- **Hacer conexión usando HTTP**

```
URLConnection conexion= (URLConnection) url.openConnection();  
    conexion.setRequestProperty("User-Agent",  
"Mozilla/5.0" + " (Linux; Android 1.5; es-ES) Ejemplo HTTP");
```



Internet (3): Procesar resultados

- Imprimir el resultado que proviene de Internet

```
if (conexion.getResponseCode()==URLConnection.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();
}
```



Web Services

Concepto de Web Service

- **La W3C define "servicio web" como un sistema de software diseñado para permitir interoperabilidad máquina a máquina en una red.**
- **Se trata de API que son publicadas, localizadas e invocadas a través de la web.**
- **Es decir, una vez desarrolladas, son instaladas en un servidor, y otras aplicaciones (u otros servicios web) pueden descubrirlas desde otros ordenadores de Internet e invocar uno de sus servicios.**



Protocolo y Transporte

- **Como norma general, el transporte de los datos se realiza a través del protocolo HTTP y la representación de los datos mediante XML.**
- **Sin embargo, no hay reglas fijas en los servicios web y en la práctica no tiene por qué ser así.**

Ventajas de los Web Services

- Una de las grandes ventajas de este planteamiento es que es tecnológicamente neutral, es decir, podemos utilizar un servicio web sin importarnos el sistema operativo o el lenguaje en el que fue programado.
- Además, al apoyarse sobre el protocolo HTTP, puede utilizar los sistemas de seguridad (https) y presenta pocos problemas con cortafuegos, al utilizar puertos que suelen estar abiertos (80 o 8080).

Tipos de Web Services

- **Web Services basados en SOAP**
- **Web Services basados en REST**



Servicios web basados en SOAP

- **SOAP (Simple Object Access Protocol) es el protocolo más utilizado en la actualidad para implementar servicios web. Fué creado por Microsoft, IBM y otros, aunque en la actualidad está bajo el auspicio de la W3C.**
- **Utiliza como transporte HTTP, aunque también es posible utilizar otros métodos de transporte, como el correo electrónico. Los mensajes del protocolo se definen utilizando un estricto formato XML, que ha de ser consensuado por ambas partes. A continuación se muestra un posible ejemplo de mensaje SOAP**

XML (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>
```

Desventajas de SOAP en Android

- Aunque SOAP está ampliamente extendido como estándar para el desarrollo de servicios web, no resulta muy adecuado para ser utilizado en Android.
- Esto es debido a la complejidad introducida, supone una sobrecarga que implica un menor rendimiento frente a otras alternativas como REST.
- Además, Android no incorpora las librerías necesarias para trabajar con SOAP.

Web Services basados en REST

- **En primer lugar conviene destacar que el término REST se refiere a una arquitectura en lugar de a un protocolo en concreto, como es el caso de SOAP.**
- **A diferencia de SOAP, no vamos a añadir una capa adicional a la pila de protocolos, sino que utilizaremos directamente el protocolo HTTP.**
- **Siendo estrictos, la arquitectura REST no impone el uso de HTTP; no obstante, en la práctica se entiende que un servicio web basado en REST es aquel que se implementa directamente sobre la web.**

Funcionamiento de REST

- **Transporte de datos mediante HTTP, utilizando las operaciones de este protocolo, que son GET, POST, PUT y DELETE.**
- **Los diferentes servicios son invocados mediante el espacio de URI unificado. Como ya se ha tratado en este libro, una URI identifica un recurso en Internet. Este sistema ha demostrado ser flexible, sencillo y potente al mismo tiempo. Se cree que fue uno de los principales factores que motivó el éxito de WWW.**
- **La codificación de datos es identificada mediante tipos MIME (text/html, image/gif, etc.), aunque el tipo de codificación preferido es XML (text/xml).**

JSON

JSON

Frutas:

- **Manzana** - 10 unidades
- **Pera** - 20 unidades
- **Naranja** - 30 unidades

Esto se traduciría en JSON de la siguiente manera:

```
{
  "frutas": [
    { "nombre_fruta": "Manzana" , "cantidad": 10 },
    { "nombre_fruta": "Pera" , "cantidad": 20 },
    { "nombre_fruta": "Naranja" , "cantidad": 30 }
  ]
}
```



Rogelio Ferreira Escutia

***Instituto Tecnológico de Morelia
Departamento de Sistemas y Computación***

***Correo: rogelio@itmorelia.edu.mx
 rogeplus@gmail.com***

***Página Web: http://sagitario.itmorelia.edu.mx/~rogelio/
 http://www.xumarhu.net/***

***Twitter: http://twitter.com/rogeplus
Facebook: http://www.facebook.com/groups/xumarhu.net/***