

Conexión FTP en Android

En este tutorial vamos a hacer un ejemplo de conectarnos a un servidor FTP para subir archivos, listar los existentes y descargarlos, concretamente vamos a usar imágenes que hagamos con la propia cámara de nuestro Android.

Para realizar la conexión FTP vamos a usar la librería **Apache Commons Net** (<https://commons.apache.org/proper/commons-net/>), cuya última versión es la 3.3 y podemos descargar directamente desde aquí: <http://apache.rediris.es//commons/net/binaries/commons-net-3.3-bin.tar.gz>. Una vez descargada y descomprimida, tenemos que copiar el archivo “**commons-net-3.3.jar**” al directorio “**libs**” de nuestro proyecto, que se encuentra dentro del directorio “**app**”. Si estamos usando AndroidStudio, el archivo “**build.gradle**” de la aplicación ya contendrá la orden “**compile fileTree(dir: 'libs', include: ['*.jar'])**” que hará que automáticamente se compilen todos los archivos *JAR* en el directorio “**libs**”, en caso contrario, simplemente tenemos que añadirlo dentro del grupo “**dependencies**”.



```
app ×
apply plugin: 'com.android.application'

android {
    compileSdkVersion 22
    buildToolsVersion "22.0.1"

    defaultConfig {
        applicationId "com.example.germaan.pruebaftp"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.1.1'
}
```

Ahora vamos a añadir al archivo “**AndroidManifest.xml**” los permisos que vamos a necesitar para trabajar con la aplicación:

- Guardar datos en el almacenamiento externo:
android.permission.WRITE_EXTERNAL_STORAGE
- Acceso a Internet: **android.permission.INTERNET**
- Comprobar el estado de la red: **android.permission.ACCESS_NETWORK_STATE**
- Comprobar el estado del WiFi: **android.permission.ACCESS_WIFI_STATE**

Desde la versión 3 de Android no se permite realizar conexiones en segundo plano en la actividad principal de la aplicación por motivos de seguridad, así que para realizar las operaciones de subir, listas y descargar fotos vamos a crear clases que hereden de la clase “**AsyncTask<Params,Progress,Result>**”. Esta clase usará los tres parámetros genéricos de la siguiente forma:

- **Params:** es un array de objetos del tipo que hayamos definido con los parámetros que le hayamos pasado a la función durante su llamada. Este objeto es usado por la función “**doInBackground**” que es la función que se llama automáticamente al ejecutar una instancia de la clase.
- **Progress:** es un objeto cuyo valor se va actualizando según se va completando la tarea. Generalmente es un valor numérico que se usa para informar al usuario. Se accede a él mediante el método “**onProgressUpdate**”.
- **Result:** es el objeto que contiene el resultado devuelto por el método “**doInBackground**”. Se puede acceder desde la misma clase mediante el método “**onPostExecute**” o el método “**get**” de la llamada a la ejecución de la clase.
- Tendremos un clase llamada “**AsyncTaskSubirFotos**” que herede de esta, cuando queramos ejecutar una instancia de esta clase tendremos que hacer con “**objeto.execute(params)**”, u “**objeto.execute(params).get()**” si quisiéramos obtener el resultado “**result**” devuelto por el método “**doInBackground**”.

Independientemente de la tarea a realizar el procedimiento es siempre el mismo:

1. Declarar objeto del cliente FTP:

```
private FTPClient cliente = new FTPClient();
```

2. Conectarse e identificarse en el servidor:

```
cliente.connect(MainActivity.SERVIDOR, MainActivity.PUERTO);  
cliente.login(MainActivity.USUARIO, MainActivity.PASS);
```

3. Obtener el código de respuesta del servidor y comprobar que es válido:

```
reply = cliente.getReplyCode();  
  
if (FTPReply.isPositiveCompletion(reply)) {
```

4. Establecer el tipo de archivos binario y entrar en modo pasivo:

```
cliente.setFileType(org.apache.commons.net.ftp.FTP.BINARY_FILE_TYPE);  
cliente.enterLocalPassiveMode();
```

5. Si queremos subir archivos, deberemos usar el método **“storeFile”**:

```
BufferedInputStream bis = new BufferedInputStream(new FileInputStream(params[0]));  
  
resultado = cliente.storeFile(params[0].getName(), bis);  
bis.close();  
  
return resultado;
```

6. Si queremos listar los archivos, deberemos usar el método **“listFiles”**:

```
private FTPFile[] archivos = cliente.listFiles();  
  
return archivos;
```

7. Si queremos descargar archivos, deberemos usar el método **“retrieveFile”**:

```
String archivoRemoto = archivos[0];  
File archivoDescargado = new File(MainActivity.RUTA + archivoRemoto);  
OutputStream out = new BufferedOutputStream(new FileOutputStream(archivoDescargado));  
  
resultado = cliente.retrieveFile(archivoRemoto, out);  
out.close();  
  
return resultado;
```

8. Cuando finalicemos nuestra tarea siempre debemos cerrar sesión y desconectarnos del servidor:

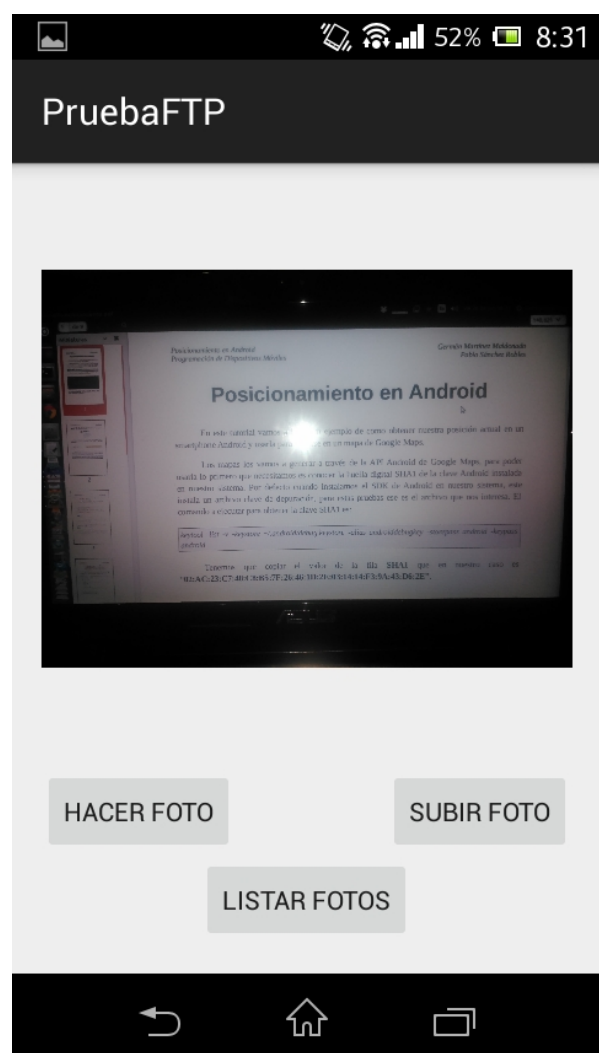
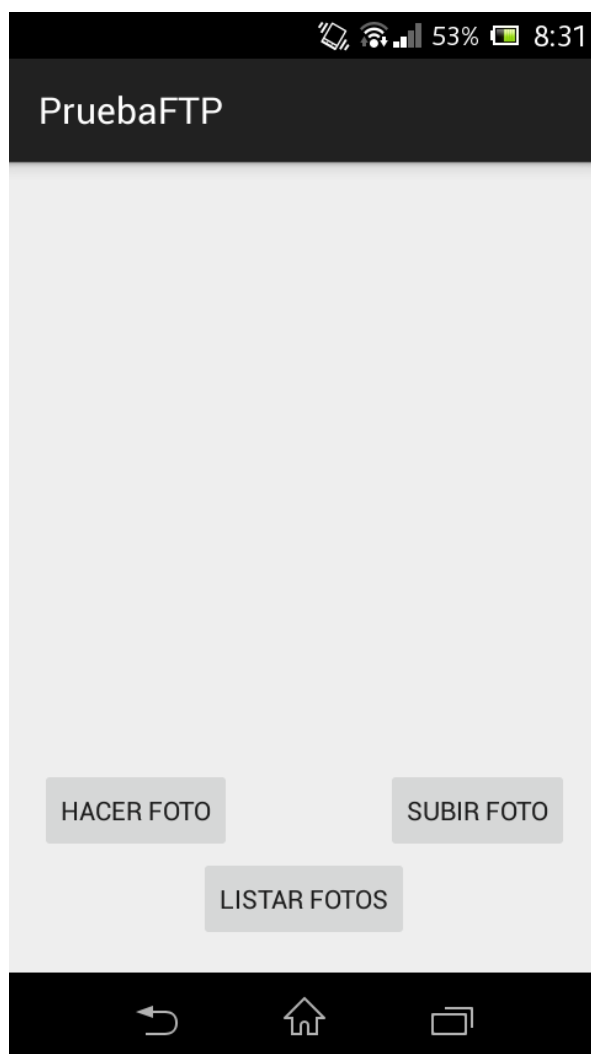
```
cliente.logout();  
  
if (cliente.isConnected()) {  
    cliente.disconnect(); }
```

9. Ahora solo nos quedará ejecutar el método de la tarea que queramos realizar:

```
AsyncTaskSubirFotos task = new AsyncTaskSubirFotos();  
fotoSubida = task.execute(archivoFoto).get();  
  
AsyncTaskListarFotos task = new AsyncTaskListarFotos();  
  
AsyncTaskDescargarFoto task = new AsyncTaskDescargarFoto();  
fotoDescargada = task.execute(fotoSeleccionada).get();
```

El código completo del ejemplo se puede encontrar en GitHub (<https://github.com/germaaan/ProgramacionDispositivosMoviles/tree/master/TutorialFTP/AndroidStudioProject>). La aplicación resultante se vería como en las siguientes imágenes:

1. Cuando pulsemos el botón **“HACER FOTO”** se abrirá nuestra aplicación de cámara de fotos y cuando capturemos una imagen, está aparecerá en la ventana de la aplicación. Si pulsamos el botón **“SUBIR FOTO”** esta imagen será subida a nuestro servidor mediante *FTP*.



Si pulsamos el botón **“LISTAR FOTOS”** se iniciará una nueva actividad que mostrará por pantalla un listado con todas las fotos en el servidor, si seleccionamos alguna, estará será descargada en segundo plano y mostrada en la ventana de una actividad de la aplicación.

