

# Programação Para Dispositivos Móveis I

## CONNECTIVIDADE

2024/\_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

# Índice

- Conectividade;
- Conectividade em Android;
- Informação da conexão;
- Conexão HTTP;
- Leitura Adicional.

# Que tipos de conexão podem existir em Android?

# Conectividade

- Funcionalidade básica para dispositivos móveis;
- Permitem a comunicação com o “exterior”;
- Não implica apenas vias de comunicação com a Internet.

# Conectividade

- Internet:

- 5G/4G/3G/2G
- Wi-Fi

- Outras:

- Bluetooth
- NFC
- IR
- ...

# Conectividade em Android

As operações de comunicação em Android devem ser feitas com cuidados adicionais:

- Necessário **explicita permissão** do utilizador;
- Devem ser feitas fora da UI thread;
- Devem ter em consideração o contexto da aplicação;
- Devem estar otimizadas para conservação de recursos (ex: bateria).


Existem vários componentes na plataforma Android para operações em background que devem ser utilizadas.

# Conectividade em Android

## Permissões [AndroidManifest.xml]

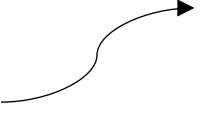
```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Permite obter o estado das redes de comunicação (Wi-Fi e 4G)



```
<uses-permission android:name="android.permission.INTERNET" />
```

Permite utilizar a comunicação à internet (seja por Wi-Fi ou por 4G)



# Conectividade em Android

## ConnectivityManager

Connectivity Manager é um serviço responsável por obter o estado de conectividade à Internet.

```
ConnectivityManager connectivityManager =  
    (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
```

Tem a capacidade para notificar as aplicações quando **existem alterações** no estado da conectividade através de callbacks.

<https://developer.android.com/reference/android/net/ConnectivityManager>



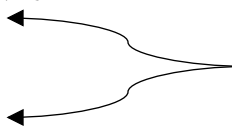
# Conectividade em Android

## ConnectivityManager

```
ConnectivityManager connectivityManager =  
    (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);  
Boolean isWifiConnected = false;  
Boolean isMobileConnected = false;
```

```
for(Network network : connectivityManager.getAllNetworks()){  
    NetworkInfo networkInfo = connectivityManager.getNetworkInfo(network);  
    if(networkInfo.getType() == ConnectivityManager.TYPE_WIFI){  
        isWifiConnected = networkInfo.isConnected();  
    }  
  
    if(networkInfo.getType() == ConnectivityManager.TYPE_MOBILE){  
        isMobileConnected = networkInfo.isConnected();  
    }  
}
```

Conexão que  
estamos a verificar



```
Log.d(DEBUG_TAG, "Wi-Fi Connected: " + isWifiConnected);  
Log.d(DEBUG_TAG, "Mobile Connected: " + isMobileConnected);
```

# Conectividade em Android

ConnectivityManager -> NetworkInfo

getState() - retorna o estado da conexão para o tipo especificado

- NetworkInfo.State.CONNECTED
- NetworkInfo.State.DISCONNECTED
- NetworkInfo.State.CONNECTING
- NetworkInfo.State.DISCONNECTING
- NetworkInfo.State.UNKNOWN

isConnected() - indica se existe conectividade e se é possível haver transferência de dados;

isConnectedOrConnecting() - indica se existe conectividade ou se a mesma está a ser estabelecida;

# Conectividade em Android

## Conexão HTTP

Criar um objeto URL com o endereço a comunicar

```
URL url = new URL("http://www.mywebsite.com");
```

## Abrir conexão HTTP

```
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
```

## Definição dos tempos de timeout

```
connection.setConnectTimeout(3000);  
connection.setReadTimeout(3000);
```

## Escolher o tipo de método HTTP a utilizar (GET/PUT/POST/DELETE)

```
connection.setRequestMethod("GET");
```

# Conectividade em Android

## Conexão HTTP

Estabelecer a conexão e obter o tamanho do ficheiro  
(se o tamanho for inferior a 0 é porque ocorreu um erro)

```
connection.getContentLength();
```

Obter o código/mensagem da resposta HTTP

```
connection.getResponseCode();  
connection.getResponseMessage();
```

Obter resposta HTTP

```
InputStream inputStream = connection.getInputStream();
```

# Conectividade em Android

## Conexão HTTP

Os pedidos de conexão HTTP devem ser feitos em background. Devemos procurar os componentes adequados para as operações:

- ExecutorService
- DownloadManager
- Volley
- Retrofit

A tentativa de conexão na UI Thread pode resultar na exceção `NetworkOnMainThreadException`.

# Outros Recursos para Comunicação

Podemos consultar outras bibliotecas para outros tipos de comunicação em Android:

- Bluetooth <https://developer.android.com/guide/topics/connectivity/bluetooth>
- NFC <https://developer.android.com/guide/topics/connectivity/nfc>
- USB <https://developer.android.com/guide/topics/connectivity/usb>

# Leitura Adicional

- Conectividade Android:

<https://developer.android.com/guide/topics/connectivity>

<https://developer.android.com/training/basics/network-ops/connecting.html>

- Volley para operações de comunicação:

<https://developer.android.com/training/volley>

<https://github.com/google/volley>

# Programação Para Dispositivos Móveis I

## CONNECTIVIDADE

2024/\_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , [rmb@estg.ipp.pt](mailto:rmb@estg.ipp.pt)

Carlos Aldeias, [cfpa@estg.ipp.pt](mailto:cfpa@estg.ipp.pt)

Adaptação do conteúdo dos slides de João Ramos [jrmr@estg.ipp.pt](mailto:jrmr@estg.ipp.pt) e Fábio Silva [fas@estg.ipp.pt](mailto:fas@estg.ipp.pt)