



++

45697056

Sistemas para Internet

Desenvolvimento Mobile e IOT - Android

45697056

Bluetooth

45697056

...



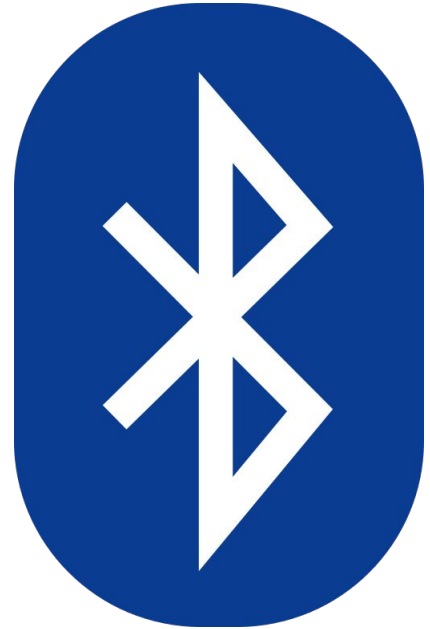
Bluetooth

Muitas vezes precisamos de uma comunicação entre dispositivos.

Podemos optar por enviar ou receber dados através de redes Wi-Fi e 3G.

Porém, o Android também disponibiliza uma API para comunicação via Bluetooth, uma alternativa para transmissão de dados em curta distância.

Em nossa aula, vamos listar dispositivos pareados e trocar algumas informações com o dispositivo selecionado.



Inserir no arquivo AndroidManifest.xml as permissões para manipular o Bluetooth:

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

Inserir um Spinner, EditText e um Button no layout da Activity Principal.

<Spinner

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/spDispositivos"/>
```

<EditText

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/txtInformacao"/>
```

<Button

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Enviar"  
    android:onClick="enviar"/>
```

Referenciar os componentes no Java:

```
public class MainActivity extends AppCompatActivity {  
  
    Spinner spDispositivos;  
    EditText txtInformacao;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        spDispositivos = (Spinner) findViewById(R.id.spDispositivos);  
        txtInformacao = (EditText) findViewById(R.id.txtInformacao);  
    }  
  
    public void enviar(View view) {  
    }  
}
```

Abaixo das variáveis declaradas anteriormente, incluir as variáveis necessárias para descoberta e manipulação dos dispositivos bluetooth pareados com o Android.

```
//Variáveis necessárias para a comunicação Bluetooth  
BluetoothAdapter bluetooth = BluetoothAdapter.getDefaultAdapter();  
BluetoothSocket soquete = null;  
OutputStream saida = null;  
Set<BluetoothDevice> dispositivosPareados;
```

Bluetooth - Passo 05

Vamos agora, preencher o spinner com os dispositivos já pareados. Incluir no método **onCreate** um **ArrayAdapter** para o spinner:

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
adapter.add("Selecione um dispositivo");

if (bluetooth != null) {

    if (!bluetooth.isEnabled()) {
        Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        int REQUEST_ENABLE_BT = 1;
        startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
    }

    dispositivosPareados = bluetooth.getBondedDevices();

    if ( dispositivosPareados.size() > 0 ) {
        for (BluetoothDevice item : dispositivosPareados) {
            adapter.add(item.getName());
        }
    }
}

spDispositivos.setAdapter(adapter);
```

Vamos criar um método chamado **criarSoqueteBluetooth** que terá como parâmetro um **BluetoothDevice** e será responsável pela criação de um socket para comunicação com este device.

```
private BluetoothSocket criarSoqueteBluetooth(BluetoothDevice dispositivo) throws IOException {  
    Method metodo;  
    BluetoothSocket tmpSoquete = null;  
  
    try {  
        metodo = dispositivo.getClass().getMethod("createRfcommSocket", new Class[] {int.class});  
        tmpSoquete = (BluetoothSocket) metodo.invoke(dispositivo, 1);  
    } finally {  
        return tmpSoquete;  
    }  
}
```


Bluetooth - Passo 07

No evento do botão enviar, vamos percorrer os dispositivos pareados, comparar com o que foi selecionado pelo usuário e então enviar o dado através do socket criado com o método anterior.

```
for (BluetoothDevice item : dispositivosPareados) {
    if (spDispositivos.getSelectedItem().toString().equalsIgnoreCase(item.getName())) {
        try {
            BluetoothDevice dispositivoRemoto = bluetooth.getRemoteDevice(item.getAddress());
            soquete = criarSoqueteBluetooth(dispositivoRemoto);
            soquete.connect();

            bluetooth.cancelDiscovery();
            saida = soquete.getOutputStream();

            byte[] buffer = txtInformacao.getText().toString().getBytes();
            saida.write(buffer);
            saida.close();
            soquete.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Bluetooth - Finalizado



A nossa integração com bluetooth está finalizada.

Agora é possível enviar comandos para uma placa Arduino por exemplo =)



Copyright © 2017 Prof. Douglas Cabral <douglas.cabral@fiap.com.br> <https://www.linkedin.com/in/douglascabral/>

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).