Now, let's look at the code for our app:

Once we've finished the layout, let's now look at how to create the code and make our app workable.

#### For that:

- 1. Click on the **Blocks** button (right side of the screen).
- 2. On the **Blocks** screen, create the first block according to **Figure 8**:

```
when <code>IpSelectBT v</code> .Before Picking

do set <code>IpSelectBT v</code> . Elements v to btClient v . Addresses And Names v
```

Figure 8 – First Block.

# This first block tells us the following:

When the **Select Bluetooth** button is clicked, a list of all Bluetooth devices available on your phone will be shown. You should click on the Bluetooth for the project.

1. Now create the second block according to **Figure 9**:

```
when [IpSelectBT v After Picking selection do set [IbIMessage v . Text v to call btClient v . Connect address [IpSelectBT v . Selection v then set [IbIMessage v . Text v to v Connected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Disconnected v else set [IbIMessage v . Text v to v Di
```

Figure 9 – Second Block.

#### This second block tells us the following:

After selecting the corresponding Bluetooth, it will be checked if the connection was successful. If so, the message "Connected" will be shown on the label, otherwise, the message "Disconnected" will be shown.

1. Now create the next block according to **Figure 10**:

```
when btnRed v .Click
do set lblValueColor v . Text v to v 255, 0, 0 v
set layoutColors v . Background Color v to call btClient v .Send 1 Byte Number
number 1
```

#### Figure 10 – Block referring to the btnRed button.

# This block tells us the following:

When the **btnRed** button is clicked, the red color will be shown as a background in the **layoutColors** component and its respective decimal value in the **lblValueColor** label, and at the same time the application's Bluetooth will send the numerical value "1" to Arduino's Bluetooth.

1. Now create the next block according to **Figure 11**:

```
when btnGreen v .Click

do set lblValueColor v . Text v to t 0, 255, 0 "

set layoutColors v . Background Color v to call btClient v .Send 1 Byte Number

number (2)
```

Figure 11 – Block referring to the btnGreen button.

# This block tells us the following:

When the **btnGreen** button is clicked, the green color will be shown as a background in the **layoutColors** component and its respective decimal value in the **lblValueColor** label, and at the same time the application's Bluetooth will send the numerical value "2" to Arduino's Bluetooth.

1. Now create the next block according to **Figure 12**:

```
when btnBlue v .Click
do set IbiValueColor v . Text v to v 0, 0, 255 v
set IayoutColors v . Background Color v to call btClient v .Send 1 Byte Number
number (3)
```

Figure 12 – Block referring to the btnBlue button.

# This block tells us the following:

When the **btnBlue** button is clicked, the blue color will be shown as the background in the **layoutColors** component and its respective decimal value in the **lblValueColor** label, and at the same time the application's Bluetooth will send the numerical value "3" to the Bluetooth of the Arduino.

1. Now create the next block according to **Figure 13**:

```
when btnYellow v.Click

do set IbiValueColor v. Text v to t " 255, 255, 0 "

set IayoutColors v. Background Color v to call btClient v.Send 1 Byte Number

number (4)
```

Figure 13 – Block referring to the btnYellow button.

#### This block tells us the following:

When the **btnYellow** button is clicked, the yellow color will be shown as a background in the **layoutColors** component and its respective decimal value in the **lblValueColor** label, and at the same time the application's Bluetooth will send the numerical value "4" to Arduino's Bluetooth.

1. Now create the next block according to **Figure 14**:

```
when btnBrown v .Click

do set lblValueColor v . Text v to v 128, 0, 0 v set layoutColors v . Background Color v to call btClient v .Send 1 Byte Number number 5
```

Figure 14 – Block referring to the btnBrown button.

### This block tells us the following:

When the **btnBrown** button is clicked, the brown color will be shown as a background in the **layoutColors** component and its respective decimal value in the **lblValueColor** label, and at the same time the application's Bluetooth will send the numeric value "5" to Arduino's Bluetooth.

1. Now create the next block according to **Figure 15**:

```
when btnPurple v .Click
do set liblValueColor v . Text v to v 255, 0, 255 v
set layoutColors v . Background Color v to call btClient v .Send 1 Byte Number
number (6)
```

Figure 15 – Block referring to the btnPurple button.

### This block tells us the following:

When the **btnPurple** button is clicked, the lilac color will be shown as a background in the **layoutColors** component and its respective decimal value in the

**lblValueColor** label, and at the same time the application's Bluetooth will send the numeric value "6" to the Bluetooth of the Arduino.

1. Finally, create the last block according to **Figure 16**:

```
when btnDisconnect · .Click
do call btClient · .Disconnect
set (lblMessage · . Text · to b " Desconnected "
```

Figure 16 – Block referring to the btnDisconnect button.

### This block tells us the following:

When the **btnDisconnect** button is clicked, Bluetooth will be disconnected.



### Download the files .aia and .apk from the link:

www.simuladosetutoriais.com/arquivos\_projetos/KodularLEDRGBluetoothEnglish.aia www.simuladosetutoriais.com/arquivos\_projetos/KodularLEDRGBluetoothEnglish.apk

