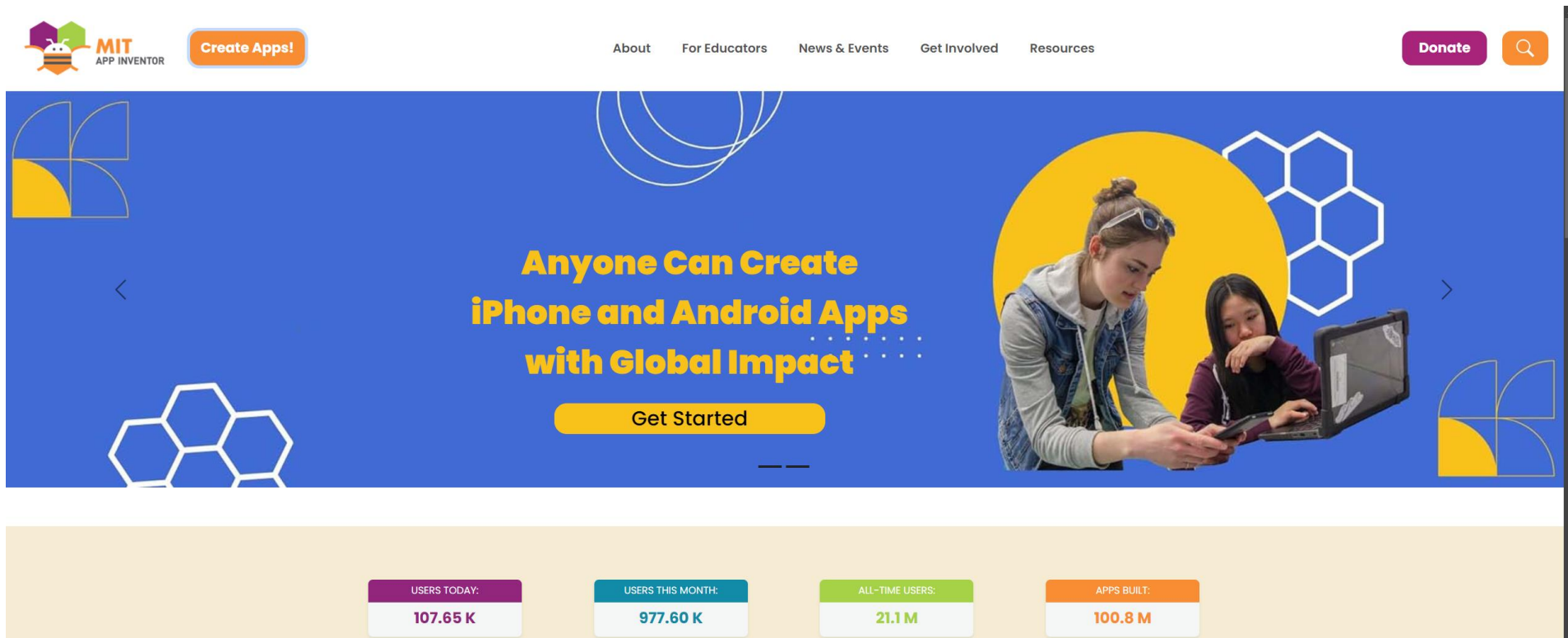


# Aplicativo para conexão mqtt

Prof. Me. Hélio Esperidião

# Click em Create Apps



The image shows the MIT App Inventor website banner. At the top left is the MIT App Inventor logo, which includes a stylized robot head and the text 'MIT APP INVENTOR'. To its right is an orange button that says 'Create Apps!'. Further right is a navigation menu with links: 'About', 'For Educators', 'News & Events', 'Get Involved', and 'Resources'. On the far right of the top bar is a purple 'Donate' button and an orange search icon. The main banner has a blue background with white geometric patterns (hexagons and circles). In the center, the text reads 'Anyone Can Create iPhone and Android Apps with Global Impact' in yellow and white. Below this text is a yellow 'Get Started' button. To the right of the text is a photograph of two young women looking at a laptop. The banner is flanked by white arrows pointing left and right. At the bottom of the banner, there are four statistics displayed in colored boxes: 'USERS TODAY: 107.65 K' (purple), 'USERS THIS MONTH: 977.60 K' (teal), 'ALL-TIME USERS: 21.1 M' (green), and 'APPS BUILT: 100.8 M' (orange).

**MIT APP INVENTOR** [Create Apps!](#)

[About](#) [For Educators](#) [News & Events](#) [Get Involved](#) [Resources](#) [Donate](#) [Search](#)

**Anyone Can Create iPhone and Android Apps with Global Impact**

[Get Started](#)

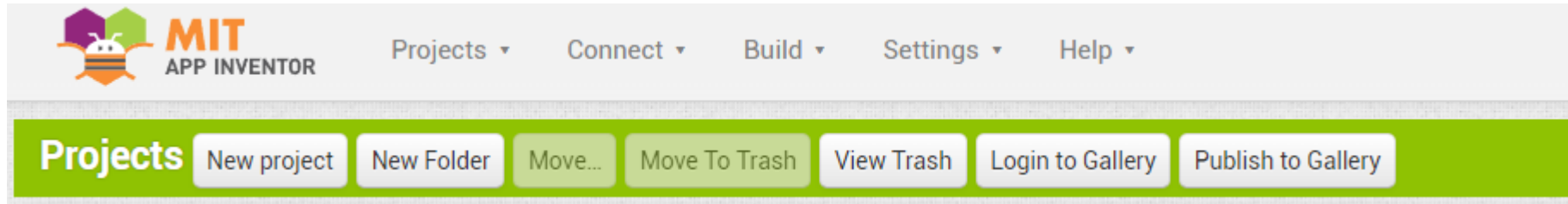
**USERS TODAY:** 107.65 K

**USERS THIS MONTH:** 977.60 K

**ALL-TIME USERS:** 21.1 M

**APPS BUILT:** 100.8 M

Click em new Project



# Click em ok

**Create new App Inventor project**

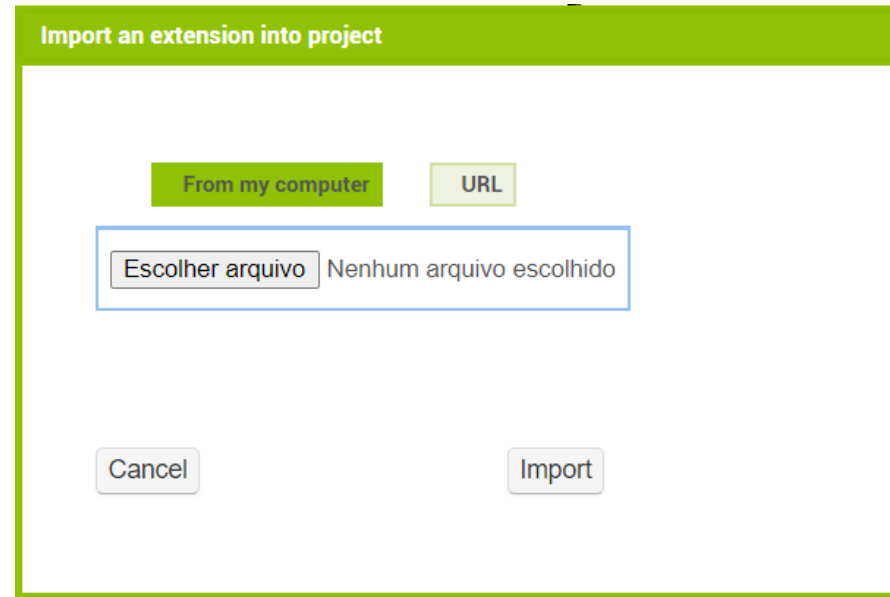
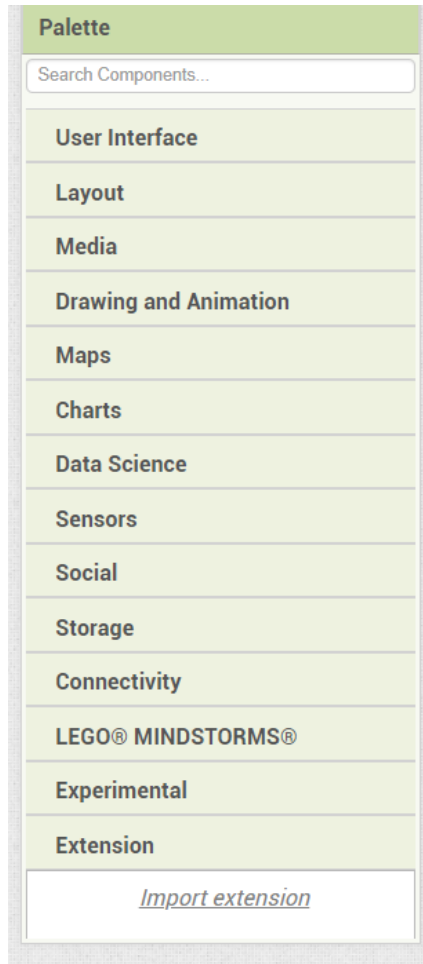
Project name:

Toolkit:  ?

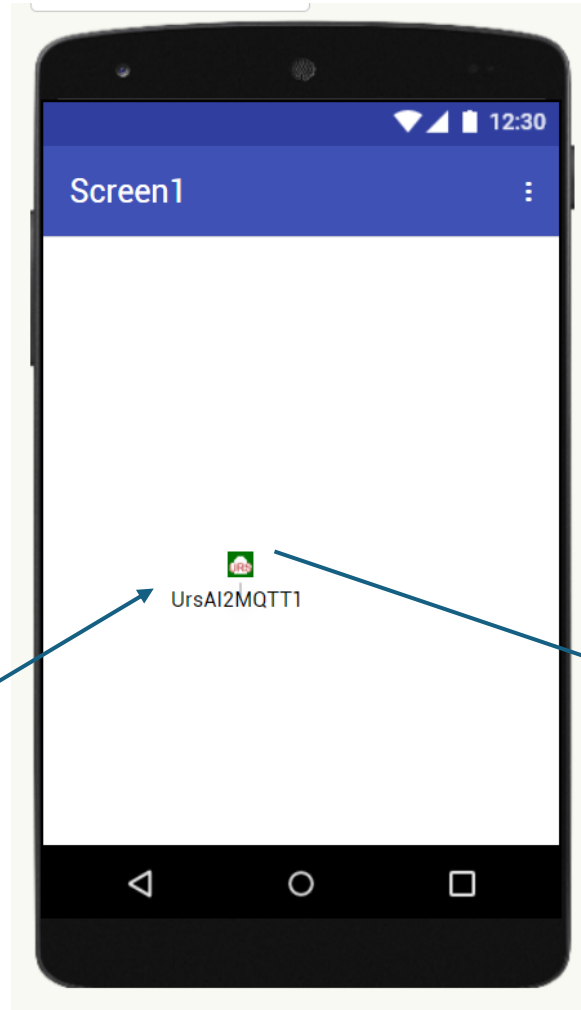
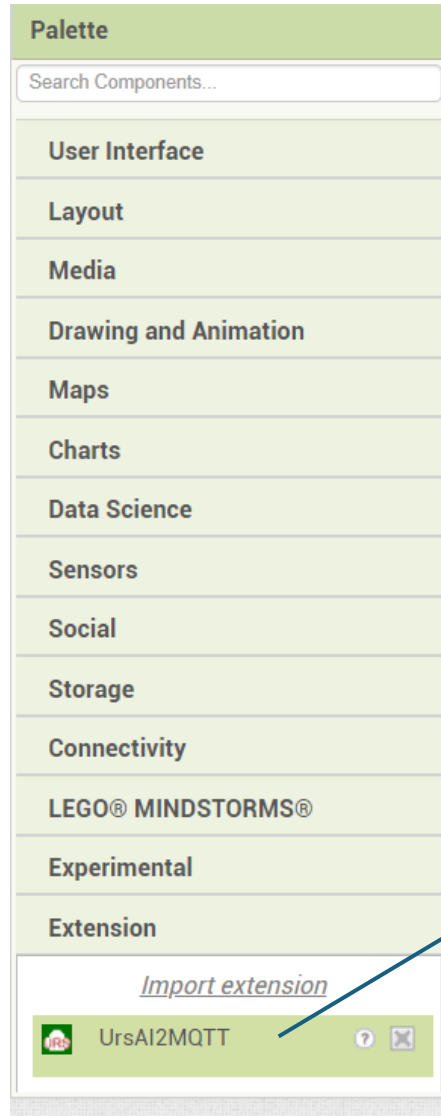
Theme:  ?

# Click em **Import Extension**

- Procure o arquivo:  
**de.UllisRoboterSeite.UrsAI2MQTT.aix**  
no material da aula



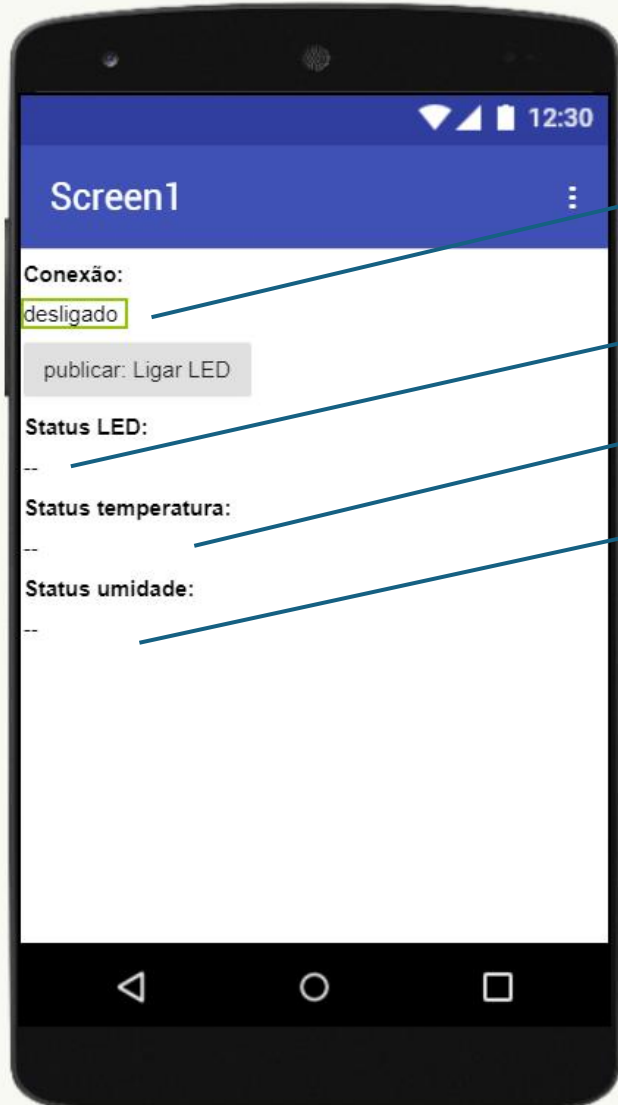
# Arraste o mqtt para a Screen 1



☐ Display hidden components in Viewer

Phone size (320 x 505) ▼

Android 5+ Devices ▼



All Components ▼

- Screen1
  - Label1
  - Label2
  - Button1
  - Label3
  - Label4
  - Label5
  - Label6
  - Label7
  - Label8
  - UrsAI2MQTT1

Rename

Delete

Media

Upload File ...

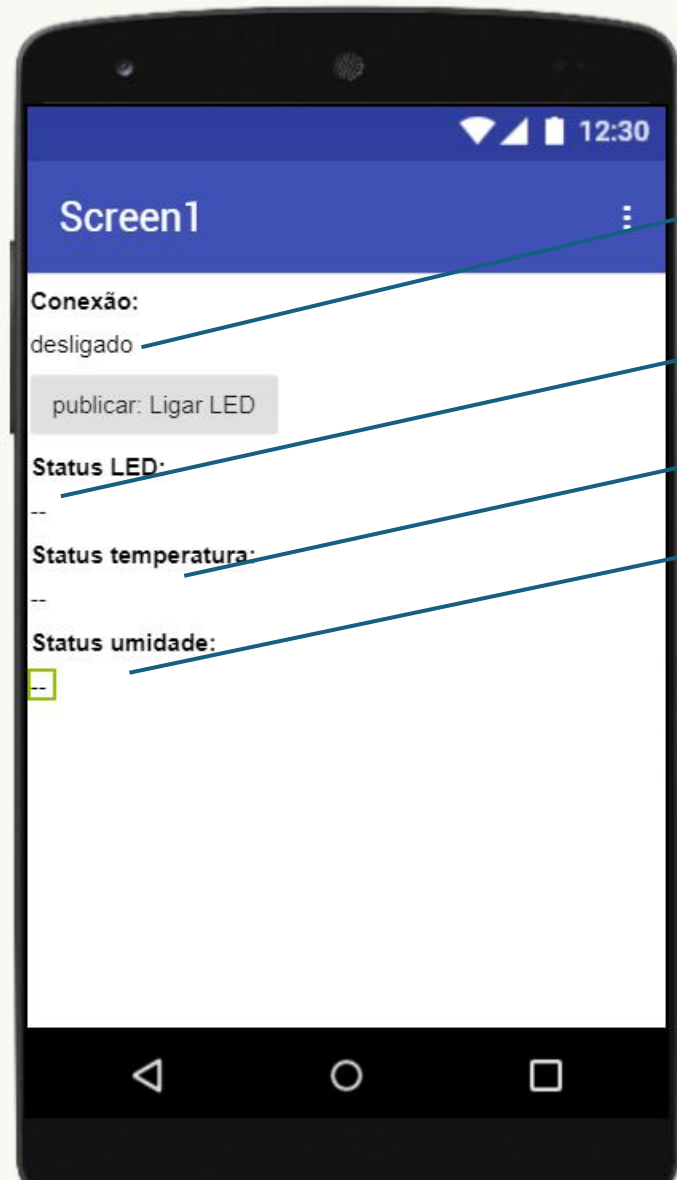
# Faça o rename dos componentes



☐ Display hidden components in Viewer

Phone size (320 x 505) ▼

Android 5+ Devices ▼



- Screen1
  - Label1
  - lblConexao
  - Button1
  - Label3
  - lblLed
  - Label5
  - lblTemperatura
  - Label7
  - lblUmidade
  - UrsAI2MQTT1

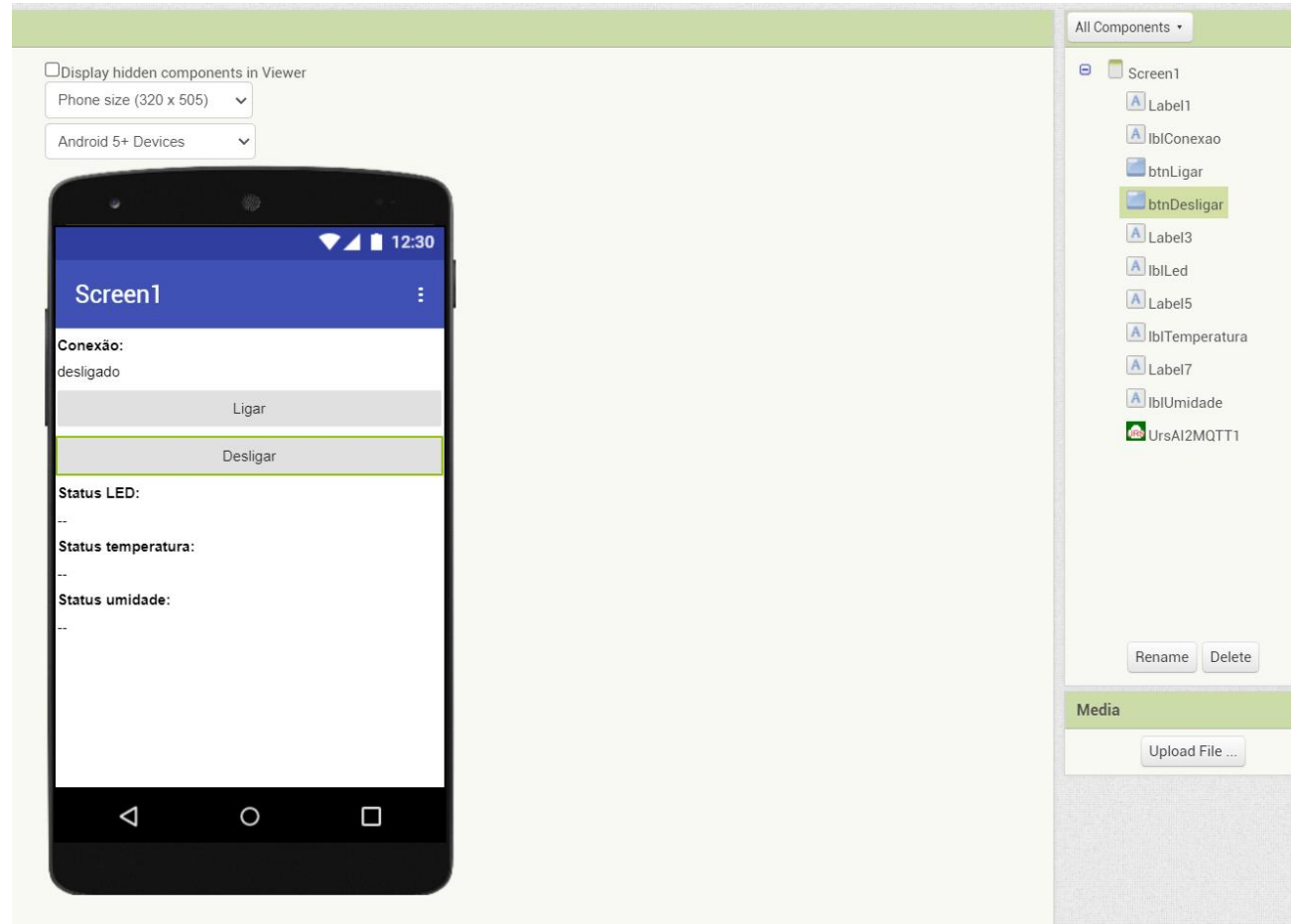
Rename

Delete

Media

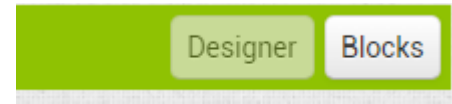
Upload File ...

# Versão final dos nomes e componentes



# Programação por blocos

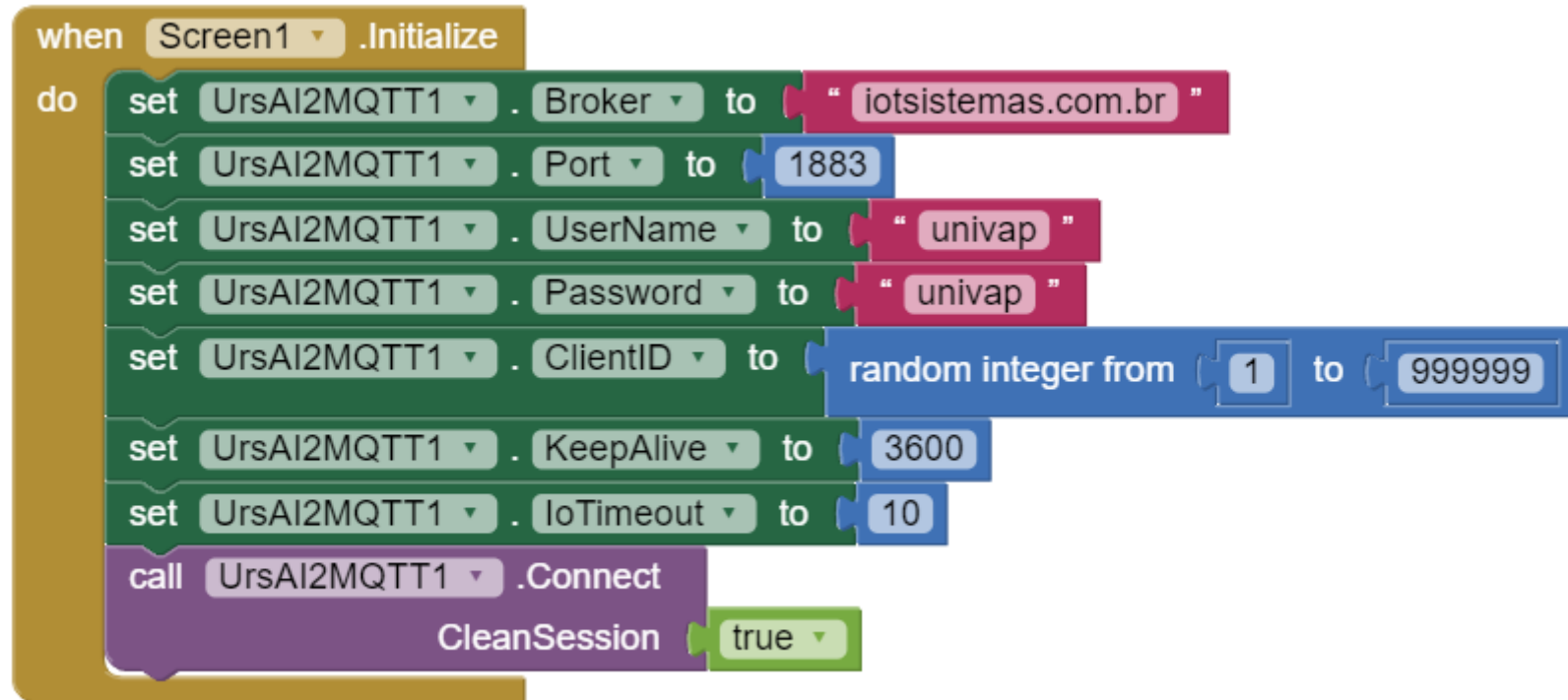
- Click em **Blocks** para programar



# Bloco: initialize

- É executado quando a tela 1 é carregada na memória
- Algoritmo desse bloco:
  - Inicializar as variáveis de conexão com o broker.
  - **Broker**: endereço do broker mqtt(servidor)
  - **Porta**: porta de rede do broker
  - **userName**: nome de usuario
  - **Password**: senha
  - **ClientId**: identificação única do cliente, não deve haver cliente igual
  - **keepAlive**: tempo que o cliente fica conectado no broker: 3600 = 1 hora
  - **IoTimeOut**: tempo máximo para o broker responder

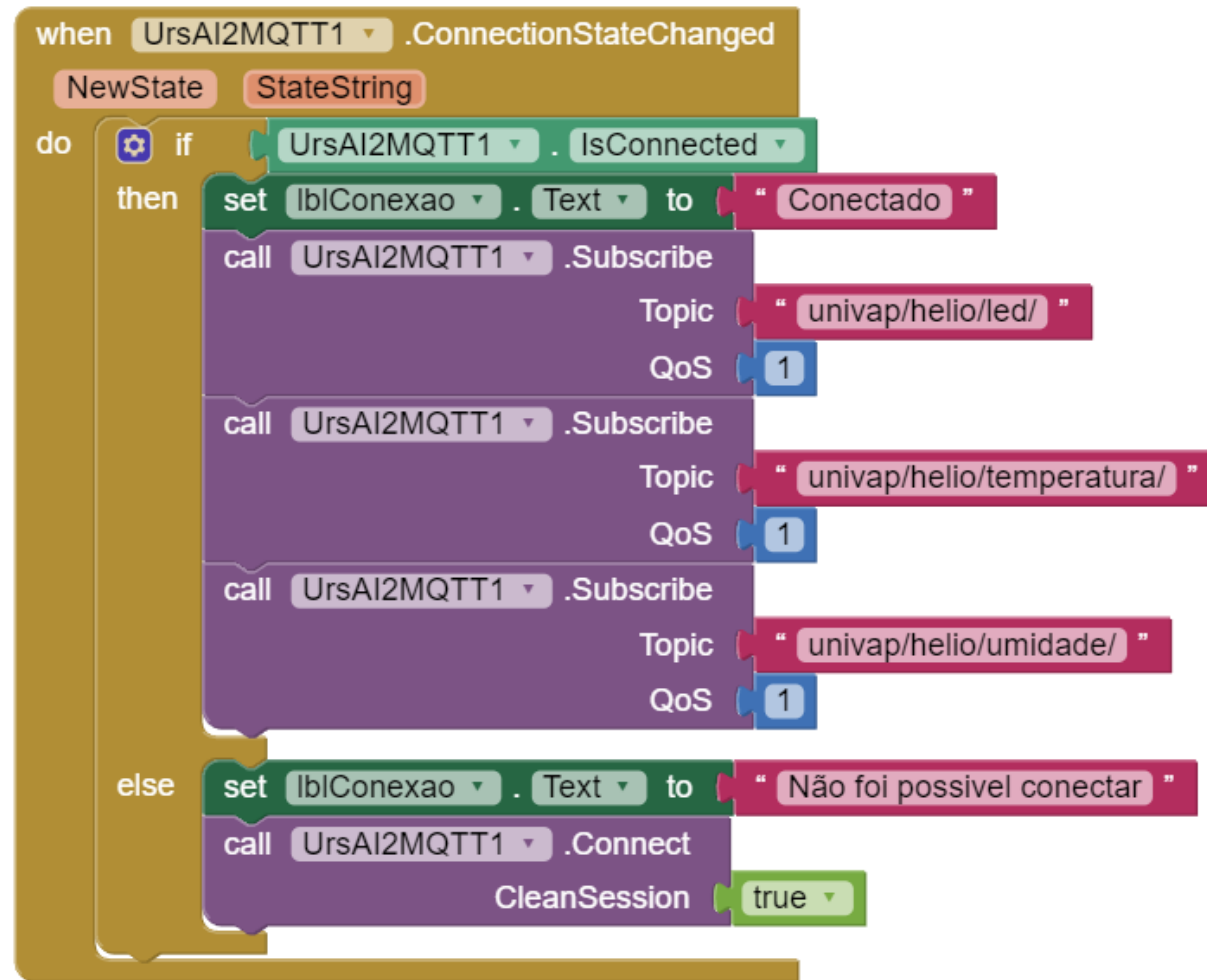
# Bloco: initialize



# Bloco: `connectionStateChange`

- É um evento que detecta variações no estado na conexão.
  - Algoritmo do bloco:
    - Se conseguir conectar
      - Inscrever-se nos canais.
        - `univap/helio/led/`
        - `univap/helio/temperatura/`
        - `univap/helio/umidade/`
    - Senão conseguir conectar ou desconectar
      - Tentar conectar novamente.

# Bloco: connectionStateChange

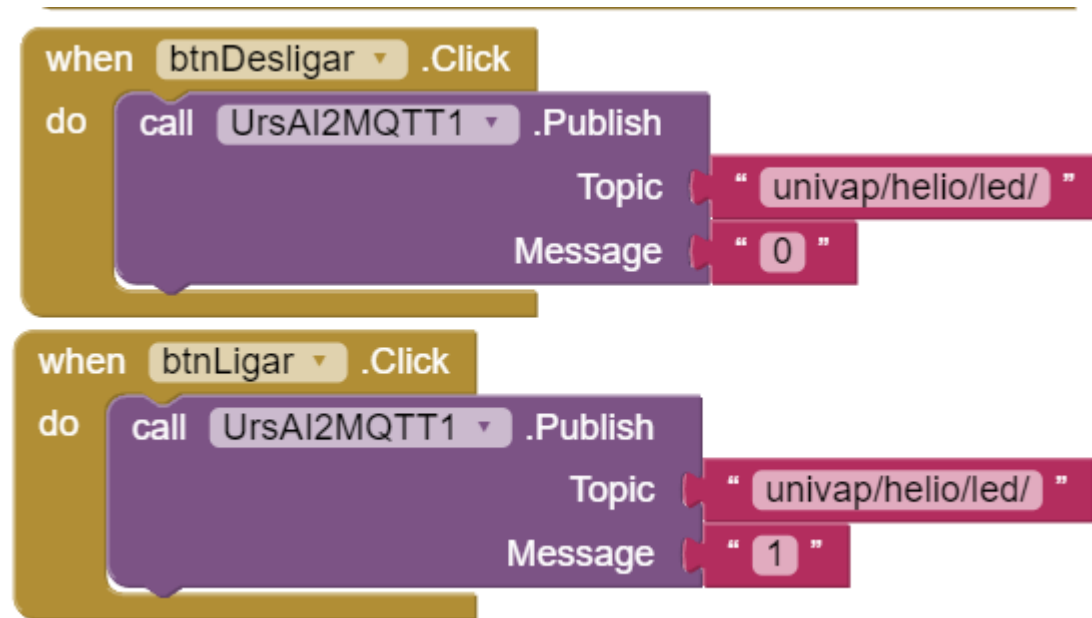


# Blocos de click nos botões

- Código é executado quando é efetuado algum click em dos botões: ligar/desligar led.
- Algoritmo
  - Se efetuar o click no botão de ligar
    - Publica no canal univap/helio/led/
      - o valor 1
  - Se efetuar o click no botão de desligar
    - Publica no canal univap/helio/led/
      - o valor 0



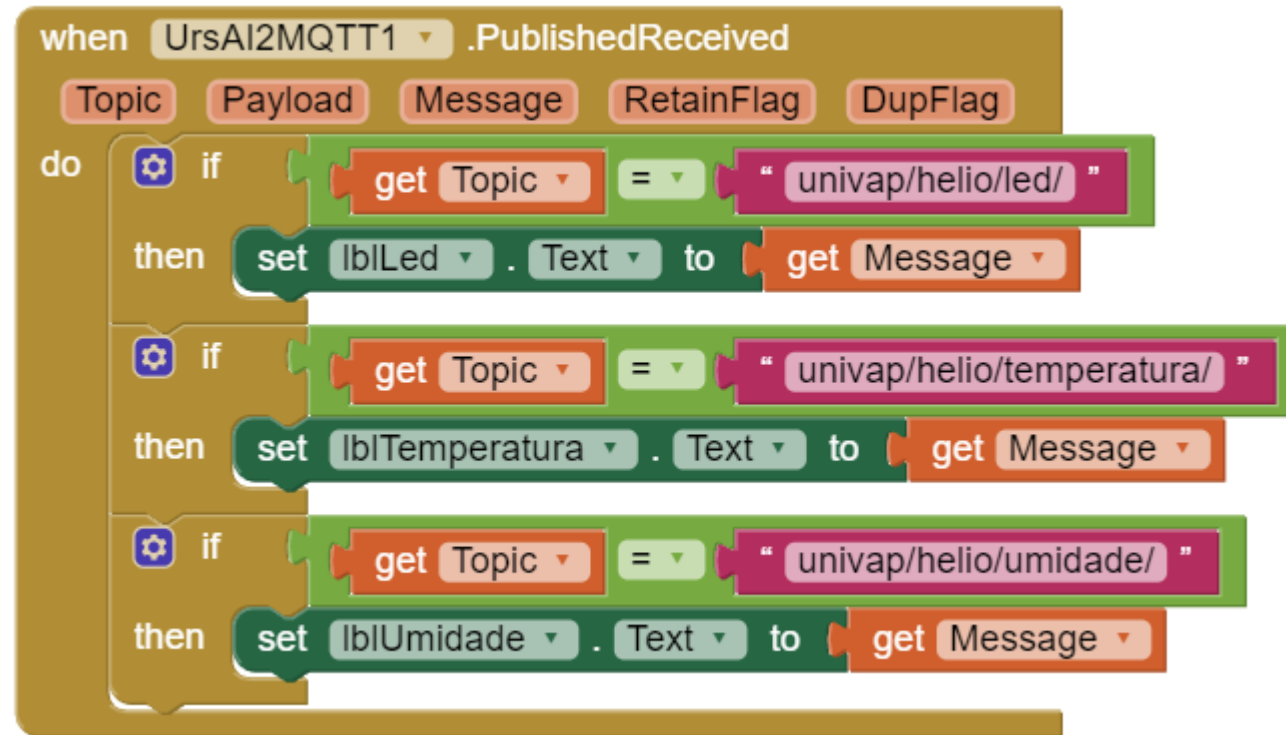
# Blocos de click nos botões



# Bloco: PublishedReceveid

- É executado automaticamente quando é recebida uma publicação em um canal inscrito.
- Algoritmo
  - Se receber uma publicação do tópico: univap/helio/led/
    - Atualiza o label `lblLed`
  - Se receber uma publicação do tópico: univap/helio/temperatura/
    - Atualiza o label `lblTemperatura`
  - Se receber uma publicação do tópico: univap/helio/umidade/
    - Atualiza o label `lblUmidade`

# Bloco: PublishedReceveid



```

when Screen1.Initialize
do
  set UrsAI2MQTT1.Broker to "iotsistemas.com.br"
  set UrsAI2MQTT1.Port to 1883
  set UrsAI2MQTT1.UserName to "univap"
  set UrsAI2MQTT1.Password to "univap"
  set UrsAI2MQTT1.ClientID to random integer from 1 to 999999
  set UrsAI2MQTT1.KeepAlive to 3600
  set UrsAI2MQTT1.KeepAlive to 10
  call UrsAI2MQTT1.Connect
  CleanSession true

```

```

when UrsAI2MQTT1.ConnectionStateChanged
  NewState
  StateString
do
  if UrsAI2MQTT1.IsConnected
  then
    set lblConexao.Text to "Conectado"
    call UrsAI2MQTT1.Subscribe
      Topic "univap/helio/led/"
      QoS 1
    call UrsAI2MQTT1.Subscribe
      Topic "univap/helio/temperatura/"
      QoS 1
    call UrsAI2MQTT1.Subscribe
      Topic "univap/helio/umidade/"
      QoS 1
  else
    set lblConexao.Text to "Não foi possível conectar"
    call UrsAI2MQTT1.Connect
    CleanSession true
  end

```

```

when UrsAI2MQTT1.PublishedReceived
  Topic
  Payload
  Message
  RetainFlag
  DupFlag
do
  if get Topic = "univap/helio/led/"
  then
    set lblLed.Text to get Message
  end
  if get Topic = "univap/helio/temperatura/"
  then
    set lblTemperatura.Text to get Message
  end
  if get Topic = "univap/helio/umidade/"
  then
    set lblUmidade.Text to get Message
  end

```

```

when btnDesligar.Click
do
  call UrsAI2MQTT1.Publish
    Topic "univap/helio/led/"
    Message "0"

```

```

when btnLigar.Click
do
  call UrsAI2MQTT1.Publish
    Topic "univap/helio/led/"
    Message "1"

```

# Referência

- <https://ullisroboterseite.de/android-AI2-MQTT-en.html#down>
- <https://www.youtube.com/watch?v=R9VF7U2wLMc&t=149s>
- <https://ullisroboterseite.de/android-AI2-PahoMQTT.html>