

AULA 2  
INTERNET DAS COISAS

SISTEMA DE AVISO DE LUMINOSIDADE

IoT - Internet of Things

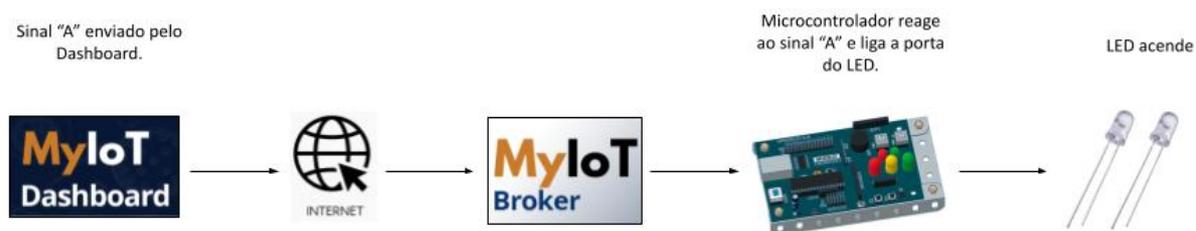
Modelix Robotics

# Índice

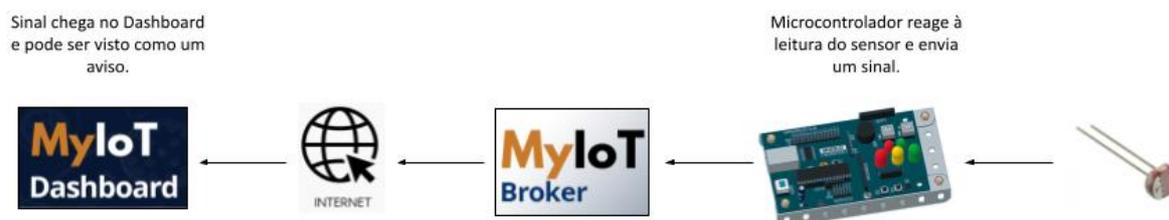
|   |    |
|---|----|
| 1.0. Introdução .....   | 2  |
| 2.0. Recurso de avisos no Dashboard.....                        | 2  |
| 3.0. Projeto 2 - Sistema de aviso de luminosidade.....          | 2  |
| 3.0.1. <i>O que você precisa para criar este projeto?</i> ..... | 4  |
| 3.0.2. <i>Programação do microcontrolador</i> .....             | 4  |
| 3.0.2.1. Opção 1: Modelix System Starter.....                   | 4  |
| 3.0.2.1.1. <i>Análise dos intervalos do sensor de luz</i> ..... | 4  |
| 3.0.2.1.2. <i>Fluxograma</i> .....                              | 4  |
| 3.0.2.2. Opção 2: Modelix Flasher.....                          | 4  |
| 3.0.3. <i>Configurando os avisos</i> .....                      | 5  |
| 3.0.4. <i>Conectando o microcontrolador com o Broker</i> .....  | 8  |
| 3.0.5. <i>Vendo os avisos no Dashboard</i> .....                | 10 |
| 4.0. Resumo e futuras iterações.....                            | 11 |

## 1.0. Introdução

Na aula passada, entendemos uma comunicação básica pela internet para ligar e desligar um LED. Após um envio de sinal através da central de comunicação Dashboard, o programa chamado “broker” conectado à internet acessou o sinal e o enviou para o microcontrolador conectado ao computador. Esse, que estava programado para realizar uma função caso este sinal fosse comunicado à ele, ligou e desligou o LED.



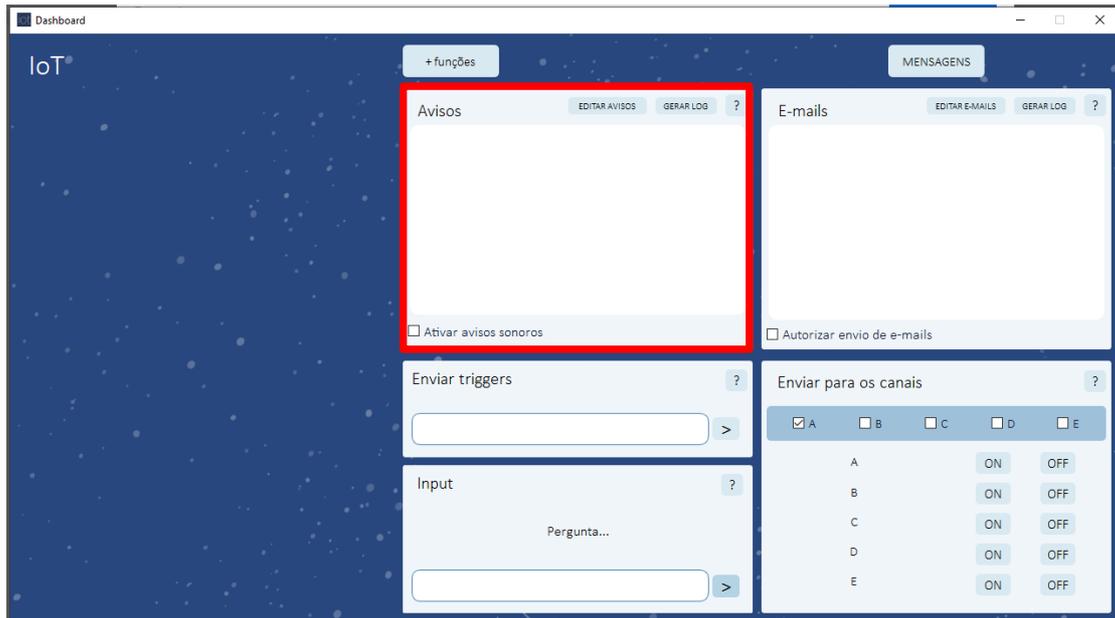
Agora, iremos aprender como gerar uma comunicação onde a informação parte do microcontrolador e é então enviada para a internet.



Para acessar a informação na internet utilizaremos novamente o *Dashboard*. Dessa vez, utilizaremos outro recurso disponível no programa: o recebimento de avisos. Para isso, iremos criar um sistema de avisos de luminosidade, onde o microcontrolador comunicará automaticamente como está a luminosidade do ambiente em que está. Como a aula 1, esta aula também apresentará como elaborar o código do microcontrolador com o Software Modelix System Starter.

## 2.0. Recurso de avisos no Dashboard

A ferramenta de avisos pode ser vista no topo direito do *Dashboard*. Esta utilidade está programada para mostrar linhas de texto conforme códigos de aviso são enviados ao *Dashboard*.



## 3.0. Projeto 2 - Sistema de aviso de luminosidade

Neste projeto iremos criar um programa que envia avisos ao Modelix MyIoT Dashboard conforme o microcontrolador percebe mudanças na luminosidade.

Como na aula 1, esta aula também decorre sobre o uso software de fluxograma Modelix System Starter. O software de fluxograma é recomendado para o entendimento da lógica de comunicação assim como a aprendizagem de programação.

### 3.0.1. O que você precisa ter para criar este projeto?

1. Um computador conectado à Internet. É necessário que o sistema operacional seja Windows.
2. Um microcontrolador Modelix 3.6 e cabo USB AB para conexão com o computador.
3. Um sensor de luminosidade. Note que o Modelix 3.6 já possui um embutido.

- Um software para programar o microcontrolador, podendo ser um IDE (Ambiente de Desenvolvimento Integrado) de C++ genérico ou um software educacional de Fluxograma como o Modelix System Starter (recomendado).
- Cadastro de usuário na Plataforma IoT da Modelix, assim como a realização do download do Pacote MyIoT (não requer instalação).
- Ter realizado a aula 1 do Pacote Modelix IoT.

Agora mãos à obra!

### **3.0.2. Programação do microcontrolador**

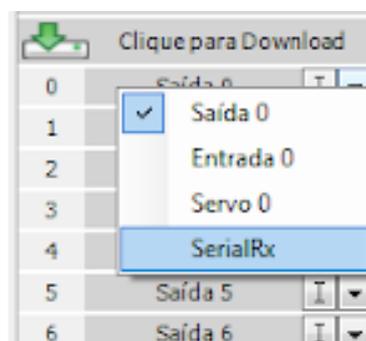
#### **3.0.2.1. Opção 1: Modelix System Starter**

Nesta seção vamos preparar o código do nosso projeto utilizando o Modelix System Starter, e então gravá-lo no Modelix 3.6.

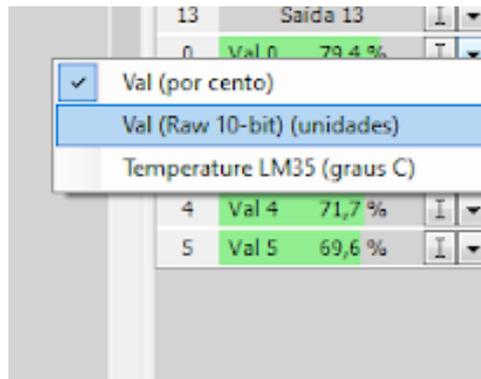
##### **3.0.2.1.1 - Análise dos intervalos do sensor de luz**

Para projetarmos nosso sensor de avisos de luz, precisamos primeiro entender como nosso sensor entende o ambiente. Nesta seção, iremos realizar a calibração do sensor para o ambiente que ele irá monitorar.

- Abra o Software Modelix System Starter. Conecte o controlador utilizando o cabo USB AB, e mude a “saída 0” para “SerialRx”.



2. No Modelix 3.6, a porta para o sensor de luz é a Val 0. Na linha desta porta, troque de Val (por cento) para Val (Raw 10-bit) (unidades). Se estiver usando este controlador, também lembre-se de ativar o sensor com o botão on-board descrito como "S. LUZ".



3. Para este fluxograma, iremos programar dois diferentes envios de sinal baseado no nível de luminosidade do ambiente em que o microcontrolador se encontra. Para isto, o primeiro passo é entender como o sensor está percebendo dois diferentes estados de luminosidade que definiremos como:

1. Luz acesa;
2. Luz apagada.

4. Primeiro iremos observar o estado de luz apagada. Deixe a luz desligada e o microcontrolador já no ambiente que o sensor de luminosidade será usado. Perceba o intervalo que o sensor lê nessa condição (linha Val 0), e anote-o (exemplo: com a luz apagada, o sensor oscila entre 40-120). **Atenção:** Esses valores não são os "certos", são somente exemplos! Anote os valores de acordo com o que você pode ler no seu programa.

5. Uma vez que só há dois estados, não é necessário observar o estado de luz acesa, pois sabemos que quando o valor estiver maior que o de luz desligada a luz estará acesa. Abaixo,

pode ser visto uma tabela que descreve a situação de cada estado de luz de acordo com as leituras do sensor.

| Condição                  | Intervalo Observado   |
|---------------------------|-----------------------|
| Intervalo 1 - Luz acesa   | $120 < \text{Val } 0$ |
| Intervalo 2 - Luz apagada | 40-120                |

6. Só que temos um problema. Uma vez que 120 é o limite superior *aproximado* do intervalo de luz apagada, podemos eventualmente ter valores como 125 ou 130 no mesmo estado (de luz apagada). Assim, para contornar o problema, adicionaremos uma margem de segurança de grandeza 50 neste limite superior. Assim, um exemplo de valores processados é:

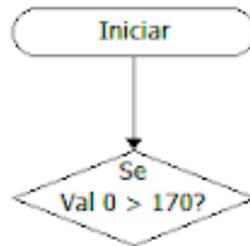
| Condição                  | Intervalo Observado   |
|---------------------------|-----------------------|
| Intervalo 1 - Luz acesa   | $170 < \text{Val } 0$ |
| Intervalo 2 - Luz apagada | 40-170                |

Observação: Há uma grande variação nos valores de luminosidade dependendo do ambiente em que o microcontrolador está. Por isso, não se esqueça de realizar esta etapa do projeto já no ambiente que você deseja monitorar.

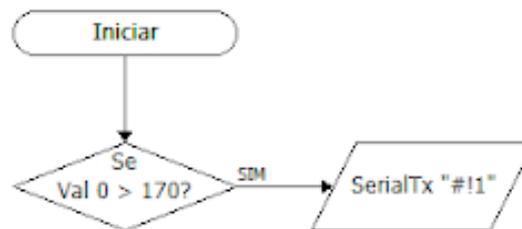
7. Pronto! Tendo finalizado a calibração dos intervalos, vamos começar a programação. Não desconecte o microcontrolador do computador ou do programa.

#### **3.0.2.1.2 - Fluxograma**

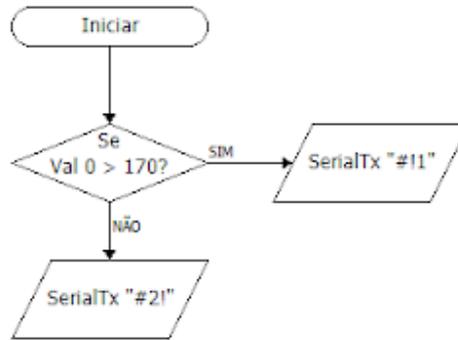
1. Insira um bloco iniciar, seguido de um bloco de decisão. Nesta decisão, iremos restringir o sistema para que, caso o sensor identifique a luz acesa, ele envie um sinal específico. Observando a tabela de calibração do item 6 da seção 3.0.2.1, sabemos que a luz estará acesa se  $Val\ 0 > 170$  (você pode ter números diferentes baseado na sua calibração, então utilize o seu valor observado). Desta forma, vamos inserir tal restrição ao bloco de decisão.



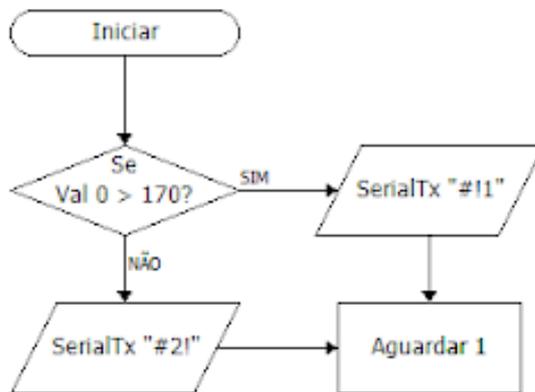
2. Agora iremos configurar o sinal que será enviado caso a luminosidade esteja acima de 170 (ou o valor que você inseriu). Insira uma caixa de Transmissão Serial, e o comando “#!1”. Conecte-o à opção SIM do bloco de decisão utilizando a ferramenta Linha.



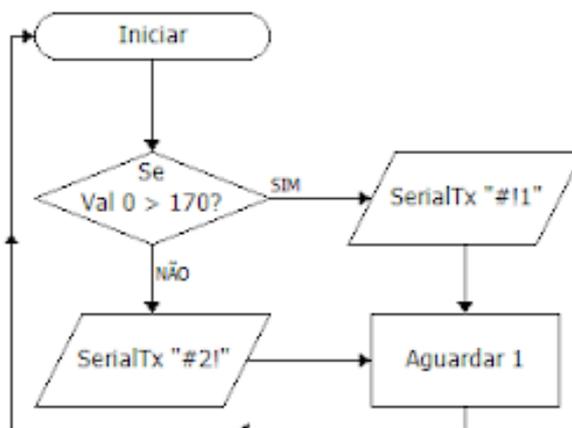
3. Agora, conectado à opção NÃO do bloco de decisão, insira outro bloco SerialRx. Enviaremos o sinal “#!2” para o estado de luz desligada (que é, em nosso exemplo,  $Val\ 0 < 170$ ).



4. Por fim, vamos adicionar um bloco aguardar de 1 segundo após os dois blocos SerialRx. Ao invés de inserir dois blocos, utilize a ferramenta de linhas para conectar ambos blocos SerialRx no mesmo bloco aguardar.



5. Termine o programa conectando o bloco aguardar com o bloco iniciar.

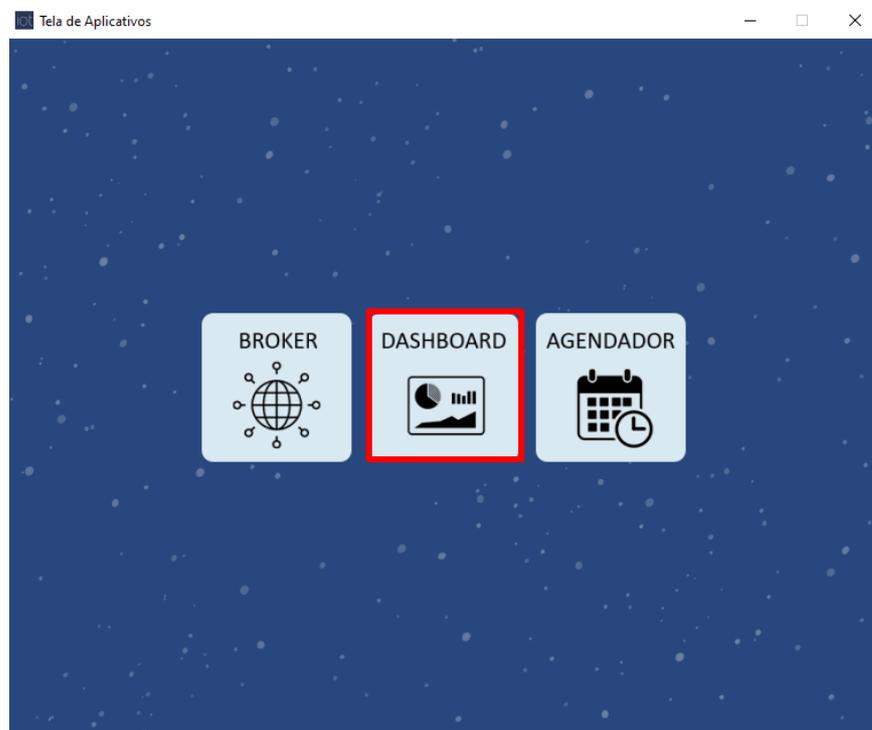


### 3.0.4. Configurando os avisos

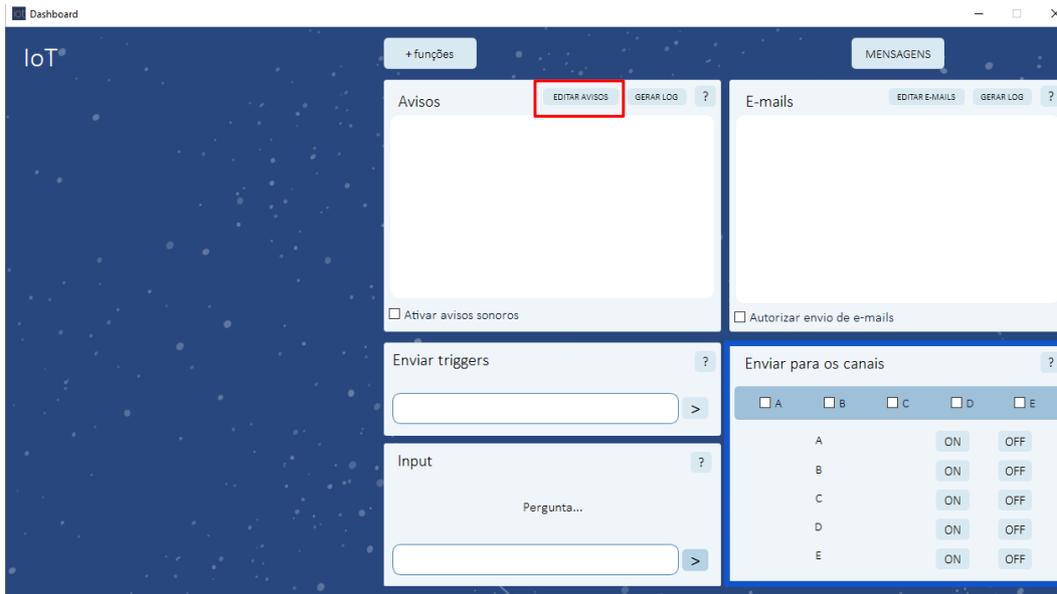
Nesta seção, iremos configurar duas mensagens para serem enviadas para a internet baseado nos comandos que programamos para serem enviados na seção anterior. Estes foram:

| Condição                  | Comando enviado pelo controlador |
|---------------------------|----------------------------------|
| Intervalo 1 - Luz acesa   | #!1                              |
| Intervalo 2 - Luz apagada | #!2                              |

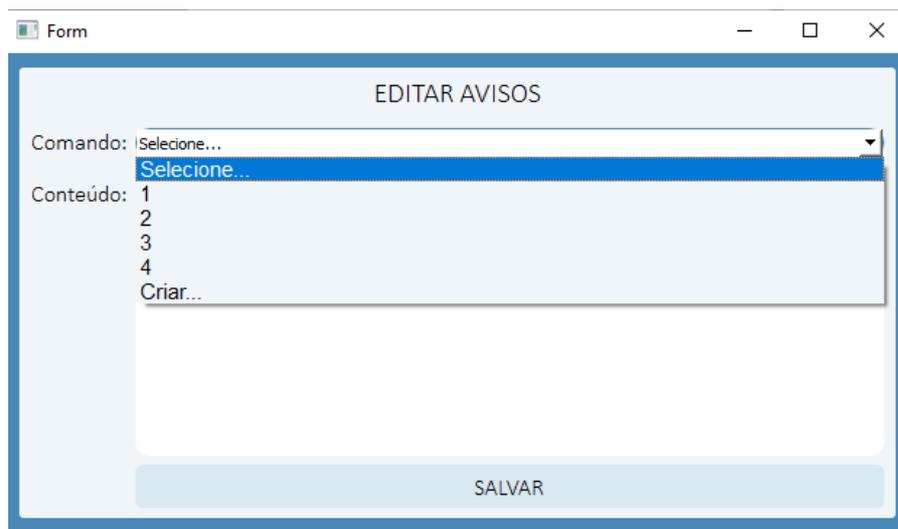
1. Abra o Pacote MyIoT,, realize o login e clique em Dashboard.

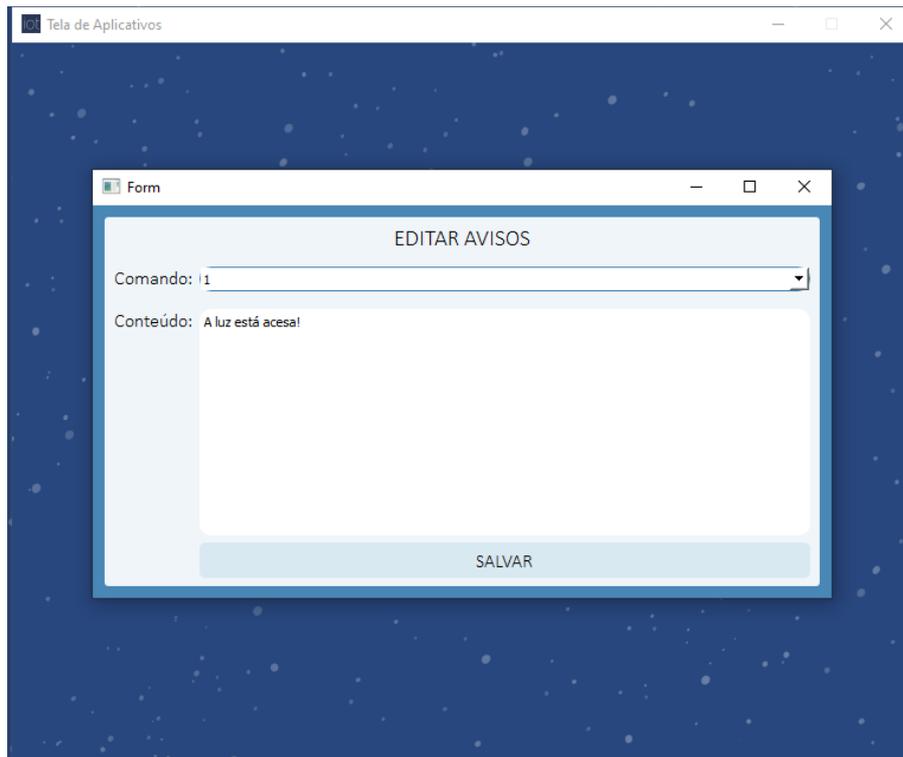


2. Clique no botão "Editar Avisos".



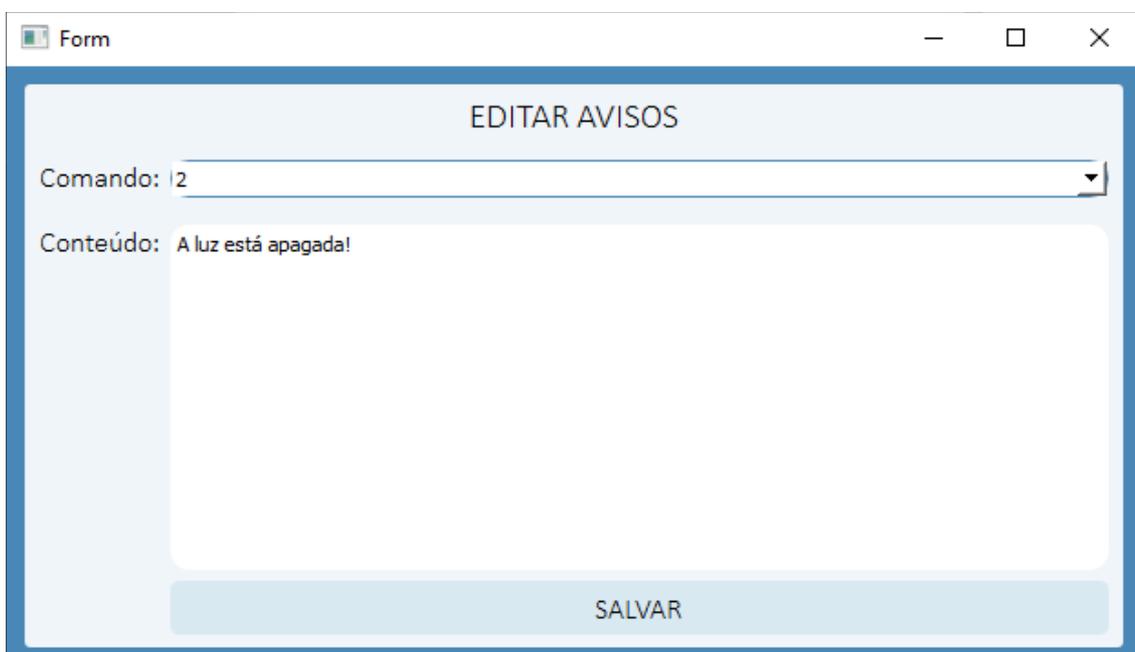
3. Na janela que abrir, clique na flecha para baixo da linha “Comando”. Selecione #!1, e preencha em “Conteúdo” o que você deseja que aquele comando avise. Para o aviso que #!1, queremos comunicar que a luz está acesa. Logo:





Após editar, clique em salvar.

4. Repita o processo para o comando “#!2”, editando o conteúdo conforme o que aquele comando deve comunicar. Consulte o item 1 desta seção para ter certeza que está preenchendo corretamente. Não se esqueça de salvar após cada alteração nos conteúdos.



5. Pronto! Os avisos já estão sincronizados aos comandos enviados pelo microcontrolador em cada situação.

### **3.0.5. Conectando o microcontrolador com o Broker**

1. Com o microcontrolador conectado ao computador, feche o Modelix System Starter ou o Modelix *Flasher*. Na tela principal do Pacote MyIoT, clique em “Broker”.
2. Clique nas setas no centro da janela, e seu microcontrolador será listado automaticamente.
3. Clique em Login. Um aviso de conectado indicará que o sistema está funcionando! Não desconecte o Modelix 3.6 do computador ou feche o aplicativo Broker, ou o sistema não funcionará.

### **3.0.6. Vendo os avisos no Dashboard:**

1. Agora que o sistema já está funcionando, só nos resta visualizá-lo de qualquer lugar no mundo. Na tela principal do Pacote MyIoT, clique em Dashboard.
2. Na caixa do topo direito da página inicial do Dashboard, descrita como “avisos”, você poderá ver os avisos que configuramos no Broker! Agora você tem um sistema de avisos em tempo real da luminosidade do ambiente em que você instalou seu microcontrolador.

### **3.0. Resumo e próximas iterações:**

Esta aula teve como objetivo ensinar como enviar sinais do microcontrolador e utilizar a ferramenta de avisos do Dashboard. Na janela de edição de avisos, perceba que há a opção “Criar” do menu de seleção da linha “Comando”. Nesta opção, você pode criar quantos e quais comandos quiser, com conteúdos específicos para cada. Para o aviso criado ser visto na caixa “Avisos” do

Dashboard, somente é necessário configurar o microcontrolador (conectado ao Broker) para que este comunique o comando configurado por você.