

WWW.

MYIOT
.SPACE

STARTUP

GUIA COMPLETO DE REFERÊNCIA

ÍNDICE

| | |
|--|-----------|
| 1. Programando dispositivos de Internet das Coisas..... | 3 |
| 1.1. Meios de criar códigos..... | 3 |
| 1.2. Criando um microcontrolador IoT..... | 4 |
| 1.3. Conceito de <i>Triggers</i> | 5 |
| 1.4. Como programar um <i>Trigger</i> em um microcontrolador?..... | 14 |
| 2. Aplicativos do Pacote..... | 15 |
| 2.1. Dashboard..... | 15 |
| 2.2.1. Envio de E-mails..... | 16 |
| 2.2.2. Parâmetros/medidas..... | 22 |
| 2.2.3. Avisos..... | 23 |
| 2.2.4. Painel de Controle..... | 26 |
| 2.2.5. Enviar para os canais..... | 27 |
| 2.2.6. Enviar Triggers..... | 28 |
| 2.2.7. Input..... | 28 |
| 2.2. Broker..... | 29 |
| 2.3. Flasher..... | 33 |
| 2.4. Agendador..... | 38 |
| 3. Mapeamento de sensores e atuadores utilizados..... | 39 |
| 2.1. LDR | 39 |
| 2.2. Sensor ultrassônico..... | 39 |
| 2.3. Sensor Infravermelho..... | 40 |
| 2.4. DHT11..... | 40 |
| 2.5. Botão..... | 41 |
| 2.6. Matriz de LED..... | 41 |
| 2.7. LCD..... | 42 |

| | |
|--|-----------|
| 2.8. Teclado Numérico..... | 43 |
| 2.9. PIR..... | 43 |
| 2.10. Motor elétrico..... | 44 |
| 2.11. Servomotor..... | 45 |
| 2.12. Joystick..... | 46 |
| 2.13. Relé..... | 46 |
| 4. Mapeamento dos microcontroladores..... | 47 |
| 4.2. ESP 32..... | 47 |
| 4.3. ESP 01..... | 48 |
| 2.4. ESP 8266 12..... | 49 |
| 2.5. ESP CAM..... | 50 |
| 2.6. Arduino Uno..... | 51 |
| 2.7. Arduino Nano..... | 53 |
| 2.7. Duemilanove..... | 54 |
| 5. Falhas..... | 55 |
| 5.1. Falha ao encontrar microcontrolador..... | 55 |

1. PROGRAMANDO DISPOSITIVOS DE INTERNET DAS COISAS

1.1 MEIOS DE CRIAR CÓDIGOS

Com IoT, cada dispositivo a ser utilizado (chamados de microcontroladores) devem ser programados especificamente para o projeto em questão. Há três maneiras de programá-los:

- **IDE de programação** - IDE (Integrated Development Environment) é um termo em inglês que significa “Ambiente Integrado de Desenvolvimento”. A ideia é de uma plataforma onde o usuário compõe de um código escrito para especificar funcionalidades para um microcontrolador de acordo com sensores, atuadores ou comunicação com a internet. Esta maneira de programar é adequada somente para usuários avançados, uma vez que é necessário conhecer a linguagem de programação C++.

- **MyIoT Flasher** - O MyIoT Flasher é um software prático que pode ser utilizado tanto por usuários avançados como iniciantes. O programa disponibiliza uma biblioteca com códigos prontos, estes que são inseridos e configurados automaticamente em um microcontrolador conectado ao computador. Cada código possui uma descrição

de seu funcionamento assim como um tutorial de uso, com suas particularidades, possíveis aplicações e conexões elétricas. Para mais informações deste software, acesse a seção 2.5 - *FLASHER*.

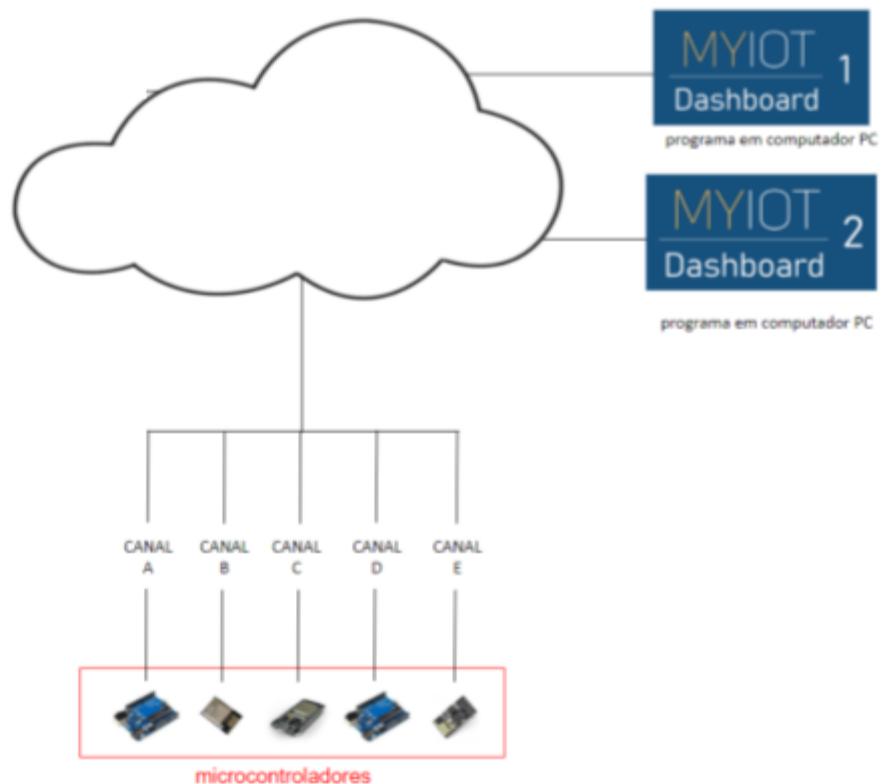
1.2 CRIANDO UM MICROCONTROLADOR IOT

Pensando no contexto de programação para Internet das Coisas, há dois passos necessários para criar um “microcontrolador IoT”:

1. Definir instruções de como o microcontrolador deve se comportar baseado nos sensores e atuadores conectados fisicamente à ele;
2. Definir instruções de como o microcontrolador deve se comportar com comandos *recebidos* da “nuvem”, assim como delinear que acontecimentos (pensando em sensores e atuadores) farão com que comandos sejam *enviados* para “nuvem”. Tais comandos são chamados de *triggers*, que serão explicados mais à fundo na próxima seção.

Um projeto IoT permite que diferentes microcontroladores comuniquem-se através da Internet, porém é necessário programar cada um individualmente para que estes interajam da maneira desejada.

Cada microcontrolador se comunica com a internet através de “ruas”, estas que chamamos de canais. Podemos nos comunicar com os canais a partir de uma “central de comunicação”, que chamamos de Dashboard (mais informações na seção 2.2 - Dashboard), assim como através dos microcontroladores em si. De forma geral, o funcionamento do sistema pode ser visualizado da seguinte forma:



1.3 CONCEITO DE *TRIGGERS*

Triggers, em inglês, é gatilho. Este conceito em programação corresponde a uma letra ou palavra que provoca o disparo de uma ação ou sequência de ações em um código. Exemplos são **B**, **b**, **#email**, **#nome_da_rotina**.

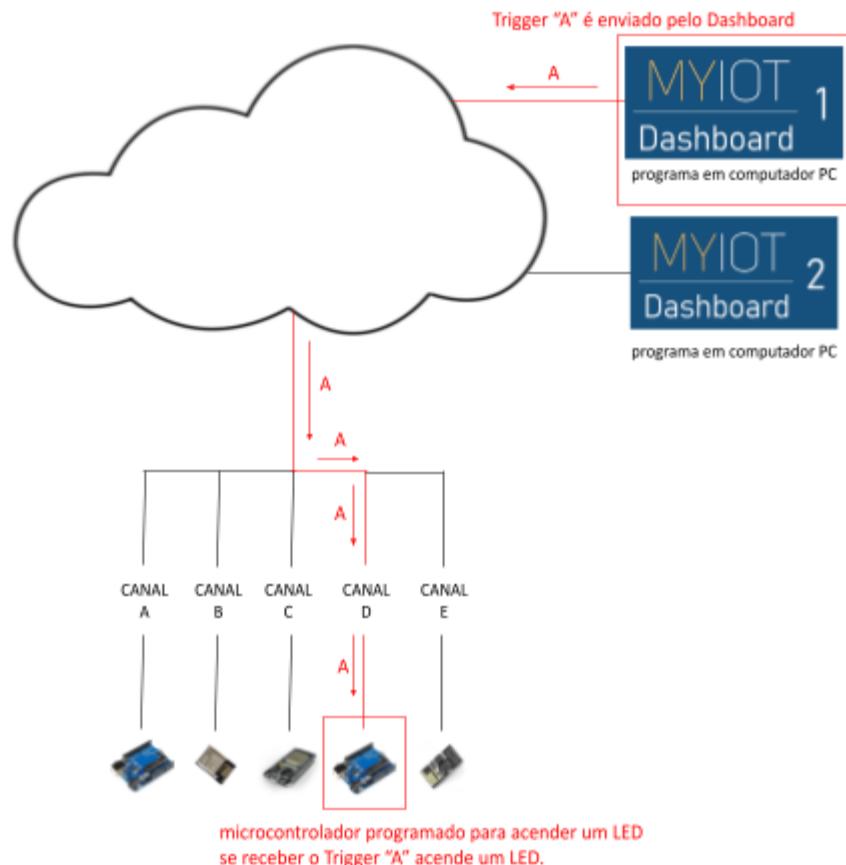
Um *trigger* pode originar na “nuvem”, ou seja, em um aplicativo conectado à internet como o *Dashboard*; ou pode originar em um microcontrolador, enviando o sinal para a “nuvem”. Em ambos os casos, o *trigger* ocasiona uma ação ou sequência de ações no *Dashboard* ou no microcontrolador. Esta ferramenta possibilita a comunicação entre microcontroladores, assim como a criação de projetos de alta funcionalidade e complexidade.

Desta forma, é essencial o entendimento deste conceito. O mecanismo é utilizado na maioria dos projetos IoT, e a maestria de seu funcionamento acarreta um grande aumento nas possibilidades de um projeto.

Em resumo, as duas formas de utilizar triggers são:

- 1. Enviando um trigger diretamente do Dashboard** - Esta opção corresponde às ferramentas de “Enviar triggers” e “Enviar para os canais” do Dashboard. Nele, é possível enviar os triggers “A” - “E”, na forma minúscula e maiúscula, assim como qualquer *trigger* padrão do sistema. Os *triggers* serão especificados à fundo nesta seção.

Exemplo:



2. Programando microcontroladores para enviarem triggers - Diferente da primeira forma de utilizar o *trigger*, enviar um *trigger* diretamente de um microcontrolador envolve conhecer os tipos de *trigger* e suas possibilidades. Uma vez que utilizando o Flasher não é necessário pensar na lógica do código, não é necessário aplicar conhecimentos sobre *triggers* nesta maneira de programar.

Assim sendo, existem os seguintes tipos de *trigger* para serem enviados **originando de microcontroladores**:

1. #email

2. **#aviso**
3. **#mensagem**
4. **#voz**
5. **#rotina**
6. **#input**
7. **#intranet**
8. **#var**
9. **#output**

1. **#email número**

Este trigger faz com que e-mails sejam enviados automaticamente pelo Dashboard. O envio do *trigger* deve sempre estar acompanhado de um número, este que está atrelado a um e-mail específico pré-configurado no Dashboard pelo usuário. O conteúdo, destinatário e endereço de e-mail de origem devem ser pré-editados na ferramenta e-mails do Dashboard. O e-mail de origem deve ser uma conta gmail com a opção de segurança ativada.

Exemplo:

1. **#email 1**

2. E-mail configurado como 1 no Dashboard, com destinatário MyIoT@teste.com.br e conteúdo “Este foi um e-mail enviado por IoT!”, é enviado.
3. Toda vez que este *trigger* com o número 1 for ativado, este mesmo e-mail será enviado. Isto acontecerá até o usuário editá-lo no Dashboard.

2. #aviso número

Este trigger faz com que avisos de texto sejam apresentados no Dashboard. Igual o *trigger* de e-mail, o *trigger* de aviso deve ser acompanhado por um número que está atrelado a um aviso específico pré-configurado no Dashboard. O texto do aviso será apresentado na janela central da ferramenta “avisos”, na parte central da tela. Este virá com data e horário de envio assim como sua origem (pois avisos podem ser enviados entre colaboradores em contas diferentes).

Exemplo:

1. #aviso 1
2. Aviso configurado como 1 no Dashboard, “Bom dia usuário！”, é enviado.
3. Toda vez que este *trigger* com o número 1 for enviado, este mesmo aviso será apresentado. Isto acontecerá até o usuário editá-lo no Dashboard.

3. #mensagem texto para ser mostrado no display e aviso do Dashboard

Este *trigger* faz com que avisos de texto sejam enviados à tela de avisos do Dashboard e apresentados por microcontroladores conectados à internet com matrizes de LED ou displays LCD acoplados. Estes microcontroladores devem utilizar flashes correspondentes ao atuador que será utilizado (Matriz de LED ou LCD). Os flashes podem ser inseridos através do aplicativo *Flasher*.

Exemplo:

1. #mensagem Boa tarde!

4. #voz texto que será lido no aviso sonoro

Este trigger faz com que avisos sonoros sejam reproduzidos no Dashboard.

Exemplo:

1. #voz Olá usuário!
2. Se o volume do computador estiver ligado e o *Dashboard* aberto, a frase “Olá usuário!” poderá ser ouvida.

5. #rotina nome da rotina a ser iniciada

Este trigger faz com que rotinas configuradas no aplicativo Programar sejam iniciadas. Rotinas podem conter um *trigger* ou uma sequência de *triggers*, estes que buscam realizar ações específicas.

Exemplo:

1. #rotina aviso_matinal!
 2. A rotina “aviso_matinal” é reproduzida. Esta contém os *triggers*:
-

```
#aviso 1  
  
#voz Um ótimo dia para você usuário!
```

3. O aviso 1, configurado como “Bom dia!” é apresentado no quadro de avisos do Dashboard. Em seguida, poderá ser ouvida a mensagem de voz “Um ótimo dia para você usuário!”

6. #input pergunta_a_ser_feita

Este *trigger* faz com que uma pergunta seja feita ao *Dashboard* no campo **Input**. Logo abaixo da pergunta, um campo de preenchimento tem o objetivo de gravar a resposta do usuário na nuvem. Esta resposta gravada pode ocasionar ações se tal for programado em um microcontrolador ou no aplicativo Programar. Para ambas opções, o código deve ter uma reação para uma leitura de texto específica esperada para a resposta.

Exemplo:

1. #input Está escuro?
2. O usuário digita a resposta “sim”, no campo input.

3. Um microcontrolador conectado à internet programado para ativar uma lâmpada através de um relé caso a resposta do input seja sim (if input = sim) ativa a lâmpada.

7. #intranet texto_a_ser_enviado_para_a_nuvem

Este *trigger* tem a mesma funcionalidade do *trigger* #input, porém sem o passo de uma pergunta ser feita no Dashboard. Quando o usuário digita o *trigger*, o texto que segue o “#intranet” é salvo na nuvem, igual a resposta de um #input. Também da mesma maneira do *trigger* apresentado anteriormente, o valor “intranet” gravado pode ocasionar ações se tal for programado em um microcontrolador ou no aplicativo Programar. Para ambas opções, o código deve ter uma reação para uma leitura de texto específica da intranet.

Exemplo:

1. #intranet ligar
2. Um microcontrolador conectado à internet programado para ativar uma lâmpada através de um relé caso o valor de intranet seja “ligar” (if intranet = ligar) ativa a lâmpada.

8. #var número_da_variávelletra_do_canal = texto_ou_número

Este *trigger* envia um **valor numérico ou escrito** para uma das **8 variáveis** atreladas a **cada canal** e consequentemente ao microcontrolador.

Exemplo:

1. #var 2A = luminoso
2. Um microcontrolador conectado à internet no canal A recebe como valor de sua variável 2 “luminoso”.
3. Um código criado no aplicativo programar define que se var2A = luminoso, o *trigger* d é enviado ao canal, este em que o microcontrolador que está configurado para desligar uma lâmpada conectada a um relé com o recebimento deste comando

.

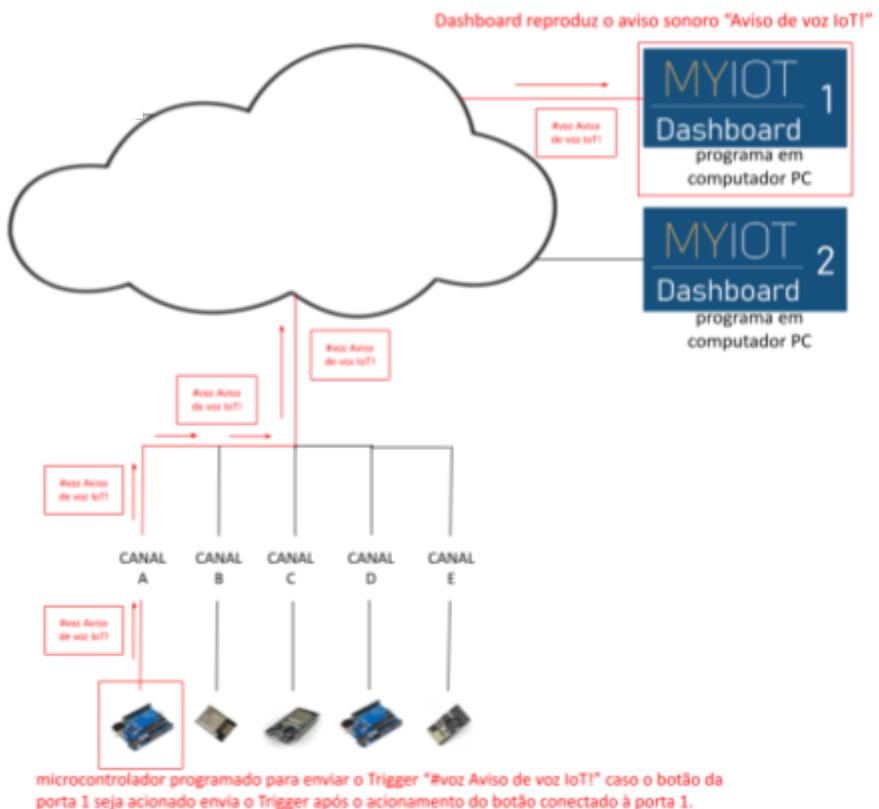
9. #output letra do canal = letra de A-E

Este trigger envia uma letra de A-E, maiúscula ou minúscula, para um canal específico atrelado a um microcontrolador. O *trigger* precisa de duas variáveis a serem especificadas pelo usuário: **qual letra a ser enviada** (A-E, maiúscula ou minúscula) e em **qual canal**.

Exemplo:

1. #output A = c
2. A letra “c” é enviada para o canal A.
3. Um microcontrolador conectado à internet configurado para ligar um motor se a letra “c” for enviada à ele liga o motor.

Exemplo de um sistema de triggers em um microcontrolador:



1.4 COMO PROGRAMAR UM TRIGGER EM UM MICROCONTROLADOR?

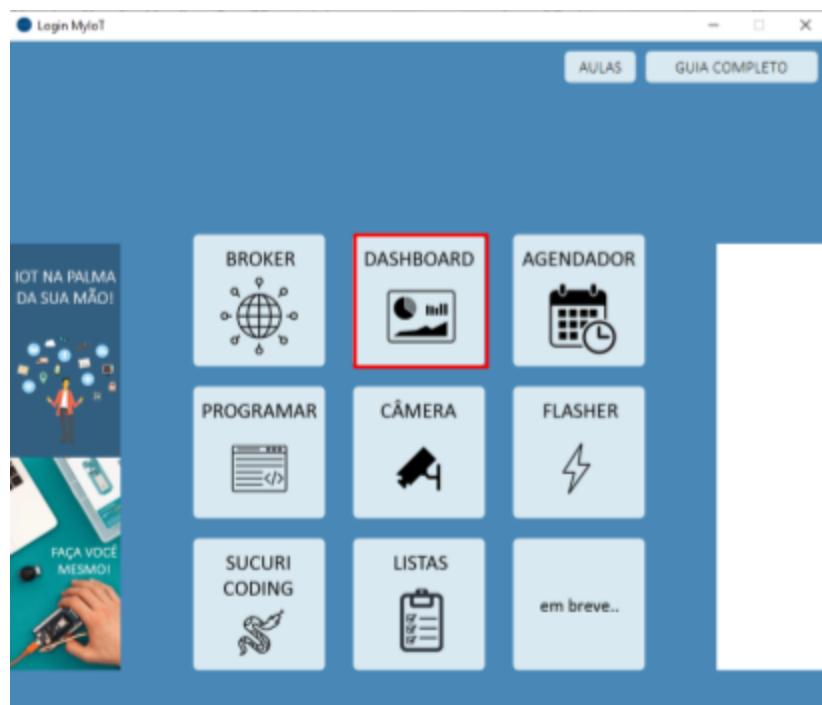
➤ Programando um Trigger em C++ por um IDE genérico

```
println "#voz Aviso de voz IoT!"
```

2. APlicativos do pacote MyIoT

2.1 - DASHBOARD

O Dashboard é a central de comunicação do sistema MyIoT. O programa consiste em uma janela de computador ou celular que permite com que o usuário visualize diversas informações sobre dispositivos IoT. Além disso, o software possibilita a interação do usuário com projetos a partir do envio de *triggers* e edição de parâmetros em tempo real. O programa vem com diversas ferramentas já embutidas, das quais é possível criar e gerenciar diferentes tipos de projetos. Para abrir o aplicativo, clique em **Dashboard** no menu principal do pacote MyIoT.



A janela principal do aplicativo é:

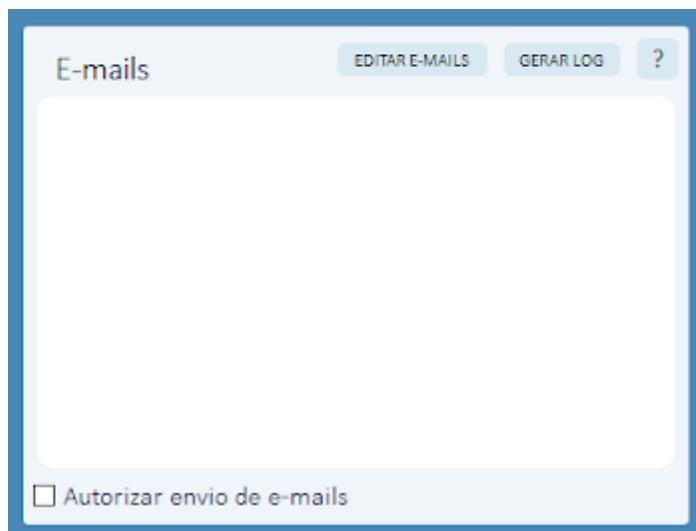
The screenshot displays the MYIOT Dashboard application window. The top navigation bar includes tabs for 'Dashboard' (selected), 'TELA DE AVISOS', 'GUIA DE USO', and 'COLABORAÇÃO'. The main interface is divided into several sections:

- Parâmetros e Medições:** A grid showing values for variables Var0 through Var7 across channels A, B, C, D, and E. Values are mostly 0, except for Var1, Var4, and Var5 which have a value of 1.
- Avisos:** A section for managing alerts with buttons for 'EDITAR AVISOS' and 'GERAR LOG'.
- E-mails:** A section for managing emails with buttons for 'EDITAR E-MAILS' and 'GERAR LOG'.
- Painel de Controle:** A control panel showing the last trigger sent to the cloud and the current signal status on channels A through E.
- Enviar triggers:** A section for sending triggers with a text input field and a '>' button.
- Enviar para os canais:** A section for sending messages to channels A through E, with 'ON' and 'OFF' buttons for each channel.
- Input:** A section for inputting questions with a text input field and a '>' button.

Suas ferramentas são:

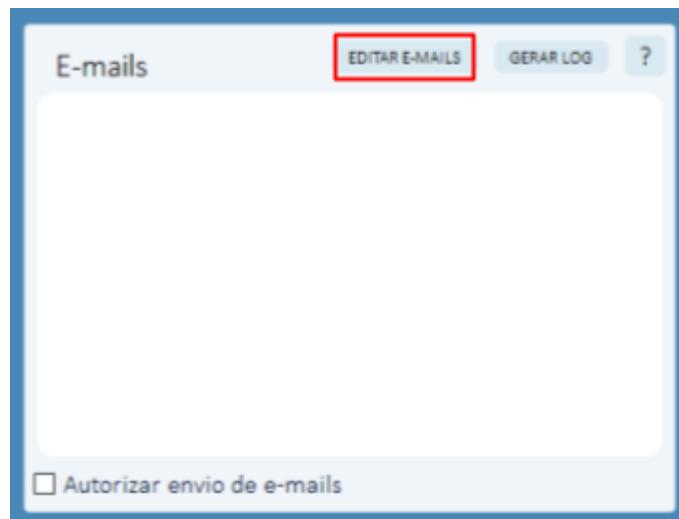
2.2.1. ENVIO DE E-MAILS

A funcionalidade de e-mails consiste no envio automático de um e-mail pré-programado caso o comando específico “#email **número**” seja enviado ao Dashboard. O número indica qual e-mail a ser enviado, uma vez que múltiplos e-mails podem ser pré-programados.

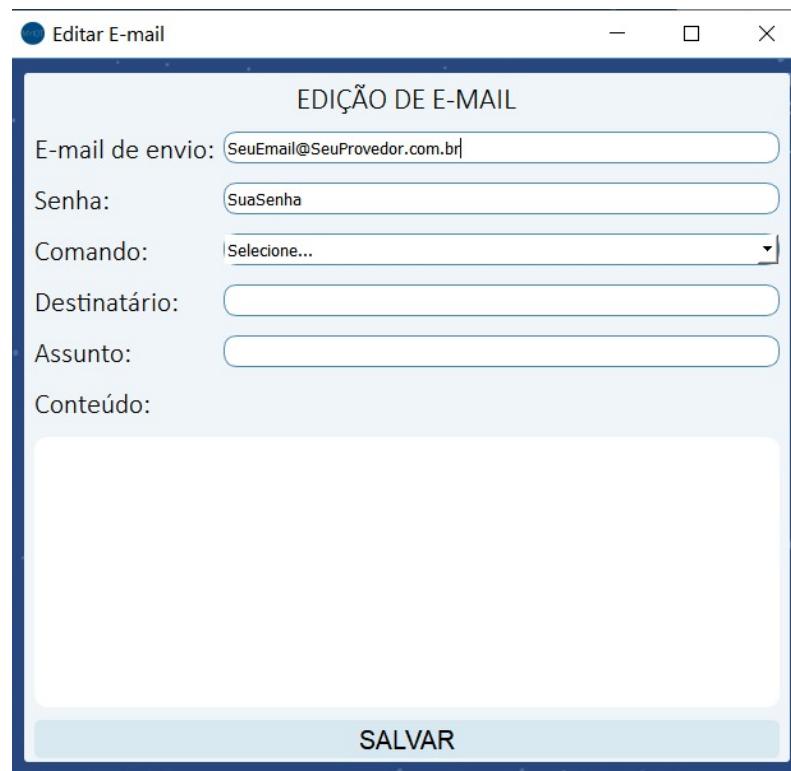


PRÉ-PROGRAMANDO OS E-MAILS

Entre no modo de edição de emails clicando em “Editar E-mail”.



Na janela que abrirá, as variáveis são as seguintes:



E-mail de envio: Endereço de email do qual os e-mails serão enviados. É necessário que esta seja uma conta Gmail com o recurso de segurança ativado. Para ativar este recurso de uma conta Gmail, siga os seguintes passos:

1. Entre na sua conta Google (<https://account.google.com/>). Caso não tenha uma, clique em “crie uma conta” e volte ao mesmo link após completar o cadastro.
2. Vá em “Segurança”, desça até “Acesso a app menos seguro” e clique em “Ativar acesso”.

The screenshot shows the Google Account security settings. On the left, a sidebar lists options: Início, Informações pessoais, Dados e privacidade, Segurança (selected), Pessoas e compartilhamento, Pagamentos e assinaturas, and Sobre. The main area displays connected devices: LG K10 LTE (Brasil - 16:15) and iPhone (Brasil - 15:18). Below these are links for 'Encontrar um dispositivo perdido' and 'Gerenciar dispositivos'. To the right, a section titled 'Acesso a app menos seguro' (Access to less secure apps) is shown, stating that less secure logins have been blocked. It shows that access is currently 'Desativado' (Disabled). A red box highlights the 'Ativar acesso (não recomendado)' (Enable access (not recommended)) button. Other third-party apps listed include academia.edu, adidas Running App Run Tracker, and DocHub - PDF Sign & Edit.

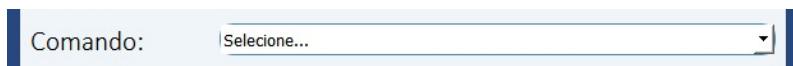
3. Na janela que abrir, selecione a opção “Permitir aplicativos menos seguros” para que esta esteja escrita “ATIVADA”.

The screenshot shows the 'Acesso a app menos seguro' (Access to less secure apps) settings page. At the top, there's a link to go back. The main content area contains a paragraph explaining that some apps and devices use less secure login technologies, making accounts vulnerable. It suggests disabling access for these apps or enabling it if users opt to use them despite risks. A blue link 'Saiba mais' (Learn more) is provided. Below this, a button labeled 'Permitir aplicativos menos seguros: ATIVADA' (Allow less secure apps: ACTIVATED) has a red box around it, and a red box also surrounds the checked checkbox icon to its right.

Senha: Senha do endereço de e-mail do qual os e-mails serão enviados.

Comando: Número correspondente ao e-mail em questão. Clique na seta para baixo e escolha um dos números listados, gravando em seguida um e-mail para este comando específico. É possível ter quantos e-mails gravados desejar, com assuntos e destinatários

diferentes. Cada email será gravado com um número único para aquele e-mail (1, 2, 3..), podendo ser chamado no *trigger* como “#email 1”.



Destinatário: Endereço de e-mail para qual o e-mail será enviado.

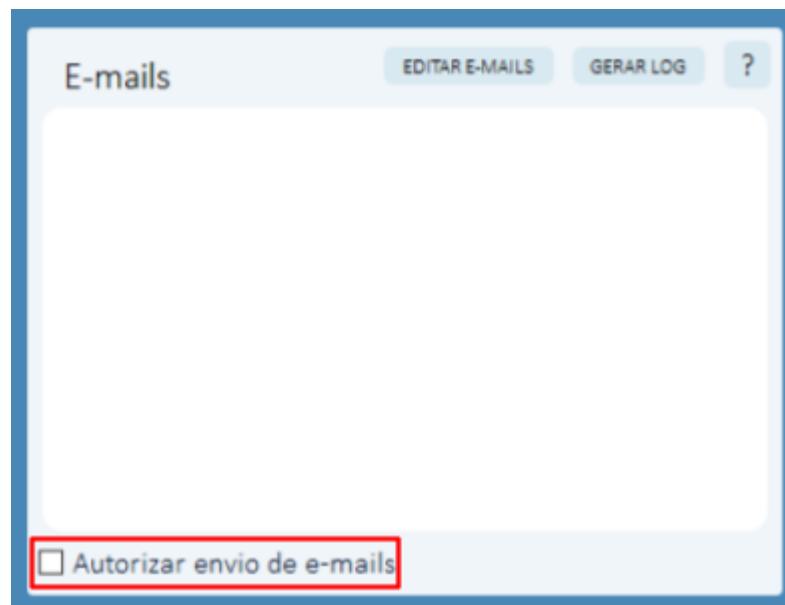
Assunto: Assunto do e-mail.

Conteúdo: Conteúdo do e-mail.

Terminando a edição de e-mails, clique em salvar e a janela fechará automaticamente.

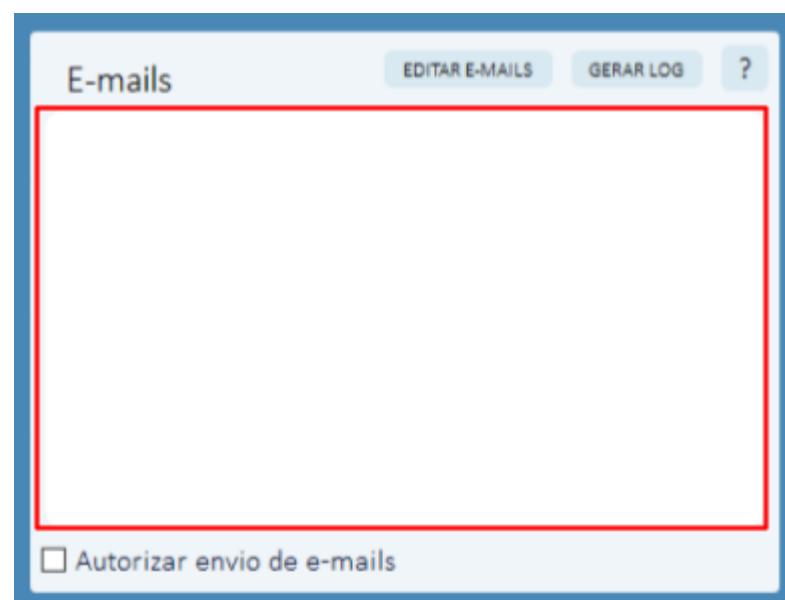
AUTORIZANDO O ENVIO DE E-MAILS

Marque ou desmarque a caixa da linha “Autorizar envio de E-mails” para permitir ou bloquear o envio de emails pelo Dashboard. Esta ferramenta é específica para o Dashboard em que é utilizada, uma vez que é possível rodar mais de um Dashboard com a mesma conta MyIoT.



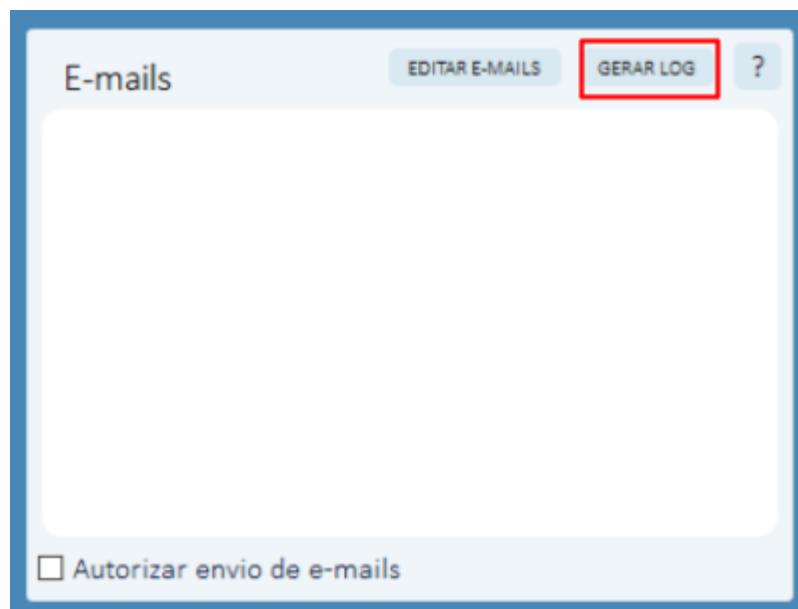
VISUALIZAÇÃO DE E-MAILS ENVIADOS

Nesta janela aparecem os e-mails enviados pelo Dashboard aberto. Uma vez que é possível ter múltiplos Dashboard abertos com a mesma conta, os e-mails enviados só aparecem no Dashboard específico por qual estes foram configurados e então enviados.



GERAR LOG

O botão Gerar Log cria um documento de texto com todos os e-mails enviados a partir do computador em questão, assim como seus conteúdos.



2.2.2. PARÂMETROS E MEDIÇÕES

A janela de parâmetros e medições apresenta dados atualizados fornecidos por sensores conectados a um microcontrolador conectado à internet, independente da localização.

| Parâmetros e Medições | | | | |
|-----------------------|-----------|-----------|-----------|---|
| Escolha o canal: | | | | |
| A | B | C | D | E |
| Var0 0 | Var1 0 | Var2 0 | Var3 0 | |
| Var4 0 | Var5 0 | Var6 0 | Var7 0 | |

A ferramenta apresenta o status em tempo real de até 8 variáveis em cada canal, correspondentes a portas de microcontroladores específicos.

2.2.3. AVISOS

A janela de avisos mostra os *triggers* de avisos enviados ao Dashboard e reproduz *triggers* de avisos sonoros (que são leituras de texto com a voz do computador).



VISUALIZAÇÃO DE AVISOS

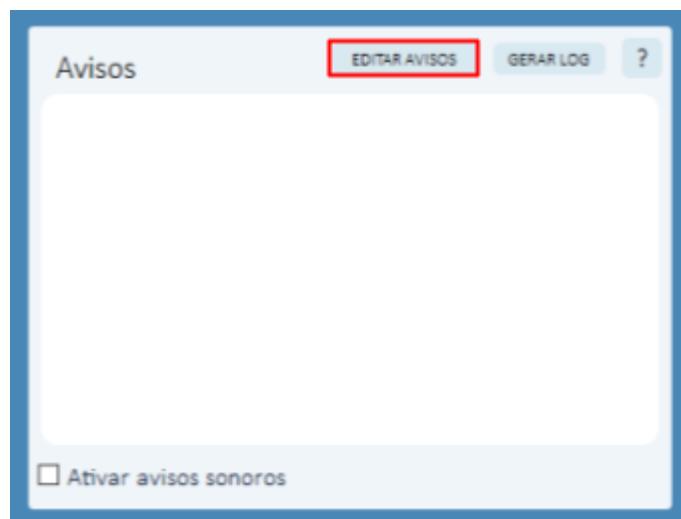
Mostra os avisos de texto baseado em acontecimentos programados em microcontroladores.

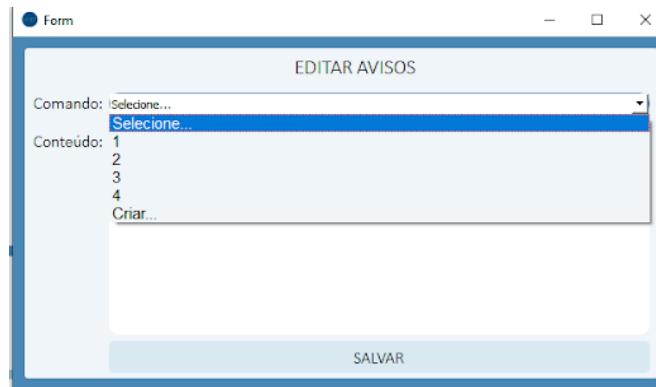
Trigger correspondente: "#aviso texto".



EDIÇÃO DE AVISOS

Para editar os avisos pré-configurados para serem acionados com o *trigger #aviso*, clique em **Editar Avisos**.

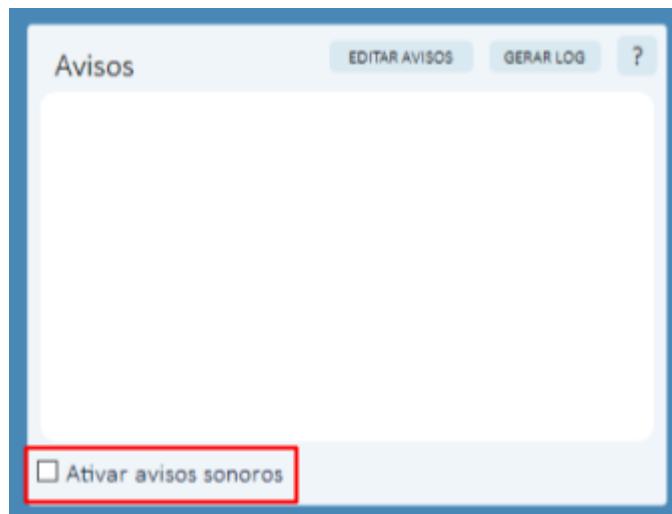




Em “comando” escolha um número, e escreva o aviso pré-configurado que deseja para aquele número. Não se esqueça de clicar em salvar após cada edição de conteúdo.

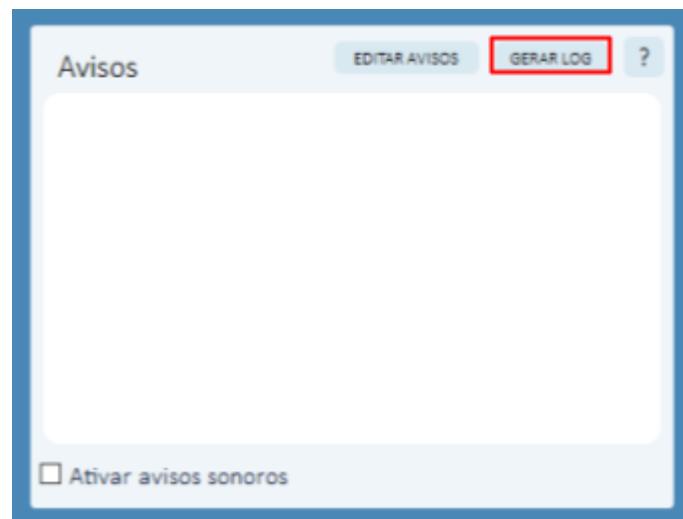
ATIVANDO AVISOS SONOROS

Marque e desmarque a caixa **Ligar avisos sonoros?** para habilitar a ferramenta. O Dashboard reproduz avisos de áudio caso o *trigger #voz* seguido de texto seja enviado à ele. **Exemplo:** “#voz Aviso de voz IoT!”.



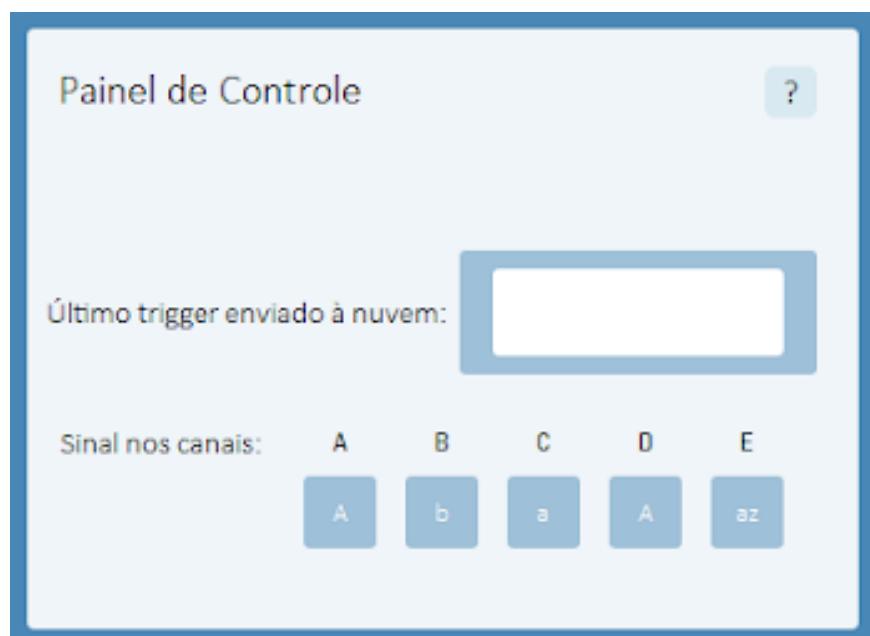
GERAR LOG

O botão Gerar Log baixa um arquivo de texto com o histórico de avisos recentes.



2.2.4. PAINEL DE CONTROLE

A janela Painel de Controle mostra os *triggers* “correndo” através dos canais e no Dashboard em questão.

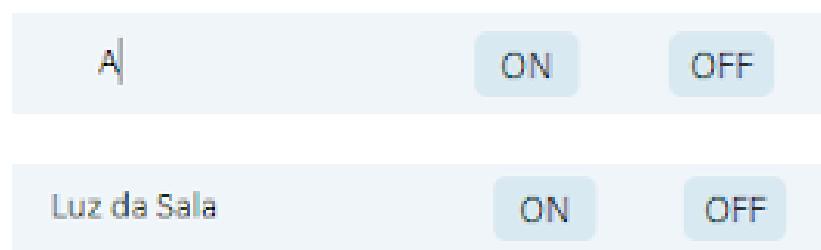


2.2.5. ENVIAR PARA OS CANAIS

Esta janela permite com que *triggers* possam ser enviados para canais específicos a serem escolhidos pelo usuário. Uma vez que cada canal corresponde a um microcontrolador específico, esta ferramenta permite uma comunicação direta com um microcontrolador programado para reagir a estes comandos.



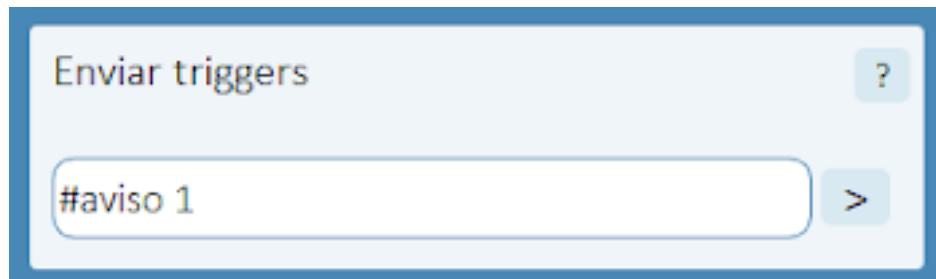
Cada botão envia as letras (A-E [ON]; a-e [off]) como um *trigger* para o microcontrolador. Ao clicar nas letras à esquerda, é possível editar o texto para uma descrição específica, como “Luz da Sala”.



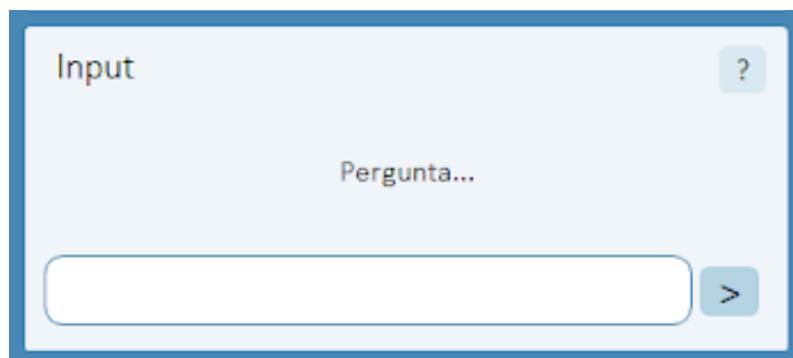
Mesmo mudando a descrição, o *trigger* enviado será a letra original daquela coluna em maiúsculo e minúsculo. Teste e visualize o valor dos botões no painel de controle. No exemplo acima, ao clicar em “ON” na linha **Luz da Sala** o *trigger* enviado será “A”.

2.2.6. ENVIAR TRIGGERS

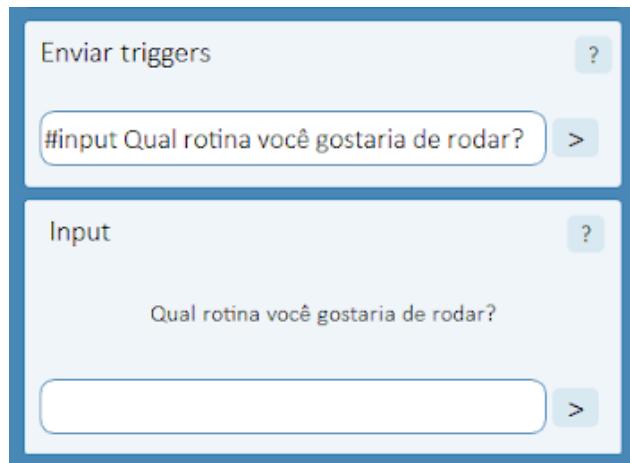
A janela **Enviar triggers** permite que qualquer *trigger* possa ser enviado para o sistema. Estes podem ser vistos em detalhe na seção 1.3.



2.2.7. INPUT



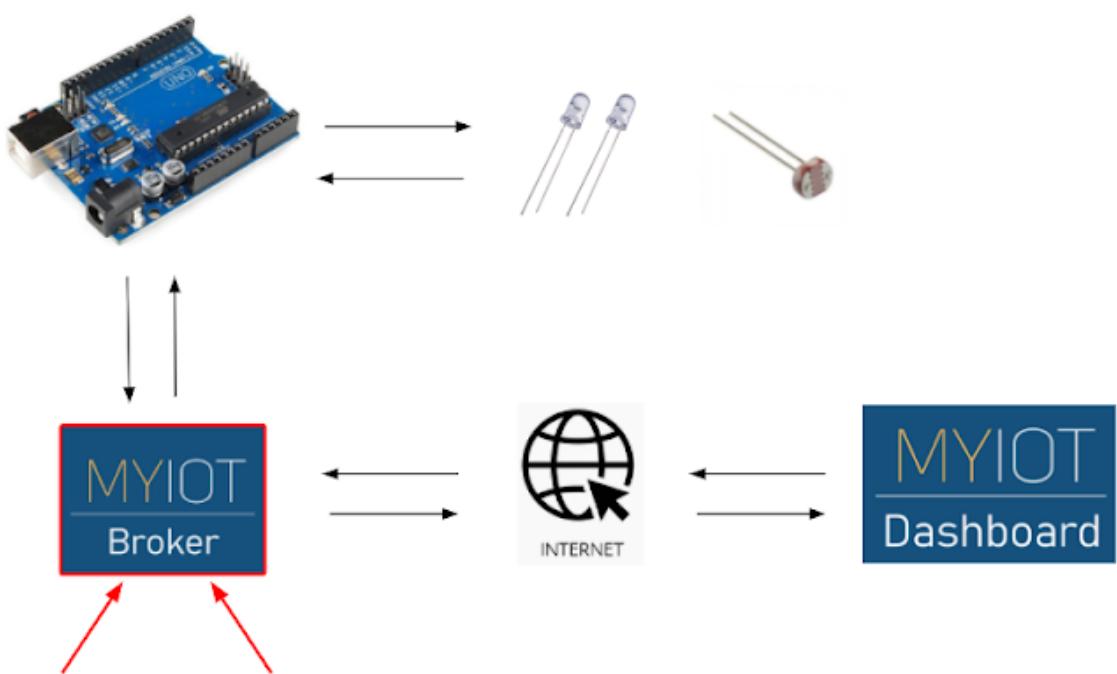
Esta janela funciona como um sistema de perguntas e respostas, com o propósito de ocasionar ações específicas em um sistema baseado na resposta do usuário. A pergunta aparecerá em **Pergunta...** se o comando `#input pergunta` for enviado ao sistema.



A resposta desencadeará ações se a palavra da resposta (exatamente igual) tiver sido previamente programada no aplicativo **Programar** para iniciar algo.

2.2. BROKER

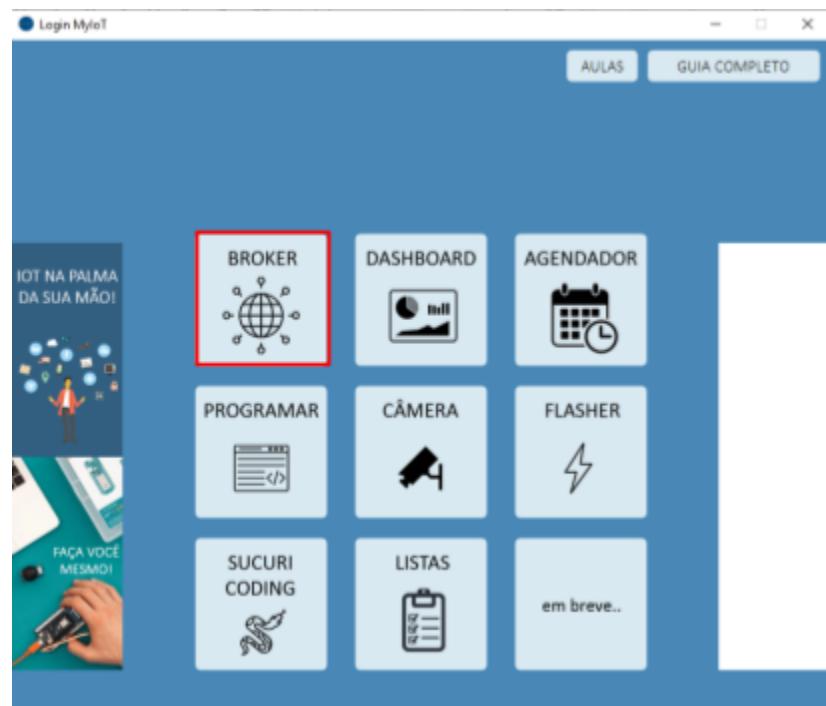
A palavra “Broker” vem do Inglês, e significa “corretor de imóveis”. O conceito gira em volta da prática de intermediação desta profissão, onde o profissional adquire conhecimento de casas e prédios e comunica informações sobre estes para alguém. Em um projeto IoT, o “Broker” é um programa de computador que adquire informações de sensores e atuadores, e então as comunica para a internet. O aplicativo tem o objetivo de intermediar um microcontrolador com à internet, possibilitando a comunicação com outros dispositivos e elementos de projeto.



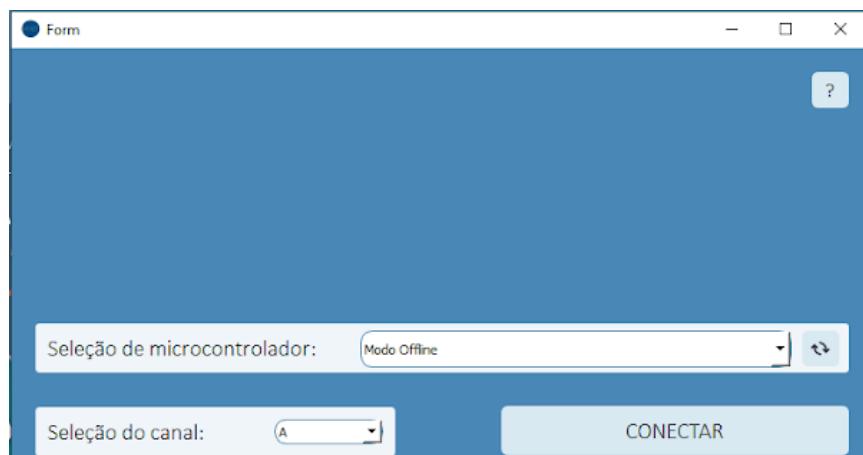
Há duas maneiras de utilizar o “Broker”:

1. **Broker Desktop** - O broker Desktop funciona em um computador PC, necessitando que o microcontrolador esteja conectado à máquina por todo o tempo que este funcione (via cabo USB AB). Para utilizá-lo:

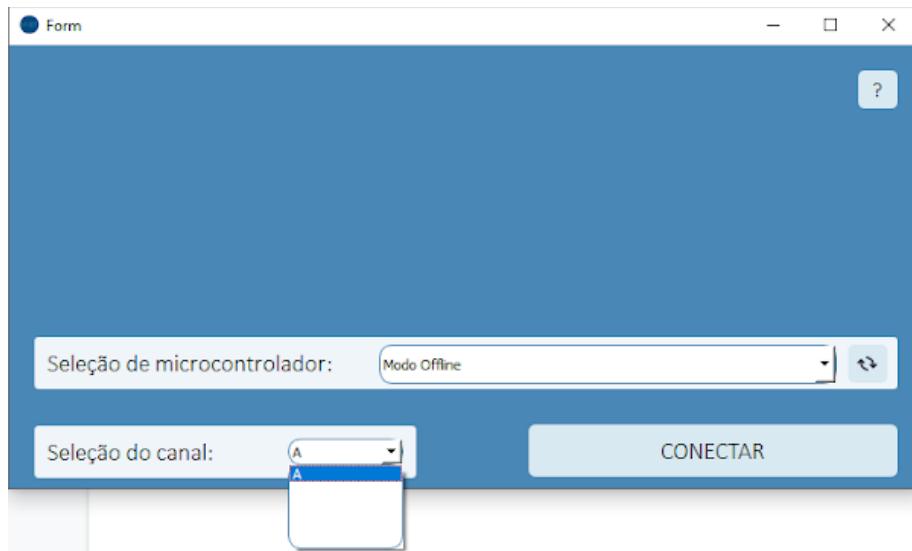
1. No menu principal do MyIoT, clique em “Broker”.



2. Ao abrir o aplicativo, clique no botão e o programa reconhecerá automaticamente o microcontrolador conectado à máquina.



3. Após o microcontrolador ser listado, selecione na primeira linha o canal que o microcontrolador utilizará. Somente é necessário preocupar-se com isso se você deseja utilizar mais de um microcontrolador conectado à um Broker: dois diferentes microcontroladores não podem estar no mesmo canal.



4. Clique em **Login**. O aparecimento do aviso **Conectado** indicará que o sistema está conectado.

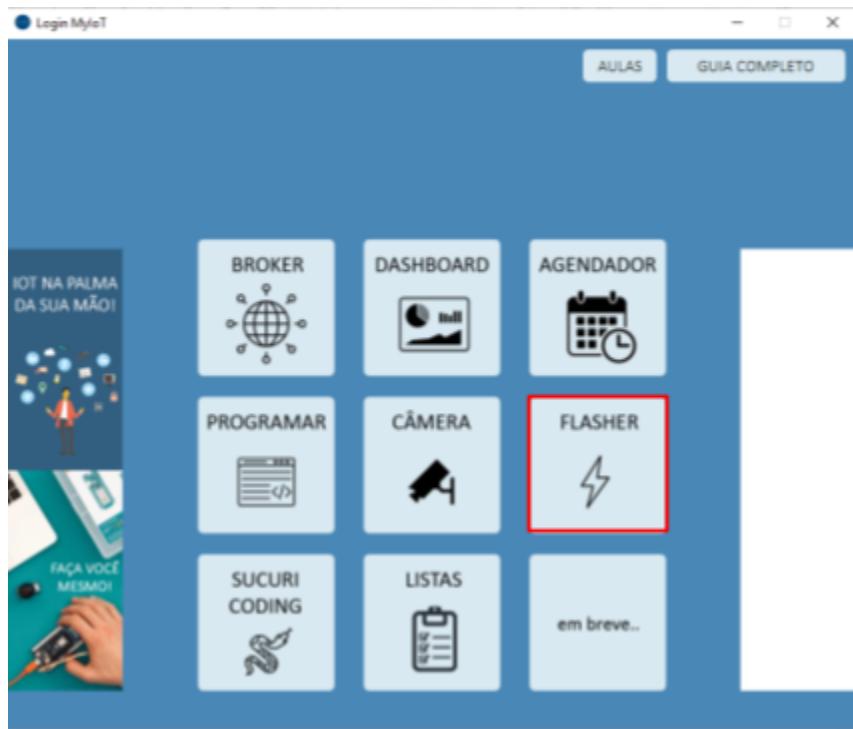


2. **Mini Broker** - O mini broker é uma maneira de possibilitar a conexão com a internet de microcontroladores de forma *Wireless*. O código é inserido em uma placa ESP (microcontrolador capaz de comunicar-se com a internet via WI-FI), e o microcontrolador original somente necessita estar conectado ao “microcontrolador Broker” e a uma fonte de energia. Para utilizá-lo, é necessário inserir o código Mini Broker na placa ESP utilizando o aplicativo MyIoT Flasher. O código Mini Broker está disponível para ESP12, ESP32 e

ESP01. Para mais informações sobre o MyIoT Flasher, acesse a seção 2.5 -

Flasher.

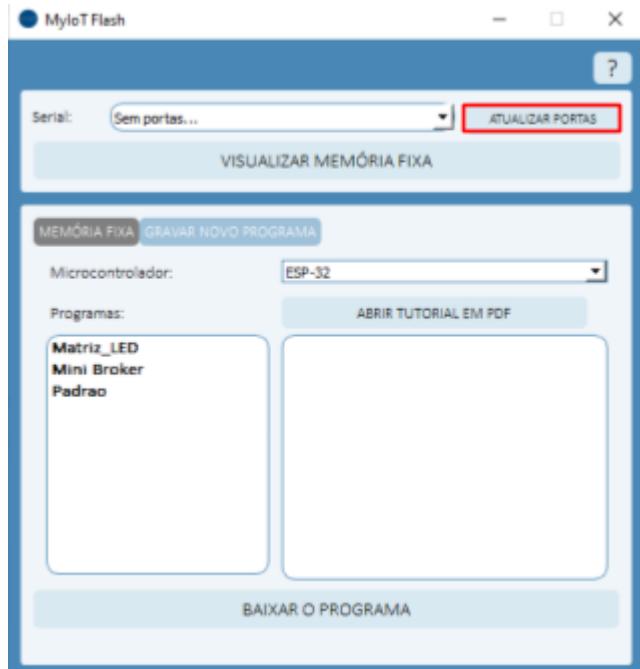
2.3. FLASHER



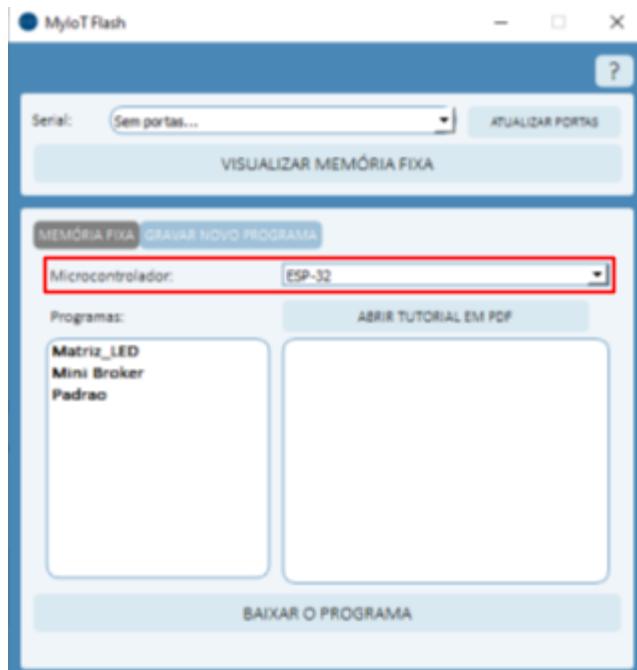
O *Flasher* é um aplicativo que insere automaticamente códigos prontos em microcontroladores. O software contém uma vasta biblioteca de códigos para diferentes controladores, com diversas utilidades e aplicações. Em geral, o programa pode ser utilizado para dois grupos:

FLASHER PARA MICROCONTROLADOR SEM COMUNICAÇÃO WI-FI

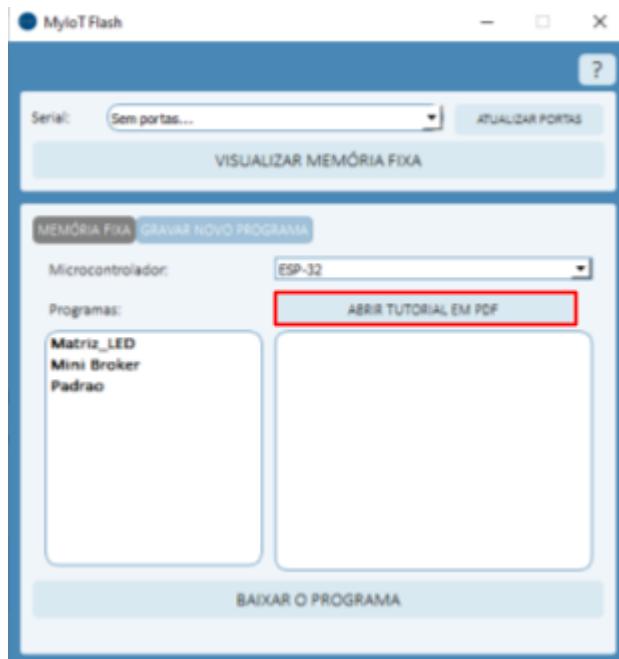
1. Clique em **Atualizar portas**. O microcontrolador conectado à máquina irá aparecer automaticamente.



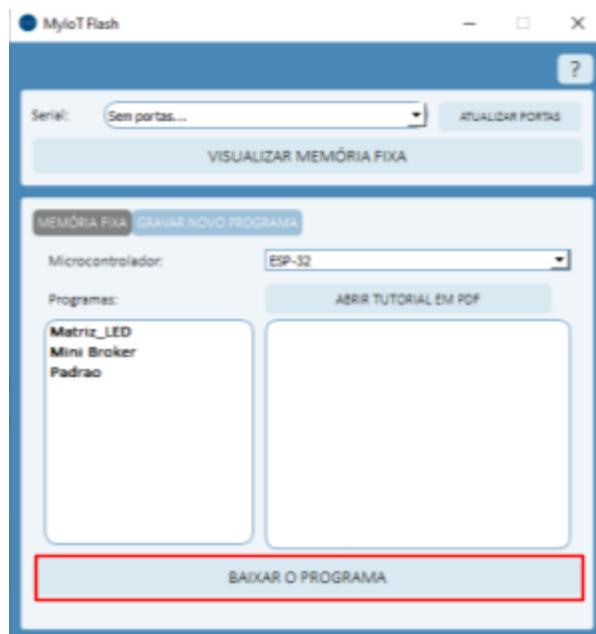
2. Em **Microcontrolador**, escolha o microcontrolador correspondente ao que será utilizado



3. Em **Programas**, escolha o código que deseja para o microcontrolador. No centro direito da tela, um breve texto descreve o código. Para mais informações sobre cada programa, clique em **Abrir Tutorial em PDF**.



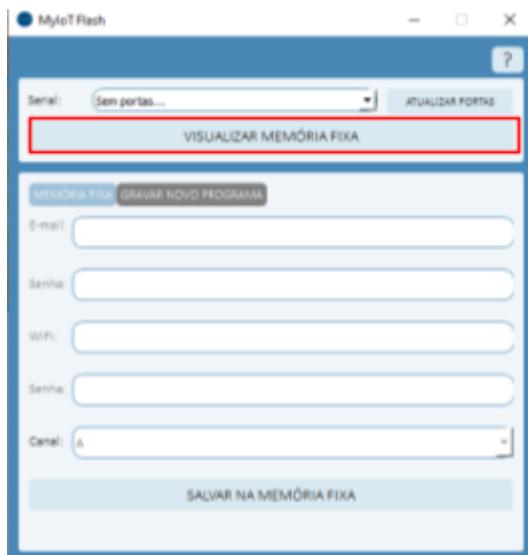
4. Clique em **Baixar o programa**. Após carregar, um aviso indicará o sucesso do Download.



FLASHER PARA MICROCONTROLADOR COM COMUNICAÇÃO WI-FI

Diferente dos microcontroladores sem comunicação WI-FI, agora é necessário inserir os dados da rede WI-FI a ser utilizada.

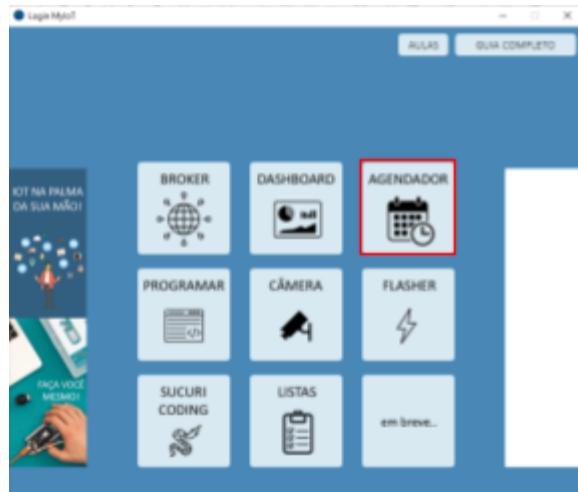
1. Clique em **Atualizar portas**. O microcontrolador conectado à máquina irá aparecer automaticamente.
2. Em **Microcontrolador**, escolha o microcontrolador correspondente ao que será utilizado.
3. Em **Programas**, escolha o código que deseja para o microcontrolador. No centro direito da tela, um breve texto descreverá o código. Para mais informações sobre cada programa, clique em **Abrir Tutorial em PDF**.
4. Clique em **Baixar o programa**. Após o carregamento, um aviso indicará o sucesso do Download.
5. Clique em **Visualizar Memória Fixa**. Uma outra aba será aberta.



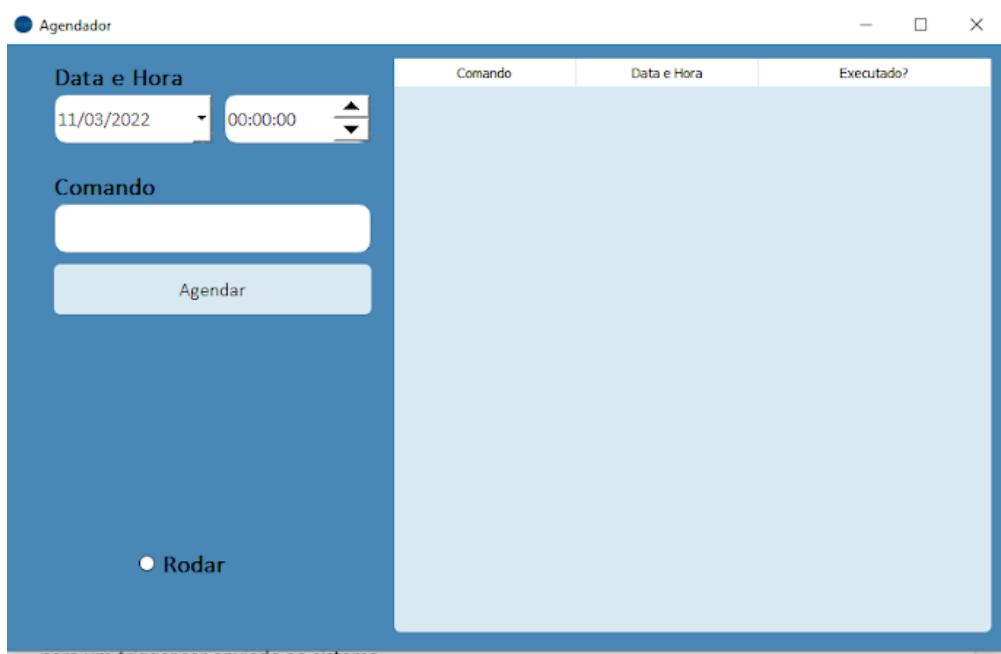
6. Os dados de e-mail e senha serão preenchidos automaticamente, pois correspondem aos dados da conta MyIoT Space utilizados para abrir o Pacote MyIoT. Não altere estes dados. Preencha o nome da rede WI-FI e a senha, atentando-se para não cometer erros de digitação.
7. Escolha o canal que o microcontrolador utilizará. Somente é necessário atentar-se a esta linha se mais de um microcontrolador for ser utilizado na mesma conta Pacote MyIoT. Para esta situação, é necessário definir um canal específico para cada microcontrolador.
8. Clique em **Salvar na Memória Fixa**. Se o botão tornar-se cinza e depois preto novamente, seu microcontrolador estará pronto!

2.4. AGENDADOR

Para abrir o aplicativo, clique em Programar no menu principal do Pacote MyIoT.



O aplicativo agendador possibilita ao usuário marcar um horário e data específicos para um *trigger* ser enviado ao sistema.



Em **Data e Hora**, escolha quando você deseja que o trigger seja enviado. Em **Comando**, digite o *trigger* em si. Marque ou desmarque **Rodar** para habilitar e desabilitar os agendamentos.

3. Mapeamento de sensores e atuadores

3.1. LDR



O sensor LDR é um componente eletrônico cuja resistência diminui conforme a incidência de luminosidade. Ele é utilizado para medir luz, comum em luzes automáticas e de emergência. Este sensor é utilizado em um Flash (código pronto do programa Flasher):

- Flash **LDR** para o microcontrolador ESP 12. Neste projeto o código está configurado para a leitura do sensor ser transmitida em tempo real para o Dashboard, dentro da ferramenta de parâmetros e medições.

3.2. SENSOR ULTRASSÔNICO



O sensor ultrassônico é um componente eletrônico capaz de perceber a distância de objetos à sua frente a partir de uma emissão de onda sonora que rebate no objeto.

Este sensor é utilizado em dois Flashes (códigos prontos do programa Flasher):

- Flash **Ultrassônico** para o microcontrolador ESP 12. Neste projeto o código está configurado para a leitura do sensor ser transmitida em tempo real para o Dashboard, dentro da ferramenta de parâmetros e medições.

- Flash **Alarme de passagem** para o microcontrolador Arduino Uno. Neste projeto o código é configurado para enviar um comando de aviso ao Dashboard se um objeto for detectado a menos de 25cm do sensor. A única conexão física necessária é a inserção do sensor ultrassônico.

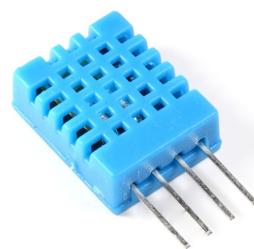
3.3. SENSOR INFRAVERMELHO



O sensor infravermelho é um sensor capaz de perceber objetos à sua frente a partir da irradiação de um feixe de luz infravermelho. Um receptor embutido percebe o rebatimento do sinal e identifica o objeto. Este sensor é utilizado no Flash:

- **Cancela**, disponível para Arduino Uno e Arduino Nano. O programa faz com que um servomotor seja ativado caso o sensor infravermelho perceba algo à sua frente.

3.4. DHT11



O sensor DHT11 é um leitor de umidade e temperatura ambiente, fornecendo digitalmente tais informações. Este sensor é utilizado no Flash (código pronto do programa Flasher):

- **DHT11** para o microcontrolador ESP 12. Neste projeto o código está configurado para a leitura do sensor ser transmitida em tempo real para o Dashboard, dentro da ferramenta de parâmetros e medições.

3.5. BOTÃO



Um botão é um sensor de toque, fornecendo uma informação digital se este está pressionado ou não.

3.6. MATRIZ DE LED



A matriz de LED é um componente eletrônico contendo 8 LEDs de altura e 32 LEDs de comprimento. A partir da ativação específica destas, é possível escrever frases e formar figuras. Este atuador é utilizado no Flash (código pronto do programa Flasher):

- **Matriz LED** para os microcontroladores ESP 12, ESP 32 e Arduino Nano. Neste projeto o código está configurado para ser apresentado na matriz textos de avisos enviados ao Dashboard. As conexões físicas para estes Flash são:

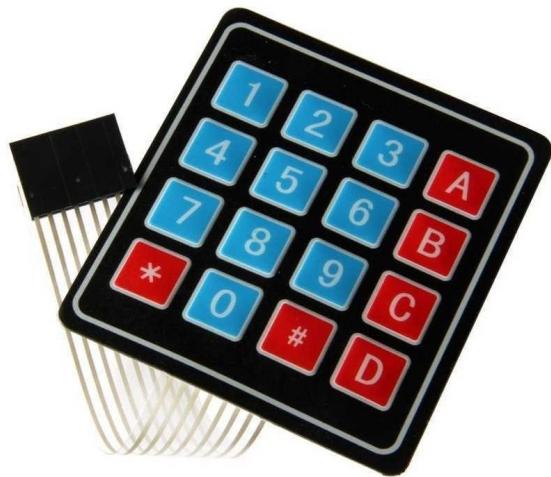
3.7. TELA LCD



A tela LCD tem o mesmo propósito da Matriz de LED, porém é capaz de apresentar textos e figuras mais refinadas. Por outro lado, sua iluminação é inferior àquela da matriz. Este sensor é utilizado no Flash (código pronto do programa Flasher):

- **LCD** para o microcontrolador Arduino Uno. Neste projeto o código está configurado para apresentar na tela avisos enviados ao Dashboard.

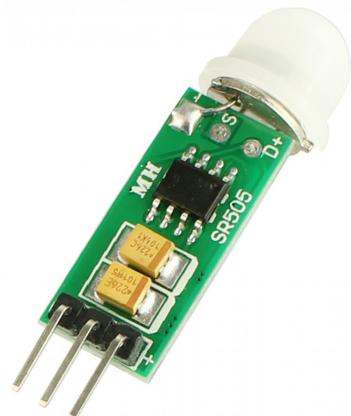
3.8. TECLADO NUMÉRICO



O teclado numérico é uma matriz 4x4 de 16 botões. Uma vez que é possível reproduzir diferentes combinações de toque, este componente “aumenta” a quantidade de botões que poderiam ser inseridos no sistema. Este sensor é utilizado no Flash (código pronto do programa Flasher):

- **Teclado Numérico** para os microcontroladores ESP 12 e Arduino Uno. Nestes projetos o código está configurado para enviar *triggers* a partir do acionamento sequencial correto dos botões.

3.9. PIR



O sensor PIR é um sensor capaz de identificar radiação infravermelha, ou seja, capaz de identificar calor. Desta forma, este sensor é muito preciso e útil para a detecção de movimento humano. Este sensor é utilizado no Flash (código pronto do programa Flasher):

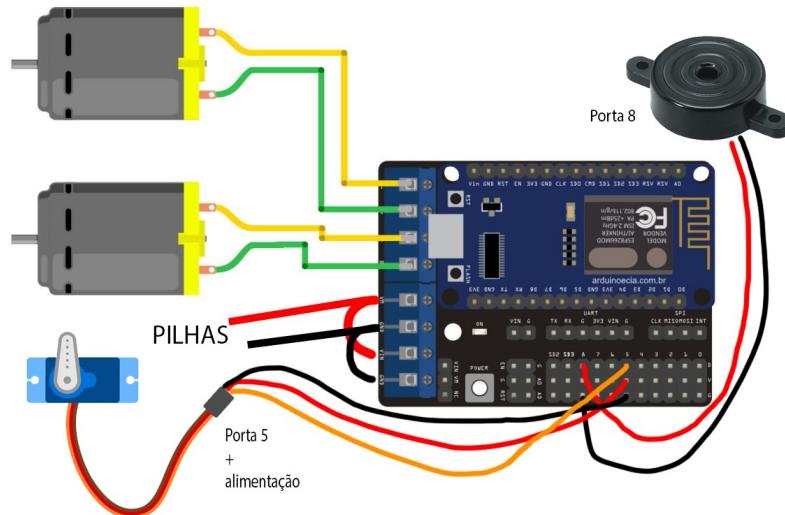
- **PIR** para o microcontrolador ESP 12. Neste projeto o código está configurado para a leitura do sensor ser transmitida em tempo real para o Dashboard, dentro da ferramenta de parâmetros e medições.

3.10. MOTOR ELÉTRICO



Um motor elétrico é um sistema de ímãs que converte energia elétrica em energia mecânica. Este tipo de motor descreve um sistema de alto valor de rotações por minuto, sendo ideal para situações onde uma rotação contínua é necessária. Este atuador é utilizado no Flash (código pronto do programa Flasher):

- **Driver Motor** para o microcontrolador ESP 12. Neste projeto o código está configurado para controlar de formas específicas dois motores. As conexões físicas para este Flash são:



3.11. SERVOMOTOR



O servomotor também é um sistema que converte energia elétrica em mecânica.

Diferente de um motor elétrico comum, este sistema trabalha com angulação, e não rotações por minuto. Desta forma, o motor é ideal para a movimentação de braços robóticos de movimentação curta e precisa. Este atuador é utilizado no Flash:

- **Cancela**, disponível para Arduino Uno e Arduino Nano. O programa faz com que um servomotor seja ativado caso o sensor infravermelho perceba algo à sua frente.

3.12. JOYSTICK



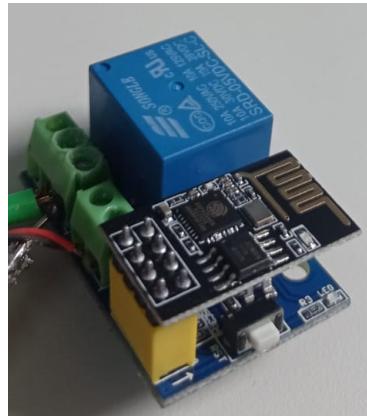
O Joystick é um sensor de toque de alta sensibilidade, permitindo o envio de sinais de intensidades e direções específicas conforme a necessidade. Ideal para o controle de sistemas de movimento, como carros de controle remoto e drones.

3.13. RELÉ



Um relé é um componente eletrônico que funciona como um interruptor. A partir da presença ou não presença de uma corrente, é possível controlar o interruptor conforme a necessidade de um projeto. Este atuador é utilizado no Flash:

- **Padrão**, disponível para o ESP 01. O programa faz com que o relé seja ligado e desligado a partir de *triggers* enviados pelo Dashboard. O ESP 01 pode ser conectado ao Relé a partir do adaptador embutido de ESP 01.



4. MAPEAMENTO DOS MICROCONTROLADORES

4.2. ESP 32



O microcontrolador ESP-32 é usualmente comercializado em uma placa de pinos com o chip original ESP. A placa possui conexão para dados via Micro USB, possui módulo WI-FI e tem um baixo consumo de energia.

No aplicativo Flash do pacote MyIoT, os seguintes Flashes estão disponíveis para este microcontrolador:

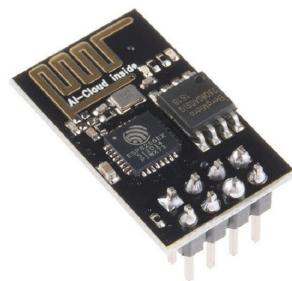
| Nome do Flash | Resumo | Sensores e atuadores |
|---------------|--|----------------------|
| Matriz de LED | Mostra na matriz avisos enviados ao Dashboard. | Matriz de LED |
| Mini Broker | Faz com que a placa possa ser conectada à | Nenhum |

| | | |
|--------|---|----------------------|
| | um microcontrolador sem comunicação WIFI e atue como um Broker | |
| Padrão | Configura um padrão para as portas, definindo-as como Inputs, Outputs e se são digitais ou analógicas | À escolha do usuário |

*Como a placa possui comunicação WIFI, todos os seus projetos são IoT.

Para mais informações, o tutorial detalhado de cada Flash pode ser acessado dentro do programa.

4.3. ESP 01



O microcontrolador ESP 01 é uma placa de pinos com o chip original ESP. A placa necessita de um adaptador USB para conexão de dados, possui módulo WI-FI e tem um baixo consumo de energia.



Adaptador USB para conexão de dados

No aplicativo Flash do pacote MyIoT, os seguintes Flashes estão disponíveis para este microcontrolador:

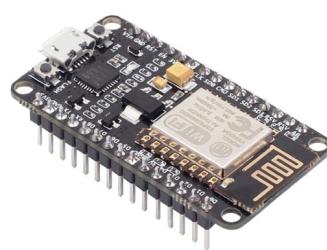
| Nome do Flash | Resumo | Sensores e atuadores |
|---------------|--|----------------------|
| Padrão | Faz com que um relé possa ser controlado pelo Dashboard. | Relé |
| Mini Broker | Faz com que a placa possa ser conectada à | Nenhum |

| | | |
|--|--|--|
| | um microcontrolador sem comunicação WIFI e atue como um Broker | |
|--|--|--|

*Como a placa possui comunicação WIFI, todos os seus projetos são IoT.

Para mais informações, o tutorial detalhado de cada Flash pode ser acessado dentro do programa.

4.4. ESP 8266 12



O microcontrolador ESP 8266 12 é uma placa de pinos com o chip original ESP. A placa possui conexão para dados via Micro USB, possui módulo WI-FI e tem um baixo consumo de energia.

No aplicativo Flash do pacote MyIoT, os seguintes Flashes estão disponíveis para este microcontrolador:

| Nome do Flash | Resumo | Sensores e atuadores utilizados |
|---------------------|---|-----------------------------------|
| DHT11 | Faz a leitura do sensor DHT11 seja apresentada em tempo real no Dashboard. | DHT11 |
| Driver Motor | Permite o funcionamento e controle de motores de 3.7v, um servomotor e um buzzer. | Dois motores, servomotor, buzzer. |
| ESP Alexa | Faz com que seja possível enviar <i>triggers</i> por áudio através da Alexa. | Alexa |

| | | |
|-------------------------|---|------------------|
| LDR | Faz a leitura do sensor LDR seja apresentada em tempo real no Dashboard. | LDR |
| Matriz de LED | Avisos enviados ao Dashboard e ao microcontrolador são lidos na Matriz | Matriz de LED |
| Mini Broker | Faz com que a placa possa ser conectada à um microcontrolador sem comunicação WIFI e atue como um Broker. | Nenhum |
| Padrão | Envia avisos ao Dashboard sobre a luminosidade de um ambiente. | Nenhum |
| PIR | Faz a leitura do sensor PIR seja apresentada em tempo real no Dashboard. | PIR |
| Teclado Numérico | Envia comandos à nuvem de acordo com a sequência de botões pressionados. | Teclado Numérico |
| Ultrassônico | Faz a leitura do sensor Ultrassônico seja apresentada em tempo real no Dashboard. | Ultrassônico |

*Como a placa possui comunicação WIFI, todos os seus projetos são IoT.

Para mais informações, o tutorial detalhado de cada Flash pode ser acessado dentro do programa.

4.5. ESP CAM



O microcontrolador ESP CAM é uma placa de pinos com o microcontrolador original ESP. A placa necessita de um adaptador para conexão para dados via Micro USB, possui módulo WI-FI e tem um baixo consumo de energia. Esta placa vem com uma câmera embutida, já conectada às portas do microcontrolador.

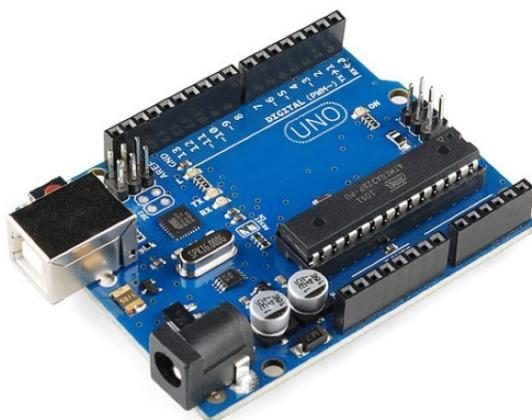
No aplicativo Flash do pacote MyIoT, os seguintes Flashes estão disponíveis para este microcontrolador:

| Nome do Flash | Resumo | Sensores e atuadores |
|-----------------|---|-------------------------------------|
| Câmera | Faz com que o IP da ESPCAM seja enviado ao Dashboard. Acessando este IP em um navegador, a câmera poderá ser vista em tempo real. | Câmera acoplada ao microcontrolador |
| Fotos com Flash | Faz com que o microcontrolador tire fotos com flash a cada 15 segundos, e então as grave em um cartão de memória. | Câmera acoplada ao microcontrolador |
| Fotos sem Flash | Faz com que o microcontrolador tire fotos sem flash a cada 15 segundos, e então as grave em um cartão de memória. | Câmera acoplada ao microcontrolador |

*Como a placa possui comunicação WIFI, todos os seus projetos são IoT.

Para mais informações, o tutorial detalhado de cada Flash pode ser acessado dentro do programa.

4.6. ARDUINO UNO



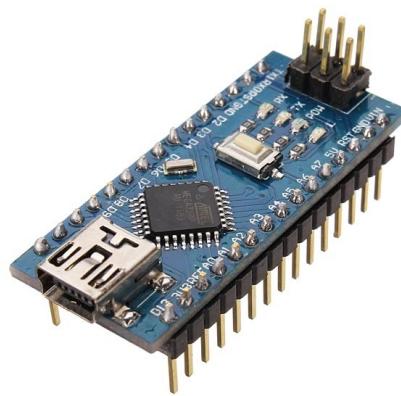
O microcontrolador Arduino Uno é uma placa de pinos com o microcontrolador original atmega. A placa possui conexão para dados via USB AB, não possui módulo WI-FI e tem um baixo consumo de energia.

No aplicativo Flash do pacote MyIoT, os seguintes Flashes estão disponíveis para este microcontrolador:

| Nome do Flash | Resumo | Sensores e atuadores | É um projeto IoT? |
|-----------------------------|--|----------------------------------|-------------------|
| Blink | Faz com que o LED verde OnBoard pisque. É um programa para testes. | LED onboard | NÃO |
| Cancela | Uma estrutura de cancela se movimenta com a leitura de um sensor. | Sensor infravermelho, servomotor | NÃO |
| Dashboard Motores | Permite o controle de motores conectados à uma ponte H. | Motores | SIM |
| Dashboard Saídas | Permite com que diferentes portas possam ser energizadas e desligadas através da ferramenta Enviar Triggers do Dashboard. | Qualquer um | SIM |
| Dashboard Saídas e Entradas | Permite com que diferentes portas possam ser controladas, ao mesmo tempo que define algumas portas para monitoramento em tempo real via Dashboard. | Qualquer um | SIM |
| Terminal Teclado | Faz com que sequências específicas apertadas no teclado envie comandos para o Dashboard. | Teclado Matricial | SIM |
| Ultrassônico | Envia a leitura em tempo real de um sensor ultrassônico conectado à uma porta específica ao Dashboard. | Sensor Ultrassônico | SIM |

Para mais informações, o tutorial detalhado de cada Flash pode ser acessado dentro do programa.

4.7. ARDUINO NANO



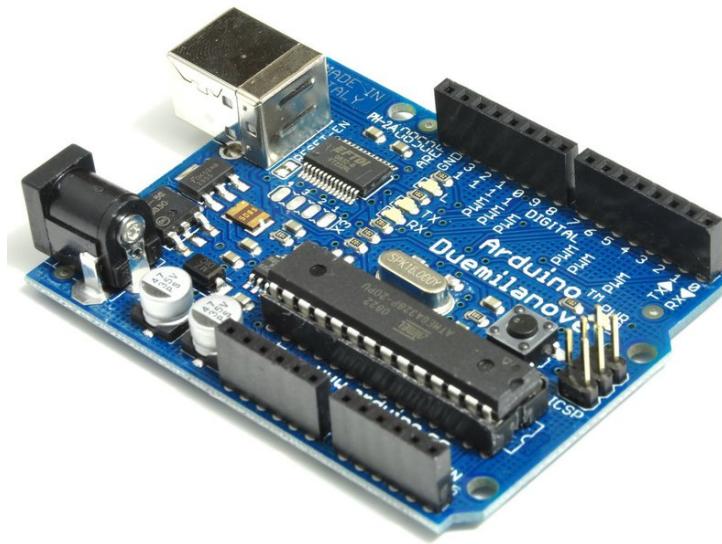
O microcontrolador Arduino Nano é uma placa de pinos com o microcontrolador original Atmega. A placa possui conexão para dados via Mini USB, não possui módulo WI-FI e tem um baixo consumo de energia. Esta placa é similar ao Arduino Uno, porém possui um tamanho mais compacto para aplicações específicas.

No aplicativo Flash do pacote MyIoT, os seguintes Flashes estão disponíveis para este microcontrolador:

| Nome do Flash | Resumo | Sensores e atuadores | É um projeto IoT? |
|---------------|--|----------------------------------|-------------------|
| Blink | Faz com que o LED verde OnBoard pisque. É um programa para testes. | LED onboard | NÃO |
| Cancela | Uma estrutura de cancela se movimenta com a leitura de um sensor. | Sensor infravermelho, servomotor | NÃO |

Para mais informações, o tutorial detalhado de cada Flash pode ser acessado dentro do programa.

4.8. ARDUINO DUEMILANOVE



O microcontrolador Arduino Duemilanove é uma placa com o chip original Atmega. A placa possui conexão para dados via USB AB e não possui módulo WI-FI.

No aplicativo Flash do pacote MyIoT, os seguintes Flashes estão disponíveis para este microcontrolador:

| Nome do Flash | Resumo | Sensores e atuadores utilizados | É um projeto IoT? |
|------------------------------------|--|----------------------------------|-------------------|
| Blink | Faz com que o LED verde OnBoard pisque. É um programa para testes. | LED onboard | NÃO |
| Cancela | Uma estrutura de cancela se movimenta com a leitura de um sensor. | Sensor infravermelho, servomotor | NÃO |
| Dashboard Motores | Triggers enviados pela ferramenta Enviar Triggers do Dashboard controlam motores conectados à portas específicas. | Motores | SIM |
| Dashboard Saídas | Triggers enviados pela ferramenta Enviar Triggers do Dashboard energizam e desligam diferentes portas do microcontrolador. | Qualquer um | SIM |
| Dashboard Saídas e Entradas | Triggers enviados pela ferramenta Enviar Triggers do Dashboard energizam e desligam diferentes portas do | Qualquer um | SIM |

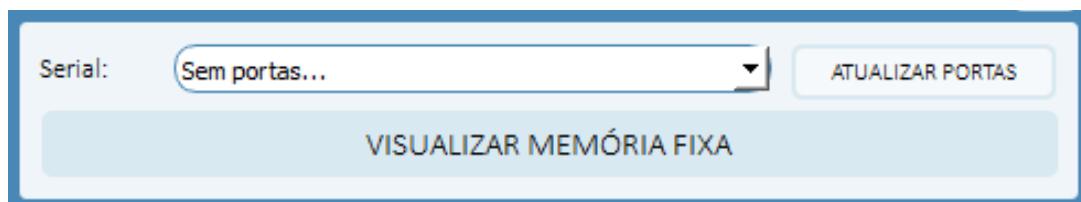
| | | | |
|------------------------|--|------------------|-----|
| | microcontrolador, ao mesmo tempo que outras portas enviam a leitura da porta em tempo real para o Dashboard. | | |
| LCD | Mostra no display avisos enviados ao Dashboard. | LCD | SIM |
| LED Inteligente | Ative e desative um LED pelo Dashboard. | LED onboard | SIM |
| Matriz de LED | Mostra na matriz avisos enviados ao Dashboard. | Matriz de LED | SIM |
| Sensor de Luz | Envia avisos ao Dashboard sobre a luminosidade de um ambiente. | LDR | SIM |
| Teclado Numérico | Envia comandos à nuvem de acordo com a sequência de botões pressionados. | Teclado Numérico | SIM |
| Dashboard Ultrassônico | Envia leituras do sensor ultrassônico ao Dashboard. | Ultrassônico | SIM |

Para mais informações, o tutorial detalhado de cada Flash pode ser acessado dentro do aplicativo, na mesma janela do local de Download do código desejado.

Para utilizar esta placa com projetos de Internet das Coisas, é necessário utilizar o Broker desktop com o cabo USB AB ou Mobile Broker em uma placa ESP 12 ou 32 para conexão wireless.

5. FALHAS

5.1. FALHA AO ENCONTRAR MICROCONTROLADOR

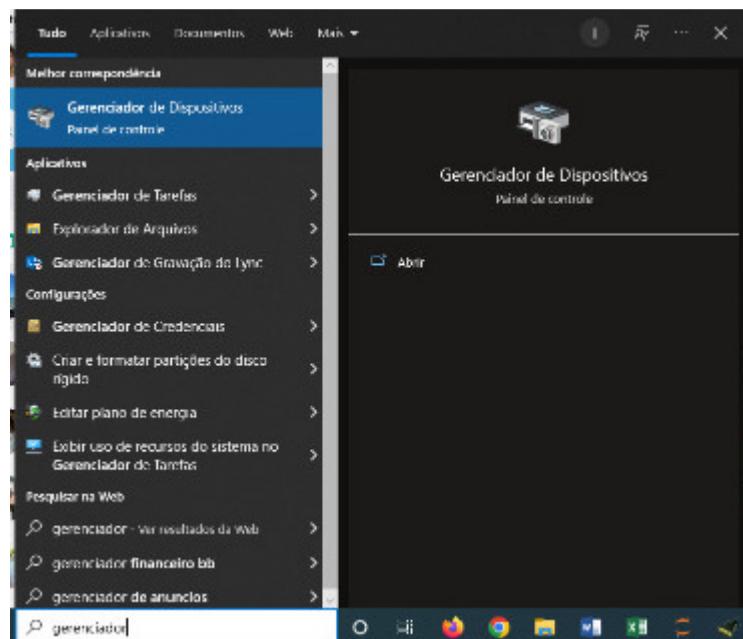


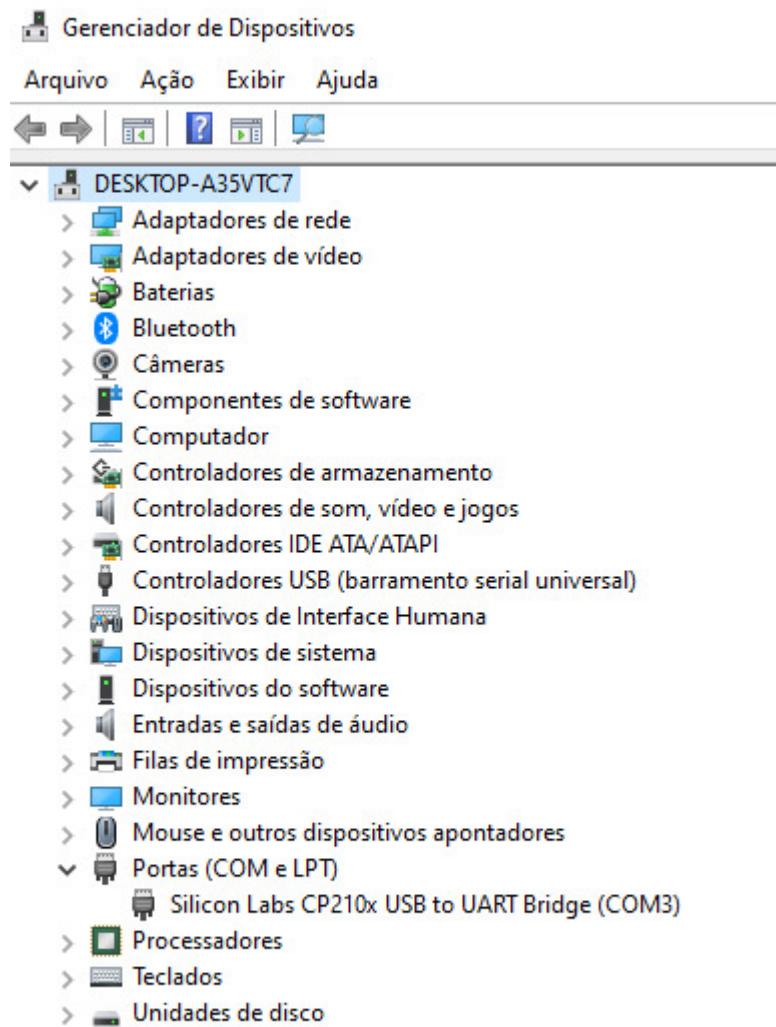
Caso um microcontrolador conectado à máquina não apareça no aplicativo *Flasher* quando o botão **Atualizar Portas** é pressionado, há dois erros comuns que podem estar acontecendo:

1. Cabo de conexão não é um cabo de dados, ou não está funcionando corretamente;
2. Driver do microcontrolador em questão está desatualizado.

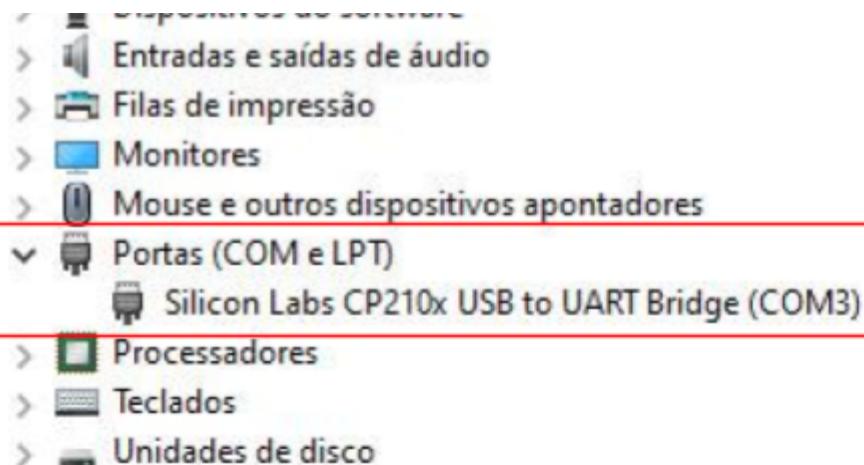
Identificando e resolvendo o problema:

1. Abra em Gerenciador de Dispositivos do Windows.

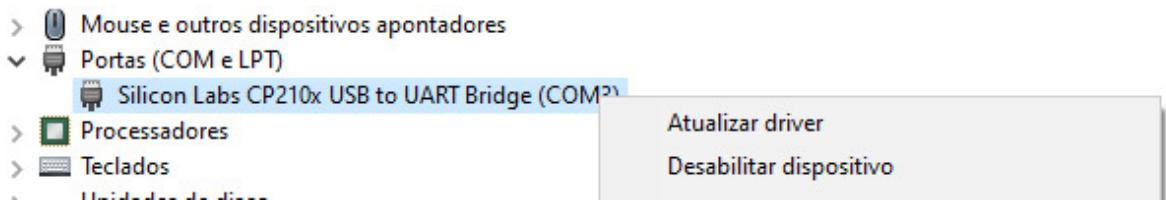




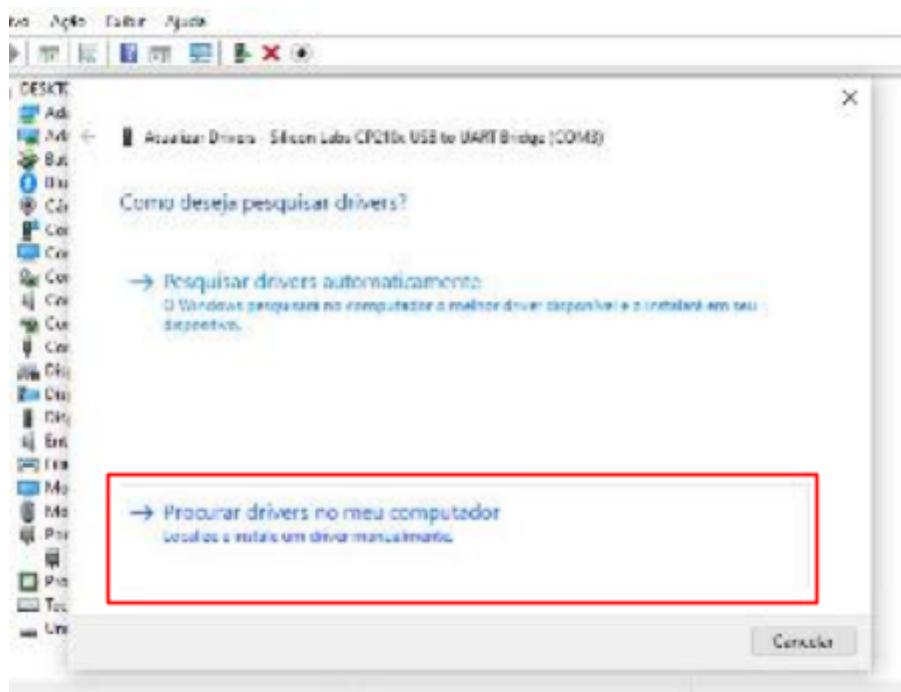
2. Confira se há um dispositivo em Portas (COM e LPT). Se não houver um, troque o cabo. Ou este não está funcionando ou é um cabo feito somente para carga, cujo interior somente possui dois fios (um cabo de dados, adequado para este uso, possui 4 fios).



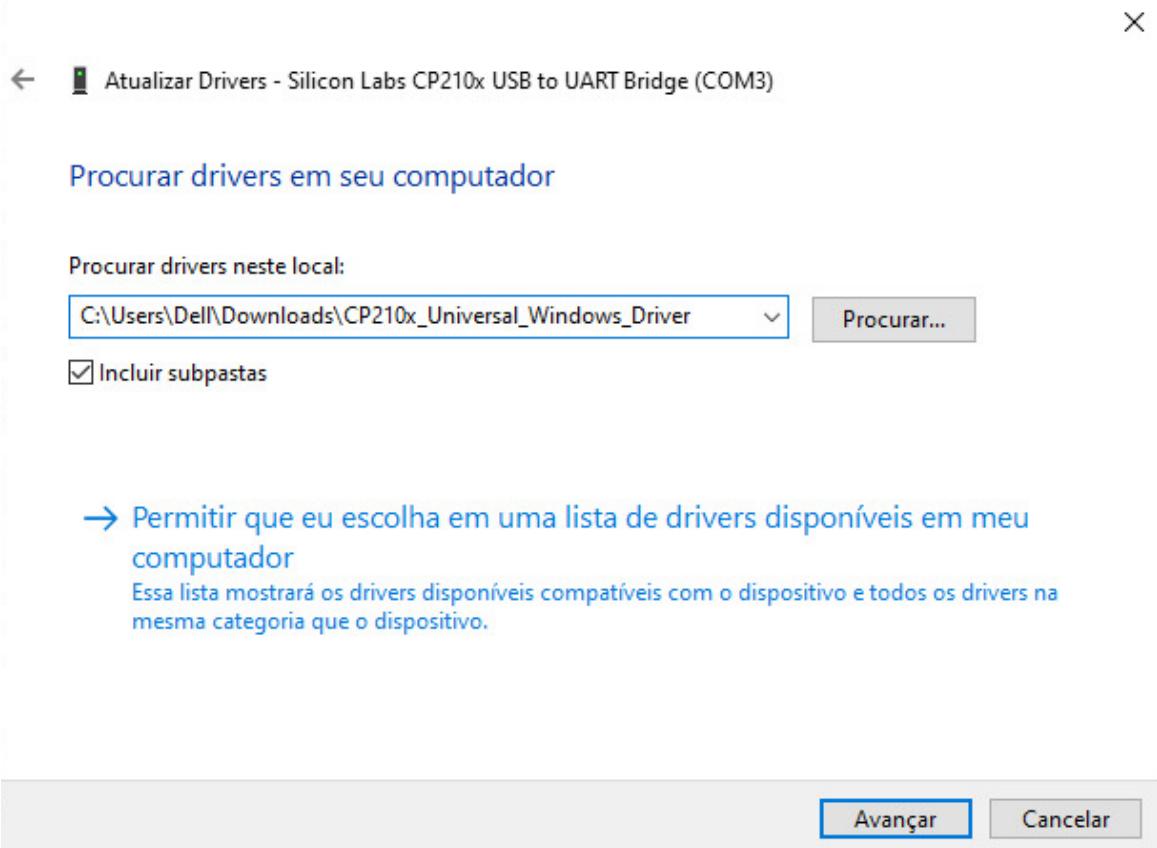
3. Se houver um dispositivo em Portas, será necessário atualizar o driver.
4. Baixe o arquivo CP210x_Universal_Windows_Driver.zip no link
5. Após o término do Download, descompacte o arquivo na área de trabalho.
6. Vá em **Gerenciador de Tarefas**, ache a linha **Portas (COM e LPT)**, clique com o botão direito sobre o dispositivo e clique em **Atualizar Driver**.



7. Selecione **Procurar drivers no meu computador**.



8. Selecione a pasta baixada na área de trabalho, confira se Incluir Subpastas está marcado e clique em **Avançar**.



9. Aguarde a instalação e pronto! Seu driver está atualizado e será possível encontrar seu microcontrolador no aplicativo *Flasher*.