



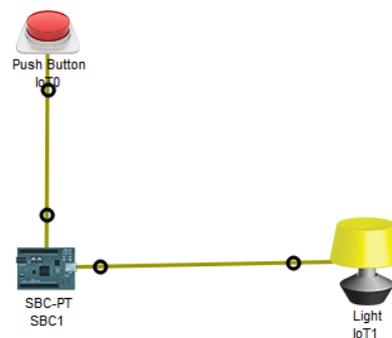
Neste guião vais aprender a **programar** uma **placa** (tipo Arduino) que **controlará** vários **componentes** de **acordo** com o **programa** que tu fizeres.

Abre o Packet Tracer

Cria um **título novo** no **final** do teu **diário** de **bordo** de nome **Guião 10 Internet das coisas programáveis**.

1ª Parte – Programação em Javascript

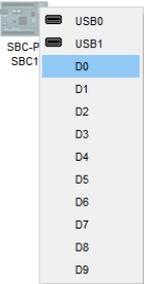
Cria o seguinte **cenário** com **alguns equipamentos inteligentes (IOT)** **interligados** por uma **controladora programável (tipo Arduino/Raspeberry)**.



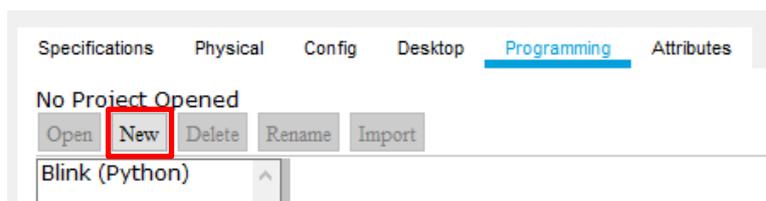
Ajuda: podes encontrar a controladora em **Components**  → **Boards** tipo **SBC** 

Podes encontrar o **botão** de **pressão** na **mesma categoria**, mas em **Sensors**  → 

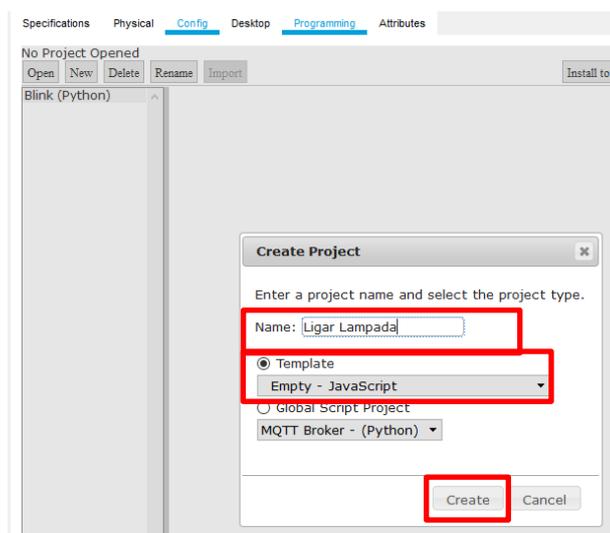
O **cadeeiro** está nos **componentes** das **casas inteligentes**.

	
Para interligar os componentes à Board deves utilizar as portas digitais D0 para o Botão de pressão e D1 para o candeeiro (Ver exemplo acima).	Os fios de ligação são os IOT Custom Cable 

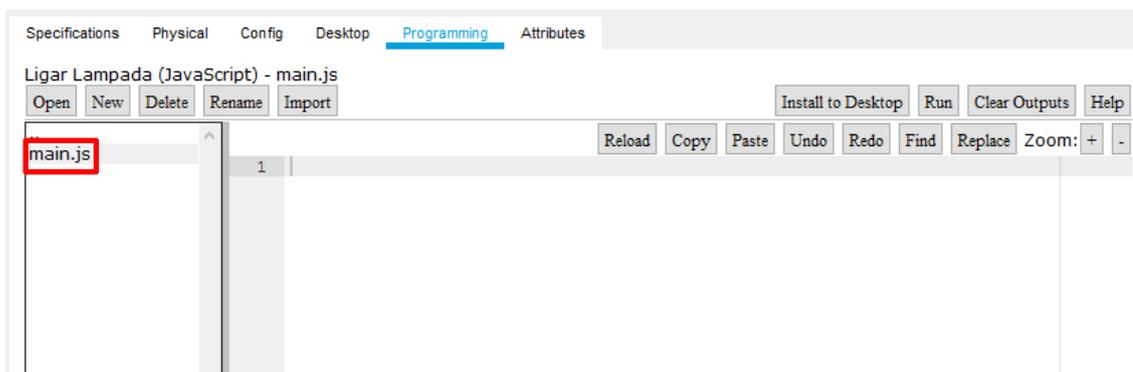
Clica sobre a **placa SBC** e **accede** ao separador **Programming** e clica em **New**.



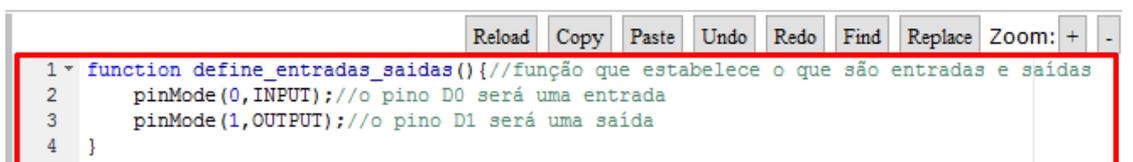
De seguida **muda** o **nome** do ficheiro para **Ligar Lampada** e **cria** um **novo Template Javascript**.



Duplo clique sobre o ficheiro **main.js** para **abrir** o editor.



Copia o seguinte **código** para o **main.js**



Como podes **verificar** esta **função** de nome **define_entradas_saidas** (que pode ter outro nome qualquer a teu gosto) só **serve** para **dizer** que os **pinos** onde **ligamos** o **botão (D0)** é uma **entrada** e o **pino (D1)** onde **ligamos** a **lâmpada** é uma **saída**.



Utilização: *pinMode(número do pino, INPUT/OUTPUT)*

ATENÇÃO: A LINGUAGEM JAVASCRIPT É CASE SENSITIVE LOGO SENSÍVEL A CARATERES MAIÚSCULAS E MINÚSCULOS. TEM ATENÇÃO COMO ESCREVES!

 Agora a lógica. Pretende-se que ao carregar no botão a lâmpada acenda e caso contrário ela apague. Acrescenta as seguintes linhas de código (abaixo) que te permitem fazer o descrito.

```
5 function loop() { //função que está sempre a correr
6   if(digitalRead(0,1)) { //se a leitura do pino digital 0 estiver 1
7     customWrite(1,2); //lança uma saída na porta digita 1 com valor 2
8   }
9   else { //caso contrário
10    customWrite(1,0); //lança uma saída na porta digita 1 com valor 0
11  }
12 }
```

No código acima é necessário ter algumas coisas em atenção.

1. A função tem de se chamar **loop** ou não irá correr para sempre.
2. A função está constantemente a ler a porta digital 0 (onde está ligado o botão) através de **digitalRead(0,1)**. Se estiver a 1 (botão pressionado) ela coloca o pino digital 1 com o valor 2 (**customWrite(1,2)**). Caso contrário (botão não está pressionado) coloca o valor da porta digital 1 a 0 (**customWrite(1,0)**).
3. Para saber como escrever para as portas temos de consultar cada elemento IOT. Como aqui só temos o candeeiro basta clicar 2X sobre o candeeiro e ler no separador especificações (Specifications):

Local Control:

- Connect device to MCU/SBC/Thing. Use the "customWrite" API per Data Specifications

Indica que para interagir com o candeeiro devemos utilizar **customWrite**.

Utilização: *customWrite(número do pino, estado)*

Para consultar os estados possíveis ler mais abaixo:

Data Specifications:

Message Format: [state]

state: 0 = off, 1 = dim, 2 = on

O valor 2 será ligado, o 1 será meia luz e o 0 desligado.

 Finalmente corre o programa em Run.



```
Ligar Lampada (JavaScript) - main.js
Open New Delete Rename Import Install to Desktop Run Clear Outputs Help
Reload Copy Paste Undo Redo Find Replace Zoom: + -
main.js
1 function define_entradas_saidas() { //função que estabelece o que são entradas e saídas
2   pinMode(0, INPUT); //o pino D0 será uma entrada
3   pinMode(1, OUTPUT); //o pino D1 será uma saída
4 }
5 function loop() { //função que está sempre a correr
6   if(digitalRead(0,1)) { //se a leitura do pino digital 0 estiver 1
7     customWrite(1,2); //lança uma saída na porta digita 1 com valor 2
8   }
9   else { //caso contrário
10    customWrite(1,0); //lança uma saída na porta digita 1 com valor 0
11  }
12 }
13 }
```

Utiliza a tecla **Alt + clique do rato** para pressionares o botão e veres a luz acender.

 **Insera uma porta inteligente** ligada a porta digital 2 (D2).

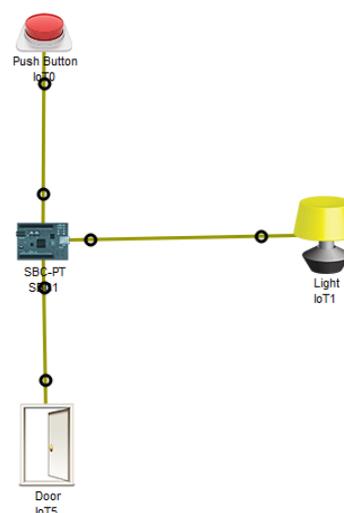
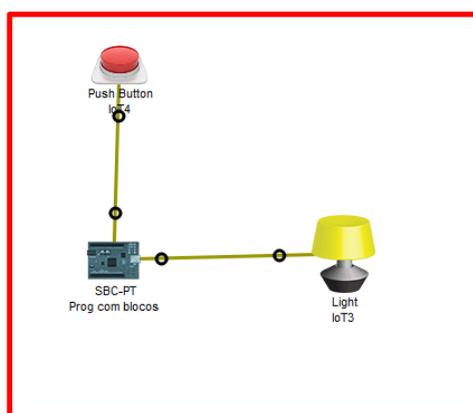
Consulta as especificações da porta para saber qual o método para acionar a porta através de programação e os estados (fechada e aberta).

A porta deve estar aberta quando a luz está apagada e vice-versa.

 **Insera no diário de bordo** um print do esquema de ligações e outro de todo o código

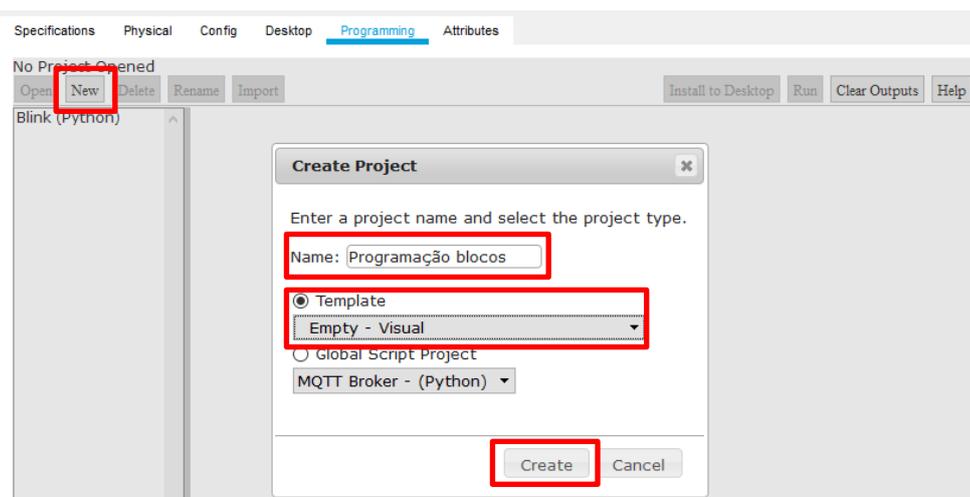
2ª Parte – Programação com Blocos

 **Cria o mesmo cenário** com outra controladora SBC, outro botão de pressão e outro candeeiro.

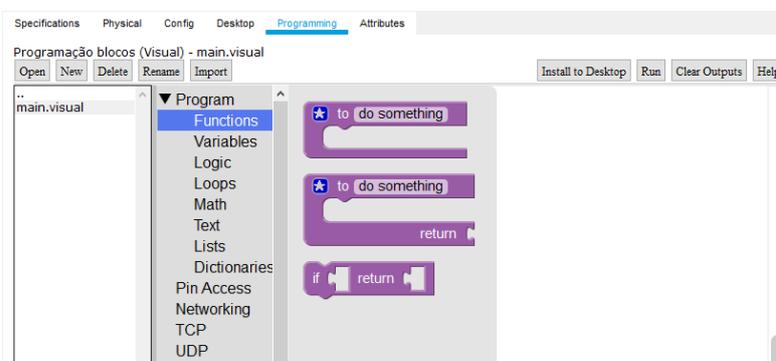


 **Vamos programar a nova placa SBC** para executar as mesmas funções da 1ª parte do guião, mas agora recorrendo a linguagem visual (blocos).

 **Cria um novo projeto visual** de nome “Programação blocos”.



👉 Clica **2X** sobre o ficheiro **main.visual**.



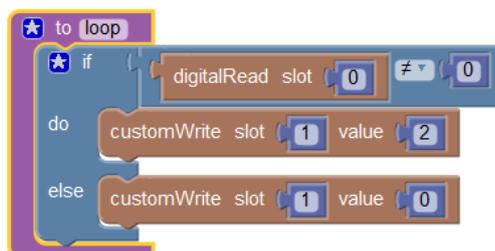
Do teu lado esquerdo tens um **menu vertical** com **todos** os **blocos existentes** listados por **categorias**.

👉 Adiciona os seguintes blocos ao programa.



A função **define_entradas_saidas** define **quais** os **pinos** a **serem utilizados** e se são **entradas** ou **saídas**, tal como na programação em javascript (JS).

👉 Adiciona os seguintes blocos ao programa para **fazer a lâmpada acender** quando **pressionamos o botão**.



Mais uma vez **para funcionar** aqui a **função** tem de se **chamar loop** para **correr para sempre**. No **entanto** há uma **ligeira diferença** relativamente à **programação** em **JS**. **Aqui testamos** se a **entrada digital D0** é **diferente de 0**, uma **vez** que **testar** se seria igual a **1** **não** iria **funcionar** na programação por blocos (porque o botão quando está pressionado não envia o valor 1).

-  Faz **Run** para **correres** o **programa** e **verifica** se **tudo** está a **funcionar** como **pretendido**.
-  **Acrescenta** uma **porta** e os **blocos necessários** para ter um **comportamento igual** à **porta** da **1ª parte do guião**.
-  **Inser**e no **diário** de **bordo** um **print** do **esquema de ligações** e outro de **todo** o **código**
-  **Chama** o teu **professor** para **avaliar**.