

Projeto 1: Construir e programar uma casa domótica com o Zum Kit Junior



Descrição do Projeto:

Através deste manual, aprenderá a construir e a programar uma porta automática e uma fonte de luz inteligente.

Nível de dificuldade: Fácil

Tempo de realização: 4 horas

Materiais:

- Kit Robótica Zum Kit Júnior
- Bitbloq Robotics
- Caixa de cartão
- Fios
- Fita adesiva
- Tesoura
- Cola
- Tecidos
- Bitbloq 3D (opcional)
- Impressora 3D (opcional)

Construa a sua casa domótica

Neste projeto, os alunos terão de construir uma casa com uma porta automática e uma luz inteligente.

Podem construir tanto o interior como o exterior da casa. É importante deixar esta parte ao critério dos alunos, incentivando a sua criatividade e envolvimento no projeto.

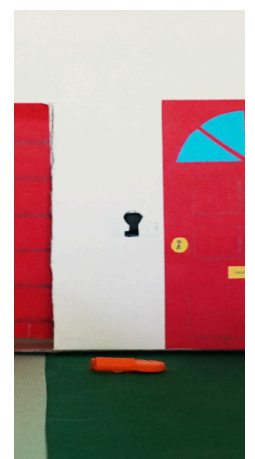
No exemplo abaixo, mostramos o exterior de uma casa, com uma porta de garagem automática e um poste de luz exterior inteligente, que ilumina o jardim.



Os elementos da casa (jardim, rua, casa...) podem ser desenhados com um editor de imagens, imagens tiradas da Internet de *Creative commons* (sem direitos de autor) para serem impressas ou usar cartolinas ou folhas coloridas e desenhar sobre elas.

Como decoração, utilizamos um carro e um poste de iluminação impressos em 3D. Colocamos os componentes necessários no poste para criar a luz inteligente.

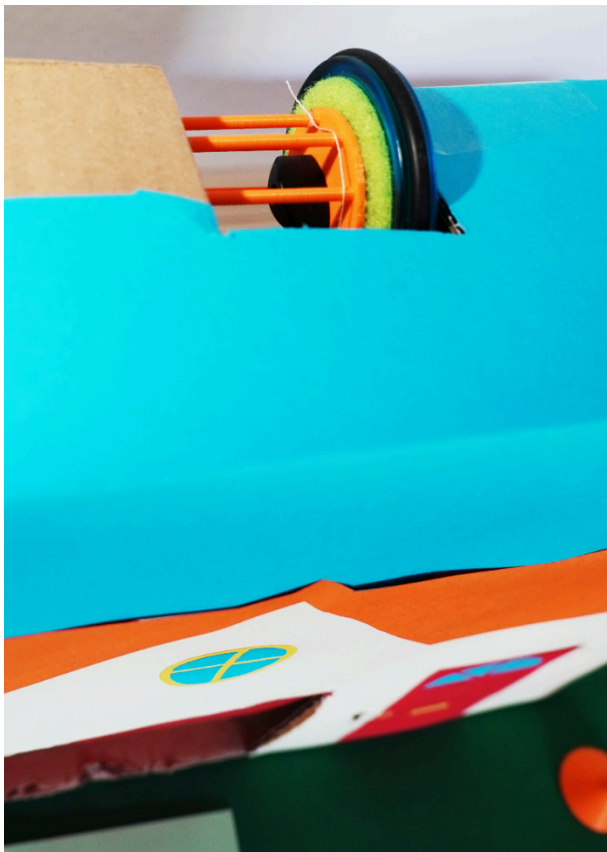
Para automatizar a abertura da porta da garagem, criamos uma fechadura com uma forma específica que ao inserir uma chave com essa forma dará início ao processo de abertura. Dentro da garagem colocamos outro botão que uma vez pressionado fecha a porta.



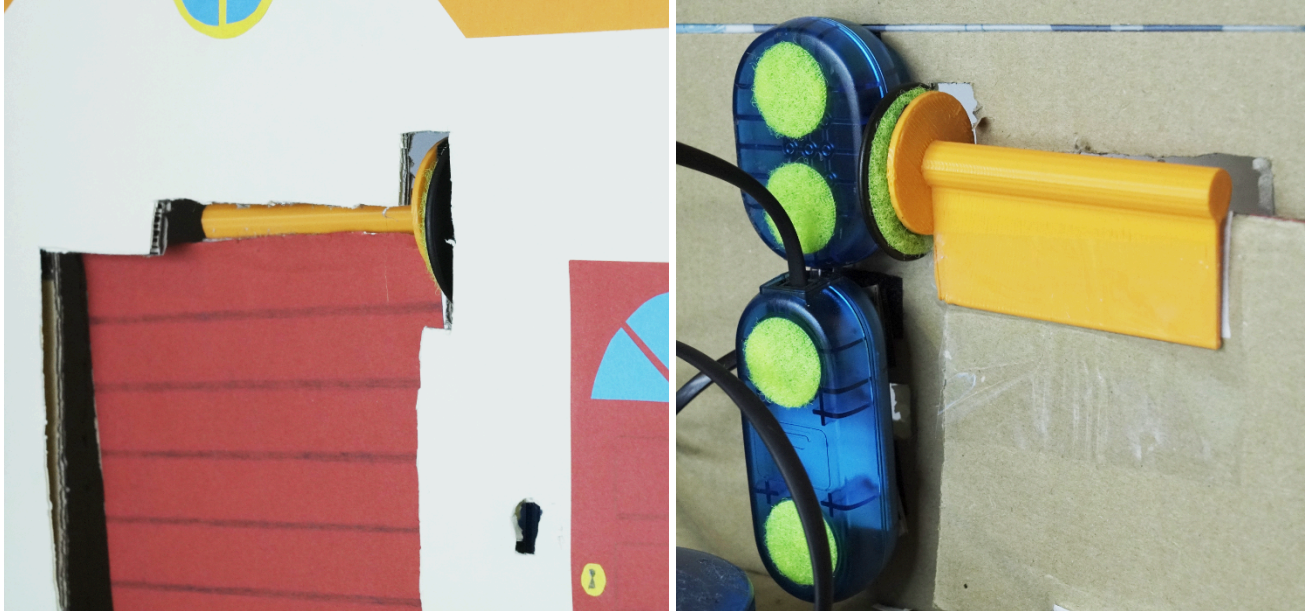
Para a porta, primeiro debemos desenhá-la, depois podemos a recortar de acordo com as nossas preferências e em último lugar usar um dos componentes do kit de robótica para fazer com que ela abra. Podemos usar dois componentes:

1) Um **motor Servo de Rotação Contínua**, amarramos um fio à porta para que esta se enrole para a abrir com a ajuda do motor.

Colocámos o motor servo de Rotação Contínua na parte superior da caixa. Precisaremos que o fio se enrole, para isso teremos que imprimir em 3D uma peça para esse efeito ou criar uma com outros materiais:



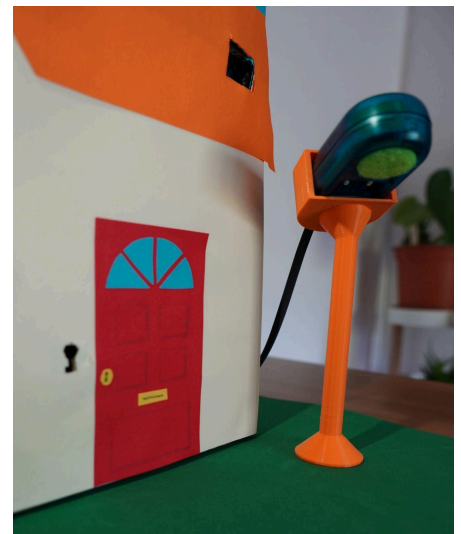
2) Um **motor Servo**, imprime-se em 3D a peça amarela, ou cria-se uma peça parecida com outros materiais. Como podemos ver nas fotos, colocamos a peça no topo da porta, para que a possa ajudar a girar 90 graus.



Para a luz inteligente podemos imprimir em 3D o desenho do poste, devemos pedir aos alunos que desenhem o seu suporte de luz em 3D para imprimir mais tarde, ou usar outros materiais.

A este poste de iluminação juntamos o LED duplo e um Sensor Múltiplo para medir a iluminação.

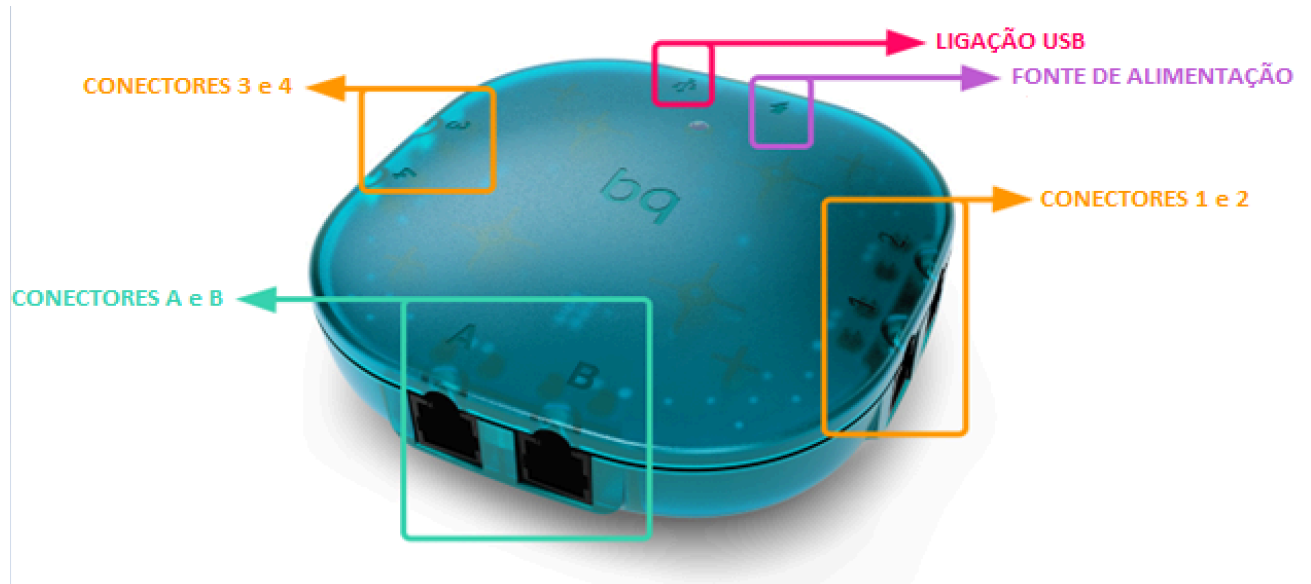
Fazemos um furo na caixa, por cima do poste, no qual metemos o Sensor Múltiplo para que possa medir a iluminação desde esse ponto.



Experimente os componentes do kit

Antes de começar a programar e a construir a casa domótica, propomos a realização de uma breve introdução e alguns exercícios simples para que os alunos aprendam a programar com os componentes do kit de robótica.

Começaremos pela explicação da placa controladora. Este é o dispositivo que guarda toda a informação que recebe através da programação e aplica essas ações aos sensores. Podemos pedir aos alunos que procurem a placa no kit e aproveitar a ocasião para explicar as suas partes:



Conectores 1, 2, 3 e 4: As entradas nas quais podemos ligar todos os componentes, **exceto o Sensor Múltiplo e o Display**.

Entradas A e B: Podem-se conectar todos os componentes.

Ligação USB: Utilizámos para ligar ao computador ou a baterias externas.

Fonte de alimentação: Na qual iremos ligar as baterias ou pilhas.

Em seguida, explicaremos os diferentes tipos de componentes, que podemos dividir em sensores e atuadores:

1. Um **sensor** é um componente que nos fornece informação sobre algo que está a acontecer à nossa volta, como por exemplo o nível de luz, temperatura, um obstáculo próximo, etc. Podemos dizer que eles são equivalentes aos nossos órgãos dos sentidos (visão, tacto, audição, etc.).
2. Um **atuador** é um componente que pode executar uma ação, como emitir uma luz, som ou mover-se. Podemos dizer que são equivalentes à fala, aos membros, etc.

Também podemos classificar os componentes quanto ao uso de sinais digitais ou analógicos:

1. Um componente **digital** é aquele que só pode diferenciar entre dois estados (0/1, verdadeiro/falso, ligado/desligado). Por exemplo, um botão é pressionado ou não, um LED está ligado ou desligado, etc.
2. Um componente **analógico** é aquele que pode ter mais de dois valores. Por exemplo, temperatura, quantidade de luz, etc. Por norma, um sensor analógico pode ler valores entre 0 e 1023.

Uma vez explicada a matéria anterior, passaremos à realização de exercícios práticos simples, para experimentar com os componentes necessários para a casa domótica.

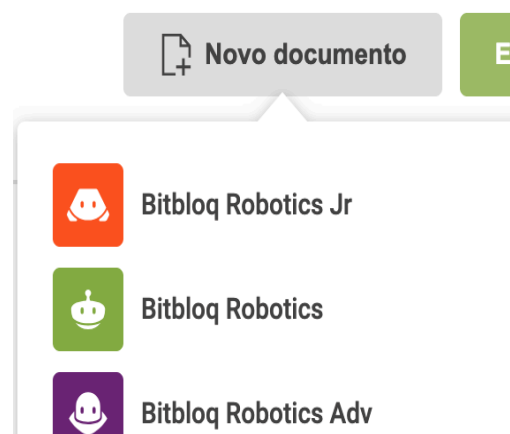
Começaremos pela programação do **LED**:

Pediremos aos alunos que procurem um LED no kit. Perguntamos se é um componente digital ou analógico e se é um sensor ou atuador.



Um **LED** (*Light Emitting Diode*) é um componente atuador capaz de emitir luz. Os LED são baratos, consomem pouca energia e podem ser muito luminosos. Só tem dois valores: ligado ou desligado, por isso é digital. No caso do Zum Kit Junior, temos LED duplos, **nos quais podemos programar duas luzes LED: uma branca e outra colorida, que pode ser vermelho.**

Em seguida, ensinar-lhe-emos a programação necessária para ligar um LED. Para fazer isso, pediremos que conectem a sua placa ao computador usando o cabo USB; ir a [Bitbloq](http://bitbloq.cc) (bitbloq.cc) e criar um novo projeto em *Novo Documento - Bitbloq Robotics*.



A interface gráfica está dividida em três símbolos principais localizados no lado esquerdo:

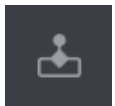


Hardware: onde selecionaremos que tipo de placa e os componentes que vamos utilizar.

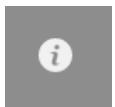


Blocos: onde iremos inserir os blocos de programação do nosso projeto. Este campo divide-se em três partes:

- **Variáveis globais e funções:** Permite declarar variáveis globais e criar funções.
- **Instruções Iniciais (Setup):** Permite configurar ações que serão executadas uma só vez no início do programa.
- **Ciclo Principal (Loop):** Tudo o que se encontra dentro deste campo será executado em loop de forma indefinida.



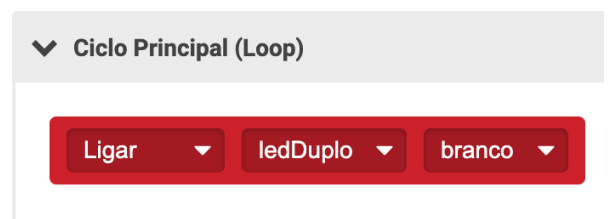
Fluxograma: onde podemos programar o nosso projeto através de diagramas de fluxo ou fluxogramas. O que programamos aqui será visto em *Blocos*, e vice-versa.




Informações do documento: Onde se podem adicionar informações sobre o projeto.

Em seguida, dentro do campo de *Hardware*, lhes diremos para irem a Placas para selecionar a que vamos usar, a Zum Junior, arrastando-a para o centro do ecrã. À qual deverão conectar o LED duplo à correspondente entrada fazendo o mesmo fisicamente.

Depois de estarem feitas as conexões, devem ir para o campo de *Blocos* e dar início à programação no Ciclo Principal (Loop) da seguinte forma:



Como o LED duplo tem dois LED (um de cor e outro branco) será necessário selecionar qual se deseja iluminar. Por exemplo, se quiserem acender os dois ao mesmo tempo, seria necessário colocar outro bloco em baixo, mas neste caso selecionaremos a *cor* em vez de *branco*..

Depois, devem clicar no botão *Carregar*,  , localizado na parte superior direita, para que a programação criada se carregue na placa e desta forma seja possível executar a programação e ver que um dos LED do LED duplo se acende.

Depois de realizar este exercício, pediremos para tentar pôr o LED duplo a piscar, apenas terão de programar para que o LED se ligue e desligue. A programação a criar poderia ser mais ou menos assim:



Geralmente, os alunos programam o *LED duplo* ligado e desligado sem usar o bloco *Esperar* e muitas vezes ficam surpreendidos que o programa não funciona. Aproveitamos a oportunidade para explicar que o programa está a funcionar. Mas, executa-se a uma velocidade extremamente rápida, ou seja, liga e desliga-se tão rápido que não é perceptível ao olho humano. Para poder conseguir verificar como se liga e desliga, temos de adicionar um comando de espera, cuja duração (em milissegundos) é possível modificá-la de acordo com as nossas preferências.

Em seguida, programaremos o **sensor de luz**, um dos sensores do **Sensor Múltiplo**:

Vamos pedir aos nossos alunos que procurem no kit o Sensor Múltiplo e, tal como o LED, perguntamos-lhes se é um componente digital ou analógico e se é um sensor ou atuador.



Com o **Sensor Múltiplo** podemos medir a temperatura, medir distâncias, distinguir cores e ter conhecimento da quantidade de luz. Só pode ser conectado às entradas A ou B.

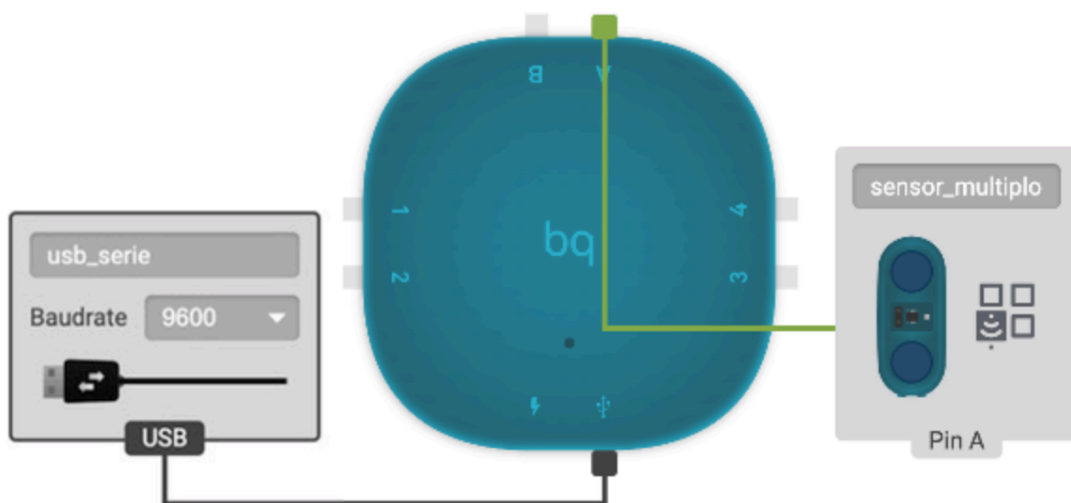
O sensor **de luz do Sensor Múltiplo** envia valores entre 0 (escuridão máxima) e 99 (luminosidade máxima).

Para verificar como funciona este sensor, faremos um programa com os alunos que lhes permita ver os valores que recebe através da entrada USB do computador.



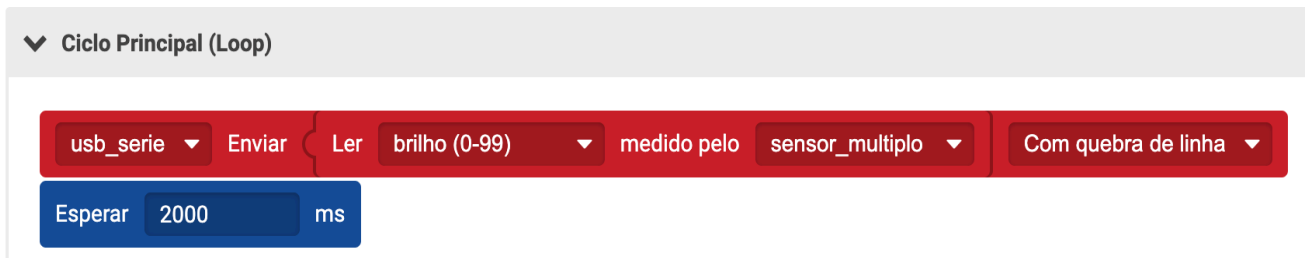
Através da **entrada USB** a placa controladora comunica-se com outro dispositivo, como o computador ou telemóvel. É muito útil verificar o valor de um sensor ou de uma variável, ver o estado de um programa e, finalmente, corrigir e compreender melhor os possíveis erros que possam existir.

Devem conectar o **Sensor Múltiplo** (física e virtualmente) e o **USB serial**, com o cabo USB, como componente no campo *Hardware* de Bitbloq da seguinte forma:



É importante lembrar que o Sensor Múltiplo **só se pode conectar nas entradas A ou B** da placa controladora.

Em seguida, começaremos a programar no campo Ciclo Principal, e, através da porta USB veremos a leitura do sensor de luz com uma espera de alguns milissegundos:



Depois, carregamos a programação na placa e damos um clique em Ver - Monitor porta série, no menu superior esquerdo. Desta forma, aparecerá uma caixa de diálogo que mostrará os valores de luz detetados pelo sensor na sala de aula.



Para verificar como os valores mudam, podemos pedir aos alunos que tapem o sensor de luz e depois o coloquem perto de uma luz para observar como os valores se vão alterando.

Por último, vamos propor aos alunos que criem um sistema de iluminação inteligente da sua casa. Para isso, eles terão de programar o LED duplo para que só se acenda quando o sensor de luz detete que já é noite, ou seja, que haja menos luz. Para programá-lo, devem usar uma **instrução condicional**, de modo que, se o valor detetado pelo sensor de luz for menor que X (por exemplo, 50), a luz acende-se e, se o valor for maior, desliga-se. A programação a usar poderia ser mais ou menos assim:

▼ Ciclo Principal (Loop)



O valor da luz que capta o sensor LED duplo dependerá da iluminação da sala onde estamos. Portanto, antes de programar a fonte de luz inteligente é importante tenhamos em conta a quantidade de luz que irá receber o sensor em cada entrada.

Passaremos agora à programação da **porta automática**. Primeiro, preparamos os componentes que vamos usar. Realizaremos algumas atividades simples com os alunos para que aprendam a trabalhar com o sensor de distância do Sensor Múltiplo, o motor Servo e o motor Servo de Rotação Contínua.

Conforme indicamos na construção, os alunos terão que decidir se querem usar o motor Servo ou o Servo de Rotação Contínua.

Vamos começar por programar o **sensor de distância**:

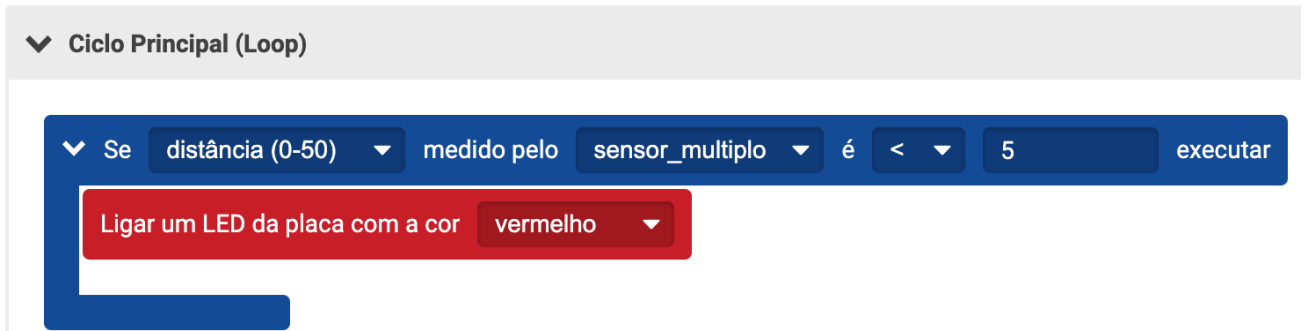
Com o **Sensor Múltiplo** podemos medir a temperatura, medir distâncias, distinguir cores e



ter conhecimento da quantidade de luz. Só pode ser conectado às entradas A ou B.

No caso da deteção de objetos, o Sensor Múltiplo usa a luz refletida para interpretar a **distância** entre um objeto e o sensor, consegue medir até 50 centímetros, aproximadamente.

Para saber como funciona este componente, pediremos aos alunos que programem o seguinte: quando o sensor de distância detetar um objeto muito próximo (por exemplo, a cerca de 5 centímetros de distância), acende-se o LED da placa.



Possivelmente os alunos não programarão o que é que o LED deve fazer quando o sensor deteta uma distância maior. Para fazer isso, teremos que programar com o bloco, *senão, executar*, para que nesse caso o LED se desligue.



Na programação da porta automática da casa, utilizaremos o sensor de distância para indicar que se deteta a existência de um carro a porta abre, ou seja, ativa o motor Servo ou o motor Servo de Rotação Contínua.

Em seguida, vamos programar o **motor Servo**:

Pediremos aos alunos que procurem este componente no kit e explicamos-lhes o que é.



O **motor Servo** é um pequeno motor que é capaz de girar entre 0 e 180 graus, não pode fazer voltas completas. É um atuador e um componente digital.

O mais interessante deste tipo de motor é que decidimos para que posição ou ângulo se deve deslocar.



Propomos um exercício para perceber o funcionamento do motor Servo, que consiste em programar a sua deslocação para os ângulos 0°, 45°, 90°, 135°, 180°.

Em primeiro lugar, conectamos-lo a uma entrada da placa. Depois, para determinar a posição correta, iremos programar nas Instruções Iniciais - *Setup* (para que quando se execute o programa só o faça uma vez) o motor Servo para que se ponha numa determinada posição, por exemplo a 90°. Assim que se coloque nessa posição, podemos desenhar e cortar uma seta de papel e colá-la no motor para nos indicar a posição dos 90°. Desta forma, saberemos em que posição se encontra em cada momento.



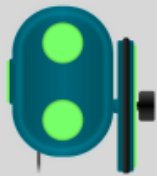
A seguir, pediremos aos alunos para programar o motor Servo para as diferentes posições (0°, 45°, 90°, 135° e 180°). É importante inserir um bloco de espera debaixo de cada bloco de *Mover*, caso contrário, o motor Servo se moverá tão rápido que apenas será visível o ângulo inicial e final.

Além disso, ao inserir uma espera debaixo de cada bloco protege o motor Servo de movimentos bruscos.



Para programar a porta automática com o motor Servo, podemos fazer com que esta se abra de forma progressiva, movendo-se em diferentes ângulos, de 0° a 90°.

Outra forma de abrir a porta seria com o **motor Servo de Rotação Contínua**.



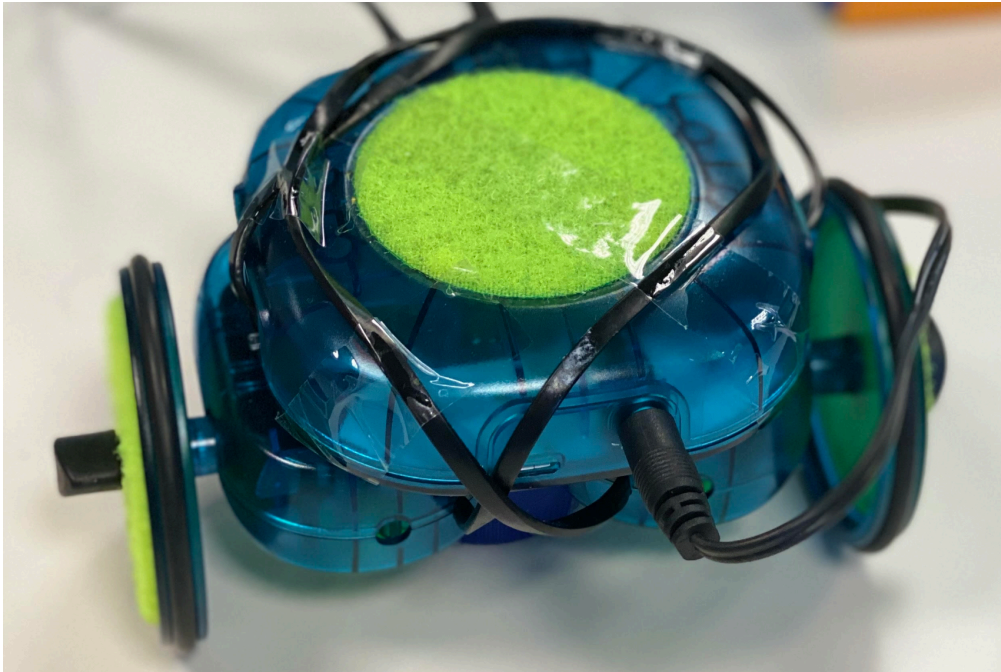
Um **motor Servo de Rotação Contínua** é um motor cujo circuito eletrónico nos permite controlar a direção e a velocidade de rotação. Ao contrário do motor Servo, este não pára numa determinada posição, gira continuamente.

Para provar este componente, podemos pedir aos alunos que programem o motor servo para a andar para a frente, parar, retroceder e parar novamente. Entre cada bloco de movimento, devemos inserir uma espera e indicar o tempo de execução dessa ação ou movimento. Para o efeito, pode-se realizar o seguinte programa:

```

    ▼ Ciclo Principal (Loop)
    Girar servoRC em sentido horário velocidade alta
    Esperar 2000 ms
    Parar servoRC
    Esperar 2000 ms
    Girar servoRC em sentido antihorário velocidade alta
    Esperar 2000 ms
    Parar servoRC
  
```

Em seguida, podemos pedir-lhes para construir um pequeno robot, juntando os dois motores servos com fita-cola (como se fosse um carrinho) e programa-se o seu movimento. Ver o exemplo da seguinte imagem:



Programar a casa domótica

Assim que os alunos aprendam a trabalhar com estes componentes, terão de programar a porta que construíram. Para fazerem isso, a programação poderá variar dependendo do motor que se usa, um motor Servo ou um motor Servo de Rotação Contínua.

No exemplo, foi utilizado um motor servo de Rotação Contínua, que puxa um fio que, por sua vez, sobe ou desce a porta. A programação utilizada é a seguinte:

▼ Ciclo Principal (Loop)

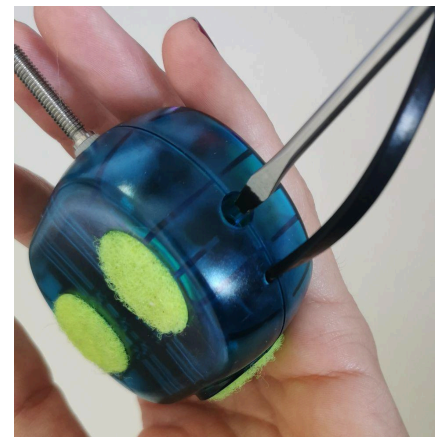
```

Girar servoRC em sentido horário velocidade alta
Esperar 1500 ms
Parar servoRC
Esperar 2000 ms
Girar servoRC em sentido antihorário velocidade alta
Esperar 1500 ms
Parar servoRC
Esperar 2000 ms
    
```

O motor Servo de Rotação Contínua rodará durante 1,5 segundos para abrir a porta. Depois, pára e espera 2 segundos para o carro passar. A última ação, fecha a porta (rodando em sentido contrário durante 1,5 segundos) e pára mais 2 segundos. Como está dentro do Ciclo Principal, a programação será repetida continuamente.

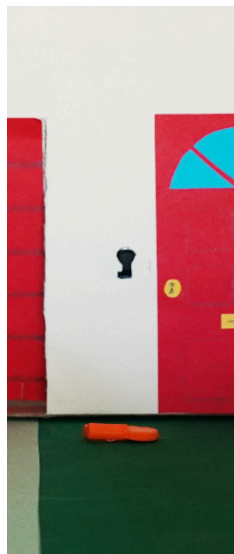
A programação pode variar dependendo da posição do motor servo e do comprimento do fio, pelo que os alunos terão de modificar o tempo de espera até que a porta se abra e feche corretamente.

Se o motor Servo se mover quando deveria de estar parado é porque precisa de ser calibrado. Para fazer isso, teremos que criar um programa em que o motor Servo esteja parado carregá-lo para a placa. Se verificamos que continua a mover-se, com uma chave de fendas apertamos o parafuso lateral lentamente até que o motor pare.

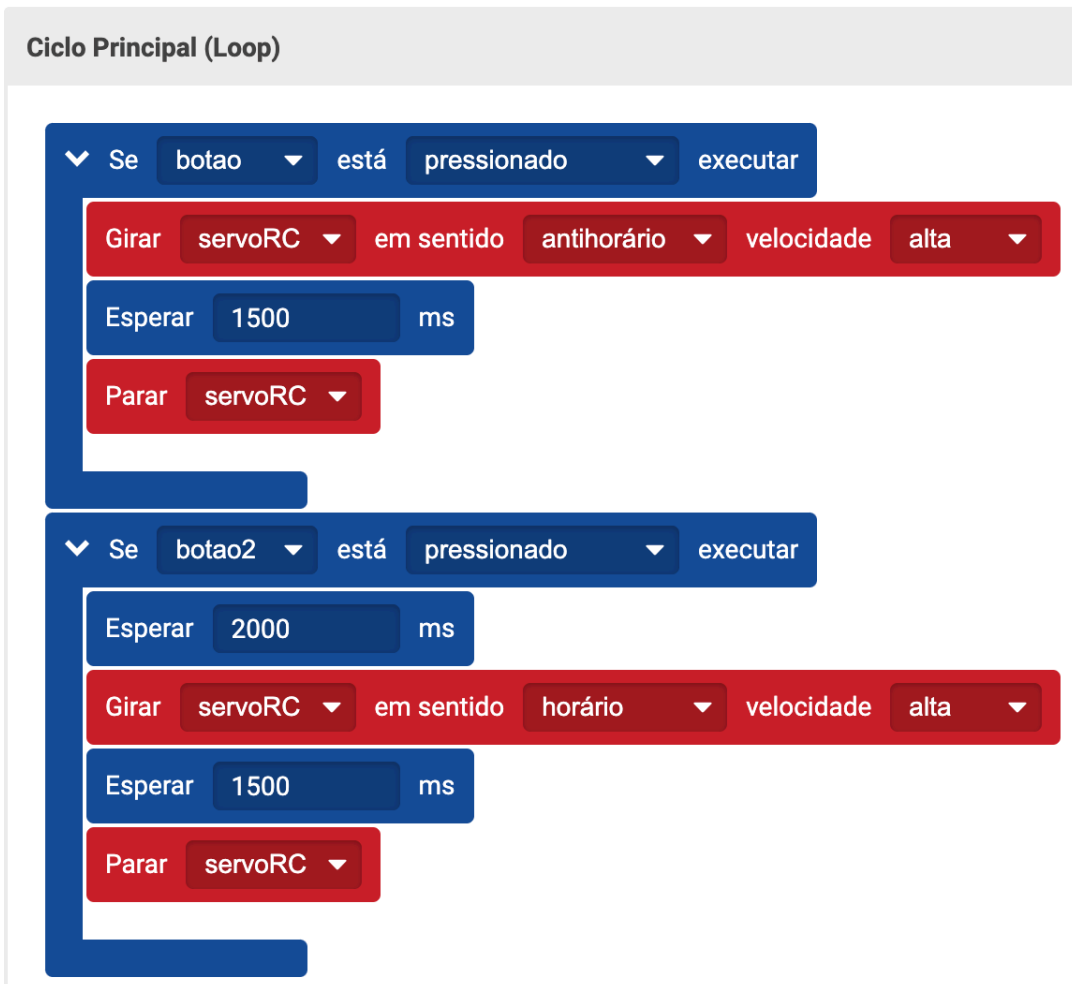


Mal tenham a programação para a abertura da porta preparada, devem integrá-la com um botão, para que a porta abra quando se pressiona o botão.

Neste exemplo, colocamos o botão atrás da abertura de uma fechadura, para que apenas uma chave com essa forma possa abrir a porta.



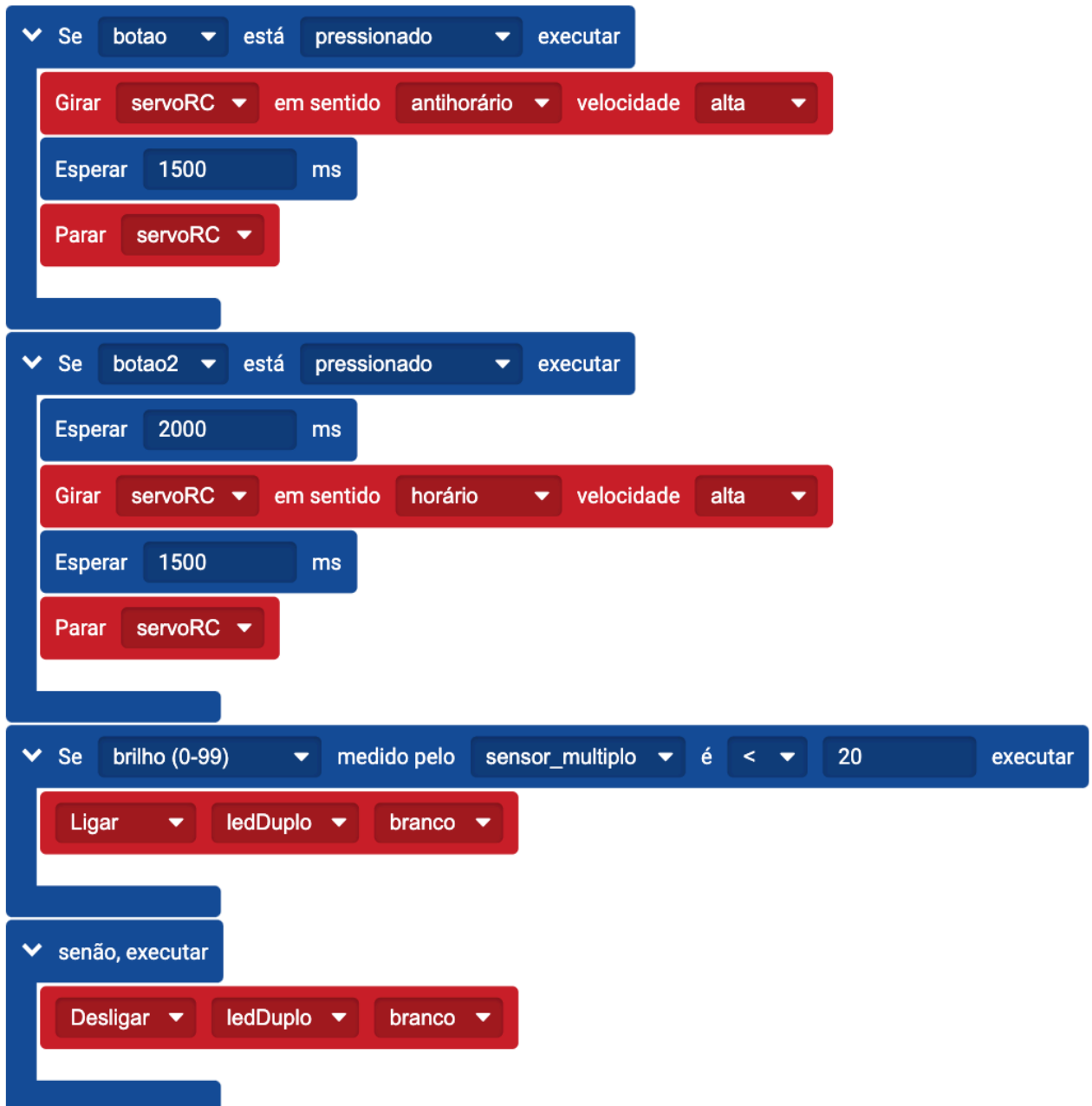
Terão de criar uma programação com a seguinte função: quando se pressiona o botão, o motor Servo de Rotação Contínua começa a girar e abre a porta e quando terminar a ação pára. Quando a porta estiver aberta, entramos na garagem e pressionamos outro botão interno para fechar a porta.



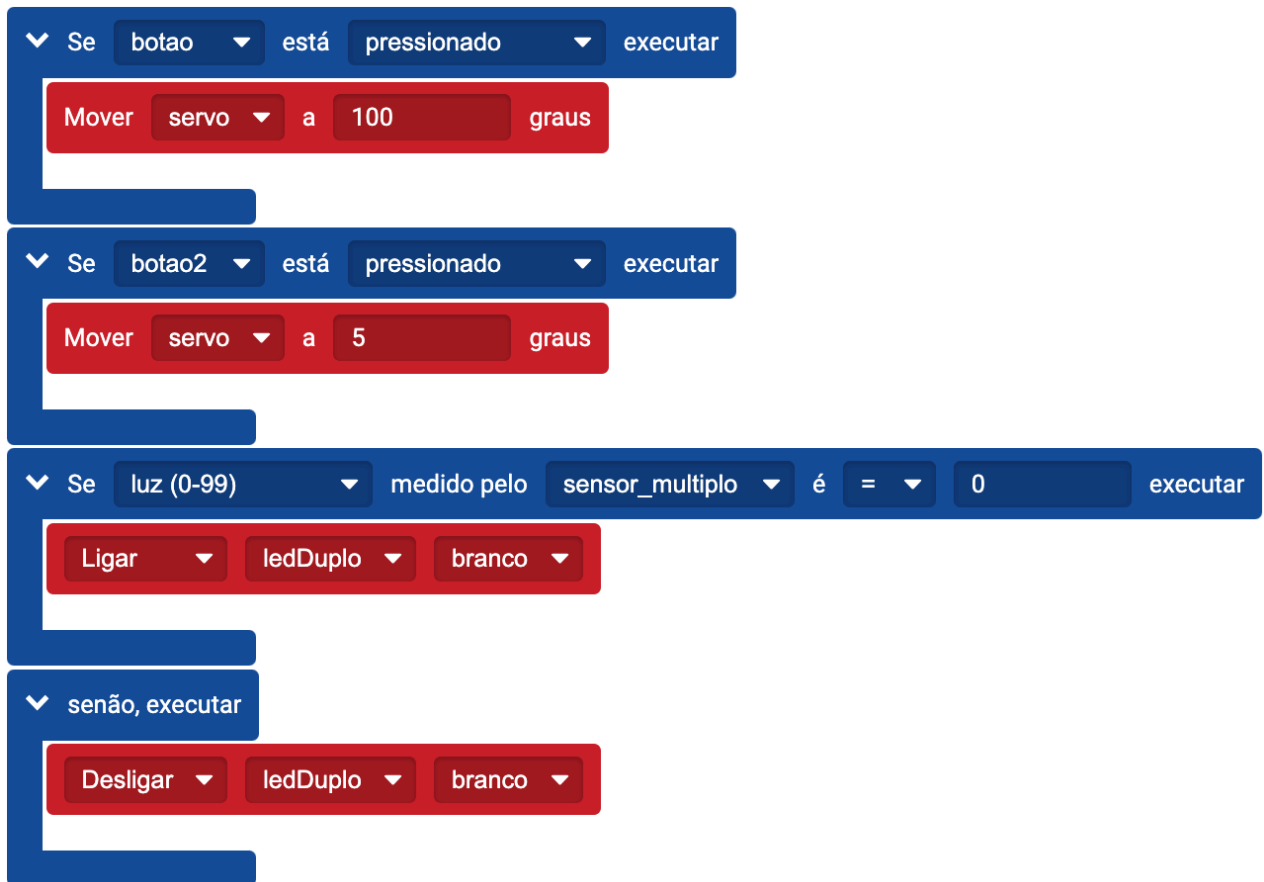
Quando tivermos a programação dos dois elementos (luz inteligente e porta automática) preparada, os alunos deverão juntar os dois programas e fazê-los trabalhar ao mesmo tempo. Para fazer isso, devem unicamente adicionar a programação da luz debaixo da programação da porta.

A programação completa para este exemplo usando um **motor Servo RC** seria a seguinte:

Ciclo Principal (Loop)



Se estivermos a usar o **motor Servo**, neste caso a programação poderia ser a seguinte:



Se os alunos tiverem terminado de programar os dois elementos propostos, pediremos que melhorem a sua casa programando outros elementos da sua imaginação.

Podem-se programar algumas melhorias com mais elementos do kit de robótica, como:

- Alarme.
- Cortinas ou persianas automáticas.
- Elevador.
- Alimentador automático para animais de estimação.
- Campainha para a casa.
- Usar outro sensor múltiplo que detete o carro a passar e abra a porta.

Planificação sugerida:

1ª Sessão: Experimentar o kit: ligar e pôr a piscar um LED e ver os valores de luz da sala de aula com ajuda do sensor de luz. Programar uma fonte de luz inteligente.

2ª Sessão: Construir uma casa. Desenhar um candeeiro ou poste de iluminação e inserir uma luz inteligente.

3ª Sessão: Experimentar com o sensor de distância, motor Servo de Rotação Contínua e com o motor Servo. Programar uma porta automática.

4ª Sessão: Melhorias e correção de erros.



Contacto e Apoio pedagógico: retotech@fundacionendesa.org