

# Projeto 1:

## Construa e programe a sua casa domótica



### ***Descrição do projeto:***

*Com este guia, aprenderá a construir e programar uma porta automática e um candeeiro inteligente.*

***Nível de dificuldade: Fácil***

***Tempo estimado: 4 horas***

### **Materiais:**

- Kit de robótica
- Bitbloq robotics
- Caixa de cartão
- Corda
- Fita adesiva
- Tesoura
- Cola

### **Opcionais:**

- Impresora 3D
- BlocksCAD / FreeCAD / Tinkercad / Bitbloq 3D

## Construa a sua casa domótica

Neste projeto, os alunos têm de construir uma casa com uma porta automática e uma

lâmpada inteligente.

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

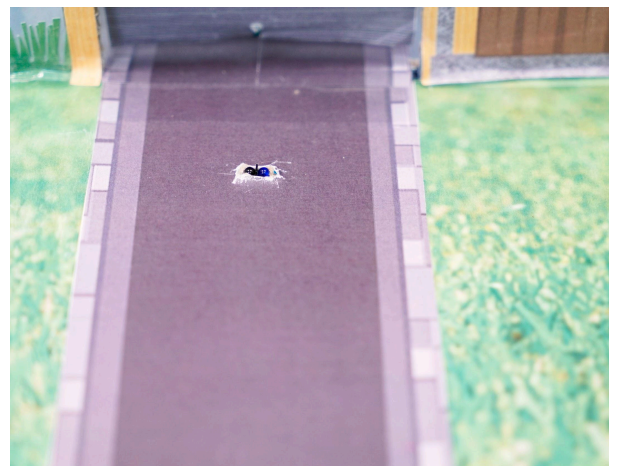
A seguir, é mostrado um exemplo do exterior de uma casa, na qual foi criada uma porta de garagem automática e um poste de iluminação inteligente que ilumina o jardim.



Os elementos da casa (jardim, estrada, casa...) podem ser desenhados com um editor de imagens, pesquisando na Internet imagens Creative Commons (sem direitos de autor) para imprimir ou pintar sobre o cartão ou material utilizado.

Para decorar, foi projetado em 3D um carro e um poste de iluminação, no qual serão incorporados os componentes necessários para criar uma luz inteligente.

Para automatizar a abertura da porta da garagem, foi colocado um sensor IR sob a estrada para que este detecte a presença de um carro.

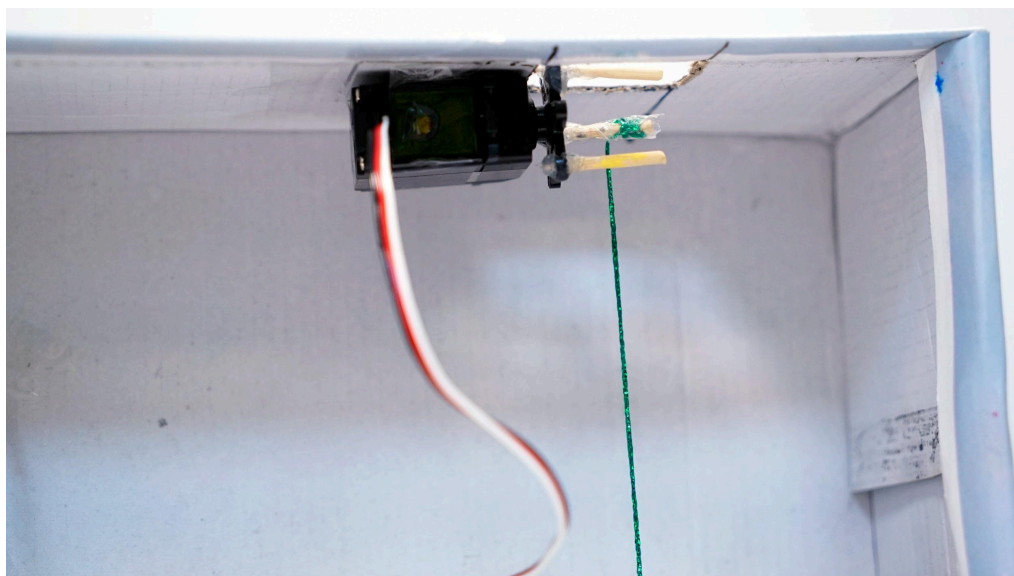


Para abrir a porta, primeiro teremos de recortá-la e, em seguida, utilizar algum dos componentes do kit de robótica para puxá-la. Podemos utilizar dois componentes:

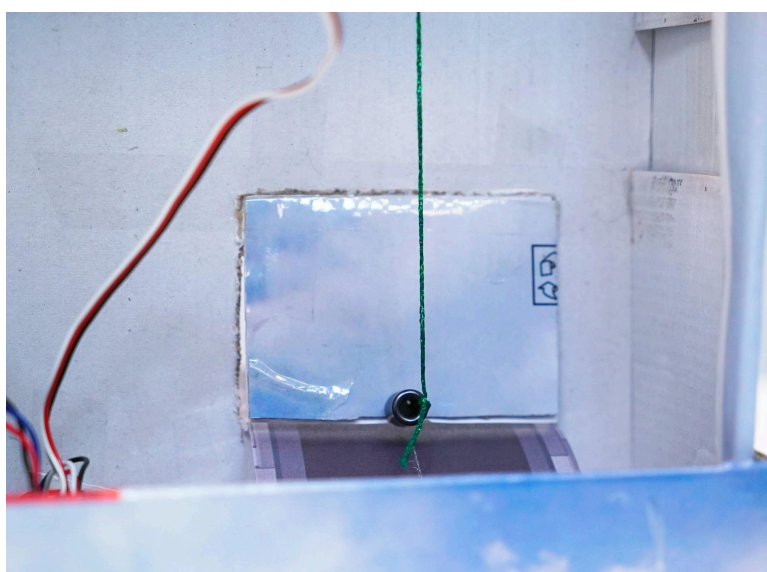
- 1) Um servo de rotação contínua, amarrando uma corda à porta para que ela se enrole e abra.

Colamos o servo de rotação contínua na parte superior da caixa e, para que a corda possa ser enrolada, colocamos uma cabeça na qual colamos três varas (nas quais a

corda será enrolada). Amarramos a corda a uma das varas, conforme mostrado na imagem a seguir:

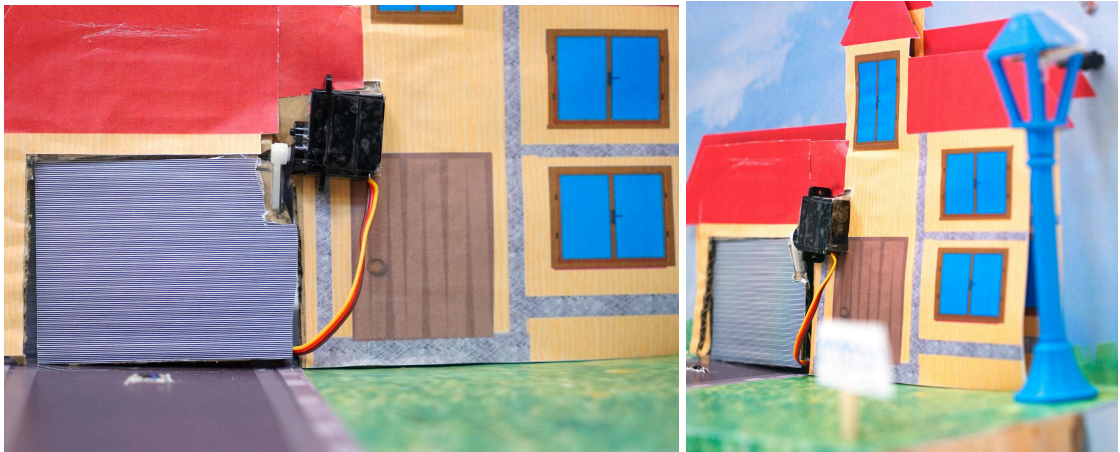


Por último, adicionaremos um peso à parte de trás da porta para que esta feche completamente.



2) Um minisservo, colando o cabeçote à porta, conforme mostrado na imagem a seguir:



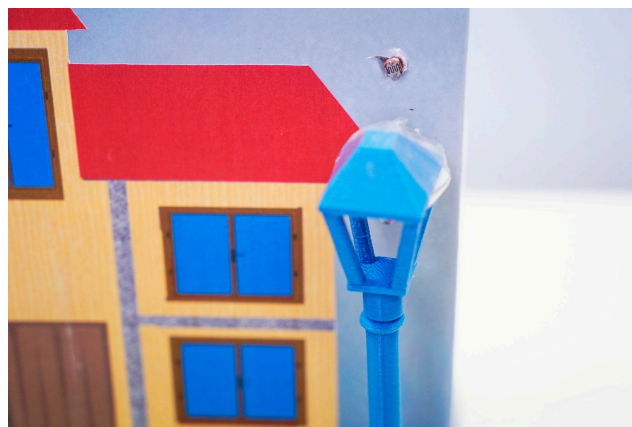


Para a lâmpada inteligente, foi procurado no Thingiverse o design de um poste de iluminação  
<http://www.thingiverse.com/thing:1361590>  
 e impresso em 3D, mas podemos pedir aos alunos que desenhem a sua própria lâmpada ou poste de iluminação em 3D para imprimir posteriormente.



Um LED e um sensor de luz (LDR) foram incorporados a essa lâmpada.

Na caixa, acima da lâmpada, foi feito um pequeno orifício e introduzida a fotorresistência do LDR, para que ele possa verificar a luz existente. Também foi feito um orifício na caixa abaixo da lâmpada para passar os cabos.

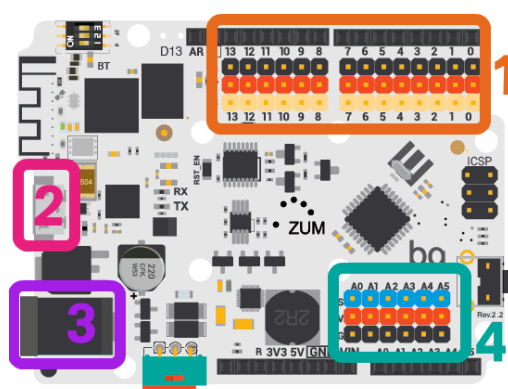


# Experimentando os componentes do kit

Antes de começar a programar a casa domótica, sugerimos fazer uma breve introdução e alguns exercícios simples para que os alunos aprendam a programar os componentes do kit de robótica que serão utilizados.

Começaremos explicando o que é a placa controladora. Este é o dispositivo que armazena todas as informações que recebe através dos sensores e que controla como agir. Podemos pedir aos alunos que procurem a placa no kit e aproveitar para explicar as suas partes:

- 1. Pinos digitais:** onde ligaremos os componentes digitais (aqueles que têm apenas dois valores: 0 e 1).
- 2. Ligação USB:** que utilizaremos para ligar ao computador.
- 3. Entrada de alimentação:** onde ligaremos as pilhas ou baterias.
- 4. Pinos analógicos:** onde ligaremos os componentes analógicos (aqueles que podem ter valores entre 0 e 1023).



A seguir, explicaremos que existem vários tipos de componentes, que podemos dividir em sensores e atuadores:

- Um **sensor** é um componente que nos fornece informações sobre algo que ocorre ao nosso redor, como, por exemplo, o nível de luz, a temperatura, um obstáculo próximo, etc. Poderíamos dizer que eles são equivalentes aos nossos órgãos sensoriais (visão, tato, audição, etc.).
- Um **atuador** é um componente que pode realizar alguma ação, como emitir uma luz, um som ou mover-se. Poderíamos dizer que são equivalentes à fala, aos membros, etc.

Também podemos classificar os componentes de acordo com o uso de sinais digitais ou analógicos:

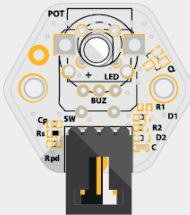
- Um componente digital é aquele que só consegue diferenciar entre dois estados (0/1, verdadeiro/falso, ligado/desligado). Por exemplo, um botão está pressionado ou não, um LED está aceso ou apagado, etc.
- Um componente analógico é aquele que pode ter mais de dois valores. Por exemplo, a temperatura, a quantidade de luz, etc. Um sensor analógico

geralmente pode ler valores entre 0 e 1023.

Depois de explicar tudo isso, faremos exercícios simples para experimentar os componentes que serão necessários para a sua casa domótica.

Começaremos programando o LED:

Pediremos aos alunos que procurem um LED no kit. Perguntaremos o que eles acham que 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 LEDs são baratos, consomem muito pouca energia e podem ser muito luminosos. Tem apenas dois valores: ligado ou desligado, pelo que é digital. Devemos ligá-lo a um dos pinos digitais da nossa placa.

Devemos indicar que observem os cabos de ligação do LED. Verão três cabos: um branco (sinal), um vermelho (tensão) e um preto (terra). Para os ligar, as cores dos cabos têm de coincidir com as dos pinos da placa.



A seguir, vamos ensinar-vos a programar o acendimento de um LED. Para isso, pedimos que liguem a placa ao computador usando o cabo USB, abram o Bitbloq ([bitbloq.cc](http://bitbloq.cc)) e cliquem em Novo - documento Bitbloq Robotics.

 Novo documento



Bitbloq Robotics Jr



Bitbloq Robotics

A interface está dividida em quatro guias principais localizadas à esquerda:

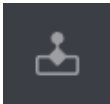


**Hardware:** onde escolheremos qual placa vamos usar e quais componentes utilizaremos.

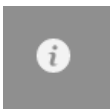


**Blocos:** onde programaremos o nosso projeto por blocos. Esta área está dividida em três partes:

- Variáveis globais e funções: Permite declarar variáveis globais e criar funções.
- Instruções iniciais (Setup): Permite configurar ações que são executadas apenas uma vez no início.
- Loop principal: Tudo o que estiver dentro será executado em loop indefinidamente.



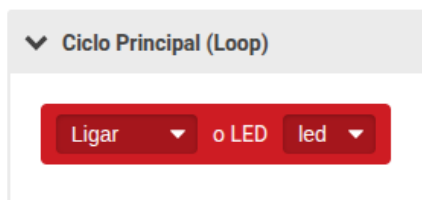
**Diagramas:** onde programamos o nosso projeto por diagramas de fluxo. O que programarmos aqui será refletido em Blocos e vice-versa.




**Informações sobre o projeto:** onde adicionaremos as informações sobre o projeto.

Indique aos alunos que, no separador Hardware, selecionem a placa que vamos utilizar (**Zum Core ou Zum Core 2.0**), arrastando-a para o centro do ecrã. Em seguida, eles deverão conectar o LED ao mesmo pino ao qual o conectaram fisicamente. Devemos alertar os alunos para que evitem conectar os componentes nos pinos 0 e 1, pois esses pinos são os que a placa utiliza para se comunicar com o computador e podem causar problemas.

Depois de fazer as ligações, vá para a parte Blocos e programe na secção Loop principal da seguinte forma:



Em seguida, clique no botão Carregar , localizado na parte superior direita, para que o programa seja carregado na placa e possam observar que o LED acende.

Após realizar este exercício, devemos pedir-lhes que tentem programar o LED para piscar, para o que só terão de programar para que acenda e apague. A programação seria semelhante à seguinte:



Geralmente, os alunos programam Ligar e Desligar o LED sem utilizar o bloco Esperar, e costumam ficar surpreendidos quando não funciona. Aproveitaremos para explicar que o programa é executado a uma velocidade muito alta e que, na verdade, está a funcionar, mas acende e apaga tão rapidamente que não conseguimos perceber. Para poder observar como acende e apaga, é necessário incorporar uma pequena espera, cuja duração (em milissegundos) pode ser modificada conforme desejado.

Em seguida, programaremos o sensor de luz ou LDR:

Pediremos aos alunos que procurem o sensor de luz no kit e, assim como no caso do LED, perguntaremos o que eles acham que é, um componente digital ou analógico, e se é um sensor ou um atuador.



O **sensor de luz**, também conhecido como LDR ou fotorresistência, é um sensor analógico que nos fornece uma medida da intensidade da luz. Como todos os sensores analógicos, ele pode fornecer valores entre 0 e 1023, embora o do kit retorne valores entre 0 (escuridão máxima) e 800 (iluminação máxima).

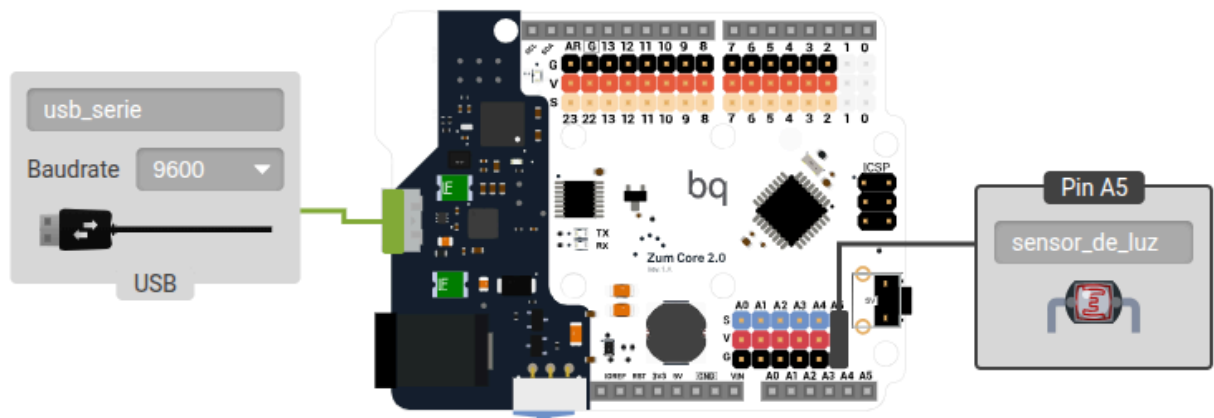
Para verificar como funciona este sensor, faremos com os alunos um programa que permita ver pela porta série do computador os valores que ele recebe.



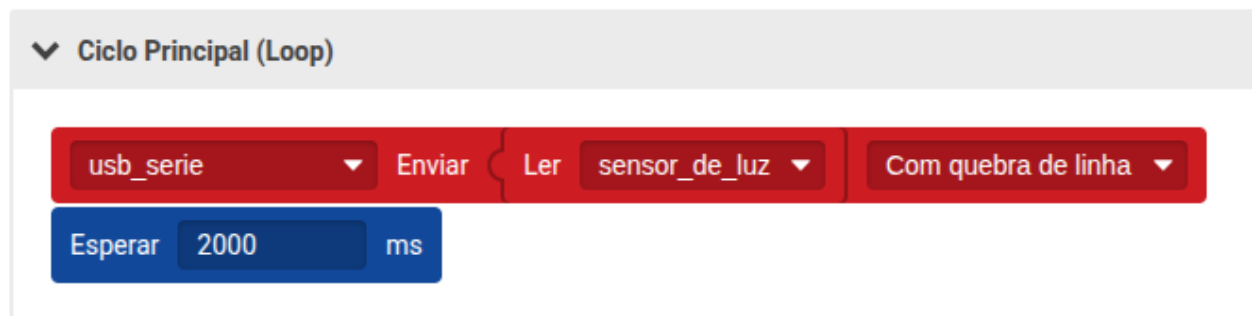
A **porta serial** é o meio que a placa controladora utiliza para comunicar com outro dispositivo, como o computador ou o telemóvel. É muito útil para verificar rapidamente o valor de um sensor ou de uma variável, ver o estado de um programa e, em definitiva, corrigir e compreender melhor os possíveis erros que este possa ter.

Deve ligar o sensor de luz (física e virtualmente) e a porta série, correspondente ao USB, como mais um componente na secção Hardware do Bitbloq, da seguinte forma:





A seguir, começaremos a programar na secção Loop principal (Loop) que mostre pela porta série a leitura do sensor de luz e uma espera de alguns segundos:

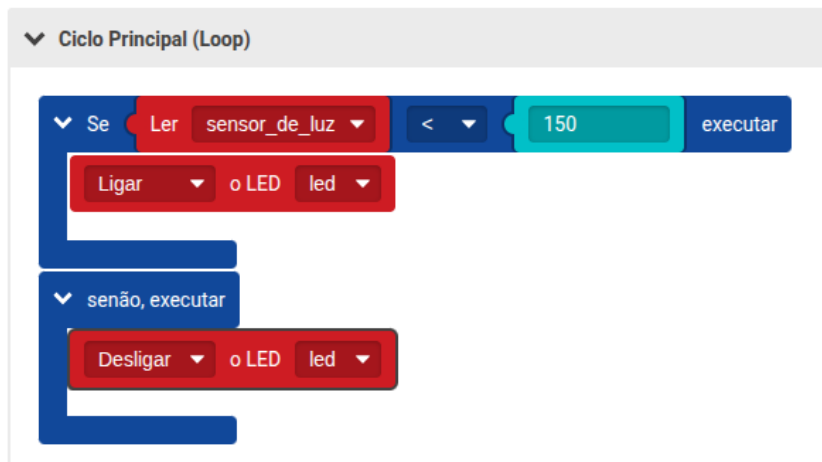


Em seguida, carregamos o programa na placa e clicamos em Ver→ Monitor porta série, na parte superior esquerda da interface. Desta forma, os valores de luz detectados pelo sensor de luz na sala de aula aparecerão numa janela pop-up.

Para verificar como os valores mudam, podemos pedir aos alunos que cubram o sensor de luz e, em seguida, o coloque perto de uma fonte de luz para observar como os valores diminuem ou aumentam.



Por último, proporemos aos alunos que criem uma lâmpada inteligente para suas casas. Para isso, eles terão que programar o LED para acender apenas quando o sensor de luz detectar que é noite ou que está escuro. Para programá-lo, eles deverão usar uma sentença condicional, de modo que, se o valor detectado pelo sensor de luz for menor que X (por exemplo, 150), o LED acenda e, se o valor for maior, ele se apague. A programação deve ficar semelhante à seguinte:



O valor da luz dependerá da iluminação da sala onde nos encontramos. Por isso, antes de programar uma luz inteligente, é importante verificar através da porta série a quantidade de luz que existe na nossa sala de aula.

A seguir, serão apresentados os componentes que servirão para programar a porta automática. Para isso, realizaremos com os alunos algumas atividades simples para que aprendam a programar o **sensor infravermelho (IR)**, o **minisservo** e o **servo de rotação contínua**.

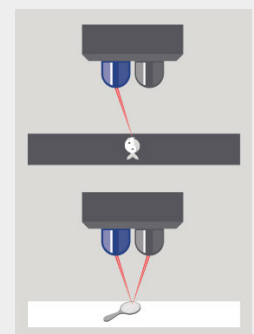
Começamos programando o sensor infravermelho ou IR:



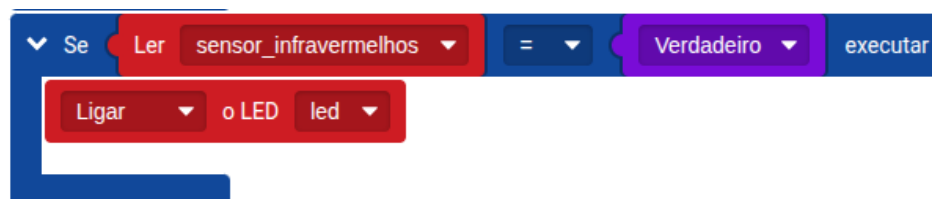
Um sensor infravermelho ou IR é um dispositivo que emite luz infravermelha, detectando a quantidade de luz refletida. Desta forma, é capaz de diferenciar entre branco e preto.

Este componente é um sensor digital, pois retorna 0/1 ou verdadeiro/falso.

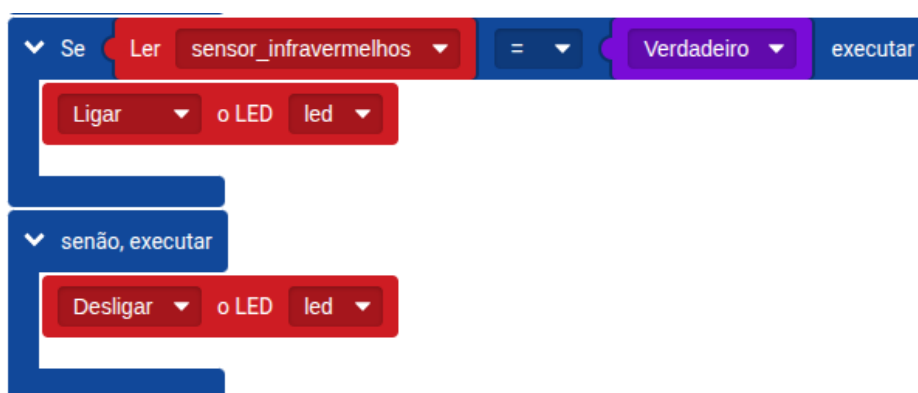
Funciona de forma que retorna 1 ou verdadeiro quando detecta a luz refletida no alvo e 0 ou falso quando não a detecta, ou seja, quando a cor é preta.



Para saber como funciona este componente, pediremos aos alunos que programem que, quando o sensor infravermelho (IR) detectar que há algo, ou seja, quando detectar 1 ou verdadeiro, um LED acenda.



É possível que os alunos não programem o que o LED deve fazer quando o sensor IR detecta 0 ou falso. Para isso, teremos que programar com o bloco de caso contrário, o LED se apaga:

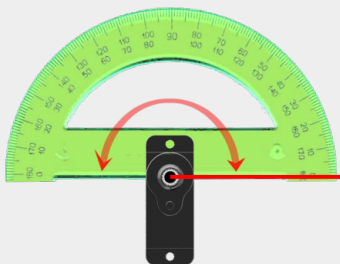


Depois de termos a programação, para verificar se funciona, teremos que pegar um papel preto e colocá-lo no IR. Quando o papel estiver sobre o IR, o LED acenderá; se o retirarmos, ele se apagará.

Na programação da porta automática da casa, utilizaremos o IR para indicar que, se detectar que há um carro, a porta se abrirá, ou seja, ativará o minisservo ou o servo de rotação contínua.

Em seguida, programaremos o **minisservo**:

Pediremos aos alunos que procurem este componente no kit e explicaremos o que é.



O **Miniservo** é um pequeno motor capaz de girar entre 0 e 180 graus, pelo que não consegue dar voltas completas. É um atuador e um componente digital.

O interessante neste tipo de motores é que somos nós que decidimos para que posição ou ângulo devem mover-se.

Proporemos um exercício para introduzir o funcionamento do minisservo, que consiste em programar o movimento do minisservo nos ângulos 0°, 45°, 90°, 135°, 180°.

Para saber a posição do servo, colocaremos uma das suas cabeças e ligá-la-emos a um pino digital da placa. Depois disso, para saber qual é a posição correta, programaremos no Setup (para que isso seja feito apenas uma vez ao executar o programa) que o minisservo vá para uma posição, por exemplo, 90°. O que normalmente acontece é que a cabeça do servo está mal colocada, por isso vamos removê-la e colocá-la voltada para o centro, procurando que fique num ângulo de 90°.

Em seguida, pediremos aos alunos que programem o minisservo para ir para as diferentes posições (0°, 45°, 90°, 135° e 180°). É importante que coloquem um bloco Esperar abaixo de cada bloco de movimento, pois, caso contrário, o minisservo se moverá tão rapidamente que só será possível ver o ângulo inicial e final.

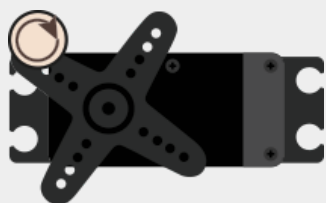
Além disso, adicionar uma espera abaixo de cada bloco protege o miniservo de movimentos bruscos.



Para programar a porta automática com o Miniservo, podemos fazer com que ela se mova em diferentes ângulos, por exemplo, de 10° a 100°, para que se abra progressivamente.

**Nota:** para não forçar os mini servos, é recomendável não programá-los nas suas posições extremas, ou seja, nem em 0° nem em 180°, e fazê-lo em 10° ou 170°.

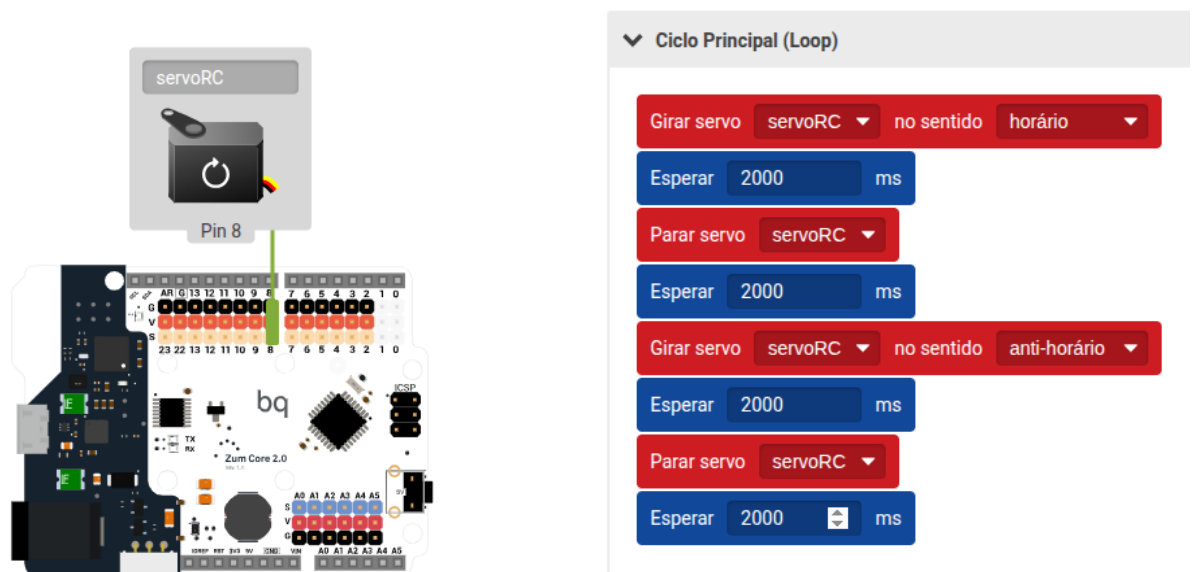
Outra forma de fazer com que a porta se mova é com o **servo de rotação contínua**.



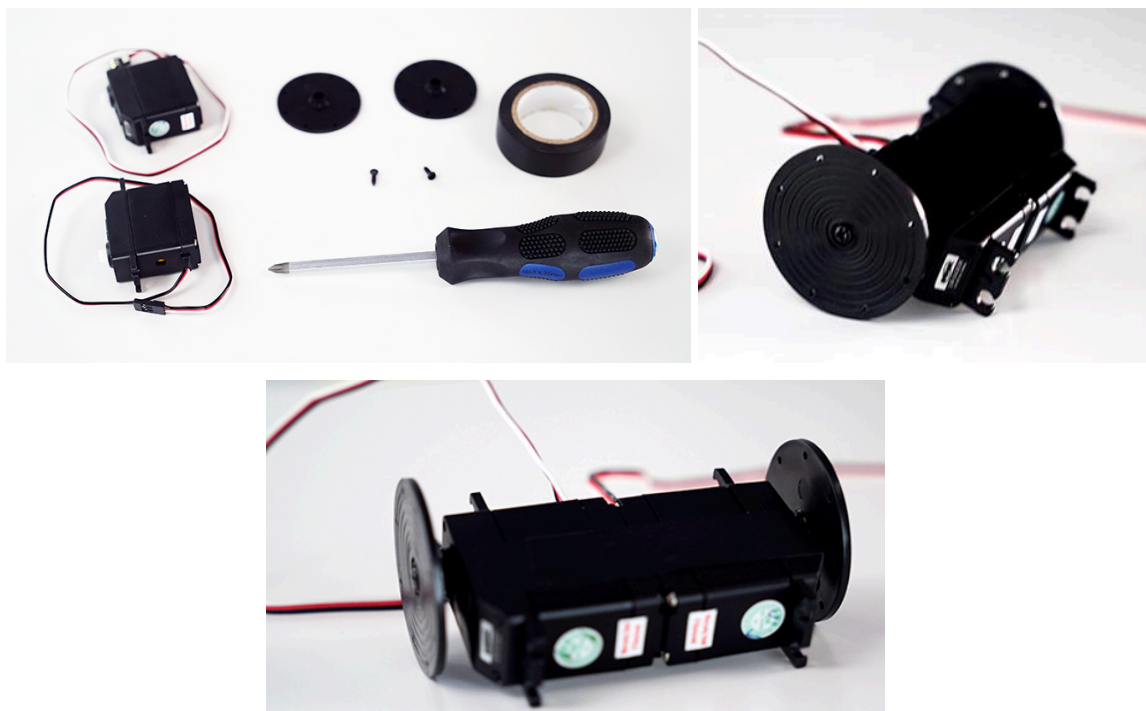
Um **servo de rotação contínua** é um motor cujo circuito eletrônico nos permite controlar a direção de rotação. Ao contrário do miniservo, ele não para em uma posição, mas gira continuamente.

Para experimentar este componente, podemos pedir aos alunos que programem o servo para avançar, parar, recuar e parar novamente. Entre cada bloco de movimento, eles devem adicionar uma espera para indicar quanto tempo essa ação ou movimento

deve durar. Para isso, será feita uma programação semelhante à seguinte:



Depois disso, podemos pedir que construam um pequeno robô no qual coloquem dois servos com fita adesiva nas suas cabeças (como se fossem rodas) e programem o seu movimento. Um exemplo é o seguinte:

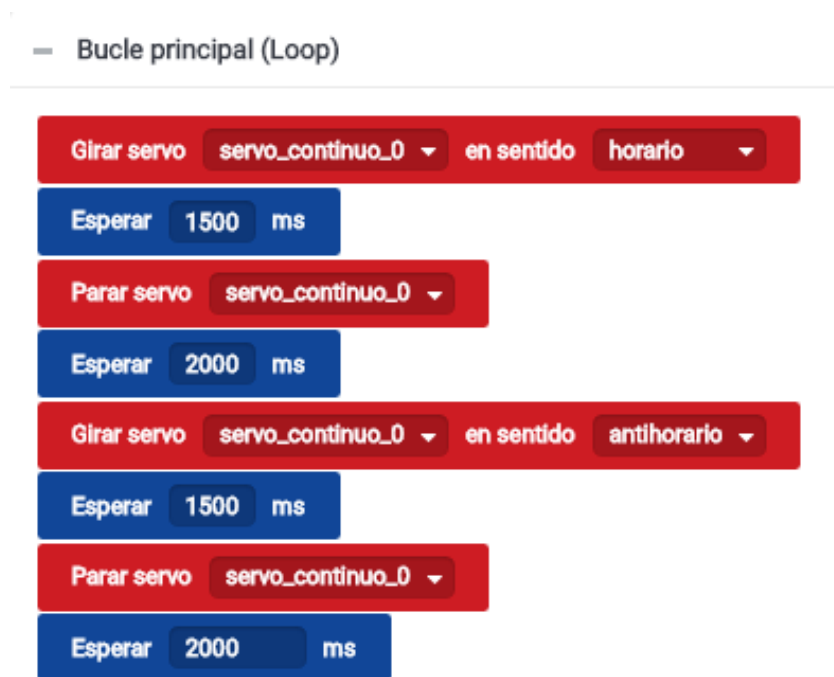




## Programa sua casa domótica

Depois de conhecerem o funcionamento destes componentes, terão de programar a porta que construíram. Para isso, a programação irá variar consoante se utilize um minisservo ou um servo de rotação contínua.

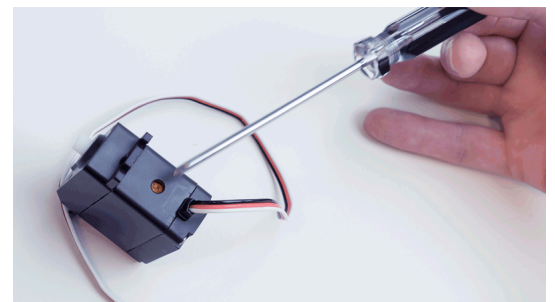
No exemplo, foi utilizado um servo de rotação contínua, de modo que este puxa uma corda que, por sua vez, sobe ou desce a porta. A programação utilizada é a seguinte:



O **servo de rotação contínua** gira durante 1,5 segundos para abrir a porta. Em seguida, pára e espera 2 segundos para que o carro passe. Por fim, fecha a porta (girando no sentido contrário durante 1,5 segundos) e pára por mais 2 segundos. Por estar dentro do Loop, a programação será repetida continuamente.

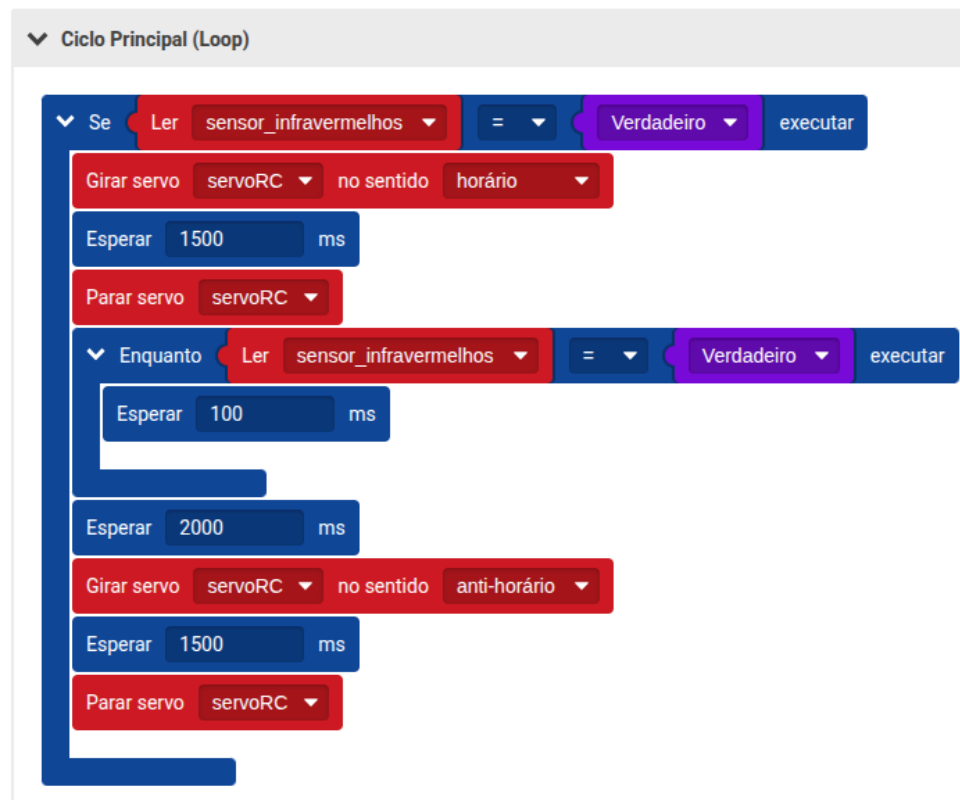
A programação irá variar dependendo da posição do servo e do comprimento da corda, pelo que os alunos terão de ir modificando o tempo de espera até conseguirem que a porta se abra e feche corretamente.

É possível que o servo se mova quando deveria estar parado e seja necessário calibrá-lo. Para isso, devemos criar um programa que pare o servo e carregá-lo na placa. Verificaremos se ele se move e, caso isso ocorra, pegamos uma chave de fenda plana e giramos lentamente em uma das direções até que ele pare.



Depois de terem a programação da abertura da porta, deverão integrá-la com o **sensor IR**, de modo que a porta só se abra quando detectar que há um carro.

A programação que terão de criar terá a seguinte função: quando o IR deteta que há um carro (verdadeiro), o **servo de rotação contínua** gira para abrir a porta e pára quando termina. Uma vez que a porta está aberta, temos de indicar que, enquanto o IR detetar que o carro está em cima, deve esperar que o carro passe.



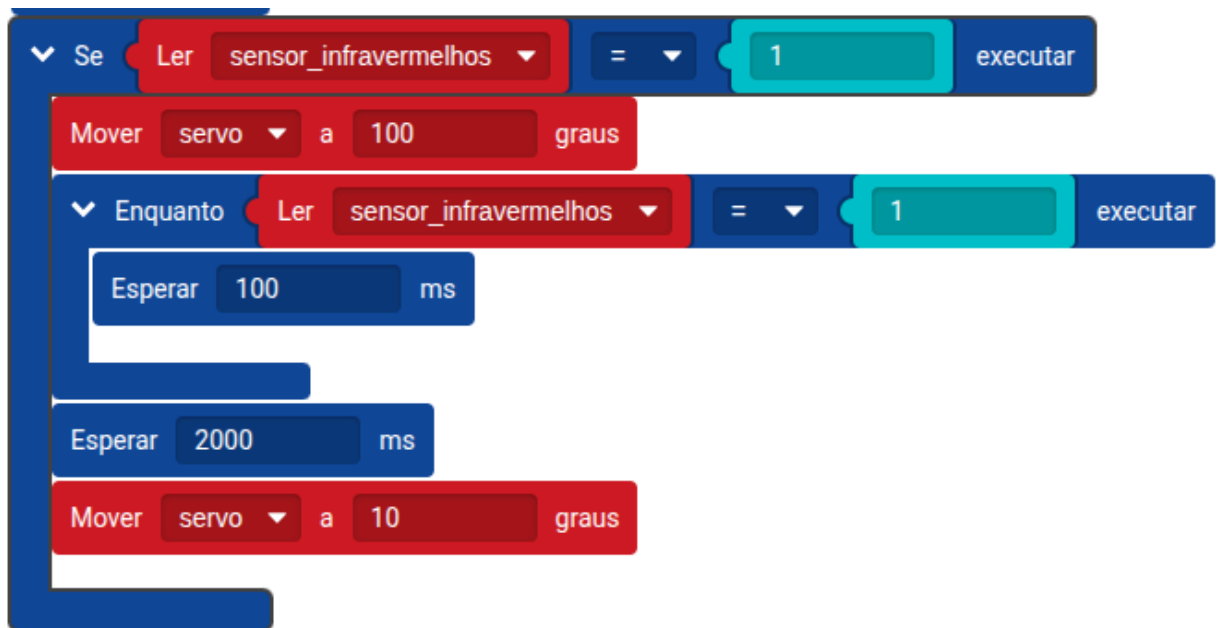
Quando não há um carro, ou seja, o IR detecta falso ou 0, ele sai do loop While e executa os blocos seguintes, ou seja, espera um pouco (2 segundos) para dar tempo ao carro de passar, gira o servo para baixar a porta e, quando termina de fechar, para. Desta forma, quando voltar a detetar que há um carro, realizará o mesmo processo.

Depois de terem a programação dos dois elementos (luz inteligente e porta automática), devem juntá-los para que possam funcionar ao mesmo tempo. Para isso, basta adicionar a programação da luz abaixo da programação da porta. A programação completa do exemplo é a seguinte:



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

Se tivéssemos usado o miniservo, teríamos que substituir o interior do condicional do sensor IR por estes blocos:



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

Algumas melhorias que podem ser programadas são as seguintes:

- Alarme.
- Cortinas ou persianas automáticas.
- Elevador.
- Comedouro automático para animais de estimação.
- Campainha.

Para programar essas melhorias, eles precisarão conhecer os demais componentes do kit. Para isso, podemos pedir aos alunos que vejam as diferentes atividades e tutoriais disponíveis em [bitblog.cc](http://bitblog.cc).

### **Tempo sugerido:**

1ª Sessão: Experimentação com o kit: acender e piscar um LED e ver os valores de luz da sala de aula com o sensor de luz. Programar uma luz inteligente.

2ª Sessão: Construção de uma casa. Design de candeeiro ou poste de iluminação para incorporar luz inteligente.

3ª Sessão: Experimentação com sensor de distância, servo de rotação contínua e miniservo. Programação de uma porta automática.

4ª Sessão: Incorporação de melhorias e correção de erros.