

Proxecto 1:

Constrúe e programa a túa casa domótica co Zum Kit Junior



Descrición do proxecto:

Con esta guía aprenderás a construír e programar unha porta automática e unha lámpada intelixente.

Nivel de dificultade: Fácil

Tempo estimado: 4 horas

Materiais:

- Kit de robótica Zum Kit Junior
- Bitbloq robotics
- Caixa de cartón
- Corda
- Cinta adhesiva
- Tesoiras
- Pegamento
- Velcro
- Bitbloq 3D (opcional)
- Impresora 3D (opcional)

Constrúe a túa casa domótica

Neste proxecto os alumnos teñen que construír unha casa que teña unha porta automática e unha lámpada intelixente.

Pódese construír tanto o interior coma o exterior da casa. É importante que deixemos esta parte á elección dos alumnos, fomentando a súa creatividade e implicación no proxecto.

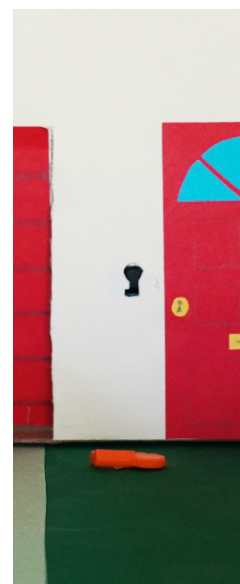
A continuación, amósase un exemplo do exterior dunha casa, na que se creou unha porta de garaxe automática e unha farola intelixente que ilumina o xardín.



Os elementos da casa (xardín, estrada, casa...) pódense deseñar cun editor de imaxes, buscar en Internet imaxes *Creative commons* (sen dereitos de autor) para imprimir ou utilizar cartulinas e/ou folios de cores e debuxar sobre eles.

Para decorar, imprimíuse en 3D un coche e unha farola. Sobre ésta última incorporáronse os compoñentes necesarios para crear unha luz intelixente.

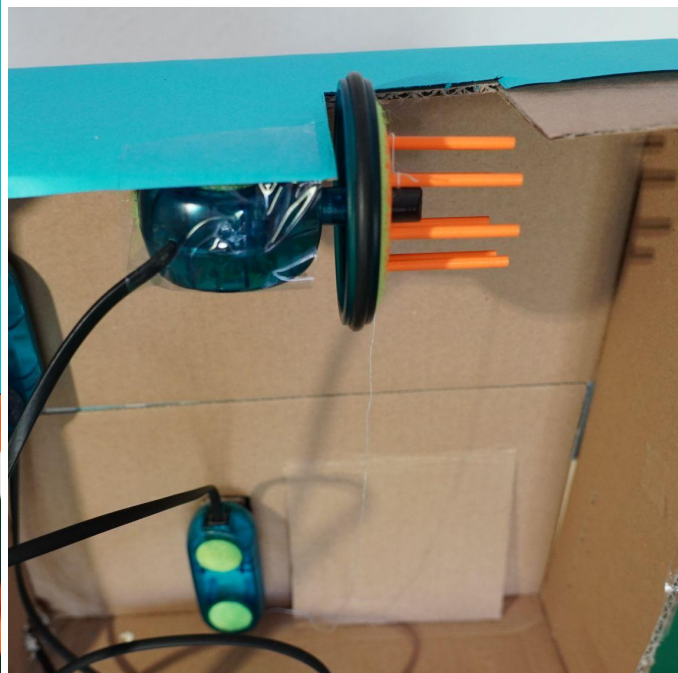
Para automatizar a apertura da porta do garaxe, colocouse unha fechadura xunto a ela cunha forma concreta, que ao introducir unha chave con esa forma fai comezar o proceso de apertura. Dentro do garaxe teremos outro botón que, ao premelo, fechará a porta.



Para a apertura da porta, primeiro deberemos recortala e logo utilizar algún dos compoñentes do kit de robótica para que tire dela. Podemos utilizar dous compoñentes:

1) Un Servo de rotación continua, atando unha corda á porta co fin de que se vaia enrollando para abríla.

Pegaremos o servo de rotación continua na parte superior da caixa. Necesitaremos que a corda se vaia enrollando, polo que teremos que imprimir en 3D o cabezal adxunto ou ben crear un con outros materiais dunha forma similar:



2) Un **Miniservo**, imprimindo o cabezal adxunto en 3D, ou ben creando un similar con outros materiais. Como podemos observar nas fotos, colocariase na parte superior da porta, para que esta poida xirar 90 graos



Para a lámpada intelixente pódese imprimir en 3D o deseño adxunto, pero podemos pedir aos alumnos que deseñen a súa propia lámpada ou farola en 3D para imprimila posteriormente, ou ben usar outros materiais.

Á devandita farola incorporóuselle un LED dobre e o Sensor múltiple, que medirá a luminosidade.

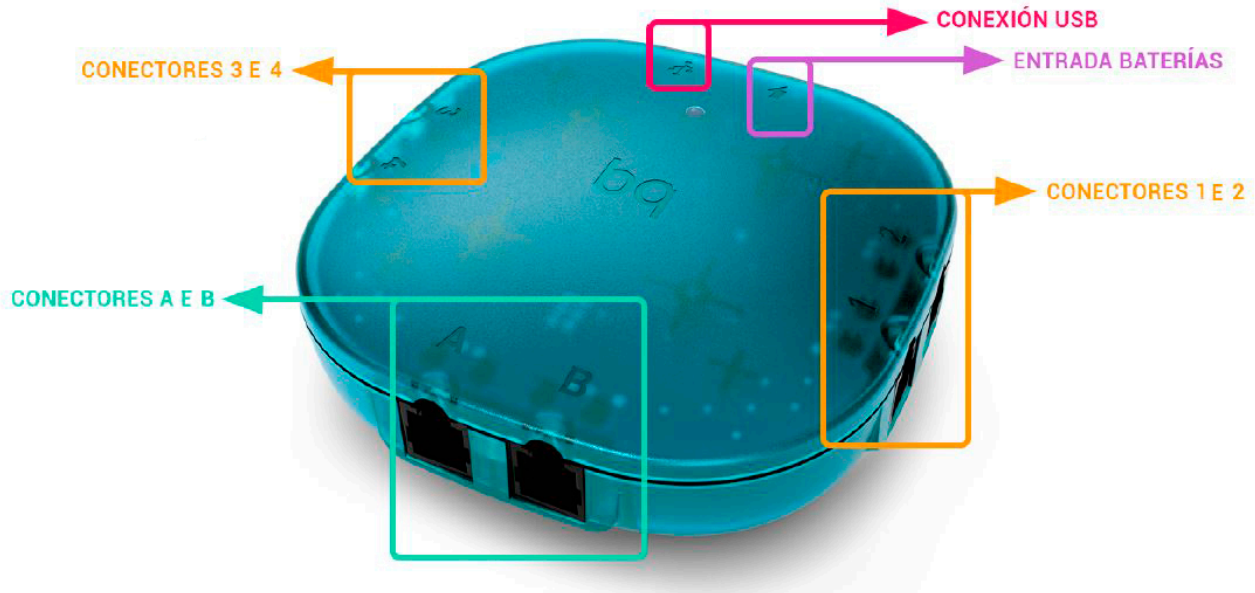
Na caixa, enriba da farola, fíxose un burato e introduciuse o Sensor múltiple, co fin de que poida comprobar a luz que hai.



Experimentando cos compoñentes do kit

Antes de comenzar a programar a casa domótica, propoñemos realizar unha breve introdución e realizar uns sinxelos exercicios co fin de que os alumnos aprendan a programar os compoñentes do kit de robótica que se van utilizar.

Comezaremos explicando que é a placa controladora. Esta é o dispositivo que garda toda a información que recibe a través dos sensores e a que controla como actuar. Podemos pedirles aos alumnos que busquen a placa no kit e aproveitar para explicar as súas partes:



Pins 1, 2, 3 e 4: Nos que podemos conectar tódolos compoñentes, **menos o Sensor múltiple eo Display**.

Pins A e B: Nos que podemos conectar tódolos compoñentes.

Conexión USB: Que utilizaremos para conectar co ordenador e/ou baterías externas.

Entrada de alimentación: Na que conectaremos as pilas ou baterías.

A continuación, explicaremos que existen diversos tipos de compoñentes, que podemos dividir en sensores e actuadores:

- Un **sensor** é un compoñente que nos proporciona información sobre algo que ocorre ao redor, como por exemplo o nivel de luz, a temperatura, un obstáculo cercano, etc. Poderamos dicir que son equivalentes aos nosos órganos dos sentidos (vista, tacto, oído, etc.).
- Un **actuador** é un compoñente que pode realizar algunha acción, como por exemplo emitir unha luz, un son ou moverse. Poderíamos dicir que son equivalentes á fala, ás extremidades, etc.

Tamén podemos clasificar os compoñentes en canto a se utilizan sinais dixitais ou analóxicas:

- Un compoñente **dixital** é aquel que só pode diferenciar entre dous estados (0/1, verdadeiro/falso, prendido/apagado). Por exemplo, un botón está pulsado ou non o está, un LED está prendido ou apagado, etc.
- Un compoñente **analóxico** é aquel que pode ter máis de dous valores. Por exemplo, a temperatura, a cantidade de luz, etc. Un sensor analóxico, xeralmente pode ler valores de entre 0 e 1023.

Unha vez explicado todo o anterior, procederemos a realizar exercicios sinxelos para experimentar cos compoñentes que necesitarán para a súa casa domótica.

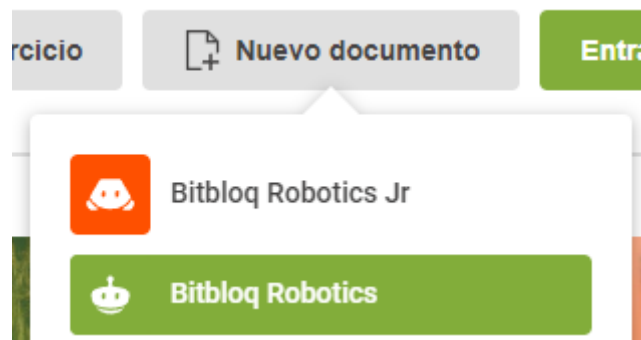
Comezaremos programando o **LED**:

Pediremos aos alumnos que busquen no kit un LED. Preguntarémolles que cren que é, un compoñente dixital ou analóxico, e se é un sensor ou actuador.



Un **LED** (*Light-Emitting Diode*) é un compoñente actuador capaz de emitir luz. Os LED son baratos, gastan moi poca enerxía e poden chegar a ser moi luminosos. Só ten dous valores: prendido ou apagado, por lo que é dixital. No caso del Zum Kit Junior, temos **LED dobres**, nos cales podemos programar dous LED: un branco e outro de cor, podendo ser vermello.

A continuación, ensinarémolles a programar que se prenda un LED. Para iso, pedirémolles que conecten a súa placa ao ordenador utilizando o cable USB, abran Bitbloq (bitbloq.cc) e fagan clic en *Novo - documento Bitbloq Robotics*.



A interface divídese en tres pestanas principais situadas á esquerda:

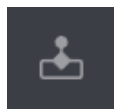


Hardware: onde elixiremos que placa imos usar e que compoñentes utilizaremos.



Bloques: onde programaremos o noso proxecto por bloques. Este área está dividida en tres partes:

- Variables globais e funcións: Permite declarar variables globais e crear funcións.
- Instrucións iniciais (*Setup*): Permite configurar accións que se executan só unha vez ao principio.
- Bucle principal (*Loop*): Todo o que se encontre dentro executarase en bucle de forma indefinida.



Diagramas: onde programaremos o noso proxecto por diagramas de fluxo. O que programemos aquí verase reflectido en *Bloques*, e viceversa.



Información do proxecto: onde engadiremos a información sobre o proxecto.

Indicarémoslles que na pestana *Hardware* seleccionen a placa que imos utilizar (Zum Junior) arrastrándoa ao centro da pantalla. A continuación, deberán conectar o LED dobre ao mesmo pin ao que o teñan conectado físicamente.

Una vez que teñen as conexións realizadas, deberán ir á parte de *Bloques* e programar no apartado *Bucle principal* (*Loop*) da seguinte maneira:

▼ Bucle principal (Loop)

Encender ▼

ledDoble ▼

blanco ▼

Como o LED dobre ten dous LED (cor e branco) terase que seleccionar o que se queira iluminar. Por exemplo, se se quixeran iluminar os dous á vez, se tería que colocar o mesmo bloque debaixo, seleccionando neste caso *cor* en lugar de *branco*.

A continuación, deberán dar ao botón *Cargar* , situado na parte superior dereita, para que o programa se cargue na placa e poidan observar que un dos LED do LED dobre se prende.

Tras realizar este exercicio, deberemos pedirilles que intenten programar que o LED dobre chisque, para o cal só terán que programar que se prenda e se apague. A programación sería algo similar á seguinte:



Xeralmente, os alumnos programan *Prender e Apagar o LED dobre* sen utilizar o bloque *Esperar*, e soe sorprendellos que non funcione. Aproveitaremos para explicarlles que o programa se executa a moitísima velocidade e realmente si está funcionando, só que se prende e se apaga tan deprisa que non podemos percibilo. Para poder observar como se encende e se apaga, hai que incorporar unha pequena espera, cuxa duración (en milisegundos) poderán modificar canto queiran.

A continuación programaremos o **sensor de luz**, un dos sensores do **Sensor múltiple**:

Pediremos aos alumnos que busquen no kit o Sensor múltiple e, ao igual que o LED, preguntarlles que cren que é, un compoñente dixital ou analóxico, e se é un sensor ou actuador.



Co **Sensor múltiple** podemos detectar a temperatura, medir distancias, distinguir cores e coñecer a cantidade de luz. Sé se poderá conectar nos pins A ou B.

O **sensor de luz** do multisensor devolve valores de entre 0 (máxima escuridade) e 99 (máxima iluminación).

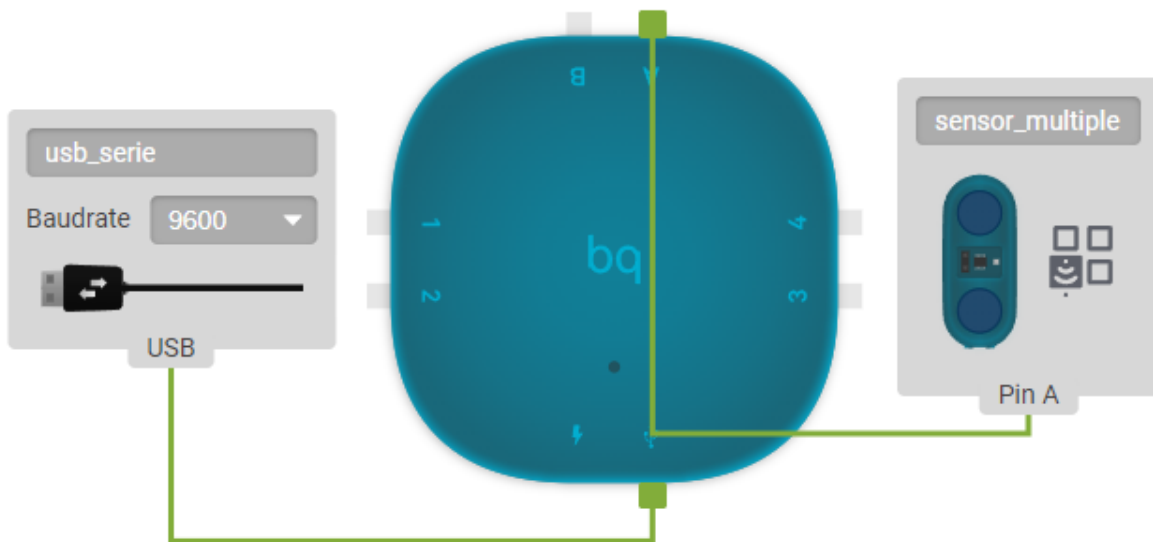
Para comprobar como funciona este sensor, faremos cos alumnos un programa que

permita ver polo porto serie do ordenador os valores que recibe.



O **puerto serie** é o medio que téñ a placa controladora para comunicarse con outro aparato como o ordenador ou o teu móvil. É moi útil para comprobar dunha ollada o valor dun sensor ou unha variable, ver o estado dun programa e, en definitiva, corrixir e entender mellor os posibles erros que este teña.

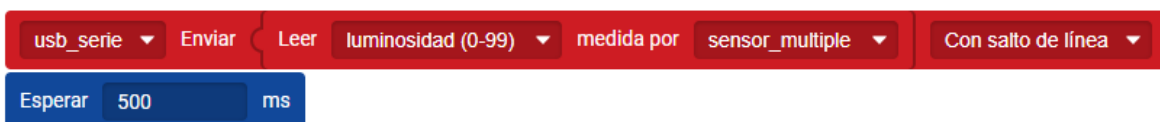
Deberán conectar o **Sensor múltiple** (física e virtualmente) e o **USB serie**, correspondente con el USB, como un compoñente máis no apartado de *Hardware* de Bitbloq da seguinte forma:



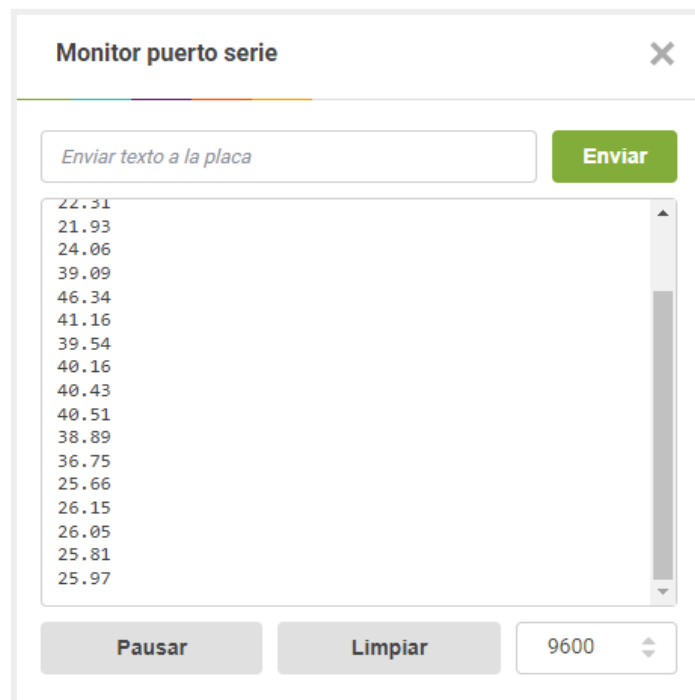
É importante recordar que o Sensor múltiple **só pode conectarse aos pins A ou B** da placa controladora.

A continuación, comezaremos a programar no apartado *Bucle principal (Loop)* que amose por porto serie a lectura do sensor de luz e unha espera duns milisegundos:

▼ **Bucle principal (Loop)**



Tras isto, cargaremos o programa á placa e prememos en *Ver*→ *Monitor porto serie*, da parte superior esquerda da interface. Desta forma, aparecerán os valores de luz detectados polo sensor de luz no aula nunha ventá emerxente.



Para comprobar como cambian os valores, podemos pedirlle aos alumnos que tapen o sensor de luz e logo o poñan preto dun foco de luz para observar como os valores disminúen ou aumentan.

Por último, proporemos aos alumnos facer a lámpada intelixente da súa casa. Para iso, terán que programar que o LED dobre só se encenda cando o sensor de luz detecte que é de noite ou haxa escuridade. Para programalo, deberán utilizar unha sentencia condicional, de maneira que se o valor detectado polo sensor de luz é menor que X (por exemplo 50), se encenda o LED dobre, e se o valor é maior, se apague. Debería quedar unha programación similar á seguinte:

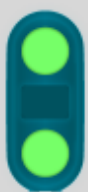


O valor de luz dependerá da iluminación da sala onde nos encontremos. Por iso, é importante antes de programar unha luz intelixente, ver por porto serie a cantidade de luz que hai na nosa aula.

A continuación introducíranse os compoñentes que servirán para programar a **porta automática**. Para iso, realizaremos cos alumnos unhas actividades sinxelas para que aprendan a programar o sensor de distancia do Sensor múltiple, o Miniservo e o Servo de rotación continua.

Como se indicou naa construción, os alumnos terán que elixir se utilizarán o Miniservo ou Servo de rotación continua.

Comezamos programando o **sensor de distancia**:



Co **Sensor múltiple** podemos detectar a temperatura, medir distancias, distinguir cores e coñecer a cantidade de luz. Só se poderá conectar nos pins A ou B.

No caso de detectar obxectos, o sensor múltiple utiliza a luz reflectida para interpretar a **distancia** que hay entre o obxecto e o sensor, ata aproximadamente 50 centímetros.

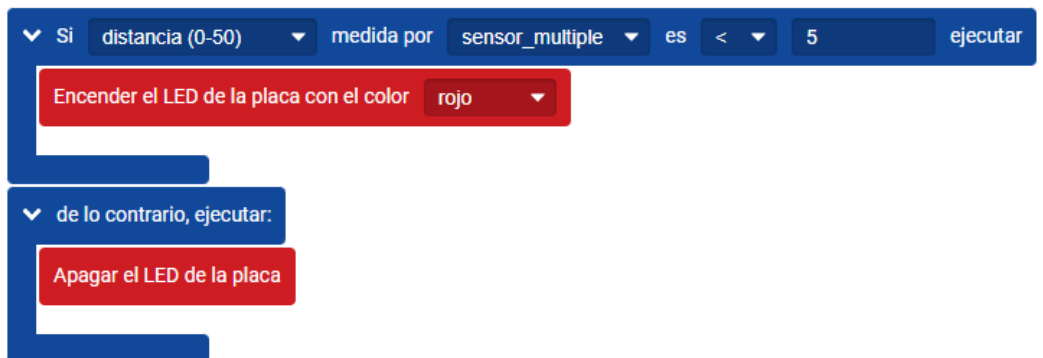
Para coñecer como funciona este compoñente, pediremos aos alumnos que programen que cando o sensor de distancia detecte un obxecto moi preto (por exemplo, máis ou menos a 5 centímetros), se prenda o LED da placa.

▼ Bucle principal (Loop)



É posible que os alumnos non programen que ten que facer o LED cando o sensor detecta que a distancia do obxecto é maior. Para iso, teremos que programar co bloque *pola contra* que o LED se apague.

▼ Bucle principal (Loop)



Na programación da porta automática da casa, utilizaremos o sensor para indicar que se detecta que hai un coche a puerta se abra, é dicir, active o Miniservo ou o Servo de rotación continua.

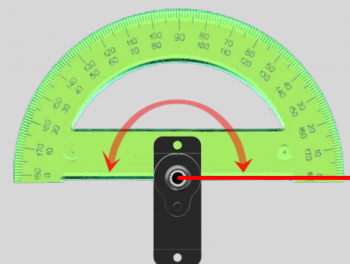
A continuación programaremos o **Miniservo**:

Pediremos aos alumnos que busquen no kit este compoñente e explicaremos que é.



O **Miniservo** é un pequeno motor que é quen de xirar entre 0 e 180 graos, polo que non pode dar voltas completas. É un actuador e un compoñente dixital.

O interesante deste tipo de motores é que nós decidimos a que posición ou ángulo deben moverse.



Proporemos un exercicio para introducir o funcionamento do Miniservo, consistente en programar o movemento do mesmo aos ángulos 0°, 45°, 90°, 135°, 180°.

Conectarémolo a un dos pins da placa. Tras isto, para saber cal é a posición correcta, programaremos no *Setup* (para que só o faga unha vez ao executarse o programa) que ol Miniservo vaia a unha posición, por exemplo 90°. Unha vez se mova a esa posición, podemos debuxar e recortar unha frecha de papel e pegala no servo apuntando a 90°. Desta forma, xa sabemos en que posición se encontra en cada momento.



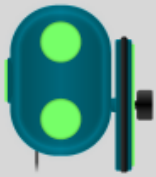
Tras isto, pediremos aos alumnos que programen que o Miniservo vaia a diferentes posicións (0°, 45°, 90°, 135° e 180°). É importante que debaixo de cada bloque de movemento poñan un bloque *Esperar*, xa que senón o Miniservomoverase tan a prisa que só se verá o ángulo inicial e final.



Ademais, engadir unha espera debaixo de cada bloque protexe ao Miniservo de movementos bruscos.

Para programar a porta automática co Miniservo, podemos facer que vaia a diferentes ángulos, dende 0° a 90°, para que se abra de forma progresiva.

Outra forma de facer que a porta se mueva é co **Servo de rotación continua**.



Un **Servo de rotación continua** é un motor cuxo circuito electrónico permítenos controlar a dirección e velocidade de xiro. A diferenza do Miniservo, non se deté nunha posición, senón que xira continuamente.

Para experimentar con este compoñente, podemos pedir aos alumnos que programen que o servo avance, se pare, retroceda e volva parar. Entre cada bloque de movemento, deberán engadir unha espera para indicar canto tempo ten que realizar esa acción ou movemento. Para iso, realizarase unha programación similar á seguinte:



Tras isto, podemos pedirilles que constrúan un pequeno robot no que peguen dous servos con cinta adhesiva cos seus cabezais (a modo de roda) e programen o seu movemento. Un exemplo é o seguinte:



Programa a túa casa domótica

Unha vez que coñecen o funcionamento destes compoñentes, terán que programar a porta que construíron. Para iso, a programación variará en función de se se utiliza un Miniservo ou un Servo de rotación continua.

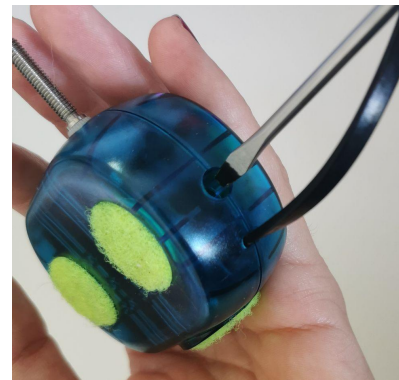
No exemplo, utilizouse un Servo de rotación continua, de maneira que este tira dunha corda que á súa vez sube ou baixa a porta. A programación que se utilizou é a seguinte:



O servo de rotación continua xira durante 1,5 segundos para abrir a porta. Despois para e espera 2 segundos para que pase o coche. Por último, cerra a porta (xirando en sentido contrario durante 1,5 segundos) e para outros 2 segundos. Ao estar dentro do *Loop* a programación repétirase continuamente.

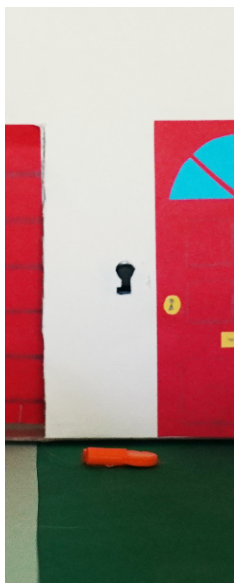
A programación variará dependendo da posición do servo e da lonxitude da corda, polo que os alumnos terán que ir modificando o tempo de espera ata conseguir que se abra e cerre a porta correctamente.

É posible que o servo se mova cando ten que estar parado e haxa que calibralo. Para iso, deberemos facer un programa que pare o servo e cargalo á placa. Comprobaremos se se move e en caso de que así sexa, colleremos un desparafusador plano e xiraremos lentamente nun dos sentidos ata que se pare.

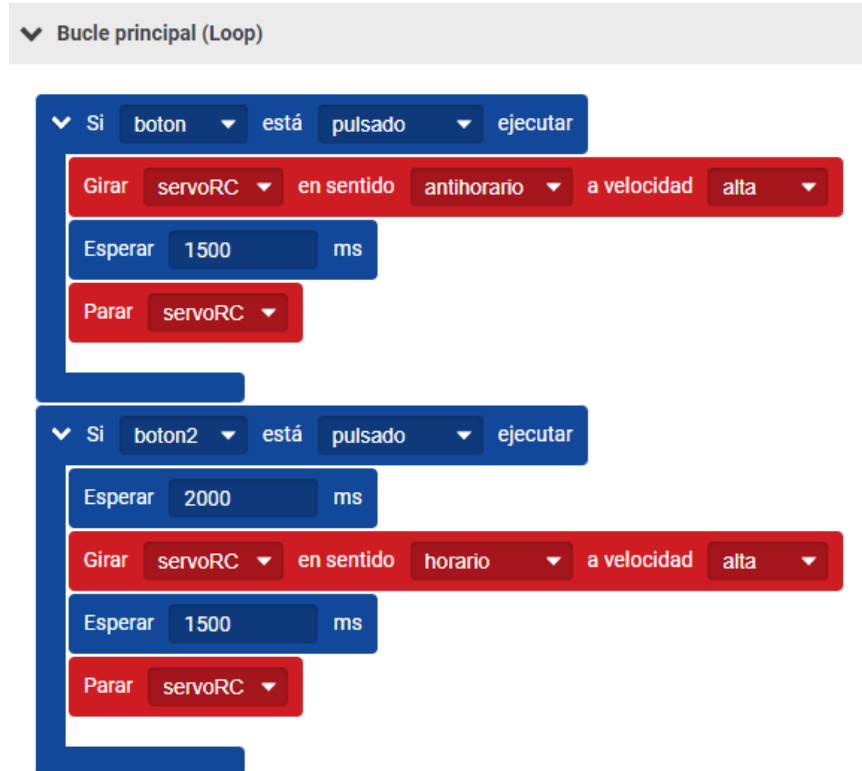


Unha vez que teñen a programación da apertura de porta, deberán integrala co botón, de maneira que a porta só se abra cando este sexa premido.

Neste exemplo, colocamos o botón de apertura tra-la forma dunha fechadura, de maneira que só unha chave con esa forma poida abrir a porta.



A programación que terán que crear terá a seguinte función: cando o botón é premido, o servo de rotación continua xira para abrir a porta e para cando termina. Unha vez que a porta está aberta, entramos ao garaxe e premeremos o botón de dentro para cerrar a porta.

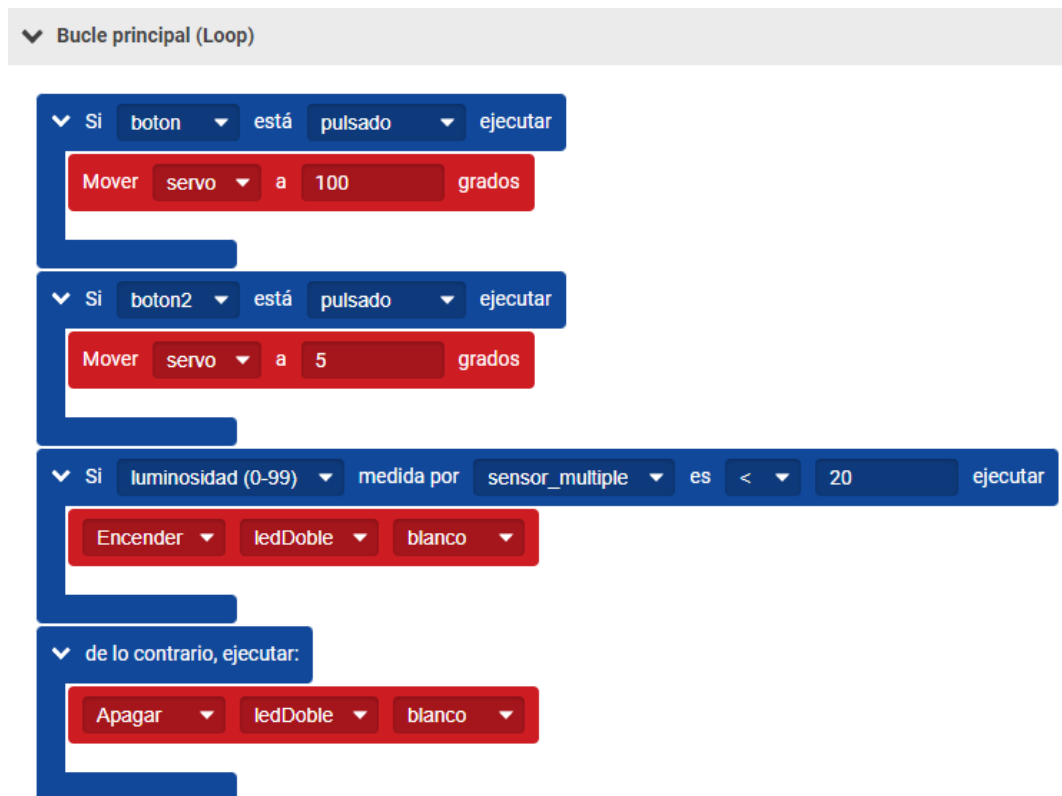


Unha vez que teñen a programación dos dous elementos (luz intelixente e porta automática),deberán xuntalas para que poidan funcionar á vez. Para iso, só terán que engadir a programación da luz debaixo da programación da porta.

A programación completa do exemplo usando un **servo RC** é a seguinte:



Se tivesemos usado o **Miniservo**, a programación sería a seguinte:



Se os alumnos terminaron de programar os dous elementos que se propoñen, pedirémoslles que melloren a súa casa programando outros elementos que se lles ocorran.

Algunhas melloras que se poden programar con máis elementos do kit de robótica son as seguintes:

- Alarma.
- Cortinas ou persianas automáticas.
- Ascensor.
- Comedeiro automático para mascotas.
- Timbre para a casa.
- Utilización doutro sensor múltiple para facer a apertura detectando o coche.

Temporalización suxerida:

1ª Sesión: Experimentación co kit: encender e chiscar un LED e ver valores de luz do aula co sensor de luz. Programar unha luz intelixente.

2ª Sesión: Construción dunha casa. Deseño de lámpada ou farola para incorporar luz intelixente.

3ª Sesión: Experimentación con sensor de distancia, Servo de rotación continua e Miniservo. Programación dunha porta automática.

4ª Sesión: Incorporación de melloras e corrección de erros.