

ROBÓTICA

AULA 18

Primeiros Passos Módulo 3



Imagem gerada com I.A.

Girabot



Diretoria de Tecnologia e Inovação

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Andrea da Silva Castagini Padilha

Validação de Conteúdo

Cleiton Rosa

Darice Alessandra Deckmann Zanardini

Revisão Textual

Kellen Pricila dos Santos Cochinski

Ilustrações em 3D

Roberto Carlos Rodrigues

Projeto gráfico, diagramação e geração de imagens IA

Edna do Rocio Becker

Apoio Técnico

Equipe UFMS

2024

SUMÁRIO

Introdução	2
Objetivos	3
Roteiro da aula	4
1. Contextualização	4
2. Roteiro de montagem	5
3. Feedback e finalização	23
Referências bibliográficas	24

Girabot

Introdução

Você já ouviu falar de biomimética? Essa palavra é a junção de bio=vida e mimese= palavra grega que se refere à imitação, lembra das aulas de ciências, dos animais que fazem mimetismo (cobra-coral-falsa que “imita” a parente que é venenosa)? A biomimética, também conhecida como bioinspiração, é um campo da ciência e tecnologia que se inspira na natureza para criar soluções inovadoras em diversos campos, como arquitetura, engenharia, robótica, medicina, agricultura, entre outros. Por exemplo, a criação de turbinas eólicas inspiradas na fisiologia das nadadeiras de baleias jubarte.

Nesta aula, você vai conhecer um pouco sobre o movimento dos girassóis, chamado de fototropismo, e relacionar com a Robótica. Pense aí, será que é possível colocar o fototropismo do girassol em painéis solares? Quais benefícios e obstáculos há nessa ideia?

Imagem gerada com I.A.

Fonte: Pexels, 2024.

Objetivos desta aula

- Compreender a movimentação do girassol conforme incidência de luz;
- Desenvolver um protótipo de um modelo de "robô girassol" que, a partir da detecção de luz pelo sensor, se movimenta com o auxílio de servomotor;
- Associar a programação do servomotor com as condições de luz captadas pelos sensores LDR presentes no protótipo.

Lista de materiais

- placa microcontroladora Arduino;
- Protoboard;
- 01 servomotor;
- 02 sensores de LDR;
- 03 jumpers macho-macho;
- 06 jumpers macho-fêmea.



Roteiro da aula

1. Contextualização

Com relação à Robótica, a biomimética pode ajudar na construção de soluções automatizadas com:

- Design funcional em robôs, como o exemplo das asas de libélulas que dão ao animalzinho o máximo de elevação com o menor arrasto, que faz delas ótimas no voo. Engenheiros usam o design das asas das libélulas para desenvolver robôs voadores mais eficientes e manobráveis.
- Sistemas autônomos de ventilação e climatização de prédios simulando os ninhos de cupins.

A biomimética é um campo em rápido crescimento com o potencial de revolucionar o campo da robótica. Ao se inspirarem na natureza, os estudantes de Robótica podem desenvolver robôs mais eficientes, adaptáveis e resilientes que podem ter um impacto positivo no mundo. Essa é uma oportunidade para refletirmos sobre o cuidado que todos devemos ter com a natureza e a sustentabilidade.

Nesta aula vamos utilizar um exemplo de biomimética, que tem a ver com o heliotropismo das plantas. Heliotropismo é o movimento da planta na direção do sol, como ocorre com os girassóis (experimente registrar a posição de um girassol pela manhã e depois à tarde).

Agora que você já aprendeu um pouquinho sobre biomimética e heliotropismo, vamos à montagem.

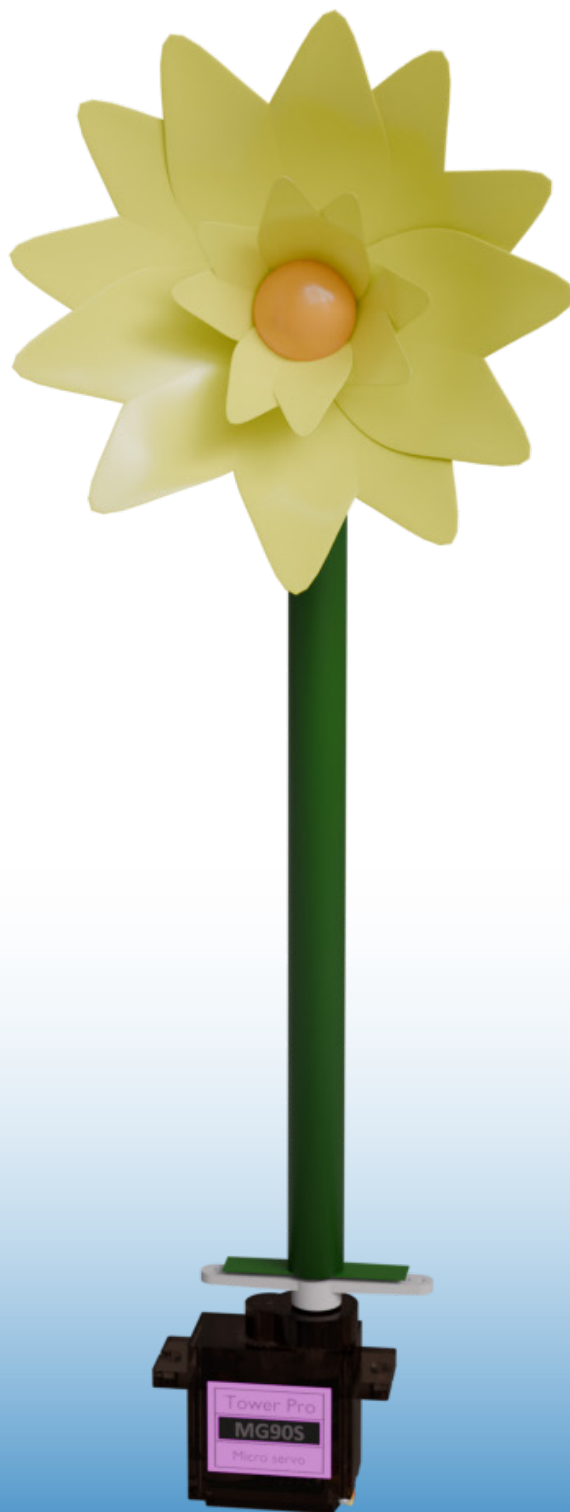


Imagem gerada com I.A.

2. Roteiro de montagem

A montagem deste protótipo é bastante simples. Você vai precisar dos dois sensores LDR, seis jumpers macho-fêmea, para ligá-los na Protoboard e Arduino, e três jumpers para ligar o servomotor no protótipo.

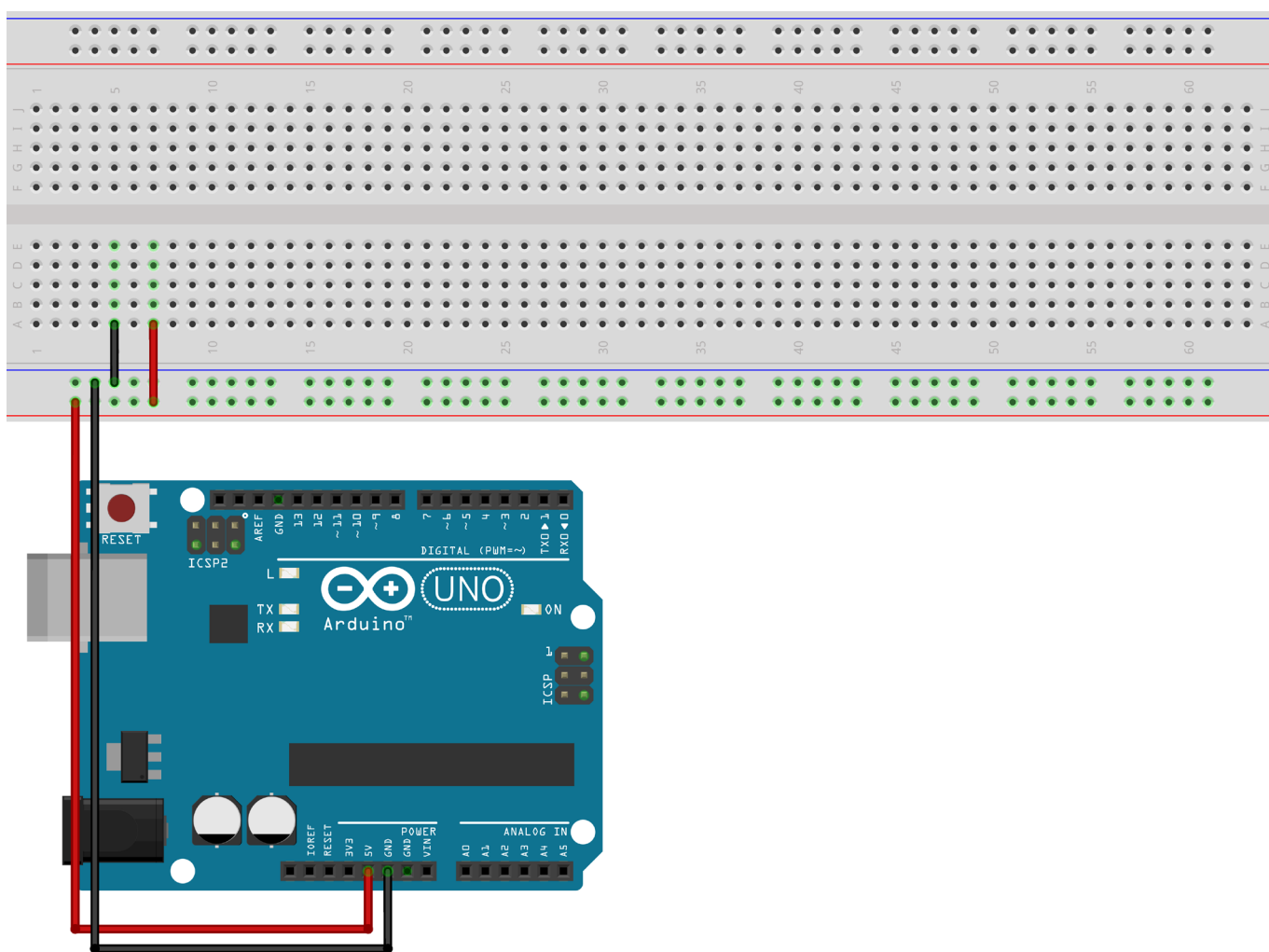
Uma dica: se tiver canudos de papel (que são mais firmes) ou palitos de churasco, improvise um girassol para ser acoplado ao servomotor. O palito ou canudo é a haste da flor e as pétalas e sépalas podem ser feitas com papel colorido ou folha sulfite branca. Desenhe o miolo da flor e cole essa composição na haste. Depois, cole com fita crepe ou cola quente a flor sobre a pá do servomotor (ao final terá que deixar o componente como foi encontrado originalmente).



Fonte: Seed/DTI/CTE, 2024

Energize a protoboard ligando um jumper na lateral negativa à porta GND do Arduino e outro jumper da lateral positiva da Protoboard para a porta 5V. Insira um jumper para energizar uma coluna como negativa e um segundo jumper para energizar outra coluna positivamente, como mostra a figura a seguir.

Figura 01 – Montagem inicial e energização da Protoboard.

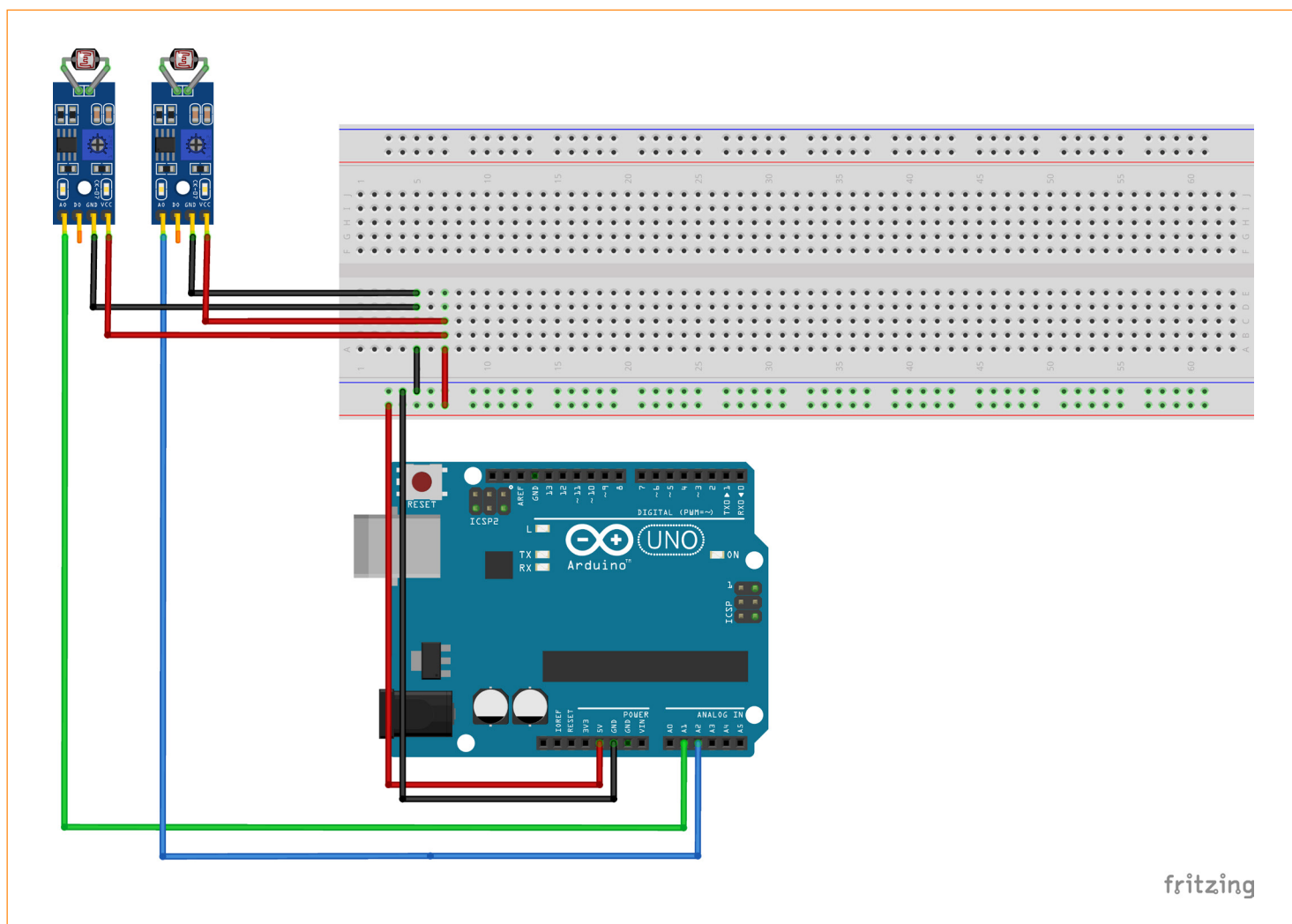


fritzing

Fonte: Fritzing, 2024.

Após essa etapa, faremos a conexão dos sensores de LDR (utilizando os jumpers macho-fêmea) nas portas analógicas determinadas para facilitar a movimentação do girassol robô. Perceba que na sugestão de montagem na Figura 02, o jumper macho-fêmea da porta positiva de cada um dos sensores é conectada na Protoboard, na coluna positiva. E cada um dos jumpers ligados aos terminais GND dos sensores têm sua outra ponta conectada na coluna energizada pelo jumper negativo, inserida anteriormente. O terminal analógico do sensor 1 (sensor direito) deve ser conectado na porta analógica A1. E a porta analógica A2 deve ter o jumper do sensor 2 (sensor esquerdo). Observe a montagem na Figura 2.

Figura 02 - Ligação dos sensores LDR.

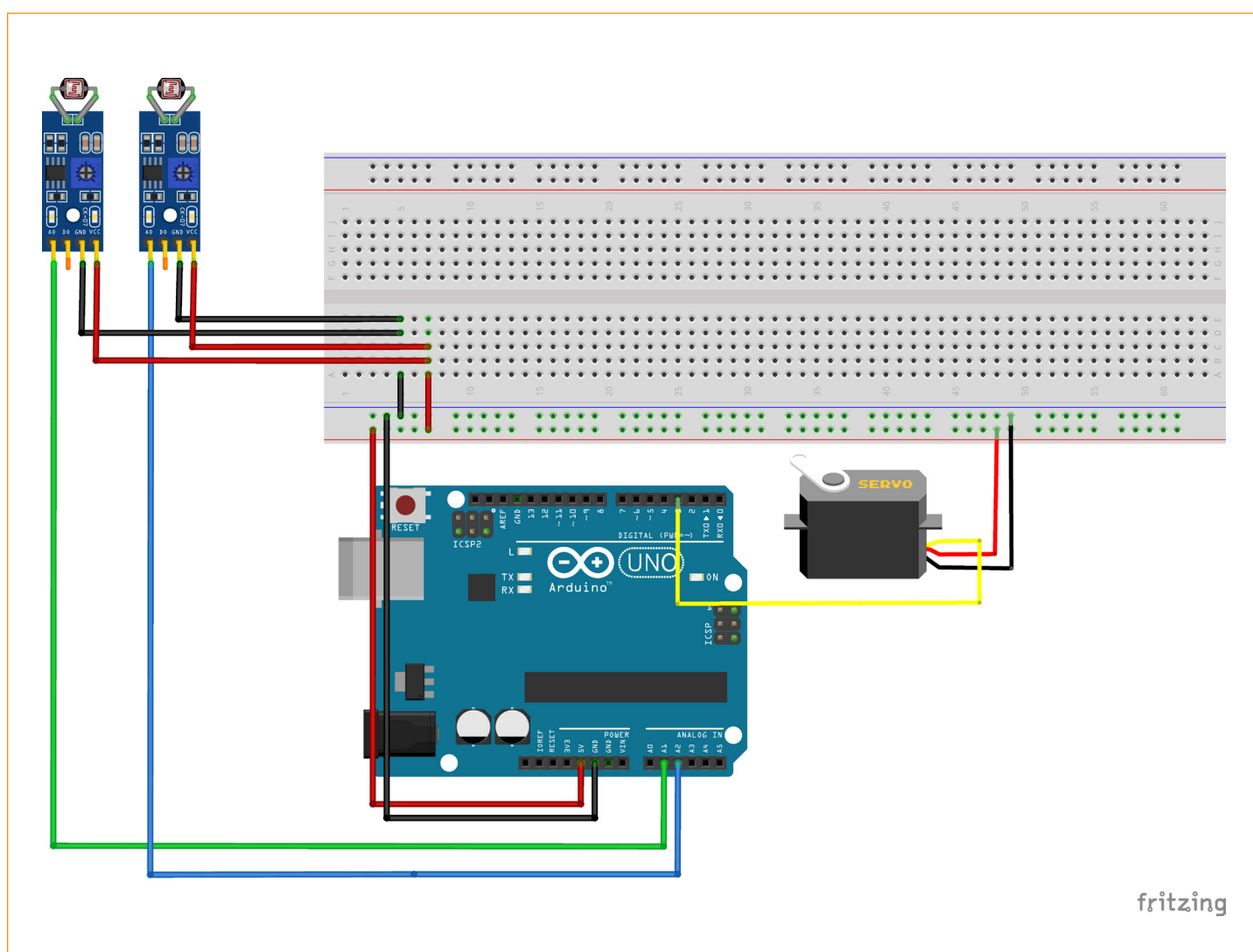


Fonte: Fritzing, 2024.

Girabot

Por fim, conectamos o servomotor na base do caule do protótipo de girassol para permitir os movimentos do Girabot (que você e seus colegas construíram com o canudo e papel). Ele deve ser conectado sobre a aleta plástica do servomotor, com cuidado para não danificar o componente. Realize as ligações dos jumpers para cada terminal do servomotor e utilize a uma saída PWM digital (possui um ~) para controlar a programação do servo motor. Observe a Figura 3 para ver a montagem completa do protótipo.

Figura 3 – Montagem completa do protótipo.



Fonte: Fritzing, 2024.

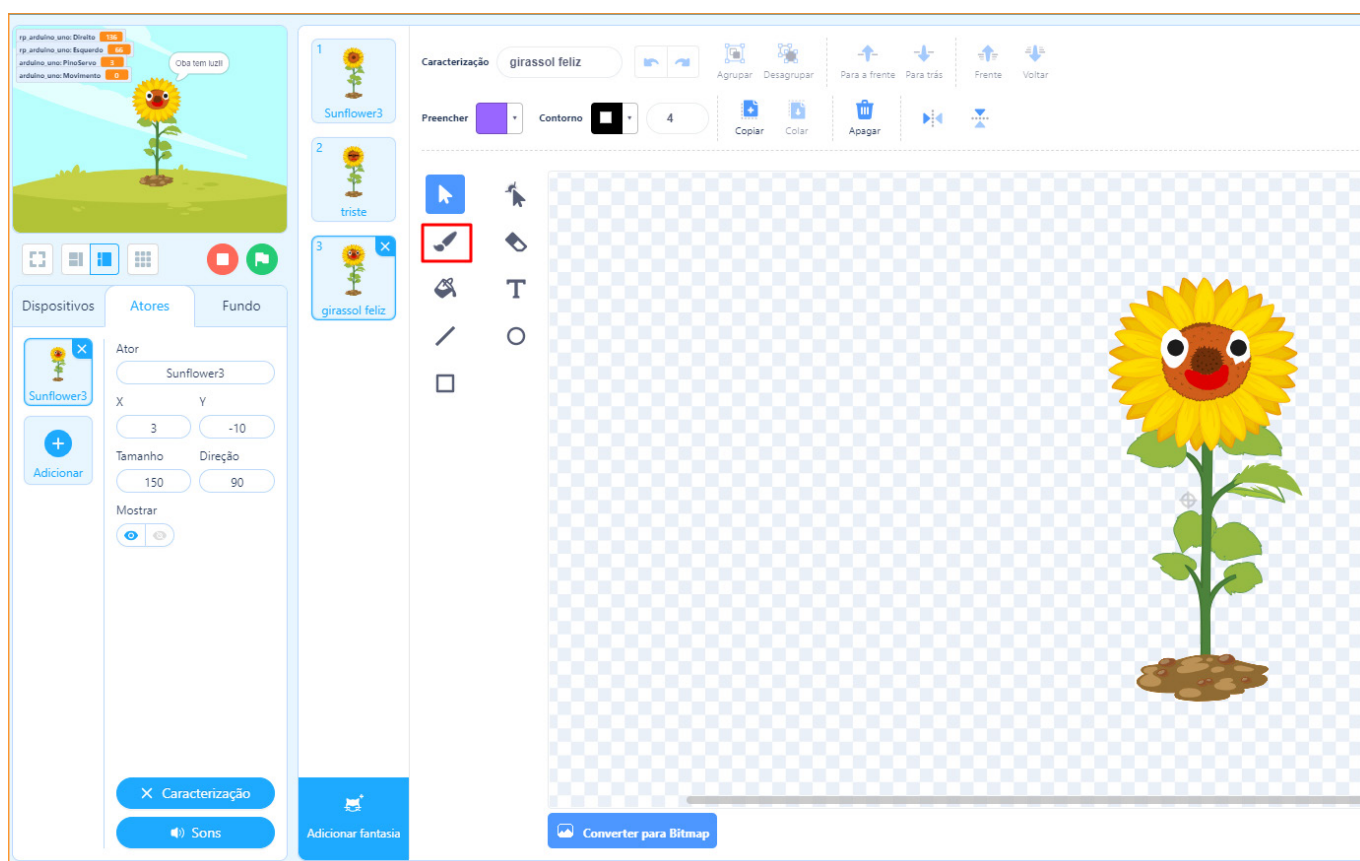
Agora vamos para a programação!

Programação

Antes de começar a programar, é importante esclarecer que essa é uma proposta que combina a ação dos componentes no mundo físico, com o espelhamento dentro do mBlock, no digital. Você e seus colegas fizeram já um experimento desses na aula **16 - Abelhinha**. Então, para que tudo funcione e faça sentido, o primeiro passo é clicar em **Atores** e começar nossa construção no digital.

Apague o panda e adicione um novo ator. O girassol que escolhemos como exemplo está com o nome Sunflower. Você vai criar fantasias para ele clicando em “Caracterização”. Essa ação é necessária para que quando o girassol estiver recebendo luz fique “Muito feliz” e quando não recebe, fique “Triste”. Duplica o ator Sunflower5 e selecione no painel o pincel. Com ele você pode fazer olhos e uma boca sorrindo como mostra a Figura 04.

Figura 04 – Painel de Caracterização dos atores.



Fonte: mBlock 5, 2024.

Duplique novamente o ator para criar a versão “Triste”. Agora, como é possível ver na Figura 04, você deve ter 3 versões do girassol. Feche a janela de edição, clicando sobre caracterização novamente. Você pode inserir um fundo para ambientar sua animação com programação. Uma sugestão é o fundo grassland4 (Figura 05).

Figura 05 – Fundo grassland4.



Fonte: mBlock 5, 2024.

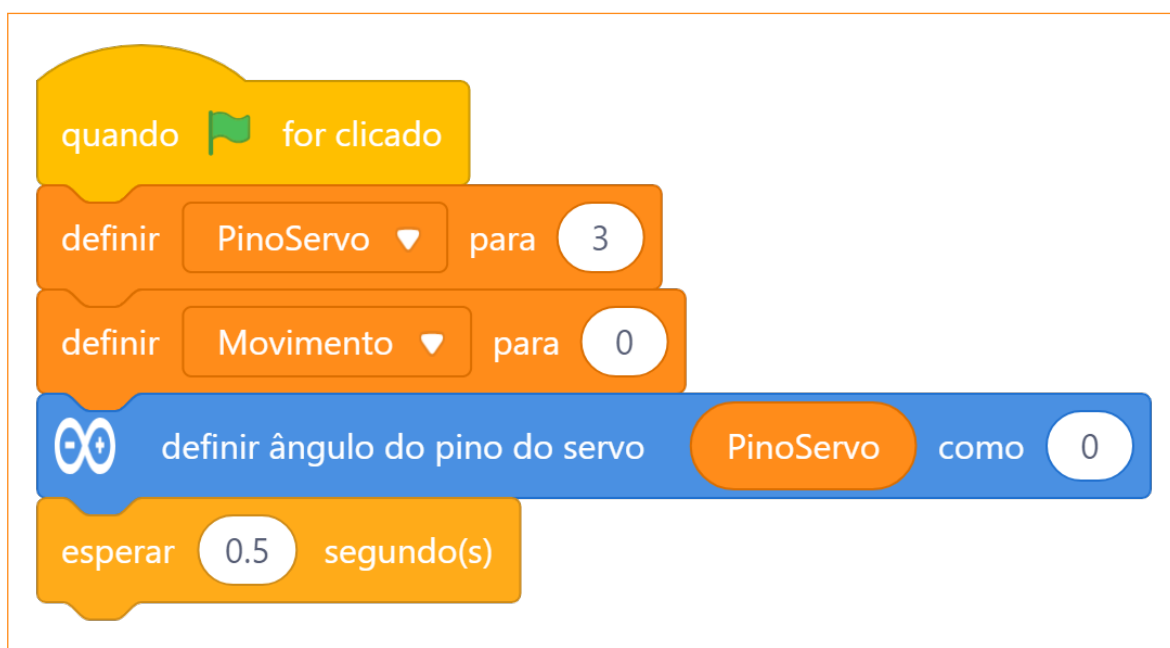
Agora de volta à programação. Este protótipo solicita a criação de quatro variáveis, uma vez que vocês estão trabalhando com dois sensores e um servomotor. A ideia é programar o servomotor para que, quando um dos sensores LDR perceber mais luz que o ângulo dele, se mova para a direção percebida. Isso vai ocorrer no protótipo, no girassol construído e conectado na aleta do servomotor.

Criaremos neste projeto quatro variáveis que serão utilizadas para guardar valores importantes para o funcionamento do código:

- Movimento: é uma variável utilizada para definir os valores da movimentação do girassol através de somas e subtrações que representam o valor em grau do posicionamento da pá do servomotor.
- Esquerdo: registra o valor recebido pela leitura analógica do sensor de LDR do lado esquerdo.
- Direito: registra o valor recebido pela leitura analógica do sensor de LDR do lado direito.

Inicialmente, antes do loop, vamos definir o valor das variáveis **<PinoServo>** e **Movimento**. Além disso, indicaremos para o servomotor posicionar-se no ângulo 0°. Após isso, um bloco de **<espera de 0,5 segundos>** apenas para não entrar no loop. Importante lembrar também, que essa programação será tanto digital na tela do mBlock, quanto no mundo físico dos componentes. Para isso, o bloco de inicialização deve ser o **<quando a bandeira verde for clicada>**. Essa montagem inicial pode ser vista na Figura 6.

Figura 6 – Definição das variáveis PinoServo e Movimento.



Fonte: mBlock 5, 2024.

Abaixo desses blocos será inserido o loop eterno dentro do qual irão os comandos para os sensores e servomotores. Esse loop é definido pelo bloco **<repetir para sempre>**.

Vamos agora à sequência que será incorporada dentro do bloco repetir para sempre. Primeiro, é necessário definir para o Arduino qual dos sensores LDR se referem ao lado direito e esquerdo. Então, dentro da categoria variáveis, selecione o bloco **<definir ... para>** e selecione na seta para baixo, a palavra Direito. Esse sensor será ligado na porta analógica A1. Para isso ocorrer, vá à categoria Pin e selecione o bloco **<ler pino analógico (A)>** e insira o numeral 1. Em seguida, conecte o bloco **esperar 0,2 segundos**. Você pode duplicar esses dois blocos (com o botão direito do mouse), e alterar de

Direito para Esquerdo e o pino analógico de 1 para 2. Agora, seu código já definiu as portas de todos os componentes. Nessa programação você vai usar vários blocos <Se então... Senão>. São chamados de loops que vão indicar o que o Arduino deve fazer com as condições e leituras que os sensores farão.

No primeiro loop, o bloco <Se então... Senão> deve ser configurado da maneira como mostra a Figura 5. Dentro do espaço após o “se” serão colocados blocos de operação. Aqui a informação ao Arduino é que se os sensores LDR perceberem uma quantidade de luz alta (indicada por números menores que 200) eles farão algo. Para colocar na linguagem do mBlock, você precisará localizar na cate-

goria operadores, e arrastar um bloco **menor que**. O primeiro espaço será ocupado pela variável que se refere ao sensor da direita, coloque ali a variável Direito e depois do sinal complete com o numeral 200. Duplicue o bloco, e altere a variável Direito para Esquerdo. Agora retorne à categoria Operadores, porque o Arduino precisa entender que se um sensor LDR ou outro estiver nessas condições será feita uma ação. Para isso, selecione o bloco “ou” e o encaixe depois do espaço “se” no bloco <Se então... Senão>. No primeiro espaço desse operador “ou” coloque o bloco com a variável Direito e no segundo espaço, o bloco com a variável Esquerdo, como mostra a Figura 7.

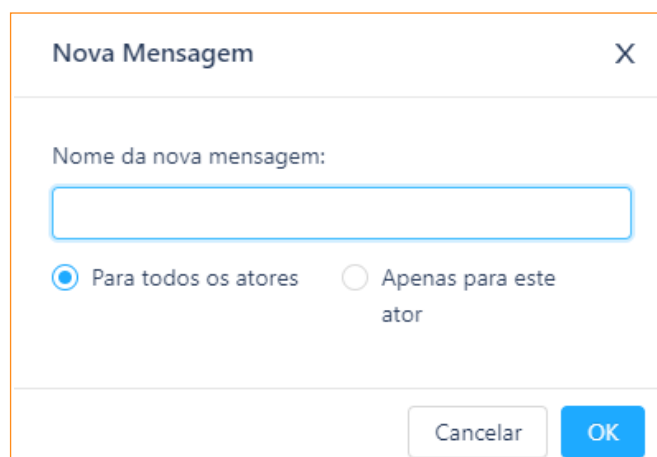
Figura 7 - Configuração do primeiro loop.



Fonte: mBlock 5, 2024.

A ação que o Arduino deve executar se a condição acima for atendida é: **Definir a situação “Muito Feliz”** e **<esperar 1 segundo>**. Mas o que significa essa situação Muito Feliz? Ela tem a ver com a fantasia criada anteriormente para o girassol. Agora, será necessário programar o que ocorre quando a situação muito feliz é chamada no código. Para isso, clique em **Atores**. Selecione o girassol. Chame o bloco **<quando Eu receber...>**. Clique na flecha para baixo. Abrirá uma janela para você editar a mensagem, como mostra a Figura 8.

Figura 8 – Janela nova mensagem.

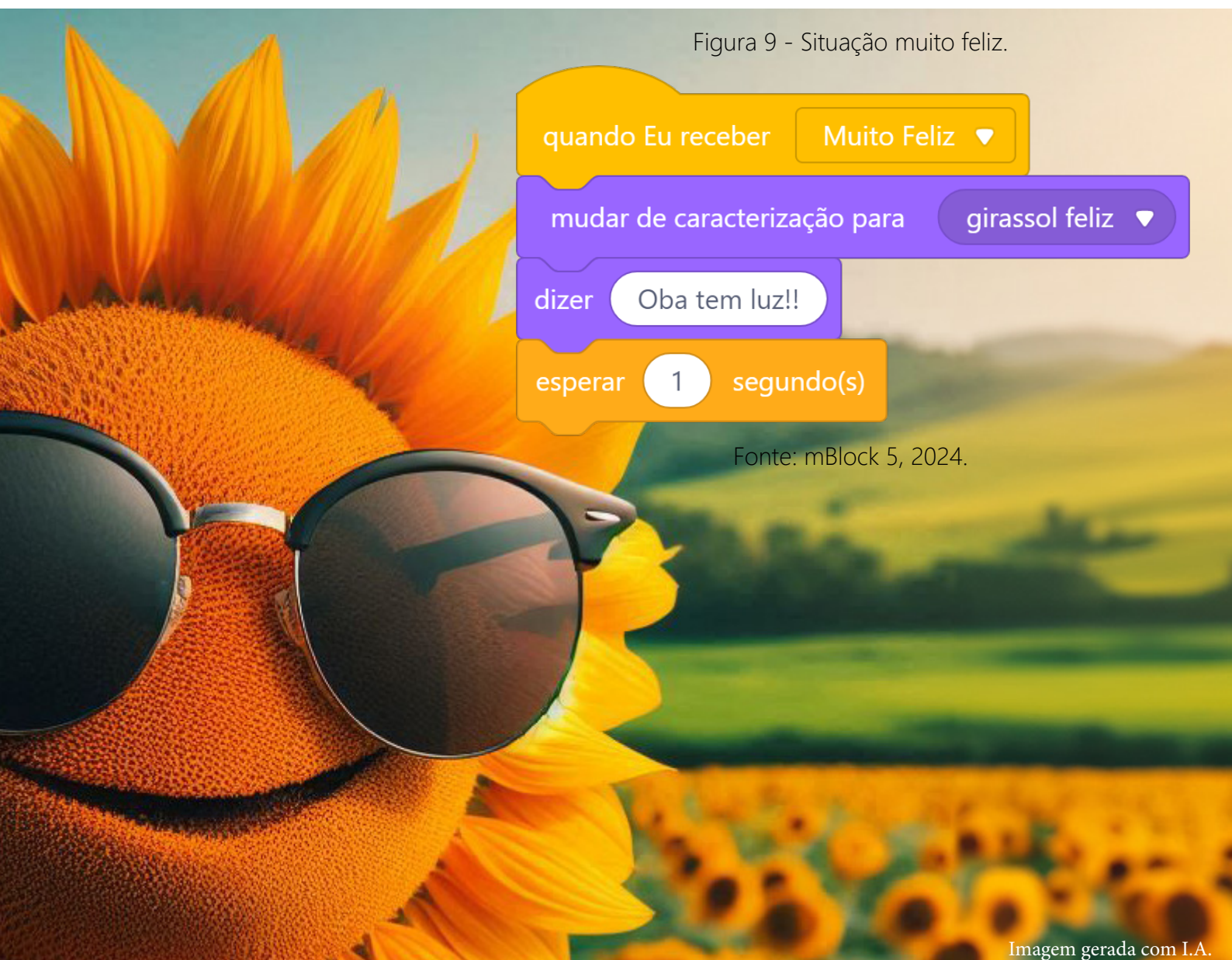


Fonte: mBlock 5, 2024.

Imagem gerada com I.A.

Selecione “Nova mensagem”, escreva “Muito Feliz” e clique em Ok. Repita esses passos para criar as situações “Triste” e “Normal”. Agora, volte ao bloco <quando Eu receber...> e selecione “Muito Feliz”. Quando for chamada a situação Muito Feliz, o Arduino executará a mudança de caracterização para o girassol feliz que vocês alteraram, ao inserir o bloco <mudar de caracterização para “girassol feliz”> colocando o sorriso. Depois, ainda na categoria aparência, você pode selecionar o bloco <dizer...> e colocar uma mensagem que reflita o estado de espírito do girassol por ter luz. E, para finalizar, insira o bloco <esperar 1 segundo>, como mostra a Figura 9.

Figura 9 - Situação muito feliz.



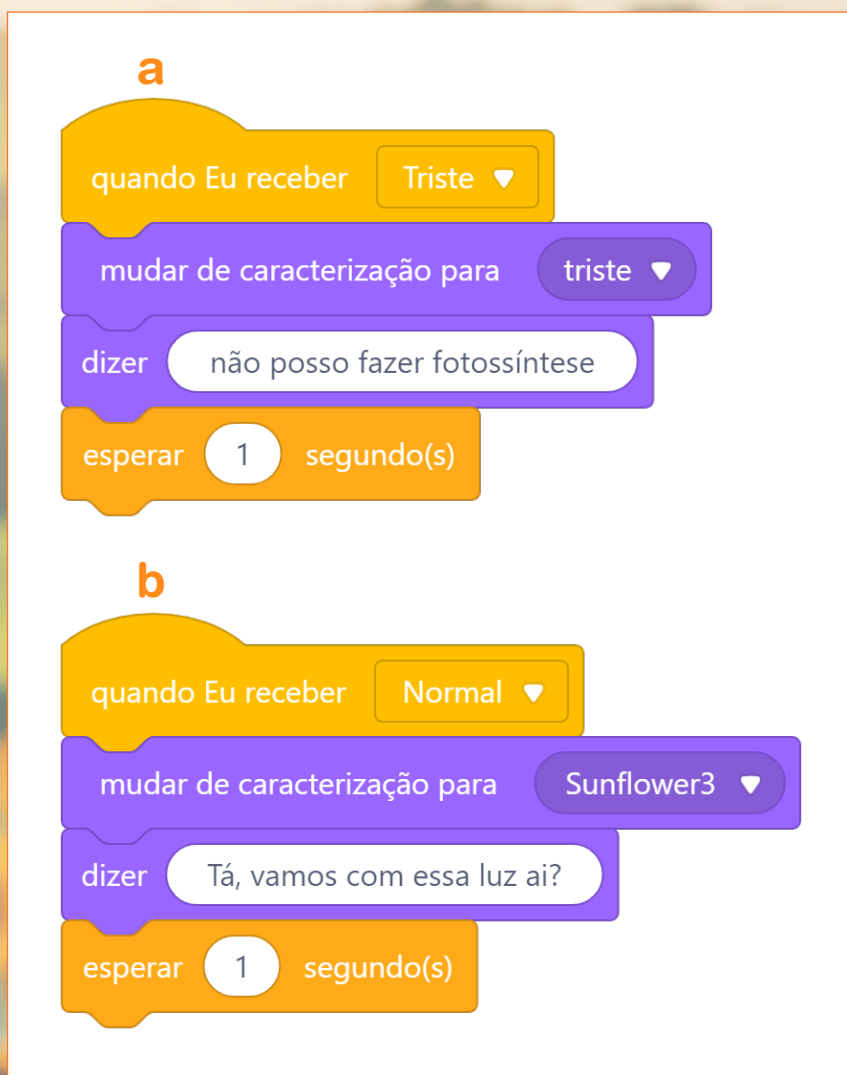
Fonte: mBlock 5, 2024.

Imagem gerada com I.A.

Girabot

Esse bloco pode ser duplicado, para programar as outras duas situações. A primeira delas, é alterar o “Muito feliz” para “Triste”. No bloco mudar de caracterização, selecione o girassol triste que vocês também editaram no início da programação. E no bloco dizer, coloque outra frase que faça sentido. Já na última situação, será o Normal. Então a caracterização do girassol deve ser a original. As figuras 10a e 10b mostram a programação dessas duas situações, que serão chamadas no código posteriormente.

Figura 10 a e b - Situações triste e normal.

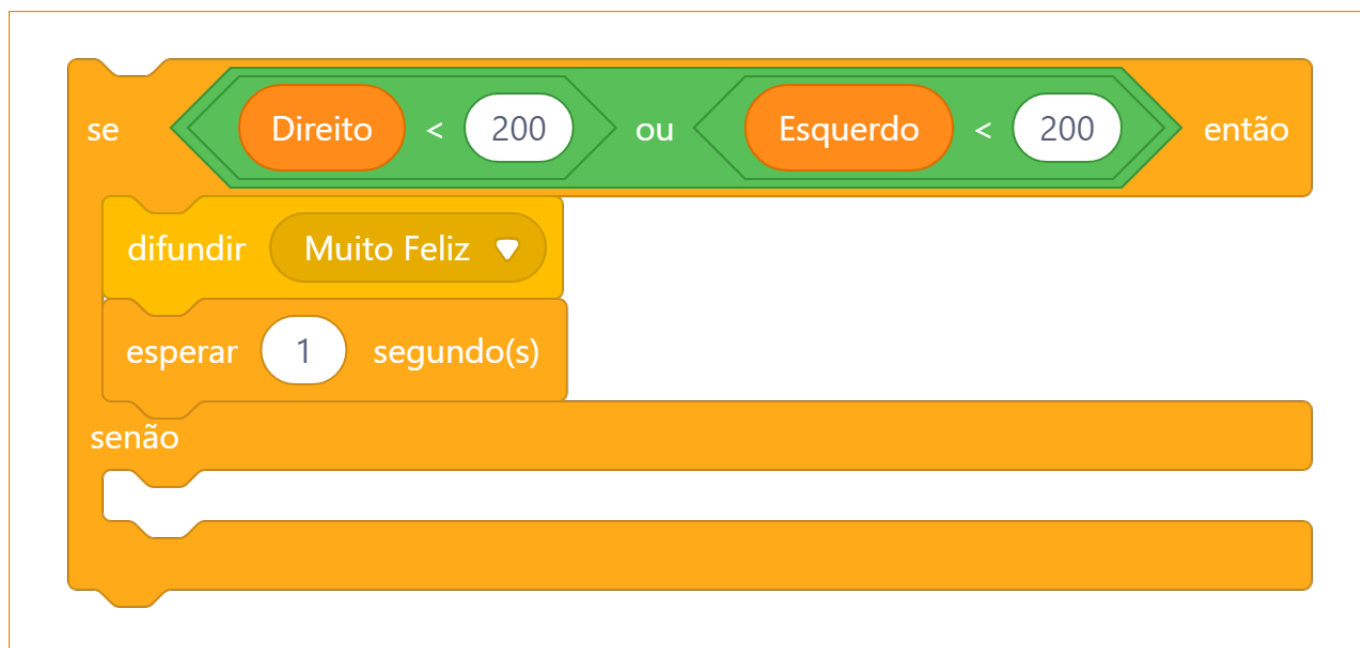


Fonte: mBlock 5, 2024.

Imagem gerada com I.A.

Agora, clique sobre **Dispositivos** para retornar à programação do Arduino. Na categoria Evento, selecione o bloco definir e selecione Muito Feliz. Depois conecte o bloco <esperar 1 segundo>. A montagem dessa parte deve ficar como mostra a Figura 11.

Figura 11 - Inserção da situação muito feliz quando a planta recebe bastante luz.

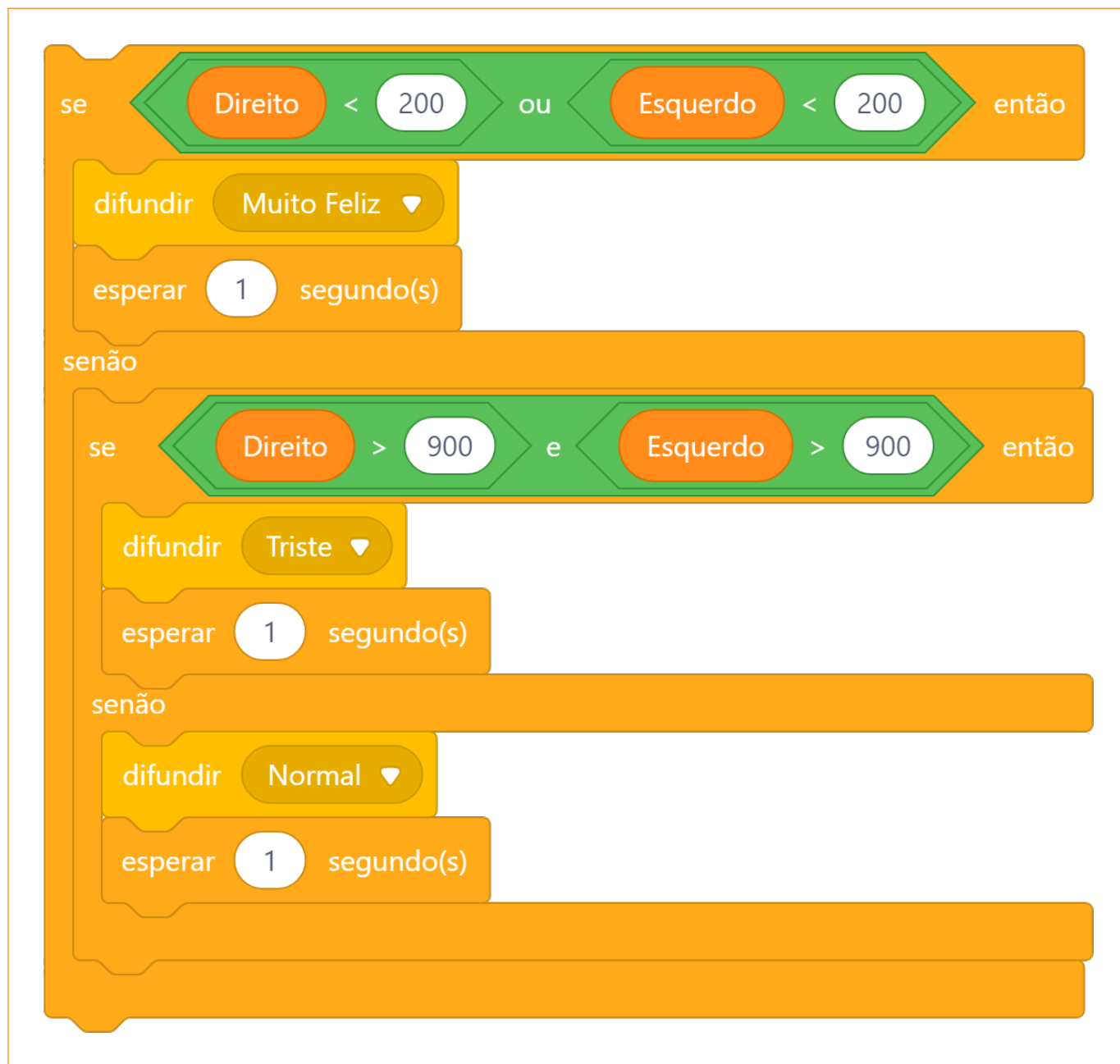


Fonte: mBlock 5, 2024.

Agora, após o “senão” será inserido o que o Arduino vai fazer no caso da condição de “muita luz (menor que 200)” não ocorrer. Se o ambiente estiver mais escuro, gerará um número maior que 200, podendo chegar até 1023. No código, você pode utilizar qualquer número maior que 200, entendendo que será necessário, em todo caso, calibrar o Trimpot (parafuso no quadradinho azul) dos dois sensores para aferir a sensibilidade. Você verá que a sugestão indicada para o ambiente sem luz é maior que 900.

Um detalhe importante! Como há mais situações a serem programadas, no espaço após o se... Então, que seria o “senão”, você deverá colocar outro bloco <Se então... Senão>. E dentro do espaço após o “se”, a condicional já explicada acima, dos sensores captarem pouca luz e retornar um número maior que 900. Caso essa condição seja atendida, o que acontece com o protótipo e com a nossa animação no mBlock? Será chamada a situação Triste, a partir do bloco <difundir “Triste”> e depois o bloco <esperar 1 segundo(s)>. Após o “senão”, insira o bloco <difundir “Normal”> e o bloco <esperar 1 segundo(s)>. Veja a programação desses blocos na Figura 12.

Figura 12 - Inserção da situação triste.

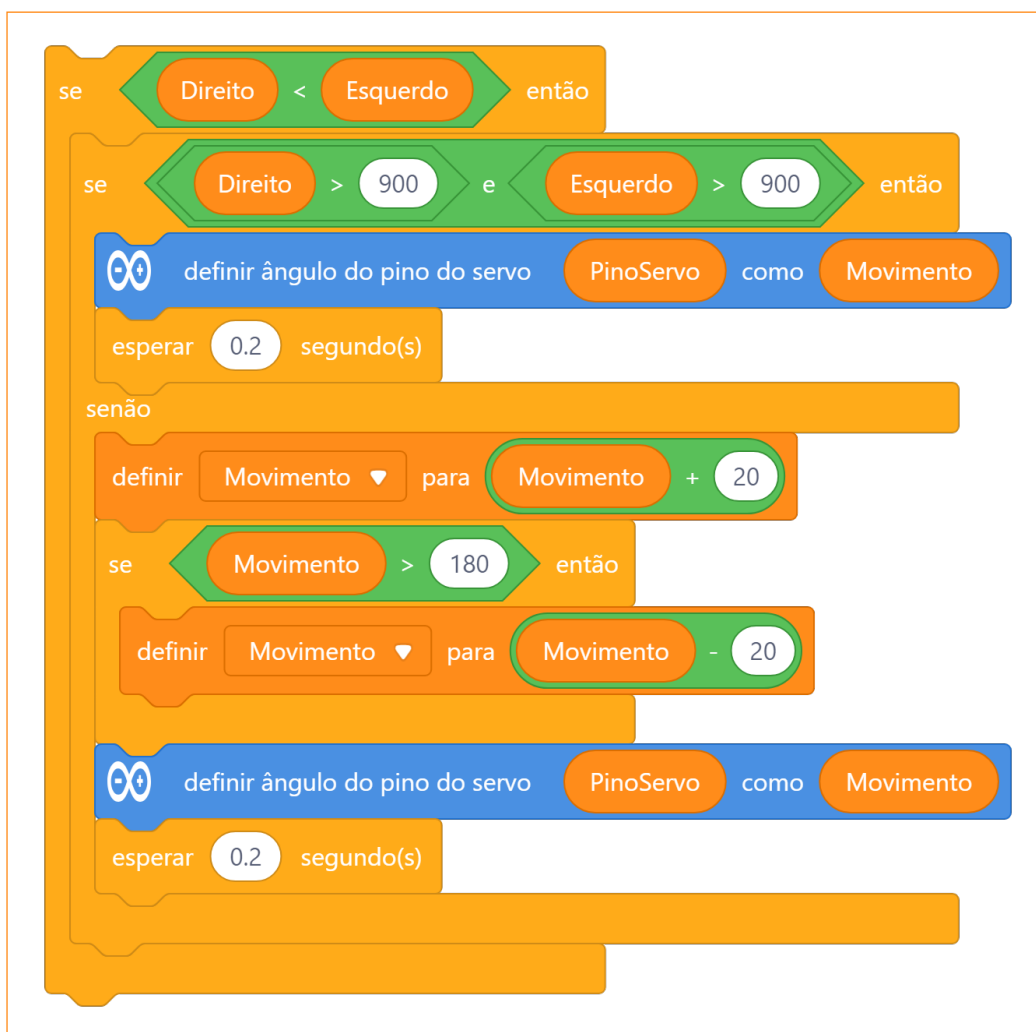


Fonte: mBlock 5, 2024.

Até o momento, programamos o que o Arduino fará no mBlock, quais alterações fará na animação inserida com o ator. Mas queremos que o protótipo físico do girassol gire em direção ao sensor que recebe mais luz. Para isso, você irá criar outros loops com os blocos <Se então... Senão> e <Se... então>. Para começar vamos definir que, se a incidência de luz for maior em um dos sensores LDR, que o servomotor deverá alterar seu ângulo “movendo” o girassol de papel em uma direção (esquerda ou direita).

Para isso, precisaremos novamente de um bloco <Se então... Senão>. E no espaço da condicional após o “se”, procurar o bloco **operador menor**. Inserir no primeiro espaço a variável **Direito** e depois do sinal de **menor** a variável **Esquerdo**, como mostra a Figura 13.

Figura 13 - Movimentação do protótipo para a direita.



Fonte: mBlock 5, 2024.

Então, se o sensor **Direito** está recebendo mais luz que o **Esquerdo**, o Arduino tem que verificar duas situações. A primeira, que essa quantidade de luz é pouca, então o protótipo fica parado, e se não cumprir essa condição, o movimento do servomotor é acrescido de 20 no sentido da direita (positivo). Para escrever esse primeiro loop, será necessário outro bloco <**Se então... Senão**>.

Arraste o bloco operador **maior que**, e insira no primeiro campo a variável **Direito** e no segundo campo o número 900. Duplique esse bloco e altere a variável para **Esquerdo** no primeiro campo. Agora arraste um novo bloco **operador "e"**. Coloque o bloco operador **maior que** da variável **Direito** no primeiro campo e, o bloco operador **maior que** do **Esquerdo**, no segundo campo. Após o "então", insira o bloco <**definir o ângulo do servomotor**> e ao invés do número da porta relacionada, selecione e conecte a variável "**PinoServo**". Após o "como" coloque a variável "**Movimento**." Isso quer dizer que o servomotor conectado na porta 3 se movimenta como programado na variável movimento (que aqui não recebe ação, portanto, não se move). Após conecte um tempo de **espera de 0,2 segundos** com o bloco correspondente.

Se essa condição (os dois sensores no escuro) não for contemplada, então, o Arduino deverá fazer o colocado após o "senão". Para fazer o protótipo se mover para a direita, deverá indicar o movimento com uma soma no ângulo. Para isso, pegue o bloco <**definir "Movimento" para**> na categoria variáveis. No espaço após o "para", conecte um **bloco operador de soma**. No primeiro espaço, coloque a variável **Movimento** e no segundo espaço, após o sinal de mais, o número 20 (veja a Figura 13, acima).

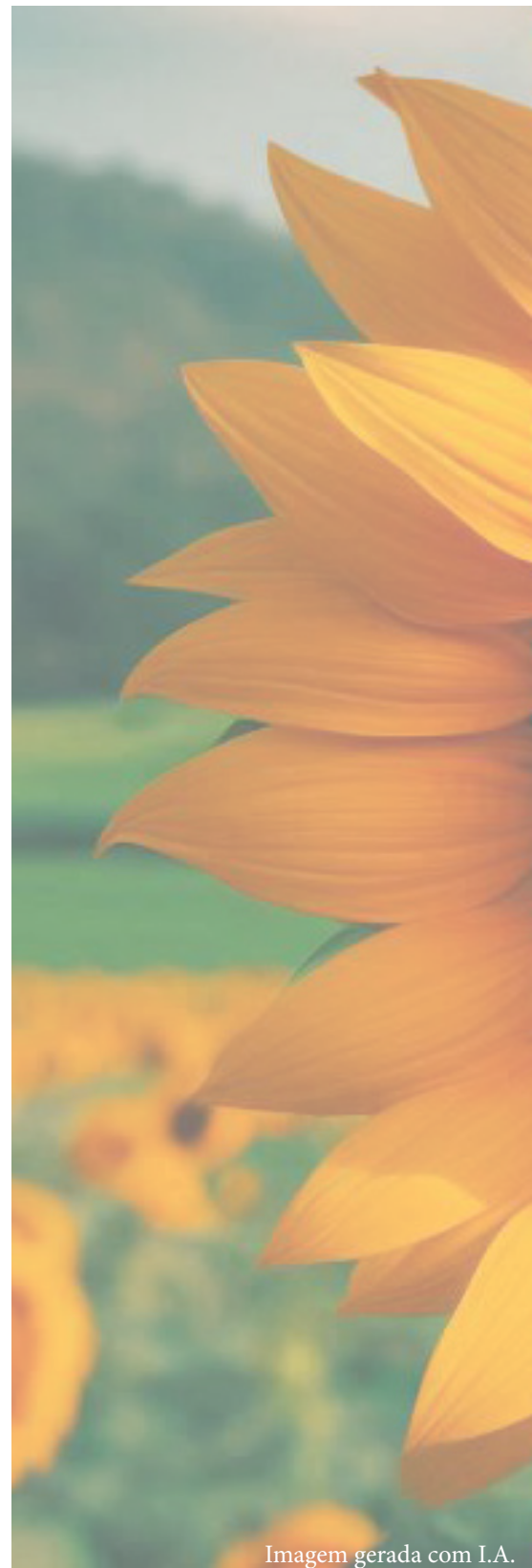
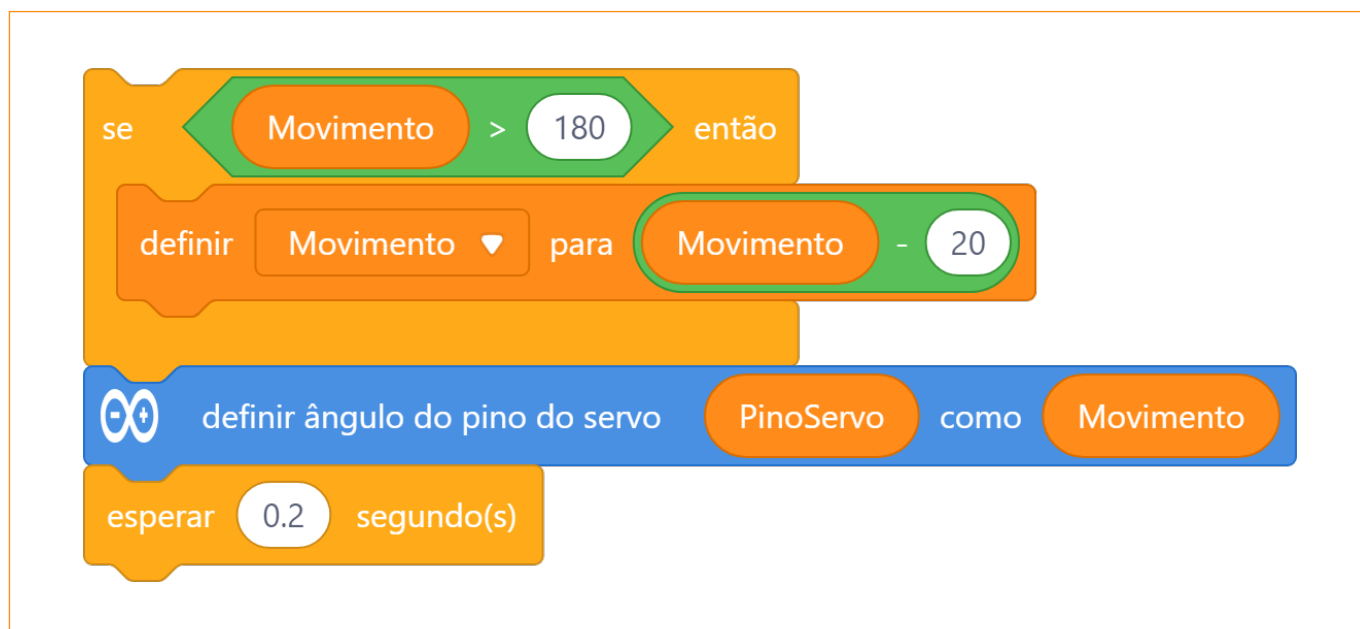


Imagem gerada com I.A.

A Figura 14 mostra o detalhe do que já foi conectado na Figura 13 (não precisa arrastar esses blocos). Pegue um **operador maior que**, insira a variável **"Movimento"** e depois o número 180. Ou seja, quando atingir 180°, faça a ação determinada depois do "então". Aqui a ação é colocar o bloco <**definir "Movimento" para...**> e trazer o **operador de subtração**. Colocar no primeiro campo a variável **"Movimento"** e no segundo campo o número 20. Em outras palavras, quando chegar a 180° recue em 20° para a esquerda e pare (o bloco <**definir ângulo do pino do servo "PinoServo" como "Movimento"**> define isso). Depois desse bloco, insira o bloco <**esperar 0,2 segundos**>.

Figura 14 - Análise da programação feita na Figura 13.



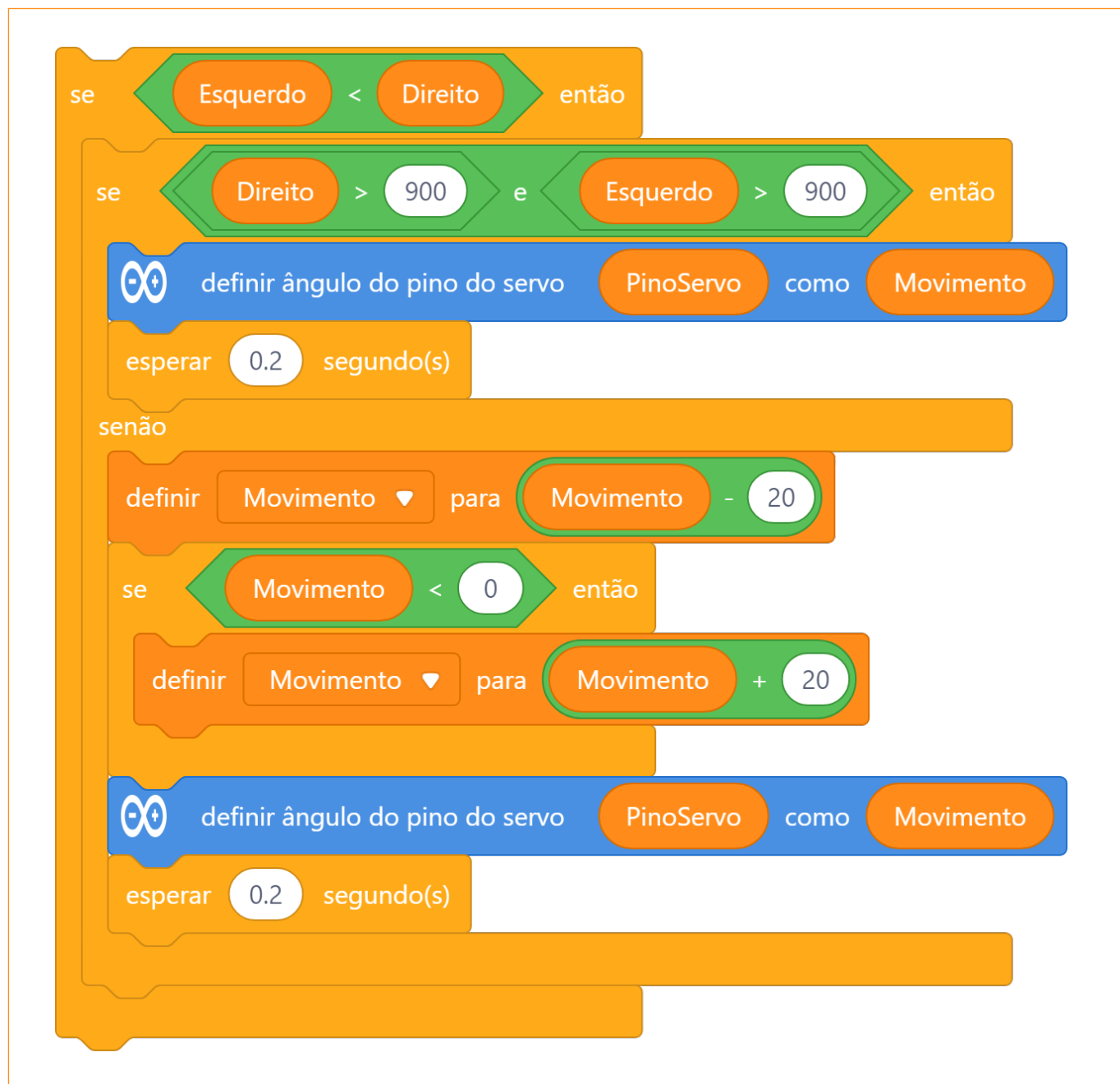
Fonte: mBlock 5, 2024.



Imagem gerada com I.A.

O que foi feito para o lado direito deve ser feito também para o lado esquerdo. A lógica é a mesma, com pequenos detalhes que serão alterados. A programação dessa parte será anexada abaixo dos blocos indicados na Figura 13. Observe a Figura 15 para ver quais blocos devem ser adicionados.

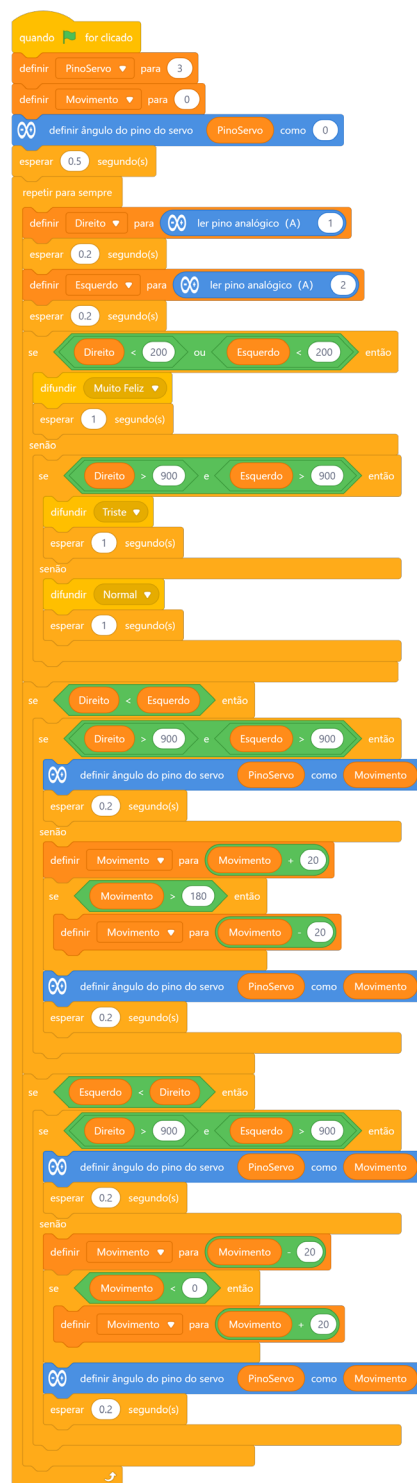
Figura 15 - Programação da movimentação do protótipo para a esquerda.



Fonte: mBlock 5, 2024.

Perceba que a diferença nessa sequência é que a variável **Esquerdo é menor que** variável **Direito**. Mantém então que se os sensores estiverem recebendo pouca luz (com isso números acima de 900) não há movimentação do servo. E a outra mudança está se, essa condição não for atendida (que há mais luminosidade no sensor Esquerdo), será definido **“Movimento”** dentro do operador de subtração (o servo vai movimentar em ângulos para a esquerda). Então, é preciso pensar que essa movimentação pode levar até o ângulo 0°. E quando isso ocorrer, é necessário corrigir o ângulo para continuar a programação anterior. Assim, é colocado outro bloco **<Se então... Senão>** e dentro dele o bloco operador menor. Insira no primeiro campo a variável “Movimento” e no segundo o número 0. Depois de então, o bloco **<definir “Movimento” para...>** e a expressão no bloco **operador de soma**, **“Movimento” + 20**. Ainda na sequência do bloco **<Se então>** inserir o bloco **<definir ângulo do pino do servo “PinoServo” como “Movimento”>** e depois o bloco de **<esperar 0,2 segundos>**. Na Figura 16, você pode ver a programação completa.

Figura 16 - Programação completa.



Fonte: mBlock 5, 2024.

Desafio

Que tal os desafios a seguir?

a) Incluir no protótipo um display OLED que apresente a mesma mensagem que foi programada para o ator quando ele está recebendo luz ou quando não está fazendo fotossíntese.

b) Prototipar, na fase de ideias e projeto no papel, como essa programação e componentes poderiam otimizar a captação de luz solar em placas fotovoltaicas.

c) Pensar em como a robótica com a biomimética e heliotropismo podem impactar de forma positiva a agricultura de precisão e a exploração espacial.

3. Feedback e finalização

a. Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado.

b. Analise nessa observação dos projetos o que houve de diferenças entre os protótipos (material do girassol, frases, e mesmo da programação).

c. Quais as dificuldades encontradas na elaboração desse projeto? Como superaram os obstáculos?

d. Como os componentes conjugados nessa proposta poderiam ser utilizados em outras frentes, como na agricultura ou na criação de robôs?

e. Dentro da proposta dos múltiplos loops, de que forma você e seus colegas poderiam otimizar o código?

f. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de Robótica.

Referências Bibliográficas

REIS, Ney R. S.; SOUZA, Lucia Helena R. **Robótica e biomimética: a tecnologia aliada à natureza**. Revista Scientiarum Historia, 2019, v2:e102. Disponível em: <https://revistas.hcte.ufrj.br/index.php/RevistaSH/article/view/102/81>. Acesso dia 16 mai. 2024.

ROCHA, Bruno M; SILVA, Kiany F.D. **Potencial dos sistemas responsivos para soluções de caráter ecológico na Arquitetura, Urbanismo e Design - Seguidor Solar Girassol**. Relatório final de Pesquisa. UFES. Disponível em: <https://www.territorios.org/rede/pdf/kianyrelatorio.pdf>. Acesso dia 16 mai. 2024.

Sociedade Internacional de Biomimética. Disponível em: <https://www.biomimicry.org/v>. Acesso dia 16 mai. 2024.

Tecmundo. **5 tecnologias robóticas inspiradas totalmente na natureza**. Disponível em: <https://www.tecmundo.com.br/biotecnologia/43231-5-tecnologias-roboticas-inspiradas-totalmente-na-natureza.htm>. Acesso dia 16 mai. 2024.

YOUTUBE. **Girassol Seguidor de luz com Arduino**. Disponível em: <https://www.youtube.com/watch?v=wufE2xWhG-Q>. Acesso dia 16 mai. 2024.



Imagem gerada com I.A.

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL (UFMS)
FACULDADE DE COMPUTAÇÃO (FACOM)

PROFESSORES

- Amaury Antônio de Castro Junior
- Anderson Corrêa de Lima
- Glauder Guimarães Ghinozzi
- Graziela Santos de Araújo
- Said Sadique Adi

ESTUDANTES

- Arthur Henrique Andrade Farias - Ciência da Computação
- Bruno Pereira Wesner da Silva - Engenharia de Computação
- Fernanda das Neves Merqueades Santos - Ciência da Computação
- Gabriel Pereira Falcão - Ciência da Computação
- Jenniffer Oliveira Checchia - Ciência da Computação
- Leonardo Vargas de Paula - Sistemas de Informação
- Marcos Gabriel da Silva Rocha - Engenharia de Computação
- Maria Paula do Nascimento Santos - Engenharia de Computação
- Nathanael Martins Wink - Ciência da Computação
- Victor Luiz Marques Saldanha Rodrigues - Ciência da Computação

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

- Adilson Carlos Batista
- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva
- Viviane Dziubate Pittner

Os materiais, aulas e projetos da “Robótica Paraná”, foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica. Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - CompartilhaIgual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)



DTI - DIRETORIA DE TECNOLOGIA E INOVAÇÃO