



Atenção aos sinais!

[[Cidades inteligentes]]

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

Darice Alessandra Deckmann Zanardini

Viviane Dziubate

Revisão Textual

Kellen Pricila dos Santos Cochinski

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Apoio Técnico

Equipe UFMS

2025

SUMÁRIO

Introdução	2
Objetivos	4
Roteiro da aula	5
1. Contextualização	5
2. Montagem e programação	9
3. Feedback e finalização	26
Referências bibliográficas	26

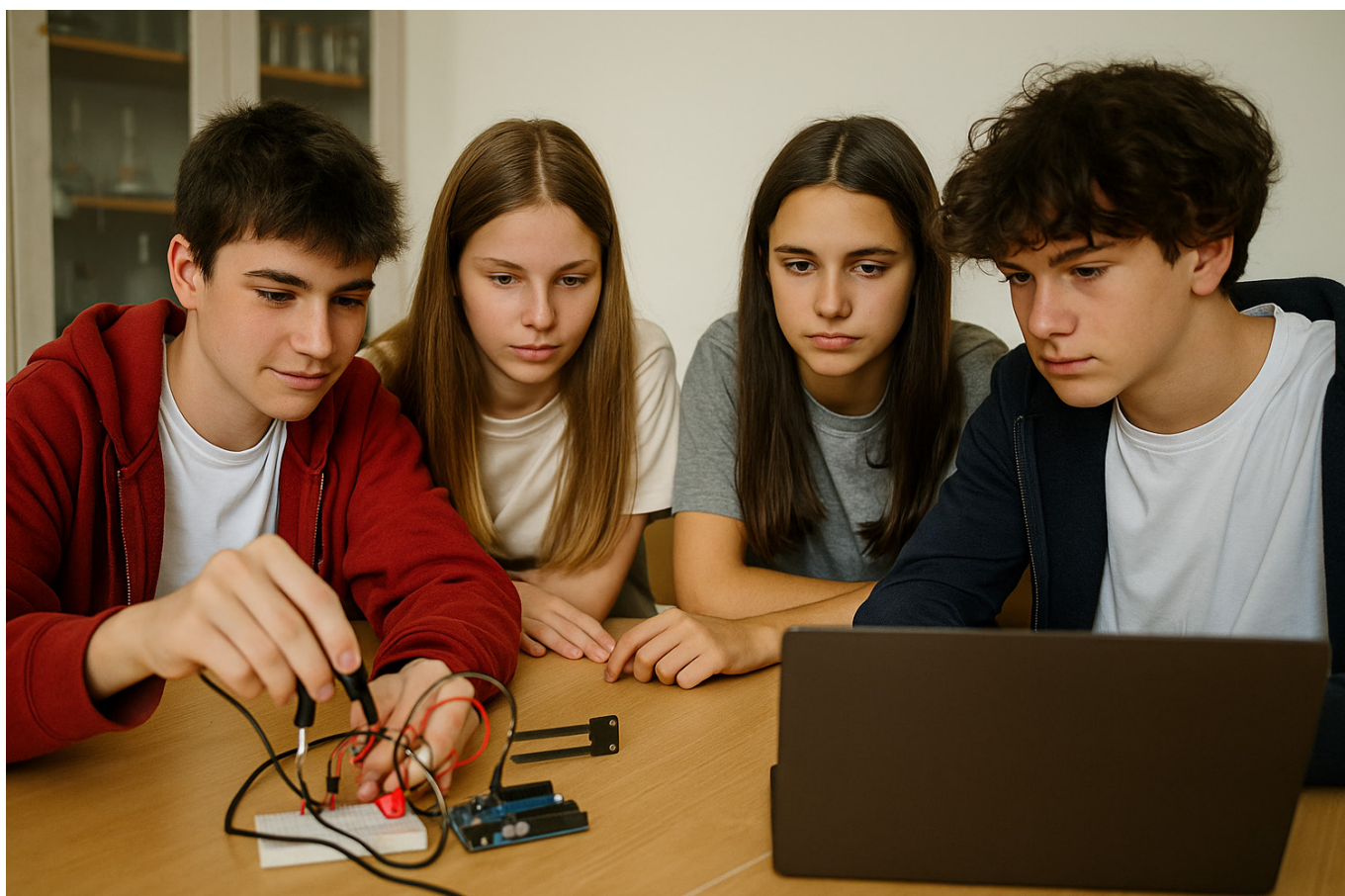
Atenção aos sinais!

[[Cidades inteligentes]]

Introdução

Ao longo deste ano, você e seus colegas desenvolveram uma série de protótipos para tornar uma casa inteligente, automatizando-a para que consumisse menos energia e fosse mais funcional. Tudo isso com os componentes do kit de Robótica Educacional. Vocês perceberam que a automação residencial é uma realidade concreta. Será que é possível automatizar também partes de uma cidade, para alertar quanto a perigos climáticos, ambientais, entre outros?

Essa é a proposta e desafio desta aula, Atenção aos sinais - cidades inteligentes. A relevância desse tema se encontra nos noticiários e manchetes, frente ao aquecimento global e mudanças climáticas. Todos os anos, ocorrem no Brasil e no mundo todo, situações de alagamentos, desmoronamentos e deslizamentos de terra. Nesta aula, você e seus colegas, com os componentes do kit de Robótica e o Arduino, criarão um protótipo de alerta para situações de risco.



Atenção aos sinais!

[[Cidades inteligentes]]



Os livros de geografia descrevem os deslizamentos de terra como movimentos de massa que ocorrem principalmente em encostas com solos pouco compactos e alta capacidade de absorção de água. Quando o solo fica saturado devido às chuvas intensas, especialmente em solos formados sobre rochas impermeáveis, a água se acumula, aumentando a umidade e reduzindo a coesão do solo, o que pode levar ao seu deslizamento morro abaixo.

A umidade elevada do solo é um dos principais indicadores de risco iminente de deslizamento. Portanto, monitorar a umidade em pontos estratégicos das encostas pode fornecer dados valiosos para alertas preventivos. O sensor de umidade do solo, acoplado ao Arduino e ao comparador LM393, permite detectar níveis críticos de umidade, sinalizando quando o solo está próximo da saturação, o que pode desencadear deslizamentos. Embora o sensor não cubra grandes áreas, sua aplicação em múltiplos pontos estratégicos permite mapear zonas de risco com maior precisão. A combinação de sensores distribuídos, análise geológica e monitoramento climático contribui para um sistema eficaz de prevenção de deslizamentos, salvando vidas e protegendo comunidades vulneráveis.

Atenção aos sinais!

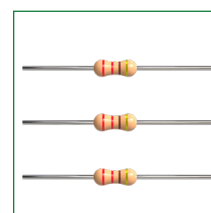
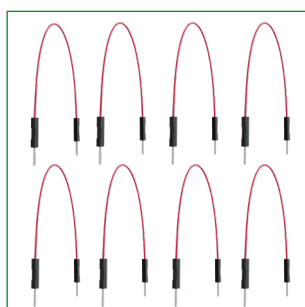
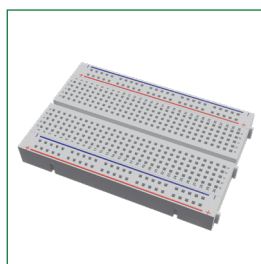
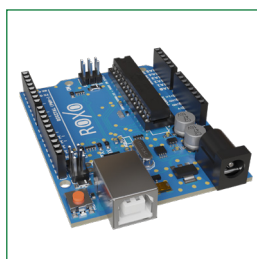
[[Cidades inteligentes]]

Objetivos desta aula

- Entender o papel e a importância da robótica e dos sensores na automação e monitoramento ambiental;
- Compreender como os sensores de umidade medem a quantidade de água no solo e como essa informação pode ser utilizada para prever situações de deslizamentos;
- Conhecer ferramentas e técnicas para monitorar dados em tempo real e analisar padrões que possam indicar riscos de deslizamentos.

Lista de materiais

- 1 Arduino Uno;
- 1 Sensor de umidade do solo;
- 8 Jumpers;
- 1 Protoboard;
- 3 LEDs de cores distintas para definir cada nível de umidade;
- 3 Resistores 220R;



Atenção aos sinais!

[[Cidades inteligentes]]



Roteiro da aula

1. Contextualização

Quando se fala em Cidades Inteligentes ou *Smart Cities*, entende-se que é uma cidade eficiente, conectada e sustentável. Ela utiliza as *Tecnologias da Informação e Comunicação* (TIC) para coletar dados em tempo real e gerenciar a cidade de forma mais inteligente. Isso é possível graças a sensores, câmeras, aplicativos e internet, usados para ajudar a resolver problemas, economizar recursos, preservar o meio ambiente e melhorar a qualidade de vida dos moradores.



Atenção aos sinais!

[[Cidades inteligentes]]

VOCÊ SABIA?

O termo “Cidade Inteligente” começou a ganhar força na década de 1990, com a popularização da internet e dos computadores. Inicialmente, o foco era muito centrado na infraestrutura tecnológica: como conectar prédios, instalar fibras ópticas e modernizar as redes de comunicação. Para ser “inteligente”, a cidade precisa colocar as pessoas no centro de tudo e incluir a sustentabilidade, a democracia e a justiça social, com a melhoria dos serviços públicos como saúde, educação e segurança.

Não existe uma única “certificação” ou um órgão mundial que carimba uma cidade como “inteligente”. Na prática, essa definição vem de diversos especialistas, rankings e organizações que analisam as cidades com base em vários critérios. Um dos mais conhecidos é o índice IESE Cities in Motion, que avalia as cidades em 9 dimensões:

- **Capital humano:** estratégias para o desenvolvimento pessoal e profissional dos cidadãos, com foco na educação de qualidade.
- **Governança:** eficácia da intervenção estatal na gestão de recursos, transparência e ética do governo.
- **Planejamento urbano:** cidades bem-organizadas, com soluções para a conectividade e autossuficiência dos bairros, o planejamento de infraestrutura, o sistema de gestão de resíduos e a distribuição de energia, moradia digna e espaços públicos de qualidade.
- **Tecnologia:** as tecnologias atuam como carro-chefe, já que o desenvolvimento tecnológico proporciona soluções para sustentabilidade, segurança pública, desenvolvimento humano, etc.
- **Meio ambiente:** iniciativas que ajudem a conter a poluição e problemas ambientais, promovam o uso de energias renováveis e a gestão de resíduos.
- **Coesão social:** pertencimento da população como iniciativas como desenvolvimento comunitário, combate ao preconceito e acessibilidade;
- **Mobilidade e transporte:** transporte público eficiente, bem-planejado, integrado e com baixas emissões de carbono, otimizam a qualidade de vida dos cidadãos.
- **Economia:** ambiente favorável à economia local, à criação de planos estratégicos e estímulo ao empreendedorismo e geração de empregos.
- **Reconhecimento internacional:** refere-se aos planos estratégicos de turismo e infraestrutura que promovam o reconhecimento internacional da cidade.

Atenção aos sinais!

[[Cidades inteligentes]]

No Rio de Janeiro, uma das cidades brasileiras consideradas *Smart City*, foi criado o Centro de Operações (COR) que, por meio de uma plataforma de monitoramento, integra dados de diferentes fontes para ajudar a gerenciar crises, trânsito e condições climáticas. Junto com a NASA e o Google, o COR desenvolveu um algoritmo de matriz de decisão composto por variáveis como mobilidade; chuva; deslizamentos e inundações; zonas de calor; e eventos extremos, além de informações enviadas pelos cidadãos. Dessa forma, ele consegue acompanhar e divulgar os estágios da situação operacional da cidade, todos divididos em escalas de 1 a 5, inspiradas na Escala Richter, com extremos que variam da normalidade aos estágios de crise, dependendo da intensidade e interação das circunstâncias. Essa solução é essencial para pessoas que vivem em regiões com risco de deslizamentos, pois quando esses ocorrem deixam prejuízos físicos, mas com o alerta a tempo pode salvar vidas.

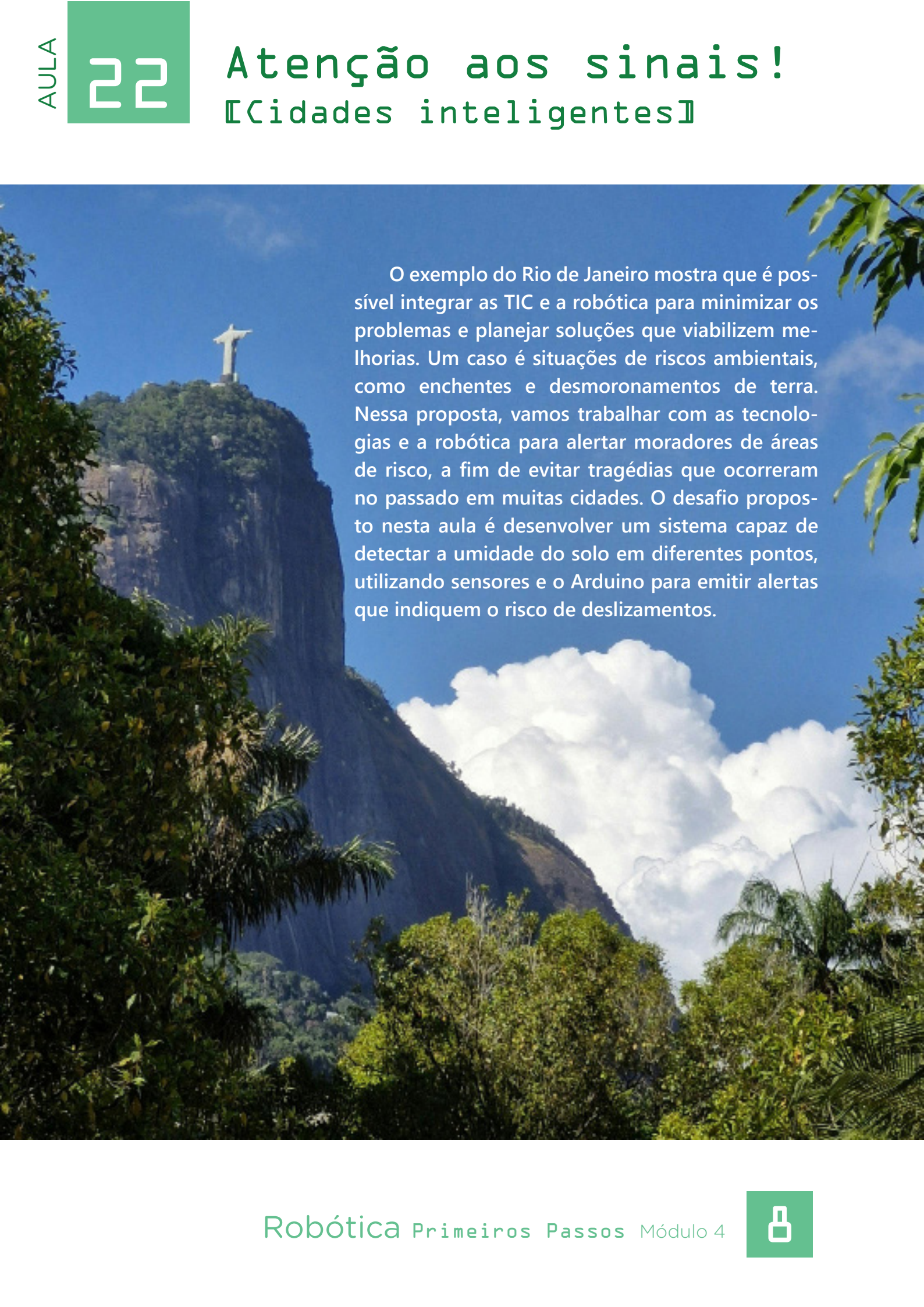
Figura 1 – Deslizamento de terra



Fonte: Franklin Peña Gutierrez – Pexels, 2016.

Atenção aos sinais!

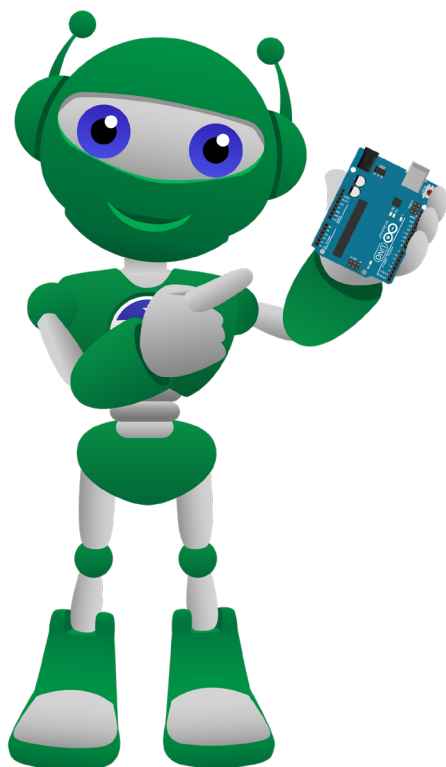
[[Cidades inteligentes]]

A scenic view of the Christ the Redeemer statue perched atop Corcovado Mountain in Rio de Janeiro. The mountain's steep, rocky slopes are partially covered in lush green vegetation. The foreground is filled with dense tropical foliage, including palm trees. The sky is a vibrant blue, dotted with large, fluffy white clouds. The overall atmosphere is bright and clear, suggesting a sunny day.

O exemplo do Rio de Janeiro mostra que é possível integrar as TIC e a robótica para minimizar os problemas e planejar soluções que viabilizem melhorias. Um caso é situações de riscos ambientais, como enchentes e desmoronamentos de terra. Nessa proposta, vamos trabalhar com as tecnologias e a robótica para alertar moradores de áreas de risco, a fim de evitar tragédias que ocorreram no passado em muitas cidades. O desafio proposto nesta aula é desenvolver um sistema capaz de detectar a umidade do solo em diferentes pontos, utilizando sensores e o Arduino para emitir alertas que indiquem o risco de deslizamentos.

Atenção aos sinais!

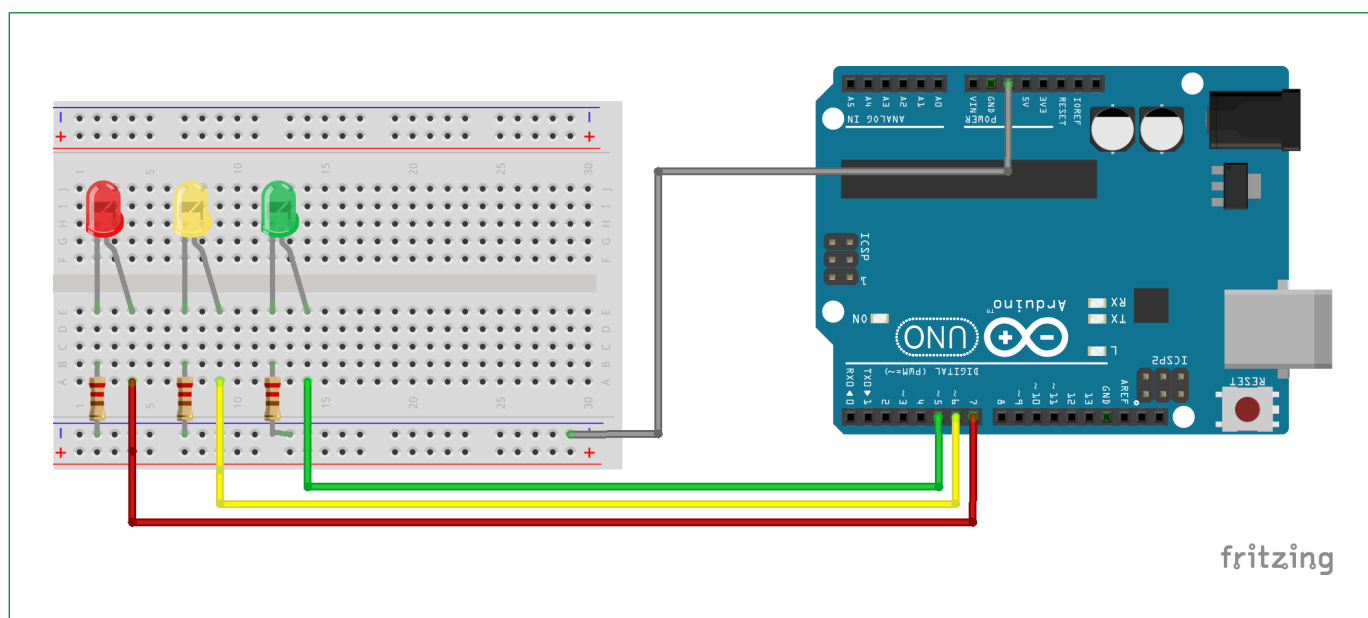
[[Cidades inteligentes]]



Montagem

A montagem dos componentes para essa aula é parecida com o que foi feito nas aulas 18, 19 e 20 - Plantas Felizes. A proposta é conectar à protoboard 3 LEDs de cores diferentes, que irão indicar o nível de umidade do solo no qual está o sensor. Lembre-se de conectar um resistor 220 Ω associado a cada LED. Conecte os LEDs nas portas digitais 5, 6 e 7. Lembre-se de energizar a protoboard com jumper macho-macho na lateral negativa, a qual terá um terminal de cada resistor colocado. A montagem dos LEDs e resistores pode ser conferida na Figura 2.

Figura 2 - Protótipo com LEDs



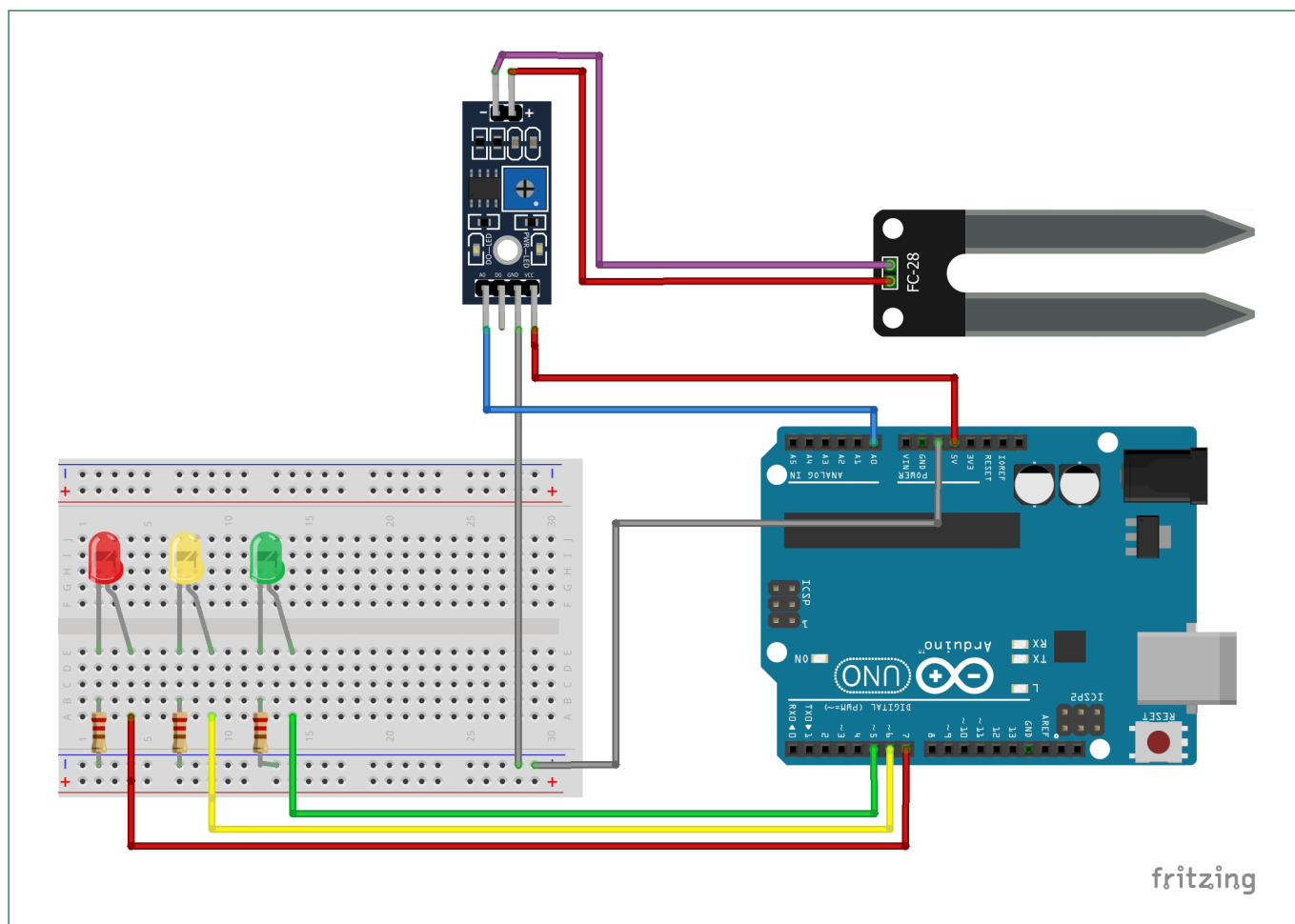
Fonte: Fritzing, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

Agora, conecte o sensor de umidade na placa Arduino, conectando a porta A0 do sensor à porta A0 do Arduino. Conecte a porta GND do sensor na lateral negativa da protoboard (já energizada) e o terminal VCC na porta 5V do Arduino. A montagem completa pode ser vista na Figura 3.

Figura 3 - Protótipo completo



Fonte: Fritzing, 2025.

Por fim, providencie uma amostra de areia ou terra, na qual será testada a umidade por meio do sensor.

Atenção aos sinais!

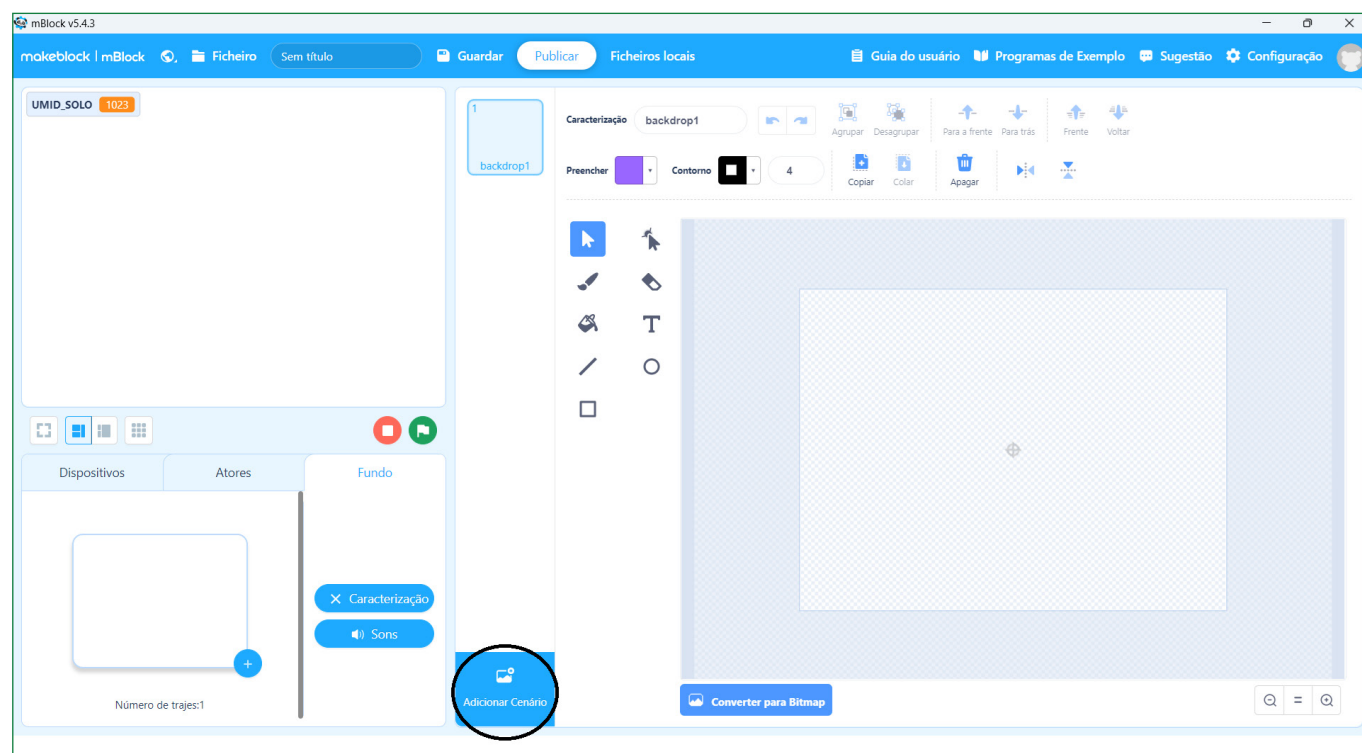
[[Cidades inteligentes]]

Programação

Na programação em blocos do nosso circuito, podemos iniciar criando um projeto referente a essa aula 22, nomeando-o como Cidade Inteligente. Altere para o Modo Viver do mBlock, para termos as respostas do sensor em tempo real.

Antes de iniciarmos a programação em si, precisamos ajustar os cenários que iremos utilizar, para isso, assim que abrir o mBlock, vá em “Fundo” e em seguida vá em “Caracterização”, deverá abrir uma tela parecida com a Figura 4, então vá em “Adicionar caracterização”.

Figura 4 - Tela de caracterização



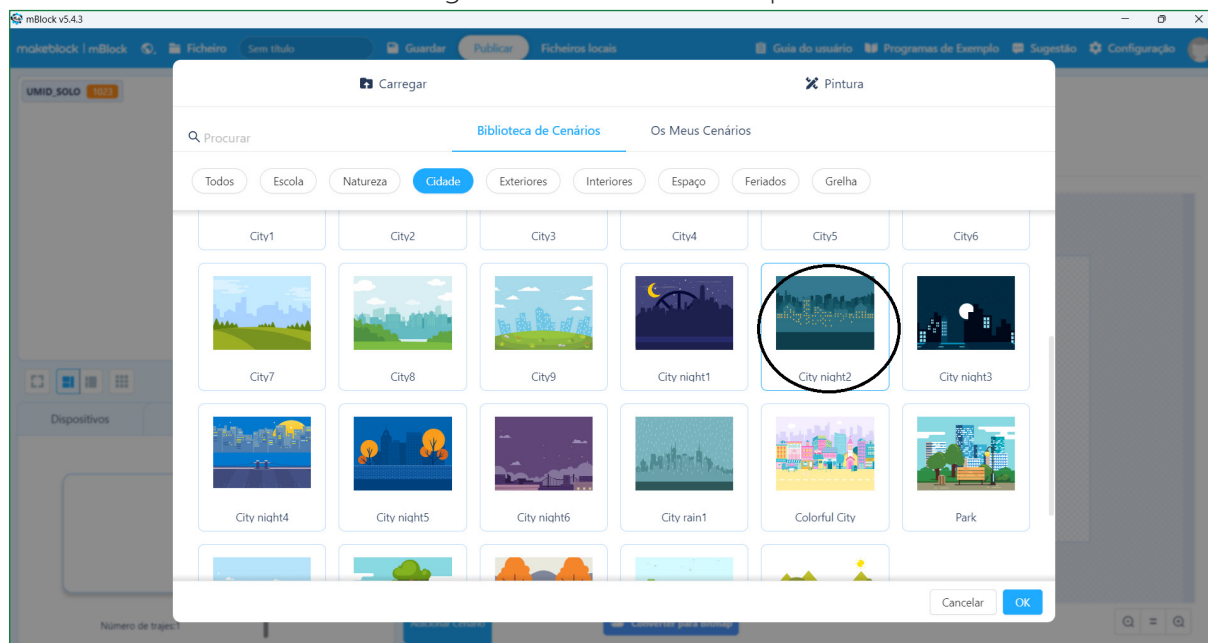
Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

A seguir, adicionaremos o cenário “City night 2” como mostra a Figura 5.

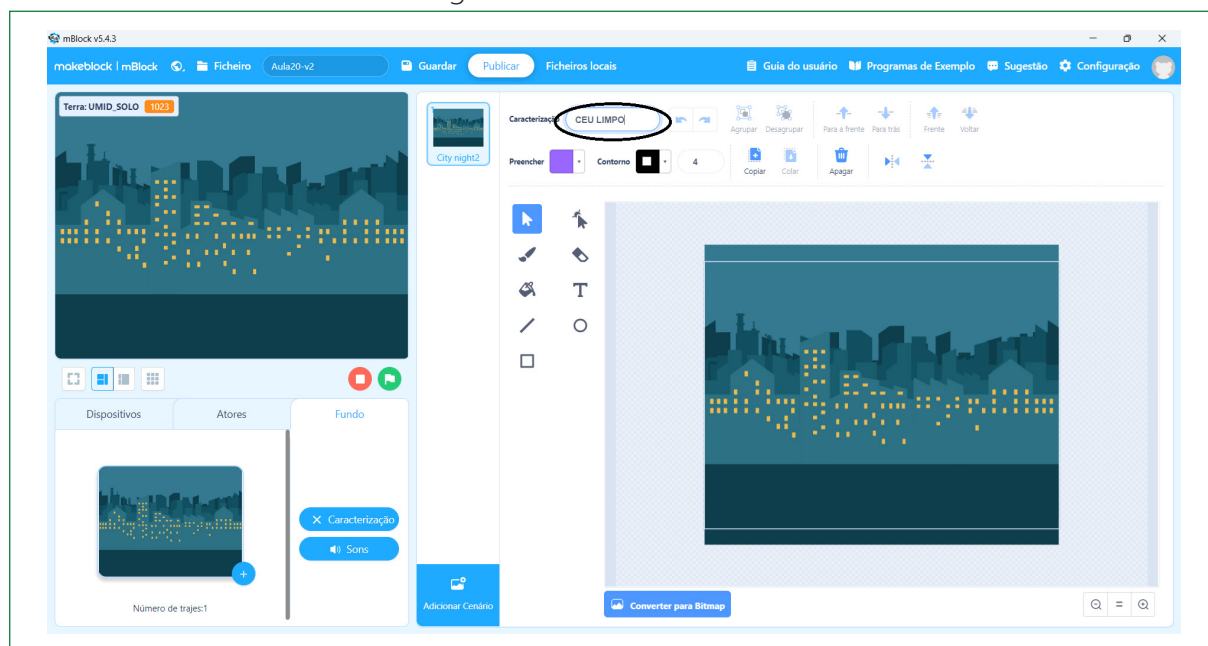
Figura 5 - Cenário céu limpo



Fonte: mBlock, 2025.

Para facilitar nosso entendimento, vamos alterar o nome dele (Figura 6).

Figura 6 - Nome do cenário



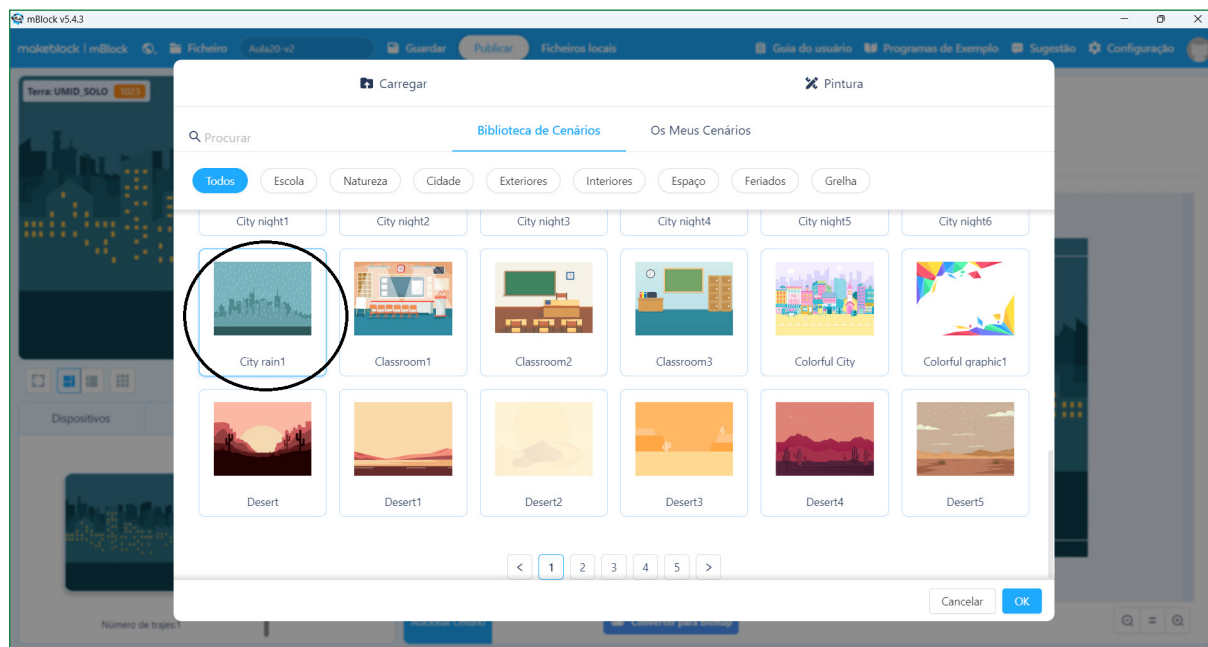
Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

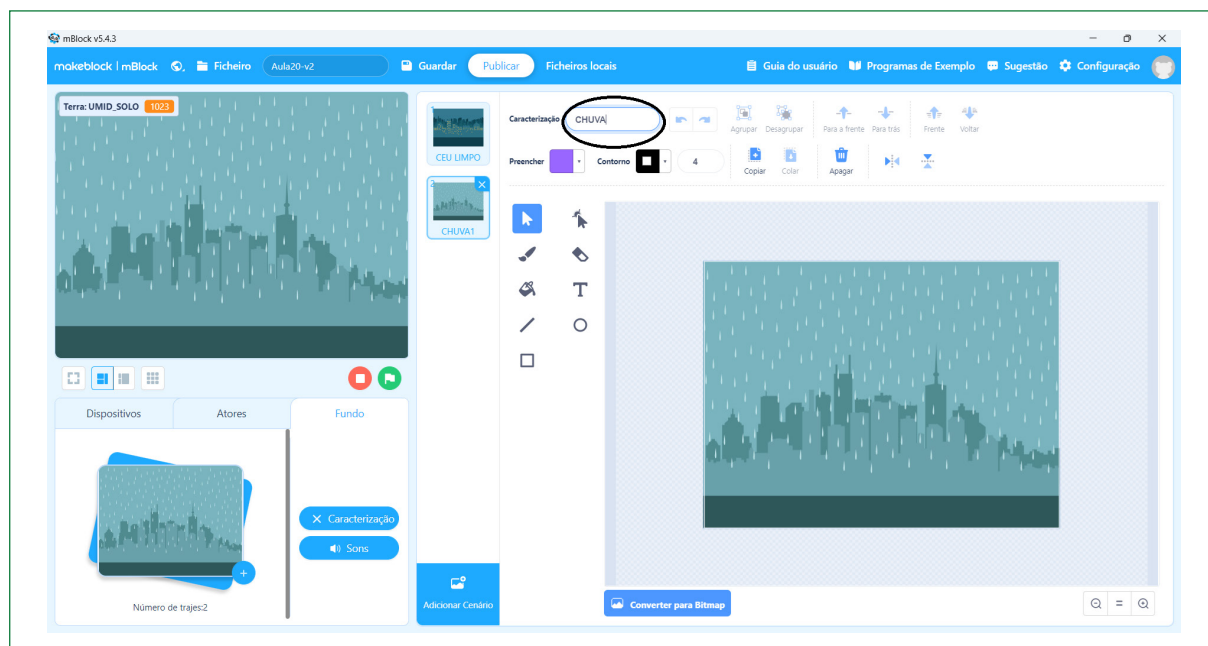
Agora, adicione o cenário de chuva da mesma maneira que o anterior e renomear, como mostram as Figuras 7 e 8.

Figura 7 - Cenário chuva



Fonte: mBlock, 2025.

Figura 8 - Renomeando o cenário



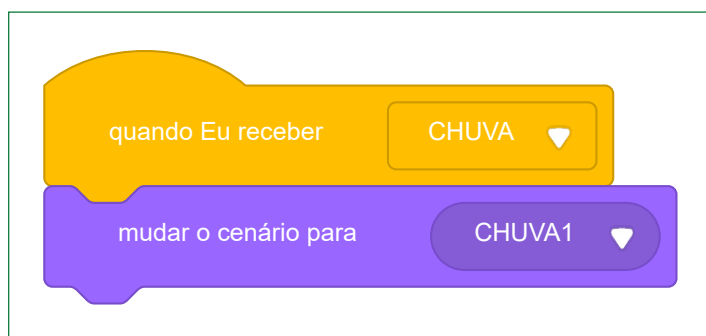
Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

Figura 9 - Programação do cenário

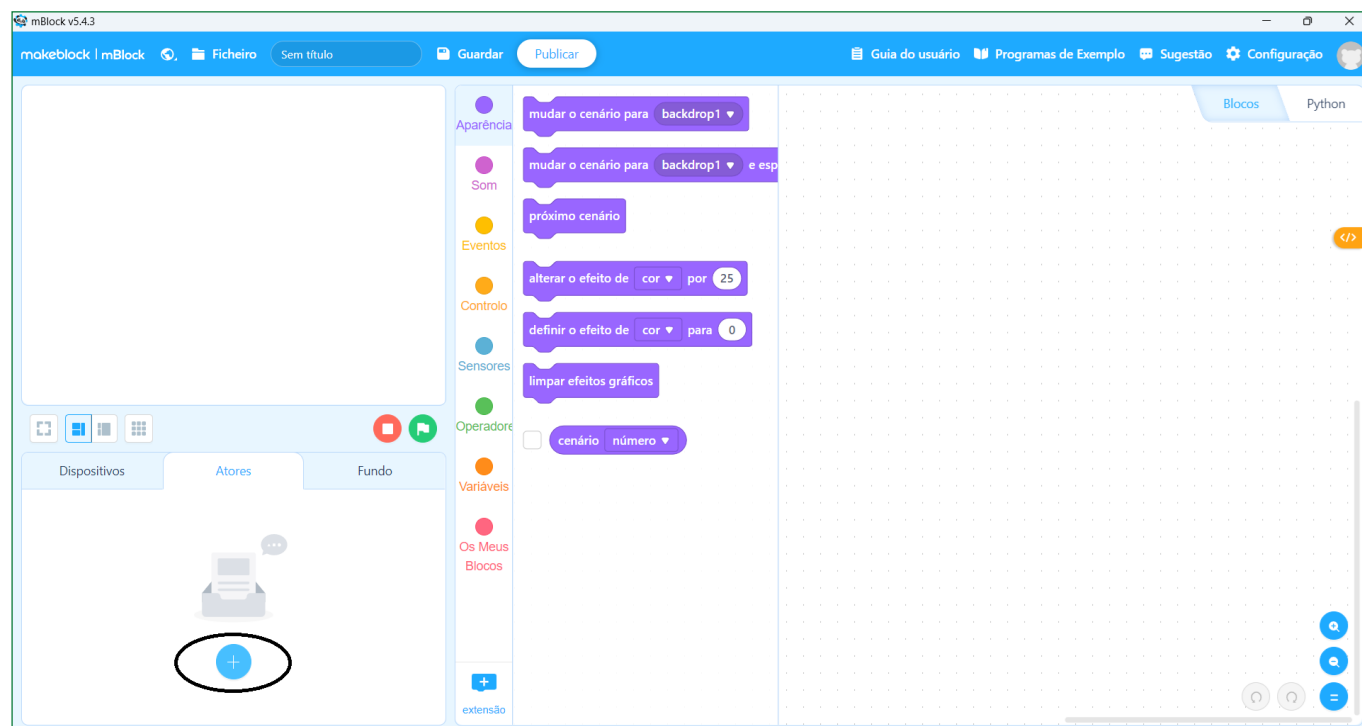
Agora, no cenário, programaremos para que, em caso de chuva, a imagem mude para cidade com chuva. Para isso, basta seguir a Figura 9. Quando receber a mensagem “Chuva” significa que começou a chover e o cenário deve mudar.



Fonte: mBlock, 2025.

Agora que o nosso cenário está programado, vamos programar o Ator “Terra”, para que ele apareça em 2 situações, sutilmente quando identificar a chuva e o led ficar amarelo e bem grande na tela quando a chuva ficar intensa e o led ficar vermelho, indicando assim um alto risco de deslizamento. Para isso, baixe a imagem sugerida para o ator, disponível nesse link: <http://bit.ly/4mYpkEX> e siga os passos mostrados nas Figuras 10, 11 e 12.

Figura 10 - Adicionando Ator Terra (parte 1)

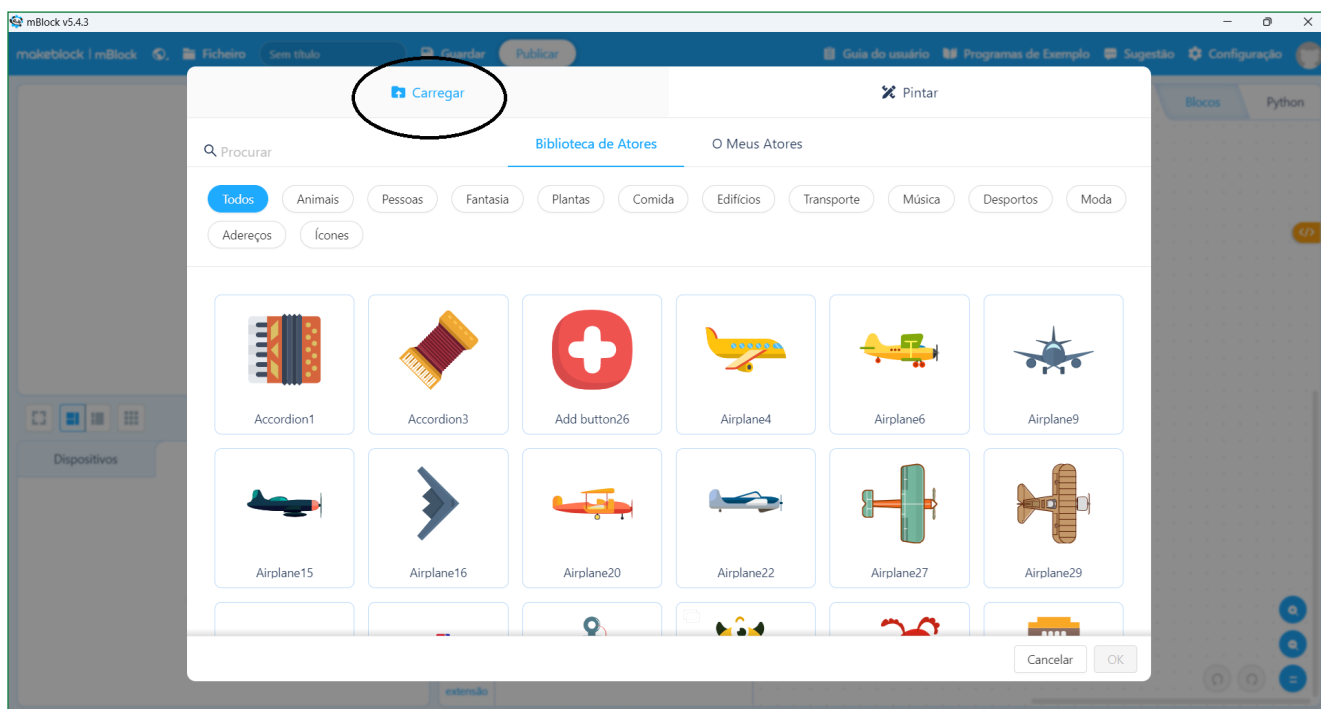


Fonte: mBlock, 2025.

Atenção aos sinais!

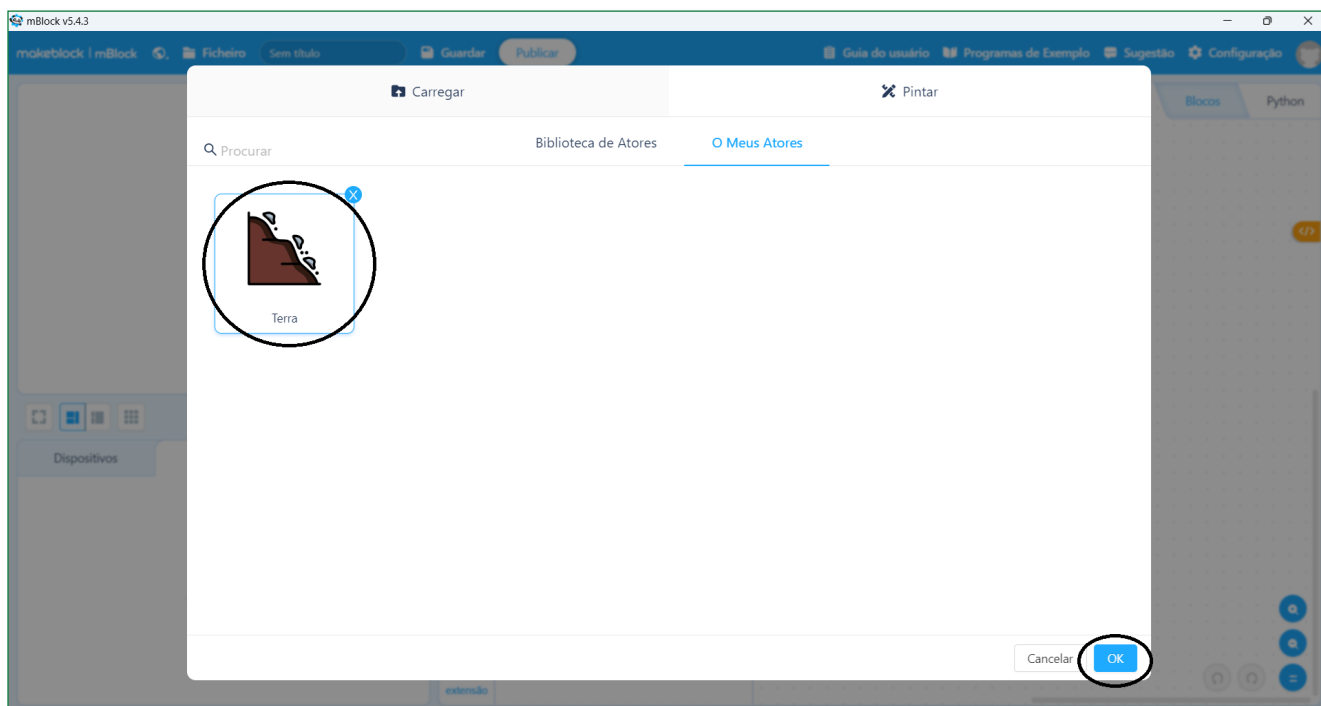
[[Cidades inteligentes]]

Figura 11 - Adicionar Ator Terra (parte 2)



Fonte: mBlock, 2025.

Figura 12 - Adicionar Ator Terra (parte 3)



Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

Vamos dar início à programação do ator Terra. Ele só aparecerá se o cenário estiver no modo “CHUVA”, e se o sensor de umidade identificar leitura maior que 1000.

O funcionamento é o seguinte:

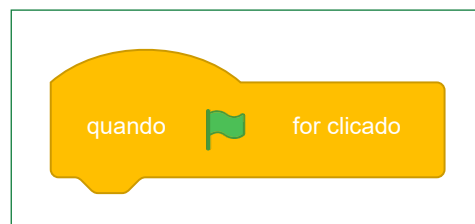
- Se a leitura for **MAIOR** que **1000** (solo seco): o risco de deslizamento é zero.
- O cenário exibe um céu limpo.
- O **LED verde** acende, indicando segurança.
- O ator Terra **NÃO** aparece.

Para isso, inicie o código com o bloco indicado na Figura 13 e, em seguida, insira o bloco < **repetir para sempre** >, como mostra a Figura 14.

Agora, vamos começar as parametrizações, lembrando que o cenário “CHUVA” será apresentado. **IMPORTANTE!** A programação será feita no ator TERRA. Primeiro, será necessário verificar o cenário, para isso insira o bloco < **se... então...** > e dentro do parâmetro, insira o bloco < **operador igual** >. No primeiro campo desse bloco, insira o bloco < **cenário- nome** > e no segundo escreva CHUVA 1, como mostra a Figura 15.



Figura 13 - Início do código.



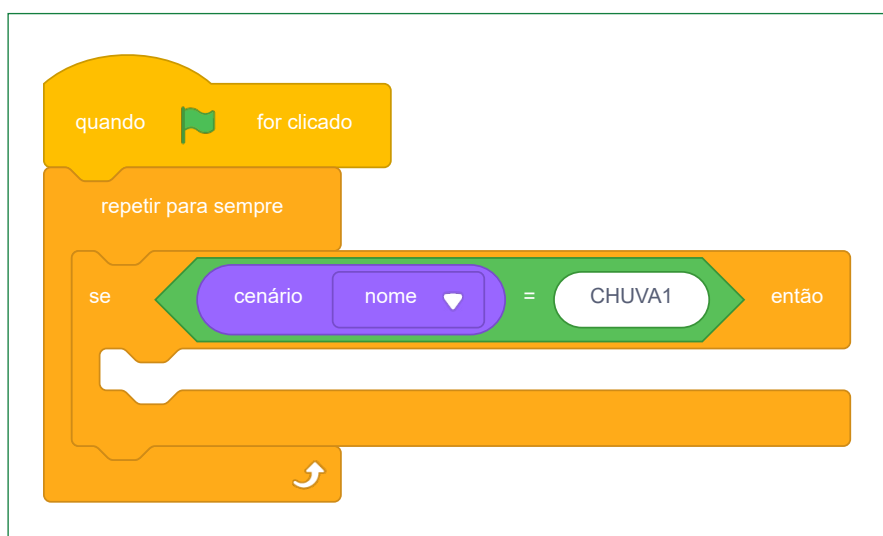
Fonte: mBlock, 2025.

Figura 14 - Repetir para sempre.



Fonte: mBlock, 2025.

Figura 15 - Verificando cenário



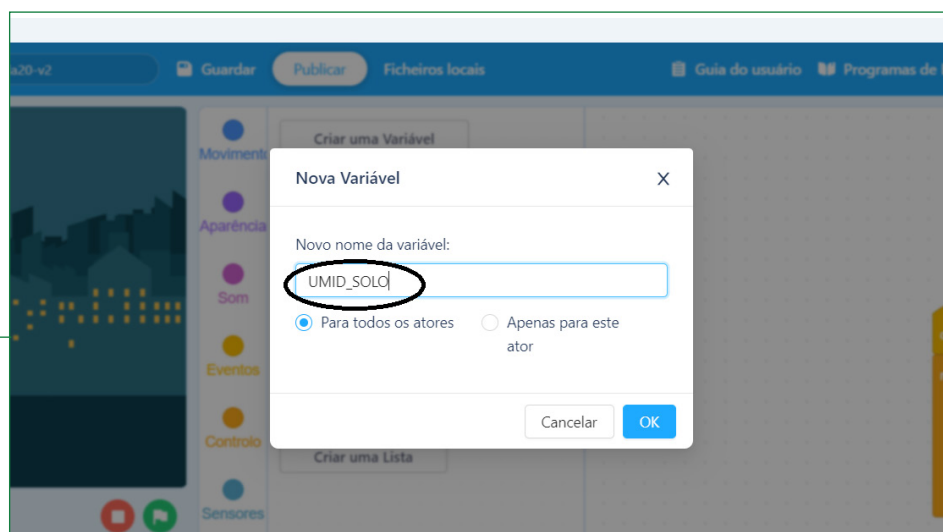
Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

Figura 16 - Criando variável

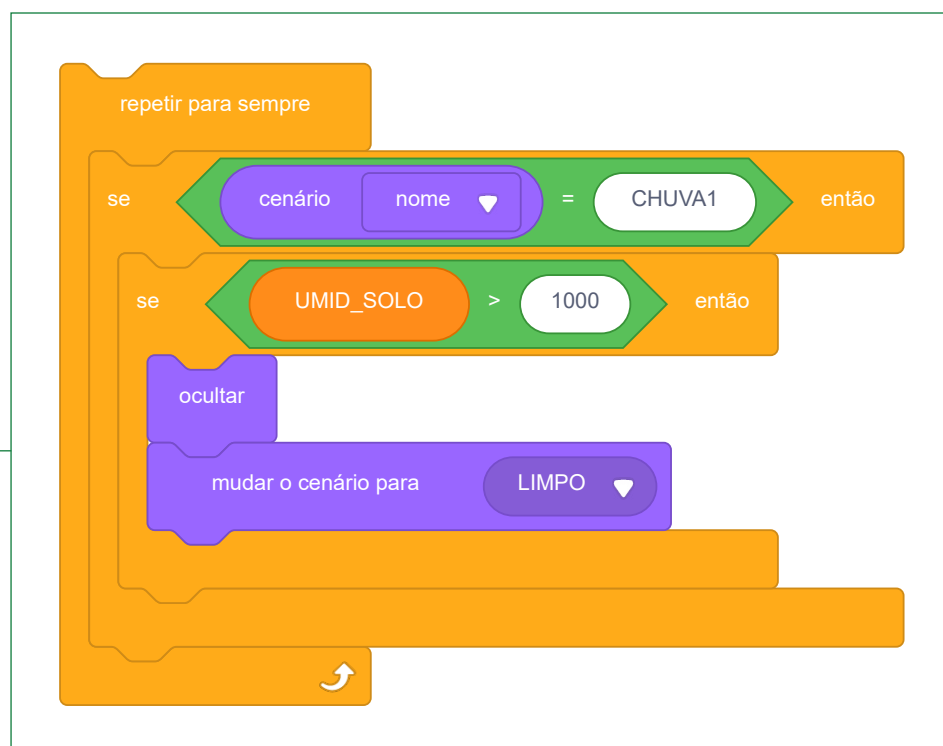
Agora, precisamos de uma variável para armazenarmos o valor lido pelo sensor, então vamos criá-la, como mostra a Figura 16.



Fonte: mBlock, 2025.

Figura 17 - Primeiro parâmetro

Caso a minha variável "UMID_SOLO" receba um valor maior que 1000, o ator deverá ficar oculto e o cenário deve ficar em "LIMPO", como apresentado na sequência de blocos da Figura 17.



Fonte: Autores.

Atenção aos sinais!

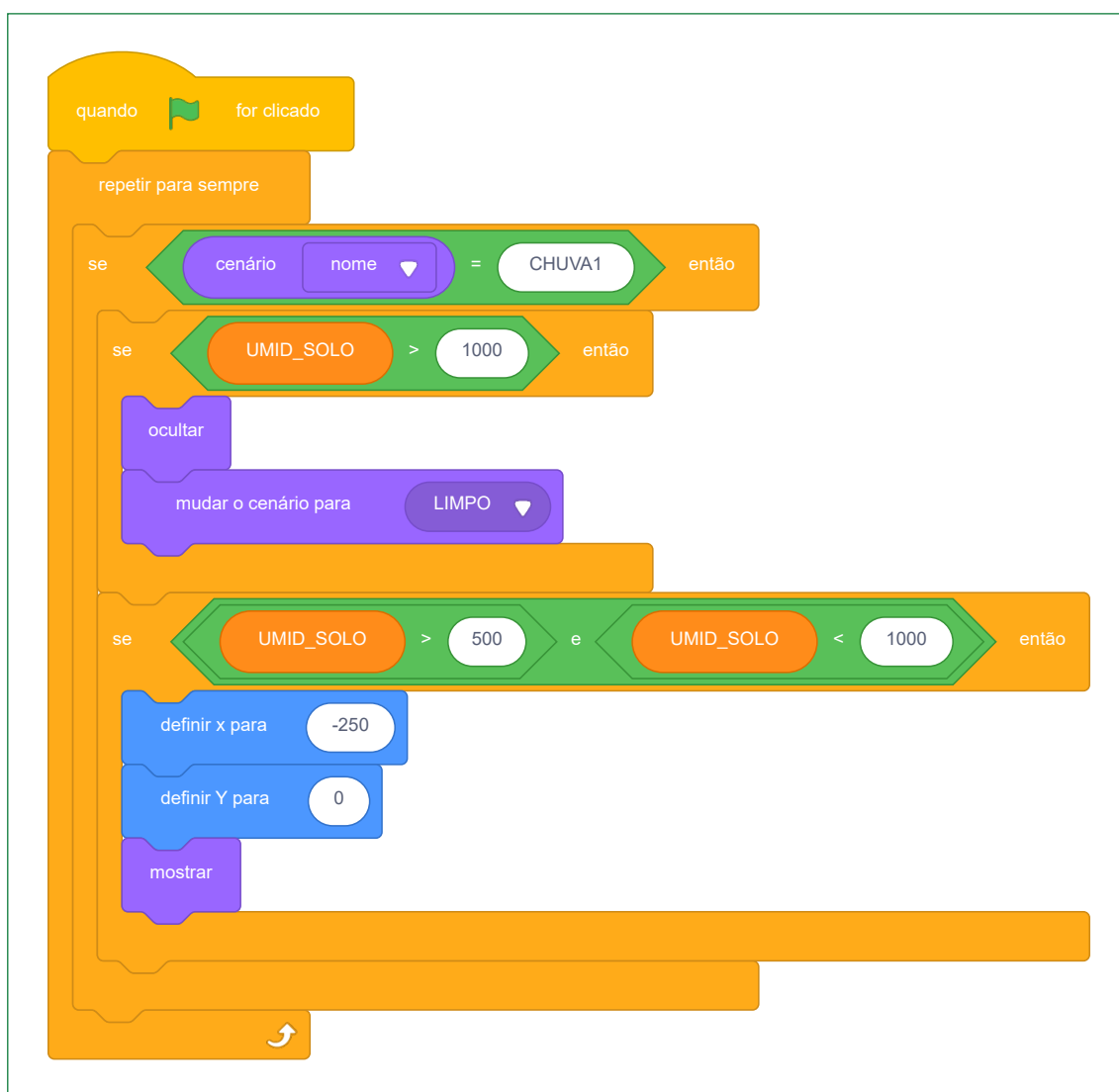
[[Cidades inteligentes]]

Agora, se for um valor maior que 500 e menor que 1000, o risco é médio, então vamos deslocar um pouco o nosso ator e fazer com que ele apareça. Então, para solo úmido, temos:

- O risco de deslizamento é mediano.
- O cenário muda para **CHUVA**.
- O **LED amarelo** acende, indicando atenção.
- O ator **Terra** aparece parcialmente na tela.

A Figura 18 mostra a segunda parametrização com os blocos <definir X para -250>, <definir y para 0> e <mostrar>.

Figura 18 - Segunda parametrização



Fonte: mBlock, 2025.

Atenção aos sinais!

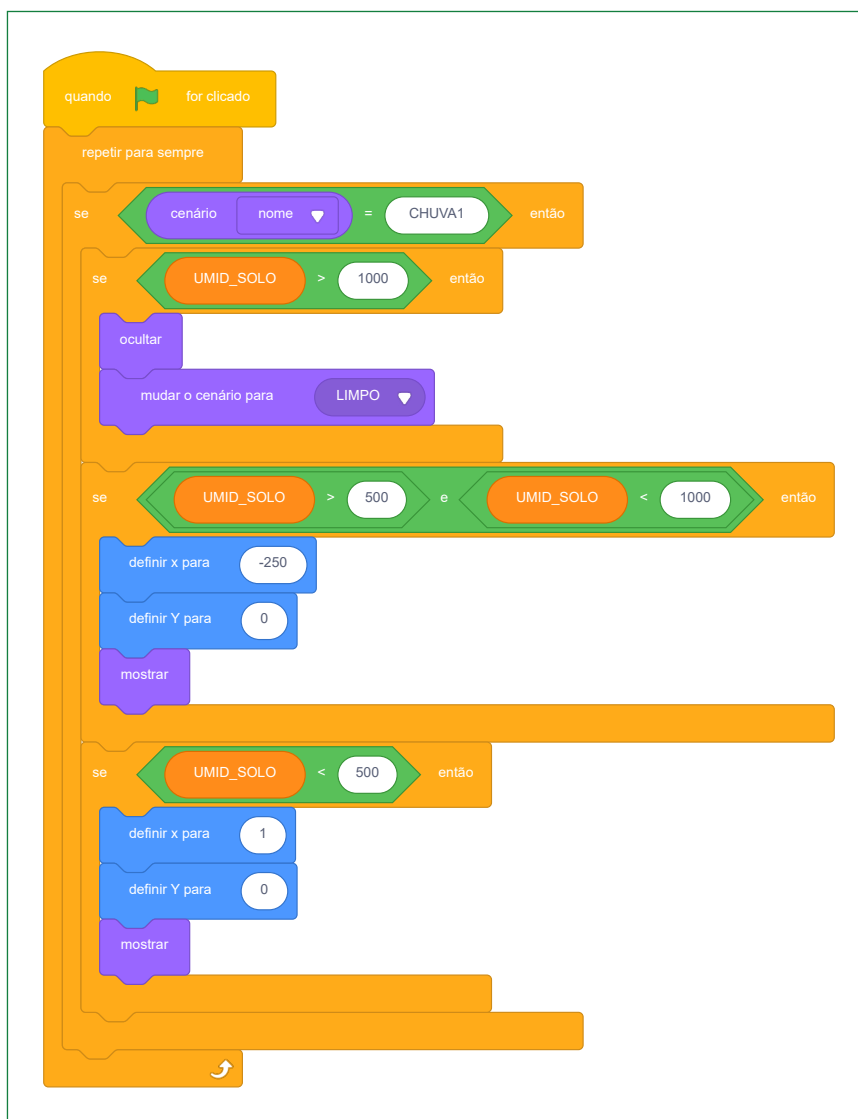
[[Cidades inteligentes]]

A última parametrização mostrará o que ocorre se a leitura for MENOR que 500 (solo encharcado):

- O risco de deslizamento é altíssimo.
- O cenário continua em **CHUVA**.
- O **LED vermelho** acende, indicando perigo.
- O ator **Terra** ocupa a tela inteira.

Nessa programação, serão adicionados os blocos <definir X para 1>, <definir y para 0> e <mostrar>.

Figura 19 - Última parametrização



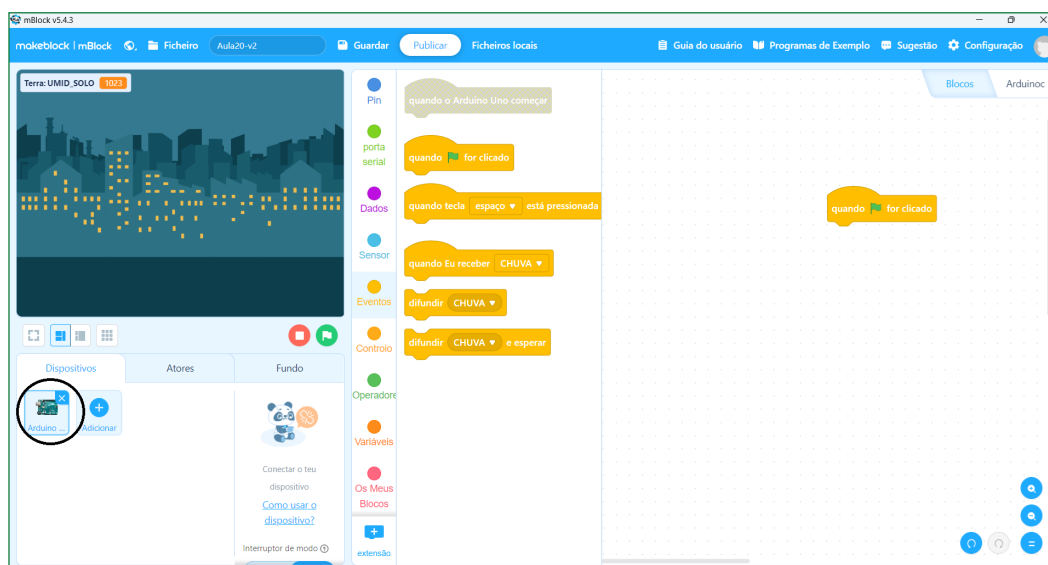
Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

Vamos configurar o nosso sensor de umidade do solo, basta clicar em dispositivos (Figura 20). Lembrando que utilizaremos o “Modo viver”.

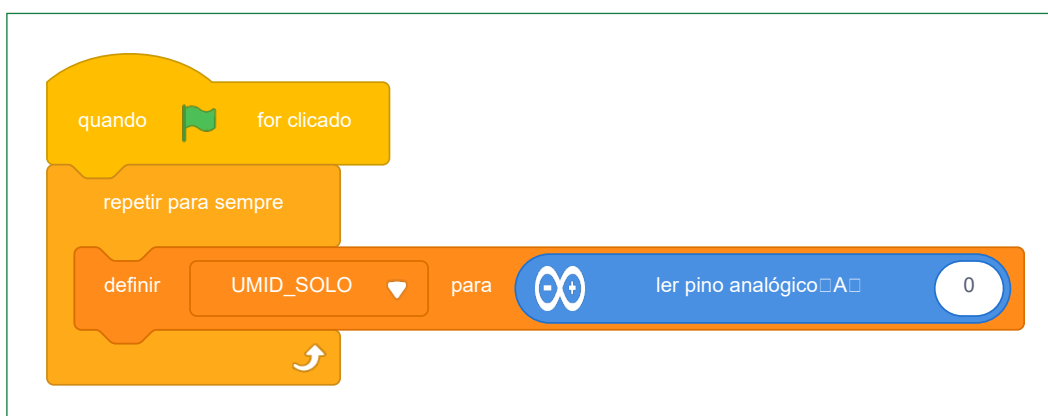
Figura 20 - Sensor de umidade



Fonte: mBlock, 2025.

Seguindo a mesma linha da Figura 14, vamos repetir para sempre a leitura do sensor. Nesse sentido, utilizaremos as parametrizações que vimos anteriormente para: armazenar a leitura na variável “UMID_SOLO” e acender os LEDs. A Figura 21 apresenta o bloco <definir UMID_SOLO para> e, dentro dele, o bloco <ler pino analógico (A) 0>, pino no qual foi conectado o sensor.

Figura 21 - Lendo a variável umidade do solo



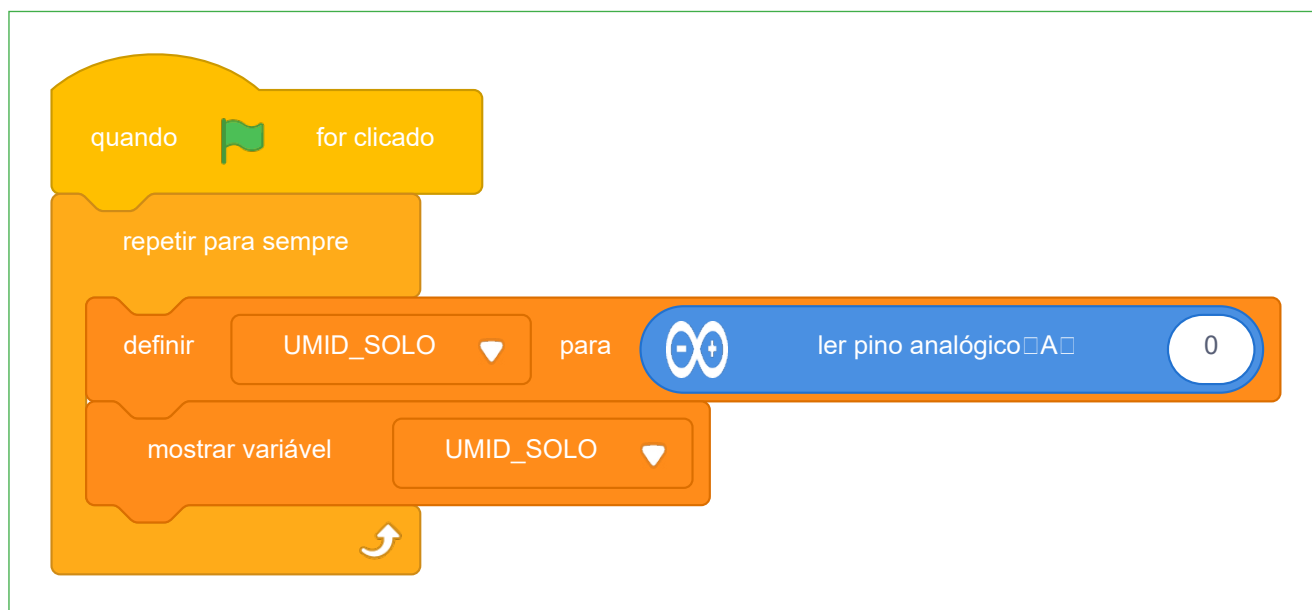
Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

Para facilitar a nossa compreensão, vamos mostrar o que nosso dispositivo está lendo, assim saberemos se está tudo ocorrendo bem com o nosso programa. Para tanto, insira o bloco <**mostrar variável UMID_SOLO**> como mostra a Figura 22.

Figura 22 - Mostrando a variável



Fonte: mBlock, 2025.



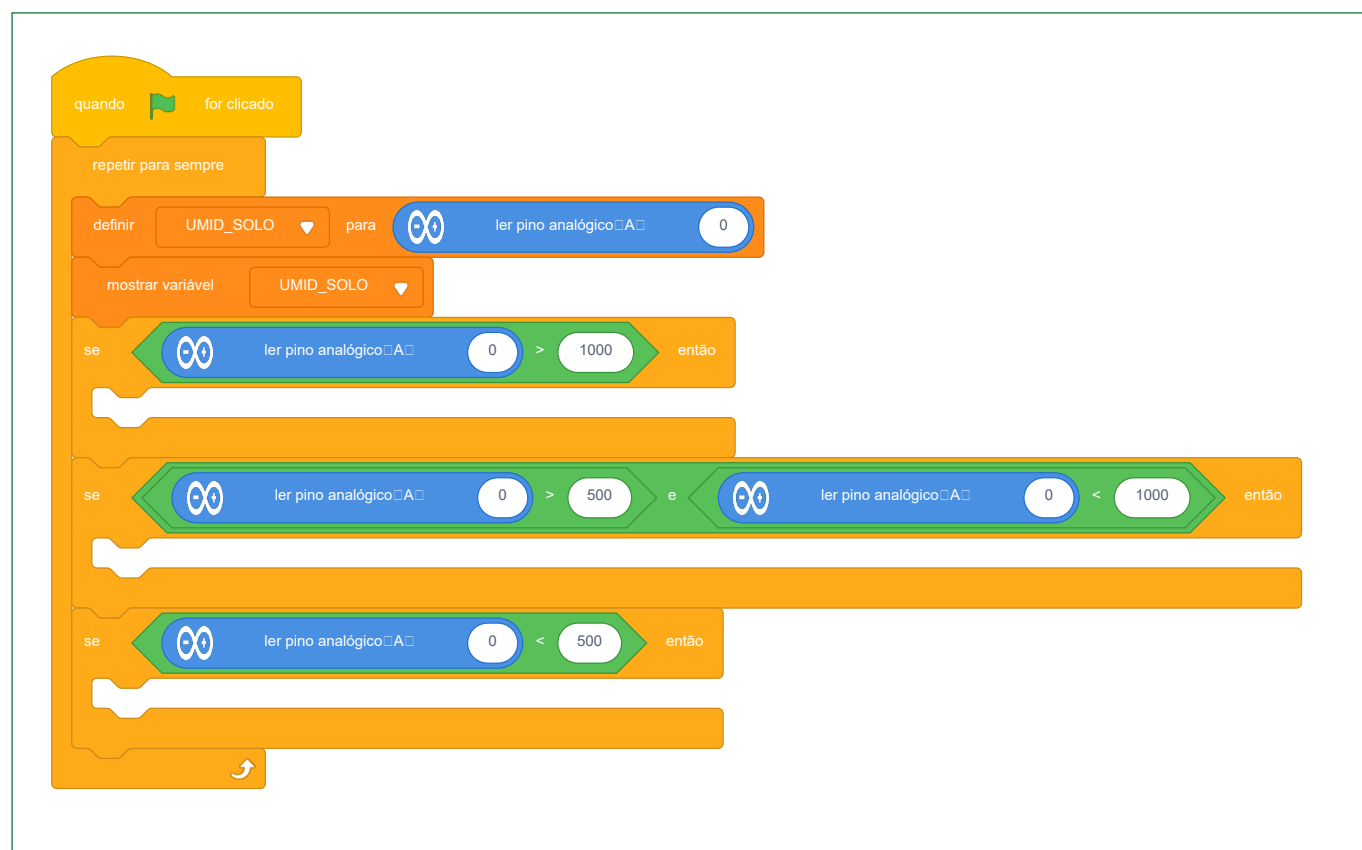
Atenção aos sinais!

[[Cidades inteligentes]]

Vamos repetir os passos da parametrização do ator TERRA, agora para o dispositivo. Para isso, duplique 3 vezes o bloco <se... então>. O primeiro será o parâmetro para terra SECA (MAIOR que 1000). Insira o bloco <operador maior>, e dentro dele o bloco <ler pino analógico (A) 0> e escreva 1000 no segundo campo do operador maior.

O segundo parâmetro precisará do bloco < operador e...>, no primeiro campo, o bloco <operador maior> e no segundo <operador menor>. Repita o bloco <ler pino analógico (A) 0> dentro de cada bloco e escreva o numeral conforme está na Figura 23.

Figura 23 - Inserção das condições



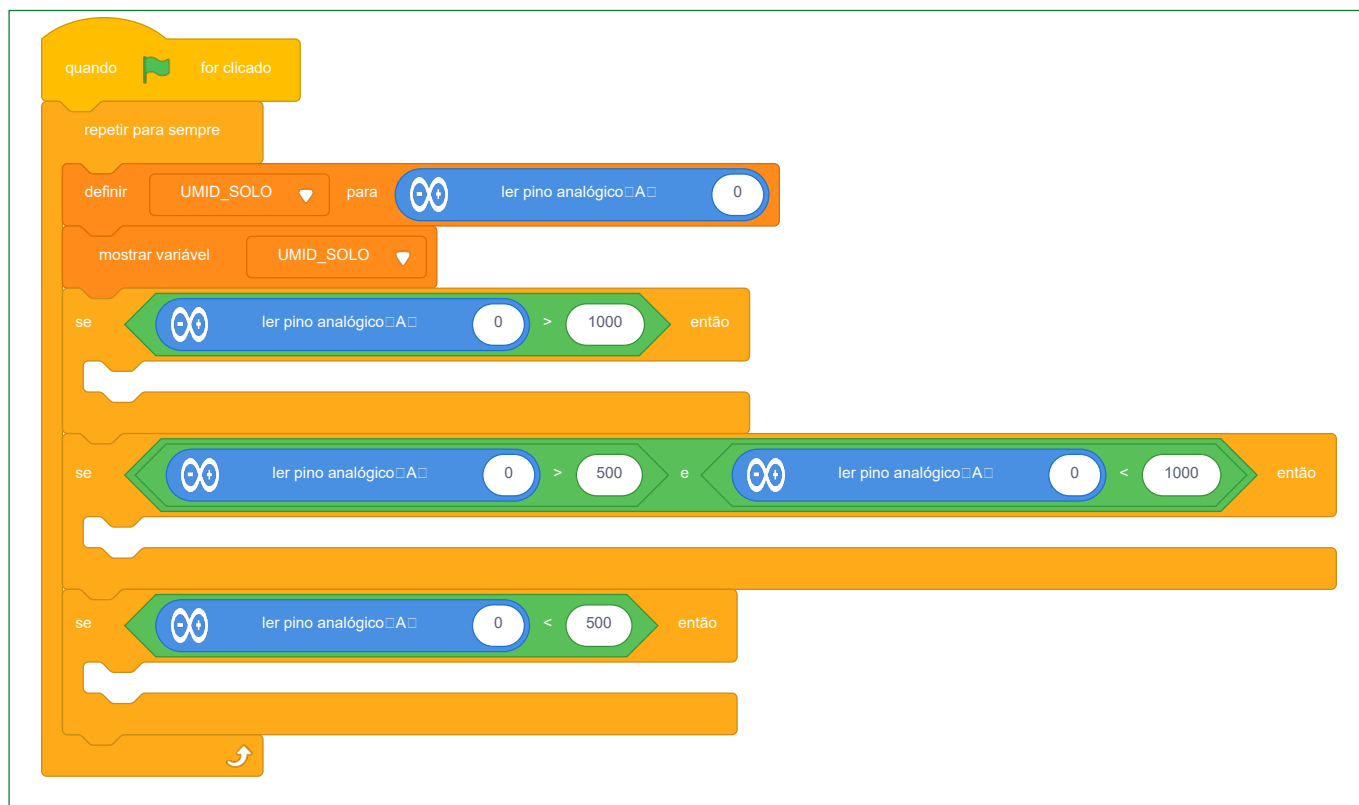
Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

No primeiro “se”, é quando o risco de deslizamento está zero, então só iremos acender o LED Verde, e a programação ficará como mostra a Figura 24.

Figura 24 - Definindo o LED verde como aceso



Fonte: mBlock, 2025.

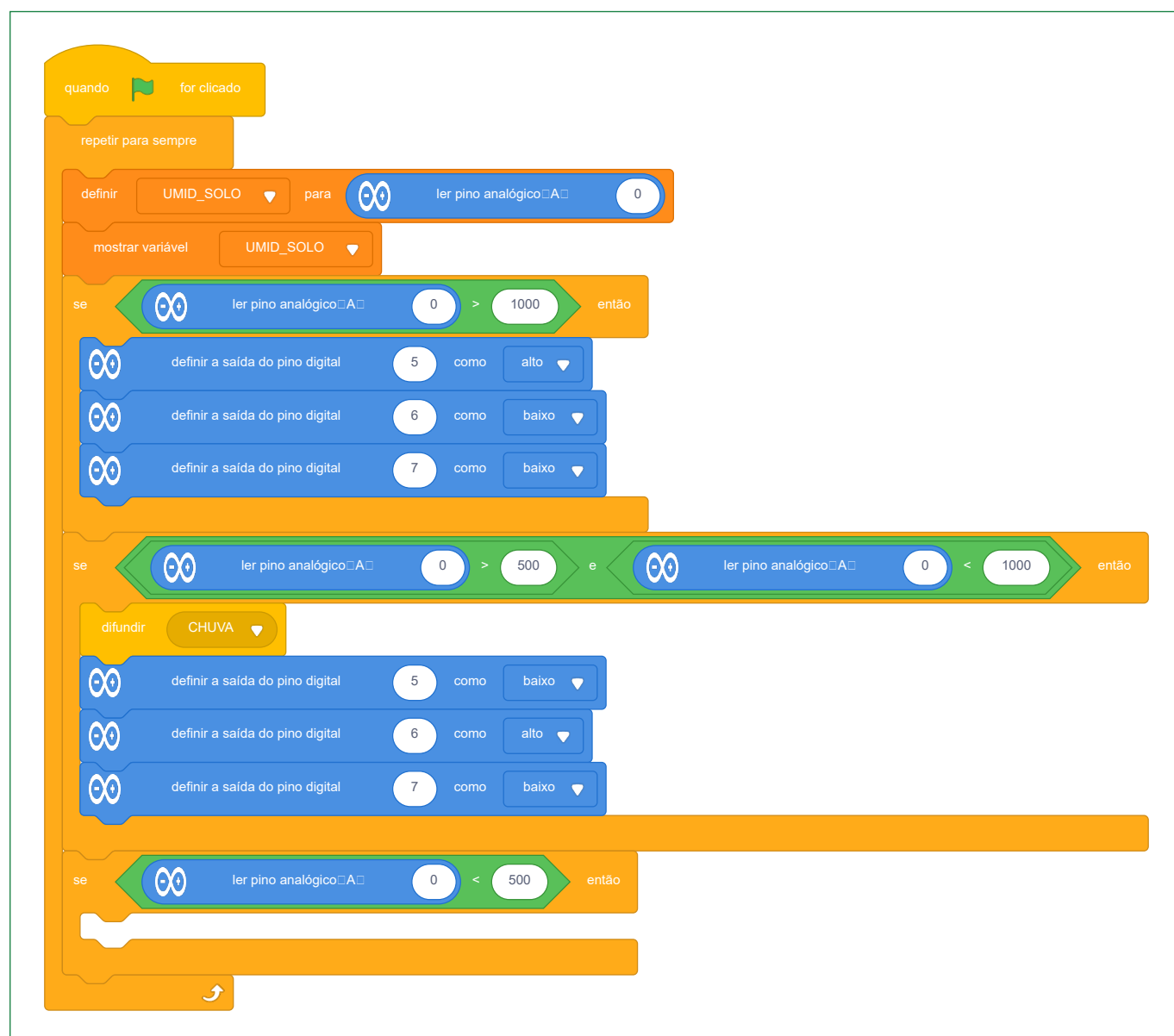


Atenção aos sinais!

[[Cidades inteligentes]]

Já o segundo “SE”, é o risco mediano, então temos que “Difundir a mensagem” de “CHUVA” para que ele entre no cenário chuvoso, também precisamos que ele acenda o LED amarelo. Observe a programação na Figura 25.

Figura 25 - Ligando o LED amarelo e mudando o cenário



Fonte: mBlock, 2025.

Atenção aos sinais!

[[Cidades inteligentes]]

Para fecharmos e finalizarmos o código completo, quando meu sensor identifica algum valor abaixo de 500, o LED vermelho deve acender e “Difundir a mensagem” de “CHUVA”. Observe a programação completa na Figura 26.

Figura 26 - Código completo



Fonte: mBlock, 2025.

Pronto, agora nosso código está funcionando e fazendo o sistema de alerta de deslizamento para a cidade inteligente. Utilize um recipiente com areia ou terra e um pouco de água para testar seu protótipo de alerta para deslizamentos.

Atenção aos sinais!

[[Cidades inteligentes]]

Desafios:

Que outros componentes poderiam ser úteis nesse protótipo, para alertar as pessoas em uma situação de risco como a desta aula?

Pense com seus colegas em inserir um buzzer ou outros componentes que ajudem a população a se preparar para momentos de chuvas que podem levar a deslizamentos.

E se...

O projeto não funcionar?

- Verifique se a programação está orientada corretamente ao ator, cenário e dispositivos.
- Observe a conexão dos jumpers nos pinos e se estão corretamente endereçados no código.

3. Feedback e finalização

Para finalizar nossa aula, compartilhem com seus colegas de equipe como foi sua experiência, respondendo às seguintes questões:

1. Vocês já haviam pensado sobre como a robótica pode ser utilizada em situações de desastres naturais?
2. Já passaram por situações como as descritas nessa aula, ou conheceram alguém que passou? A robótica poderia ajudar essas pessoas?
3. Já conhecia o conceito de *Smart City*?
4. Qual foi a sua parte favorita ao desenvolver o projeto de hoje? A montagem, a programação ou a fase de testes?

5. Vocês encontraram algum desafio durante a montagem ou programação? Como a equipe conseguiu superar esse obstáculo?
6. De que maneira o projeto que vocês criaram hoje se conecta com o nosso dia a dia ou com outras áreas do conhecimento?
7. Se vocês tivessem mais tempo e componentes, que novas funções ou melhorias vocês adicionariam ao protótipo?
8. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de Robótica.

REFERÊNCIAS

ARDUINO. **Documentação de Referência da Linguagem Arduino**. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 27, mai. 2024.

BOSH. **Cidade Inteligente**. Disponível em: <https://www.bosch.com/stories/smart-city-challenges/>. Acesso em: 25 ago. 2025.

G20 BRASIL. **Cidades inteligentes, democráticas e sustentáveis: o legado do G20 no Brasil**. Disponível em: <https://www.gov.br/g20/en/news/smart-demo-cratic-and-sustainable-cities-the-legacy-of-the-g20-in-brasil#:~:text=Meanwhile%2C%20Brazilian%20cities%20such%20as,cities%20in%20the%20Global%20South>. Acesso em: 25 ago. 2025.

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

- Bruno Pereira Wesner da Silva - Engenharia de Computação
- Caetano de Medeiros Santana - Sistemas de Informação
- Fernanda das Neves Merqueades Santos - Ciência da Computação
- Filipe de Andrade Machado - Ciência da Computação
- Gabriel Pereira Falcão - Ciência da Computação
- Guilherme Siqueira Fiani - Engenharia de Software
- Jenniffer Oliveira Checchia - Ciência da Computação
- Maria Paula do Nascimento Santos - Engenharia de Computação
- Pedro Paulo de Oliveira Andrade - Ciência da Computação
- Vinicius Wagner da Silva - Engenharia de Software

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
- Enzo Enrico Giacomini Piolla
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Regeane Vaz Guedes
- 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/)



ALERTA

RISCO DE DESLIZAMENTO

DEFESA
CIVIL

199