

Desafios e oportunidades da nova era  
Manual para ensinar alunos mais velhos

# ARTIE



# ARTIE

## *Desafios e oportunidades da nova era: Manual para ensinar alunos mais velhos*

### Autores

Željko Krnjajić  
Janko Radigović

### Editor

Hrvatski robotički savez, Croatia  
“Artificial Intelligence in Education - challenges and opportunities of the new era: development of curriculum, guides for teachers and e-courses for students”, project number 2020-1-HR01-KA201-077800 under Erasmus+ Programme.

### Consultores/Revisores

Katarzyna Garbacik  
Andrzej Garbacik  
Bogusław Klimczuk  
Ivana Ružić  
Ana Pina  
Christina Eirini Karvouna

### Design gráfico e ilustrações

Christina Eirini Karvouna

### Tradutores

Jura Cmrečak (Croata)  
Bogusława Denys (Polonês)  
Ana Pina (Português)  
Christina Eirini Karvouna (Holandês)



3	Inteligência artificial em robótica
11	Deteccção e reconhecimento de rosto no Scratch
19	Programando deteccção de rosto no Scratch
26	Programando reconhecimento facial no Scratch
41	ODeteccção e classificação de objetos para iniciantes
49	Programação de deteccção de objetos no Scratch
57	Projeto com classificação de objetos
63	Reconhecimento e geração de fala para iniciantes no Scratch
68	Programação de reconhecimento de fala no Scratch
73	Programando geração de fala no Scratch
77	Objeto de projeto de controle de voz
83	Introdução ao hardware - microcontrolador, câmera e controlador do motor
93	Interface do microcontrolador com câmera e computador, início do trabalho
100	Montando o robô
119	Programação de movimentos do robô
128	Programação de robôs - rastreamento de objetos
140	Programando um robô - seguimento de linha



# Plano de aula 1

**Duração: 90 minutos**

## Tópicos

inteligência artificial em robótica, inteligência artificial, robótica

## Objetivos

familiarizar-se com a IA na robótica e compreendê-la

## Resultados

compreender como a IA é utilizada na robótica

## IA em robótica

Mostramos aos alunos o que é um robô com IA e onde é que eles já são utilizados

O professor anuncia o tema e inicia o debate:

Poderá a IA na robótica mudar o futuro?

O que é a inteligência artificial na robótica?

Já utilizamos robôs com IA na nossa vida quotidiana? Como? Onde?

### Apresentação do objetivo da aula:

Hoje vamos aprender o que é uma IA na robótica e onde podemos utilizá-la no dia a dia



<http://erasmus-artie.eu>



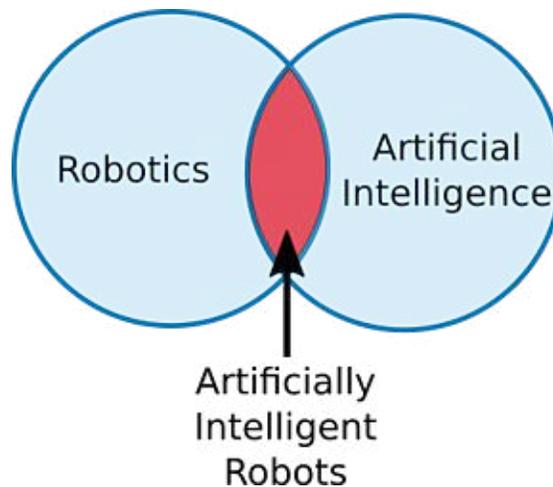
# PARTE PRINCIPAL

## Tópicos para debate:

- O que é um robô?
- Todos os robôs funcionam com IA?
- Conheces alguns exemplos de IA na robótica?
- Os robôs com IA podem substituir os humanos?

A primeira coisa a esclarecer é que a robótica e a inteligência artificial não são, de todo, a mesma coisa. De facto, os dois domínios são quase totalmente distintos.

Um diagrama de Venn dos dois domínios teria o seguinte aspeto:



Como se pode ver, há uma pequena área em que os dois domínios se sobrepõem: Inteligência Artificial e Robots. É nesta sobreposição que as pessoas por vezes confundem os dois conceitos.

O que é a Inteligência Artificial?

A inteligência artificial (IA) é um ramo da ciência da computação. Envolve o desenvolvimento de programas de computador para realizar tarefas que, de outra forma, exigiriam a inteligência humana. Os algoritmos de IA podem ser utilizados para a aprendizagem, a percepção, a resolução de problemas, a compreensão da linguagem e/ou o raciocínio lógico. De acordo com o pai da Inteligência Artificial, John McCarthy, é também "A ciência e a engenharia de fazer máquinas inteligentes, especialmente programas de computador inteligentes".



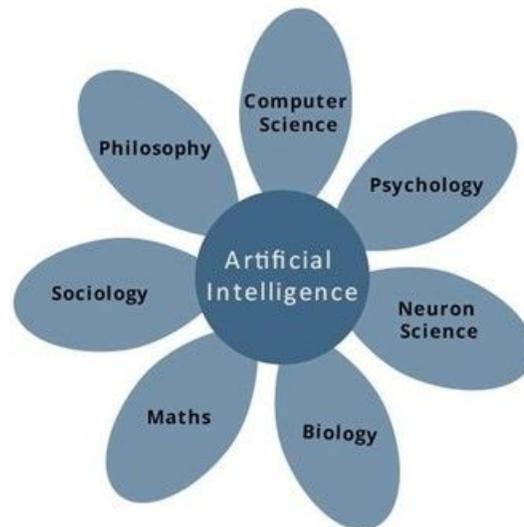
Assim, podemos dizer que a Inteligência Artificial é uma forma de fazer com que um computador, um robô controlado por computador ou um software pensem de forma inteligente, da mesma forma que os humanos inteligentes pensam.

### Objetivos da IA

- Criar Sistemas Especializados - Os sistemas que exibem um comportamento inteligente, aprendem, demonstram, explicam e aconselham os seus utilizadores.
- Implementar a inteligência humana nas máquinas - Criar sistemas que compreendam, pensem, aprendam e se comportem como humanos.

### O que contribui para a IA?

A inteligência artificial é uma ciência e uma tecnologia baseada em disciplinas como a informática, a biologia, a psicologia, a linguística, a matemática e a engenharia. Um dos principais objetivos da IA é o desenvolvimento de funções informáticas associadas à inteligência humana, como o raciocínio, a aprendizagem e a resolução de problemas. Dos seguintes domínios, um ou vários podem contribuir para a construção de um sistema inteligente.



### O que é a robótica?

A robótica é um ramo da IA, que é composto por Engenharia Elétrica, Engenharia Mecânica e Ciências da Computação para a conceção, construção e aplicação de robôs.

### O que são robôs?

Os robôs são agentes artificiais que actuam num ambiente do mundo real.

São máquinas programáveis que, normalmente, são capazes de realizar uma série de acções de forma autónoma ou semi-autónoma.



## Objetivo

Os robôs têm como objetivo manipular objectos, percebendo, apanhando, movendo, modificando as propriedades físicas de um objeto, destruindo-o ou produzindo um efeito, libertando assim a mão de obra de executar funções repetitivas sem se aborrecer, distrair ou esgotar.

## Aspectos da robótica

Os robôs têm uma construção mecânica, forma ou formato concebido para realizar uma determinada tarefa.

Têm componentes elétricos que alimentam e controlam a maquinaria.

Contêm algum nível de um programa informático que determina o quê, quando e como o robô faz alguma coisa.

## Inteligência artificial na robótica

A IA na robótica ajuda os robôs a realizar tarefas cruciais com uma visão semelhante à humana para detetar ou reconhecer vários objectos. Os robôs são desenvolvidos através de formação em aprendizagem automática e é utilizado um grande número de conjuntos de dados para treinar o modelo de visão por computador, de modo a que a robótica possa reconhecer os vários objectos e executar as acções em conformidade com os resultados correctos. A visão por computador é simplesmente o processo de perceção das imagens e dos vídeos disponíveis em formatos digitais. A IA na robótica não só ajuda a aprender o modelo para realizar determinadas tarefas, como também torna as máquinas mais inteligentes para atuar em diferentes cenários. Eis alguns exemplos das máquinas mais avançadas de Robôs humanóides, industriais e de serviço que estão a mudar o futuro com a ajuda da Inteligência Artificial.

## Sophia

O robô mais avançado da Hanson Robotics, Sophia, personifica os nossos sonhos para o futuro da IA. Como uma combinação única de ciência, engenharia e arte, Sophia é simultaneamente uma personagem de ficção científica criada por humanos que representa o futuro da IA e da robótica, é uma plataforma para a investigação avançada em robótica e IA

A personagem de Sophia capta a imaginação de audiências globais. É a primeira cidadã robô do mundo e a primeira embaixadora da inovação robótica do Programa das Nações Unidas para o Desenvolvimento. Sophia é agora um nome conhecido, com aparições no Tonight Show e no Good Morning Britain, para além de falar em centenas de conferências em todo o mundo.

Conheça a Sophia <https://www.youtube.com/watch?v=BhU9hOo5Cuc>





### Digit

O Digit foi concebido para ajudar a cuidar das pessoas nas suas casas, ajudar na resposta a catástrofes e entregar encomendas à porta de casa. Com os seus membros ágeis e um tronco repleto de sensores, o Digit pode navegar em ambientes complexos e realizar tarefas como a entrega de encomendas. Em maio de 2019, a Ford Motor Company e a Agility anunciaram uma parceria para desenvolver uma solução logística de última milha que combina a tecnologia de veículos autónomos da Ford e o Digit da Agility.



### Pepper

O Pepper é o primeiro robô humanoide social do mundo capaz de reconhecer rostos e emoções humanas básicas. O Pepper foi adotado por mais de 2000 empresas em todo o mundo. Perfeito para os sectores do retalho e das finanças, o Pepper tem inúmeras funcionalidades, incluindo o aumento do tráfego na loja, atraindo a atenção dos compradores, criando experiências memoráveis na loja, estimulando as compras e retendo os clientes. O Pepper também pode recolher dados abrangentes para enriquecer a base de clientes e gerar conhecimentos sobre os compradores.



<http://erasmus-artie.eu>

### Atlas

O Atlas é o robô humanoide mais dinâmico do mundo, construído pela BostonDynamics, uma empresa que foi anteriormente propriedade da Google e atualmente da SoftBank. O Atlas está a tornar-se mais sofisticado de ano para ano, graças ao seu hardware de última geração e ao algoritmo que lhe permite compreender rapidamente as instruções. Com as suas 28 articulações hidráulicas, 1,9 metros de altura e 176 libras de peso, o robô pode realizar actos impressionantes e aterradores, incluindo navegar em terrenos irregulares, saltar num percurso de parkour e dar cambalhotas. Todas estas actividades demonstram uma agilidade de nível humano, pelo que o robô pode ser perfeito para operações de busca e salvamento e para executar tarefas humanas em ambientes onde os humanos não poderiam sobreviver.





### Spot

O Spot é um cão-robô concebido para utilizações industriais, como o transporte de mercadorias num armazém e a inspeção de um local remoto com um ambiente desfavorável para os operadores humanos. Pode correr a 5,2 pés por segundo, tem câmaras de 360 graus e pode funcionar a temperaturas que variam entre 4 e 113 Fahrenheit. Com a sua API e interface de carga útil flexível, o robô pode ser facilmente personalizado para as tarefas desejadas. O Spot também é fabricado pela BostonDynamics e está agora a ser alugado a empresas elegíveis.



f

### HRP-5P

O HRP-5P é um robô humanoide avançado concebido para funcionar de forma autónoma e realizar trabalhos pesados em ambientes perigosos. Está equipado com sensores ambientais e reconhecimento de objectos, planeamento e controlo de movimentos de corpo inteiro e descrição de tarefas e gestão de execução. O HRP-5P baseia-se em mais de 20 anos de investigação de humanóides no AIST. Nesses 20 anos, o instituto criou 4 outros robôs que são os antecessores do HRP-5P.



<http://erasmus-artie.eu>

### Surena IV

O Surena IV é a quarta geração da série de robôs humanóides Surena, desenvolvida pela Universidade de Teerão, no Irão. Com uma altura de 1,5 m e um peso de 1,5 kg, este robô é capaz de andar a uma velocidade de 0,43 milhas por hora. Os seus sensores de força personalizados na parte inferior dos pés ajudam o robô a caminhar sobre superfícies irregulares, ajustando o ângulo e a posição de cada pé.





### Aquanaut

O Aquanaut é um avançado transformador subaquático não tripulado que pode transformar-se de um ágil submarino de longa distância num robô semi-humanoide capaz de executar tarefas complexas de manipulação subaquática. Concebido pela Houston Mechatronics Inc, o Aquanaut pode inspecionar infra-estruturas submarinas de petróleo e gás, operar válvulas e utilizar ferramentas submarinas com apenas alguns cliques do rato. Operando completamente sem amarras e sem navios de apoio, o Aquanaut pode viajar mais de 124 milhas em modo submarino, tem uma velocidade máxima de 7 nós e uma profundidade operacional máxima de 984 pés.



f

### Stuntronic robot

Um robô Stuntronic é um duplo animatrónico concebido para entreter as multidões nos parques temáticos e estâncias da Disney. Com os seus sofisticados sensores integrados, pode tomar as suas próprias decisões em tempo real - tudo isto enquanto voa a 60 pés no ar. Sabe quando deve dobrar os joelhos para dar uma cambalhota, quando deve puxar os braços para se torcer e até quando deve abrandar a rotação para garantir uma aterragem perfeita.



<http://erasmus-artie.eu>

### Handle

O Handle é outro robô da Boston Dynamics. Com o seu software de visão de aprendizagem profunda, este robô consegue identificar e localizar caixas, descarregar camiões, paletizar e despaletizar com o premir de um botão. A sua mobilidade permite-lhe operar em várias células de trabalho, deslocando-se pelas instalações juntamente com o fluxo de mercadorias. Pode recolher até 360 caixas/hora.





Mostre aos alunos um vídeo (opcional) e debata-o;  
<https://www.youtube.com/watch?v=Jky9I1ihAkg>

Atualmente, estamos a utilizar a IA na robótica, nos cuidados de saúde, na agricultura, na indústria automóvel, nos armazéns, nas cadeias de abastecimento, etc.

Mais tarde, construiremos o nosso próprio robô com IA e treiná-lo-emos para fazer a deteção e o reconhecimento de rostos, a deteção de objectos e o reconhecimento da fala.

ARTIEbot é um robô móvel com uma câmara e capacidades de IA desenvolvido exclusivamente para este projeto.

Vamos mostrar-lhe como o fazer e como o utilizar:

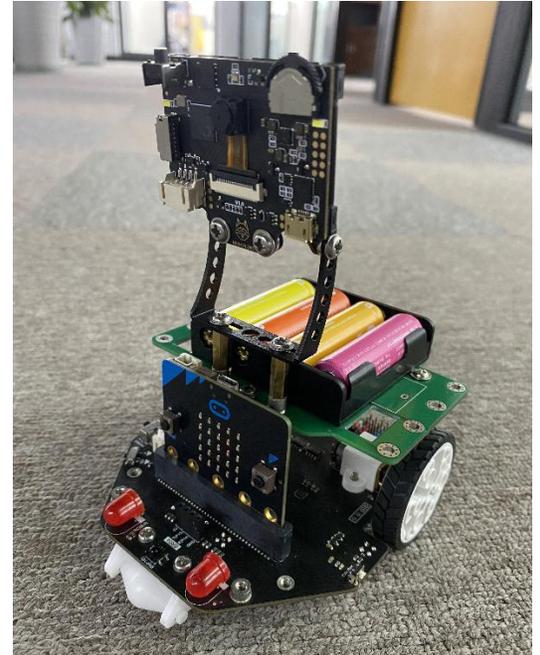
- Deteção de rostos
- Reconhecimento de rostos
- Seguimento do rosto
- Deteção de objectos
- Seguimento de objectos
- Seguimento de linhas

A Inteligência Artificial finalmente chegou e a maioria de nós já a utiliza ativamente no dia a dia (mesmo sem saber). As gerações futuras precisam, antes de mais nada, entender como usar a IA! Só então eles poderão usá-lo para facilitar o aprendizado e resolver problemas do mundo real.

A Inteligência Artificial (IA) e a Robótica estão fortemente conectadas hoje.

A IA na robótica é cada vez mais utilizada na vida quotidiana e tem sido fundamental em vários domínios, como indústrias, militar, medicina, exploração, entretenimento.

Lembre-se de que a IA é provavelmente a tecnologia mais poderosa já inventada pelo homem. Pode ser usado tanto para coisas boas quanto para coisas ruins. No final, cabe a nós como usá-lo.



## CONCLUSÃO

Atualmente, a Inteligência Artificial (IA) e a Robótica estão fortemente ligadas. A IA na robótica é cada vez mais utilizada na vida quotidiana e tem sido fundamental em vários domínios, como a indústria, o exército, a medicina, a exploração e o entretenimento.



## Plano de aula 2

**Duração: 90 minutos**

### Tópicos

Deteção e reconhecimento de rostos para principiantes em Scratch

### Objetivos

saber o que é a deteção e o reconhecimento de rostos para principiantes no Scratch

### Resultados

Saiba o que é a deteção e o reconhecimento de rostos através de exemplos simples no Scratch, Qual é a diferença entre a deteção e o reconhecimento de rostos

## Detecção e reconhecimento facial para iniciantes no Scratch

Os alunos devem ser solicitados a definir o que são reconhecimento facial e deteção facial. Dê-lhes a oportunidade de descrever a diferença.

Os termos deteção facial e reconhecimento facial às vezes são usados de forma intercambiável, mas existem algumas diferenças importantes. Para ajudar a esclarecer as coisas, vejamos o termo deteção facial e como ele difere do termo reconhecimento facial.

#### Apresentação do objetivo da aula:

Introdução à deteção e reconhecimento de rostos para principiantes através de exemplos de diferentes aplicações.



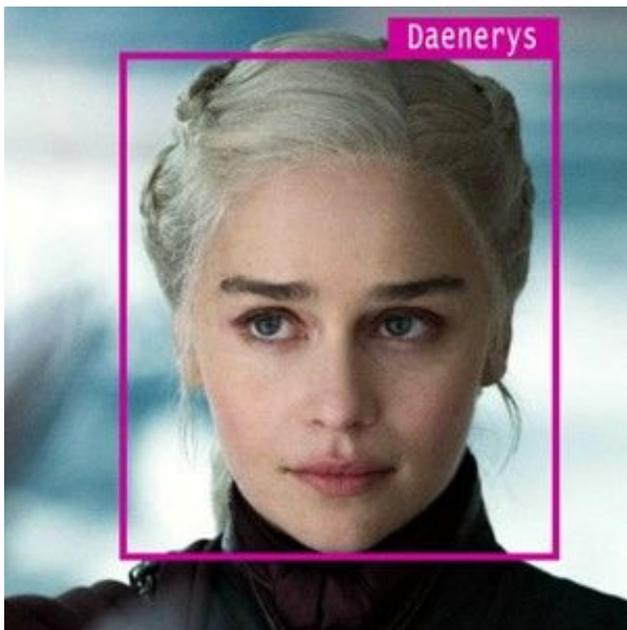
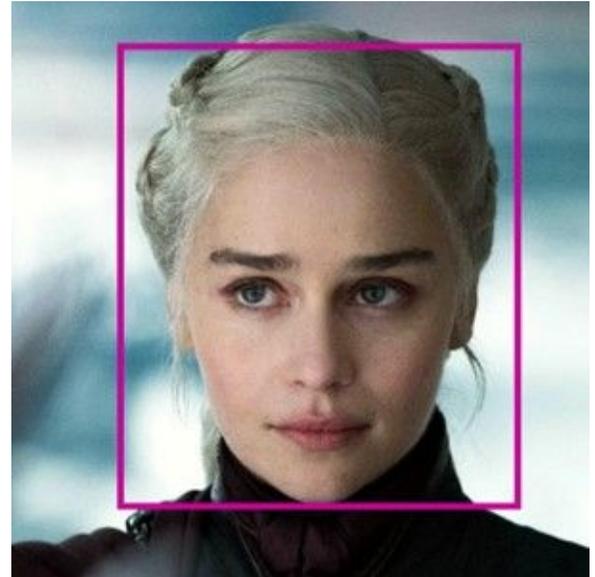
<http://erasmus-artie.eu>

11



# PARTE PRINCIPAL

A deteção de rosto é um processo que determina a presença de rosto(s) numa imagem fixa ou num clip de vídeo. Por exemplo, esta função está disponível na maioria dos softwares de câmaras de smartphones. Mas o módulo de deteção de rostos não determina qual o rosto que está na imagem.



A Deteção de Rosto não memoriza ou guarda características faciais. Se o software detetar um rosto de uma determinada pessoa na imagem e mais tarde encontrar o mesmo rosto noutra imagem, não determinará que o rosto pertence à mesma pessoa, apenas detectará a presença de um rosto nas imagens. O software pode fornecer dados sobre a idade e o sexo de uma pessoa em cada fotograma, mas não mais do que isso. O software de deteção de rostos não pode reconhecer pessoas específicas. Pelo contrário, o reconhecimento facial está relacionado com a identificação e o reconhecimento de pessoas. O objetivo do reconhecimento facial baseado em software é efetuar a identificação de pessoas que aparecem numa imagem fixa ou num clip de vídeo, comparando-os com uma base de dados. Para garantir uma identificação bem sucedida, os rostos correspondentes devem ser previamente introduzidos na base de dados.



### Scratch/Scratch baseados e outros aplicativos para usar

#### Scratch (ML4KIDS)

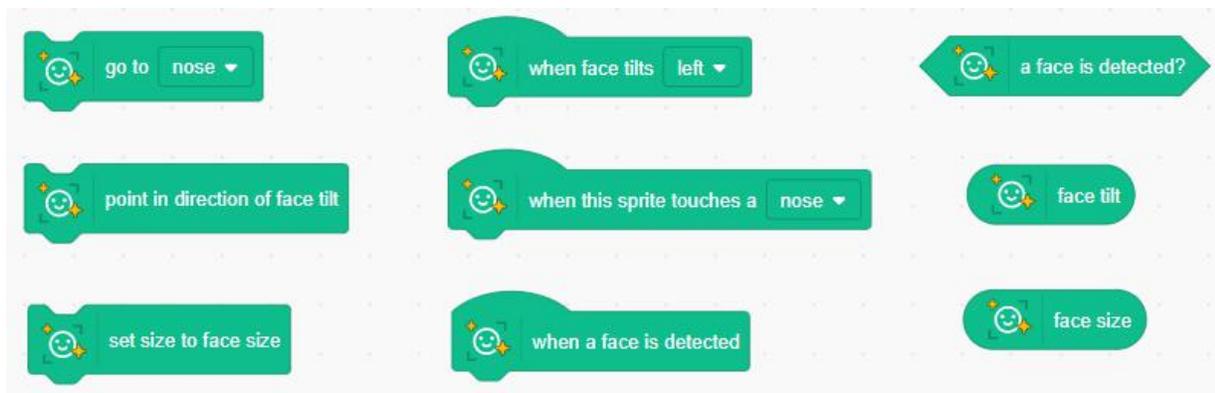
<https://machinelearningforkids.co.uk/scratch3/>

Está disponível uma extensão de detecção de faces com 3 blocos do tipo repórter. Utilize-a em combinação com a extensão de detecção de vídeo para ligar/desligar o vídeo da câmara e definir a transparência.



#### Scratch (MIT) - <https://lab.scratch.mit.edu/face/>

Clique em "Try it out" (Experimentar) e terá 9 blocos para o manuseamento do reconhecimento facial.



<http://e.erasmus-artie.eu>



**Scratch (MITMEDIALAB)** - <https://mitmedialab.github.io/prg-extension-boilerplate/create/>  
 Carregue a extensão Face sensing e terá 9 blocos para o tratamento da detecção de rostos. Alguns blocos são utilizados para o reconhecimento de expressões faciais e sentimentos. Também pode utilizar a extensão Teachable machine em combinação com a máquina Google Teachable.



**Makeblock (mBlock)** - <https://ide.mblock.cc/>

Carregue os serviços cognitivos e as extensões de detecção de vídeo e terá uma variedade de blocos. Não existe um bloco específico para a detecção de rostos, mas é possível detetar uma pessoa dentro de um bloco de reconhecimento. Mas há muitos blocos disponíveis para lidar com emoções, idade, género, sorriso, cor do cabelo, óculos e até uma situação em que se cobre uma parte do rosto.

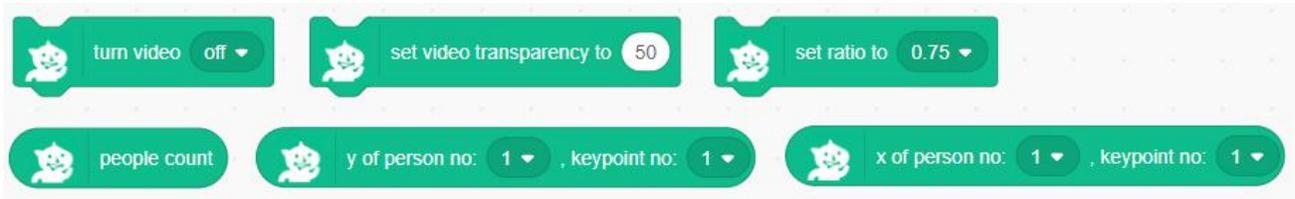
O Makeblock também fornece a extensão Teachable machine (que não está relacionada com o Google), onde pode treinar até 3 classes e utilizá-la para reconhecimento facial ou detecção de objetos.



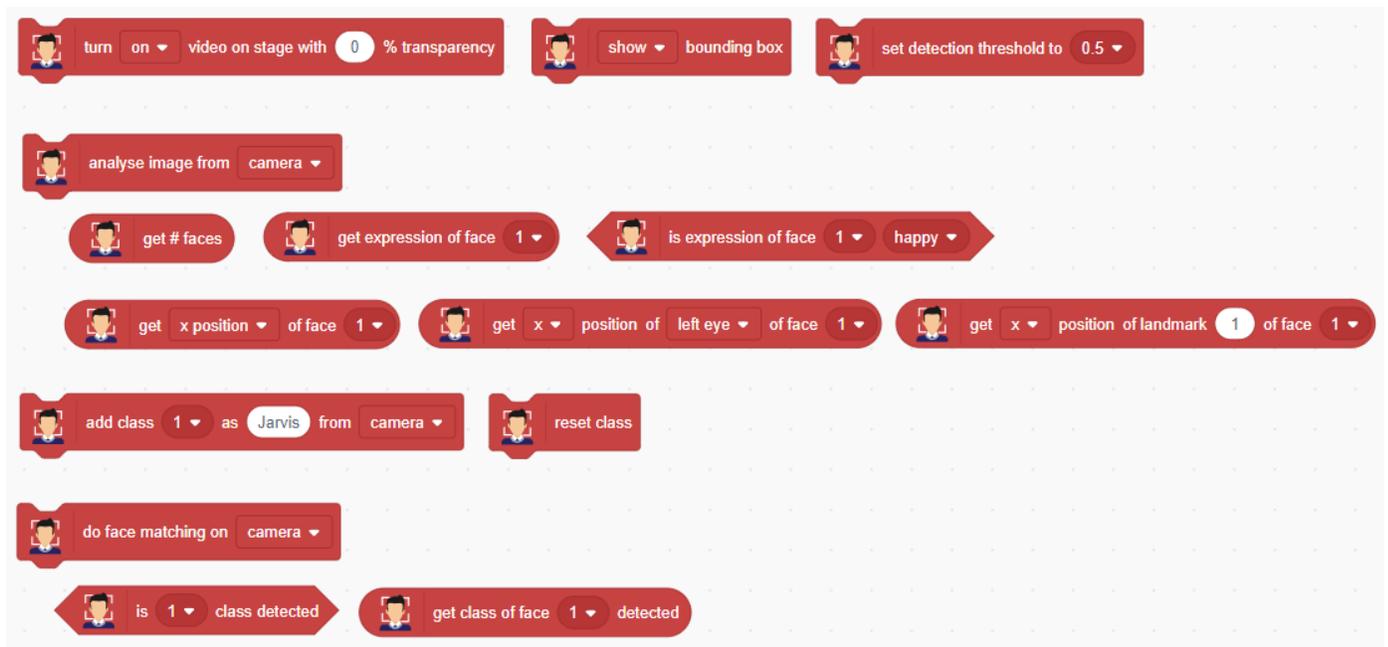


### Stretch3 (github.io) - <https://stretch3.github.io/>

Carregue a extensão Facemesh2Scratch para utilizar 3 blocos para a detecção de rostos (existem 3 blocos adicionais para o tratamento de vídeo). A principal característica é a capacidade de detecção múltipla de rostos e pode detetar mais do que uma pessoa no fluxo da sua câmara.



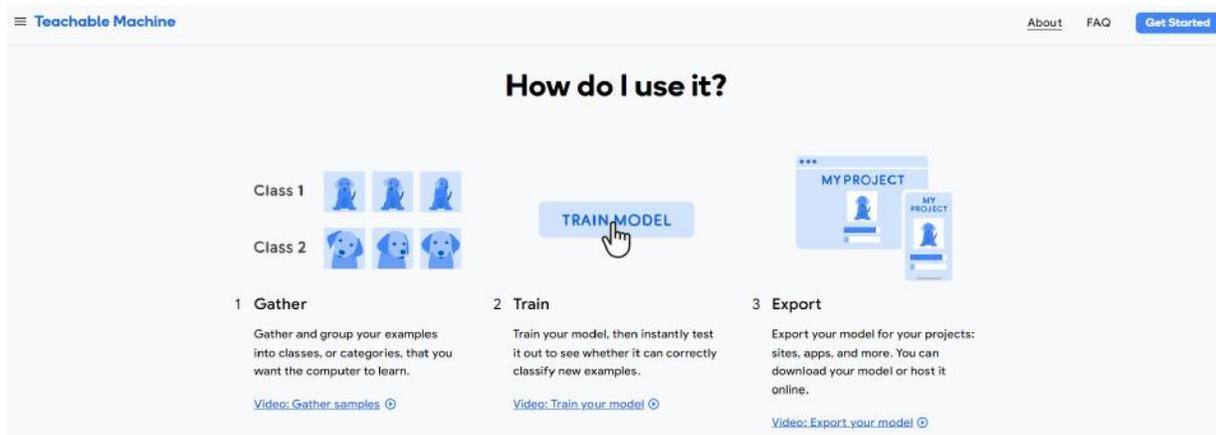
A última aplicação a utilizar é a PictoBlox, uma aplicação do tipo desktop, que deve ser instalada primeiro a partir de <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb) Após a instalação, utilize a extensão Face Detection e verá um verdadeiro tesouro - pode detetar vários rostos e as suas expressões faciais. Além disso, existe uma funcionalidade para treinar classes que nos conduzem diretamente ao reconhecimento facial.





**Teachable machine (Google)** - <https://teachablemachine.withgoogle.com/>

Esta aplicação é utilizada para treinar o seu modelo e utilizá-lo para o reconhecimento facial em combinação com a extensão da máquina Teachable disponível no Scratch (MITMEDIALAB)

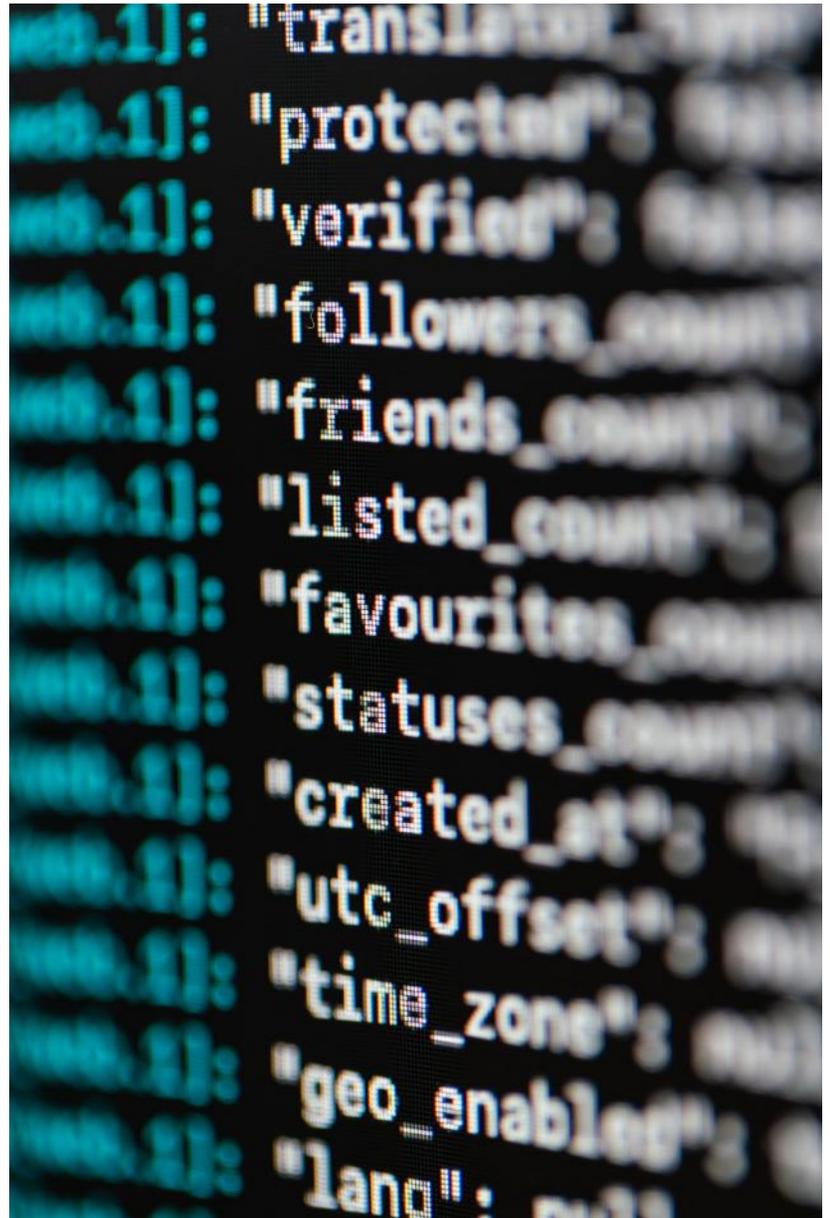


Vimos três aplicações diferentes, mas também uma muito semelhante para detecção e reconhecimento facial. A Detecção de Rosto difere do Reconhecimento Facial (os termos não devem ser usados de forma intercambiável) porque a Detecção de Rosto envolve apenas a detecção de um rosto dentro de uma imagem ou vídeo digital. Significa simplesmente que o sistema de detecção de rosto pode identificar que existe um rosto humano presente numa imagem ou vídeo – não pode identificar a pessoa. A detecção facial é um componente dos sistemas de reconhecimento facial – o primeiro estágio do reconhecimento facial é detectar a presença de um rosto humano em primeiro lugar. A detecção de rosto também pode ser usada em câmeras para ajudar no foco automático – você provavelmente já percebeu que em algumas câmeras digitais e telefones, uma pequena caixa aparecerá ao redor dos rostos das pessoas detectadas na imagem, permitindo que a câmera priorize o foco nessas pessoas. A identificação da presença de um rosto humano é feita por meio de fórmulas e algoritmos.

Normalmente, a primeira coisa que um sistema de detecção de rosto procura são os olhos, pois são uma das características mais fáceis de identificar. Em seguida, também pode procurar a presença de boca, sobrancelhas, nariz e narinas. A detecção facial é uma parte importante do processo de reconhecimento facial, no entanto, do ponto de vista da segurança, não há nenhum benefício independente de um sistema de detecção facial – ele simplesmente reconhece a presença de um rosto, mas não tem ideia sobre a identidade desse rosto. O reconhecimento facial está a desempenhar um papel vital numa vasta gama de indústrias, especialmente no controlo de fronteiras e na aplicação da lei. A identificação precisa de indivíduos está ajudando a melhorar a segurança nos aeroportos e em vilas e cidades em todo o mundo, e isso só pode ser feito usando sistemas de reconhecimento facial líderes de mercado.



Como podemos ver, a tecnologia de reconhecimento facial está a mudar o mundo em que vivemos e parece que estamos apenas a arranhar a superfície em termos de aplicações potenciais de software de reconhecimento facial. Embora as utilizações do reconhecimento facial possam parecer intermináveis, precisamos de garantir que esta tecnologia é utilizada de forma adequada e responsável.



<http://erasmus-artie.eu>

## CONCLUSÃO

A deteção de rosto é um processo que determina a presença de um rosto em uma imagem ou videoclípe enquanto o reconhecimento facial se refere à identificação e reconhecimento de pessoas.



## Plano de aula 3

**Duração: 90 minutos**

### Tópicos

Programando detecção de rosto no Scratch

### Objetivos

Para aprender a programar a detecção facial com exemplos carregados

### Resultados

Saber como escrever um programa para detecção de rosto usando Scratch

## Programando detecção de rosto no Scratch

Como programar a detecção de rosto no Scratch?

Revisaremos o que aprendemos sobre detecção de rosto na última aula.

Pergunte a seus alunos sobre suas experiências com detecção de rosto.

Antes de conduzi-los, pergunte se eles sabem como fazer um programa de detecção de rosto.

O professor apresenta aos alunos a programação da detecção de rosto no Scratch.

Vamos examinar alguns exemplos simples de programação de detecção de rosto em aplicativos baseados em Scratch e Scratch.

Você provavelmente já percebeu que alguns aplicativos do seu smartphone desenharam um retângulo ao redor do rosto como resultado da detecção de rosto. Também é possível fazer isso no Scratch.

#### Anúncio do objetivo da aula:

Compreender o software de detecção facial e seu uso examinando exemplos específicos.



<http://erasmus-artie.eu>

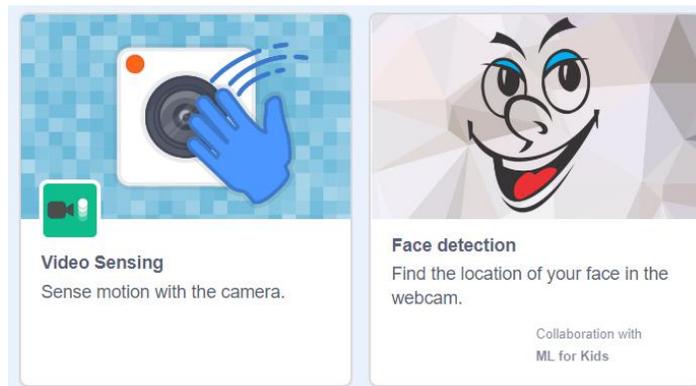


# PARTE PRINCIPAL

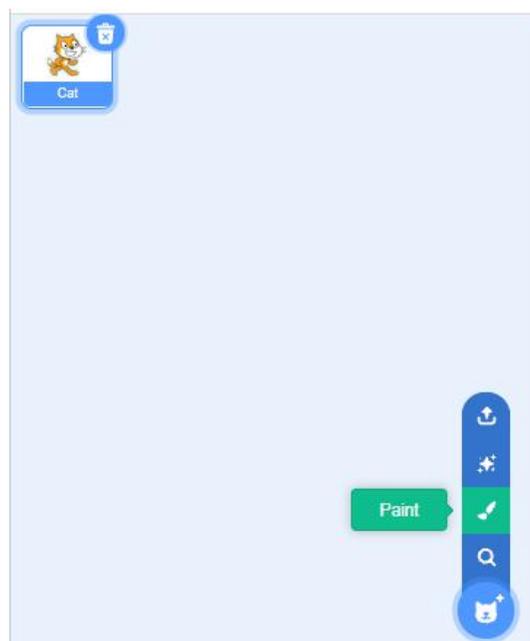
## Primeiro projeto- SCRATCH (ML4KIDS):

Etapa 1: abra seu navegador Chrome e vá para:  
<https://machinelearningforkids.co.uk/scratch3/>

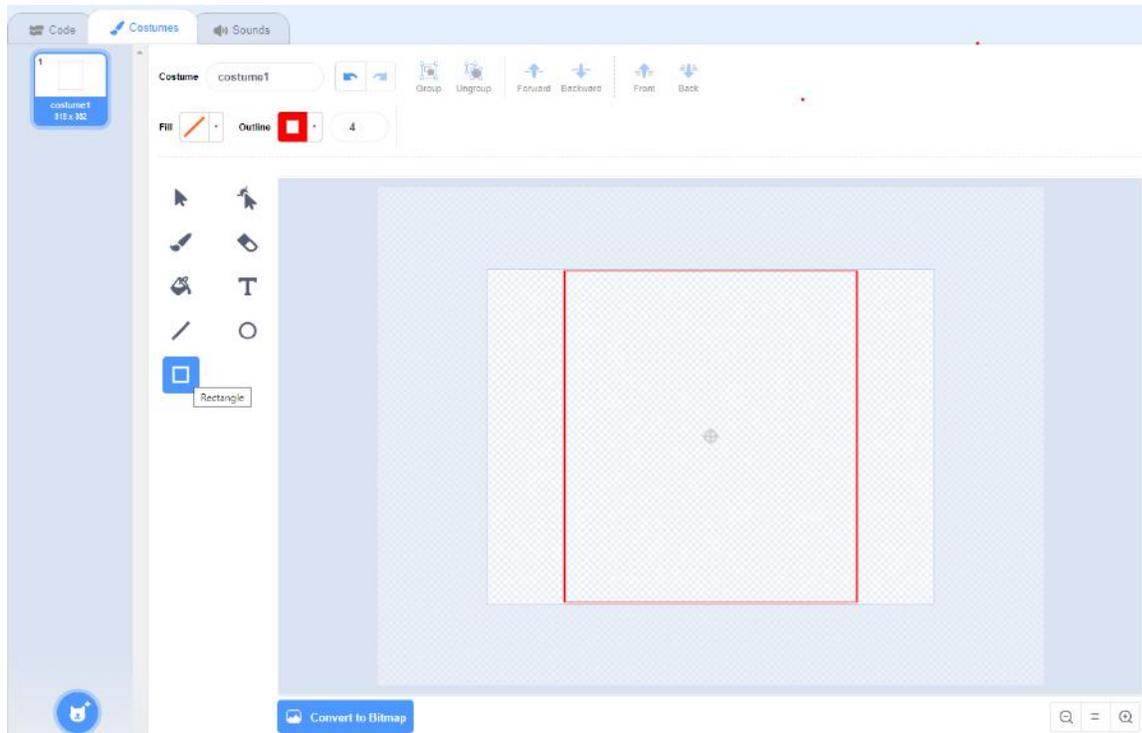
Etapa 2: carregue extensões de detecção de vídeo e detecção de rosto e conecte sua webcam (se você não tiver uma integrada)



Etapa 3: Exclua o sprite Cat clicando em ícone da lixeira (canto superior direito) e escolha a opção de pintar um novo sprite

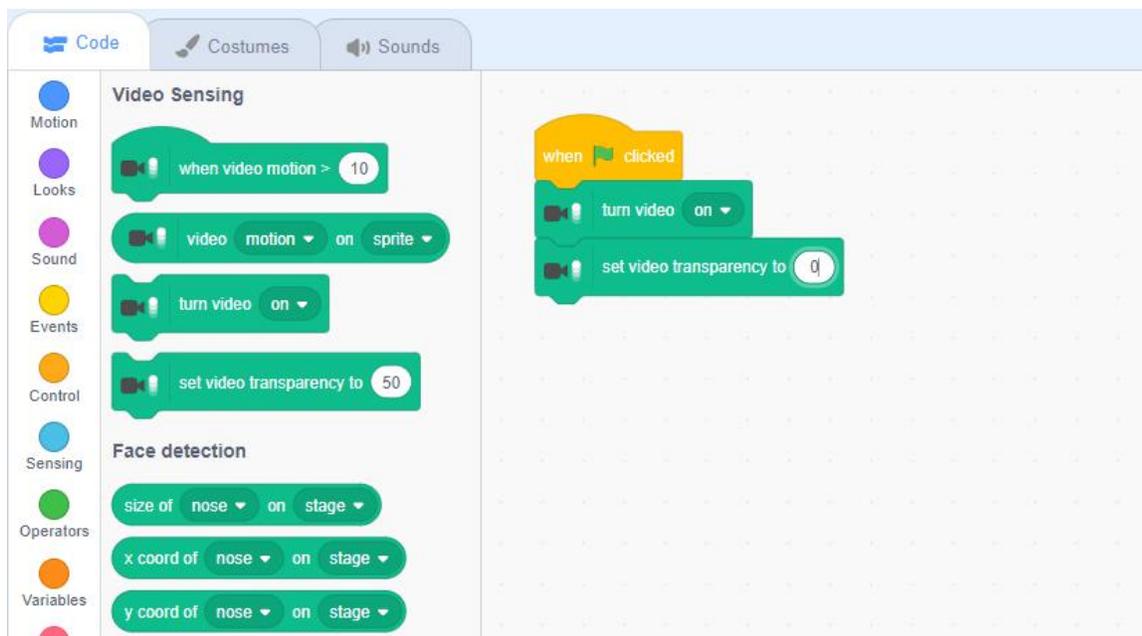


Etapa 4: Desenhe um retângulo (será usado como caixa delimitadora) sem preenchimento e defina o contorno para vermelho (4) como na imagem abaixo



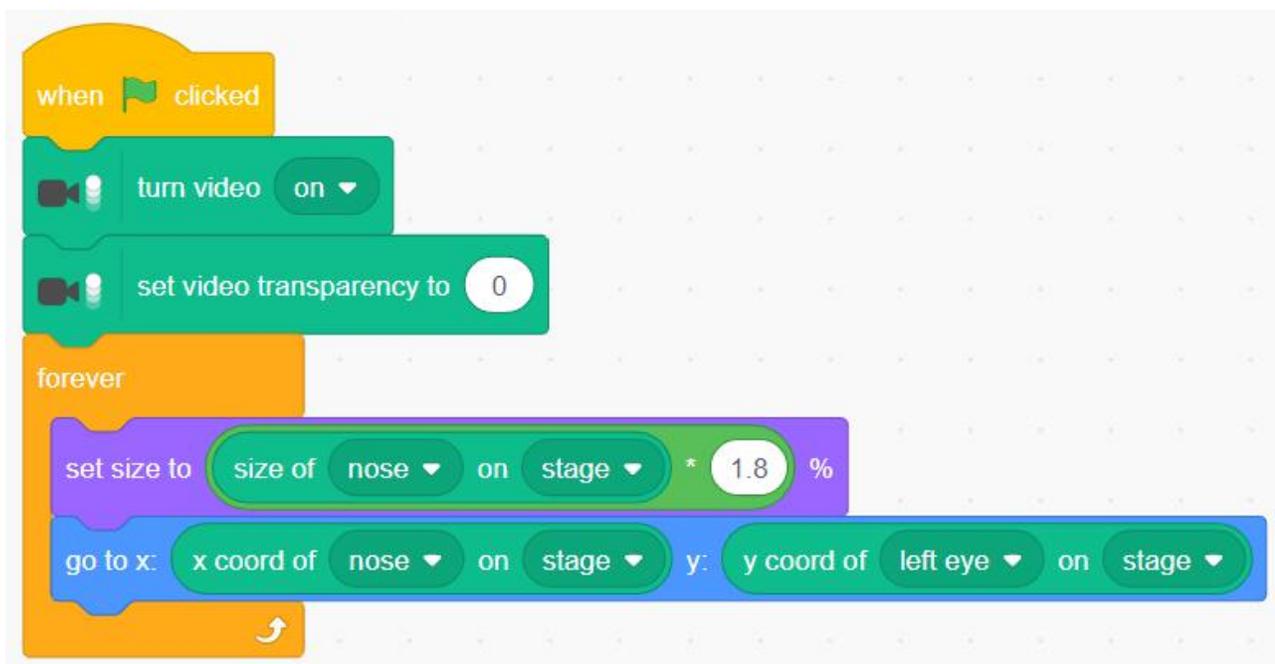
f

Etapa 5: Mude para a guia Código e comece a programar. Primeiro ligamos o vídeo e definimos a transparência como 0 (não transparente).



<http://erasmus-artie.eu>

21



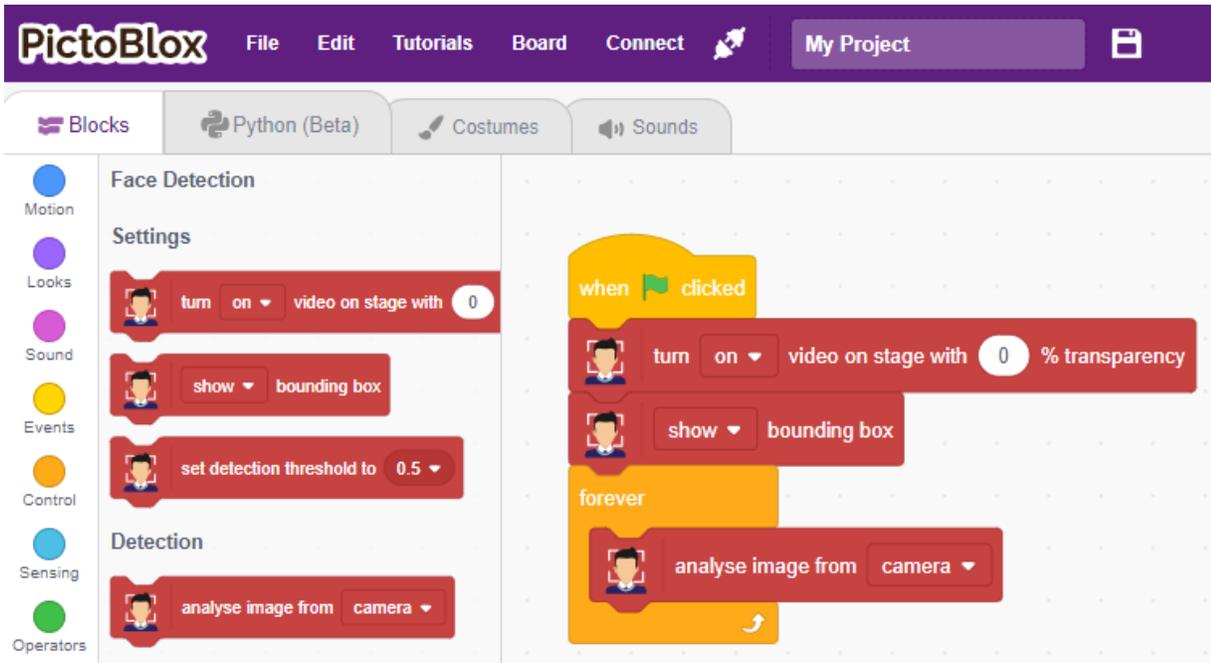
### Terceiro projeto com PICTOBLOX (aplicativo Desktop):

Etapa 1a: Como não há GUI on-line disponível, você deve instalar o PictoBlox em: <https://thestempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

**f** Etapa 2a: Abra o PictoBlox e escolha a expansão de detecção de rosto



Passo 3a: Use blocos como na imagem abaixo e é muito simples descobrir como funciona. Agora temos uma caixa delimitadora como bloco e não há necessidade de desenhar um retângulo. Mas a principal característica é que ele pode detectar vários rostos. Chame alguém para se juntar a você na frente da câmera para ver como funciona. Verifique o bloco de tipo de repórter get # faces para ver quantos rostos são detectados.

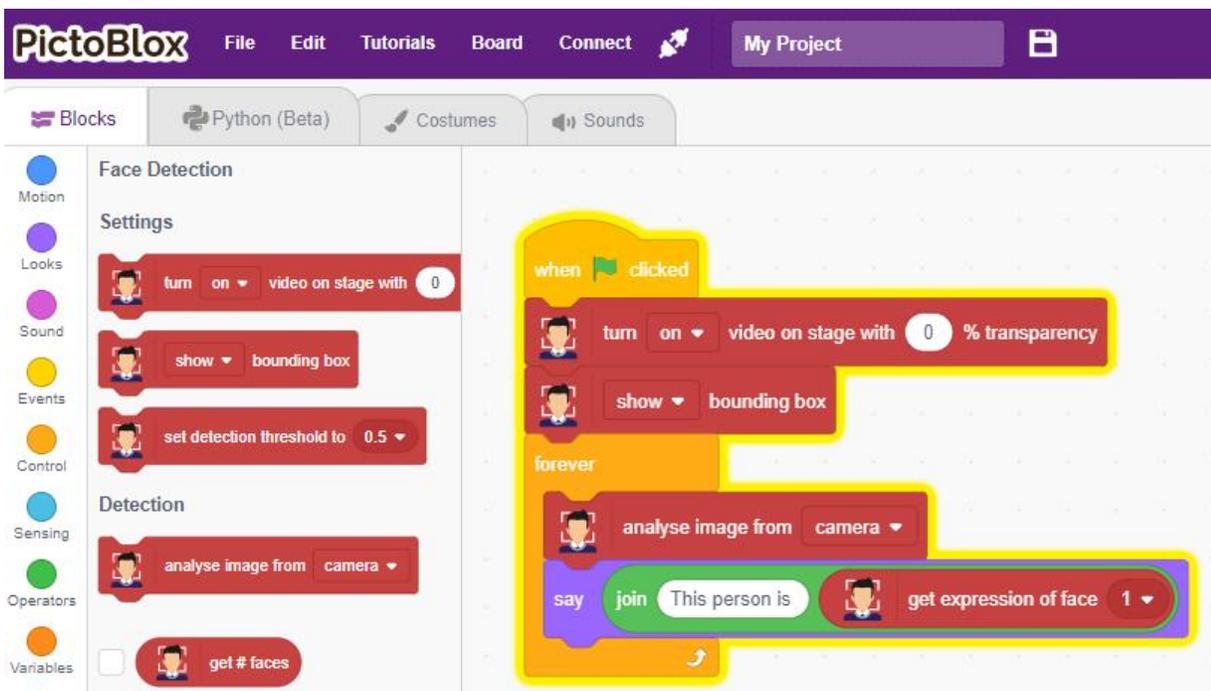


Etapa 4a: E vamos incrementar isso usando o operador join para exibir a expressão facial de uma pessoa. Você pode explorar ainda mais como funciona com múltiplas faces.



<http://erasmus-artie.eu>

24





Hoje em dia, o software de detecção facial é usado em quase todos os campos, desde dispositivos móveis para filtros faciais de bate-papo até vários aplicativos de segurança. A detecção de rosto ajuda a reconhecer rostos, idade, expressões, gênero, localização e muitos outros recursos. Detecção de rosto é um termo mais amplo dado a qualquer sistema que possa identificar a presença de um rosto humano em uma imagem visual. A detecção de rosto tem inúmeras aplicações, inclusive na contagem de pessoas, marketing on-line e até mesmo no foco automático da lente de uma câmera. Seu objetivo principal é sinalizar a presença de um rosto. O reconhecimento facial tornou-se mais significativo e relevante nos últimos anos devido às suas potenciais aplicações. Uma vez que os rostos são altamente dinâmicos e apresentam mais questões e desafios a resolver, investigadores no domínio do reconhecimento de padrões, visão computacional e inteligência artificial propuseram muitas soluções para reduzir tais dificuldades, de modo a melhorar a robustez e a precisão do reconhecimento.

Hoje, a detecção facial é usada em:

Aplicativos do mundo real (Amazon Rekognition: os recursos incluem verificação de usuários, contagem de pessoas e moderação de conteúdo, frequentemente usados por empresas de mídia, empresas de análise de mercado, sites de comércio eletrônico e soluções de crédito, BioID: solução compatível com GDPR usada para evitar fraudes on-line e roubo de identidade, Cognitec: reconhece rostos em transmissões de vídeo ao vivo, com clientes que vão desde a aplicação da lei até o controle de fronteiras, FaceFirst: uma solução de segurança que visa usar DigitalID para substituir cartões e senhas, Trueface.ai: serviços que vão até a detecção de armas e são utilizados por vários sectores incluindo educação e segurança...)

Diagnósticos médicos

Captura criminosa

Vigilância e conformidade

## CONCLUSÃO

A detecção facial é usada em vários sistemas sofisticados, e é possível criar exemplos simples em ferramentas como Scratch e PictoBlox.





## Plano de aula 4

**Duração: 90 minutos**

### Tópicos

Programando reconhecimento facial no Scratch  
Interpretando resultados do algoritmo de reconhecimento facial

### Objetivos

Uso prático do reconhecimento facial

### Resultados

Saber como escrever um programa para reconhecimento facial usando Scratch,  
Explorando as possibilidades de extensão de reconhecimento facial no Scratch

## Programando reconhecimento facial no Scratch

O objetivo desta lição é aprender como programar o reconhecimento facial no Scratch?

Tópicos para discussão:

O que é reconhecimento facial?

Como funciona o reconhecimento facial?

Que aplicações práticas pode ter?

Converse com seus alunos sobre o que aprenderam sobre reconhecimento facial.

Eles sabem alguma coisa sobre como criar um programa de reconhecimento facial antes de você liderá-los?

O professor apresenta aos alunos a programação de reconhecimento facial no Scratch e os instrui sobre como fazer upload de imagens e depois usá-las em programas de reconhecimento.

#### Anúncio do objetivo da aula:

Através de exemplos de um programa, você compreenderá melhor os programas de reconhecimento facial e seu uso.



# PARTE PRINCIPAL

Passo 1: Abra o seu navegador Web e descarregue as 20 imagens de: <https://bit.ly/daenerys-data>

Serão utilizadas para treinar a Classe 1

Passo 2: Abra o seu navegador Web e descarregue as 20 imagens de: <https://bit.ly/arya-data>

Será utilizada para treinar a turma 2

Passo 3: Abra o seu navegador Web e acesse a: <https://teachablemachine.withgoogle.com/>

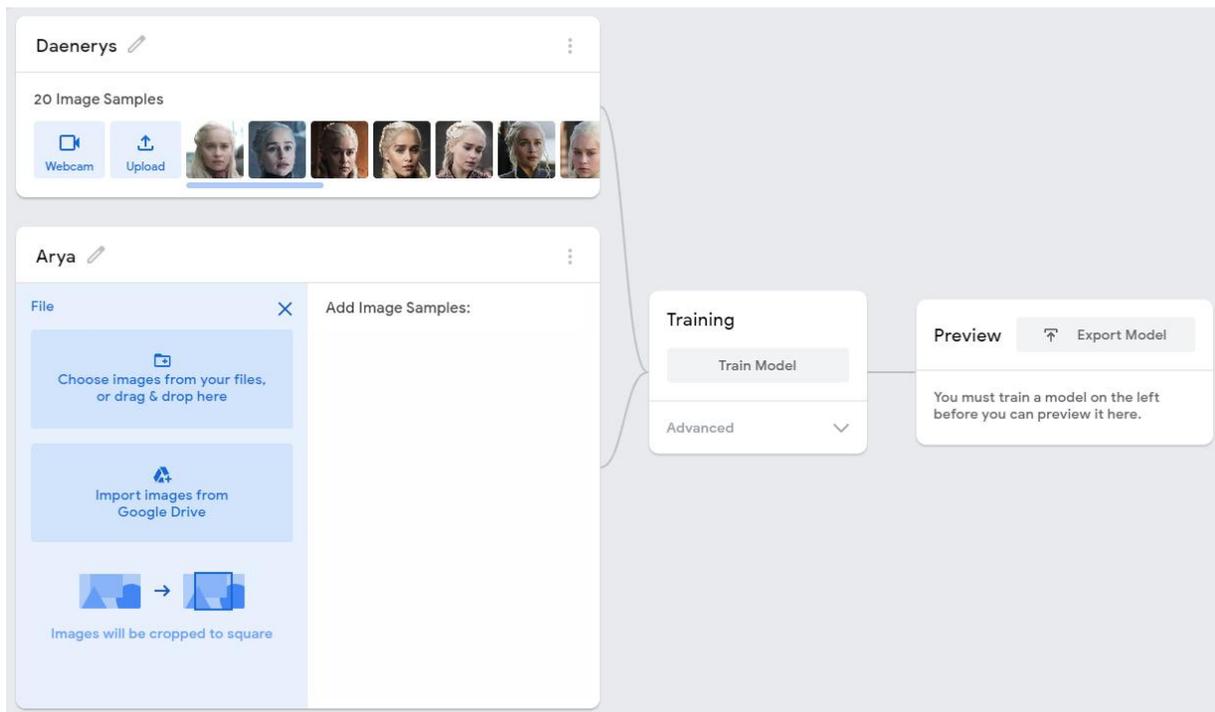
Passo 4: Clique em Get started (Começar)

Passo 5: Escolha o projeto de imagem

Passo 6: Seleccione o modelo de imagem Standard

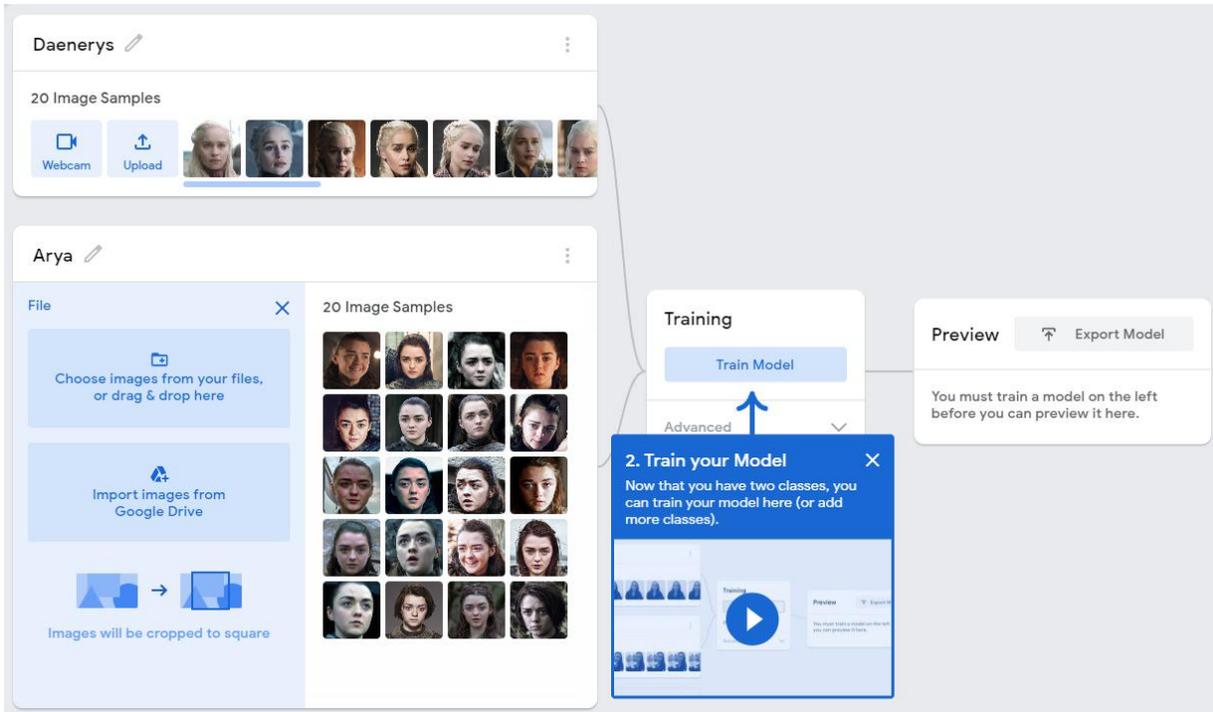
Passo 7: Altere o nome da Classe 1 para Daenerys e o da Classe 2 para Arya. Carregue as imagens de Daenerys para os ficheiros Daenerys e as imagens de Arya para os ficheiros Arya,

f como mostra a figura abaixo





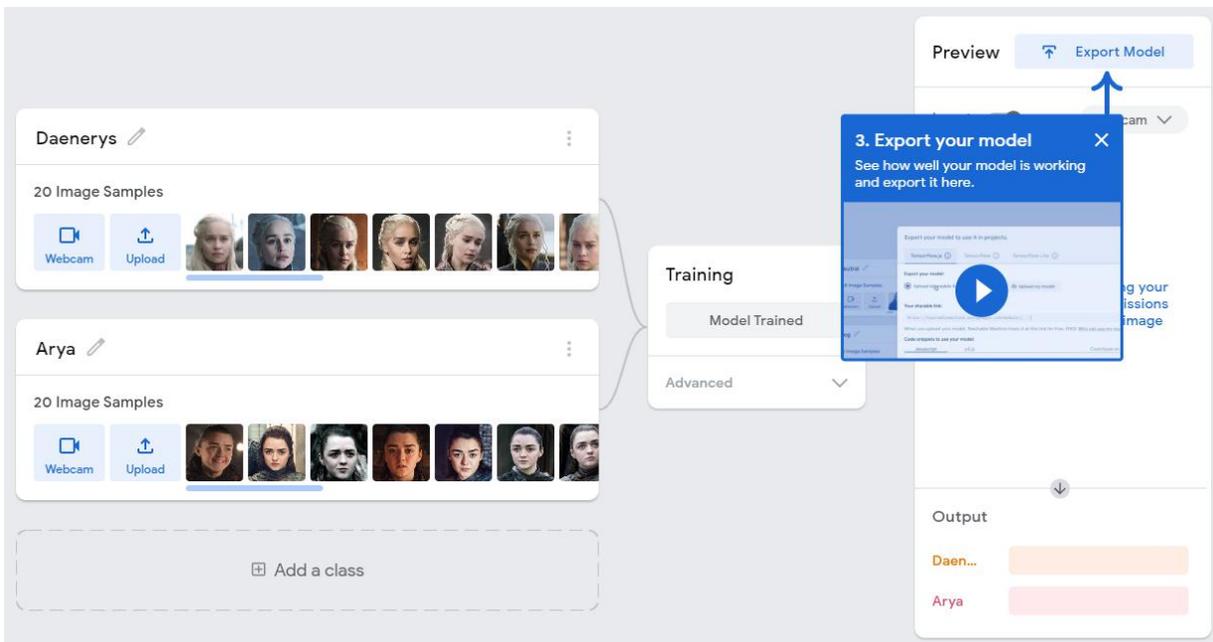
Passo 8: Treine o seu modelo. Não altere os separadores do browser durante o processo de treino.



Passo 9: Exportar o modelo. Na janela pop-up, seleccione carregar para a nuvem (terceira opção) e o Google alojará os seus dados gratuitamente.



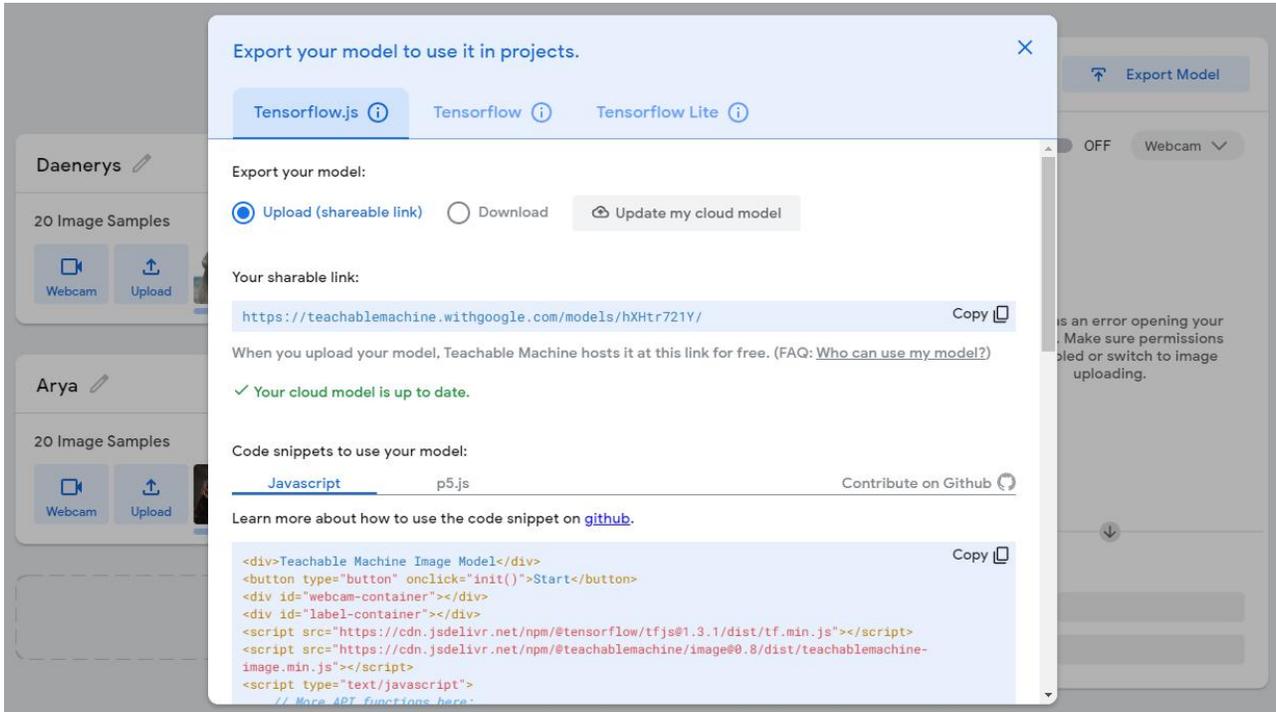
<http://erasmus-artie.eu>





Passo 10: Copie a hiperligação indicada no campo de texto abaixo - este é o URL do seu modelo.

Neste caso, foi <https://teachablemachine.withgoogle.com/models/hXHtr721Y/>



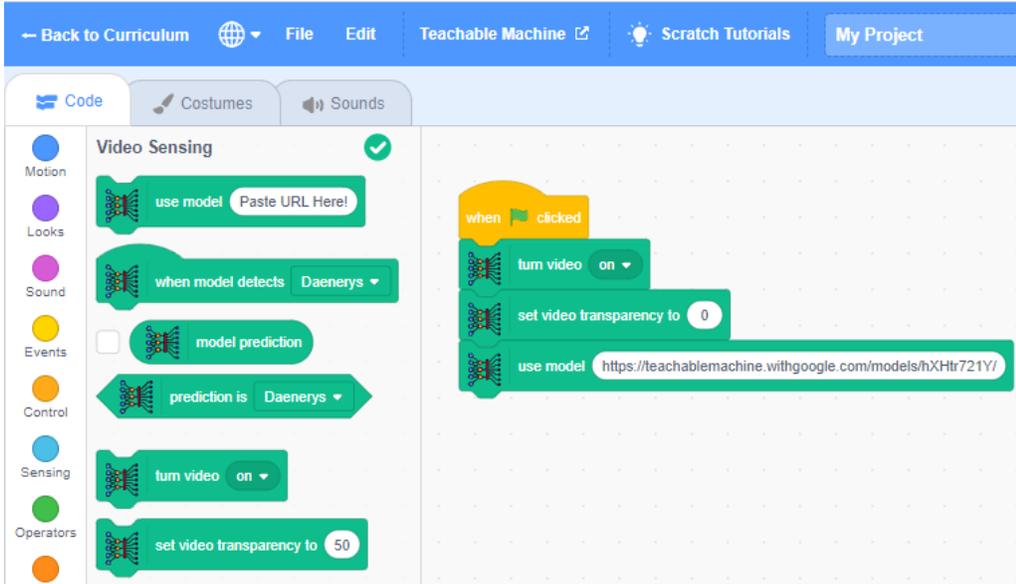
Passo 11: O seu modelo está pronto a ser utilizado

Passo 12: Abra a GUI do Scratch em: <https://mitmedialab.github.io/prg-extension-boilerplate/create/> e carregar a extensão Teachable Machine.



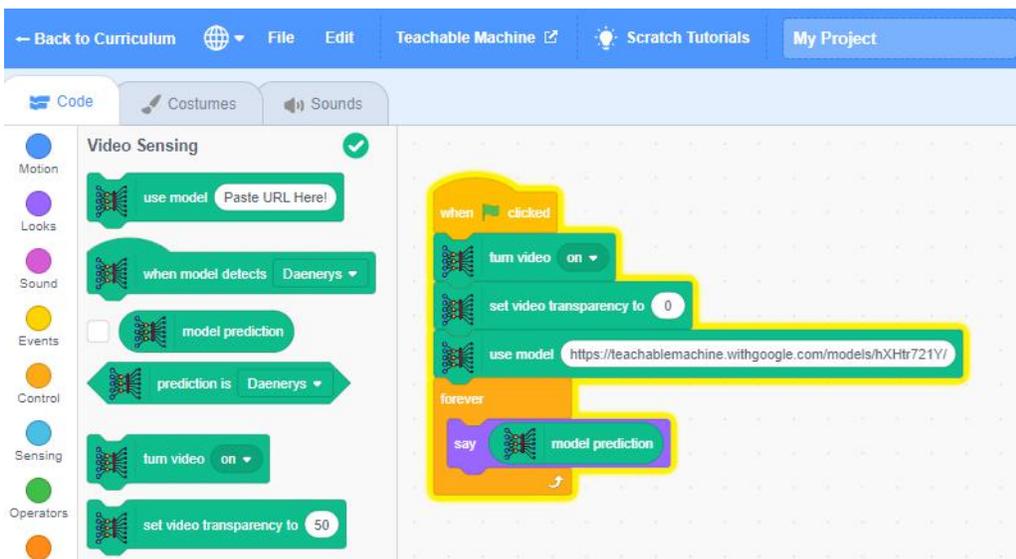


Passo 13: Em primeiro lugar, temos de ativar o vídeo da câmara e definir a transparência para 0 (não transparente). A seguir, utilizamos o bloco URL do modelo e colamos aí a nossa hiperligação para o modelo.



Passo 14: Os últimos blocos são um loop (para sempre) e um bloco que contém um bloco do tipo repórter com o resultado da previsão. Tenho quase a certeza de que não vai conseguir colocar a verdadeira Daenerys ou Arya em frente à sua câmara, por isso use o seu smartphone com as imagens delas e aponte-o para a câmara para ver os resultados. Podes treinar o modelo com as tuas próprias imagens ou com as imagens dos teus amigos.

\*\*\* Certifica-te de que não tiras a fotografia de ninguém sem a sua autorização.



http://erasmus-artie.eu



## PICTOBLOX (aplicação de ambiente de trabalho):

Passo 1a: Uma vez que não existe uma interface gráfica em linha, é necessário instalar o PictoBlox a partir de: <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

Passo 2a: Abrir o PictoBlox e selecionar a expansão da deteção de rostos



Etapa 3a: Abra o seu navegador Web e descarregue as 20 imagens de: <https://bit.ly/daenerys-data>

Estas serão utilizadas para treinar a Classe 1

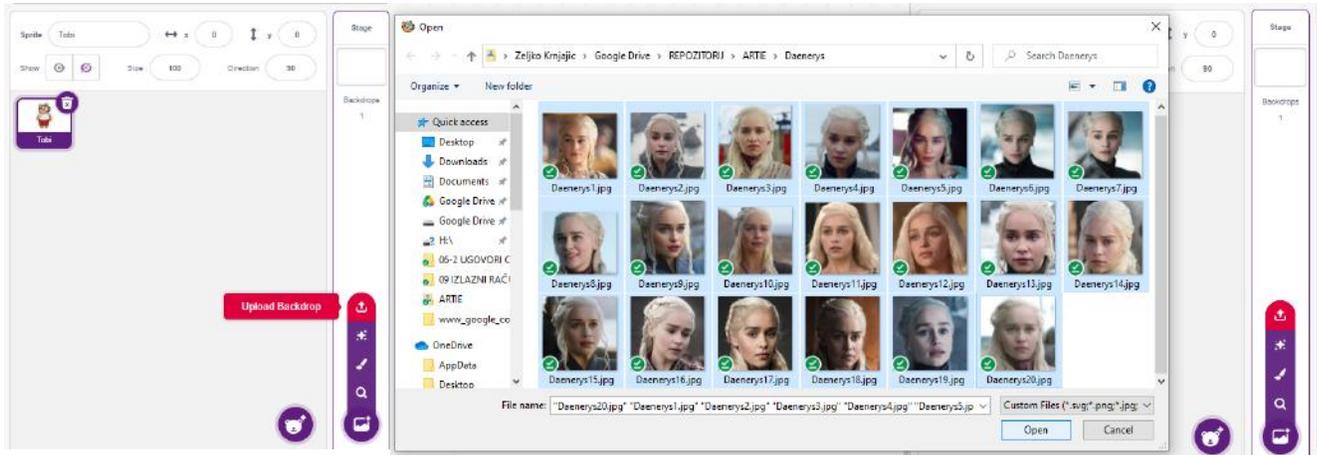
Passo 4a: Abra o seu navegador Web e descarregue as 20 imagens de: <https://bit.ly/arya-data>  
Será utilizada para treinar a turma 2

Etapa 5a: Esconder o ator Tobí do palco, clicando no ícone como mostra a figura abaixo.

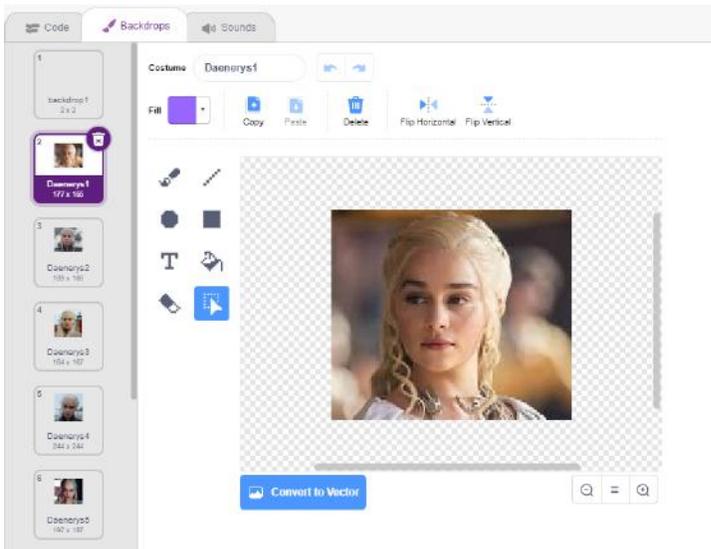




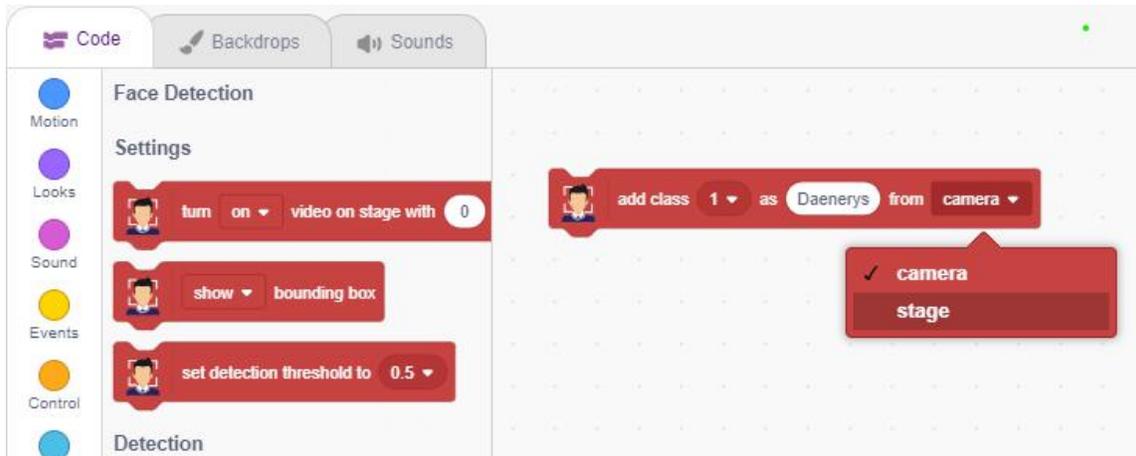
Passo 6a: Carregar todas as imagens de Daenerys para o pano de fundo (Carregar pano de fundo - Selecionar todas as imagens - Abrir)



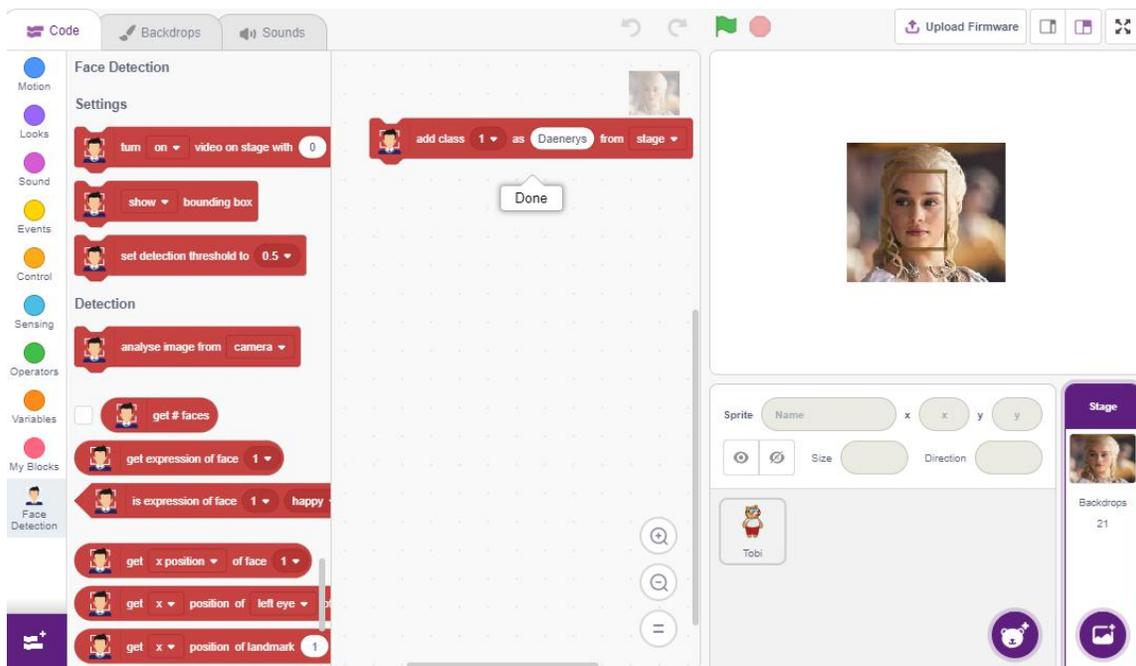
Passo 7a: Seleccione a imagem Daenerys1 e mude para o separador Code (Código)



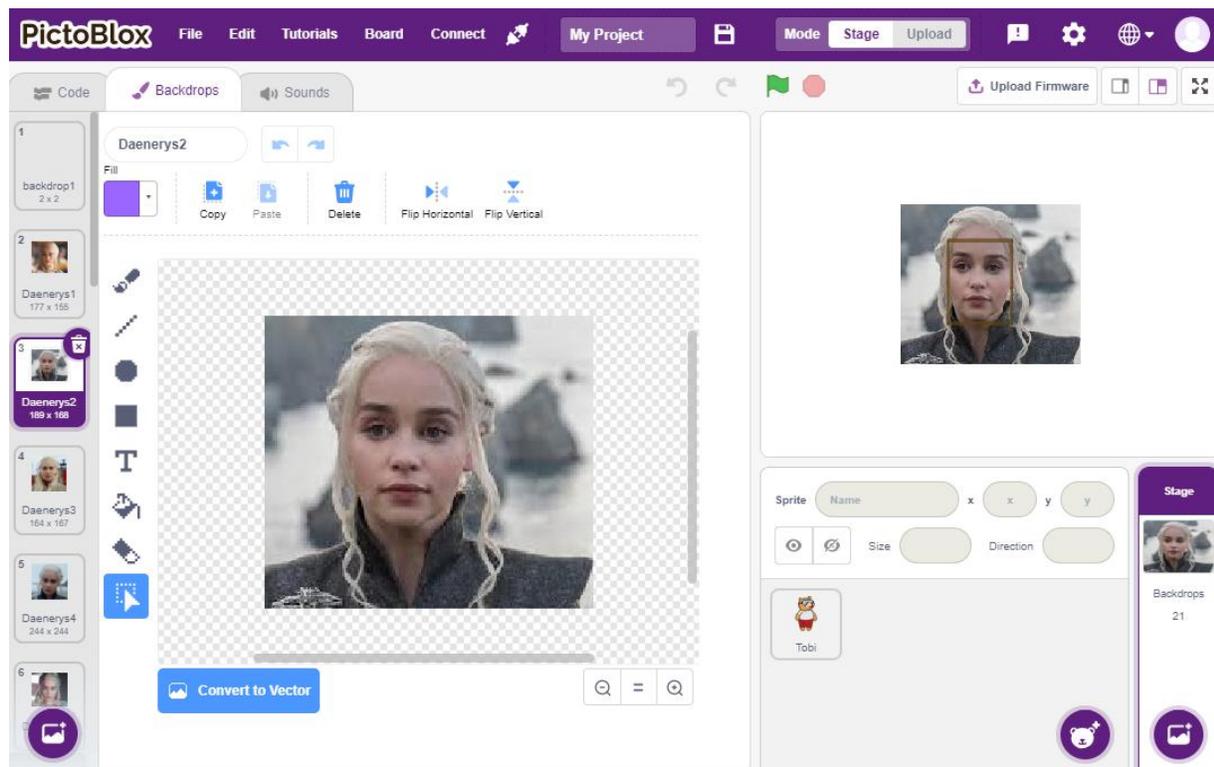
Passo 8a: No grupo Face Detection (Deteção de faces), procure o bloco Add Class (Adicionar classe), arraste-o e largue-o na área de código e mude o nome da classe de Jarvis para Daenerys e mude a fonte da câmara para o palco, como mostra a figura abaixo.



Etapa 9a: Depois de fazer as alterações, basta clicar no bloco Adicionar aula para começar a treinar. Receberá uma mensagem de "Concluído" e verá a caixa delimitadora no rosto da Daenerys. O treino com a 1ª imagem está concluído e terá de repetir estes passos para todas as outras imagens.



Passo 10a: Volte ao separador Backdrops e selecione a imagem Daenerys 2.

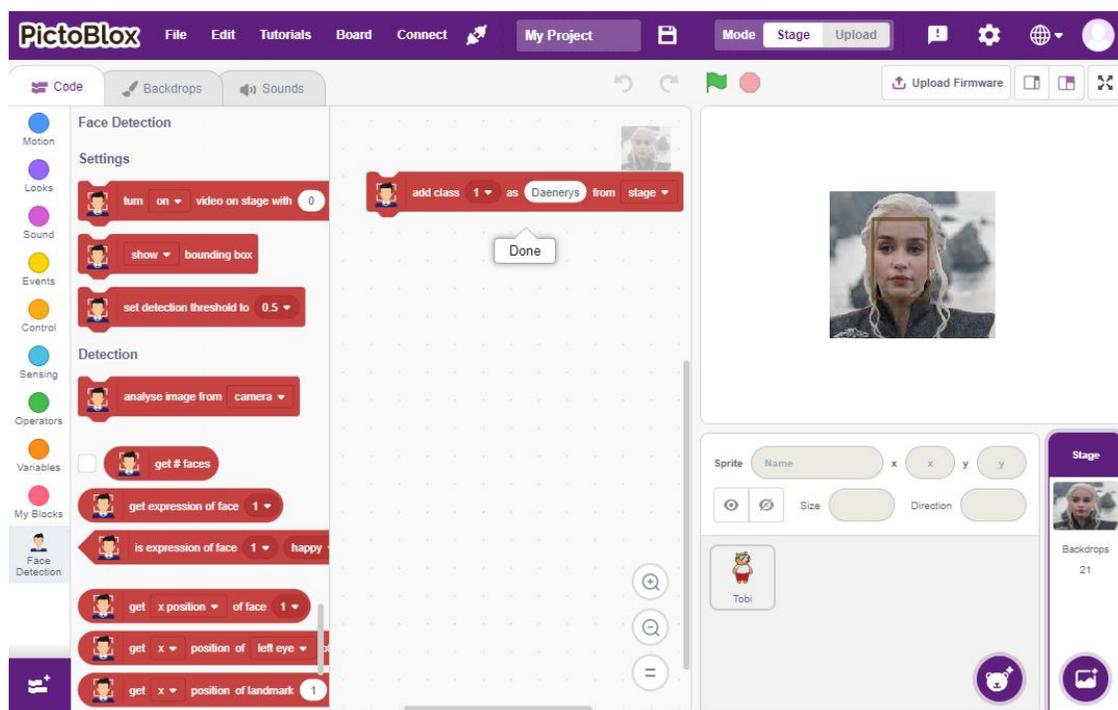


Passo 11a: Voltar ao separador Code (Código) e clicar novamente em add class block (adicionar bloco de classe)

f

<http://erasmus-artie.eu>

34

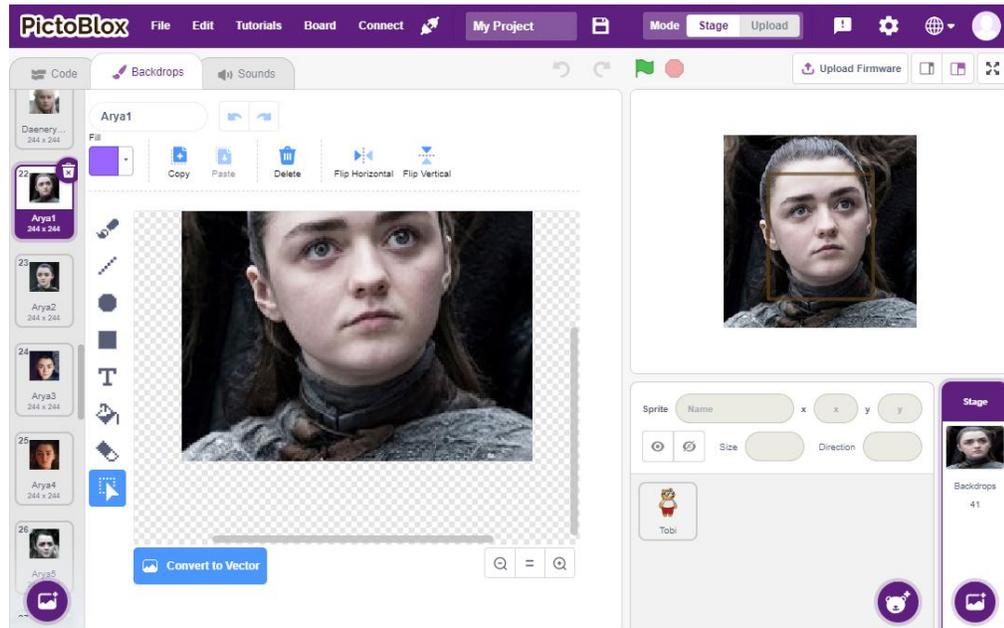




Passo 12a: Repetir os passos 10a e 11a para cada uma das imagens (até Daenerys20)

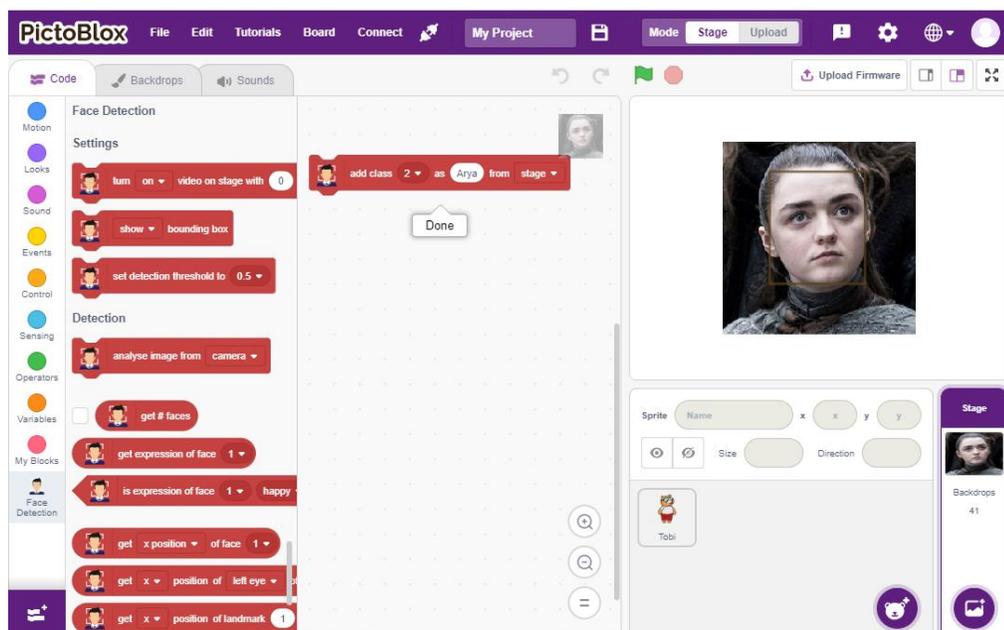
Passo 13a: Agora, carregue todas as imagens de Arya para o backdrop da mesma forma que fez no Passo 6a

Passo 14a: Seleccione a imagem Arya1



f

Passo 15a: Volte ao separador Código e altere (isto é importante porque estamos a formar a 2ª classe) o nome da classe de adição de Daenerys para Arya. A fonte continua a ser a mesma - do palco. Depois, cliquem nesse bloco para treinar a primeira imagem da classe Arya.





Passo 16a: Mudar para o separador Backdrops e seleccionar a imagem Arya2

Passo 17a: Mudar para o separador Code (Código) e clicar em add class block (adicionar bloco de classe)

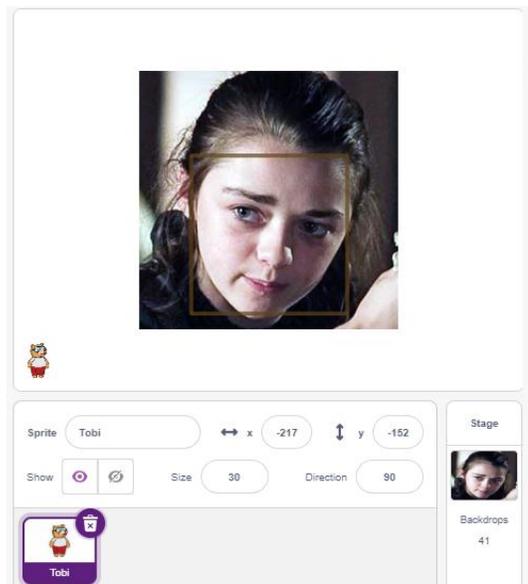
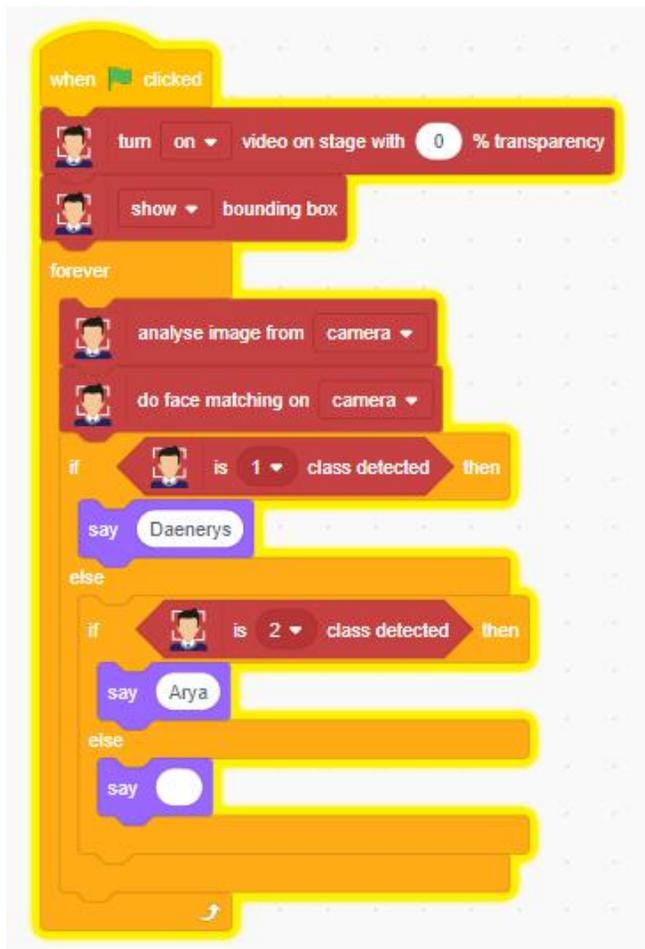
Passo 18a: Repita os passos 16a e 17a para cada uma das imagens (até Arya20)

Etapa 19a: O modelo está agora pronto para ser testado. Seleccionar o ator Tobi e torná-lo visível (mostrar). Definir o tamanho do Tobi para 30% e deslocá-lo do centro para o canto.

Passo 20a: Ligue a sua câmara (se não tiver uma) e comece a codificar. Ligar o vídeo com 0% de transparência e mostrar a caixa delimitadora. O bloco seguinte é um loop infinito e o Tobi mostra o resultado do reconhecimento facial com base no resultado do reconhecimento facial. É um bloco duplo if-else e o último else case devolve uma cadeia vazia se não for reconhecida nenhuma cara.

Etapa 21a: Iniciar o programa. Utilize o seu smartphone com imagens da Daenerys ou da Arya e aponte-o para a câmara para ver os resultados. Pode treinar o modelo com as suas próprias imagens ou com imagens dos seus amigos.

\*\*\* Certifica-te de que não tiras a fotografia de ninguém sem a sua autorização.





## PROJETO COM RECONHECIMENTO FACIAL

Aqui está um pequeno projeto que é basicamente um caso de utilização do reconhecimento facial. Preparámos um modelo treinado de 8 celebridades femininas para que a Teachable Machine o compare com o seu rosto. Temos Adriana Lima, Emilia Clarke, Gal Gadot, Natalie Portman, Selena Gomez, Emma Stone, Zoe Saldana, Maisie Williams. Terás de descobrir qual o rosto da lista que tem mais semelhanças contigo.

Passo 1b: Abra o Scratch GUI em:

<https://mitmedialab.github.io/prg-extension-boilerplate/create/> e carregar a extensão Teachable Machine.



Passo 2b: Primeiro, definimos o tamanho e a posição do sprite e, em seguida, activamos o vídeo da câmara e definimos a transparência para 0 (não transparente). A seguir, utilizamos o bloco URL do modelo e colamos esta hiperligação do modelo: <https://teachablemachine.withgoogle.com/models/smuBDQTuY/>





Passo 3b: A seguir, vamos utilizar um bloco de tipo de evento que é acionado por correspondência de rosto. E basta acrescentar-lhe um bloco "You look like...".

```
when clicked
  set size to 50 %
  go to x: -200 y: -120
  turn video on
  set video transparency to 0
  use model https://teachablemachine.withgoogle.com/models/smuBDQTuY/

when model detects Adriana
  say You look like Adriana Lima
```

Passo 4b: Adicione esta combinação de blocos para cada celebridade da lista.

```
when clicked
  set size to 50 %
  go to x: -200 y: -120
  turn video on
  set video transparency to 0
  use model https://teachablemachine.withgoogle.com/models/smuBDQTuY/

when model detects Adriana
  say You look like Adriana Lima

when model detects Emilia
  say You look like Emilia Clarke

when model detects Gal
  say You look like Gal Gadot

when model detects Natalie
  say You look like Natalie Portman

when model detects Selena
  say You look like Selena Gomez

when model detects Emma
  say You look like Emma Stone

when model detects Zoe
  say You look like Zoe Saldana

when model detects Maisie
  say You look like Maisie Williams
```





Passo 5b: Ligue/ligue a sua câmara Web, inicie o código e veja quem é o seu par!

Passo 6b: Quer alterar o modelo atual com o seu próprio modelo? Existe um ótimo conjunto de dados de imagens de celebridades em: <https://www.kaggle.com/hereisburak/pins-face-recognition>

Poderá ser-lhe pedido que se registe no Kaggle antes de o poder descarregar.

Ou pode recolher imagens manualmente através da pesquisa no Google. Depois de recolher todas as imagens que pretende - utilize a máquina Teachable em:

<https://teachablemachine.withgoogle.com/> para treinar o seu modelo da mesma forma que fez no cenário Programar o reconhecimento facial no Scratch.

O reconhecimento facial é uma tecnologia capaz de identificar ou verificar um sujeito através de uma imagem, vídeo ou qualquer elemento audiovisual do seu rosto. Geralmente, esta identificação é utilizada para aceder a uma aplicação, sistema ou serviço. Antes de começarmos a programar o reconhecimento facial, temos de recolher as fotografias de um rosto específico e treinar o modelo. Utilizámos uma aplicação chamada Teachable Machine. A Teachable Machine é uma ferramenta baseada na Web que torna a criação de modelos de aprendizagem automática rápida, fácil e acessível a todos.

Reparou que este projeto é totalmente baseado na Web e que não é necessária a instalação de software?

Atualmente, muitos consumidores em todo o mundo interagem regularmente com a tecnologia de reconhecimento facial.

### Principais funcionalidades do reconhecimento facial:

Verificação de identidade:

Identificar indivíduos e aplicar regras específicas com base na categoria em que se inserem, por exemplo, VIP, visitante registado, lista de bloqueio, empregado ou estudante. Utilizar as informações para melhorar e automatizar processos como (1) controlo de acesso, (2) proteção de segurança, (3) saudações a clientes ou visitantes e (4) relógios de ponto de funcionários.

eKYC e prevenção de spoofing:

Validar a identidade de uma pessoa utilizando uma fotografia ou captura de vídeo em direto com uma identificação digitalizada (e verificada). A isto chama-se eKYC (electronic Know Your Customer) e é amplamente utilizado em BFSI ou casos semelhantes.

Autorização:

Identificar se um indivíduo está numa base de dados pré-autorizada para (1) levantar dinheiro de um ATM, (2) aceder a um gabinete médico que contenha medicamentos seguros, ou (3) desbloquear maquinaria dispendiosa que exija operadores bem treinados.

Segmentação e análise de clientes:

Para publicidade inteligente, analisar as características de uma pessoa que se encontra em frente a um sinal digital, como o género, a idade e a emoção.





### Medidas de saúde:

Confirmar se uma pessoa está a usar corretamente uma máscara, bem como verificar se não tem febre antes de conceder acesso a um edifício ou restaurante.

Quando olhamos para os mercados verticais, 10 indústrias destacam-se como estando prontas para integrar o reconhecimento facial e, em muitos casos, já o estão a adotar:

1. Fabrico e armazenamento
2. Banca, serviços financeiros e seguros (BFSI)
3. Escritórios inteligentes
4. Casas e complexos residenciais inteligentes
5. Retalho
6. Transportes públicos e aeroportos
7. Instalações de cuidados de saúde
8. Escolas e universidades
9. Hotelaria
10. Restaurantes e bares

## CONCLUSÃO

O reconhecimento facial é usado em vários sistemas sofisticados, e é possível criar exemplos simples em ferramentas como Scratch e PictoBlox.





## Plano de aula 5

**Duração: 90 minutos**

### Tópicos

---

Deteção e classificação de objectos para principiantes no Scratch

### Objetivos

---

Aprender o que é a deteção e classificação de objetos para principiantes no Scratch

### Resultados

---

Saiba o que é a deteção e a classificação de objetos através de um exemplo simples no Scratch, Qual é a diferença entre deteção de objectos e classificação de objetos

## Detecção e classificação de objetos para iniciantes no Scratch

A deteção e classificação de objetos têm atraído muita atenção nas últimas décadas.

No campo da visão computacional, uma das dúvidas mais comuns tem a ver com a diferença entre classificação de imagens, deteção de objetos e segmentação de imagens.

#### **Apresentação do objetivo da aula:**

Introdução à deteção e classificação de objetos para iniciantes através de exemplos de diferentes aplicações.

Comecemos por perceber o que é a classificação de imagens:

Considere a imagem abaixo:





# PARTE PRINCIPAL

Você o reconhece instantaneamente. É um cachorro. Dê um passo para trás e analise como você chegou a essa conclusão. Foi mostrada uma imagem a você e você classificou a classe à qual ela pertencia (um cachorro, neste caso). E é disso que se trata, em poucas palavras, a Classificação de Imagens.

Como você viu, só há um objeto aqui: um cachorro. Podemos facilmente usar um modelo de classificação de imagens e prever que há um cachorro na imagem fornecida. Mas e se tivermos um gato e um cachorro em uma única imagem.



f

<http://erasmus-artie.eu>

Podemos treinar um classificador multi-rótulo nesse caso. Mas não saberemos a localização de nenhum animal/objeto na imagem. É aí que a localização de imagens entra em cena. Ajuda-nos a identificar a localização de um único objeto na imagem fornecida. Caso tenhamos vários objetos presentes, contamos então com o conceito de Detecção de Objetos. Podemos prever a localização junto com a classe de cada objeto usando OD.

**Classification**



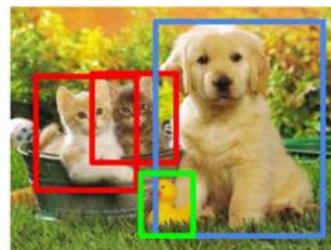
CAT

**Classification + Localization**



CAT

**Object Detection**



CAT, DOG, DUCK

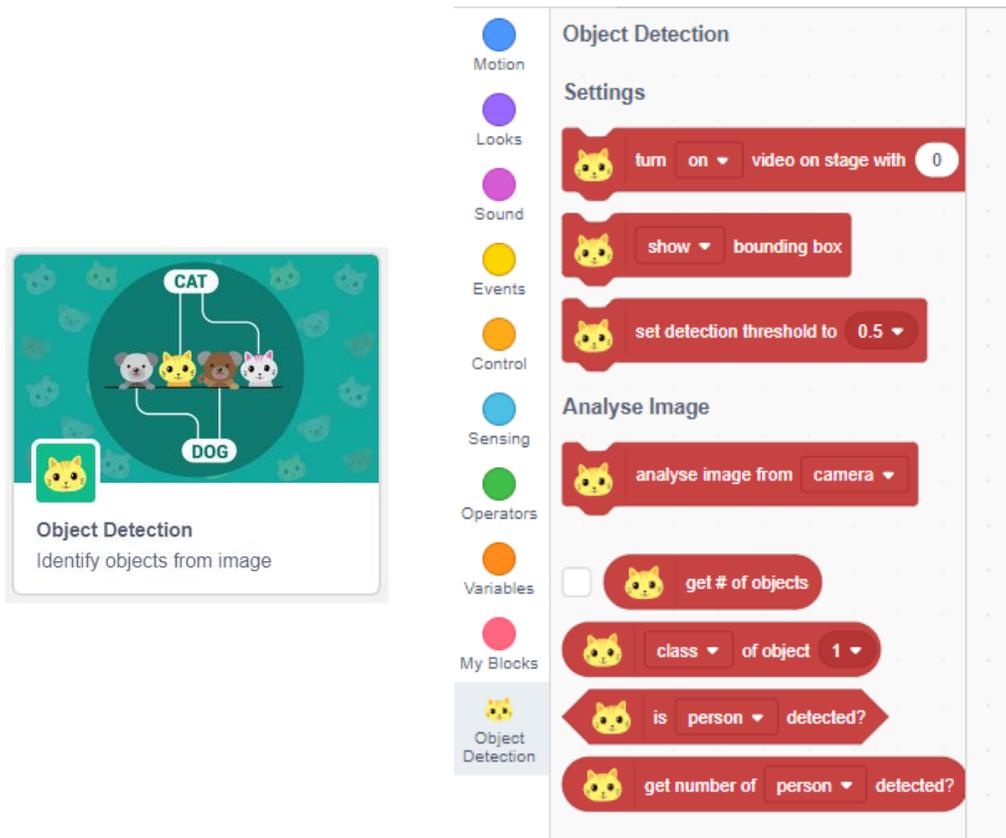


### Aplicativo com detecção de objetos

O PictoBlox é atualmente o único com capacidade de detecção de objetos. É uma aplicação de tipo desktop e deve ser instalada primeiro a partir de <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

Passo 1: Após a instalação, carregar a extensão de detecção de objetos

Passo 2: E tem estes blocos disponíveis para codificação.

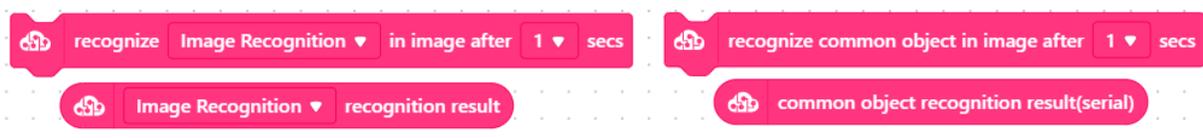


Os primeiros 3 blocos são definições do fluxo da câmara e os outros são utilizados para análise e elaboração de relatórios.

A situação da classificação de objetos é muito melhor, pois temos mais algumas aplicações para utilizar (além do PictoBlox)

**Makeblock (mBlock)** - <https://ide.mblock.cc/>

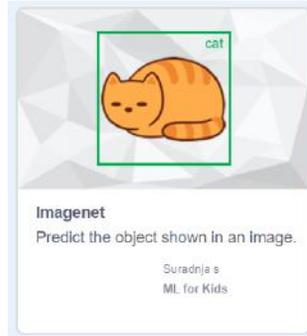
Carregue a extensão Cognitive services e encontrará 4 blocos para utilizar no reconhecimento (classificação) de objectos.



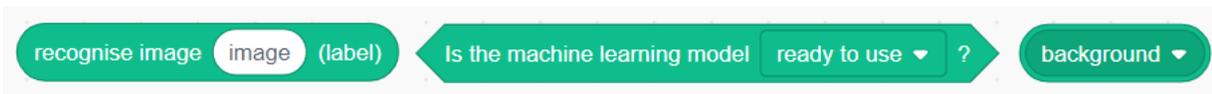


O Makeblock também fornece a extensão de máquina Teachable (que não está relacionada com o Google), onde pode treinar até 3 classes e utilizá-la para a classificação de objectos.

**Scratch (ML4KIDS)** - <https://machinelearningforkids.co.uk/scratch3/>  
Carregar Imagenet extension.



Estão disponíveis 3 blocos. Utilize-os em combinação com a extensão de deteção de vídeo para ligar/desligar o vídeo da câmara e definir a transparência. Foi treinado para reconhecer fotografias de mil objectos comuns. O modelo de aprendizagem automática é baseado no MobileNet (um modelo de aprendizagem automática concebido para dispositivos móveis, pelo que não necessita de muita potência de computação). A lista completa de objectos está disponível <https://storage.googleapis.com/download.tensorflow.org/data/ImageNetLabels.txt>

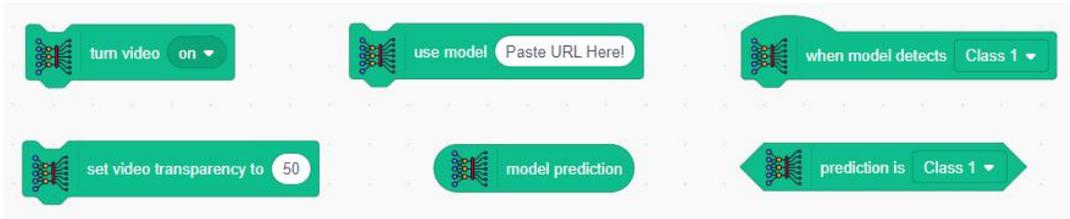


**Scratch (MITMEDIALAB)** - <https://mitmedialab.github.io/prg-extension-boilerplate/create/>  
Carregar Teachable machine extension



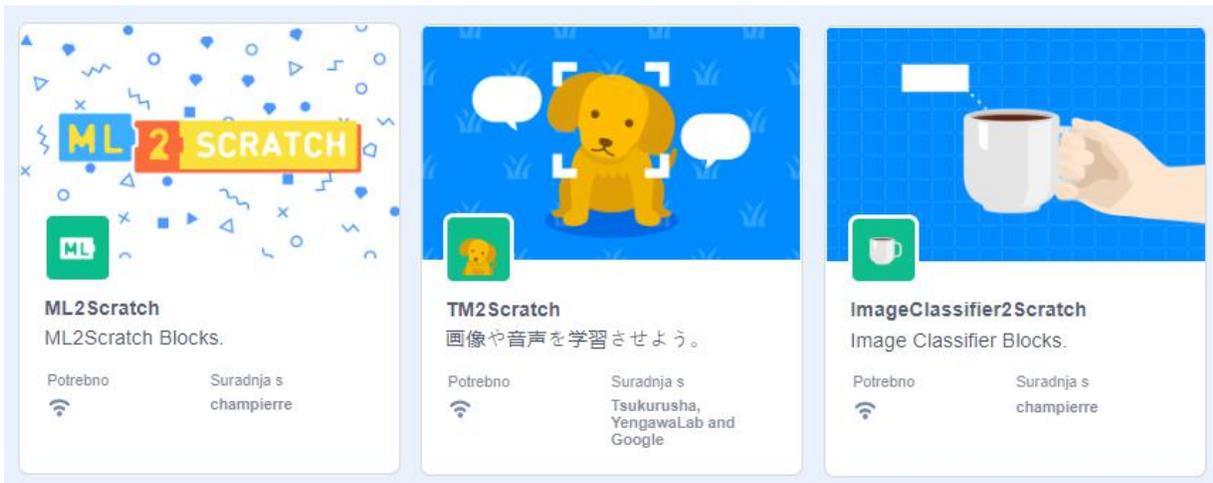


Utilize estes blocos em combinação com a máquina Google Teachable.



Stretch3 (github.io) - <https://stretch3.github.io/>

Carregue as extensões ML2Scratch, TM2Scratch e ImageClassifier2Scratch para usar muitos blocos para classificação e treinamento de objetos



### ML2Scratch blocks





### TM2Scratch blocks



### ImageClassifier2Scratch blocks

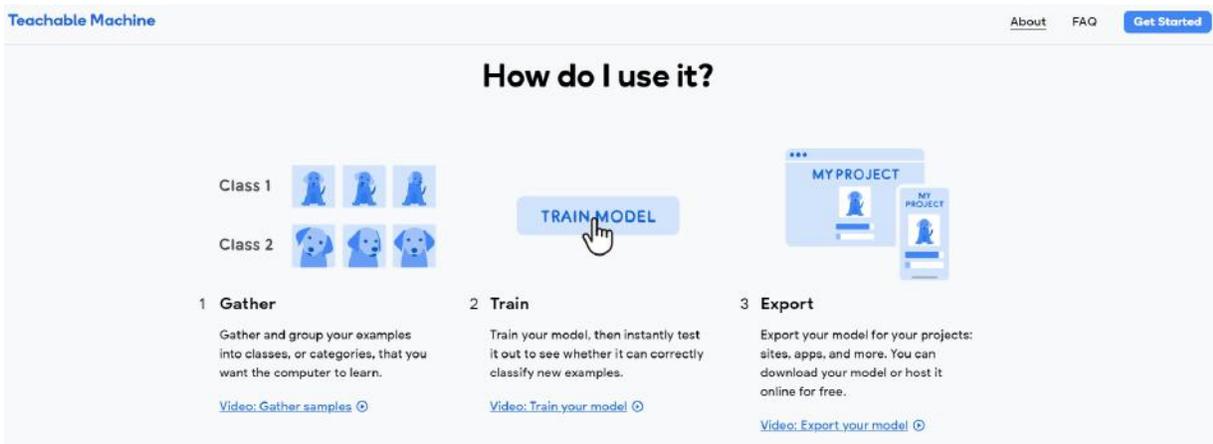


Teachable machine (Google) - <https://teachablemachine.withgoogle.com/>

Esta aplicação é utilizada para treinar o seu modelo e utilizá-lo para o reconhecimento de objectos em combinação com as extensões Teachable machine disponíveis em PictoBlox, Scratch (MITMEDIALAB) e Stretch3



<http://erasmus-artie.eu>

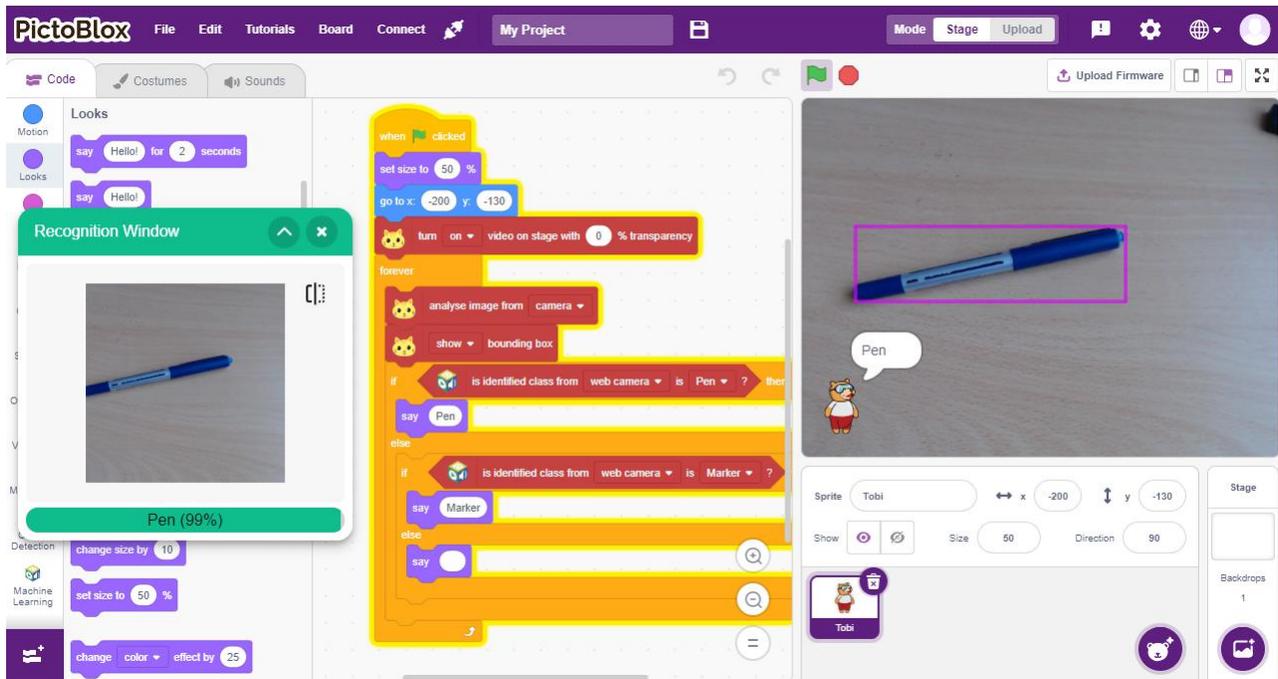


Caso de utilização para deteção de objetos - caneta ou marcador.

A máquina programável é utilizada para formar 2 classes. A ligação de um modelo já testado é:

<https://teachablemachine.withgoogle.com/models/FdWn0CA2a/>

As extensões utilizadas no PictoBlox são a deteção de objectos e a aprendizagem automática.



A detecção e classificação de objetos são duas tarefas principais para a compreensão de imagens. O reconhecimento de objetos em uma imagem requer a combinação de muitos sinais diferentes dos dados brutos da imagem. Dois tipos de informação são frequentemente usados: a aparência local que descreve o objeto em si e a representação global que captura a informação específica da imagem. Esses dois tipos de informações são frequentemente usados em duas tarefas: detecção e classificação de objetos.

### Classificação

A classificação é uma tarefa de aprendizado de máquina para determinar quais objetos estão em uma imagem ou vídeo. Refere-se ao treinamento de modelos de aprendizado de máquina para reconhecer quais classes (objetos) estão presentes. A classificação é útil no nível sim-não para decidir se uma imagem contém um objeto/anomalia ou não.

Uma tarefa separada da classificação é a localização ou determinação da posição dos objetos classificados na imagem ou vídeo.

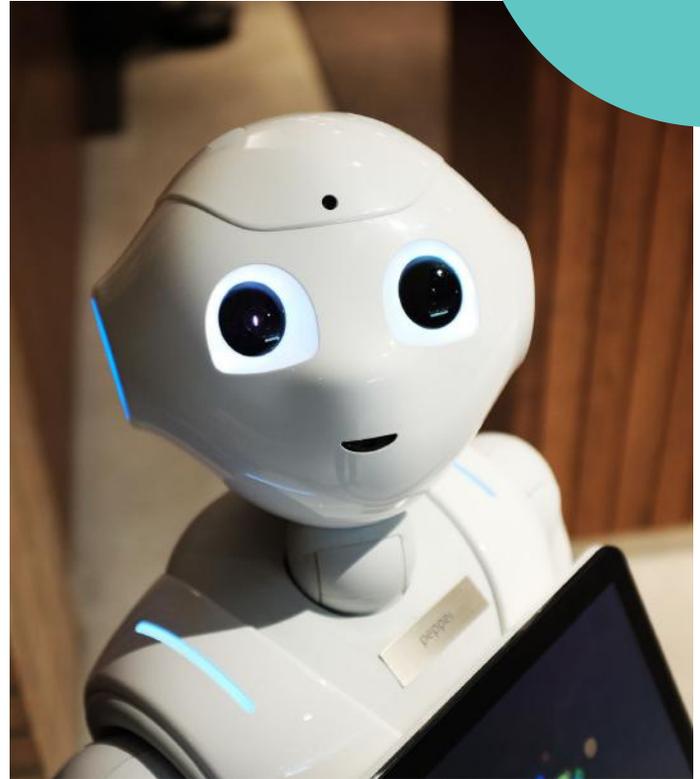




### *Detecção de objetos*

A detecção de objetos combina classificação e localização para determinar quais objetos estão na imagem ou vídeo e especificar onde eles estão na imagem. Aplica classificação a objetos distintos e usa caixas delimitadoras. A detecção de objetos é útil para identificar objetos em uma imagem ou vídeo. Os casos de uso para detecção de objetos incluem detecção facial com qualquer análise pós-deteção, por exemplo, detecção de expressão, estimativa de idade ou detecção de sonolência. Existem muitas aplicações de detecção de objetos em tempo real para gerenciamento de tráfego, como sistemas de detecção de veículos baseados em cenas de tráfego.

Conforme descrito acima, as abordagens mais populares para visão computacional são classificação e detecção de objetos para identificar objetos presentes em uma imagem e especificar sua posição.



<http://erasmus-artie.eu>

## CONCLUSÃO

A detecção e classificação de objetos são duas tarefas principais para análise de imagens.



## Plano de aula 6

Duração: 90 minutos

### Tópicos

Programando detecção de objetos no Scratch

### Objetivos

Para aprender a programar a detecção de objetos com exemplos carregados

### Resultados

Saber como escrever um programa para detecção de objetos usando Scratch

## Programando a detecção de objetos em Scratch

*Para entender a detecção de objetos, vamos revisar o que aprendemos até agora.*

*O que é detecção de objetos?*

*Como funciona a detecção de objetos?*

A detecção de objetos é uma técnica de visão computacional cujo objetivo é identificar e localizar objetos dentro de uma imagem ou vídeo. Especificamente, a detecção de objetos desenha caixas delimitadoras em torno desses objetos detectados, o que nos permite localizar onde esses objetos estão (ou como eles se movem) em uma determinada cena. Antes de começarmos a programar a detecção de objetos, devemos coletar fotos de rostos específicos e treinar o modelo. Usaremos um aplicativo chamado Teachable Machine. Teachable Machine é uma ferramenta baseada na web que torna a criação de modelos de aprendizado de máquina rápida, fácil e acessível a todos

#### Anúncio do objetivo da aula:

Compreender o software de detecção de objetos e seu uso por meio de exemplos.



<http://erasmus-artie.eu>



# PARTE PRINCIPAL

## TAREFA: É um gato ou um cachorro?

Crie um modelo e um programa que detectará se um gato ou um cachorro está no fluxo da câmera

### APLICATIVO BASEADO NA WEB (sem instalação de software)

Etapa 1: abra seu navegador, selecione e baixe imagens em: <https://bit.ly/cats-image-dataset>

Será usado para treinar a Classe 1

Etapa 2: abra seu navegador, selecione e baixe imagens em: <https://bit.ly/dogs-image-dataset>

Será usado para treinar a Classe 2

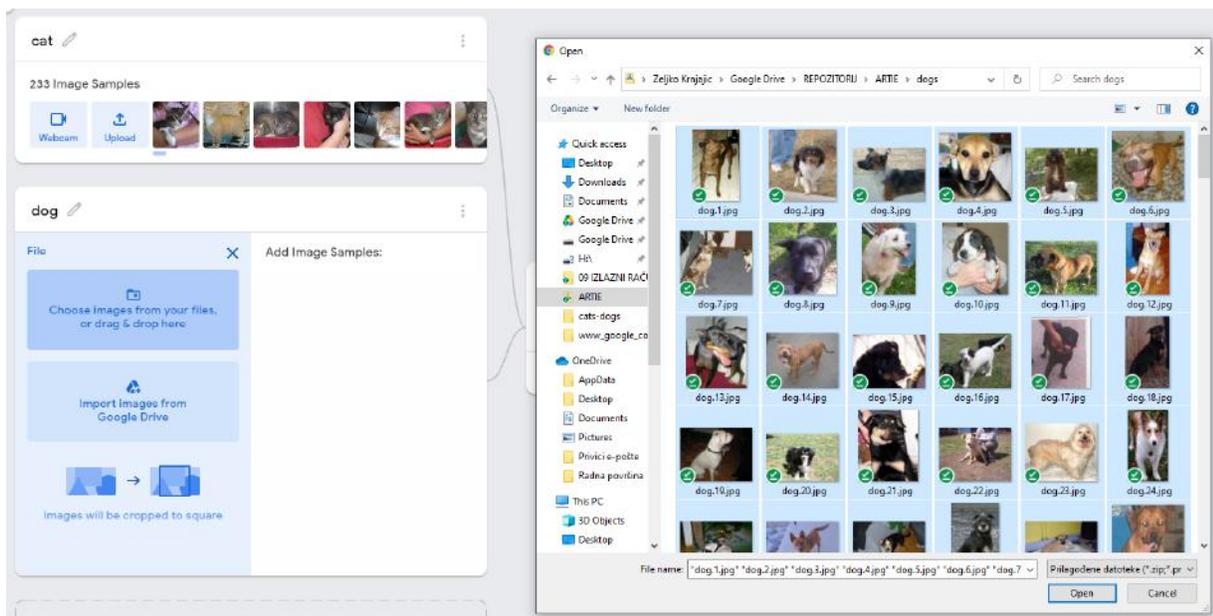
Etapa 3: Abra seu navegador e acesse: <https://teachablemachine.withgoogle.com/>

Passo 4: Clique em Começar.

Passo 5: Escolha o projeto de imagem

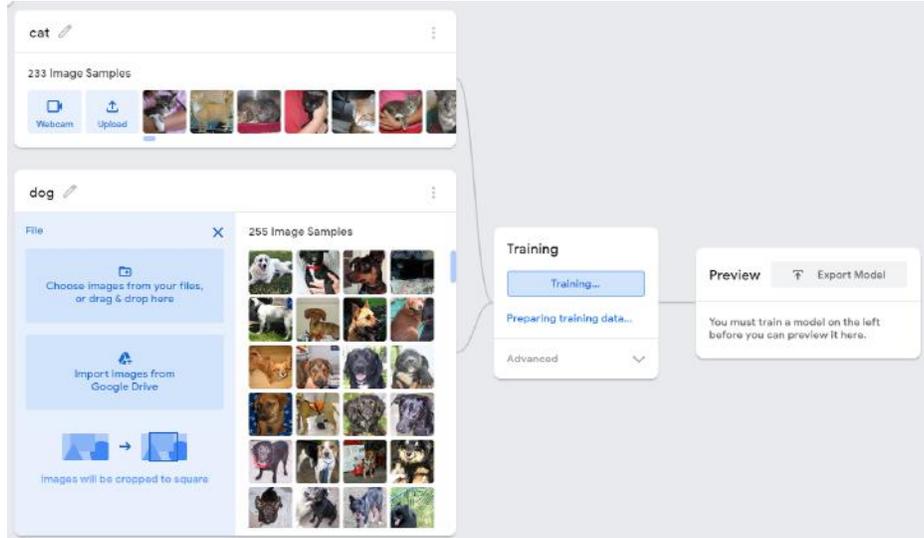
Etapa 6: escolha o modelo de imagem padrão

Etapa 7: altere o nome da Classe 1 para gato e da Classe 2 para cachorro. Faça upload de imagens de gatos para arquivos de gatos e imagens de cães para arquivos de cães, conforme mostrado na imagem abaixo

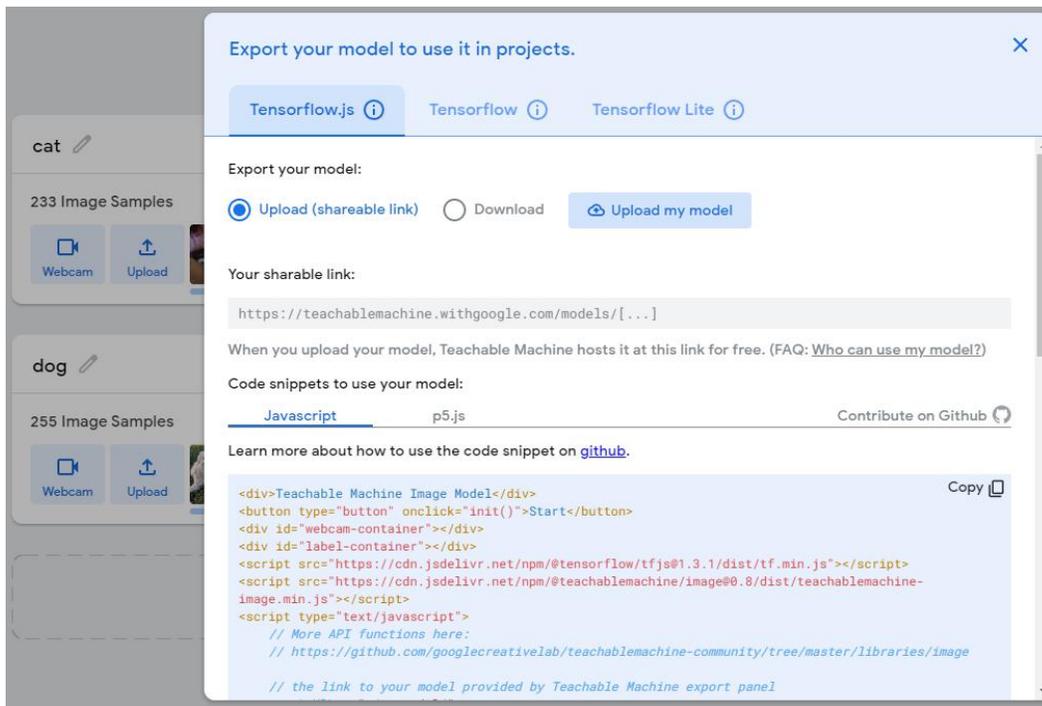




Passo 8: Treine o seu modelo. Não altere os separadores do browser durante o processo de treino.



Passo 9: Exporte o seu modelo. Na janela emergente, escolha carregar o modelo para a nuvem (a terceira opção é Carregar o meu modelo) e a Google alojará os seus dados gratuitamente.





Passo 10: Copie a hiperligação indicada no campo de texto abaixo - este é o URL do seu modelo. No meu caso, era <https://teachablemachine.withgoogle.com/models/gS4NT1mgE/>

Upload (shareable link)    Download  

Your sharable link:

<https://teachablemachine.withgoogle.com/models/gS4NT1mgE/> Copy

When you upload your model, Teachable Machine hosts it at this link for free. (FAQ: [Who can use my model?](#))

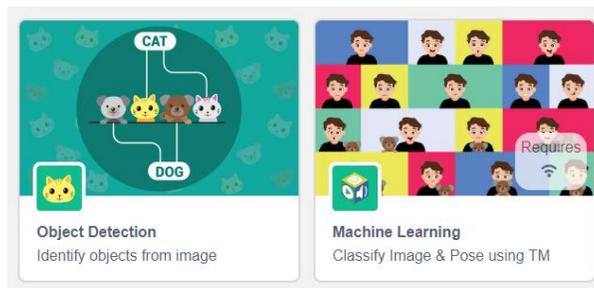
✓ Your cloud model is up to date.

Passo 11: O seu modelo está pronto a ser utilizado

### PICTOBLOX (aplicação de ambiente de trabalho):

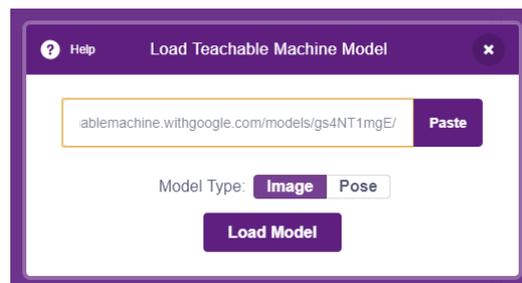
Passo 1a: Descarregar e instalar o PictoBlox, pois é atualmente o único com capacidade de deteção de objectos. É uma aplicação de ambiente de trabalho e deve ser instalada primeiro a partir de <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

Passo 2a: Carregue as extensões Object Detection and Machine Learning.



Passo 3a: Seleccione o grupo Aprendizagem automática e escolha "Carregar um modelo". Cole a ligação do modelo da máquina de aprendizagem (abaixo):

<https://teachablemachine.withgoogle.com/models/gS4NT1mgE/> e clique em Carregar modelo



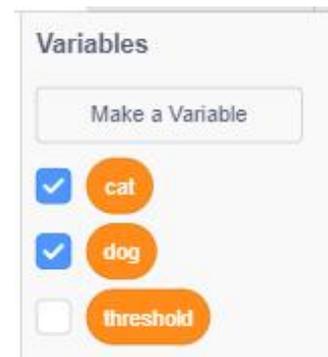
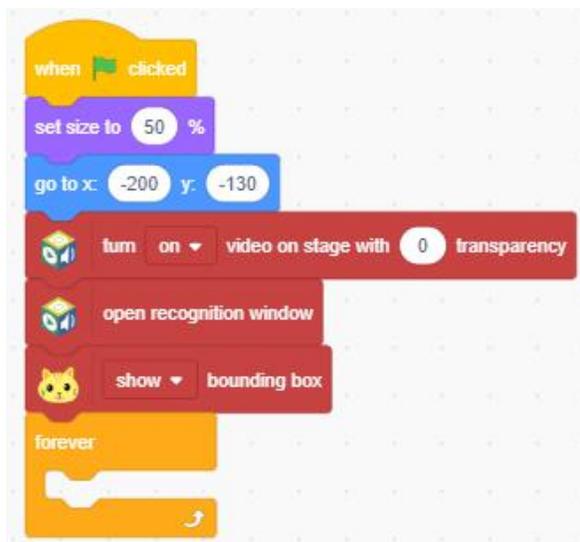


Passo 4a: Vamos fazer a classificação e a deteção de objectos simultaneamente. Primeiro, temos de iniciar o programa, redimensionar e mover o ator Tobi no canto inferior esquerdo. Os blocos seguintes da extensão Aprendizagem Automática são utilizados para ligar o vídeo e abrir a janela de reconhecimento. Utilize o vídeo invertido no palco com transparência 0 se o seu vídeo estiver invertido.



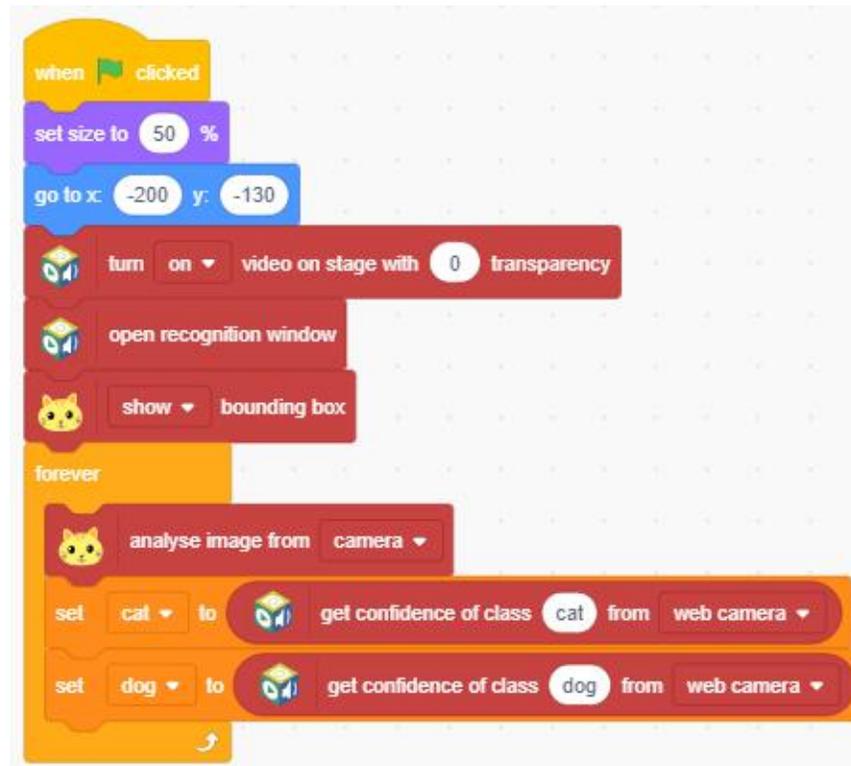
Passo 5a: A partir da extensão Object Detection (Deteção de objectos), utilize o bloco Show Bounding Box (Mostrar caixa delimitadora) para mostrar a posição de um objeto no fluxo de vídeo e adicione um bloco de loop infinito

Passo 6a: Criar 3 variáveis: gato, cão e limiar. Mostrar o gato e o cão no palco, marcando-os. Deixe o limiar desmarcado.

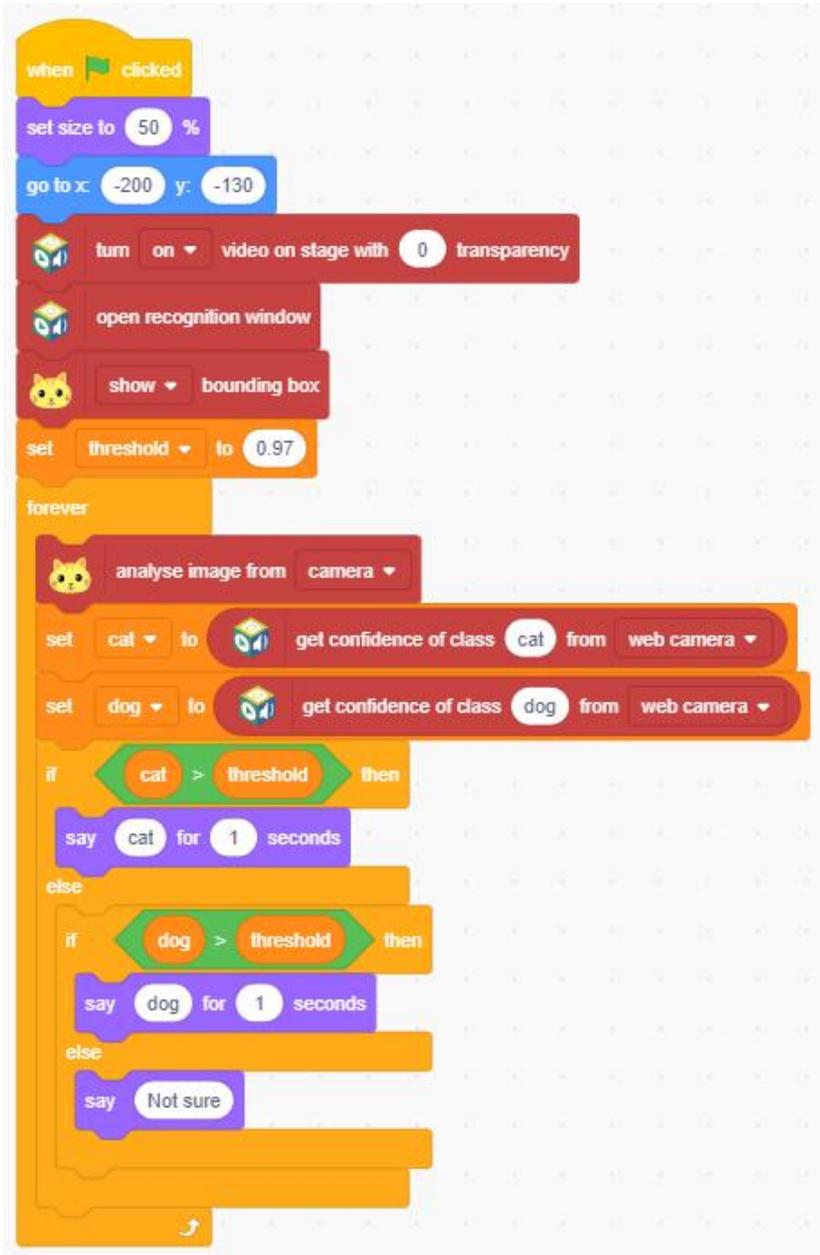




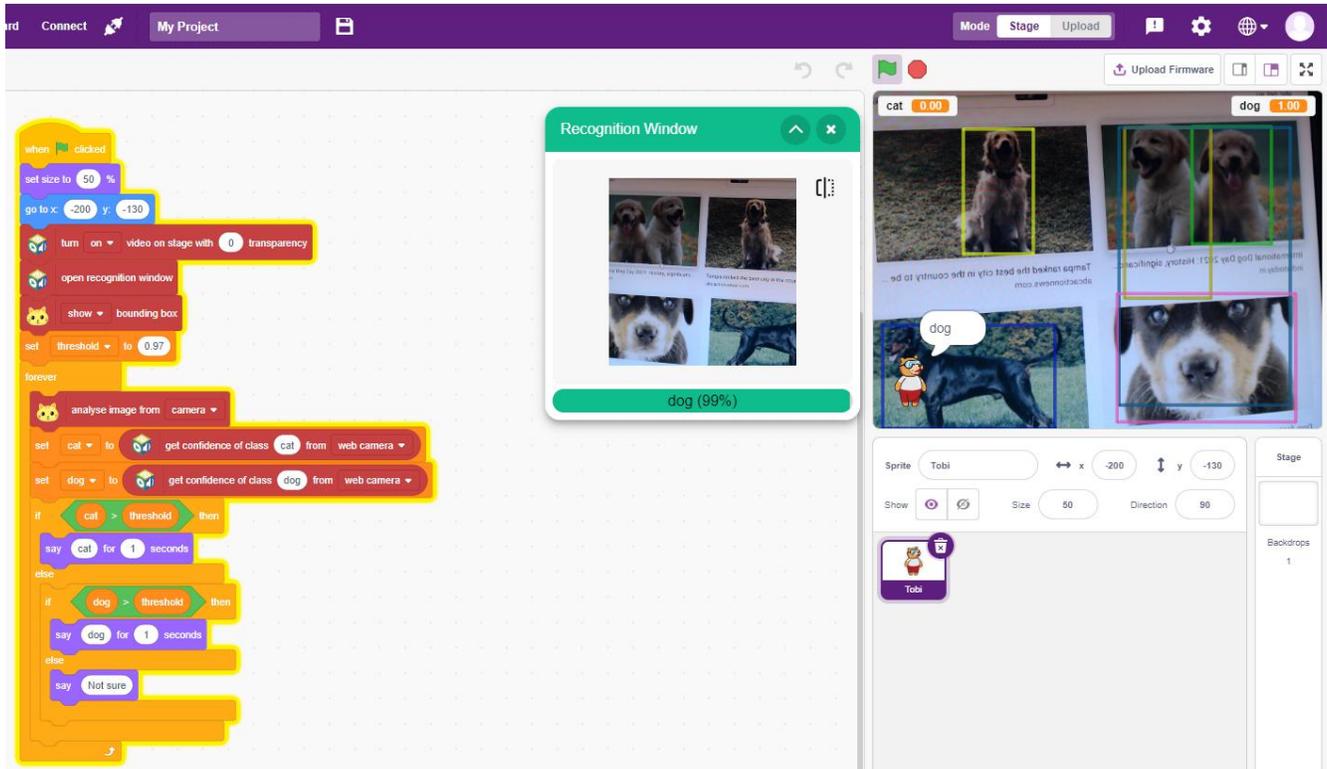
Passo 7a: Primeiro, insira o bloco de limiar definido para 0,97 antes do ciclo para sempre. Esta é a parte principal em que analisamos primeiro a imagem da câmara e armazenamos a confiança dos valores de classe em variáveis.



Passo 8a: É seguido de um bloco duplo if-then-else e de 2 blocos de detetor de classes seguidos de "gato" ou "cão". Se nada for detectado, o texto do balão de diálogo é "Não tenho a certeza".



Passo 9a: Inicie o programa e teste-o no seu cão ou gato. Se não tiver nenhum perto de si, o Google Images está aqui, basta procurar imagens de cães/gatos e apontar a sua câmara Web para o ecrã com os resultados. Verá as caixas delimitadoras dos objectos detectados. Tenta baixar ou aumentar o valor do limiar ou tenta confundir o algoritmo de deteção com algumas imagens específicas de raças de cães ou gatos.



### Então, o que exatamente é detecção de objetos?

Para responder a essa pergunta, vamos começar com a classificação das imagens. Nesta tarefa temos uma imagem e queremos atribuí-la a uma de muitas categorias diferentes (por exemplo, carro, cão, gato, humano,...), então basicamente queremos responder à pergunta “O que há nesta imagem?”. Observe que uma imagem possui apenas uma categoria atribuída a ela. Em palavras simples, a detecção de objetos é um tipo de técnica de classificação de imagens e, além de classificar, essa técnica também identifica a localização das instâncias do objeto a partir de um grande número de categorias predefinidas em imagens naturais.

## CONCLUSÃO

A detecção de objetos é uma técnica de classificação de imagens e, além de classificar, essa técnica também identifica a posição de um objeto encontrado a partir de um grande número de categorias predefinidas.





## Plano de aula 7

Duração: 90 minutos

### Tópicos

Interpretação dos resultados do procedimento de classificação de objetos

### Objetivos

Utilização prática da classificação de objetos

### Resultados

Explorar as possibilidades de extensão da aprendizagem automática no scratch

## Projeto de classificação de objetos

*O que é detecção de objetos?  
Como funciona a detecção de objetos?*

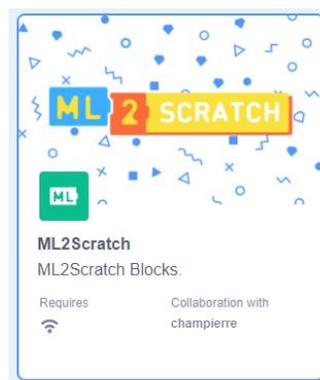
#### Apresentação do objetivo da aula:

Compreender o procedimento de classificação de objetos e a sua utilização através de projetos práticos.

## PARTE PRINCIPAL

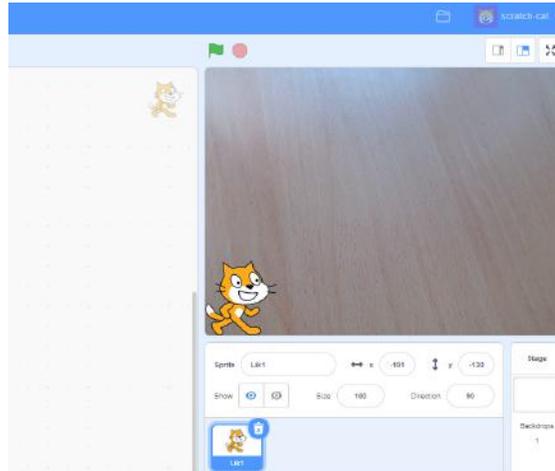
Passo 1: Ligar a câmara Web

Passo 2: Abrir o scratch e adicionar a extensão "ML2SCRATCH"

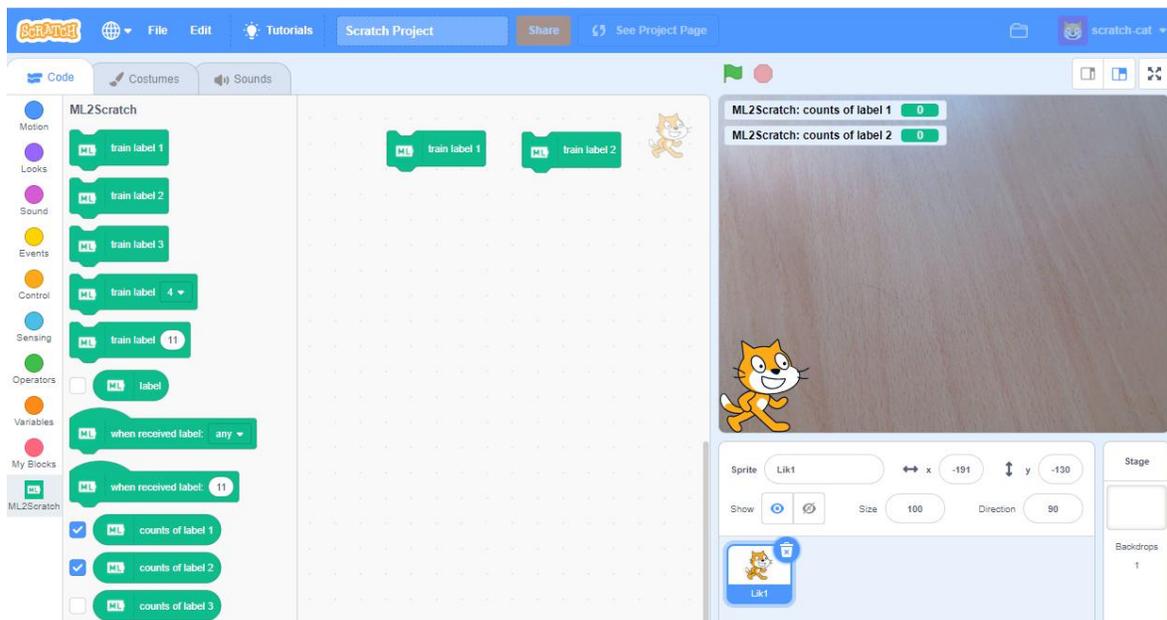




Passo 3: Limpar a área de trabalho e apontar a webcam para uma área vazia, mover o ator do gato para o canto, como mostrado



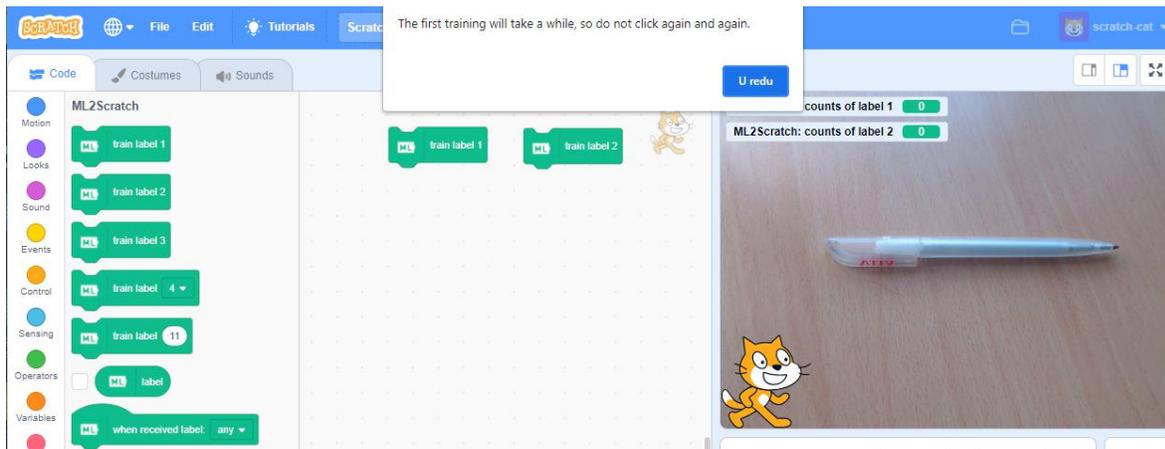
Passo 4: No grupo ML2SCRATCH, selecionar um bloco de etiquetas 1 e outro de etiquetas 2 e colocá-lo numa área de programação e verificar as contagens das etiquetas 1 e 2, como indicado





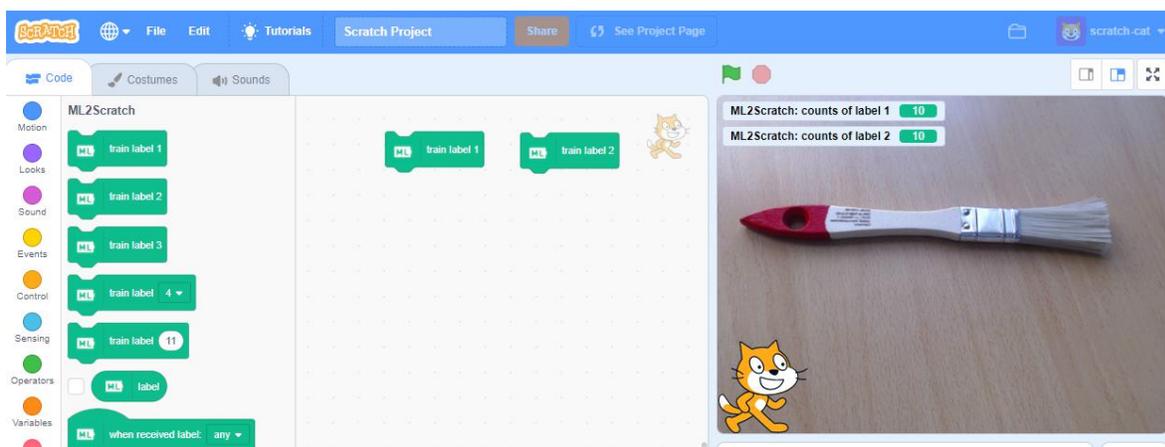
Passo 5: Preparar dois grupos de objetos para a aprendizagem automática. No nosso caso, serão utilizados pincéis e canetas para treinar as etiquetas

Passo 6: Coloque o primeiro objeto do primeiro grupo na área para a qual a webcam está apontada e clique no bloco "train label 1" (treinar etiqueta 1) - será notificado para aguardar um pouco, como mostrado, e a contagem de etiquetas mudará para 1



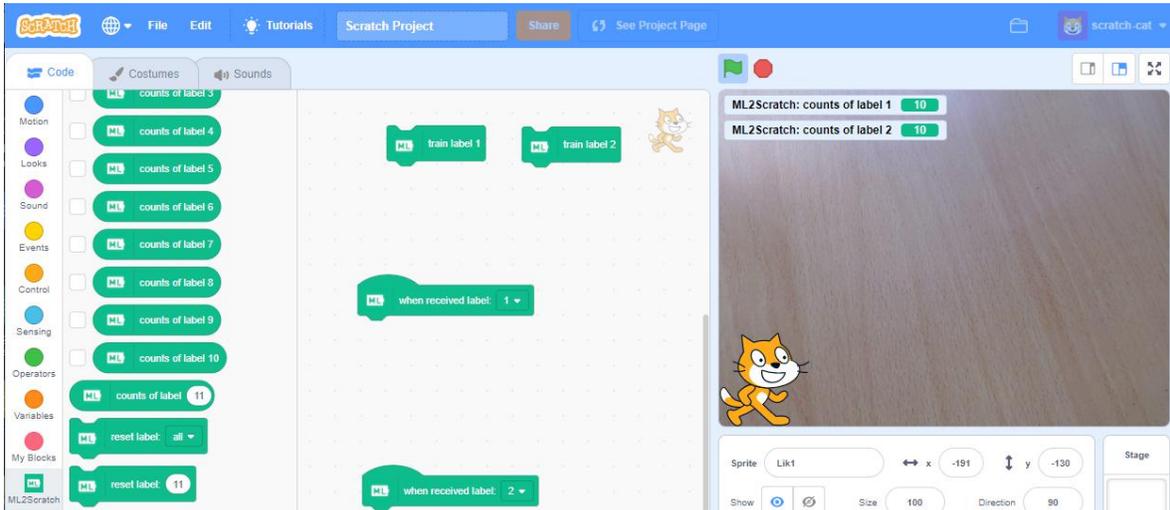
Etapa 7: Obter cerca de 10 imagens de cada objeto com formas e colocações.

Não se esqueça de aplicar a etiqueta correcta, não misture os objetos e as etiquetas.



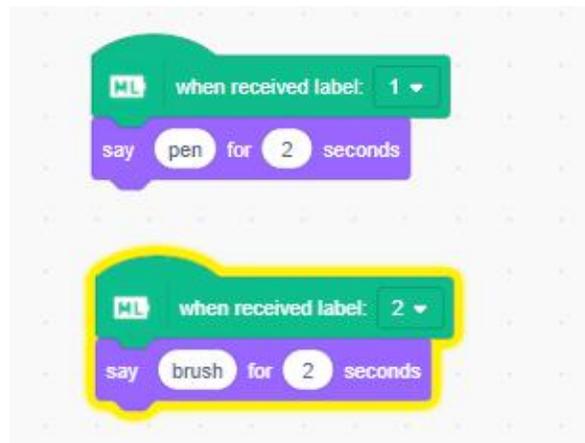


Passo 8: Agora escolhe dois blocos quando receberes a etiqueta: qualquer e muda qualquer para 1 no primeiro e muda qualquer para 2 no segundo bloco, como mostrado



f

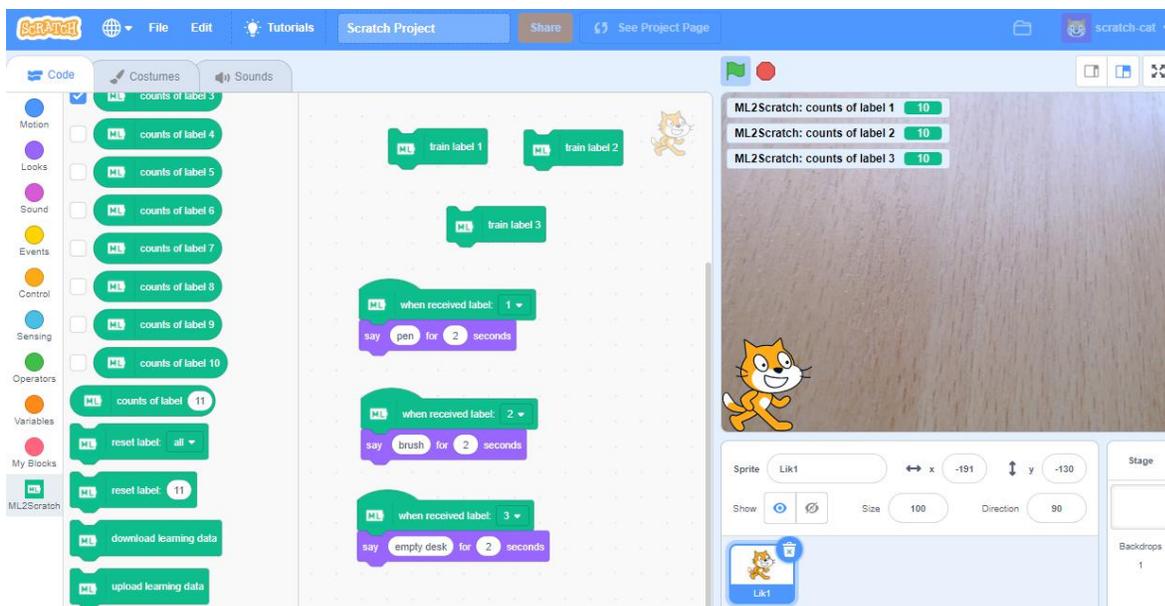
Passo 9: No grupo Look (Olhar), selecionar dois blocos "Hello! for 2 seconds" e mudar Hello! para caneta no primeiro bloco e mudar Hello! para pincel no segundo bloco, como se mostra.





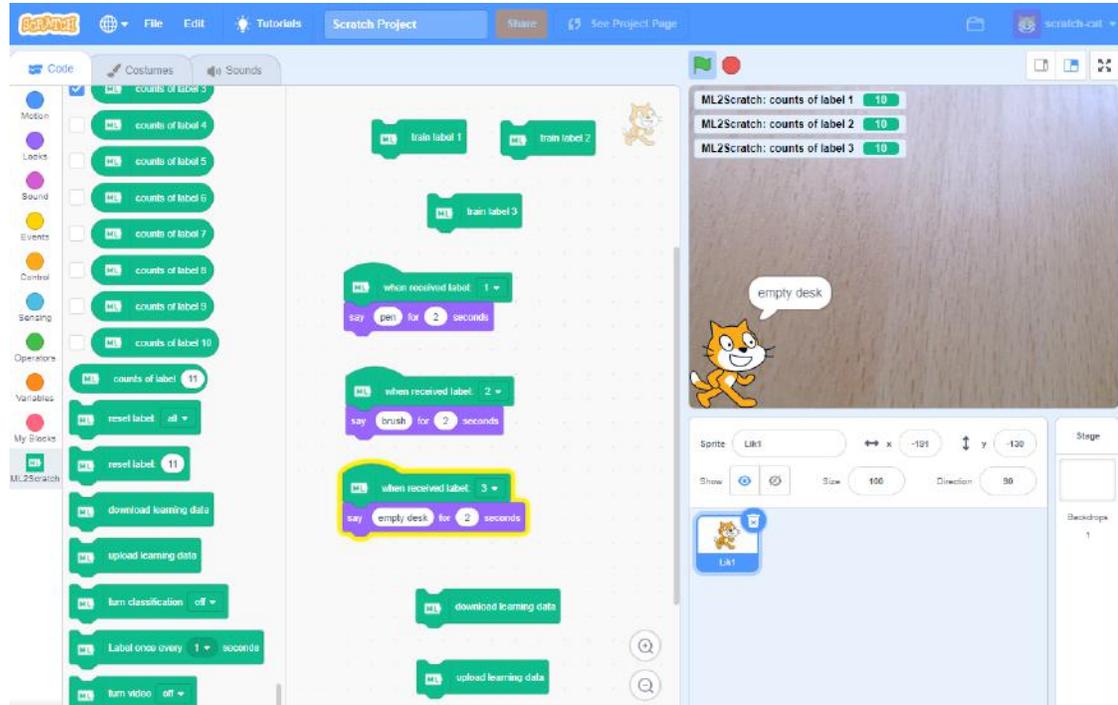
Passo 10: Coloque aleatoriamente objetos de dois grupos na área de visão da câmara e observe o que está a acontecer. Está a funcionar? O que acontece quando nenhum objeto é colocado? Como resolver o problema da secretária vazia?

Passo 11: Solução: Utiliza a terceira etiqueta, treina-a numa secretária vazia e adiciona a etiqueta quando recebida 3 blocos com a indicação de secretária vazia durante 2 segundos, como mostrado



Passo 12: Discuta a exatidão da previsão. Tente treinar mais cada etiqueta e compare-a com os resultados anteriores. A precisão melhora?

Passo 13: Utilize mais grupos de objetos para calcular mais rótulos. Descarregue e carregue a sua dana treinada com os blocos descarregar dados de aprendizagem e carregar dados de aprendizagem (basta clicar no bloco para guardar ou carregar o ficheiro .json)



Em palavras simples, classificação de imagens é uma técnica usada para classificar ou prever a classe de um objeto específico em uma imagem. O principal objetivo desta técnica é identificar com precisão as características de uma imagem.

f

<http://erasmus-artie.eu>

Em geral, as técnicas de classificação de imagens podem ser categorizadas como paramétricas e não paramétricas ou supervisionadas e não supervisionadas, bem como classificadores rígidos e suaves. Para classificação supervisionada, esta técnica fornece resultados com base no limite de decisão criado, que depende principalmente das entradas e saídas fornecidas durante o treinamento do modelo. Mas, no caso de classificação não supervisionada, a técnica fornece o resultado com base na análise do próprio conjunto de dados de entrada; os recursos não são alimentados diretamente aos modelos. As principais etapas envolvidas nas técnicas de classificação de imagens são determinar um sistema de classificação adequado, extração de características, seleção de boas amostras de treinamento, pré-processamento de imagens e seleção do método de classificação apropriado, processamento pós-classificação e, finalmente, avaliação da precisão geral. Nesta técnica, as entradas são geralmente uma imagem de um objeto específico e as saídas são as classes previstas que definem e correspondem aos objetos de entrada.

## CONCLUSÃO

A classificação de imagens é uma técnica usada para classificar ou prever a classe de um determinado do objeto na imagem. O principal objetivo desta técnica é identificar com precisão as características de uma imagem.



## Plano de aula 8

**Duração: 90 minutos**

### Tópicos

Reconhecimento e geração de voz para principiantes em Scratch

### Objetivos

Reconhecer a geração de voz para principiantes no Scratch

### Resultados

Compreender o reconhecimento e a geração de fala com a ajuda de um exemplo simples no Scratch.

## Reconhecimento e geração de voz para principiantes em Scratch

### Introdução ao reconhecimento de fala

O reconhecimento da fala é a capacidade de traduzir um ditado ou uma palavra falada em texto. É também conhecido por Speech-to-Text (STT) e Voice Recognition (reconhecimento de voz).

É conseguido através de determinados passos e o software responsável por ele é conhecido como "Sistema de Reconhecimento de Fala". Os sistemas de SR são geralmente implementados sob a forma de software de ditado e assistentes inteligentes em computadores pessoais, smartphones, navegadores Web e muitos outros dispositivos.

### Introdução à geração de fala

A geração ou síntese de fala (também abreviada como TTS, Text-to-Speech), ao contrário do reconhecimento de fala, não é uma tecnologia que explora a voz, mas sim que a produz. As vozes sintéticas são geralmente a fase final do processo e estão a tornar-se cada vez mais democráticas porque são importantes na experiência global da "voz".

### Definir o objetivo da aula:

Introdução ao reconhecimento e geração de fala para principiantes, usando um exemplo de um programa simples em Makeblock.



<http://erasmus-artie.eu>



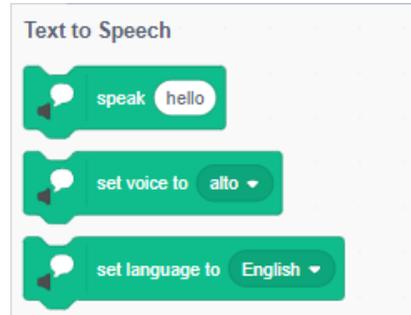
# PARTE PRINCIPAL

## Apliações

### Scratch (MIT)

<https://scratch.mit.edu/projects/editor/>

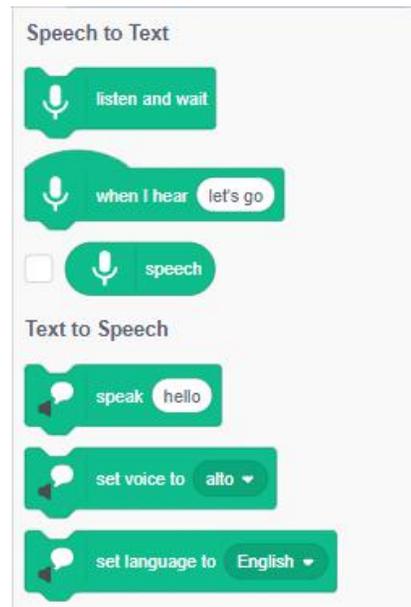
Apenas a extensão Text-to-Speech está disponível (3 blocos)



### Scratch (ML4KIDS)

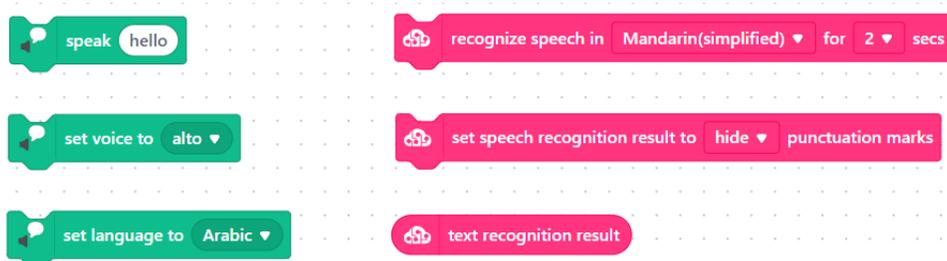
<https://machinelearningforkids.co.uk/scratch3/>

Estão disponíveis extensões de conversão de texto em voz (3 blocos) e de conversão de texto em voz (3 blocos)



### Makeblock - <https://ide.mblock.cc/>

Conversão de texto em fala (3 blocos), serviços cognitivos (3 blocos relacionados com a conversão de texto em fala)



<http://erasmus-artie.eu>



### \* BONUS

O Makeblock também inclui blocos para reconhecimento de linguagem e análise de sentimentos de texto



A análise de sentimentos é o processo de detecção de sentimentos positivos ou negativos no texto. É frequentemente utilizada pelas empresas para detetar sentimentos em dados sociais, avaliar a reputação da marca e compreender os clientes. Os modelos de análise de sentimentos centram-se na polaridade (positivo, negativo, neutro), mas também nos sentimentos e emoções (zangado, feliz, triste, etc.), na urgência (urgente, não urgente) e até nas intenções (interessado vs. não interessado).

Saiba mais sobre a análise de sentimentos: <https://monkeylearn.com/sentiment-analysis/>

Para ilustrar isto, vamos dar um exemplo com o Makeblock.

Passo 1: Abrir a página do Makeblock: <https://ide.mblock.cc/>

Passo 2: Adicionar extensões: Serviços cognitivos e Texto para discurso

Passo 3: Marque para apresentar os seguintes blocos de tipo de repórter:

- ✓ resultado do reconhecimento de fala
- ✓ resultado do reconhecimento da língua
- ✓ resultado da análise de sentimentos

Step 4: Use speech recognition result in recognize language and analyse text sentiment blocks

Step 5: Use this sequence of blocks





```

when clicked
  recognize speech in English for 2 secs
  recognize language: speech recognition result
  analyze English text sentiment speech recognition result
  wait 2 seconds
  set voice to tenor
  set language to English
  if sentiment analysis result = negative then
    speak This is not good
  if sentiment analysis result = positive then
    speak This is good

```

f

Passo 6: Ligue o microfone e os altifalantes, inicie o código e diga "I am happy" (verá a janela pop-up a gravar a sua voz durante 2 segundos - se precisar de mais, aumente o valor em reconhecer discurso em inglês para 3 segundos (ou mais). Irá ouvir "Isto é bom" ou "Isto não é bom", dependendo dos resultados da análise de sentimentos.





Passo 7: Recomeçar o código, dizer outra coisa e esperar pelos resultados da análise de sentimentos .

### Princípios básicos do reconhecimento de fala

A unidade mais pequena da linguagem falada é conhecida como fonema. A língua inglesa contém aproximadamente 44 fonemas que representam todas as vogais e consoantes que utilizamos na fala. Podemos dar o exemplo de uma palavra típica como moon, que pode ser dividida em três fonemas: m, ue, n.

Para interpretar o discurso, temos de ter uma forma de identificar os componentes das palavras faladas e os fonemas que actuam como marcadores de identificação dentro de um discurso. Deve ser utilizado um algoritmo para interpretar o discurso. O modelo oculto de Markov é um modelo matemático utilizado para o fazer. Para criar um motor de reconhecimento de fala, é criada uma grande base de dados de modelos para corresponder a cada fonema.

Saiba mais: <https://www.ibm.com/cloud/learn/speech-recognition>

### Princípios básicos da geração de fala

Ao contrário dos sistemas de reconhecimento de voz que utilizam fonemas (as unidades mais pequenas de som) para cortar frases, o TTS baseia-se nos chamados grafemas: as letras e grupos de letras que transcrevem um fonema. Isto significa que o recurso básico não é o som, mas o texto. Isto é normalmente feito em duas etapas.

Na primeira, o texto é dividido em frases e palavras (os nossos famosos grafemas) e são atribuídas transcrições fonéticas, a pronúncia, a todos estes grupos. Uma vez identificados os diferentes grupos de texto/fonética, a segunda etapa consiste em converter estas representações linguísticas em som. Por outras palavras, ler estas indicações para produzir uma voz que leia a informação.

Experimente o TTS em linha: <https://www.readspeaker.com/>

O reconhecimento da fala é a capacidade de traduzir um ditado ou uma palavra falada em texto. Também é conhecido como Speech-to-Text e reconhecimento de voz. É conseguido seguindo certos passos e o software responsável por ele é conhecido como "Sistema de reconhecimento de fala". Os sistemas de reconhecimento de voz são normalmente implementados sob a forma de software de ditado e assistentes inteligentes em computadores pessoais, smartphones, navegadores Web e muitos outros dispositivos.

## CONCLUSÃO

O reconhecimento de fala é a capacidade de traduzir o ditado ou a palavra falada em texto. A geração de fala é uma tecnologia que cria uma voz.



## Plano de aula 9

Duração: 90 minutos

### Tópicos

Programar o reconhecimento de voz no Scratch

### Objetivos

Aprender a programar o reconhecimento de voz com exemplos carregados

### Resultados

Saber escrever um programa de geração de fala utilizando o Scratch

## Programar o reconhecimento de voz no scratch

*Pergunte aos alunos o que é o reconhecimento de voz?*

*Pergunte aos alunos o que é o SIRI e o Google NOW?*

*Conhecem algum outro sistema de reconhecimento de fala?*

Nos últimos anos, a tecnologia de reconhecimento de voz tem vindo a generalizar-se cada vez mais. Esta tecnologia é frequentemente utilizada por empresas e indivíduos devido aos inúmeros benefícios que traz.

"Hey, Siri", "OK Google", etc. - O reconhecimento de voz, muitas vezes conhecido como tecnologia de reconhecimento de voz, não é um conceito novo (SRT). Refere-se a um tipo de tecnologia que pode transformar palavras faladas em formas legíveis por máquinas. Agora, pode comunicar com os seus dispositivos e fazer com que estes atuem de acordo com as suas ordens, tal como nos contos de ficção científica. Uma vez que a maioria das tecnologias de reconhecimento de voz tem uma taxa de precisão superior a 95%, não é de admirar que as estatísticas mais recentes de pesquisa por voz revelam que quase 50% de todas as pesquisas em 2022 serão efectuadas por voz.

#### Definição do objetivo da aula:

Introdução a um programa de reconhecimento de fala e à sua utilização, utilizando o exemplo de um programa.



<http://erasmus-artie.eu>

# PARTE PRINCIPAL

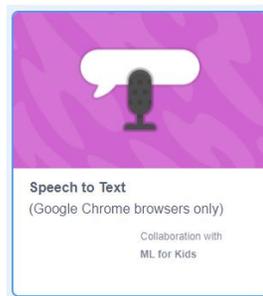
Se quisermos trabalhar com o reconhecimento de voz, existem extensões disponíveis nas aplicações Scratch e Makeblock.

## SCRATCH (ML4KIDS):

Passo 1: Abra o seu navegador Web Chrome e vá para:

<https://machinelearningforkids.co.uk/scratch3/>

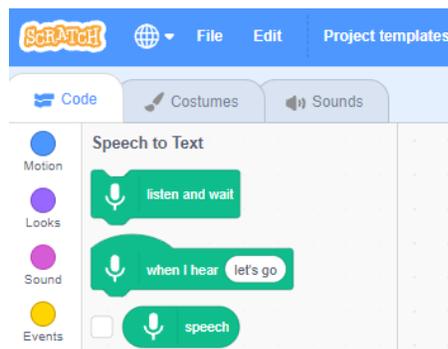
Passo 2: Carregue a extensão Speech to Text



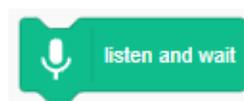
Passo 3: Verá um novo grupo na paleta de blocos chamado "Speech to Text" e 3 novos blocos



<http://erasmus-artie.eu>



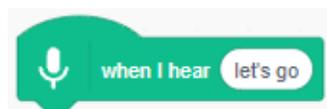
Passo 4: O bloco "Ouvir e esperar" começa a ouvir e a processar as palavras faladas.



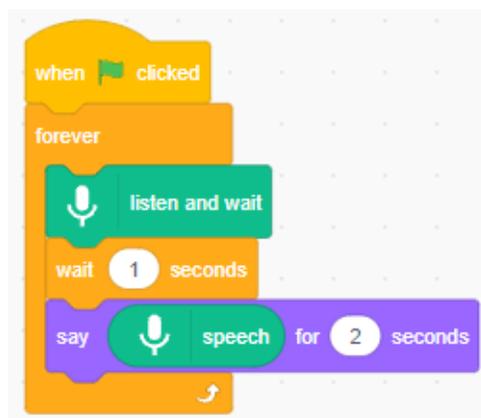
Passo 5: Os resultados do reconhecimento de voz são apresentados no bloco do relator de voz. Verifique se pretende que o resultado do reconhecimento de voz seja apresentado no palco .



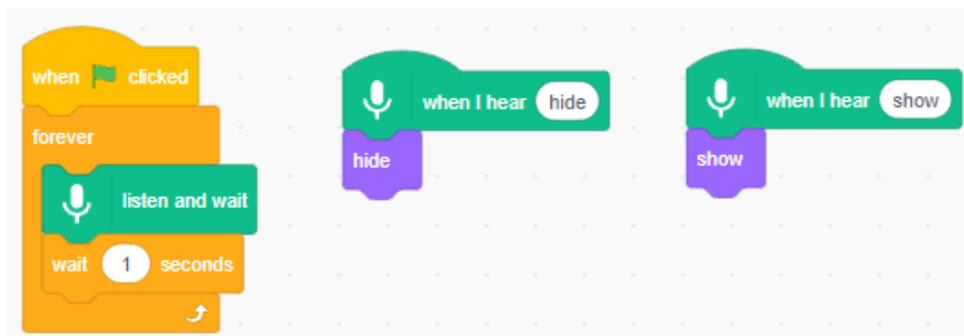
Passo 6: O último bloco é o bloco de ativação de eventos. Este bloco espera pela palavra em balão branco, como "algo" neste exemplo, e depois executa a sequência de blocos que lhe está associada.



Passo 7: Vamos então fazer um programa simples de ouvir e dizer. Tudo o que tens de fazer é colocar, ouvir e esperar num ciclo com o bloco say para veres o reconhecimento de voz em ação.



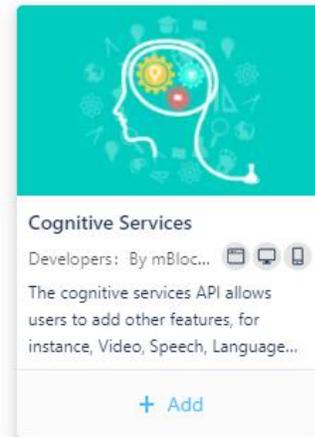
Passo 8: Também podes esperar por uma ou mais palavras específicas para ativar o(s) evento(s). Por exemplo, este jogo de esconder e mostrar.



## MAKEBLOCK

Passo 1a: Abra o seu navegador Web e acesse a:  
<https://ide.mblock.cc/>

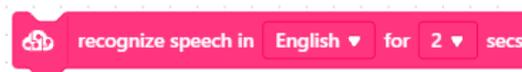
Passo 2a: Carregar a extensão Sprite dos Serviços Cognitivos



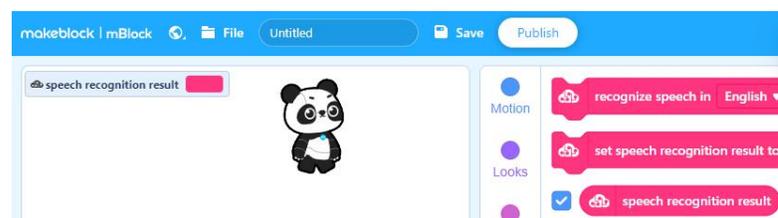
Passo 3a: Há muitos blocos neste grupo, mas apenas alguns estão relacionados com o reconhecimento de voz.



Passo 4a: Reconhecer a fala em <língua> durante <x> segundos. O bloco inicia o reconhecimento de fala durante vários segundos. A janela pop-up RECONHECIMENTO aparecerá e verá uma forma de onda ao falar.



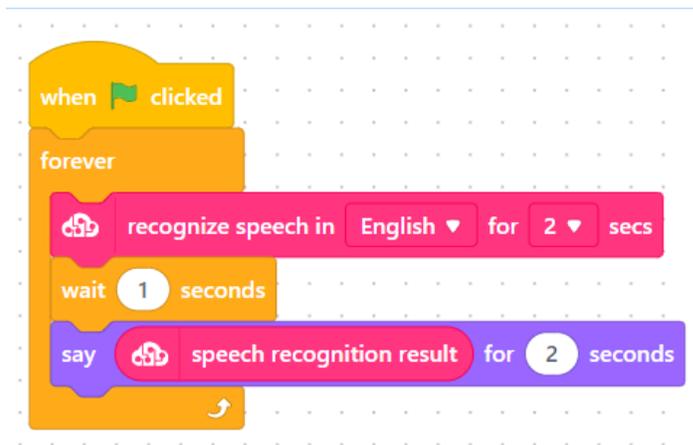
Passo 5a: Os resultados do reconhecimento de fala são exibidos no bloco de resultados do reconhecimento de fala. Verifique se pretende que o resultado do reconhecimento de voz seja apresentado no palco.



Passo 6a: o último bloco mostra ou esconde os sinais de pontuação do resultado do reconhecimento de voz



Passo 7a: Vamos fazer um programa simples de ouvir e dizer no Makeblock. Verás que é muito semelhante à versão Scratch.



A tecnologia de reconhecimento de voz faz agora parte da nossa vida quotidiana, embora ainda se limite a pedidos simples. Com o progresso da tecnologia, os investigadores poderão desenvolver sistemas mais sofisticados que interpretam o discurso conversacional. Um dia, será possível conversar com o computador da mesma forma que se conversa com um ser humano, e ele responderá com respostas racionais. A tecnologia de processamento de sinais tornará tudo isto possível. A procura de profissionais neste sector está a aumentar e muitas organizações estão à procura de pessoas brilhantes para se juntarem a elas. Muitas das novas e poderosas tecnologias e formas de comunicação dependem do processamento, interpretação e compreensão dos sinais de voz. Considerando as tendências atuais, o reconhecimento da fala continuará a ser um subconjunto do processamento de sinais em rápido crescimento nos próximos anos...



<http://erasmus-artie.eu>

## CONCLUSÃO

A tecnologia de reconhecimento de fala permite a comunicação de voz entre usuários e computadores.



## Plano de aula 10

**Duração: 90 minutos**

### Tópicos

Programar a geração de voz no Scratch

### Objetivos

Aprender a programar a geração de voz com exemplos carregados

### Resultados

Saber escrever um programa de geração de fala utilizando o Scratch

## Programar a geração de voz em Scratch

Text-to-speech (TTS) é um tipo de tecnologia de apoio que lê texto digital em voz alta. Por vezes, é designada por tecnologia de "leitura em voz alta". Com um clique de um botão ou um toque de um dedo, a TTS pode pegar nas palavras de um computador ou de outro dispositivo digital e convertê-las em áudio. O TTS é muito útil para crianças e adultos que têm dificuldades com a leitura. Mas também pode ajudar a escrever e a editar, e até a concentrar-se.

Um sistema Concept-to-Speech (CTS) converte a representação concetual de uma frase a ser dita em discurso. Embora alguns sistemas CTS consistam em módulos independentes de geração de texto e de conversão de texto em fala (TTS), a maioria dos sistemas CTS existentes melhora a ligação entre estes dois módulos com um módulo de previsão prosódica que utiliza conhecimentos linguísticos do gerador de texto para prever características prosódicas para a geração de TTS.

A geração do discurso pode transformar qualquer texto em discurso. A geração do discurso consiste em produzir mensagens faladas em resposta a sinais de um sistema de processamento de dados ou de controlo. A seleção de mensagens é produzida através da montagem de sons de fala a partir de um conjunto de fundamentos que podem ser de origem artificial ou podem ter sido extraídos através do processamento de sons produzidos por humanos.

#### Definição do objetivo da aula:

Introdução a um programa de geração de discurso e à sua utilização através de um exemplo de um programa.



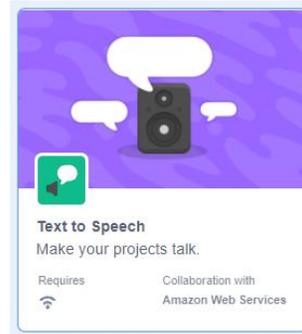
<http://erasmus-artie.eu>



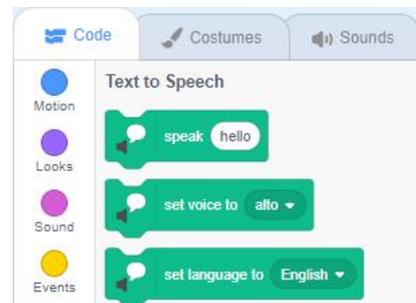
# PARTE PRINCIPAL

Passo 1: Abra o seu navegador Web Chrome e acesse a: <https://machinelearningforkids.co.uk/scratch3/>

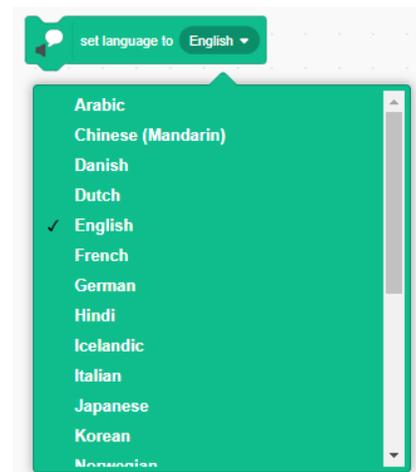
Passo 2: Carregue a extensão Text to Speech



Passo 3: Verá o novo grupo na paleta de blocos chamado "Text to Speech" e 3 novos blocos



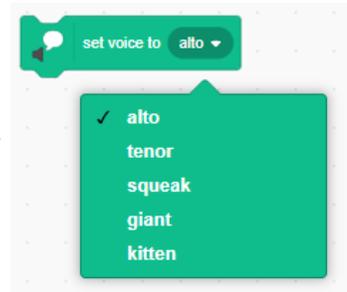
Passo 4: Vamos começar por baixo - definir Idioma para o bloco  
Este bloco define o idioma de saída - pode escolhê-lo na lista pendente



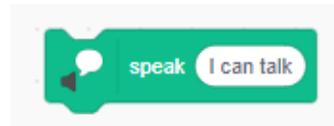


Passo 5: Bloco seguinte - definir voz para definir o tipo de voz.

Podes escolher: alto, tenor, guincho, gigante ou gatinho.



Passo 6: E o bloco mais importante - o bloco "falar". Este bloco "fala" o texto em balão branco, como "Eu posso falar" no exemplo abaixo. Altere o texto para o que quiser e clique no bloco para o ouvir. Certifica-te de que o volume do altifalante está ligado antes de testares.

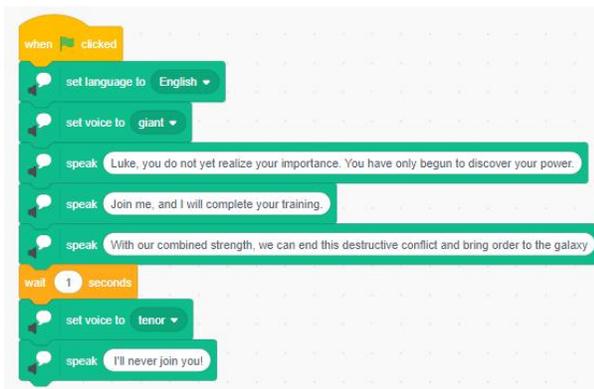


Passo 7: Basicamente, não é o trabalho mais difícil fazer com que o ator do Scratch fale. Tudo o que tem de fazer é definir a língua, a voz e começar a falar.



Passo 8: O que podes fazer com ele?

Podes recriar as famosas falas dos filmes, como esta da Guerra das Estrelas

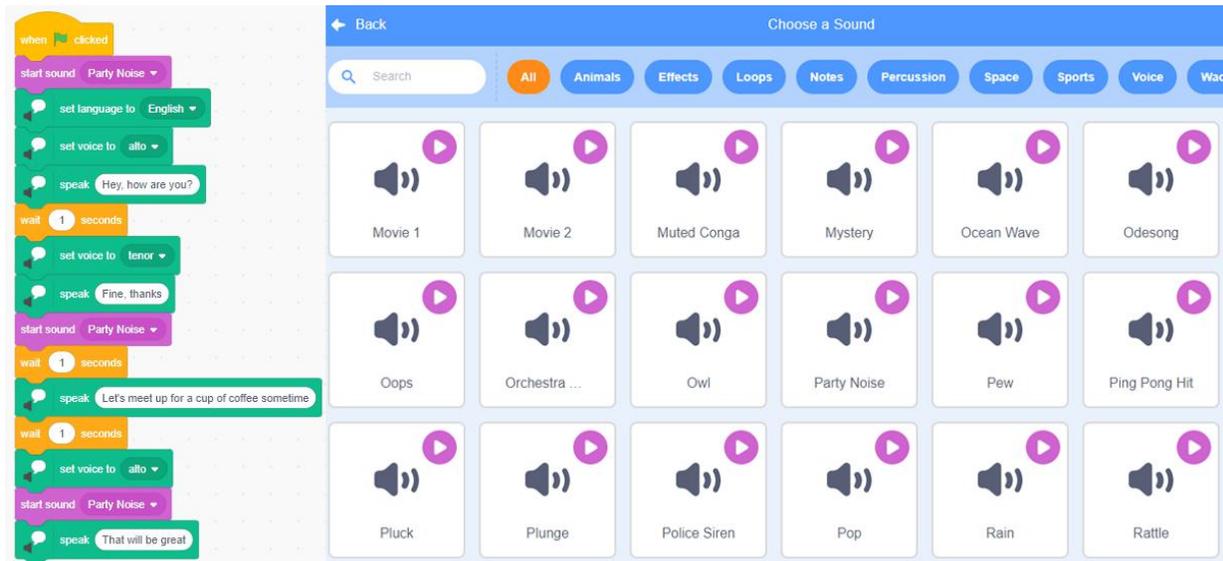


Passo 9: Ou faça o seu próprio filme, conte uma história...

Combine-o com outros clips da galeria de som para tornar a cena mais realista.



<http://erasmus-artie.eu>



Atualmente, existem muitos dispositivos de geração de fala. Os dispositivos geradores de fala permitem que as pessoas "falem" palavras e frases eletronicamente. Os dispositivos geradores de fala são dispositivos electrónicos portáteis que reproduzem palavras ou frases quando o utilizador toca num interruptor ou carrega em botões ou teclas. Alguns dispositivos "falam" palavras à medida que estas são digitadas no teclado. Os dispositivos geradores de fala permitem às pessoas que não podem utilizar a linguagem falada "falar" eletronicamente. Os dispositivos geradores de fala têm sido utilizados para ajudar as crianças autistas a comunicar desde a década de 1990.



## CONCLUSÃO

Conversão de texto em fala (Text To Speech - TTS)  
é um tipo de tecnologia que lê texto digital em voz alta.



## Plano de aula 11

**Duração: 90 minutos**

### Tópicos

---

Interpretar os resultados do algoritmo de objectos controlados por voz

### Objetivos

---

Utilização prática de objectos comandados por voz

### Resultados

---

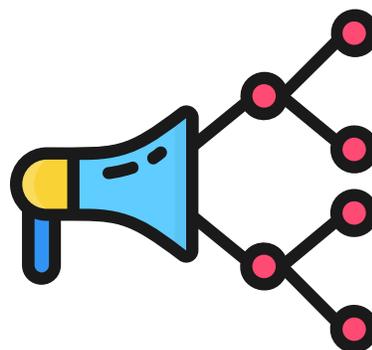
Explorando as possibilidades da extensão Speech to Text no Scratch

## Projeto com objeto comandado por voz

*Pergunte aos seus alunos se um objeto pode ser controlado por voz. É possível se beneficiar de alguma forma de objetos controlados por voz?*

### Apresentação do objetivo da aula:

Compreender o algoritmo de objectos controlados por voz e a sua utilização através de um projeto prático.



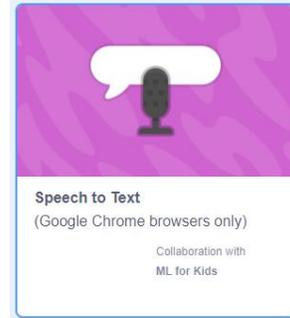


# PARTE PRINCIPAL

Através de um projeto, o professor instrui os alunos sobre os comandos e as competências do Scratch, bem como sobre a forma de treinar um modelo para converter a fala em texto.

Passo 1: Abra o seu navegador Chrome e acesse a:  
<https://machinelearningforkids.co.uk/scratch3/>

Passo 2: Carregar a extensão Speech to text  
(STT - apenas para o navegador Google Chrome)

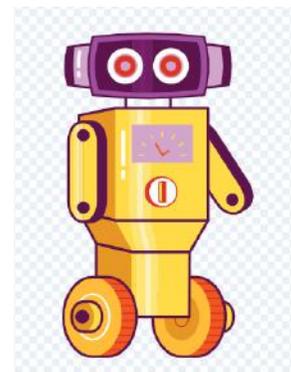


Passo 3: Apagar o ator Gato clicando no ícone do caixote do lixo do ator

Passo 4: Descarregar maze.png de:  
<https://drive.google.com/file/d/11YBBhQclhVfHYMWeLkhgAYKSwfv33pT5/view?usp=sharing> e carregá-lo no Scratch como sprite personalizado



Passo 5: Na galeria de actores, escolher Retro Robot e utilizar o segundo fato (Retro Robot b)



Passo 6: Na galeria de actores, escolha o botão Home





Passo 7: Faça 3 variáveis (para todos os sprites):

- gameover (mostra como o jogo terminou)
- xm (x posição do robô)
- ym (posição y do robô)

Passo 8: Faça uma lista e mude seu nome para: endtalk (exibe a mensagem de fim de jogo, dependendo de como o jogo terminou)

Etapa 9: mude para o sprite do labirinto e comece a codificar. Os primeiros blocos definem a posição, o tamanho e a visibilidade do labirinto. O loop eterno a seguir é um verificador de colisão. Em caso de colisão, a variável gameover é definida como 1.

Etapa 10: mude para o sprite do botão Home. O código é quase o mesmo, exceto que em caso de colisão um som pop é reproduzido e a variável game over é definida como 2.

```
when green flag clicked
  go to x: 0 y: 0
  set size to 120 %
  show
  forever loop
    if touching Retro Robot ? then
      set gameover to 1
```

```
when green flag clicked
  set size to 100 %
  go to x: 195 y: -120
  show
  forever loop
    if touching Retro Robot ? then
      play sound pop until done
      set gameover to 2
```





Passo 11: O código principal é atribuído ao código do robô Retro e há 5 threads que iniciam simultaneamente.

11.1 O código do sinalizador (inicial) define as variáveis, entradas da lista (e exclusão de todas as entradas anteriores), posição e visibilidade. No final há um bloco de escuta (da extensão STT) no loop.

11.2 Os próximos 4 threads são acionados por reconhecimento de fala e cada comando de voz reconhecido é tratado por uma sequência de blocos.

Se nenhuma colisão for detectada (gameover é 0), o robô se move em uma direção específica (alterando o valor de xm ou ym e movendo o robô para a posição calculada).

Se a colisão for detectada (gameover é 1 ou 2), a mensagem de game over (da lista) será exibida e o programa será interrompido.

```
when clicked
delete all of endtalk
insert Game over! Robot crashed! at 1 of endtalk
insert You won! Robot returned to base! at 2 of endtalk
set gameover to 0
set xm to -200
set ym to 0
go to x: xm y: ym
go to front layer
set size to 30 %
show
forever
listen and wait
```





```
when I hear up
  if (gameover = 0) then
    change ym by 20
    glide 0.25 secs to x: xm y: ym
  else
    start sound computer beeps1
    say item gameover of endtalk for 3 seconds
    stop all

when I hear left
  if (gameover = 0) then
    change xm by -20
    glide 0.25 secs to x: xm y: ym
  else
    start sound computer beeps1
    say item gameover of endtalk for 3 seconds
    stop all

when I hear right
  if (gameover = 0) then
    change xm by 20
    glide 0.25 secs to x: xm y: ym
  else
    start sound computer beeps1
    say item gameover of endtalk for 3 seconds
    stop all

when I hear down
  if (gameover = 0) then
    change ym by -20
    glide 0.25 secs to x: xm y: ym
  else
    start sound computer beeps1
    say item gameover of endtalk for 3 seconds
    stop all
```





Passo 12: Tente utilizar os comandos de voz da sua língua materna e veja como funciona.

Discuta como a ortografia (in)correcta afecta o reconhecimento de voz.

Ativar (verificar) os resultados do reconhecimento de voz.



Os robôs de seguimento de objectos, se puderem ser controlados de forma inteligente através da voz, podem ser de grande ajuda para as pessoas com deficiências físicas. É utilizado um sistema de reconhecimento de voz para reconhecer um conjunto de comandos predefinidos, como avançar, recuar, esquerda, direita e rotação num determinado ângulo de viragem. O robô navega de acordo com o sinal de comando de voz, ao mesmo tempo que segue o objeto desejado. O processamento do sinal de comando de voz é efetuado em tempo real, utilizando um servidor na nuvem que o converte em texto. O texto do sinal de comando é então transferido para o robô através da rede Bluetooth para controlar o seu acionamento diferencial. O protótipo de robô inteligente é composto por três subsistemas: o sistema de reconhecimento de voz, o sistema de localização de objectos e o sistema de controlo do movimento baseado no acionamento diferencial. A precisão e a eficiência do sistema de reconhecimento de voz são examinadas através de um conjunto de experiências. O efeito de fatores como o ruído e a distância, etc., é examinado, com resultados encorajadores. O protótipo do robô consegue reconhecer os comandos de voz dentro de um raio de alcance Bluetooth, ou seja, 10 m. São também discutidas possíveis extensões que podem levar a uma vasta gama de outras aplicações.

## CONCLUSÃO

Podemos controlar objetos em um programa de computador com a nossa voz.





## Plano de aula 12

**Duração: 90 minutos**

### Tópicos

Introdução ao hardware - microcontrolador, câmara e controlador de motor

### Objetivos

Os alunos aprendem sobre o hardware a ser utilizado no projeto principal

### Resultados

Familiarizar-se com o hardware utilizado no projeto principal

## Introdução ao hardware - microcontrolador, câmara e controlador de motor

Uma parte deste currículo está relacionada com dispositivos físicos reais com capacidades de IA. Para esse efeito, a Associação Croata de Robótica desenvolveu um pequeno robô móvel para implementar os conhecimentos de programação das aulas anteriores. Em primeiro lugar, vamos considerar em que deve consistir o nosso robot. É evidente que as peças mecânicas - caixa, rodas e motores DC - serão utilizadas como acionamento. E as partes electrónicas? Tal como a maioria dos seres vivos, o nosso robô vai precisar do cérebro (microcontrolador), dos olhos (câmara de IA) e dos nervos/impulsos (motor DC). Vamos dar uma olhadela a essas partes.

Existem 2 opções para construir um robot:

1. Kit de robô Maqueen Plus + câmara HuskyLens AI (fácil - adequado para principiantes)

micro: O Maqueen Plus é um robô educativo STEM avançado para Micro:bit. Poderoso e inteligente, este robô micro:bit tem uma gestão de energia optimizada e uma fonte de alimentação de alta capacidade: pode ser totalmente compatível com o sensor de visão AI HuskyLens, tornando-o uma ferramenta de ensino acessível.

O Micro:Maqueen Plus possui um chassis grande e estável, funcionalidades integradas e várias portas de expansão. Não só é adequado para o ensino na sala de aula, como também pode ser utilizado para sessões de formação prolongadas depois das aulas e competições de robôs.

Arduino UNO + Motor driver + Câmara HuskyLens AI (mais complexo - apenas para utilizadores experientes)



<http://erasmus-artie.eu>

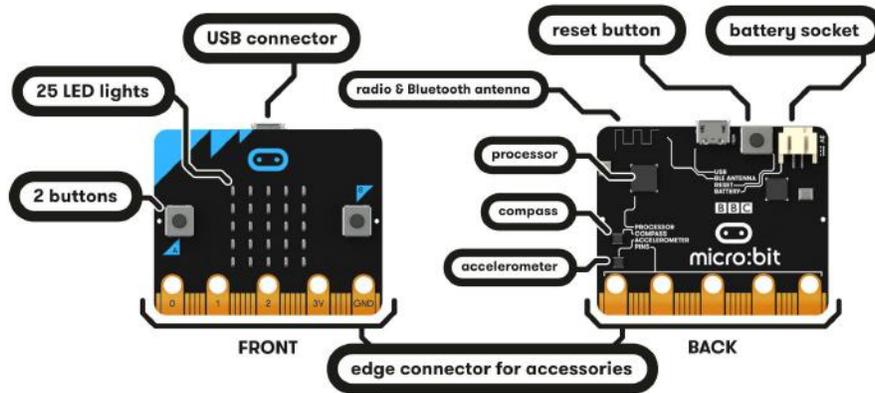


# PARTE PRINCIPAL

## Cérebro 1 - micro:bit

Um "cérebro" no mundo da eletrónica é chamado processador ou, neste caso específico, microcontrolador.

O micro:bit é um microcontrolador de bolso, fácil de usar, potente e económico, concebido para ensinar crianças e principiantes a programar, permitindo-lhes concretizar facilmente as suas ideias em jogos digitais DIY, projectos interactivos e robótica.

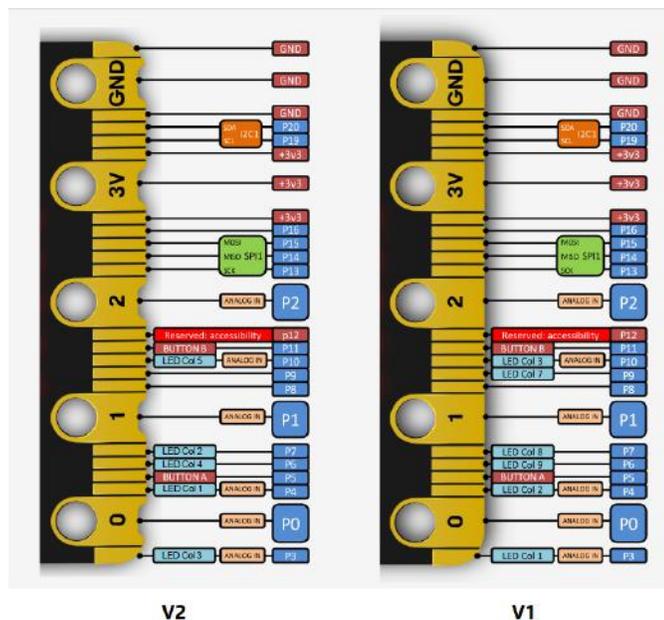


Graças às suas portas de E/S externas e suporte de hardware, o Micro:bit é adequado para a aprendizagem e desenvolvimento de vários robots.

## Layout e especificações técnicas do Micro:bit

Existem duas versões do micro:bit disponíveis no mercado - consulte este artigo para saber qual é a sua:

<https://kitronik.co.uk/blogs/resources/explore-micro-bit-v1-microbit-v2-differences>





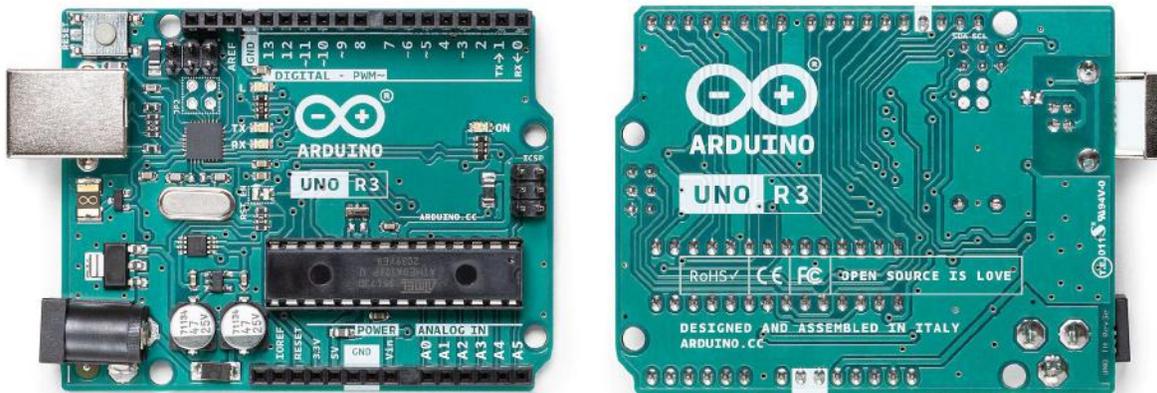
- Uma pequena placa de tamanho semelhante ao de um cartão de crédito (4cm x 5cm)
- Módulos integrados, como o acelerómetro, a bússola e o módulo Bluetooth® Smart
- Um microcontrolador de bolso
- Uma matriz de LEDs 5x5 (também suporta deteção de luz)
- Sensores de luz e temperatura e outros dispositivos de deteção comuns

Equipado com o processador M0 da ARM, o micro:bit pode executar a maioria das funções fundamentais de um robot.

Introdução ao micro: [https://www.youtube.com/watch?v=POkeI\\_2NXMo](https://www.youtube.com/watch?v=POkeI_2NXMo)

## Cérebro 2 - Arduino UNO

O Arduino UNO é o microcontrolador mais popular do mundo, com uma enorme comunidade online e muitos projetos disponíveis na Internet. Existem muitos clones do UNO no mercado. Um clone significa que a arquitetura central, em termos de eletrónica, é semelhante à do Arduino UNO, mas são introduzidas algumas modificações para proporcionarem funcionalidades adicionais à placa. Estas modificações são concebidas e desenvolvidas especialmente para os estudantes aprenderem a codificar e a arquitetura dos microcontroladores. Vejamos o Arduino UNO.



Cada um dos 14 pinos digitais do Uno pode ser utilizado como entrada ou saída e funciona a 5 volts. Cada pino pode fornecer ou receber 20 mA como condição de funcionamento recomendada e tem uma resistência pull-up interna (desligada por defeito) de 20-50k ohm. Um máximo de 40 mA é o valor que não deve ser excedido em qualquer pino de E/S para evitar danos permanentes no microcontrolador. Além disso, alguns pinos têm funções especializadas:

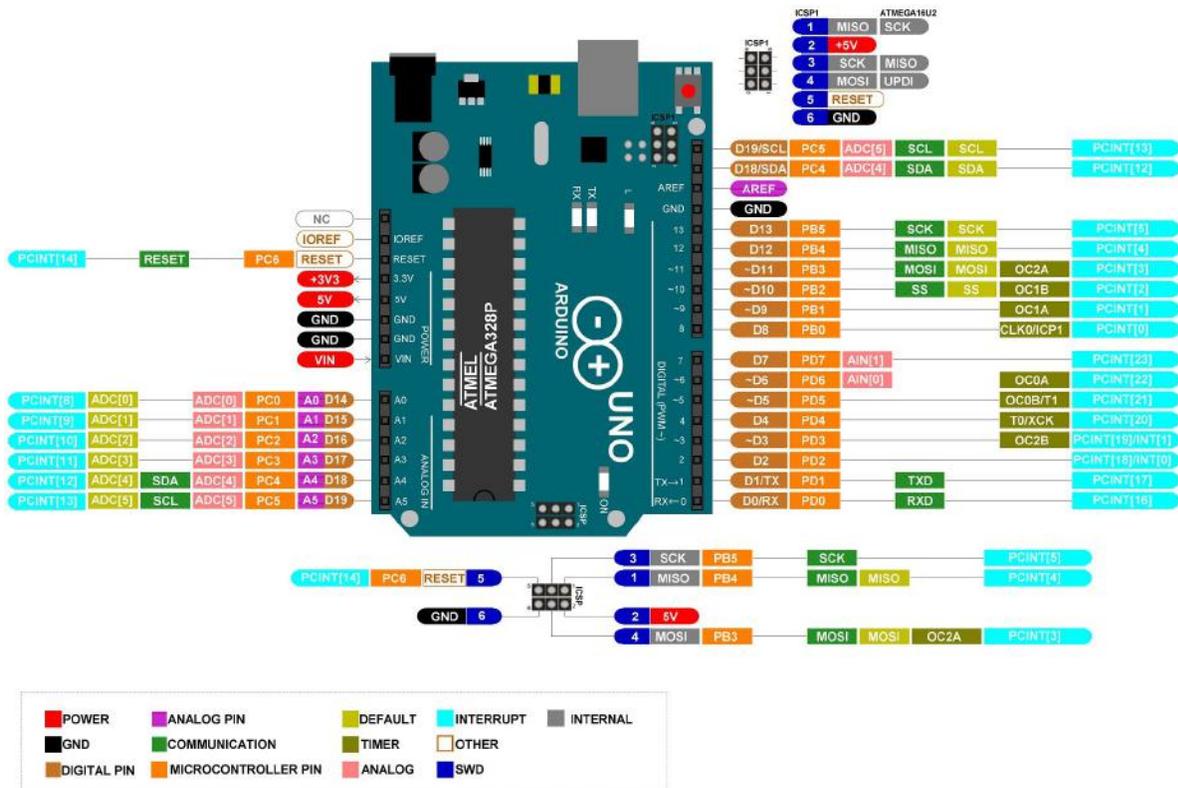
- Série: 0 (RX) e 1 (TX). Utilizados para receber (RX) e transmitir (TX) dados de série TTL. Estes pinos estão ligados aos pinos correspondentes do chip ATmega8U2 USB-to-TTL Serial.
- Interrupções externas: 2 e 3. Estes pinos podem ser configurados para acionar uma interrupção num valor baixo, numa borda ascendente ou descendente, ou numa alteração de valor.
- PWM: 3, 5, 6, 9, 10 e 11. Fornecem a função de saída PWM de 8 bits.



- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos suportam a comunicação SPI utilizando a biblioteca SPI.
- LED: 13. Existe um LED incorporado acionado pelo pino digital 13. Quando o pino tem um valor ALTO, o LED está ligado, quando o pino está BAIXO, está desligado.
- TWI: A4 ou pino SDA e A5 ou pino SCL. Suporta comunicação TWI usando a biblioteca Wire.
- O Uno tem 6 entradas analógicas, rotuladas de A0 a A5, cada uma das quais fornece 10 bits de resolução (ou seja, 1024 valores diferentes). Por defeito, medem de terra a 5 volts, embora seja possível alterar o limite superior da sua gama usando o pino AREF. Existem alguns outros pinos na placa:
- AREF. Tensão de referência para as entradas analógicas.
- Reset. Colocar esta linha em LOW para reiniciar o microcontrolador. Tipicamente usado para adicionar um botão de reset a shields que bloqueiam o que está na placa

### Esquema e especificações técnicas do Arduino UNO

Placa Arduino UNO com os pinos explicados:





MICROCONTROLLER	ATmega328P
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	14 (of which 6 provide PWM output)
PWM DIGITAL I/O PINS	6
ANALOG INPUT PINS	6
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA
FLASH MEMORY	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	68.6 mm
WIDTH	53.4 mm
-----	--

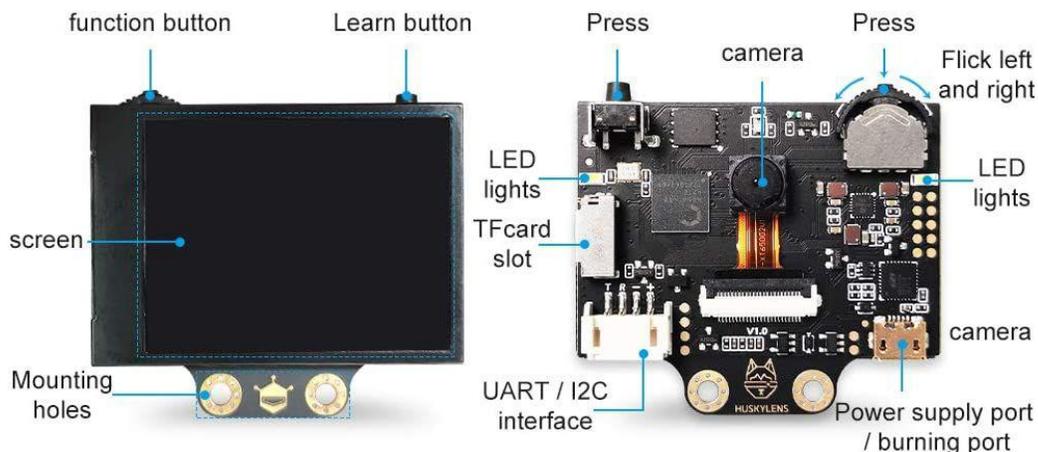
A alimentação principal do Arduino UNO é efectuada através de uma ligação USB, mas pode ser alimentada por baterias ligadas através do conector de pinos (pinos VIN e GND).

Introdução ao Arduino UNO: <https://www.youtube.com/watch?v=bniUEctJkeU>

Olhos - HuskyLens Kendryte K210 (todos os cenários)

O nosso robô móvel será capaz de detetar o ambiente com esta câmara. O HuskyLens é um sensor de visão computacional de IA fácil de utilizar com 7 funções integradas: reconhecimento facial, seguimento de objectos, reconhecimento de objectos, seguimento de linhas, reconhecimento de cores, reconhecimento de etiquetas e classificação de objectos.

Pode ser facilmente ligado a qualquer dispositivo compatível com Arduino/Arduino e micro:bit. Agora, pode realizar projectos muito criativos, mesmo sem o conhecimento de algoritmos complexos de aprendizagem automática.



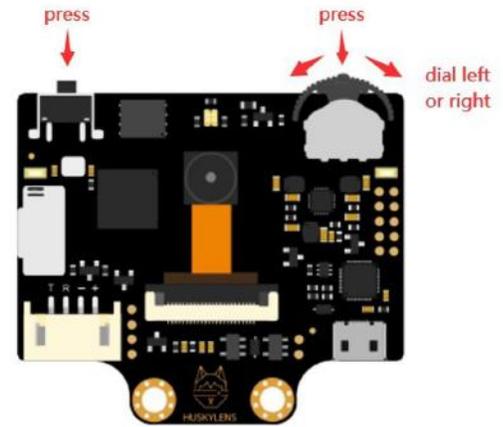


Existem dois botões no HuskyLens, o botão de função e o botão de aprendizagem. As operações básicas destes dois botões são mostradas a seguir:

Disque o "botão de função" para a esquerda ou direita para alternar diferentes funções.

Pressione rapidamente o "botão Aprendizagem" para aprender o objeto especificado; pressione longamente o "botão de aprendizagem" para aprender continuamente o objeto especificado de diferentes ângulos e distâncias; se o HuskyLens já aprendeu o objeto antes, pressione rapidamente o "botão Aprender" para esquecê-lo.

Pressione longamente o "botão de função" para entrar no menu de segundo nível (configuração de parâmetros) na função atual. Disque para a esquerda, direita ou pressione rapidamente o "botão de função" para definir os parâmetros relacionados.

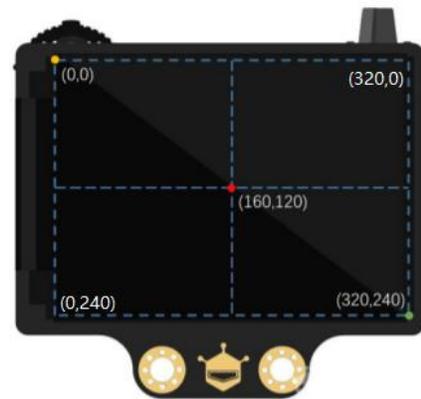


### Sistema de coordenadas

Quando o HuskyLens detecta um objeto, o alvo será automaticamente selecionado por uma moldura colorida na tela. As coordenadas da posição key da moldura colorida são atribuídas de acordo com o seguinte sistema de coordenadas. Após obter as coordenadas da porta UART/I2C, você poderá saber a posição do objeto. Formato:x,y)

Funções:

- 1) Reconhecimento facial
- 2) Rastreamento de objetos
- 3) Reconhecimento de objetos
- 4) Rastreamento de linha
- 5) Reconhecimento de cores
- 6) Reconhecimento de tags
- 7) Classificação de objetos



### Instruções de cores

Em cada função, as definições de cores de a moldura e o símbolo "+" no centro da tela são todas iguais, o que ajuda você conhece o status atual do HuskyLens.

Colour	Status
From orange to yellow, then from yellow to orange	Have not learned the object yet but ready to learn
Yellow	Learning the new object
Blue	Have learned the object and recognized it



O indicador LED RGB é usado para indicar o status da função de reconhecimento facial. Suas cores são definidas a seguir.

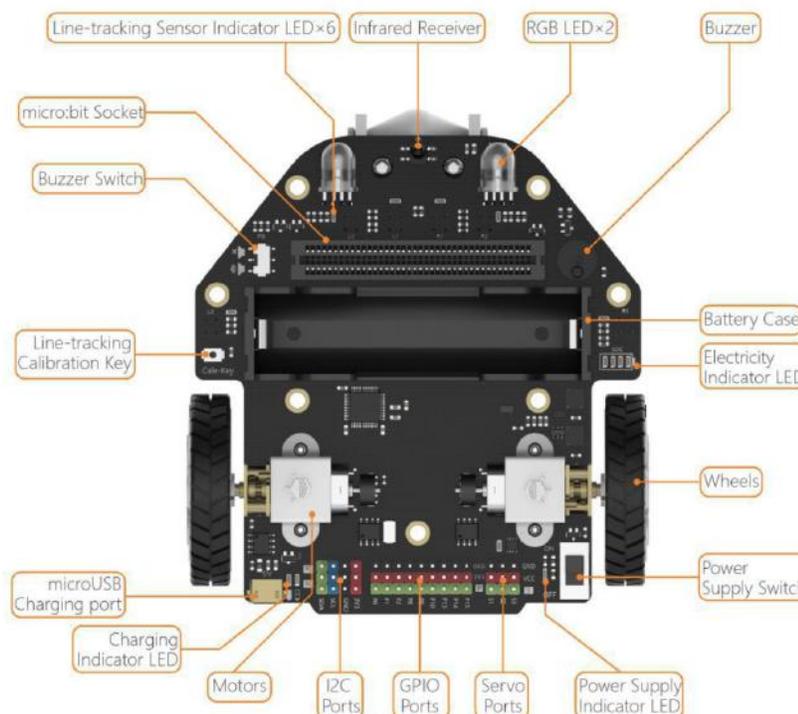
Colour	Status
Blue	Have not learned the face yet, but detected the face
Yellow	Learning the new face
Green	Have learned the face and recognized it

Manual completo: [https://wiki.dfrobot.com/HUSKYLENS\\_V1.0\\_SKU\\_SEN0305\\_SEN0336](https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336)

### Nervos/impulsos/músculos 1 - Placa do robô Maqueen Plus

O Maqueen Plus é um robô educativo inteligente e programável, concebido para principiantes. Pode ser programado com as plataformas de programação Mind+ e MakeCode. Está otimizado com uma melhor gestão da energia e uma fonte de alimentação de maior capacidade. É ideal para utilização com o sensor de visão HuskyLens AI e vem com um chassi maior e mais estável. Também inclui mais funções incorporadas e mais portas de expansão. É adequado para o ensino na sala de aula, para exercícios prolongados depois das aulas e para competições de robôs.

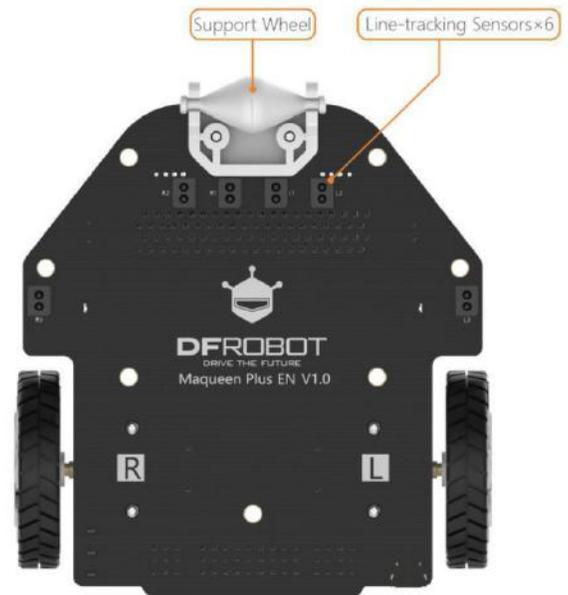
Eis os principais componentes da placa Maqueen Plus:





### Especificações técnicas do Maqueen plus

- Fonte de alimentação: Bateria de lítio 3.7V-18650
- Tensão de carregamento: 5V
- Corrente de carregamento: 900mA, Tempo de carregamento: 4h
- Indicador de bateria: 4 LEDs
- Motor de acionamento: Motor N20 260 rpm
- Buzzer \* 1
- RGB-LED \* 2
- Portas de expansão GPIO: P0 P1 P2 P8 P12 P13 P14 P15 P16
- Portas de expansão I2C \* 3
- Portas de expansão de servo \* 3
- Sensores de Seguimento de Linha \* 6
- Dados de saída do sensor de rastreamento de linha: analógico + digital
- Calibração do sensor de linha: suporte
- Sensor de receção de infravermelhos \* 1
- Sensor ultrassónico: URM10
- Placa metálica superior \* 1
- Ligações roscadas M3 \* 12
- Tamanho do mapa: 50cm\*50cm
- Dimensão: 107x100mm (4.21 x3.94")



Maqueen Plus está agora acessível através da plataforma de programação Mind+. Mind+ é uma plataforma de programação gráfica baseada em Scratch 3.0 da DFRobot, que suporta python, Arduino e outras plataformas de programação. Atualmente, a Mind+ tem sido aplicada a todos os tipos de sensores, módulos e produtos educativos relacionados.

Saiba mais sobre o Maqueen Plus:

[https://github.com/DFRobot/Maqueen\\_Plus\\_Basic\\_Tutorial/blob/master/MBT0021-EN-Maqueen%20Plus%20Basic%20Tutorial.pdf](https://github.com/DFRobot/Maqueen_Plus_Basic_Tutorial/blob/master/MBT0021-EN-Maqueen%20Plus%20Basic%20Tutorial.pdf)

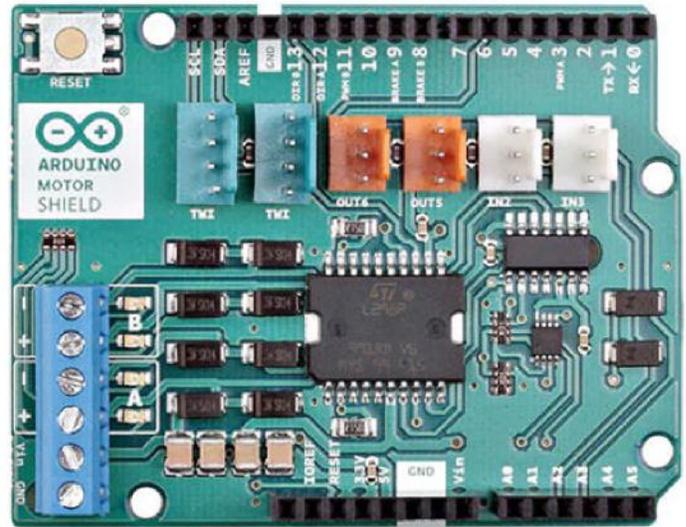
No momento em que este cenário está a ser concluído, a versão V2 deste robô está disponível no mercado.



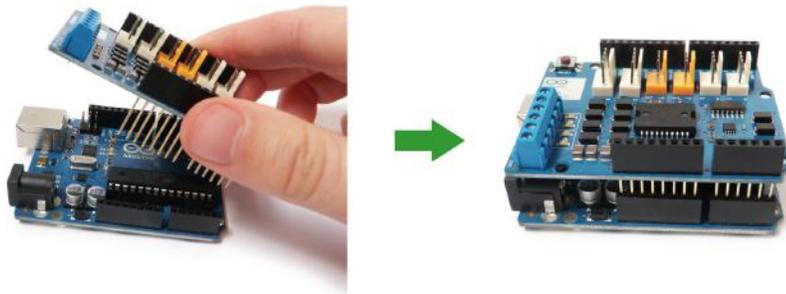


### Nervos/impulsos/músculos 2 - Arduino Motor Shield Rev 3

Existem várias maneiras de controlar um motor DC; talvez a mais fácil seja simplesmente aplicar-lhe energia. As primeiras invenções que usavam o motor DC funcionavam assim: adicione uma fonte de energia e o motor começará a girar, mude a polaridade e você muda a direção. Mas se quisermos fazer um pouco mais do que apenas fazer um motor girar a toda a velocidade em duas direções, precisamos de um circuito de controlo do motor. Mais especificamente, o driver de ponte completa dupla L298P (chip), que podemos encontrar no Motor Shield Rev3.



A proteção do motor é empilhável no Arduino UNO, o que significa que não é necessário ligar a ligação ao microcontrolador. Basta ligar os pinos da proteção (macho) aos pinos do microcontrolador (fêmea).



O shield do motor tem 2 canais, o que permite o controlo de dois motores DC ou 1 motor de passo.

Tem também 6 cabeçalhos para a ligação de entradas, saídas e linhas de comunicação do Tinkerkit. O uso desses pinos é um pouco limitado e, portanto, não é abordado neste tutorial.

With an external power supply, the motor shield can safely supply up to 12V and 2A per motor channel (or 4A to a single channel).

Com uma fonte de alimentação externa, o escudo do motor pode fornecer com segurança até 12V e 2A por canal do motor (ou 4A para um único canal).

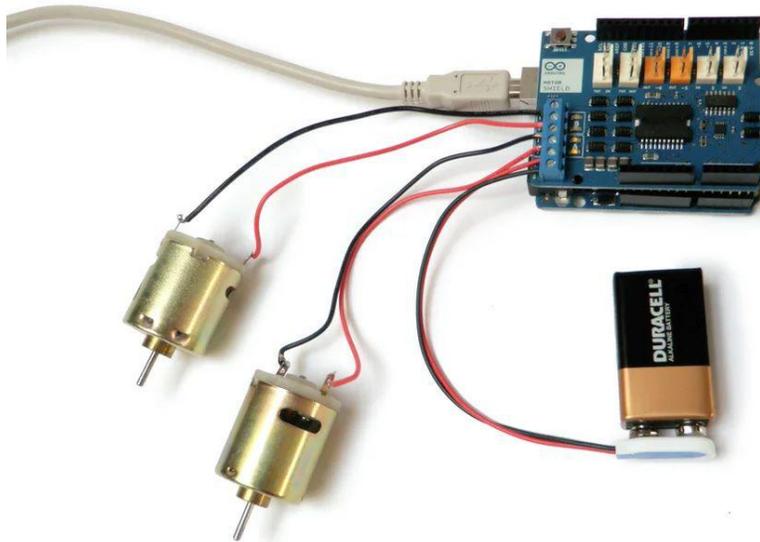
Existem pinos no Arduino que estão sempre a ser utilizados pelo shield. Ao endereçar estes pinos, pode seleccionar um canal do motor para iniciar, especificar a direção do motor (polaridade), definir a velocidade do motor (PWM), parar e ligar o motor e monitorizar a absorção de cada canal.

A repartição dos pinos é a seguinte:



Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1

E, finalmente, este é o aspeto que deve ter quando ligado:



<http://erasmus-artie.eu>

Saiba mais nesta página:

<https://www.instructables.com/Arduino-Motor-Shield-Tutorial/>

## CONCLUSÃO

Semelhante à maioria dos seres vivos, nosso robô ARTIEBot possui cérebro (microcontrolador), olhos (câmera AI) e nervos/impulsos (controlador motor).



## Plano de aula 13

Duração: 90 minutos

### Tópicos

Ligar o microcontrolador à câmara e ao PC e começar a trabalhar

### Objetivos

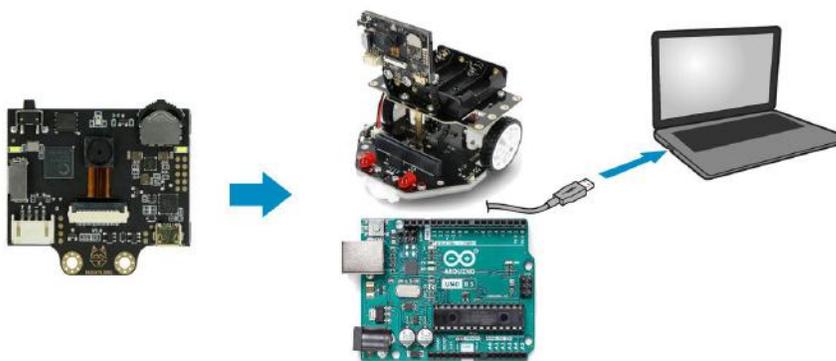
Os alunos ligam o microcontrolador à câmara e ao PC e vêem-no a funcionar

### Resultados

compreender o funcionamento da câmara

## Ligação do microcontrolador à câmara e ao PC

Antes de começarmos a programar, temos de ligar a câmara ao microcontrolador (micro:bit ou Arduino UNO) e o microcontrolador ao computador.



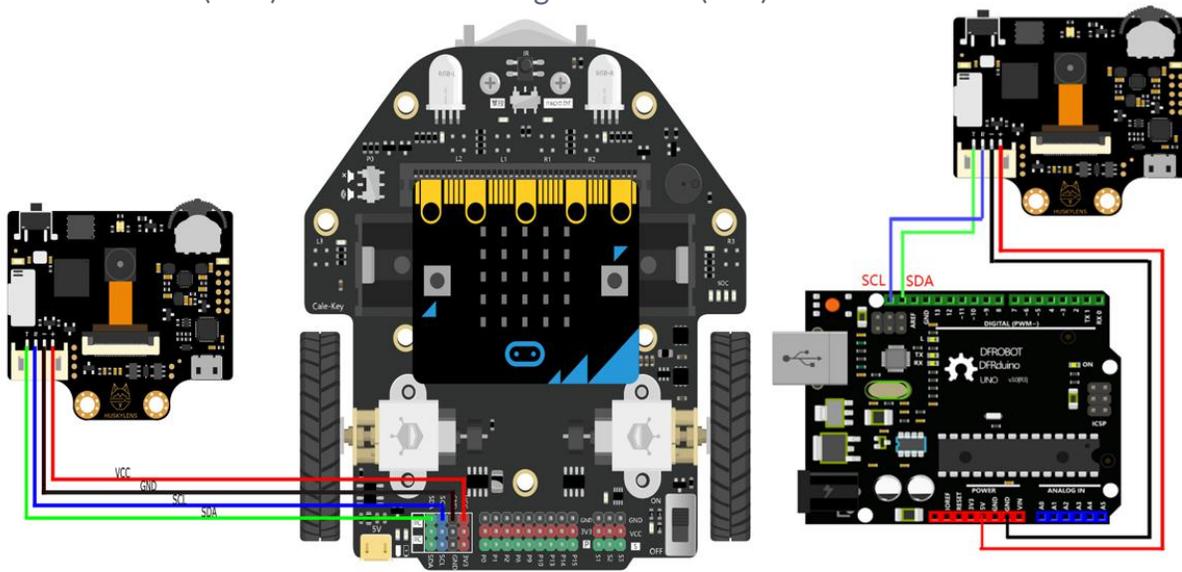
Option 1: Camera > I2Connection > micro:bit/Maqueen plus > USB cable > Laptop or PC

Option 2: Camera > I2Connection > Arduino UNO > USB cable > Laptop or PC



# PARTE PRINCIPAL

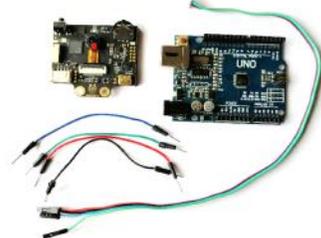
Não há muito a dizer sobre a ligação USB, mas provavelmente nunca ouviu falar de I2C. O bus I2C (Inter-Integrated Circuit) é uma interface de série de dois fios originalmente desenvolvida pela Phillips Corporation para utilização em produtos de consumo. Segue uma hierarquia mestre/escravo, em que o mestre é definido como o dispositivo que faz o relógio do barramento, endereça os escravos e escreve ou lê dados de e para os registos nos escravos. Os escravos são dispositivos que respondem apenas quando interrogados pelo mestre, através do seu endereço único. O barramento I2C utiliza apenas duas linhas bidireccionais, a linha de dados em série (SDA) e uma linha de relógio em série (SCL).



Option 1 (Maqueen plus)

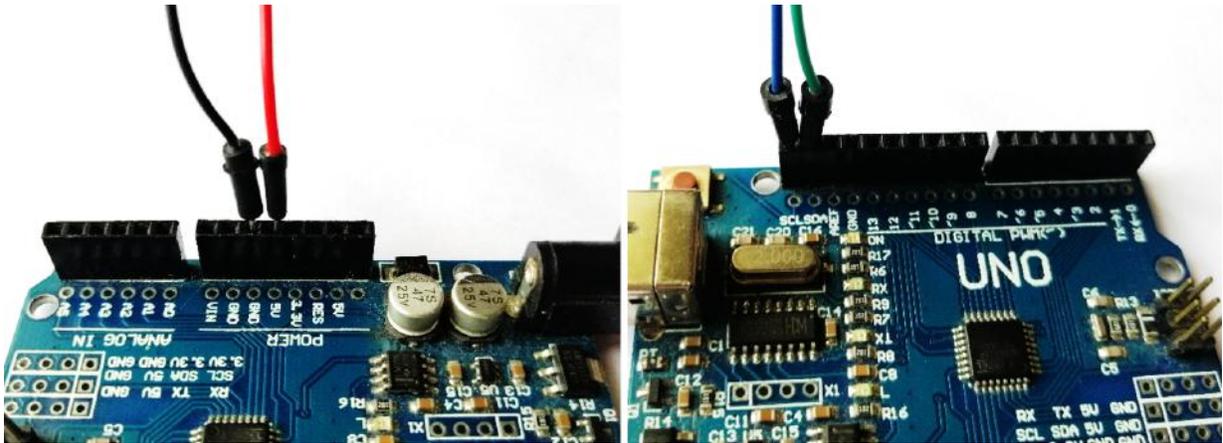
Option 2 (Arduino UNO)

Na opção 1 - a solução é muito simples e basta fazer corresponder as cores dos fios e do conetor.



Na Opção 2, será um pouco complicado ligar o cabo da câmara e o Arduino UNO, porque ambos os conectores são do tipo fêmea, pelo que precisamos de 4 fios de ligação macho para macho, que podem ser comprados em quase todas as lojas de eletrónica que vendem peças de bricolage.

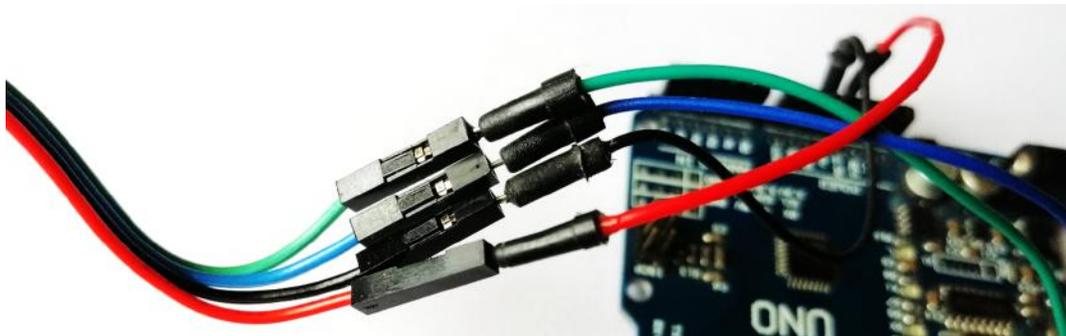




Escolha 4 jumpers (vermelho, preto, verde e azul) para corresponder aos pinos correspondentes da câmara.

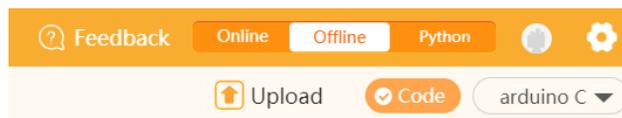
Primeiro, ligue o jumper vermelho ao pino de 5V do Arduino UNO, o jumper preto ao GND, o jumper azul ao SCL e o jumper verde ao pino SDA, como mostram as imagens abaixo.

The only thing left to do is to plug the other side of these jumpers to camera connector. Match the colours of jumpers to camera connector wires (black to black, red to red, etc...).



Só falta ligar o outro lado destes jumpers ao conetor da câmara. Faça corresponder as cores dos jumpers aos fios do conetor da câmara (preto com preto, vermelho com vermelho, etc...).

Vai a: <http://mindplus.cc/download-en.html> e descarrega a versão para o sistema operativo do teu computador. Instala e inicia o Mind+. Primeiro, muda para o modo offline.

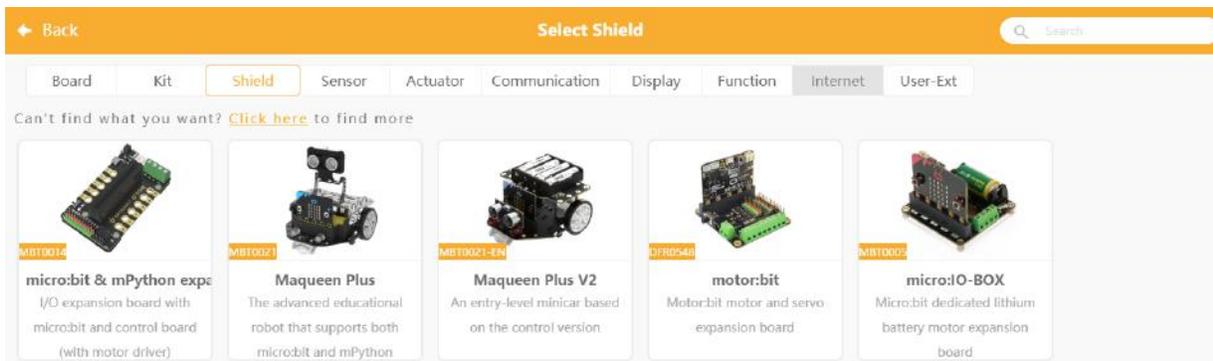




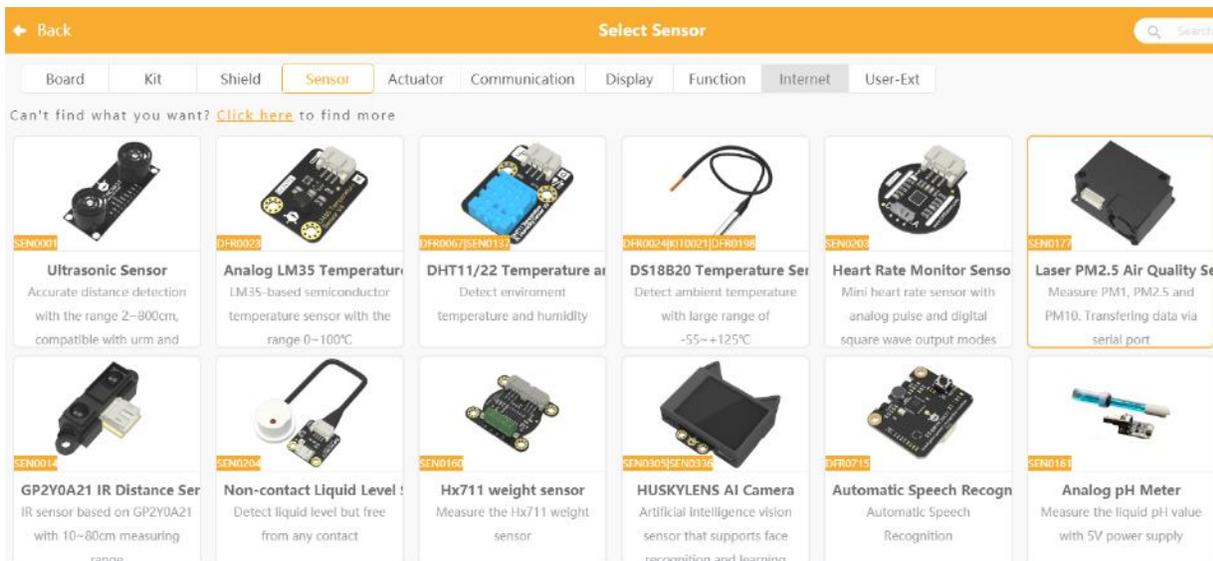
Abrir Extensões e seleccionar Quadro: Opção 1) micro:bit (se trabalhar com) Maqueen plus  
Opção 2) Arduino UNO



Apenas para a Opção 1) Mudar para o separador Escudo e seleccionar Maqueen Plus ou Maqueen Plus V2 (dependendo da sua versão)



Ambas as opções 1) e 2)  
Mudar para o separador Sensor e seleccionar o sensor - HUSKYLENS AI Camera

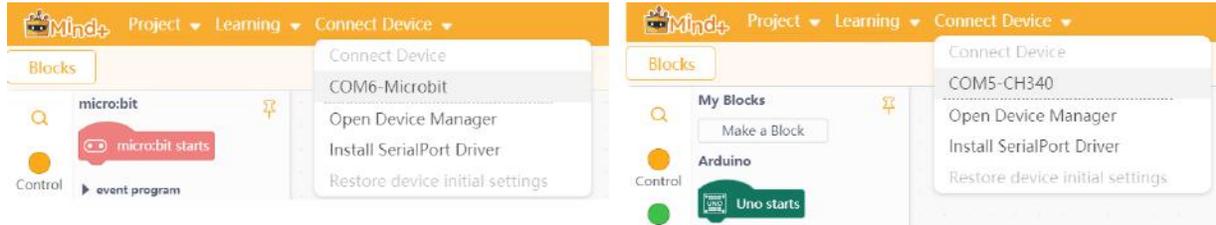


http://erasmus-artie.eu



Depois de seleccionar clique em <- Back e está pronto para usar os blocos Arduino e Sensor. Vamos testar se funciona.

Antes disso, é preciso conectar o dispositivo. Clique em Connect Device e selecione sua porta e dispositivo

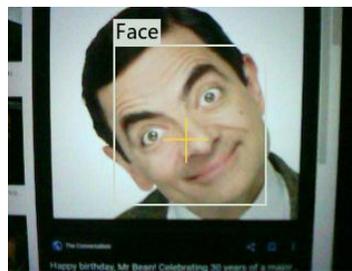


Pegue na câmara e rode o botão de função para a esquerda até aparecer a palavra "Reconhecimento de rosto" na parte superior do ecrã.



### Aprendizagem e deteção

1. Deteção de rostos: Aponte a HuskyLens para qualquer rosto. Quando é detectado um rosto, este é automaticamente seleccionado por uma moldura branca com as palavras "Rosto" no ecrã.

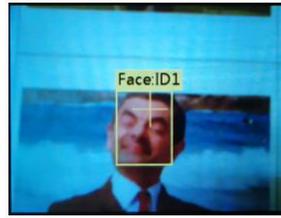


dicas: Se quiser que o HuskyLens aprenda ou reconheça o seu rosto, ou seja, tire uma selfie, não consegue ver o ecrã neste momento, pode determinar o estado de acordo com as diferentes cores do indicador RGB.

2. Aprendizagem de rostos: Aponte o símbolo "+" para um rosto, prima brevemente o "botão de aprendizagem" para aprender o rosto. Se o mesmo rosto for detectado pelo HuskyLens, será apresentada no ecrã uma moldura azul com as palavras "Face: ID0" será apresentado no ecrã, o que indica que o HuskyLens aprendeu o rosto e pode reconhecê-lo agora.



http://erasmus-artie.eu



visitar: : [https://wiki.dfrobot.com/HUSKYLENS\\_V1.0\\_SKU\\_SEN0305\\_SEN0336#target\\_15](https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336#target_15)

Vá para a Mind+ e comece a programar. Após o arranque do micro:bit/Uno, utilizar o pino de inicialização do HuskyLens até ao bloco de sucesso e configurá-lo para I2C, como se mostra na figura abaixo para cada opção.

```

micro:bit starts
  HuskyLens initialize pin until success
  Communication Method: I2C
  Address: 0x32
  SDA(Green): P20
  SCL(Blue): P19

```

```

Uno starts
  HuskyLens initialize pin until success
  Communication Method: I2C
  Address: 0x32
  SDA(Green): A4
  SCL(Blue): A5

```

Em seguida, utilize o algoritmo de comutação do bloco HuskyLens para Reconhecimento facial para ambas as opções.

```

micro:bit starts
  HuskyLens initialize pin until success
  HuskyLens switch algorithm to Face recognition

```

```

Uno starts
  HuskyLens initialize pin until success
  HuskyLens switch algorithm to Face recognition

```

Segue-se o ciclo para sempre que detecta se o rosto na câmara é reconhecido como ID Facial 0. Se o rosto for detectado na Opção 1), verá o rosto no ecrã do micro:bit. Caso contrário, verá o sinal X. Se o rosto for detectado na Opção 2), o LED integrado (D13) liga-se. Caso contrário, o LED desliga-se.

```

micro:bit starts
  HuskyLens initialize pin until success
  HuskyLens switch algorithm to Face recognition
  forever
    HuskyLens request data once and save into the result
    if HuskyLens check if ID 0 frame is on screen from the result? then
      display pattern
    else
      display pattern

```

```

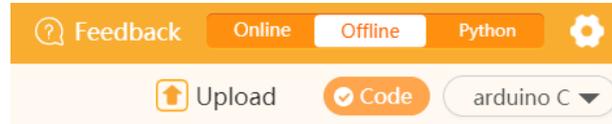
Uno starts
  HuskyLens initialize pin until success
  HuskyLens switch algorithm to Face recognition
  forever
    HuskyLens request data once and save into the result
    if HuskyLens check if ID 0 frame is on screen from the result? then
      digital pin 13 output HIGH
    else
      digital pin 13 output LOW

```





Prima o botão Upload para transferir este código para o micro:bit ou o Arduino UNO.



Depois de o código ser transferido, aponte a câmara para a face "aprendida" e o LED em D2 deve acender-se. Se a afastares do rosto, o D2 desliga-se.

Se funcionar como descrito - está tudo bem e pronto para ser utilizado no nosso robô móvel de IA.

## CONCLUSÃO

HuskyLens é um sensor de visão computacional de IA fácil de usar com 7 funções integradas: reconhecimento facial, rastreamento de objetos, reconhecimento de objetos, rastreamento de linha, reconhecimento de cores, reconhecimento de tags e classificação de objetos.

Através da porta UART/I2C, o HuskyLens pode ser conectado ao Arduino e ao micro:bit para ajudá-lo a fazer projetos muito criativos sem lidar com algoritmos complexos.





## Plano de aula 14

Duração: 90 minutos

### Tópicos

Programar o robot para se mover

### Objetivos

Aprender a programar um robô para se move

### Resultados

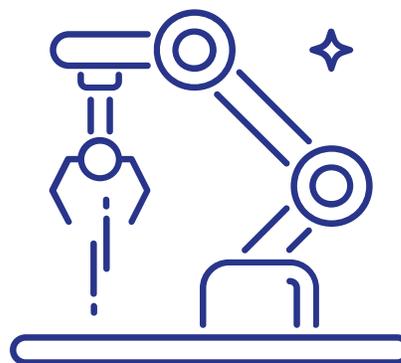
Escrever um programa para o robô mover-se

## Programação de robôs - em movimento

Sejamos realistas, os robôs são fixes. Nesta aula, fornecemos um tutorial passo-a-passo, fácil de seguir (com exemplos de código), que o acompanha através do processo de programação de um robô móvel autônomo básico para se mover.

### Definir o objetivo da aula:

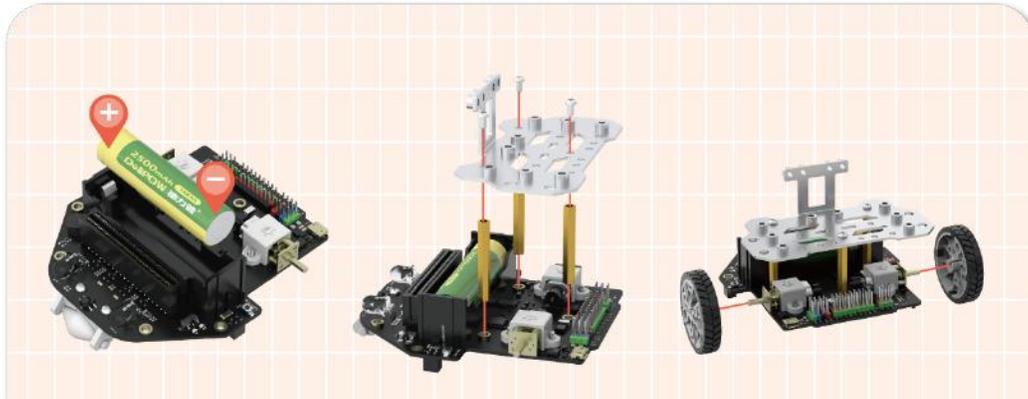
Programar o nosso ARTIEbot pela primeira vez e ver como o podemos movimentar.





# PARTE PRINCIPAL

Opção 1a - Montagem do Robô Maqueen Plus V1 (recomendado para iniciantes)

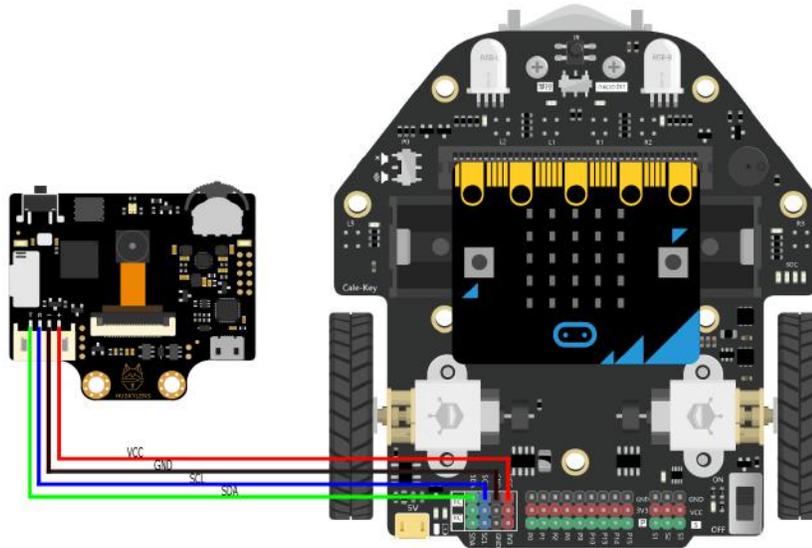


1) insira a bateria 18650 2) instale o suporte 3) prenda as rodas



4) Monte a câmera HuskyLens 5) Coloque a placa micro:bit no slot





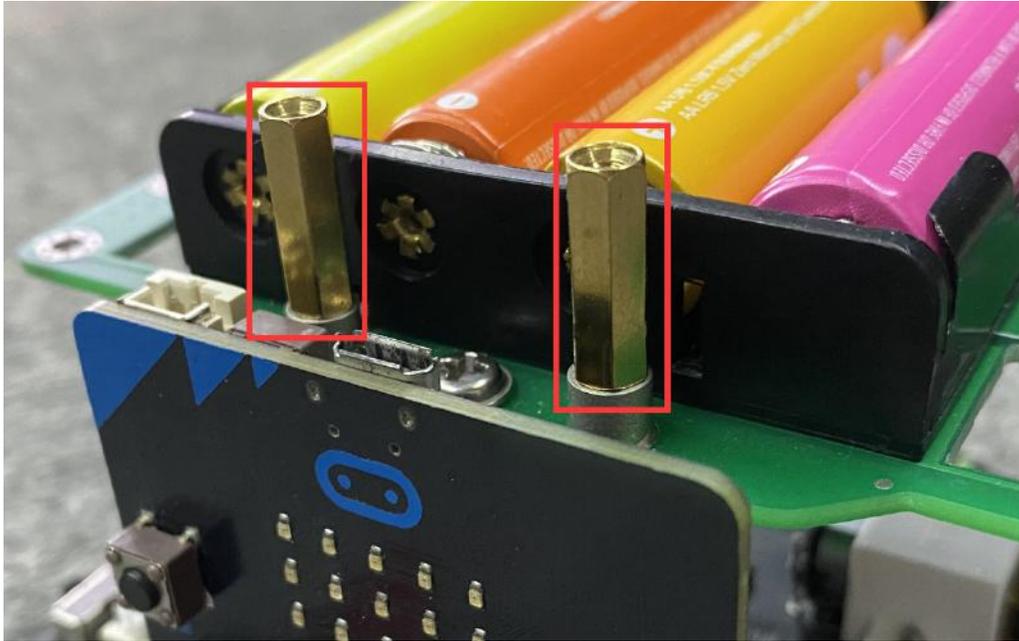
6) Conecte a interface Maqueen Plus à câmera HuskyLens

Para carregar a bateria 18650, use a microporta USB (porta de carregamento) na parte traseira do robô:

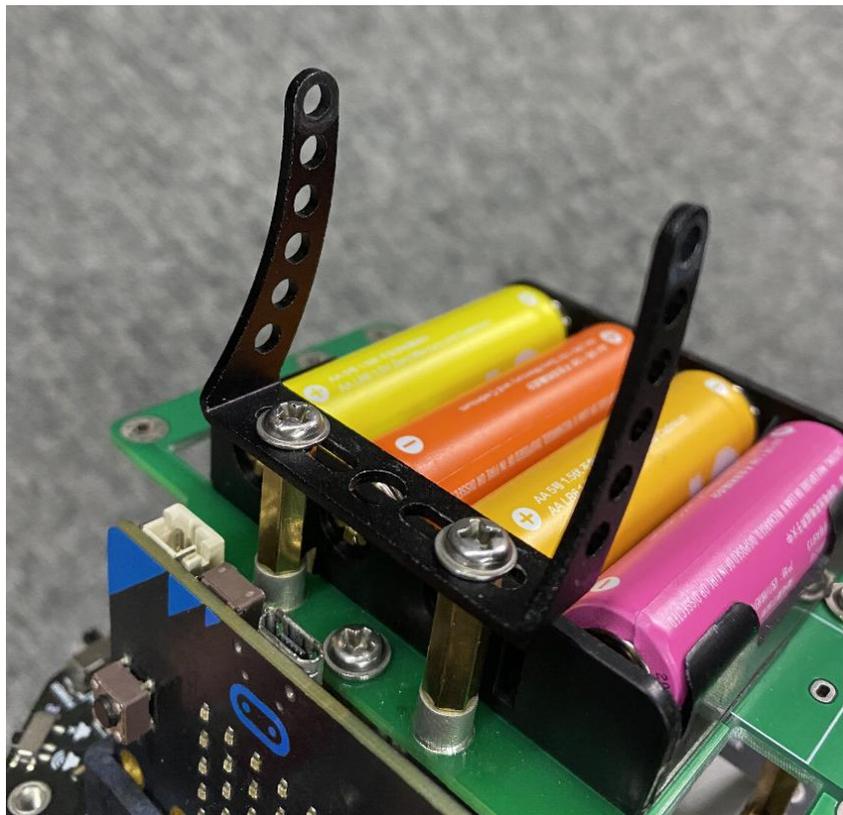


Quando a bateria está totalmente carregada, todos os LEDs acendem. Os LEDs apagarão um por um à medida que a capacidade da bateria diminui gradualmente. Se todas as luzes se apagarem, a bateria precisa ser carregada. Use um carregador de smartphone e conecte-o à porta de carregamento. Observe que a porta de carregamento serve apenas para carregar. Para programar o robô Maqueen Plus - use o micro conector USB no micro:bit.

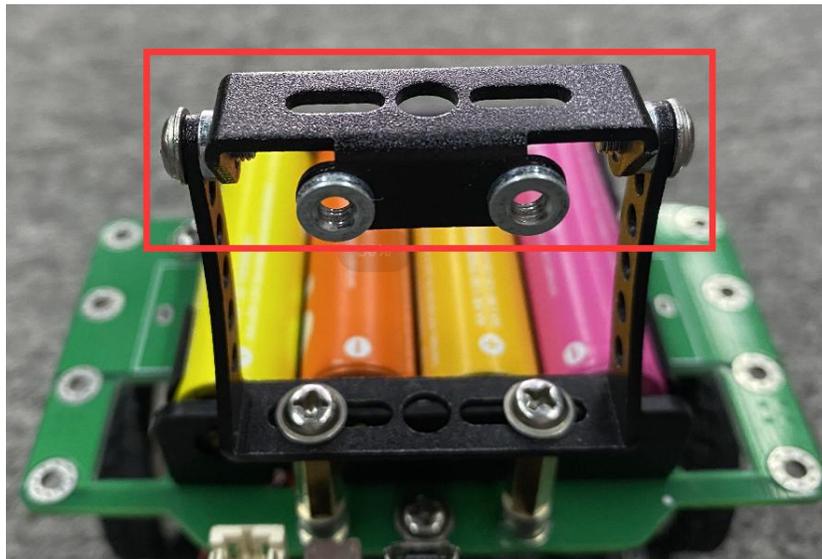
Opção 1b - Montagem do Maqueen Plus Robot V2 (também recomendado para iniciantes)  
O robô Maqueen plus V2 vem pré-montado e você só precisa colocar a câmera HuskyLens nele  
1. Instale dois espaçadores (distâncias) de cobre que estão no kit de materiais nos locais conforme mostrado na figura.



2. Prenda o suporte em arco (suporte e parafusos de montagem fornecidos com HuskyLens) ao poste de cobre com parafusos.



3. Instale o segundo suporte (o suporte e os parafusos de montagem estão incluídos no kit fornecido com a HuskyLens)

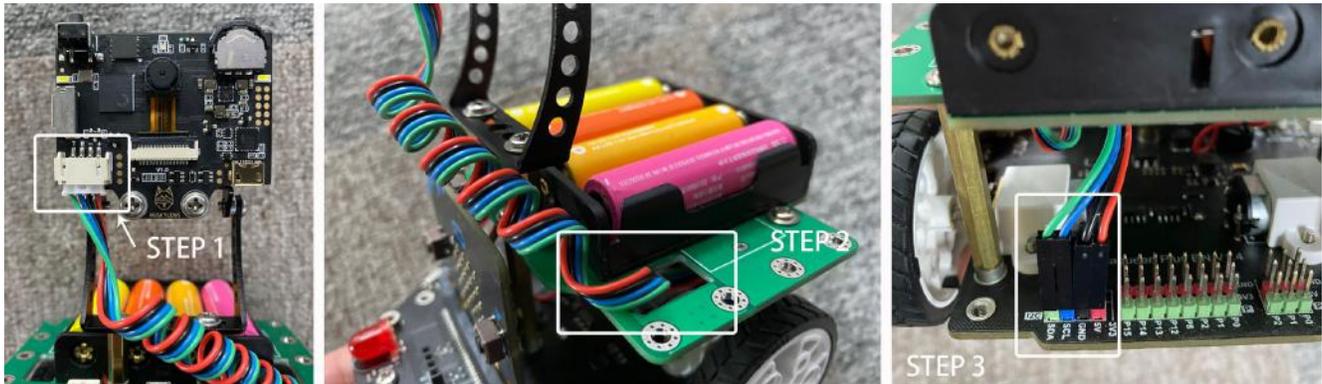


4. Conecte a câmera HuskyLens AI

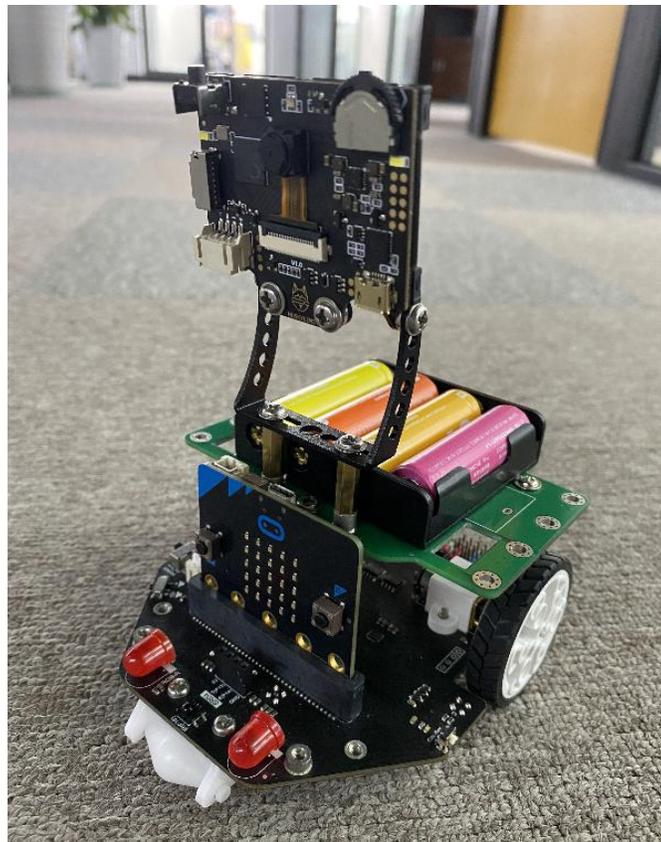




5. Conectando a câmera AI à interface do robô



6. A instalação da câmera foi concluída.





## Opção 2 - Montando um Robô Arduino (somente para usuários DIY experientes)

### Lista de materiais (lista de componentes)

- 1 x Conjunto de materiais para um carro inteligente  
<https://botland.store/chassis-for-robots/7283-chassis-rectangle-2wd-2-wheel-robot-chassis-with-dc-motor-drive-5904422335649.html>
- 1 x conjunto de jumpers  
<https://botland.store/various-wires/1022-connecting-cables-male-male-65pcs.html>
- 1 x suporte de bateria (4 x AA)  
<https://botland.store/battery-holders/173-battery-holder-4-x-aa-r6-5904422329389.html>
- Vários parafusos e porcas  
<https://botland.store/screws-and-nuts/637-screws-nuts-and-washers-set-330pcs-5410329304478.html>
- 1 com Arduino UNO (original ou clone)  
<https://store.arduino.cc/collections/boards/products/arduino-uno-rev3>  
<https://botland.store/arduino-compatible-boards-dfrobot/2683-dfduino-uno-v3-compatible-with-arduino.html>
- 1 x escudo do motor Arduino R3 (escudo do motor Arduino R3)  
<https://store.arduino.cc/collections/shields/products/arduino-motor-shield-rev3>
- 1 x câmera HuskyLens AI  
<https://store.arduino.cc/collections/dfrobot/products/gravity-huskylens-ai-machine-vision-sensor>
- 8 pilhas AA (tente usar recarregáveis)
- Abraçadeiras (se você quiser fixar fios e suportes de bateria)

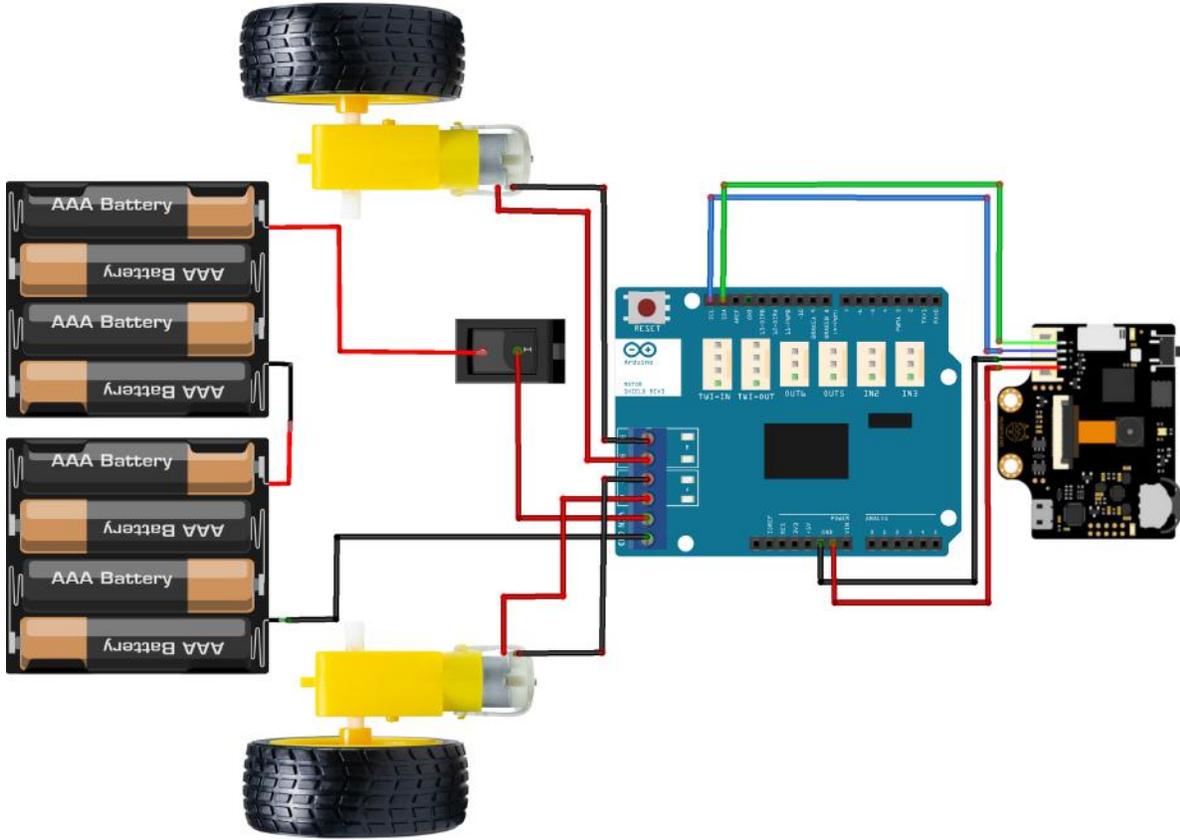
### Ferramentas necessárias

- Chaves de fenda (você precisará de uma plana e uma Phillips)
- Lemilica
- Fita isolante ou tubos termorretráteis
- Alicates de corte

### Esquema

Aqui está o esquema de fiação. Observe que usaremos um suporte de bateria duplo em uma conexão em série para que cada uma das baterias no suporte seja conectada (+ a - e - a +). 8 baterias AA devem fornecer mais tensão para alimentar o consumidor e evitar que o Arduino ou HuskyLens reinicie quando os motores começarem a funcionar. Alternativamente, você pode usar 3 baterias 18650 em uma conexão em série.



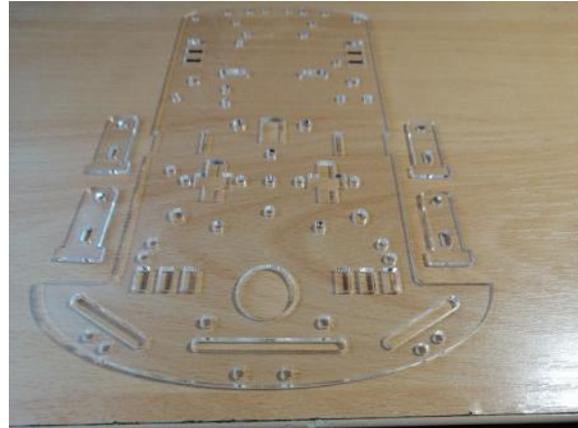


Kit de material para carro inteligente - Abra o saco plástico de peças

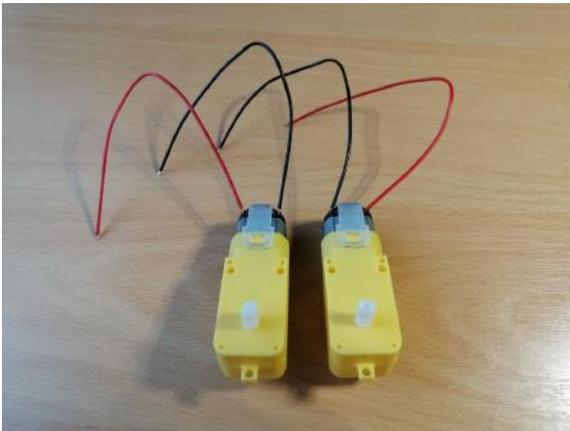




Retire a película protetora da placa e do suporte do motor



Solde os fios (ou jumpers) aos motores



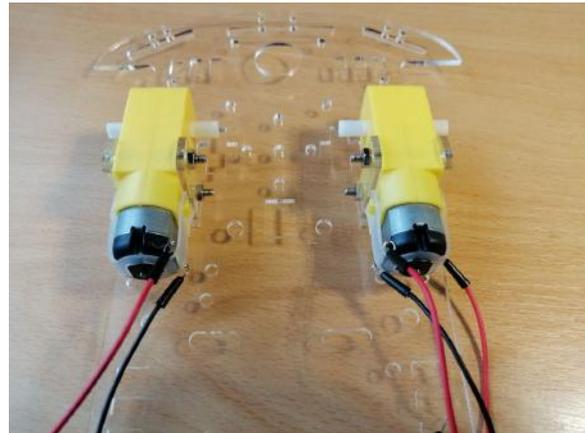
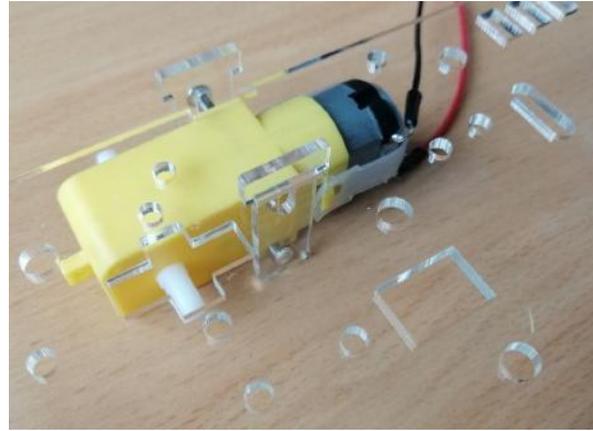
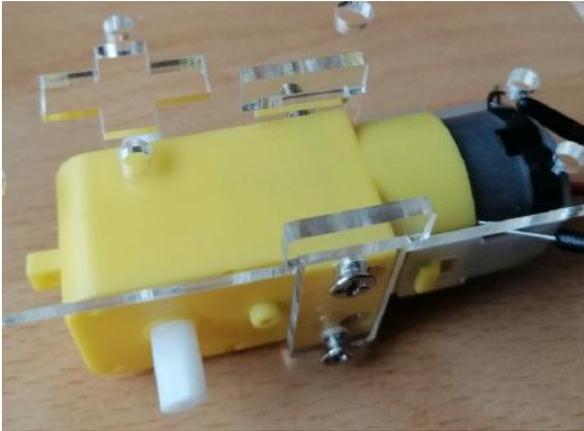
Assista a este pequeno vídeo para ver como isso é feito:  
<https://www.youtube.com/watch?v=xSWKnnvGWBs>

Peça ao seu professor ou pai para ajudá-lo se você não tiver experiência ou ferramentas anteriores.

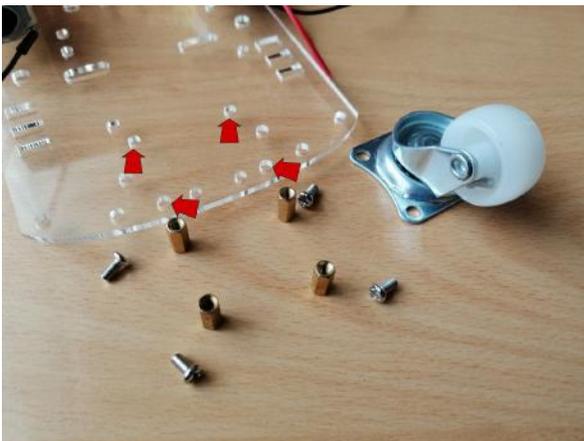


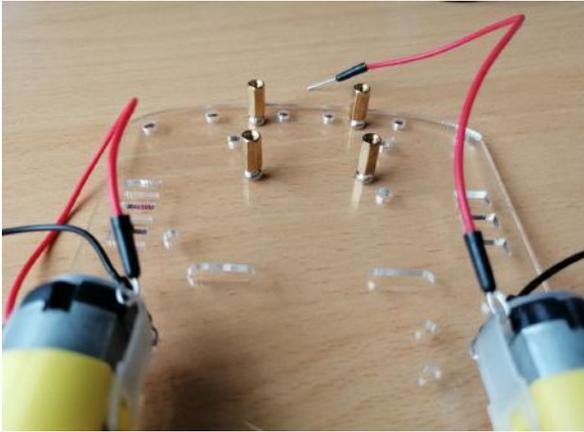


Fixe os motores aos "suportes em T" usando parafusos e porcas

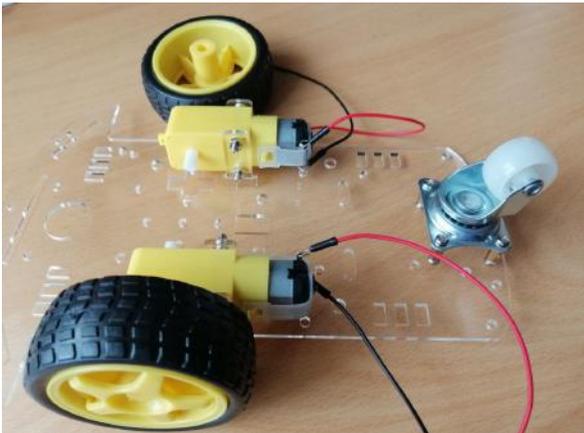


Fixe a roda pequena (traseira) usando parafusos e espaçadores.

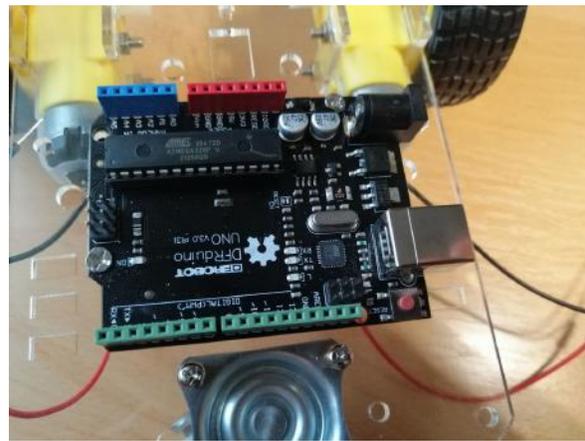
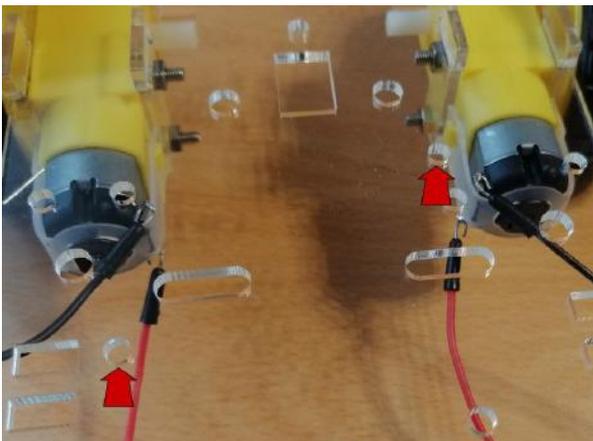


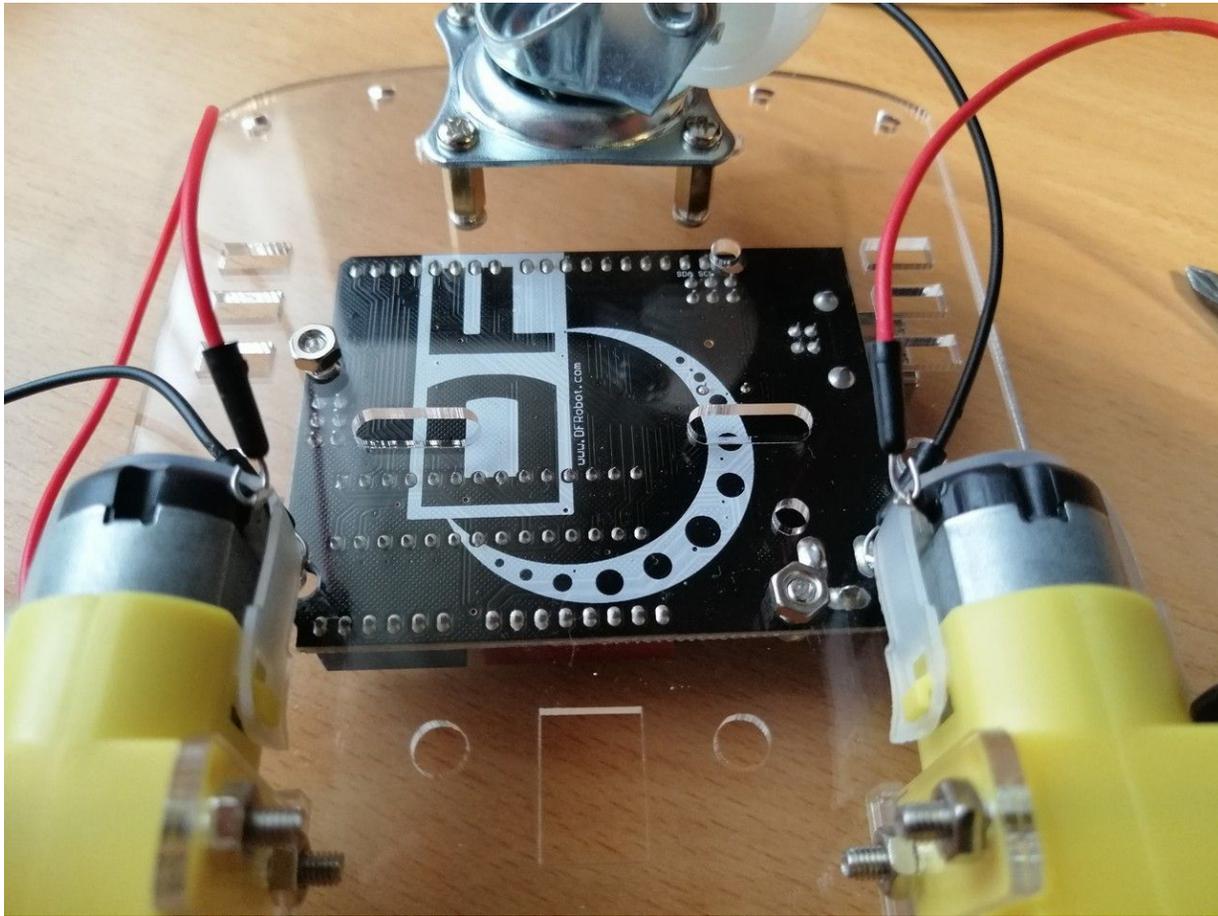


Fixe as rodas aos eixos do motor - observe o formato do eixo e o formato da abertura da roda



Anexe a placa Arduino UNO aos orifícios marcados





<http://erasmus-artie.eu>

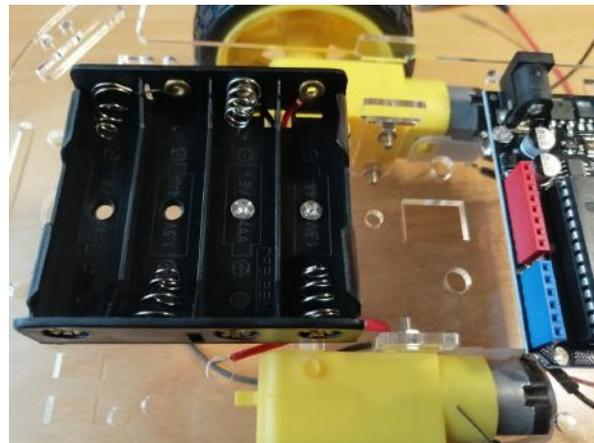
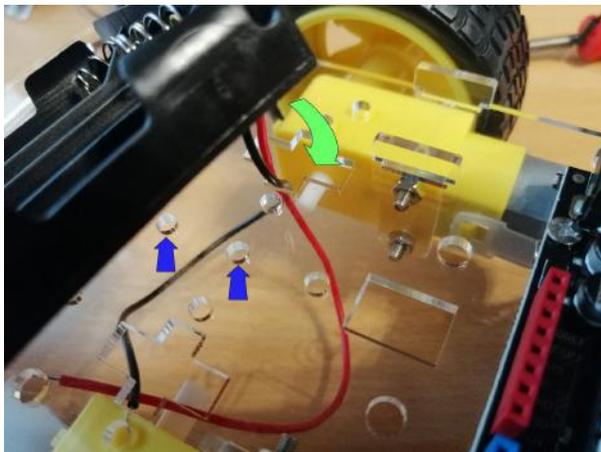
11

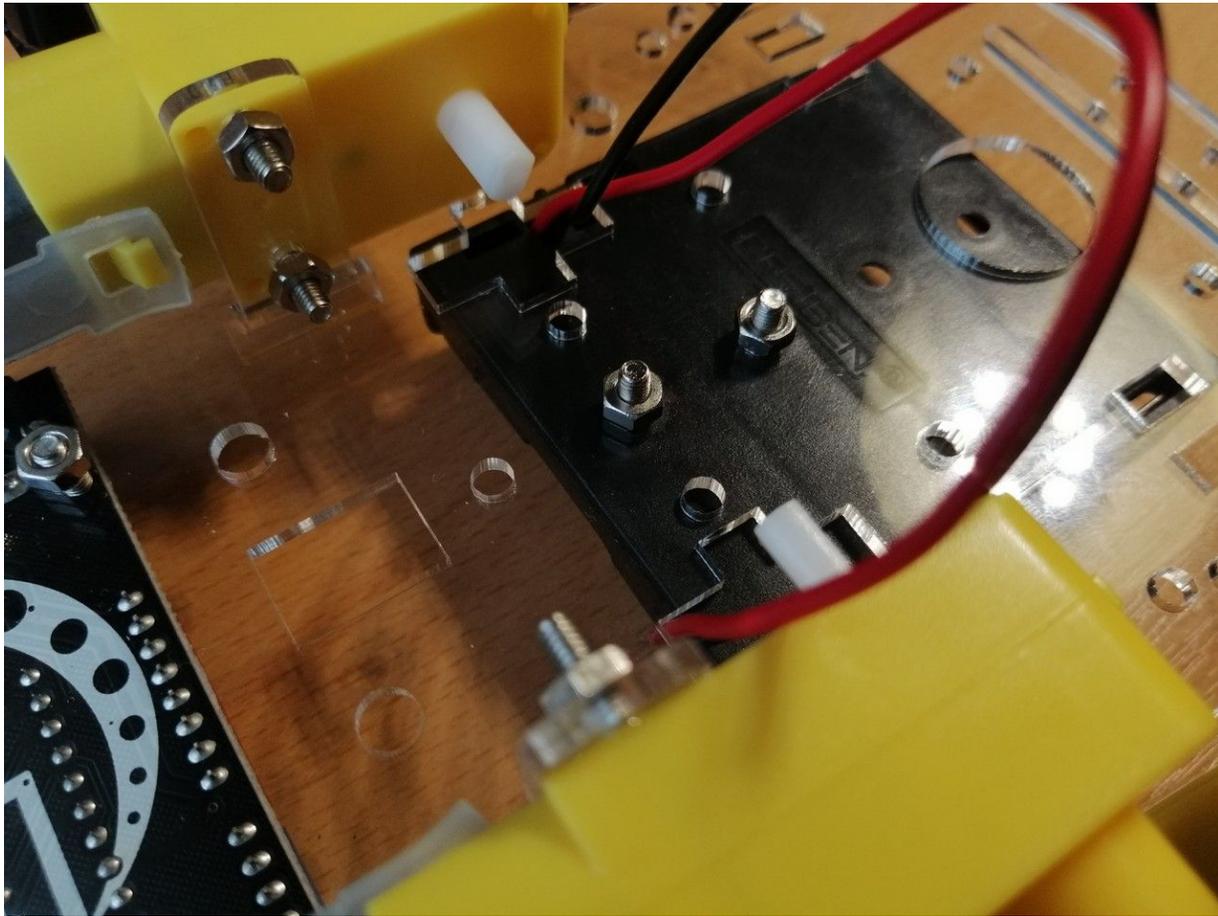
Você pode ver alguns detalhes neste vídeo, embora adotemos uma abordagem um pouco diferente:

<https://www.youtube.com/watch?v=3a-bE1VlaU8>

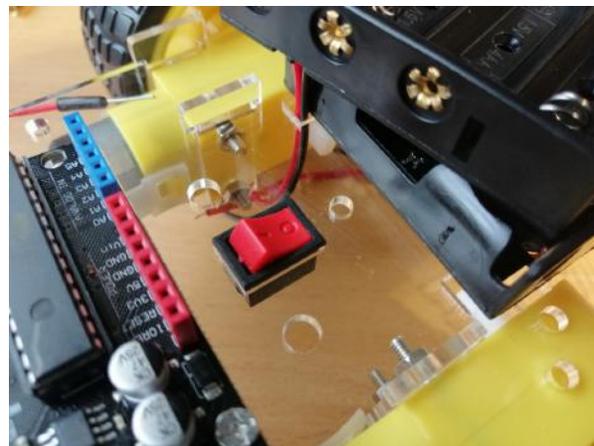
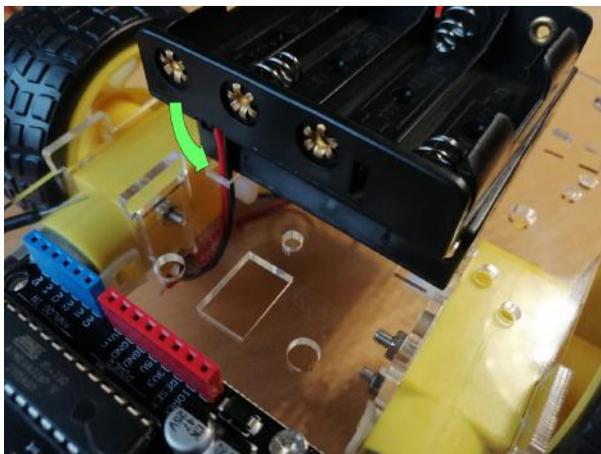
**Etapa 10) Fixe o primeiro suporte da bateria (inferior) usando os parafusos nos orifícios marcados (setas azuis)**

Primeiro, passe os cabos do suporte pela orifício (marcado com uma seta verde)



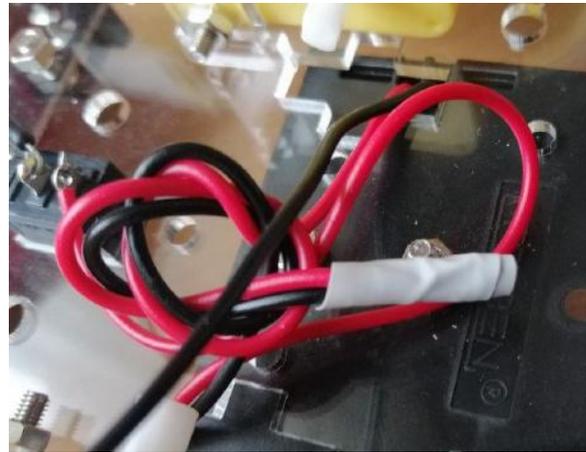
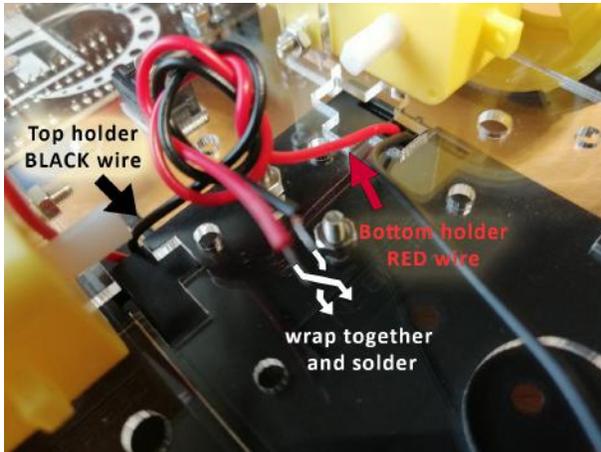


Puxe o cabo do segundo suporte de bateria (superior) através do orifício marcado (seta verde)  
Coloque a chave no orifício retangular localizado entre o Arduino UNO e o suporte da bateria

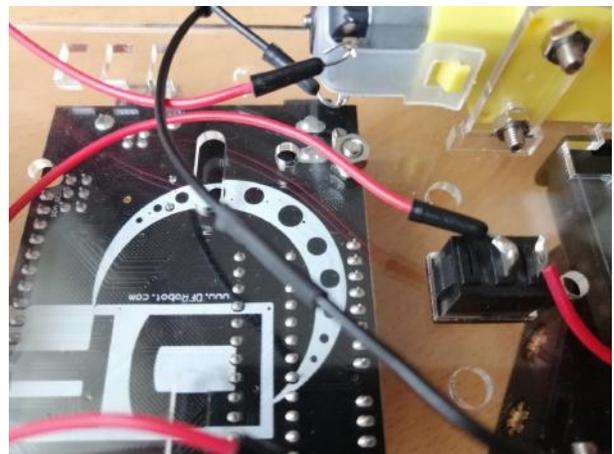
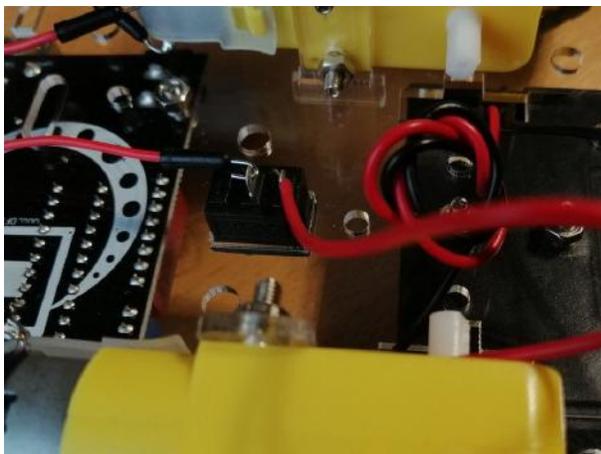




Conecte o fio vermelho do suporte inferior da bateria ao fio preto do suporte superior da bateria  
Isole a junta com tubo termorretrátil ou fita isolante.

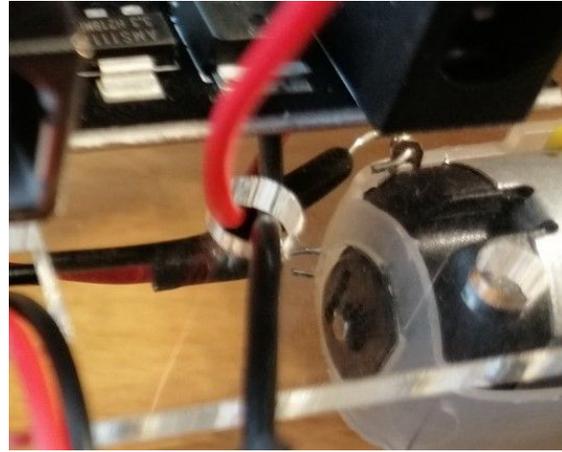
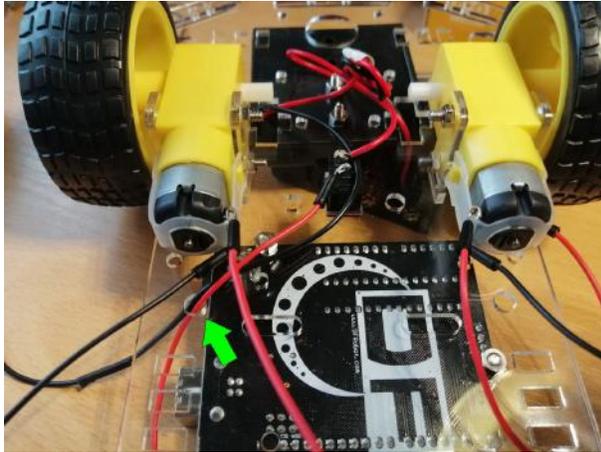


Solde o fio vermelho do suporte inferior ao switch  
Depois disso, solde o jumper (ou fio) vermelho ao segundo contato da chave. Solde o jumper preto ao suporte inferior do fio preto. Isole a junta com tubo termorretrátil ou fita isolante.

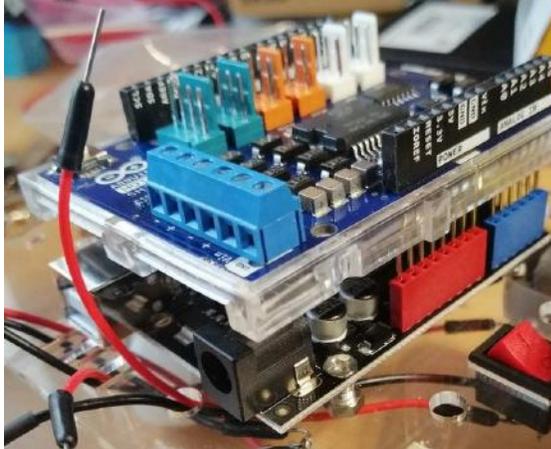




Puxe os fios de alimentação preto e vermelho através do orifício marcado (seta verde)



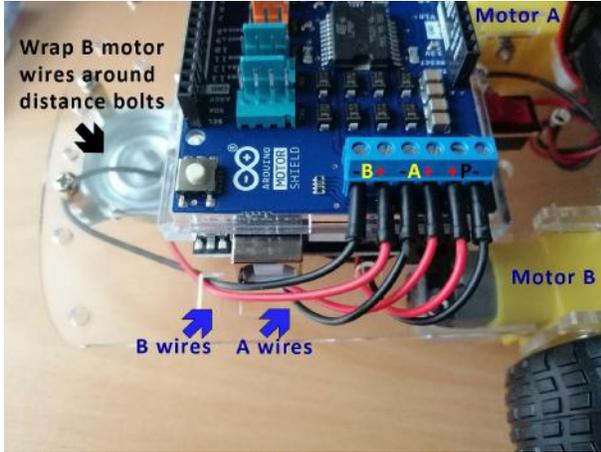
Coloque a blindagem do motor Arduino na placa Arduino UNO  
Alinhe as pernas (pinos) da blindagem e do Arduino UNO e empurre a blindagem para baixo.





Passes os fios do motor pelos orifícios marcados.

Encurte as extremidades dos jumpers com cortadores e conecte-os aos terminais de blindagem dos motores apertando os parafusos conforme indicado (- Motor B + - Motor A ++ Potência P -).



Abra a caixa da câmera HuskyLens AI e retire o kit de materiais

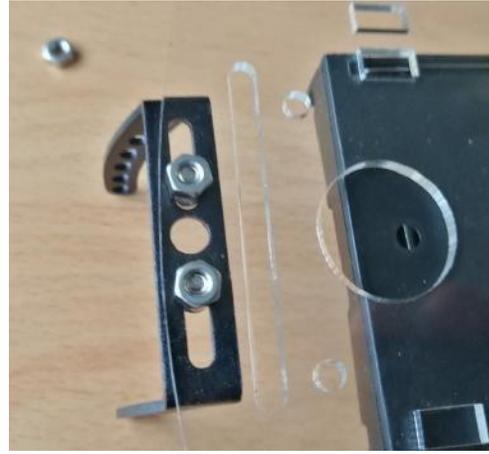


Anexe o suporte menor à câmera - use os parafusos do kit





Anexe o suporte de câmera maior ao corpo do robô



Conecte os dois suportes com parafusos e conecte o cabo da câmera

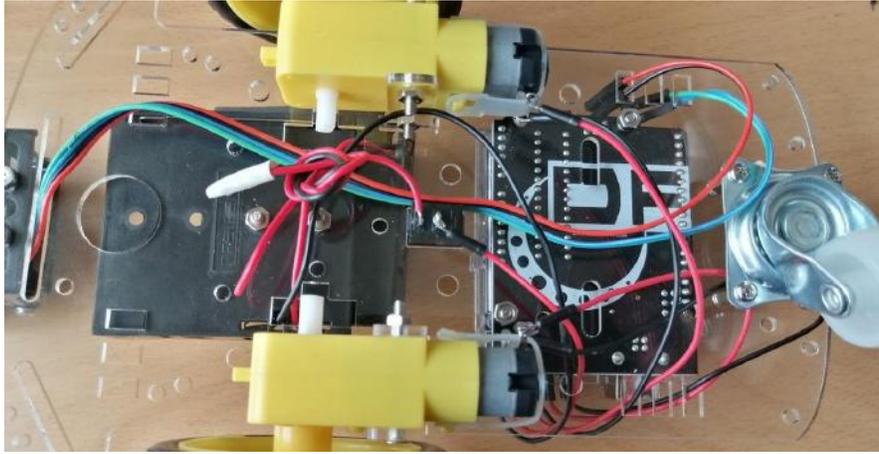


Puxe o cabo da câmera pelo orifício

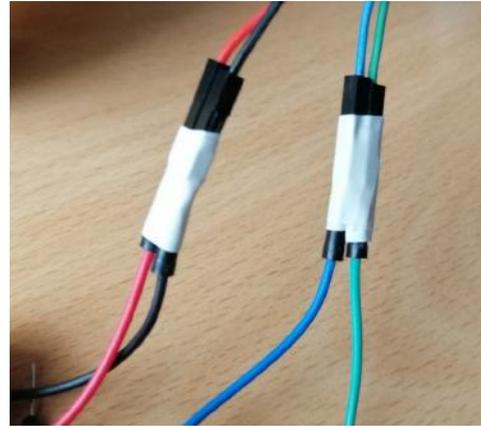
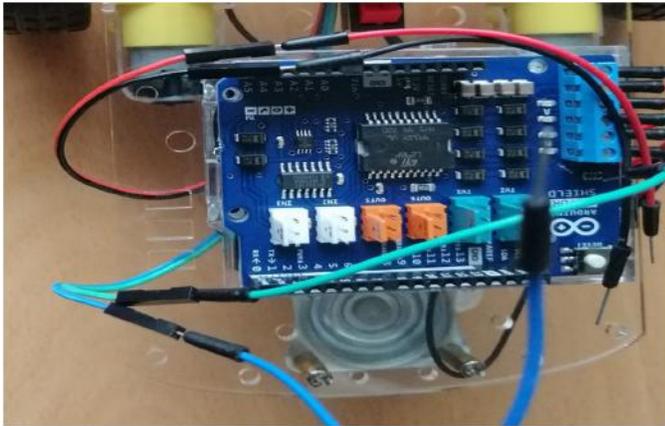




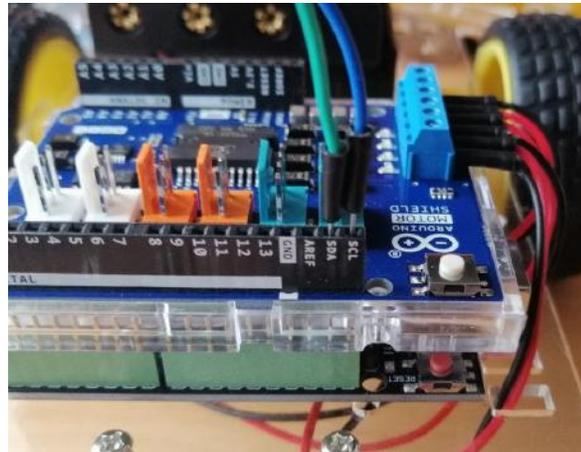
Vire o robô e passe o cabo da câmera pelos orifícios conforme mostrado na imagem abaixo



Conecte os jumpers ao cabo da câmera - use jumpers com cores correspondentes  
Prenda a junta com fita isolante



Conecte os jumpers de alimentação e as conexões I2C à blindagem do motor  
O jumper vermelho vai para 5V, preto para GND, azul para SCL e verde para porta SDA



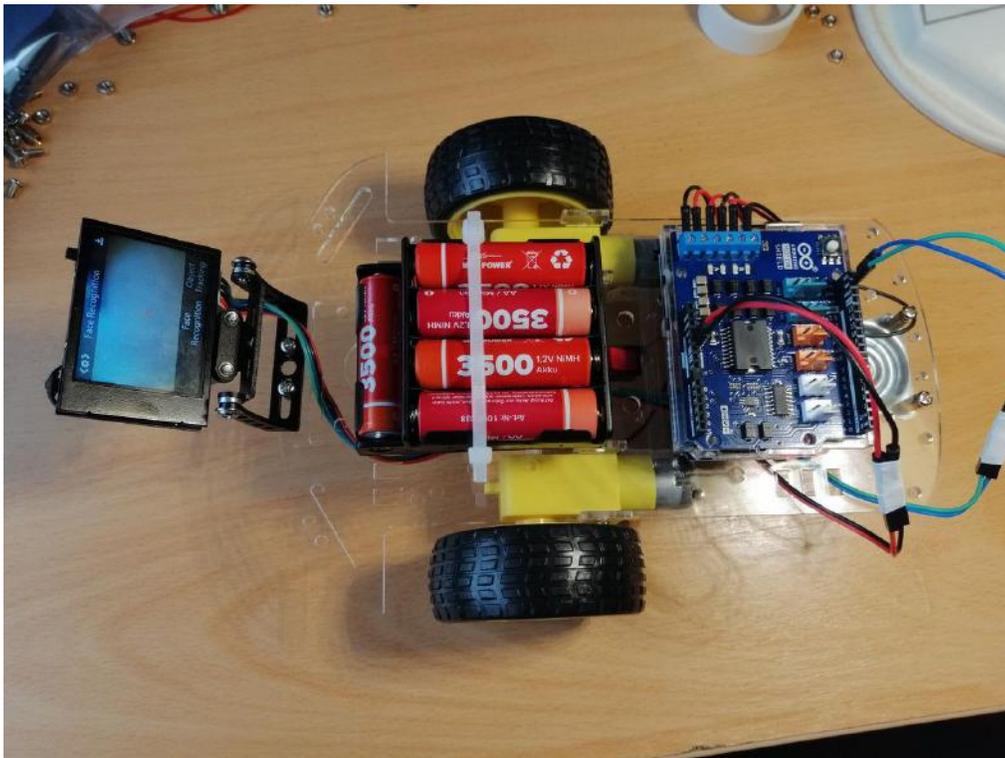


Coloque as baterias em ambos os suportes

Use duas braçadeiras para prender os suportes superior e inferior da bateria (não aperte demais - você poderá mover o suporte superior para frente e para trás)



Ligue o interruptor - se você conectou tudo corretamente - o Arduino UNO e a câmera HuskyLens serão ligados.





## Plano de aula 15

**Duração: 90 minutos**

### Tópicos

Programar o robot para se mover

### Objetivos

Aprender a programar um robô para se mover

### Resultados

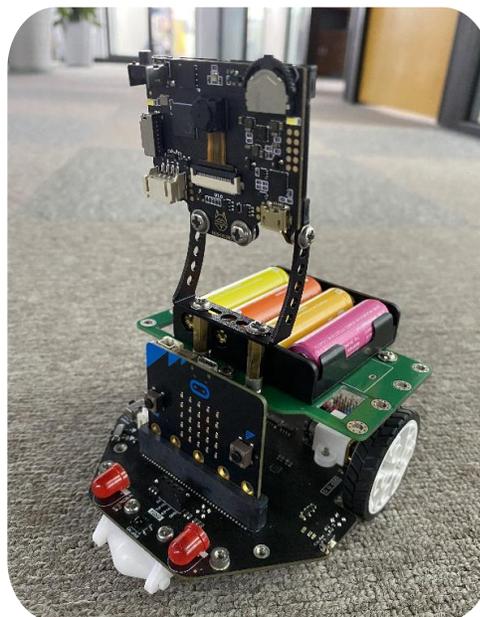
Escrever um programa para o robô mover-se

## Programação de robôs - em movimento

Robôs são legais. Nesta lição, mostramos exemplos passo a passo e fáceis de entender de programação dos movimentos de um robô móvel autônomo.

### Apresentação do objetivo da aula:

Escreva nosso primeiro programa com o qual iniciaremos o robô.

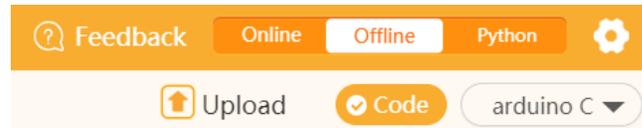


<http://erasmus-artie.eu>

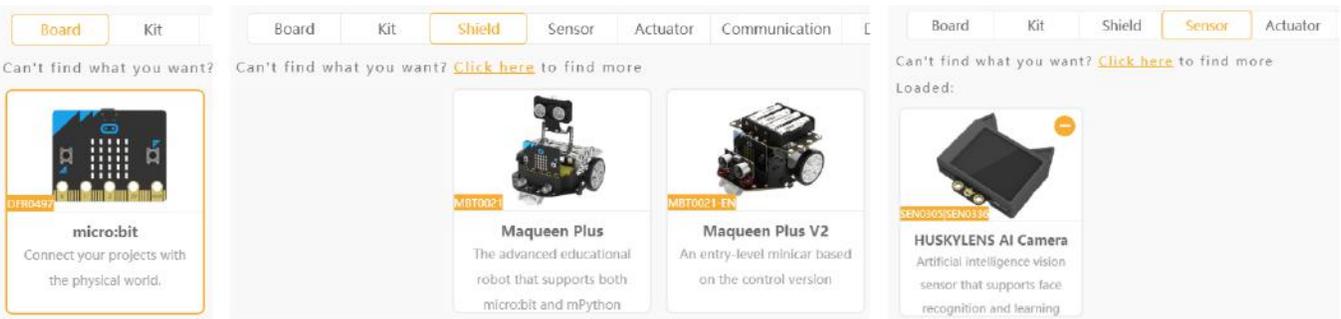


# PARTE PRINCIPAL

Passo 1 (ambas as opções): Aceda a: <http://mindplus.cc/download-en.html> e à versão descarregada para o sistema operativo do seu computador. Instale e inicie o Mind+. Depois de iniciar, mude para o modo Offline.

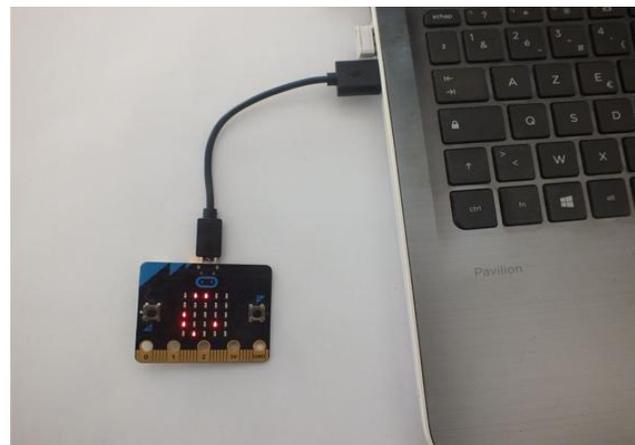


Se estiver a trabalhar com o Arduino, passe diretamente para o passo 2. -> Maqueen Plus



<http://erasmus-artie.eu>

Clique em Extensões e, no separador Placa, seleccione micro:bit, no separador Escudo, seleccione Maqueen Plus e, no separador Sensor, seleccione Câmara AI HUSKYLENS. Clique em Back e o seu software está pronto a utilizar as peças seleccionadas. Ligue o micro:bit ao seu computador através do cabo micro USB e o indicador de alimentação ligar-se-á





Clique em Ligar dispositivo e selecione micro:bit. Instale os controladores do dispositivo, se necessário.

Teste a sequência de carregamento com este código:

```

micro:bit starts
  set all motor direction rotate forward speed 200
  wait 1 seconds

```

Clique em Carregar. O robô deve avançar por um segundo e parar.

Experimente esta sequência - o robô deve se mover conforme descrito nos comentários. (para frente 2 segundos, para trás 2 segundos, rotação no sentido anti-horário 2 segundos e rotação no sentido horário 2 segundos após os quais o robô para)

```

micro:bit starts
  set all motor direction rotate forward speed 200
  wait 2 seconds
  set all motor direction rotate backward speed 200
  wait 2 seconds
  set left motor direction rotate backward speed 200
  set right motor direction rotate forward speed 200
  wait 2 seconds
  set left motor direction rotate forward speed 200
  set right motor direction rotate backward speed 200
  wait 2 seconds
  set all motor stop

```

Drive forward for 2 seconds

Drive backward for 2 seconds

Turn counter-clockwise for 2 seconds

Turn clockwise for 2 seconds



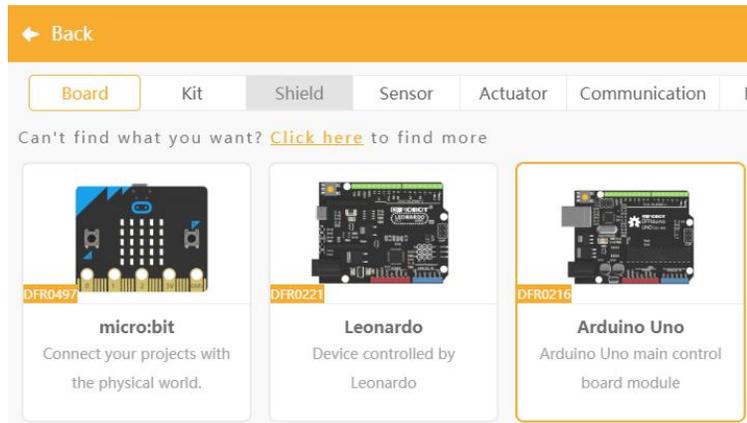
<http://erasmus-artie.eu>

Clique em Carregar. O robô deverá se mover conforme descrito nos comentários (blocos amarelos).

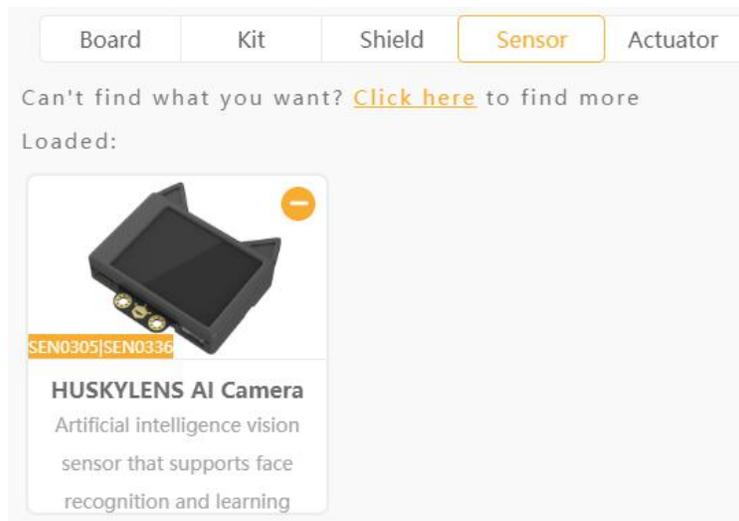
Experimente velocidades diferentes para fazer seu robô andar mais rápido ou mais devagar.



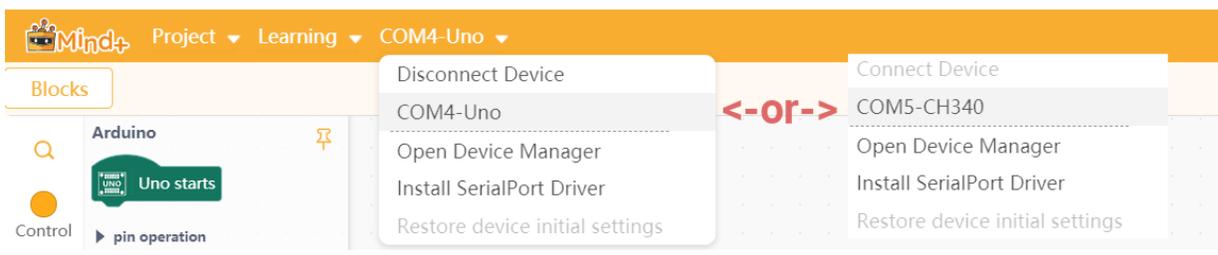
### Passo 2: Abrir Extensões e selecionar a placa - Arduino UNO



### Passo 3: Mudar para o separador do sensor e selecionar o sensor - HUSKYLENS AI Camera



Passo 4: Uma vez selecionado, clique em <- Voltar e você estará pronto para usar os blocos Arduino e Sensor. Vamos testar para ver se funciona. Antes disso, você precisa conectar o dispositivo. Conecte seu Arduino UNO via cabo USB e selecione COM X-Uno (ou CH340) dependendo do fabricante do Arduino.





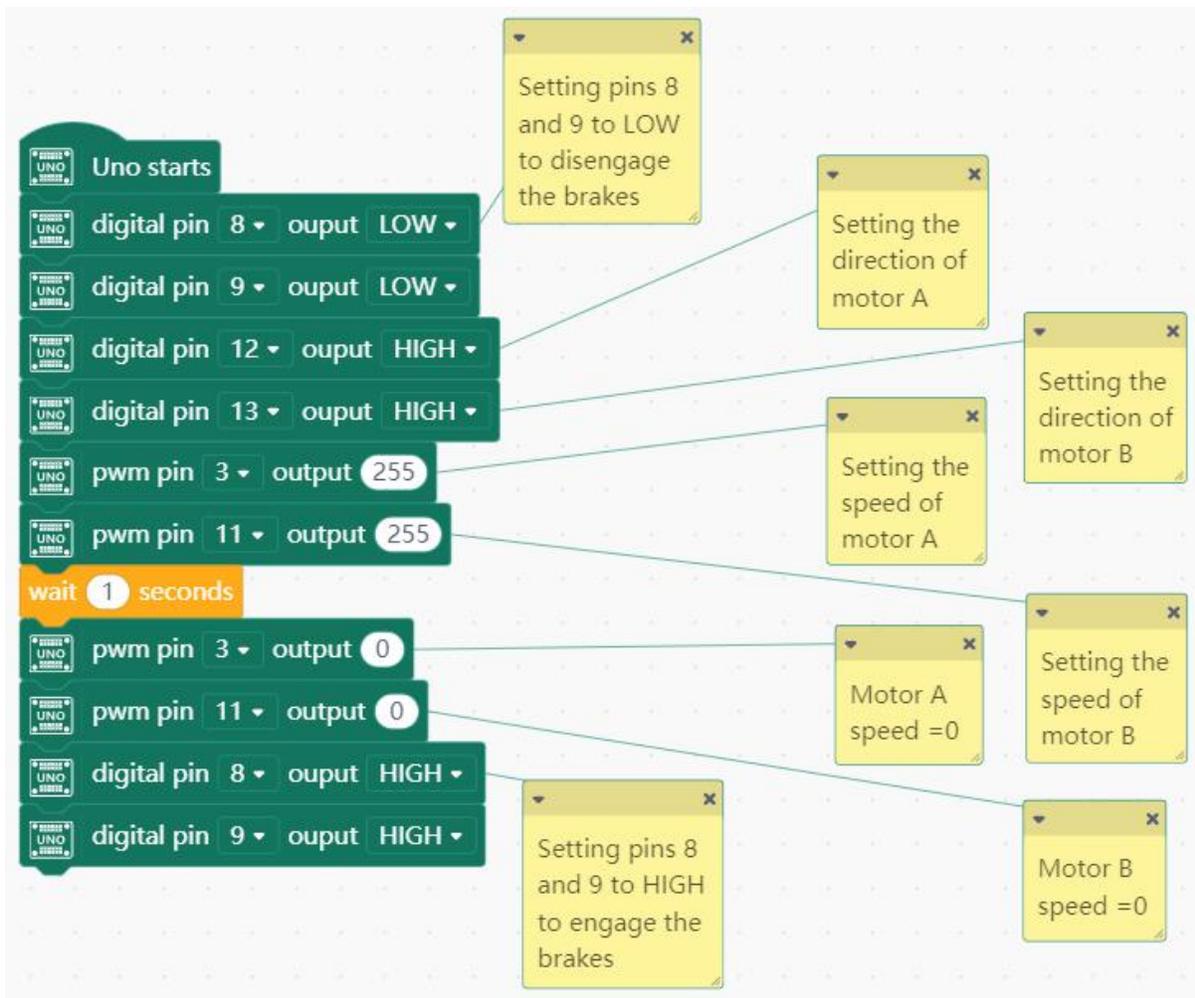
### Movimento básico

Lembra-se desta tabela do cenário de introdução ao hardware?

Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1

Dois motores (A e B) são os motores da esquerda e da direita. Os pinos digitais 12 e 13 são utilizados para mudar as direcções (HIGH - uma direcção ou LOW - direcção oposta) e os pinos PWM 3 e 11 são utilizados para definir a velocidade (0-255). Os pinos 9 e 8 activam/desactivam os travões (HIGH - travões ligados, LOW - travões desligados).

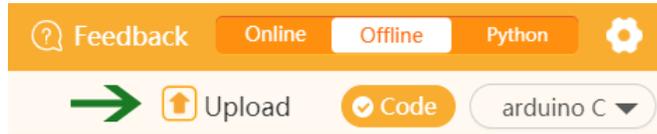
Abaixo pode ver o exemplo de código com comentários no lado direito para o ajudar a compreender o seu funcionamento



<http://erasmus-artie.eu>



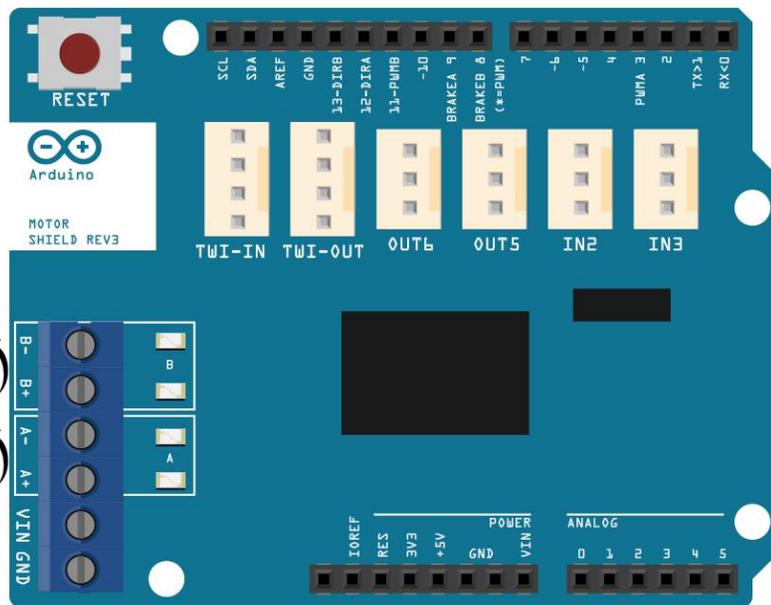
Passo 1a: Prima o botão Upload para transferir este código para o Arduino UNO.



Passo 2a: Tem cuidado, o robô começa a mover-se imediatamente após o carregamento estar concluído! O robô deve mover-se para a frente durante um segundo e depois parar.

Resolução de problemas se o robô..:

- se desloca para trás - trocar os fios vermelho e preto nos terminais de ambos os motores.
- Roda no sentido dos ponteiros do relógio - liga os fios vermelho e preto nos terminais do motor B.
- rodar no sentido contrário ao dos ponteiros do relógio - trocar os fios vermelho e preto nos terminais do motor A.



switch wires for motor B

switch wires for motor A



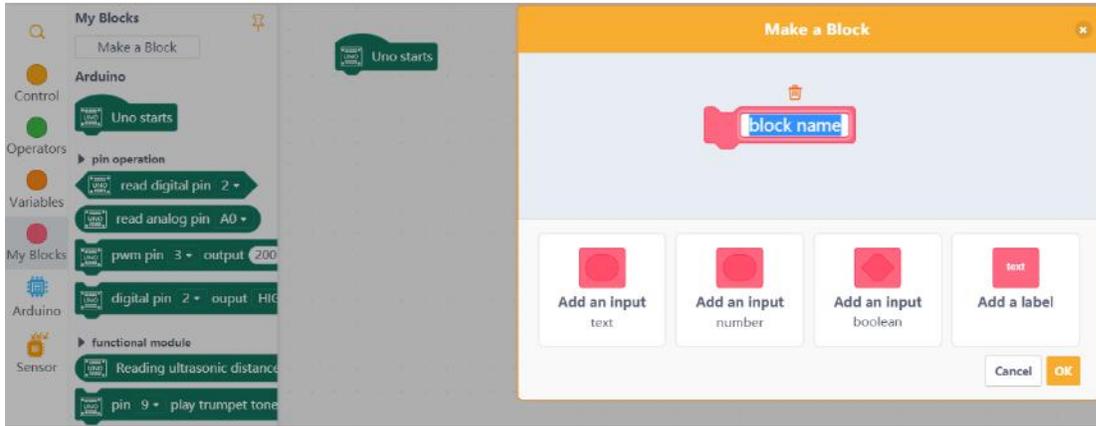
<http://erasmus-artie.eu>

Agora o robô deve estar a andar para a frente com o estado HIGH nos pinos 12 e 13. Há também uma forma mais fácil de programar o movimento - vamos utilizar os nossos blocos personalizados em vez de repetir vários blocos de definição de pinos.

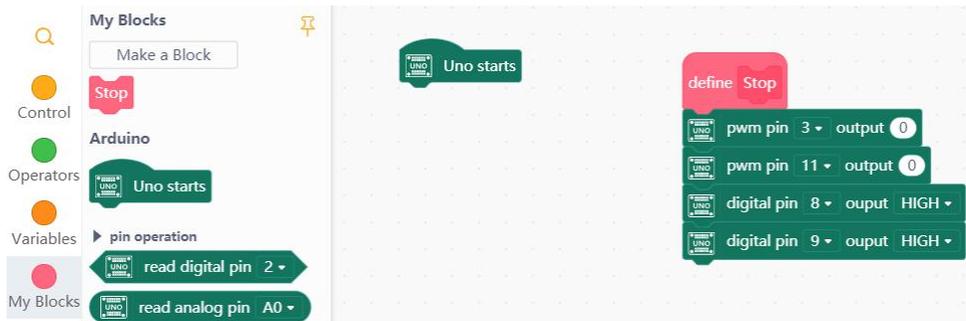
Passo 1b: Clica no grupo My Blocks (vermelho).

Passo 2b: Clicar no botão Criar um bloco e mudar o "nome do bloco" para "Paragem".

Passo 3b: Clique em OK.



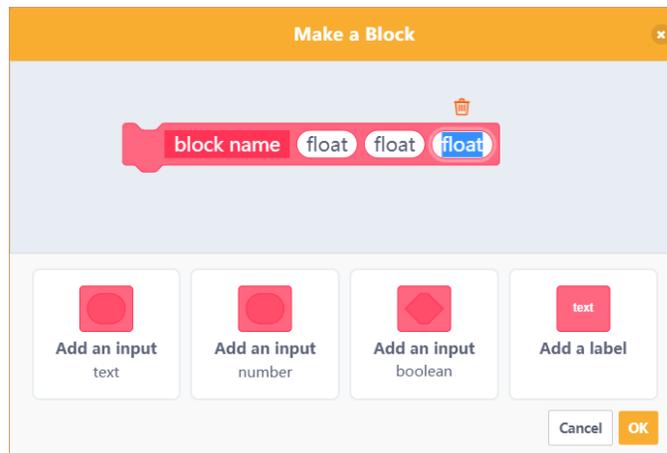
Já têm o vosso primeiro bloco, mas atualmente não está associado a nada, pelo que temos de o definir primeiro.



<http://erasmus-artie.eu>

O bloco Stop contém 4 blocos. Os dois primeiros blocos definem a velocidade de ambos os motores para 0 e os dois últimos são blocos de "travagem". Se quiseres usar a desaceleração, basta definir o estado dos pinos digitais 8 e 9 para BAIXO.

Agora é altura de criar um bloco de entrada com 3 parâmetros numéricos que manterão a velocidade de ambos os motores e a duração. Clique novamente no botão Criar um bloco e, em seguida, clique em Adicionar uma entrada - número - 3 vezes e deverá obter isto:





Altere o nome do bloco para "Drive", o primeiro flutuador para "SpeedA", o segundo flutuador para "SpeedB" e o terceiro flutuador para "Time" e clique em OK.



A ideia básica é obter os valores SpeedA e SpeedB (intervalo aceitável: -255 a 255), verificar se algum deles é negativo (ou ambos) e, se for, inverter a direção definindo o pino de direção correspondente. Mas para acionar o motor tem de usar o valor absoluto no pino PWM.

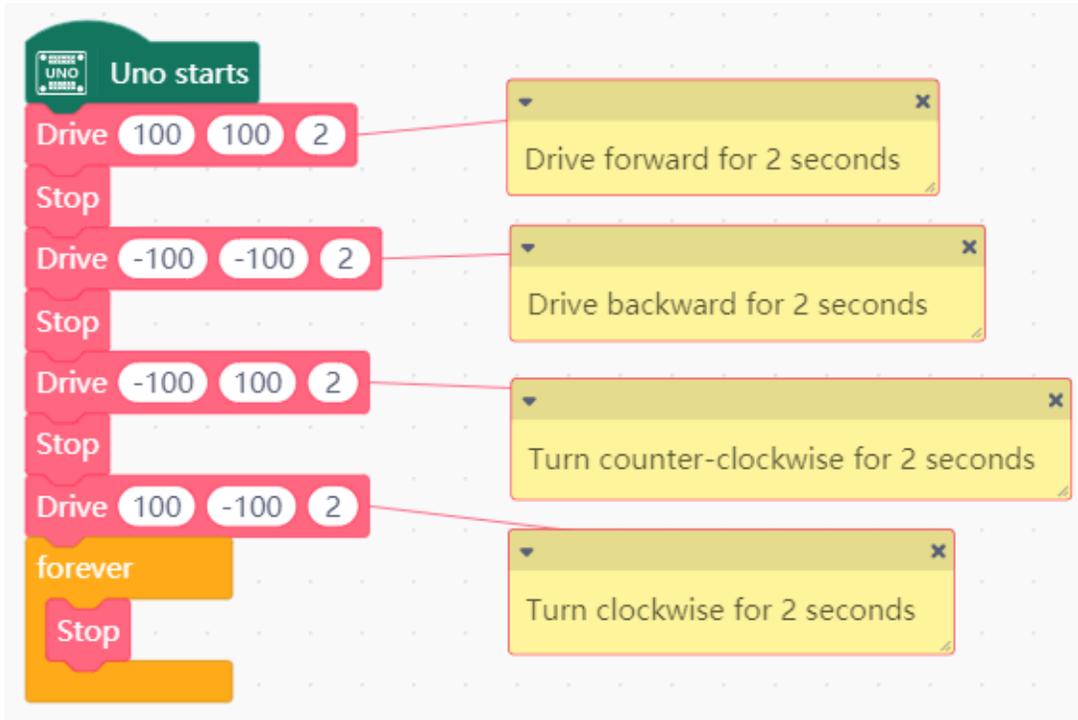
```

define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds
  
```





Então, vamos tentar mover o robot com os nossos blocos



<http://erasmus-artie.eu>

A última paragem é num loop infinito para terminar o movimento. Altera os valores no bloco Drive, carrega o teu programa no Arduino UNO e analisa a rapidez com que o teu robô se move. Está agora pronto para utilizar o seu ARTIEbot em projectos mais complexos, incluindo a câmara HuskyLens.

## CONCLUSÃO

Vamos ser sinceros, os robôs são legais. Eles podem governar o mundo um dia e, esperançosamente, nesse ponto terão misericórdia de seus insignificantes criadores (os engenheiros robóticos) e nos ajudarão a construir o paraíso no espaço. Foi uma piada, claro, mas ele sabe disso.



## Plano de aula 16

**Duração: 90 minutos**

### Tópicos

Programação de um robô para seguir objetos utilizando o rastreamento e o reconhecimento de objetos

### Objetivos

Programar um robô para seguir objetos

### Resultados

Escrever um programa para um robô com o objeto descrito a seguir

## Como preparar um robô para seguir objetos

Aprendemos como fazer um robô se mover na lição anterior.

Verifique com seus alunos se eles entenderam tudo da última aula e se estão prontos para a próxima etapa.

Vamos mover nossos robôs em direção a um objeto específico agora. Primeiro, devemos detectar esse objeto e rastreá-lo. O rastreamento de um objeto em movimento requer tecnologia de rastreamento visual de objetos, bem como operação manual.

### Apresentação do objetivo da aula:

Como preparar um robô para rastrear objetos?



# PARTE PRINCIPAL

O seguimento de objetos é uma tarefa importante na visão computacional. Refere-se ao processo de inferir continuamente o estado dos objectos em sequências de vídeo. A imagem é recolhida por uma única câmara e a informação da imagem é transmitida a um microcontrolador. Após análise e processamento, é calculada a posição relativa do objeto em movimento. Ao mesmo tempo, o robô que transporta uma câmara é controlado para rodar e seguir o objeto em tempo real.

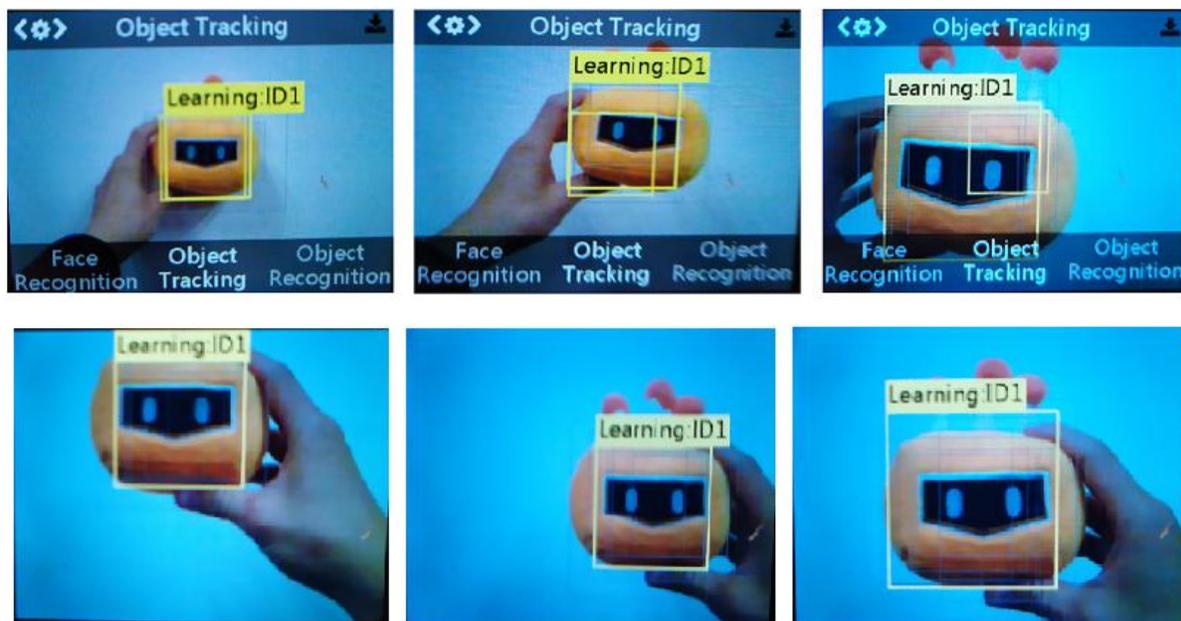
Quando o sistema de seguimento de objectos executa a função de seguimento, esta divide-se principalmente em 4 passos:

- reconhecimento do objeto
- rastreio do objeto
- análise do movimento do objeto
- controlo do robô (ou de qualquer outro sistema) com uma câmara

## Reconhecimento de objetos - Aprendizagem

Ligue o micro:bit ou Arduino UNO com a câmara HuskyLens ao seu computador portátil ou de secretária. Aponte a HuskyLens para o objeto alvo, ajustando a distância e até que o objeto esteja contido na caixa delimitadora laranja do centro do ecrã. Também é aceitável que apenas parte do objeto esteja incluída na caixa, mas com características distintas.

Em seguida, prima longamente o "botão de aprendizagem" para aprender o objeto a partir de vários ângulos e distâncias. Durante o processo de aprendizagem, a caixa cor de laranja com as palavras "Aprendizagem: ID1" será apresentada no ecrã.



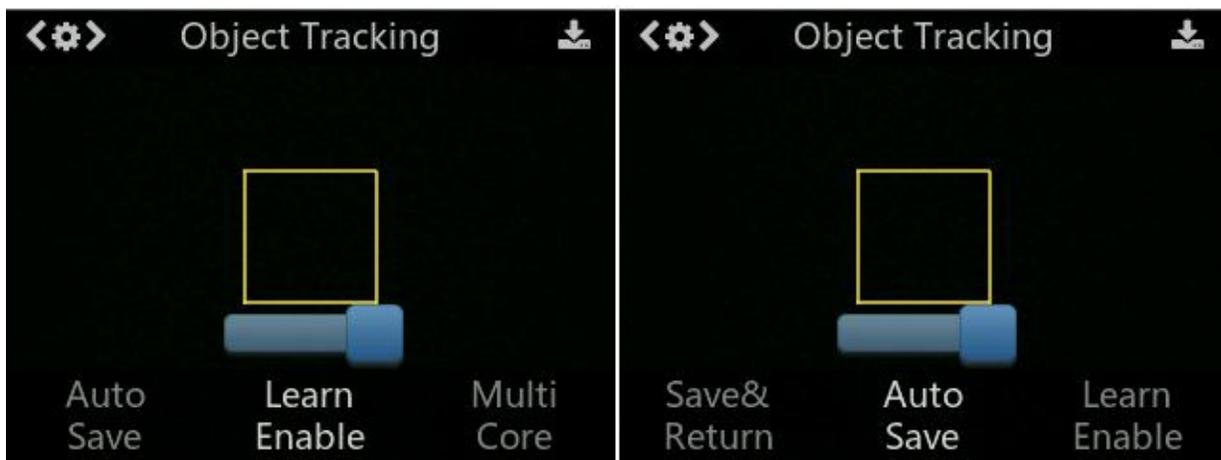


Quando o HuskyLens conseguir localizar o objeto em diferentes ângulos e distâncias, solte o "botão de aprendizagem" para concluir a aprendizagem. Se não existir uma caixa laranja no centro do ecrã, significa que o HuskyLens já aprendeu um objeto. Se pretender localizar outro objeto - selecione "Esquecer objeto aprendido" e aprenda novamente.

Na função de rastreamento de objetos, o HuskyLens pode continuar aprendendo, ou seja, enquanto a câmara vir o objeto aprendido, ela continuará aprendendo o estado atual do objeto, o que favorece a captura do objeto dinâmico. Método de operação: pressione longamente o botão de função para inserir a configuração do parâmetro da função de rastreamento de objetos.

**f**

Empurre a tecla de função para a direita para acessar o parâmetro Learn Enable, depois pressione a tecla de função brevemente para selecioná-lo e, em seguida, empurre-a para a direita para ativar Learn Enable, ou seja, o controle deslizante na barra está na posição correta. Em seguida, pressione brevemente a tecla de função novamente para confirmar este parâmetro. Saia do submenu com um toque curto no item Salvar e Retornar para salvar as alterações.



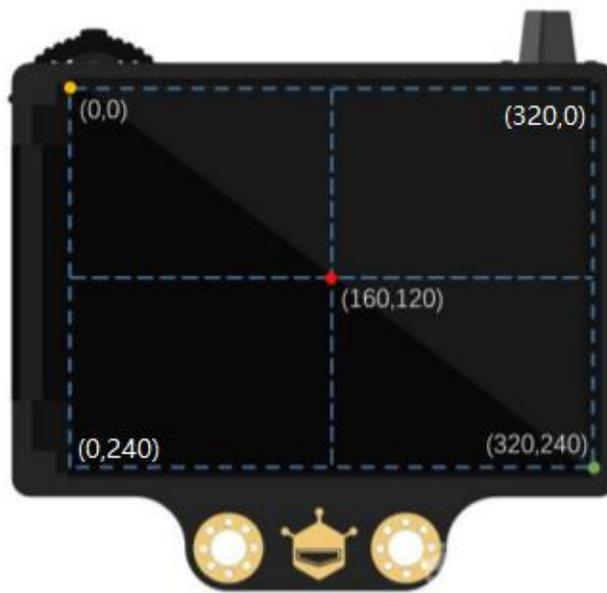


Ao reiniciar o HuskyLens, o último objeto aprendido não é salvo por padrão e você pode ativar a opção Salvar automaticamente conforme explicado anteriormente.

Como trabalhar: Da mesma forma que acima, após inserir a configuração do parâmetro, mova o controle deslizante no item Auto Save para a posição correta. Dessa forma, você só precisa aprender o objeto uma vez. Reiniciar a câmera salvará o último objeto que você aprendeu.

### Seguimento de objeto

A resolução do ecrã do sensor HuskyLens é de 320\*240, como mostra a imagem seguinte.



As coordenadas do ponto central do objeto obtidas através do programa também se encontram dentro deste intervalo. Por exemplo, se os valores das coordenadas obtidos forem (160, 120), o objeto a ser seguido está no centro do ecrã.

"Coordenadas X" e "Coordenadas Y" referem-se à posição do ponto central da caixa na coordenada do ecrã. "Object width" (Largura do objeto) e "Object height" (Altura do objeto) referem-se ao tamanho da moldura. Na função de seguimento de objectos, a moldura é quadrada, pelo que a largura e a altura são iguais.

### Testar o seguimento de objectos - Opção 1 (Maqueen Plus/HuskyLens)

Abra o Mind+ e carregue as extensões para trabalhar com o Maqueen Plus e a câmara HuskyLens. Utilize este código:



```

micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
set motor All stop

forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      serial output join "X: " HuskyLens get X center of ID 1 frame from the result in string , No-Wrap
      serial output join ", Y: " HuskyLens get Y center of ID 1 frame from the result in string , No-Wrap
      serial output join ", W: " HuskyLens get width of ID 1 frame from the result in string , No-Wrap
      serial output join ", H: " HuskyLens get height of ID 1 frame from the result in string , Wrap
    end if
  end if
  wait 1 seconds
end forever

```

### Passar para a verificação dos resultados

### Testar a localização de objectos - Opção 2 (Arduino UNO/HuskyLens)

Abra o Mind+ e carregue as extensões para trabalhar com o Arduino UNO e a câmara HuskyLens. Utilize este código com o Arduino/HuskyLens:

```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      serial output join "X: " HuskyLens get X center of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", Y: " HuskyLens get Y center of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", W: " HuskyLens get width of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", H: " HuskyLens get height of the ID 1 No. 1 frame from the result in string , Wrap
    end if
  end if
  wait 0.5 seconds
end forever

```



http://erasmus-artie.eu

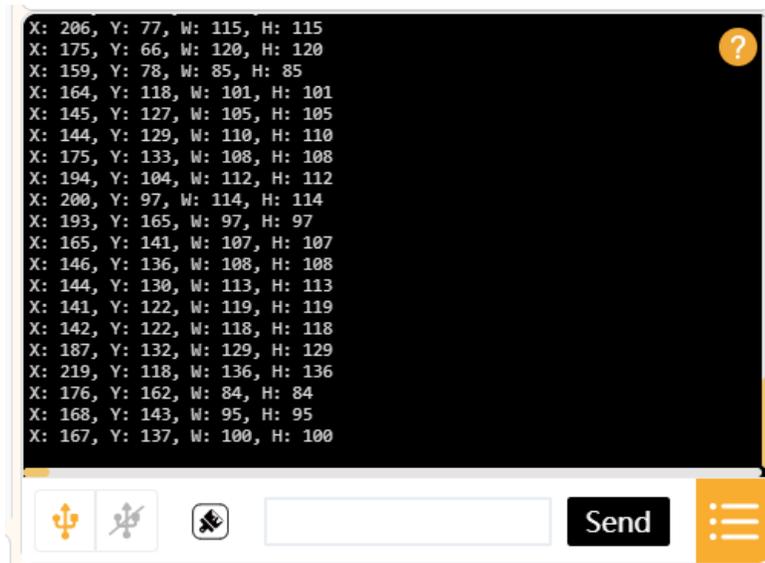


### Verificar os resultados no monitor de série (ambas as opções)

Abra o monitor de série clicando no ícone USB na parte inferior direita do ecrã da Mind+.

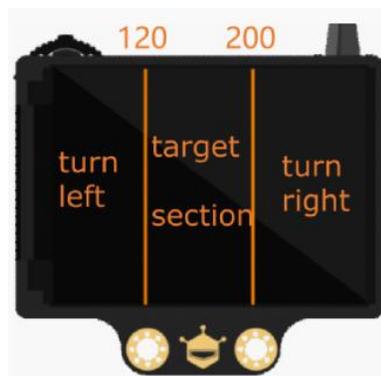


Tente mover o objeto para a esquerda e para a direita para observar a variação numérica do centro X. Mova o objeto para cima e para baixo para observar a variação numérica do centro Y. Mova o objeto para a frente e para trás para observar a variação numérica da largura e da altura.



### Análise do movimento do objeto

Como se pode ver na imagem seguinte, o ecrã está dividido em 3 secções de acordo com o eixo X do sistema de coordenadas do ecrã da câmara e a secção central é a nossa secção alvo.

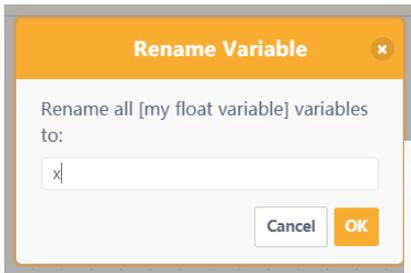


<http://erasmus-artie.eu>



Quando a câmara detecta continuamente o estado do objeto alvo na imagem, o seu centro X é 120-200, o que significa que o alvo está no centro do campo de visão e que o robô não precisa de ajustar a sua posição; o seu centro X é 0-120, o nosso robô precisa de se ajustar virando à direita; o seu centro X é 200-320, o ARTIEbot precisa de virar à esquerda para se ajustar.

Agora é altura de criar uma parte principal do código para virar o robô na direção do objeto. Ambas as opções - Renomeia a minha variável flutuante para x. Clica com o botão direito do rato na variável -> Renomear variável numérica.



```

micro:bit starts
  HuskyLens initialize pin until success
  HuskyLens switch algorithm to Object tracking
  set motor All stop
  forever
    HuskyLens request data once and save into the result
    if HuskyLens check if ID 1 is learned from the result? then
      if HuskyLens check if ID 1 frame is on screen from the result? then
        set x to HuskyLens get X center of ID 1 frame from the result
        if x >= 120 and x <= 200 then
          set motor All stop
        else if x > 0 and x < 120 then
          set motor Left move by 20 speed Forward
          set motor Right move by 80 speed Forward
        else if x > 200 and x < 320 then
          set motor Left move by 80 speed Forward
          set motor Right move by 20 speed Forward
        else
          set motor All stop
      else
        set motor All stop
    else
      set motor All stop
  
```



### Opção 1 - seguir o objeto com Maqueen Plus

Utilizar e configurar os blocos como na figura abaixo:



## Opção 2 - seguir o objeto com o ArtieBot

Primeiro, define os blocos Drive e Stop como descrito na lição anterior (Programar o robô)

```
define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds

define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0
```





Utilizar e configurar os blocos como na figura abaixo:

```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      set x to HuskyLens get X center of the ID 1 No. 1 frame from the result
      if x >= 120 and x <= 200 then
        Stop
      else if x > 0 and x < 120 then
        Drive 20 80 0.25
      else if x > 200 and x < 320 then
        Drive 80 20 0.25
      else
        Stop
    else
      Stop
  else
    Stop
  
```

```

define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0

define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds
  
```



<http://erasmus-artie.eu>

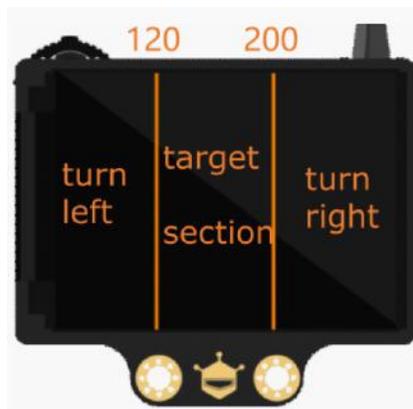
### AMBAS as opções - Verificar como funciona

Carrega o programa para o teu robot.

Faz os ajustes à velocidade do MotorA ou MotorB, se necessário.

Quando a caixa do objeto identificado estiver no centro do ecrã, o robô pára.

Quando a caixa está no lado esquerdo ou direito do ecrã, o robô ajusta automaticamente a posição para a esquerda ou para a direita até que a caixa esteja localizada na secção alvo do ecrã





## Seguir o objeto

Os nossos robôs viraram-se para os objectos, mas ainda não os seguiram. Para isso, temos de detetar o tamanho do objeto para saber se é grande (perto da câmara) ou pequeno (longe da câmara).

Crie uma nova variável (numérica) e dê-lhe o nome de h. Esta variável irá guardar a altura do objeto que estamos a seguir. Se a altura do objeto estiver entre 60 e 100, o robô manterá a posição atual. Se for inferior a 60 - está longe e temos de fazer com que o robô avance. Se for superior a 100, o robô deve deslocar-se para trás.

### Opção 1 - Código para Maqueen Plus

```
micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
set motor All stop
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      set x to HuskyLens get X center of the ID 1 No. 1 frame from the result
      set h to HuskyLens get height of the ID 1 No. 1 frame from the result
      if x >= 120 and x <= 200 and h >= 60 and h <= 100 then
        set motor All stop
      else if x > 0 and x < 120 then
        set motor Left move by 20 speed Forward
        set motor Right move by 80 speed Forward
      else if x > 200 and x < 320 then
        set motor Right move by 80 speed Forward
        set motor Left move by 20 speed Forward
      else if h > 0 and h < 60 then
        set motor All move by 80 speed Forward
      else if h > 100 and h < 240 then
        set motor All move by 80 speed Backward
    else
      set motor All stop
```





### Opção 2 - Código para Arduino (ArtieBot):

```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      set x to HuskyLens get X center of the ID 1 No. 1 frame from the result
      set h to HuskyLens get height of the ID 1 No. 1 frame from the result
      if x >= 120 and x <= 200 and h >= 60 and h <= 100 then
        Stop
      else if x > 0 and x < 120 then
        Drive 20 80 0.25
      else if x > 200 and x < 320 then
        Drive 80 20 0.25
      else if h > 0 and h < 60 then
        Drive 80 80 0.25
      else if h > 100 and h < 240 then
        Drive -80 -80 0.25
      else
        Stop
    else
      Stop
  else
    Stop
  
```

```

define Drive SpeedA SpeedB Time
digital pin 8 output LOW
digital pin 9 output LOW
if SpeedA > 0 then
  digital pin 12 output HIGH
else
  digital pin 12 output LOW
if SpeedB > 0 then
  digital pin 13 output HIGH
else
  digital pin 13 output LOW
pwm pin 3 output abs of SpeedA
pwm pin 11 output abs of SpeedB
wait Time seconds

define Stop
pwm pin 3 output 0
pwm pin 11 output 0
  
```

f <http://erasmus-artie.eu>

### Ambas as opções - Verificar o seu funcionamento

Carregue o programa no micro:bit/Arduino UNO para verificar o seu funcionamento. Faça as correções para tornar o movimento suave, ajustando a velocidade dos motores e o tempo de condução.

Depois de o HuskyLens terminar de aprender um objeto, os robôs seguem-no automaticamente e movem-se para a frente, para trás, para a esquerda e para a direita, mantendo a caixa do objeto no centro do ecrã e a uma distância adequada.

Quando o robô é utilizado como um robô de seguimento, pode ser programado para localizar qualquer alvo com a câmara HuskyLens. Isto significa



O rastreamento de objetos é a tarefa de obter um conjunto de valores após a detecção do objeto, criando um identificador único para cada um dos objetos detectados e, em seguida, rastrear cada um dos objetos conforme eles se movem e marcá-los com uma caixa no vídeo, respeitando os identificadores atribuídos. Os métodos de última geração incluem a fusão de dados de RGB e outras câmeras para fornecer rastreamento de objetos mais confiável. Agora entendemos os princípios básicos do rastreamento de objetos e aprendemos a função de rastreamento de objetos do HuskyLens. Também sabemos como usar o HuskyLens para que nosso robô possa rastrear um objeto.



## CONCLUSÃO

Para rastrear um objeto com sucesso, devemos primeiro detectá-lo e depois rastreá-lo. Rastrear um objeto em movimento requer tecnologia de rastreamento visual de objetos, bem como controle manual.





## Plano de aula 17

**Duração: 90 minutos**

### Tópicos

Robô de programação para o seguimento de linhas utilizando uma câmara de luz visível, uma câmara termográfica de infravermelhos e outros instrumentos de deteção

### Objetivos

Programar um robô para seguir objetos

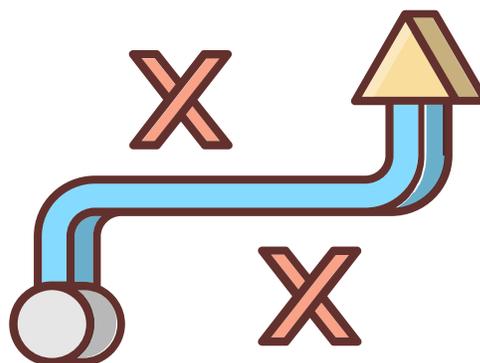
### Resultados

Escrever um programa para um robô com o objeto descrito a seguir

## Projeto com um robô - seguimento de linha

Aprendemos como programar robôs para seguir objetos na lição anterior. Certifique-se de que seus alunos entendam tudo da lição anterior e estejam prontos para a próxima etapa. Vamos aprender agora como encontrar e seguir uma linha.

**Apresentação do objetivo da aula:**  
Como preparar o robô para seguir a linha?



<http://erasmus-artie.eu>



# PARTE PRINCIPAL

O seguimento de linha refere-se ao processo de deslocação de um objeto ao longo de uma rota designada. Um robô de seguimento de linhas totalmente funcional utiliza um robô móvel como transportador, uma câmara de luz visível, um termovisor de infravermelhos e outros instrumentos de deteção como sistema de carga, uma fusão de informação multicampo de visão artificial, campo eletromagnético, GPS e GIS como sistema de navegação para movimento e seguimento autónomos do robô, e um computador incorporado como plataforma de desenvolvimento de software e hardware para o sistema de controlo.

## Sensores de seguimento de linha

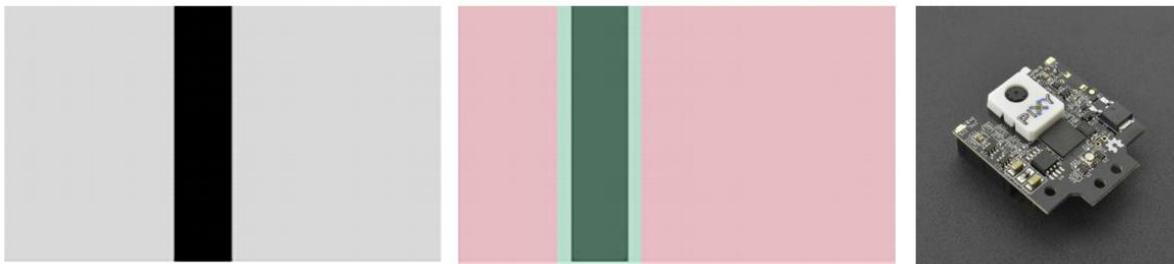
Type Comparison Items	Infrared Line Tracking Sensor	Visual Sensor
Cost	Low	High
Range of Vision	The sensor has a small range of view; it needs to be close to the ground .	The range of vision is wide, and the movement state can be adjusted in advance according to the line changes.
Environment Adaptation	When the usage environment is changed, the sensitivity of the sensor needs to be adjusted, and the adjustment process is complicated.	When the use environment is changed, only the lines need to be relearned, and the operation is simple.
Map Adaptation	Generally, only suitable for simple maps with clear background lines, black and white lines or solid lines	Suitable for maps with clear background lines, multi-colour lines, solid lines, dotted lines and other complex conditions.



<http://erasmus-artie.eu>

## Algoritmo de seguimento de linhas do HuskyLens

A função de seguimento de linhas do HuskyLens baseia-se no Pixy, um projeto de código aberto da Universidade Carnegie Mellon. O algoritmo do Pixy consegue reconhecer a cor das imagens. A sua ideia básica é utilizar o espaço de cor para remover o fundo que não interessa a todos os utilizadores e extrair o primeiro plano (como as linhas).

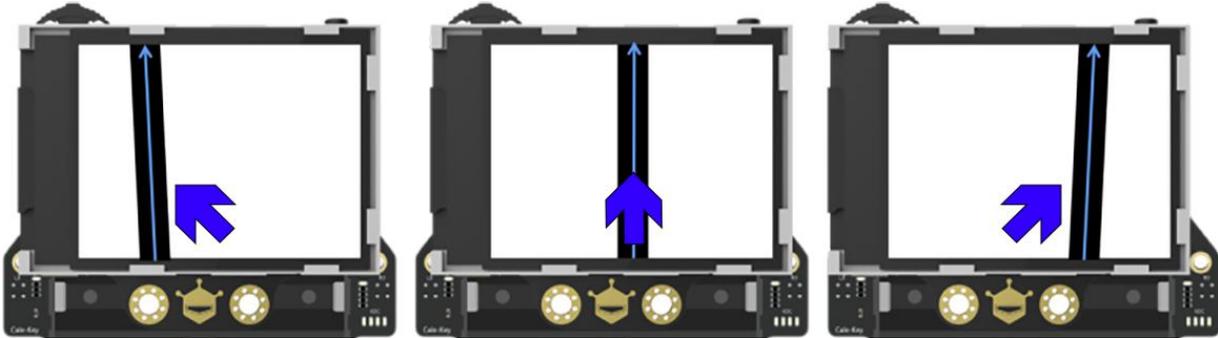


Como o robô pode seguir a linha preta no mapa de rastreamento (que tem linhas pretas em fundo branco)? Na verdade, só precisamos saber a posição relativa do ARTIEbot em relação à linha preta.



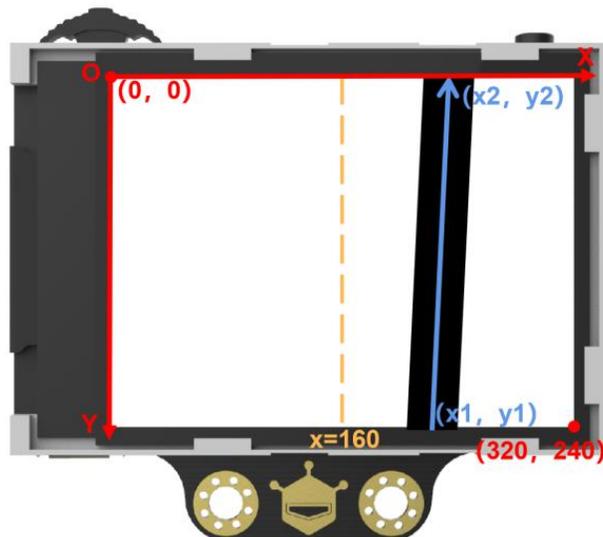
Basicamente, temos as seguintes três situações:

1. Quando o robô estiver no lado direito da linha preta, ele deverá virar à esquerda
2. Quando o robô está no centro, alinhado com a linha preta, ele deve seguir em linha reta
3. Quando o robô estiver no lado esquerdo da linha preta, ele deverá virar à direita



## Implementação

A resolução da tela HuskyLens é  $320 \times 240$ . O ponto O no canto superior esquerdo da tela é a origem das coordenadas da tela  $(0, 0)$ , a direção horizontal direita é a direção positiva do eixo X e a direção vertical para baixo é a direção positiva do eixo Y, então as coordenadas no canto inferior direito da tela são  $(320, 240)$ . A linha laranja pontilhada na imagem é o eixo central da tela e o valor x desta linha é 160. A linha preta na tela abaixo está a linha do mapa "vista" pela câmera HuskyLens. A seta azul é a direção da linha calculada pelo HuskyLens. As coordenadas do ponto inicial da seta azul são  $(x_1, y_1)$  e as coordenadas do ponto final são  $(x_2, y_2)$ .





Community Verified icon Para simplificar - só precisamos saber o ponto inicial (x1) da seta azul em relação ao eixo central ( $x=160$ ) para implementar o rastreamento de linha.

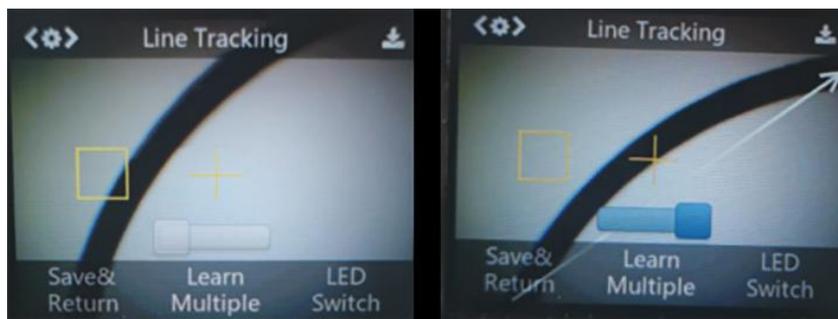
Esta função pode rastrear linhas de cores especificadas e fazer previsão de caminho. A configuração padrão é rastrear linhas de apenas uma cor e este projeto usará um rastreamento de linha de cor.

### Configurações da câmera

Passo 1: Disque o botão de função para a esquerda ou direita até que a palavra "Line Tracking" seja exibida na parte superior da tela.

Passo 2: Pressione longamente o botão de função para inserir a configuração do parâmetro da função de rastreamento de linha.

Etapa 3: disque o botão de função para a direita ou para a esquerda até que "Aprender Múltiplos" seja selecionado, pressione rapidamente o botão de função e disque-o para a esquerda para desligar a chave "Aprender Múltiplos", ou seja, o ícone quadrado no progresso barra está virada para a esquerda. Em seguida, pressione rapidamente o botão de função para concluir este parâmetro.

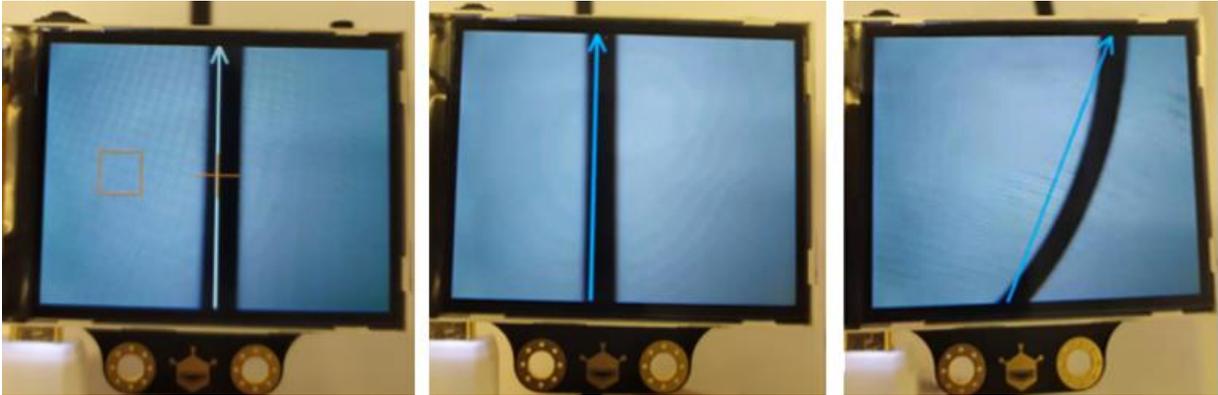


Passo 4: Você também pode ligar o LED configurando "LED Switch". Isso é muito útil em ambientes escuros. Referindo-se ao método acima, ligue o "LED Switch".

Passo 5: Disque o botão de função para a esquerda até que "Salvar e Retornar" seja selecionado e pressione rapidamente o botão de função para salvar os parâmetros e ele retornará automaticamente.

### Aprendizagem e acompanhamento

Aprendizado de linha: Aponte o símbolo "+" para a linha e, a seguir, aponte a caixa laranja para a área de fundo. Recomenda-se que não haja outras linhas na tela. Tente manter o HuskyLens paralelo à linha alvo; o HuskyLens detectará automaticamente a linha e uma seta branca aparecerá. Em seguida, pressione rapidamente o "botão de aprendizagem" e a seta branca se transforma em uma seta azul.



Ao aprender a linha, precisamos ajustar a posição do HuskyLens para ficar paralela à linha. O HuskyLens pode aprender linhas em qualquer cor que tenha um contraste de cor óbvio com o fundo, mas essa linha deve ser monocromática para manter o processo de rastreamento de linha estável.

O HuskyLens pode aprender e rastrear várias linhas com cores de linha diferentes, mas qualquer uma dessas linhas deve ser monocromática e em contraste visível com o fundo. Neste exemplo usaremos linha preta (fita isolante preta sobre fundo branco como papel ou MDF branco).

A visibilidade da linha depende muito da luz ambiente. Ao seguir a linha, tente manter a luz ambiente o mais estável possível e use o LED HuskyLens se necessário.



Open your Mind+ and load extensions.

Open your Mind+ and load extensions. When using the micro:Maqueen plus robot, make sure that you select the right version (V1 or V2).

Rename my float variable to x. Right click on variable -> Rename numeric variable.

### Algorithm

Read x1 (or x beginning) from HuskyLens Line tracking function - it is the beginning point of the blue arrow

If the black line is on the left side of the screen ( $x1 < 150$ ) - robot should turn left.

If the black line is on the right side of the screen ( $x1 > 170$ ) - robot should turn right.

If the black line is in the middle of the screen ( $150 \leq x1 \leq 170$ ) - robot should go straight.



## Opção 1 - Seguimento de linha com Maqueen Plus

Use este código:

```
micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Line tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 arrow is on screen from the result? then
      set my float variable to HuskyLens get X beginning of ID 1 arrow from the result
      if my float variable >= 150 and my float variable <= 170 then
        set motor All move by 60 speed Forward
      else if my float variable < 150 then
        set motor Right move by 90 speed Forward
        set motor Left move by 30 speed Forward
      else if my float variable > 170 then
        set motor Left move by 90 speed Forward
        set motor Right move by 30 speed Forward
      else
        set motor All stop
```





## Opção 2 - Seguimento de linha com Arduino (ArtieBot)

Primeiro defina os blocos Drive e Stop conforme descrito na lição Programando o robô. É uma boa ideia colocar os scripts usados com frequência na mochila e movê-los facilmente entre projetos.

```
define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds

define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0
  digital pin 8 output HIGH
  digital pin 9 output HIGH
```





```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Line tracking
define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 arrow is on screen from the result? then
      set x to HuskyLens get X beginning of ID 1 arrow from the result
      if x >= 150 and x <= 170 then
        Drive 90 90 0.25
      else if x < 150 then -
        Drive 40 90 0.25
      else if x > 170 then -
        Drive 90 40 0.25
      else
        Stop
  
```

f

```

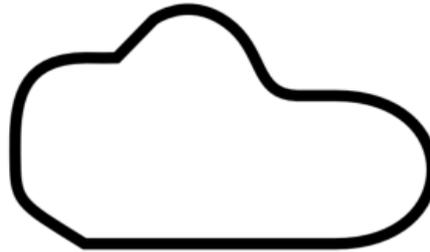
define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds
  
```

<http://erasmus-artie.eu>



## Ambas as opções – Teste seu algoritmo

Prepare a superfície branca (papel ou MDF) e use a fita isolante para fazer uma linha (veja exemplo abaixo ou faça algo semelhante).



Carregue o programa no seu robô, coloque o robô em algum lugar da linha e veja como ele se move. Segue a linha?

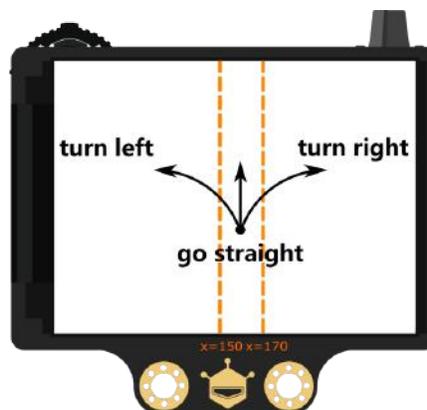
A linha está sendo perdida? Existe uma maneira de lidar com isso? As dicas a seguir irão ajudá-lo:

- Tente mudar o ângulo da câmera
- Tente obter o início X da seta ID 1 em vez do ponto final X

Adicione código à prova de falhas usando a seguinte dica:

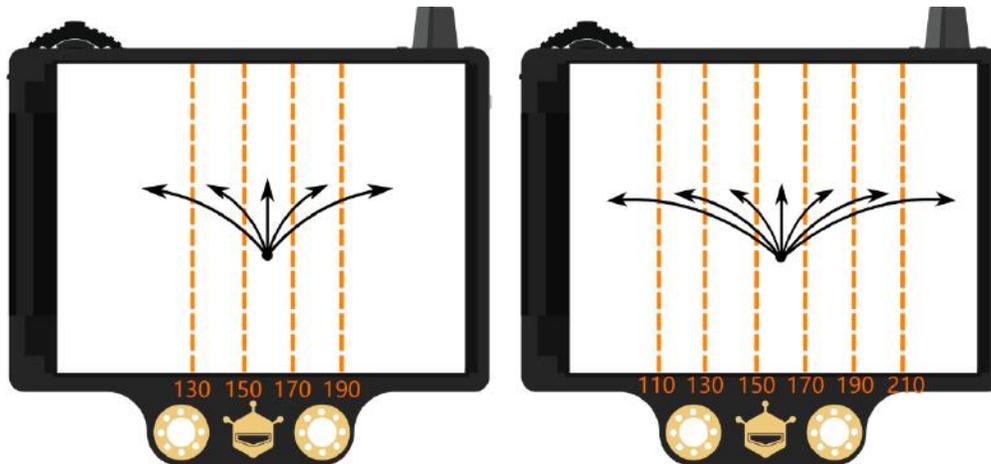
- Se uma linha estiver presente e uma seta for detectada, prossiga para o código de manipulação da seta e salve o conteúdo da variável x em uma nova variável chamada lastx
- Se a linha for perdida e a seta não for detectada, use lastx em vez de x para alcançar a linha novamente.

E finalmente o mais importante - aplique as correções para tornar o movimento suave, ajustando a velocidade dos motores e o tempo de condução. Este algoritmo analisa a posição de x; tem 3 cases como na foto abaixo.



Você pode otimizar esse algoritmo para 5 casos ou até 7 para tornar o movimento mais suave?

Veja as fotos abaixo para ter uma ideia de como fazer isso.



### Recapitulação do conhecimento

1. Entenda os princípios básicos do rastreamento de linha
2. Aprenda a aplicar a função de rastreamento de linha HuskyLens
3. Aplicar e otimizar o algoritmo de seguimento de linha

Como o nome sugere, o robô seguidor de linha é um veículo automatizado que segue uma linha visual embutida na superfície. Esta linha visual é um caminho no qual o robô seguidor de linha corre. Geralmente, ele usa uma linha preta em uma superfície branca, ou você pode ajustá-la como uma linha branca em uma superfície preta.



Normalmente, iniciantes e estudantes teriam sua primeira experiência robótica com este tipo de robô.

Nas indústrias, robôs gigantes seguidores de linha são utilizados para auxiliar o processo de produção automatizado. Eles também são usados em aplicações militares, para fins de assistência humana, serviços de entrega, etc.

Agora entendemos os princípios básicos do rastreamento de linha e sabemos como aplicar a função de rastreamento de linha HuskyLens.

Também sabemos como aplicar e otimizar o algoritmo de seguimento de linha.

Discuta com seus alunos as diferenças e semelhanças entre rastreamento e seguimento de objetos e rastreamento e seguimento de linhas.

## CONCLUSÃO

Aprendemos os princípios básicos do rastreamento de linha usando a função HuskyLens e otimizamos o algoritmo de rastreamento de linha.