

Robôs Socialmente Assistivos

Desenvolvendo Sessões de Terapia Multissensorial com o Robô **EVA**

Marcelo Marques da Rocha

Sara Luzia de Melo

Jesús Favela

Débora Christina Muchaluat Saade



Roteiro do Minicurso

■ **Introdução**

- Robôs Socialmente Assistivos (SARs)
- Aplicações em Saúde

■ **Propostas de SARs**

- Comerciais
- Acadêmicas

■ **Robô Eva**

- Arquitetura do Robô
- Principais Funcionalidades

■ **Desenvolvimento de Sessões de Terapia para o Robô EVA**

- EvaML – Linguagem baseada em XML
- EvaSIM - Simulador

■ **Atividade Prática**

Introdução

- ❖ Tecnologias robóticas têm sido utilizadas em vários ambientes com o objetivo de proporcionar uma melhor qualidade de vida para seus usuários. Robôs são empregados para realizar algum tipo de assistência e a tecnologia robótica é um método responsável por desenvolver capacidades físicas e cognitivas dos seres humanos (Shibata 2012).



Introdução

- ❖ A Robótica é considerada uma área multidisciplinar que utiliza os conhecimentos de outras ciências, tais como a Engenharia Mecânica, Engenharia Elétrica e Inteligência Artificial. Basicamente, a robótica estuda a elaboração, montagem e programação de robôs para execução de tarefas de forma automática (Giglio et.al, 2017).

Introdução

- ❖ Robôs Socialmente Assistivos (SARs) pertencem um novo campo da robótica que compreende uma classe de robôs definida pela interseção entre a classe de robôs assistivos (assistência ao usuário) e a classe de robôs sociais interativos (interação social e não física).

Introdução

- ❖ Diferente da robótica interativa, a robótica assistiva destaca-se por incluir robôs que proporcionam uma interação social, próxima e eficaz, a usuários com necessidades especiais, mais especificamente para idosos ou indivíduos com deficiências físicas, cognitivas, emocionais e sociais [Feil-Seifer and Mataric 2005, Tapus et al. 2007].

Introdução

- ❖ Robôs sociais possibilitam suporte a pessoas com Transtorno do Espectro Autista (TEA), bem como indivíduos diagnosticados com algum tipo de demência.
 - O TEA é uma condição que compromete fatores cognitivos, em especial, as habilidades sociais, o comportamento e a comunicação (Mesibov et al., 2013).
 - Uma das principais dificuldades enfrentadas pelos indivíduos com autismo é a capacidade da expressão e o reconhecimento das emoções primárias (Elder et al., 2006), (Annaz et al., 2009).

Introdução

- ❖ **As seis emoções primárias (emoções básicas) expressas pelos seres humanos foram denominadas por:**

Alegria;

Tristeza;

Medo;

Raiva;

Desprezo;

Surpresa.



Figura 1: Exemplo das seis emoções básicas
Fonte: (Ekman, 1978)

Introdução

- ❖ No âmbito do desenvolvimento das habilidades emocionais em crianças com TEA, SARs têm sido cada vez mais utilizados. Pesquisas têm verificado efeitos favoráveis nas interações criança-robô e também no reconhecimento de algumas emoções e habilidades emocionais.

- ❖ Terapias robóticas que aumentam a capacidade de imitar ações do robô, ou que despertam o desejo de imitação, estimulam o aprendizado de outras habilidades (Duquette et al. 2008).

Introdução

- ❖ Em pacientes com Alzheimer, pesquisas apresentaram contribuições na inserção dos SARs em pacientes com doença de Alzheimer, uma vez que a comunicação e interação com pacientes são estratégias eficientes para preservação das habilidades cognitivas (Tapus et al. 2009, Mordoch et al. 2013, Salichs et al. 2016, Woods et al. 2021).

Introdução

❖ Os principais desafios encontrados no desenvolvimento de SARs são:

- Definição do comportamento social dos robôs;
- Execução de diferentes tarefas;
- Elaboração de uma comunicação adequada aos usuários;
- Adaptação personalizada ao ambiente.

Perguntas



Robôs Socialmente Assistivos

❖ Em um projeto de SARs, cinco principais categorias devem ser avaliadas (Nestorov et al. 2014):

- **aparência:** a escolha da aparência física do robô conforme sua aplicação;
- **modalidade de interação:** conter múltiplas modalidades interativas;
- **interação inteligente:** comportamento adaptativo e específico do robô;
- **capacidade de tarefa:** assistência emergencial e suporte de tarefas;
- **modo de operação:** capacidade de adaptação dos robôs às mudanças do ambiente e necessidades do usuário.

Robôs Socialmente Assistivos

❖ Considerando a aparência física, os robôs possuem as seguintes classes (Martinez-Martin et al. 2020):

- **androide:** Robôs parecidos fisicamente com humanos; ;
- **mascote:** possui formas humanoides, mas aparências abstratas ou caricaturais;
- **mecânico:** formas humanoides com partes visivelmente mecânicas;
- **animais:** aparência física similar a animais de estimação;
- **não humanoide:** não há semelhança com nenhum ser vivo.

Robôs Socialmente Assistivos

❖ Exemplos SARs e suas classes conforme exposto em (Martinez-Martin et al. 2020):



Mascote: Robô Ifbot



Mecânico: Robô GIPY-1

Animal: Robô PLEO



Não-Humanoide: Robô LABO-1

Androide: Robô Zeno R-50

SARs para terapia de TEA

- ❖ Em terapias robóticas para o tratamento de indivíduos com autismo, um robô com forma humana é considerado o mais adequado na aplicabilidade em sessões terapêuticas (Ricks and Colton 2010);

- ❖ Recomenda-se que o tamanho do robô seja apropriado ao tamanho da criança com objetivo de facilitar o contato visual e, além disso, as expressões faciais dos SARs não devem ser intimidantes durante o processo de intervenções (Costa et al. 2017, Dickstein-Fischer et al. 2018).

SARs para terapia de demência

- ❖ Um projeto de SARs responsável por auxiliar no tratamento de demência, os robôs, além da interação personalizada, julga-se necessário que os robôs sejam preparados para redução de estímulos desnecessários durante a sessão de terapia, tais como ruídos e brilhos excessivos.

Robôs Comerciais

- O robô **NAO** é comumente utilizado e tornou-se um padrão em educação e pesquisa, ainda assim, também é utilizado como assistente por empresas e centros de saúde para receber, informar e proporcionar entretenimento aos visitantes.



Figura 2: Robô Androide NAO apresentado no trabalho de [Erden 2013]

Robôs Comerciais

- O **Moxie**, classe de robô mascote, foi projetado para estabelecer um relacionamento com crianças e, por meio dessa interação é possível efetuar o reconhecimento e o ensino de diferentes emoções. A interação de crianças com o Moxie desenvolve habilidades essenciais, como regulação emocional e gestão de relacionamentos, especialmente em crianças com autismo (Xiao et al. 2020, Cano et al. 2021).



Figura 3: Aparência física do robô Moxie. Fonte: Embodied - <https://embodied.com/>

Robôs Comerciais

- O robô **QT** fornece uma configuração amigável e eficaz responsável por manter a atenção das crianças e promover o ensino de várias habilidades. O trabalho de [Costa et al. 2017] utilizou o QTRobot para proporcionar o treinamento das habilidades emocionais em crianças com TEA.

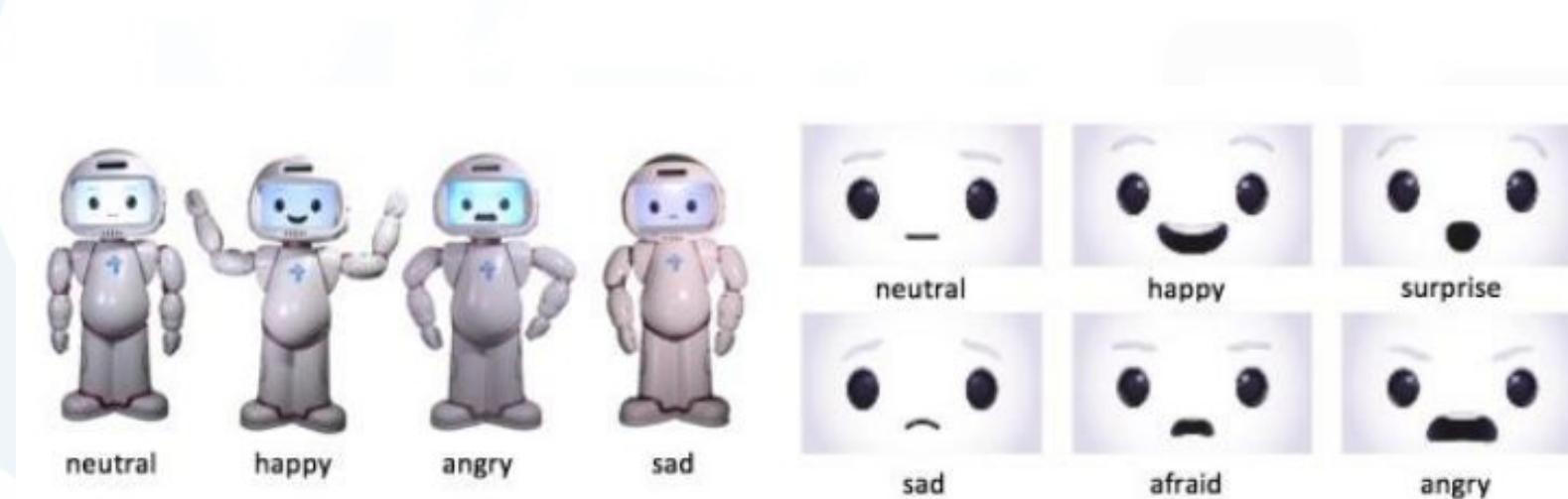


Figura 4: Robô Androide QT e suas expressões para diferentes emoções.
Fonte: (Costa et al., 2017)

Robôs Comerciais

- Outro robô denominado Pepper, possui quase as mesmas articulações de um humano e uma aparência amigável, fornece assistência as pessoas. Considerando promover habilidades sociais, o robô Pepper já foi utilizado para desenvolver um sistema de apoio à aprendizagem para comunicação e interação personalizada ao diálogo com indivíduos com TEA [Yabuki and Sumi 2018]



Figura 5: Robô Androide Pepper apresentando no trabalho de [Yabuki and Sumi 2018]

Robôs Comerciais

- Projetado especificamente para fins terapêuticos e bastante utilizado em terapias para pacientes com demência, o robô Paro, possui uma aparência semelhante a uma foca bebê, sendo assim pertence a classe de robô tipo animal [Wada et al. 2008].
- Uma análise feita sobre a utilização do robô Paro, constatou que a interação humano-robô estimula o lobo frontal do cérebro de idosos com demência, bem como a alteração dos seus estados emocionais [Shibata 2012].



Figura 6: Características do Paro. Fonte: [Shibata 2012]

Robôs Acadêmicos

- O Probolino é um dispositivo social robótico portátil de baixo custo com interação através do toque. Parece com um bicho de pelúcia, possui orelhas animadas, olhos, sobrancelhas, pálpebras, boca e pescoço. Uma tela sensível ao toque é localizada em sua barriga. Assim sendo, este robô é capaz de se comunicar e interagir com humanos, expressar atenção e emoções através do olhar e expressões faciais [Saldien et al. 2010].



Figura 7: Robô Probolino. Fonte: [Vanderborght et al. 2012]

Robôs Acadêmicos

- Com aparência de um brinquedo, temos o robô Bliss. A utilização do robô Bliss em atividades de interação para terapia do autismo fornece uma melhor interação social para crianças com TEA.
- No entanto, devido aos diferentes níveis de autismo, julga-se necessário o desenvolvimento de conteúdos adaptados, para crianças em vários níveis de desenvolvimento, que promovam maior grau de interação durante as sessões de terapia [Santatiwongchai et al. 2016].



Figura 8: Robô Mascote Bliss. Fonte: (Attawibulkul et al. 2019)

Robôs Acadêmicos

- Por fim, um outro modelo de robô, conhecido por Troy, com aparência mecânica. Esse robô possui mais de 30 ações pré-programadas disponíveis para o terapeuta aplicar em sessões de terapia para autismo [Manohar et al. 2011].



Figura 9: Robô Troy. Fonte: [Ricks et al. 2010]

Comparação entre SARs

- ✓ Dentre as características dos robôs discutidos, observa-se que a aparência física do robô desempenha um papel fundamental no projeto de um SAR e no contexto de suas aplicações.
- ✓ Para o desenvolvimento de sessões terapêuticas, robôs humanoides são considerados mais eficientes [Ricks et al. 2010, Ueyama 2015, Amanatiadis et al. 2017].
- ✓ Entretanto, na literatura revisada ainda não há um consenso em relação à aparência física no projeto de SARs.

Comparação entre SARs

Tabela 1 Análise comparativa dos robôs comerciais e acadêmicos

Robô	Uso	Forma	Aplicação em Saúde	Referência
NAO	comercial	androide	Demência e TEA	[Erden 2013]
Moxie	comercial	mascote	Desenvolvimento social e emocional	[https://embodied.com/]
QTrobot	comercial	androide	TEA	[Costa et al. 2017]
Pepper	comercial	androide	TEA	[Yabuki and Sumi 2018]
Paro	comercial	animal	Demência	[Wada et al. 2008]
Probolino	acadêmico	mascote	TEA	[Cao et al. 2015]
Bliss	acadêmico	mascote	TEA	[Attawibulkul et al. 2019]
Troy	acadêmico	mecânico	TEA	[Ricks et al. 2010]
EVA	acadêmico	mascote	Demência e TEA	[Rocha et al. 2021]

Perguntas



Robô EVA

Robô EVA

- Desenvolvido no **CICESE** (Centro de Investigación Científica y Educación Superior de Ensenada)
- Baja California, México
- Desenvolvido a partir da tese de doutorado [Cruz Sandoval 2020]
- Estendido em outros trabalhos [Mitjans 2020] e [Rocha et al. 2021]

Robô EVA (Primeira versão)

- Feito em papelão
- Raspberry PI 3
- Caixa de som
- Placa **Matrix Voice** (8 microfones e 18 LEDs RGB)
- Seis animações nos LEDs



Robô EVA (Primeira versão)

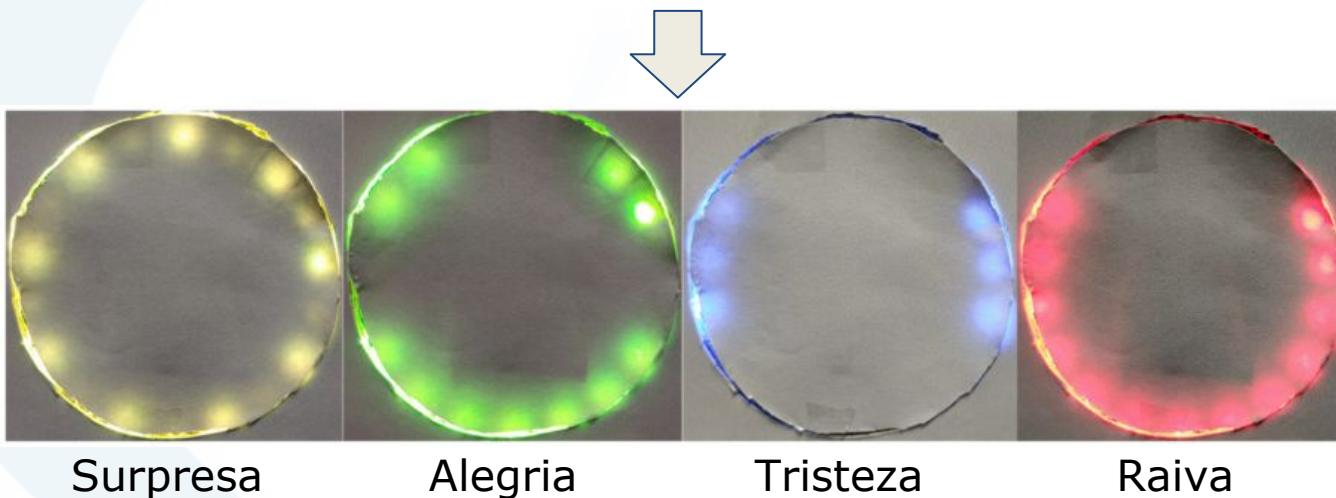
Mecanismo de Comunicação **VERBAL**

Audição

Fala

Robô EVA (Primeira versão)

Mecanismo de Comunicação **NÃO VERBAL**



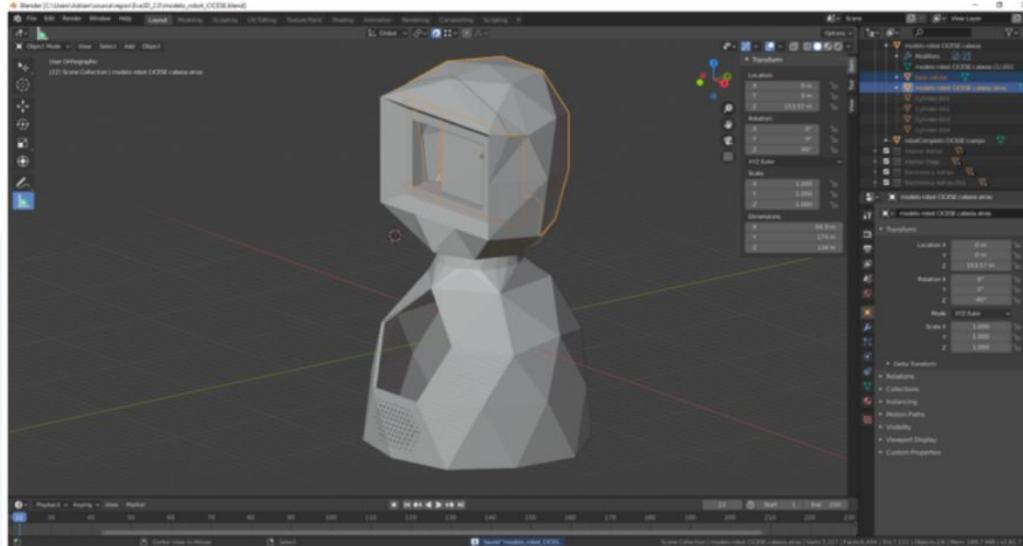
Robô EVA (Segunda versão)

- Projetado no software **Blender**
- Corpo impresso em **3D** (Filamento PLA)
- Mais dois mecanismos de comunicação
NÃO VERBAL
- Smartphone como Display

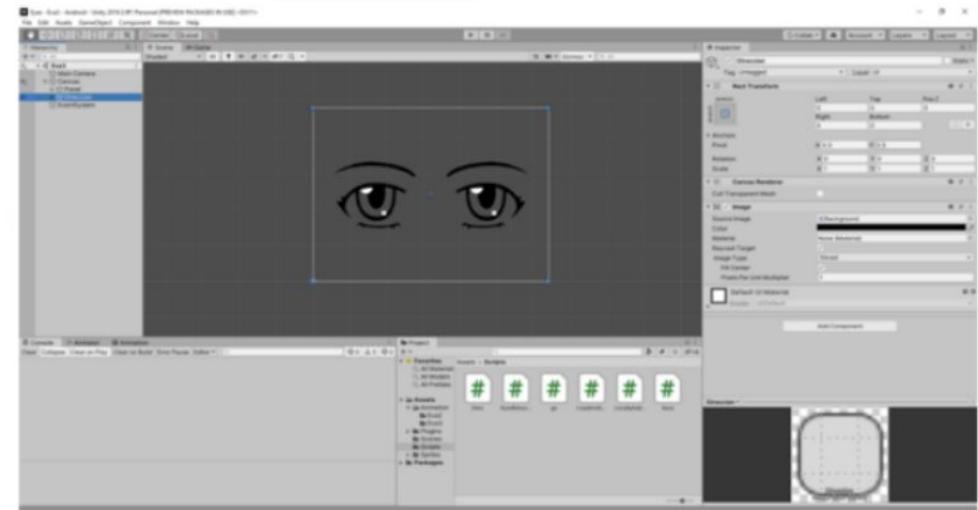


* PLA - Termoplástico biodegradável

Robô EVA (Segunda versão)



(a) Projeto do corpo do robô no software *Blender*



(b) Projeto dos olhos do robô no software *Unity 3D*

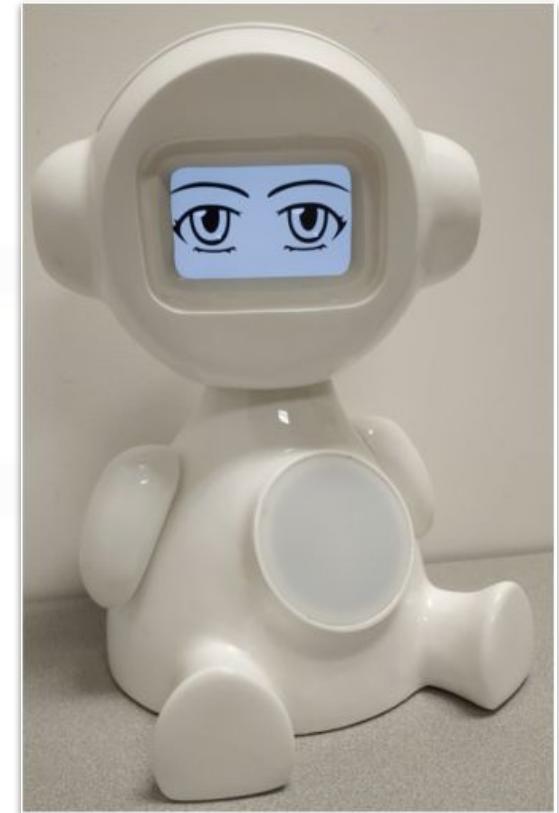
Robô EVA (Segunda versão)

- Dois novos mecanismos de comunicação **NÃO VERBAL**
 - **Olhos**
 - **Movimentação da cabeça**
- Uso dos LEDs
- **Upgrade** do Raspberry PI 3 para a versão 4



Robô EVA (Terceira versão)

- Projetado no software **Blender**
- Fabricada em fibra de vidro
- Herda todas as características *funcionais* e de *hardware*
- Design **antropomórfico** (*pernas e braços*)



Robô EVA (Versão atual)

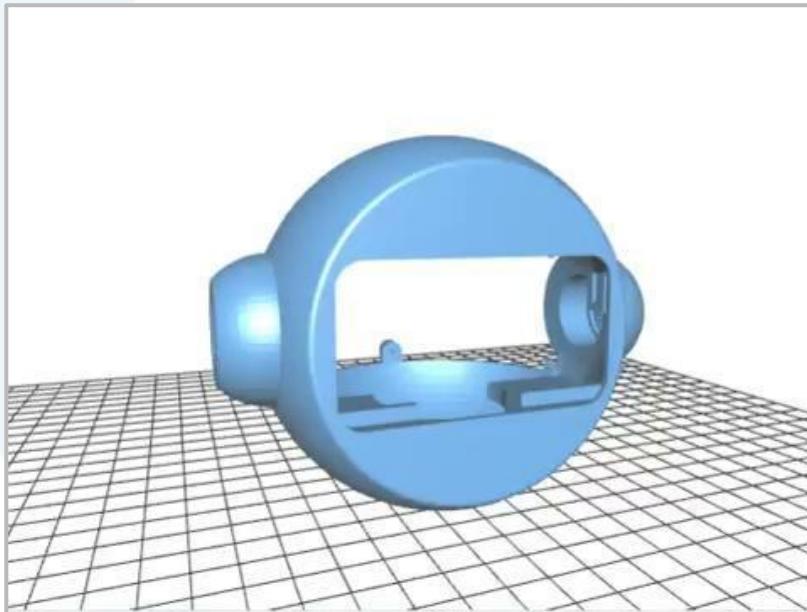


- **E.V.A** (*Embodied Voice Assistant*)
- Projeto de robótica *open-source*
- Pesquisa em Interação Humano-Robô
- Corpo impresso em **3D** (PLA)
- Arquivos do repositório do **EVA**

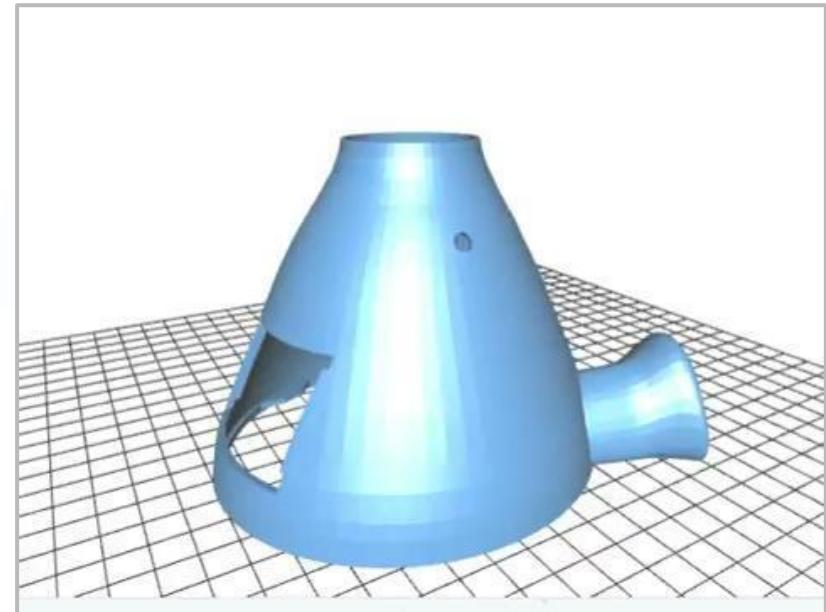
<https://github.com/eva-social-robot>

Robô EVA (Versão atual)

Arquivos para Impressão 3D



<https://drive.google.com/file/d/1h2tlCJFJuZuRUTdxE2Z4AINzIJNegEk5/view?usp=sharing>



<https://drive.google.com/file/d/1WJjmpCMm6yrY3rnzI8-obtPJUGBIQZt4R/view?usp=sharing>

Perguntas?

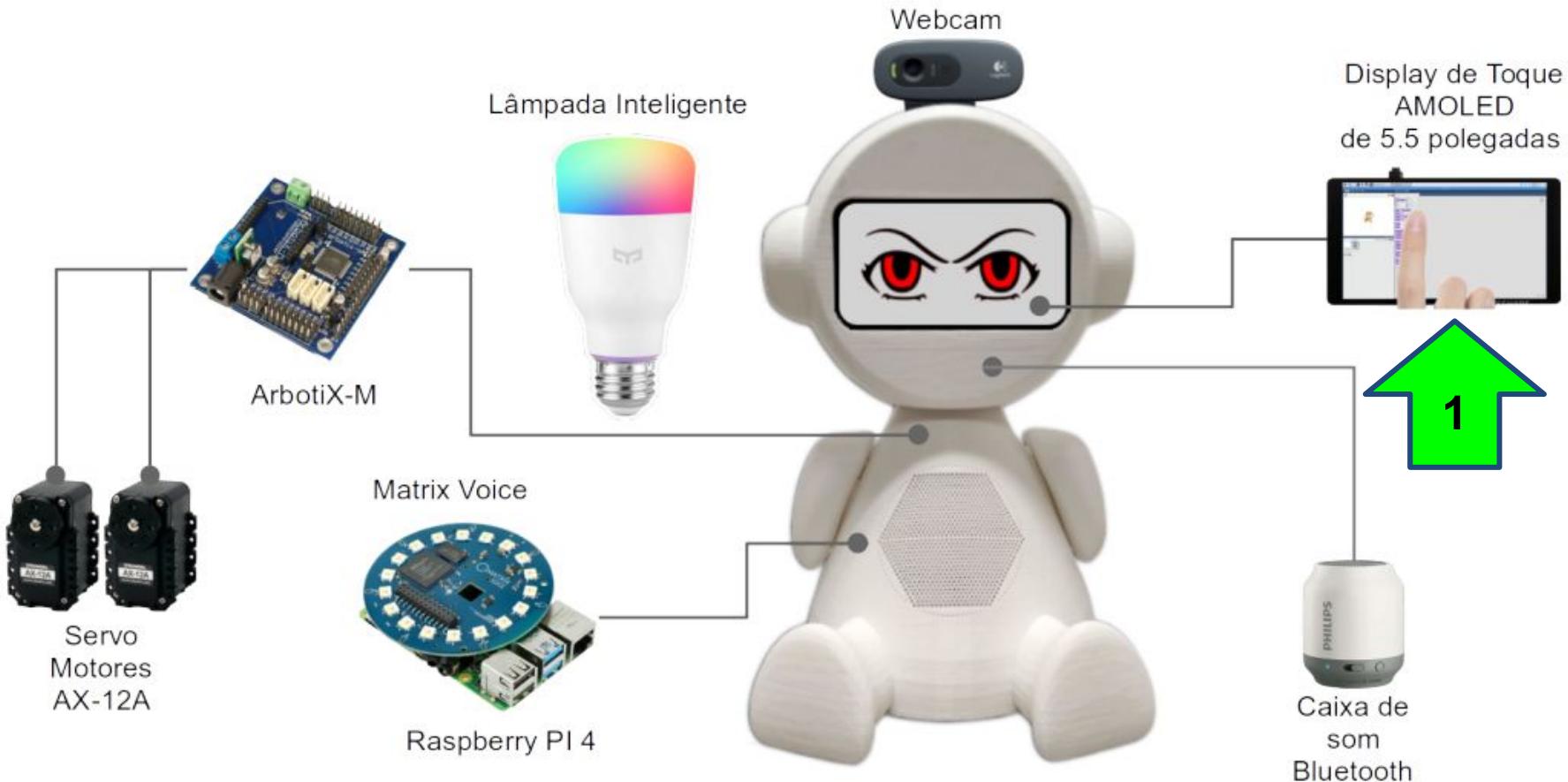
Arquitetura do Robô EVA

(Componentes de Hardware)

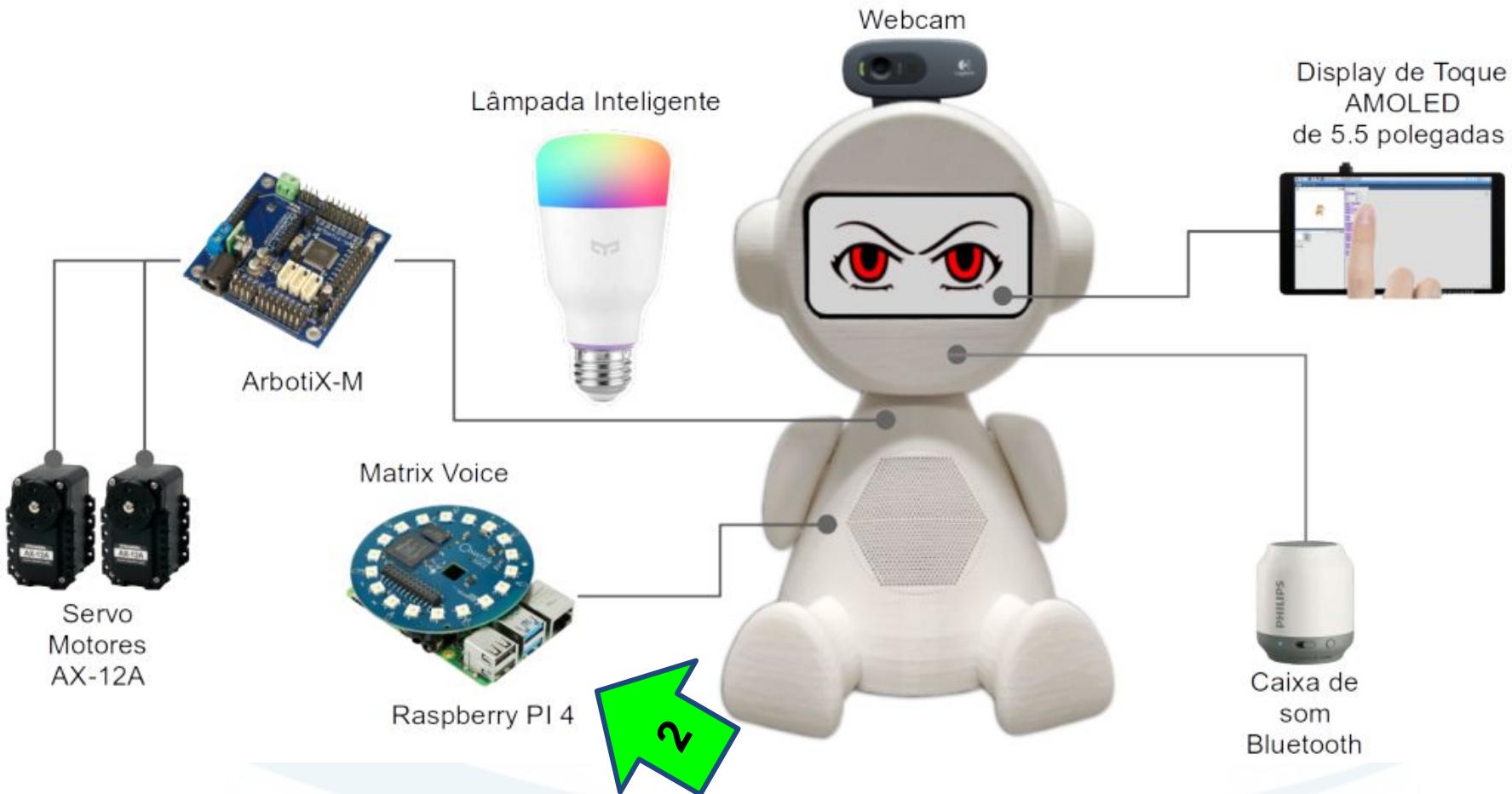
Robô EVA (Versão atual)

- Tela de toque de **5.5** de polegadas
- Raspberry PI 4
- Placa **ArbotiX-M**
- Dois servomotores *Dynamixel AX12A*
- Caixa de som bluetooth
- Placa *Matrix Voice*
- Webcam
- Lâmpada Inteligente Xiaomi

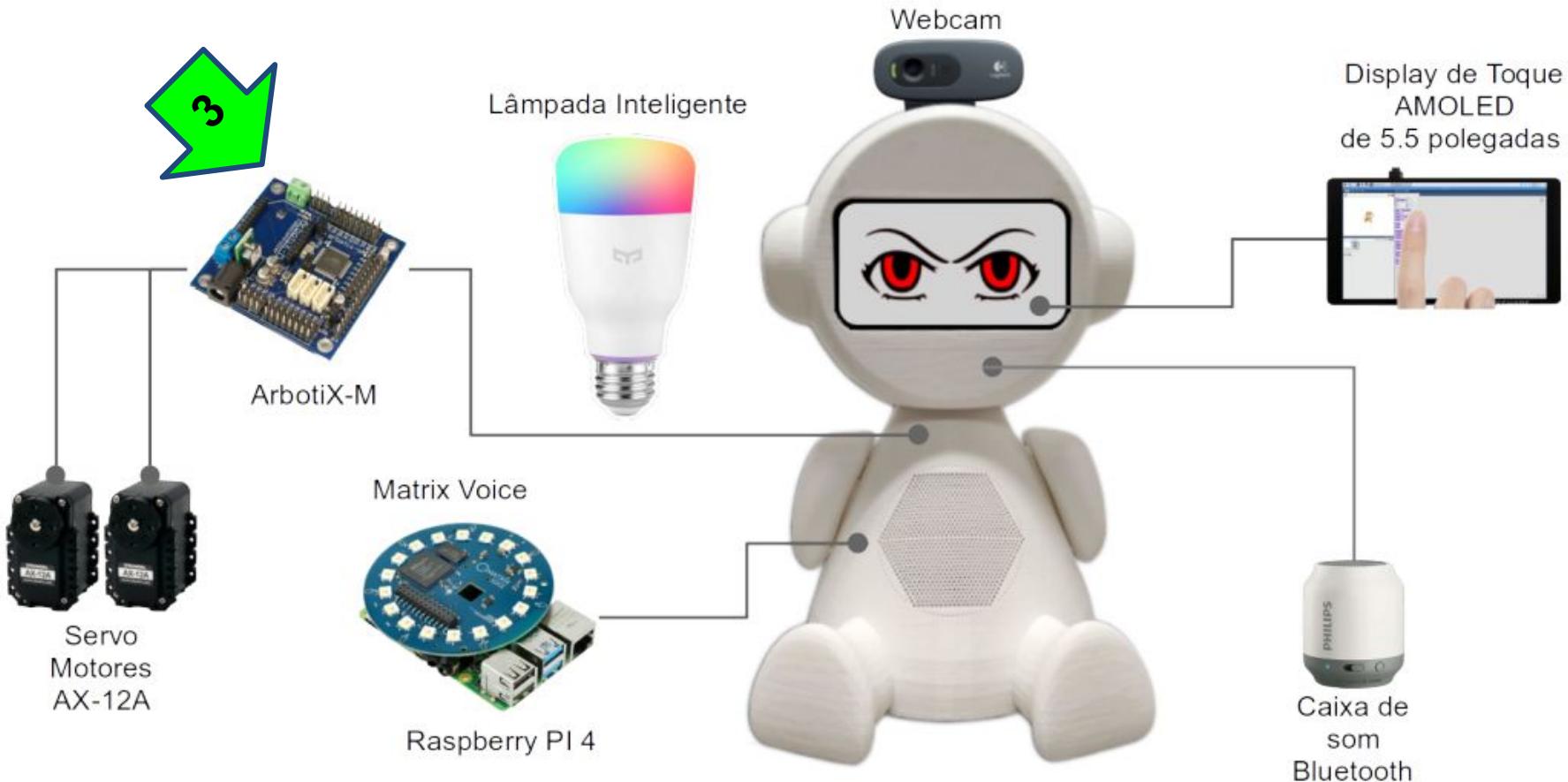
Robô EVA (Componentes de Hardware)



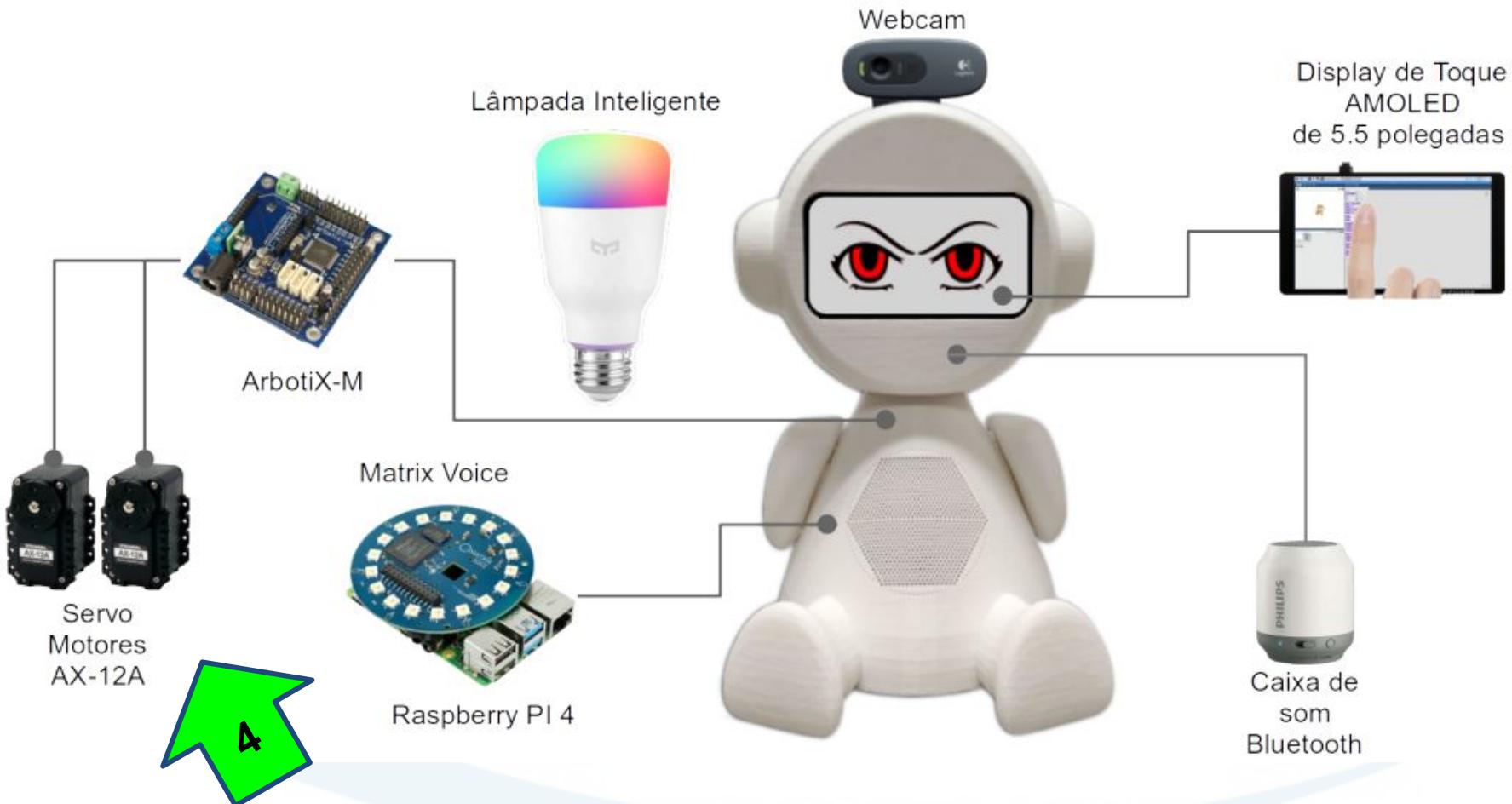
Robô EVA (Componentes de Hardware)



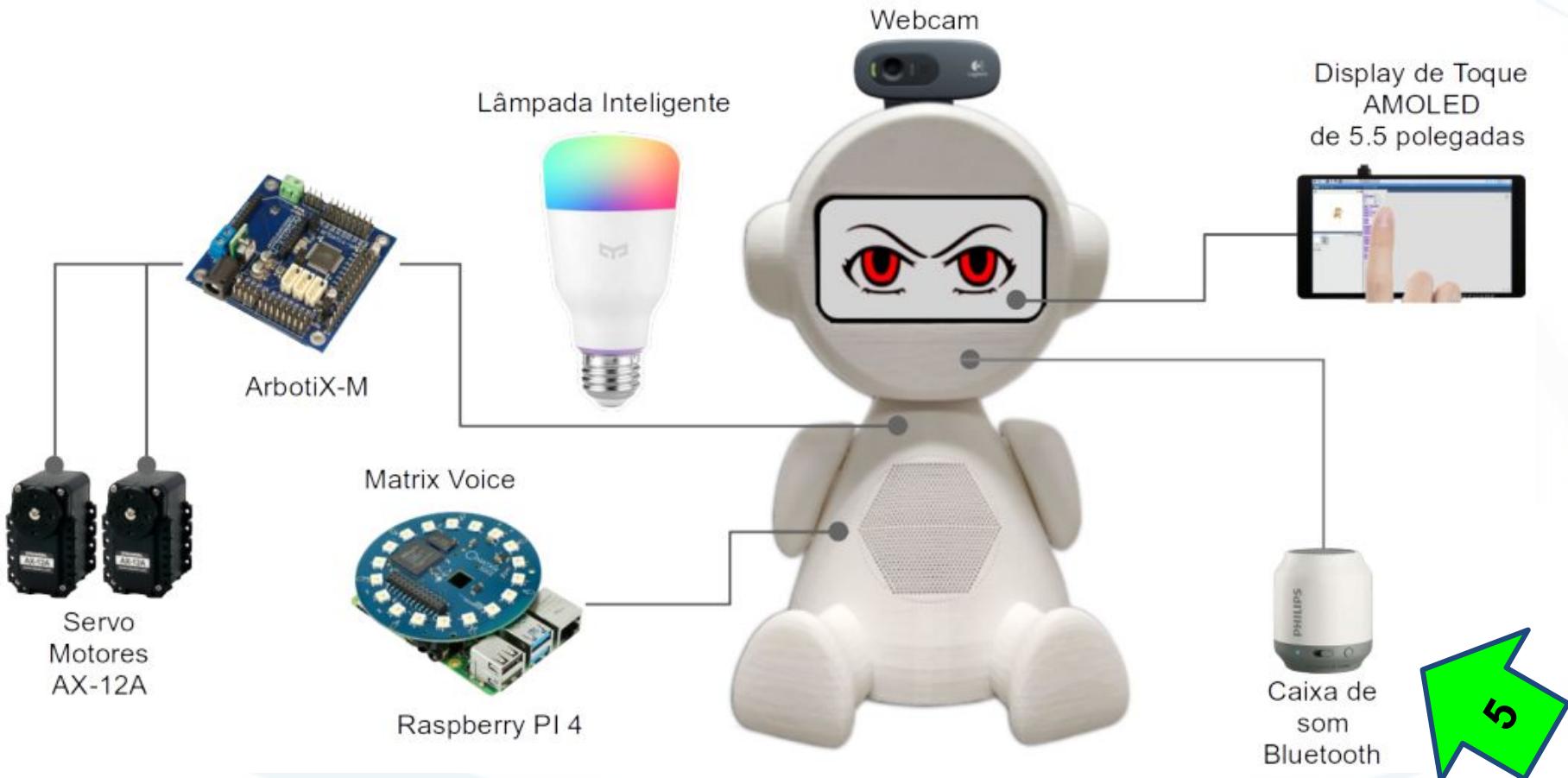
Robô EVA (Componentes de Hardware)



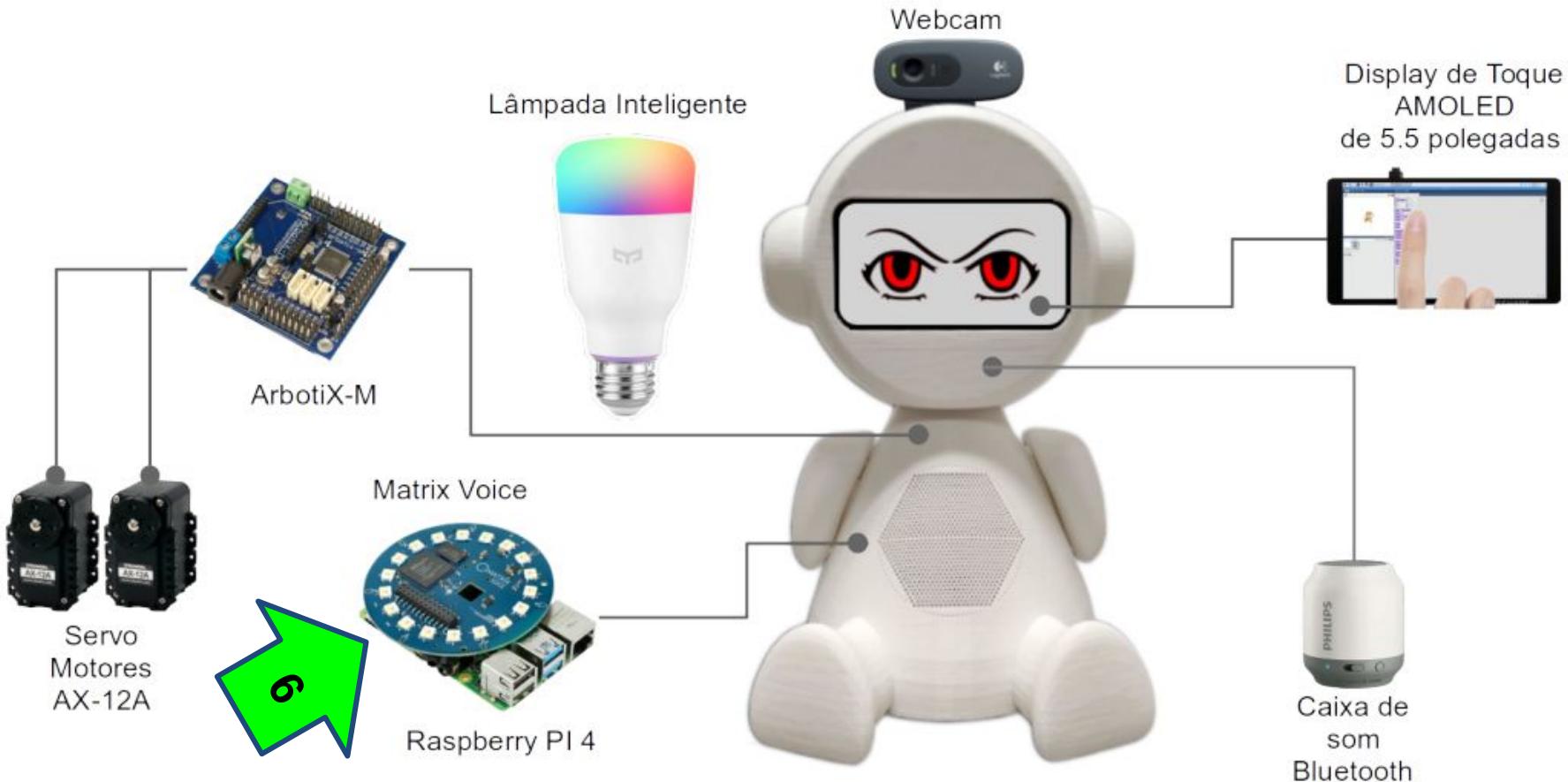
Robô EVA (Componentes de Hardware)



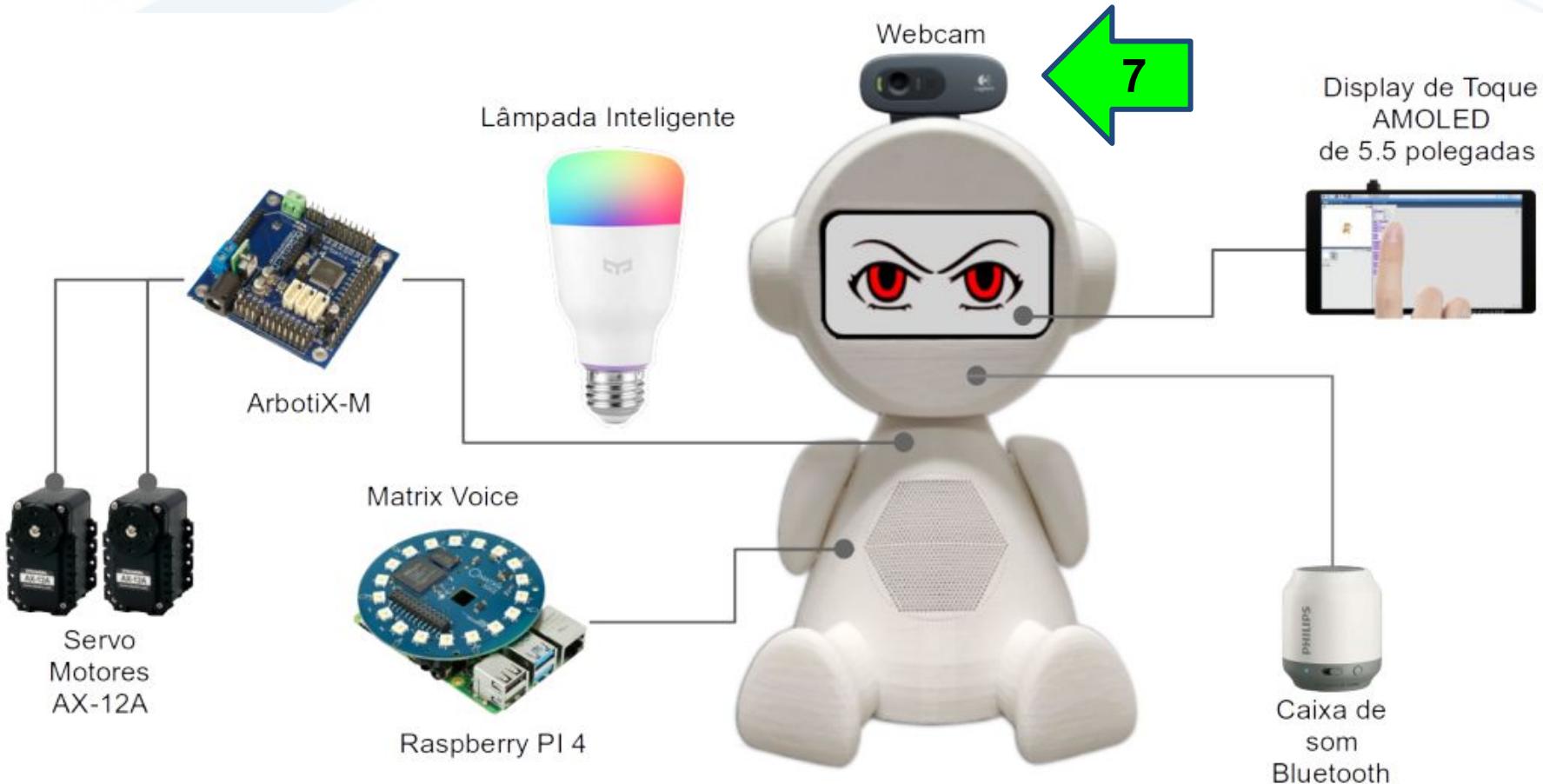
Robô EVA (Componentes de Hardware)



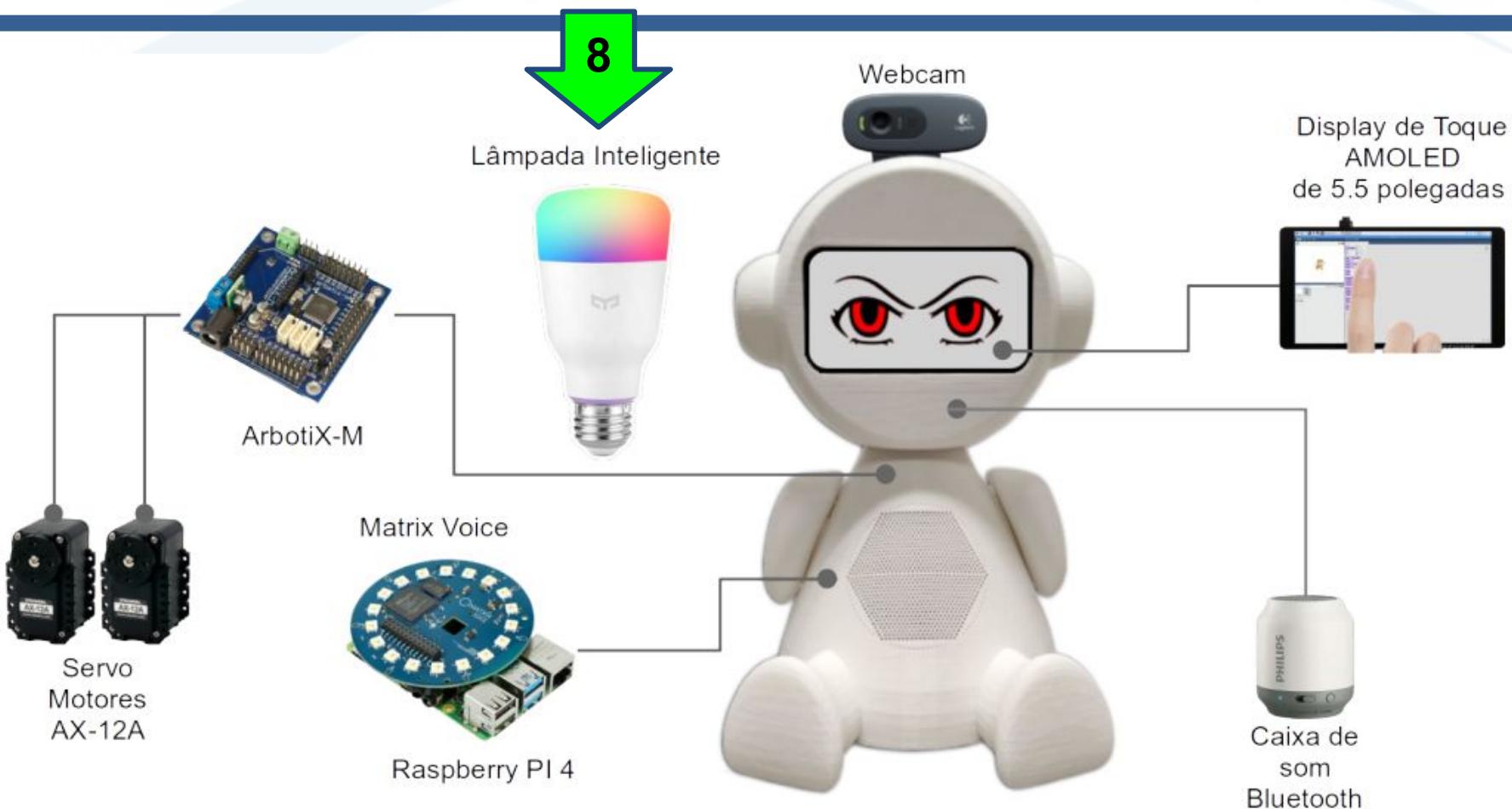
Robô EVA (Componentes de Hardware)



Robô EVA (Componentes de Hardware)



Robô EVA (Componentes de Hardware)

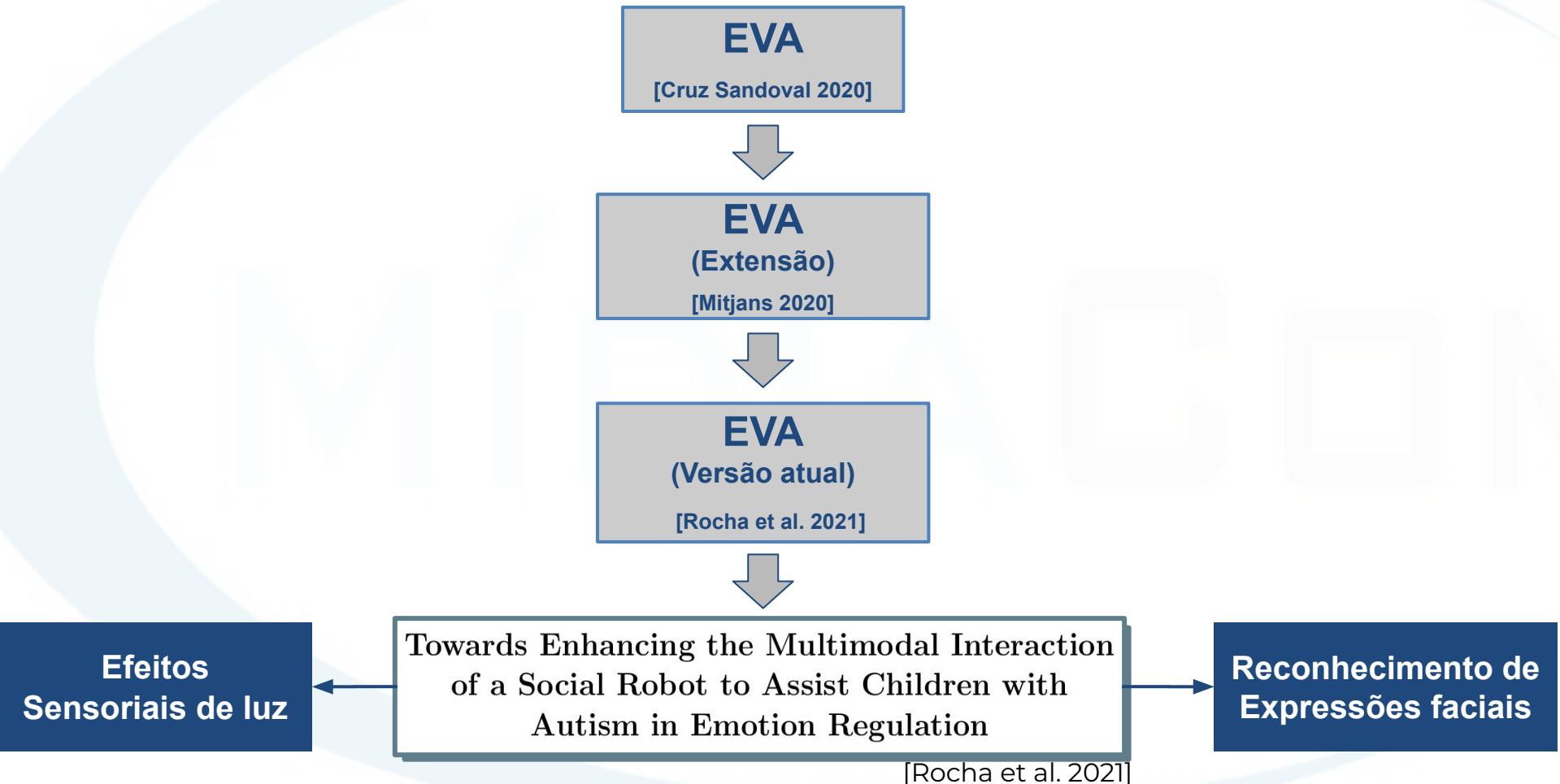


Robô EVA (Outras versões)



Uma sessão de terapia de estimulação cognitiva com três participantes sendo conduzida pelo robô Eva [Cruz-Sandoval et al. 2020]

Robô EVA



Robô EVA (Tabela comparativa)

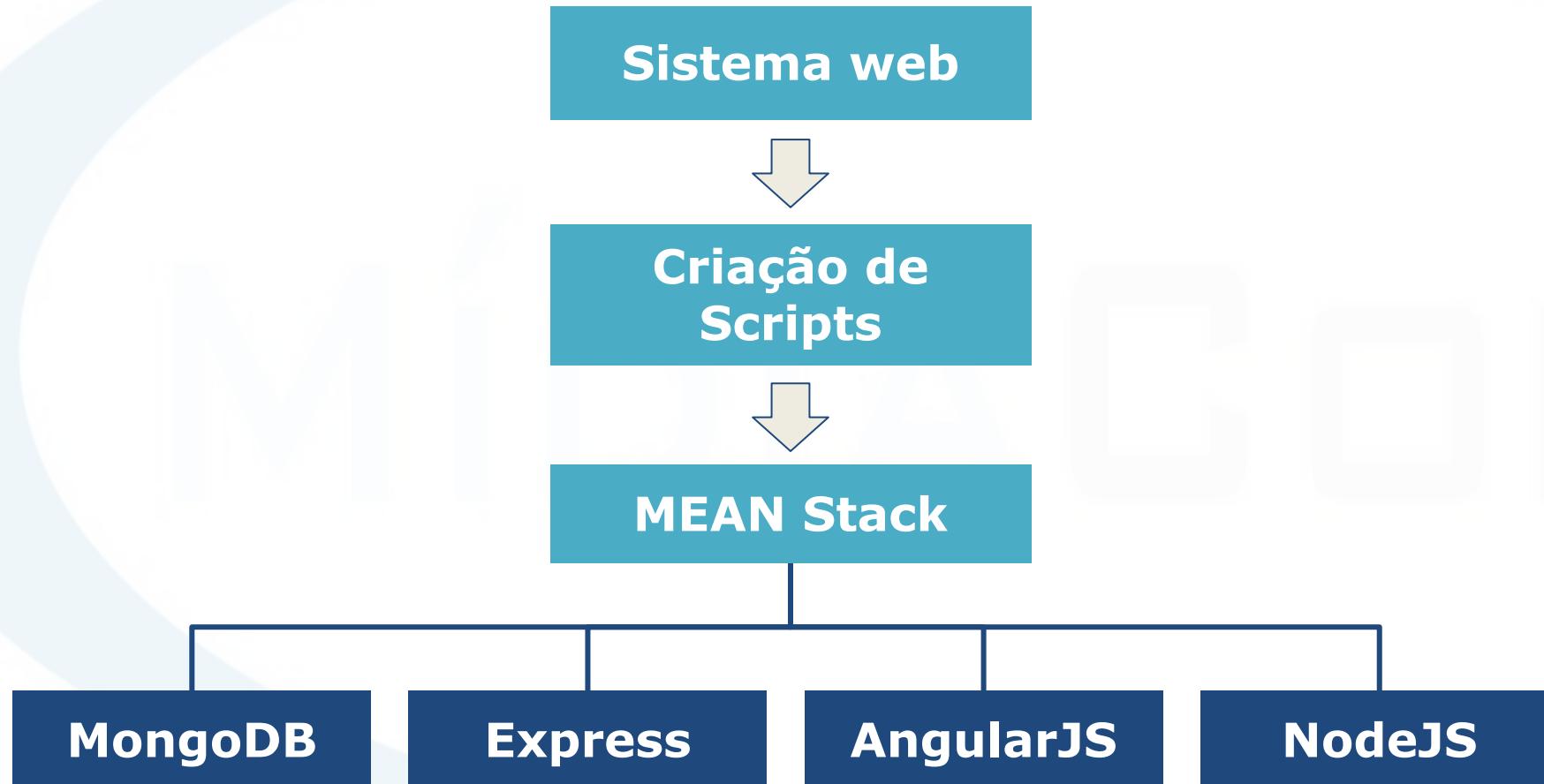
	Versão 1	Versão 2	Versão 3	Versão Atual
Material	Papelão	PLA	Fibra de Vidro	PLA
Placa de controle	Raspberry PI 3	Raspberry PI 4	Raspberry PI 4	Raspberry PI 4
Sensores	Matrix Voice	Matrix Voice	Matrix Voice	Matrix Voice
Atuadores	Caixa de som	Caixa de som, servomotores	Caixa de som , servomotores	Caixa de som , servomotores
Graus de liberdade (DoF)	0	2	2	2
Aparência	Caricatura	Caricatura	Antropomórfica	Antropomórfica
Mecanismo de comunicação não verbal	Luzes LED	Luzes LED, Olhos, Mov. da cabeça	Luzes LED, Olhos, Mov. da cabeça	Luzes LED, Olhos, Mov. da cabeça
Efeitos sensoriais de luz	-	-	-	Lâmpada inteligente Xiaomi
Reconhecimento de expressões faciais	-	-	-	Webcam

Tabela comparativa entre os modelos do robô **EVA**

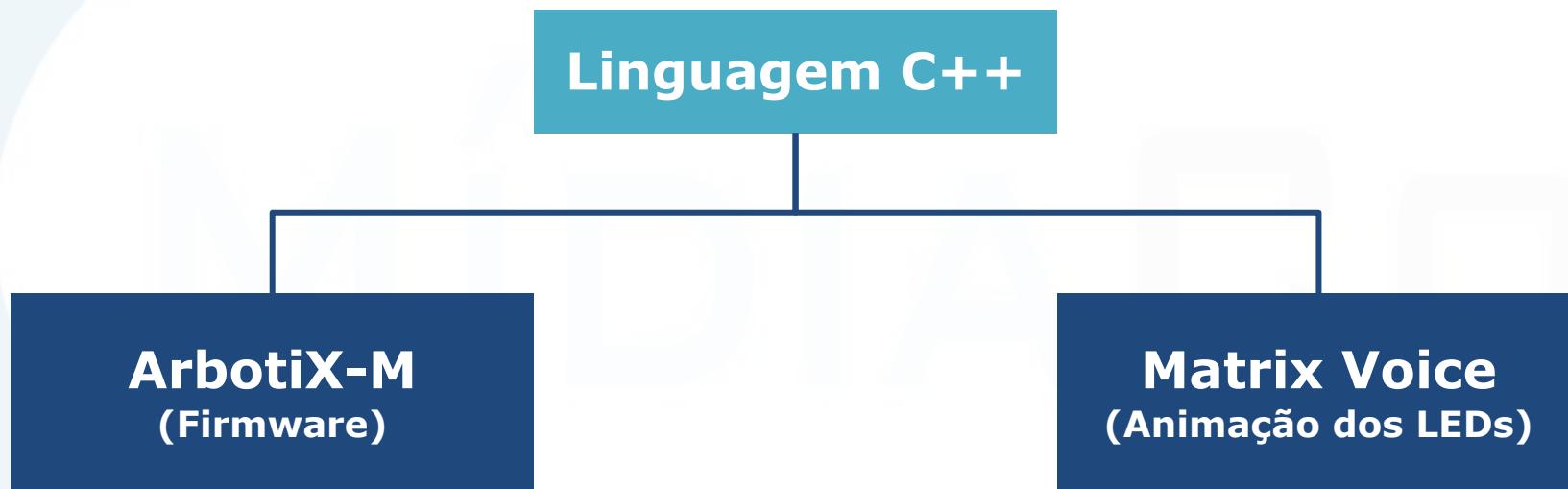
Arquitetura do Robô EVA

(Componentes de Software)

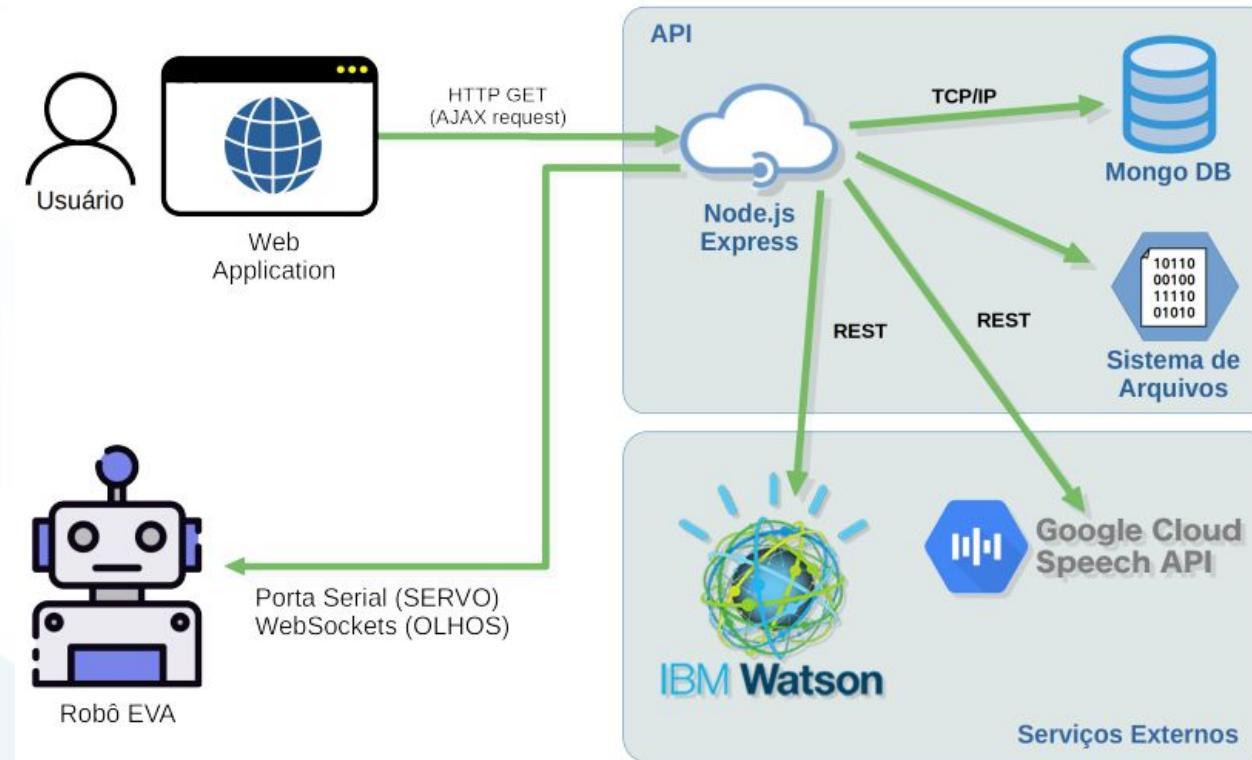
Robô EVA (Componentes de software)



Robô EVA (Componentes de software)



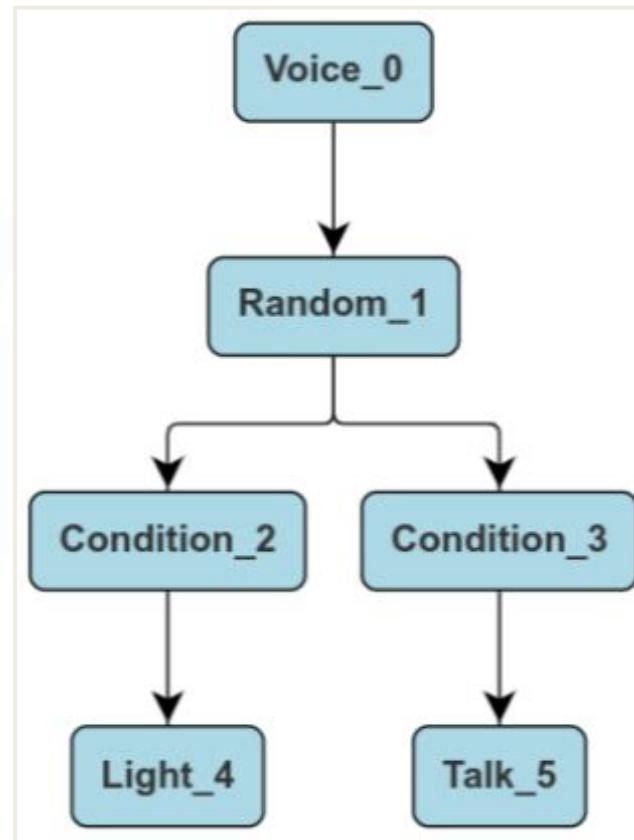
Robô EVA (Arquitetura de software do robô EVA)



Robô EVA (Criando scripts para o EVA)

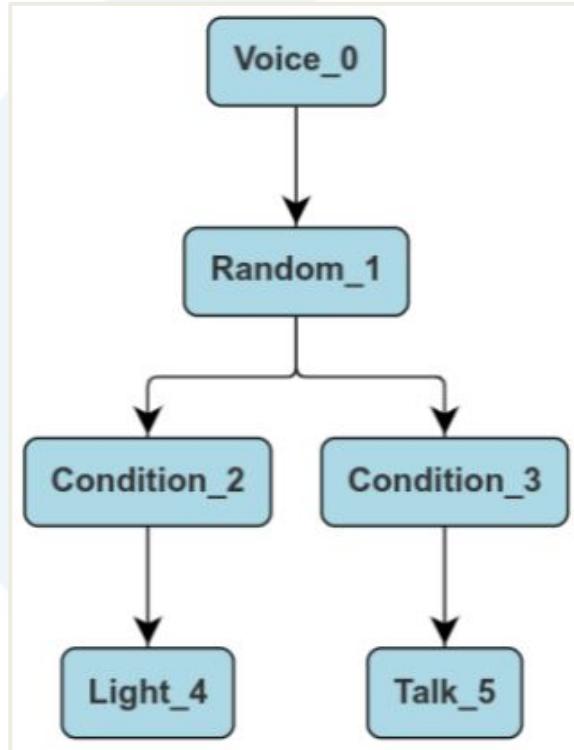
- Linguagem de Programação Visual (VPL)
- Scripts definem as sessões de terapias
- Framework web **GoJS**
- Ferramenta de arrastar e soltar
- Fácil utilização por pessoas não técnicas

Robô EVA (Criando scripts para o EVA)



Um Script na **VPL**

Robô EVA (Criando scripts para o EVA)



Um Script na **VPL**

- Grafo de interação
- Os **nós** representam comandos do robô, elementos da linguagem, estruturas condicionais etc.
- As **arestas** determinam o fluxo de execução do script (de cima para baixo ou da esquerda para a direita)

Perguntas?

Robô EVA

(Montagem Física)

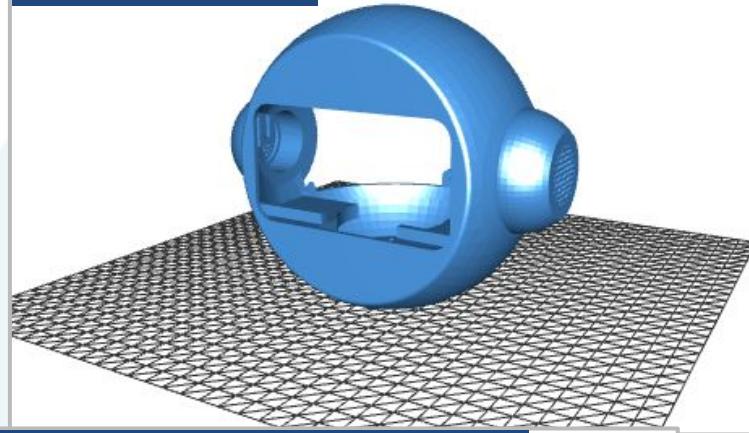
Robô EVA (Montagem física)

- Visão geral do processo de montagem
- Imagens reais (nossas) das conexões
- Configuração dos *IDs* dos servomotores

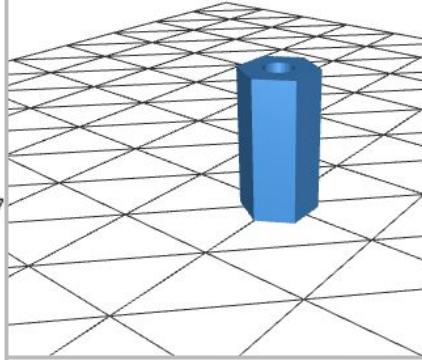
<https://github.com/eva-social-robot>

Robô EVA (Montagem física)

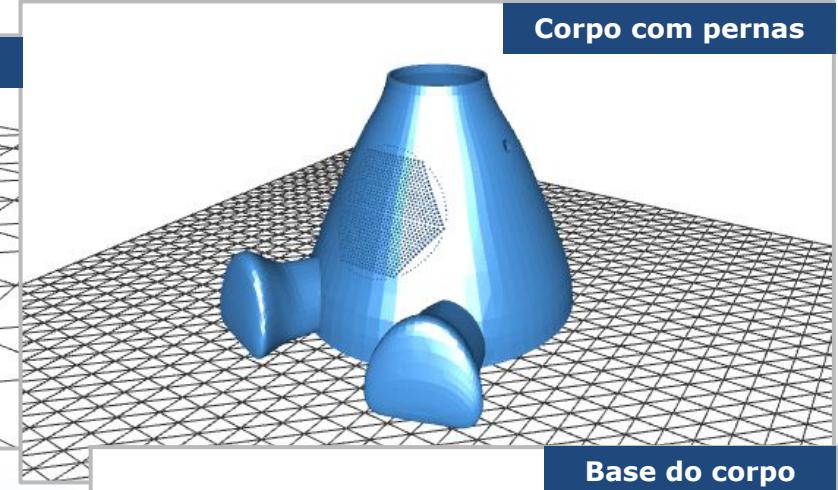
Cabeça (Frente)



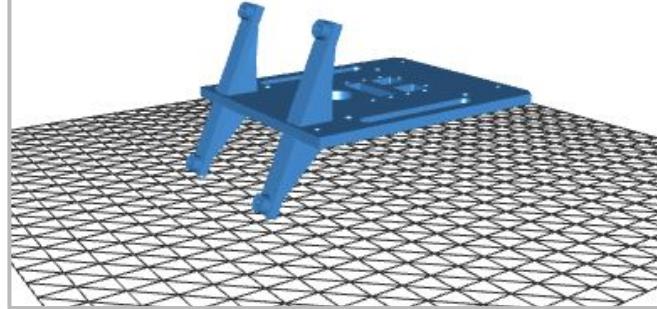
Suporte para ArbotiX-M



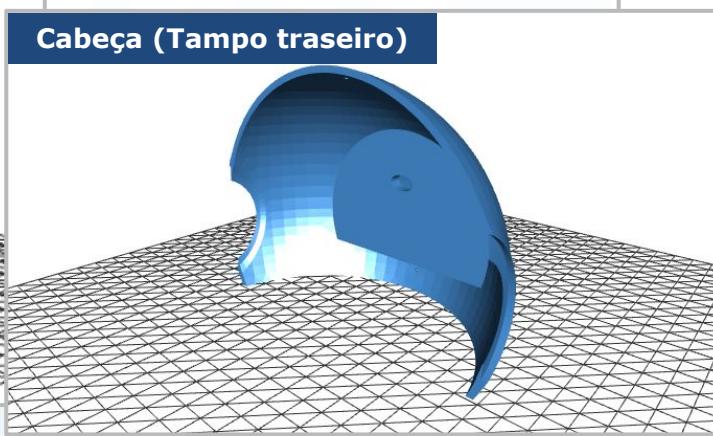
Corpo com pernas



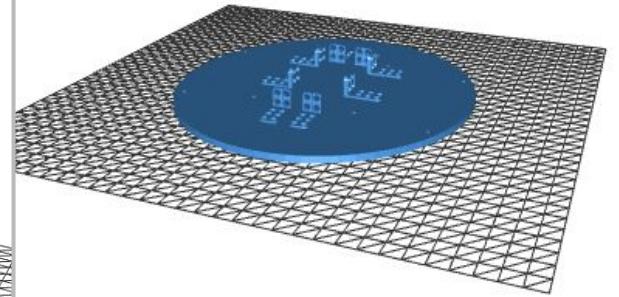
Base para os motores e pescoço



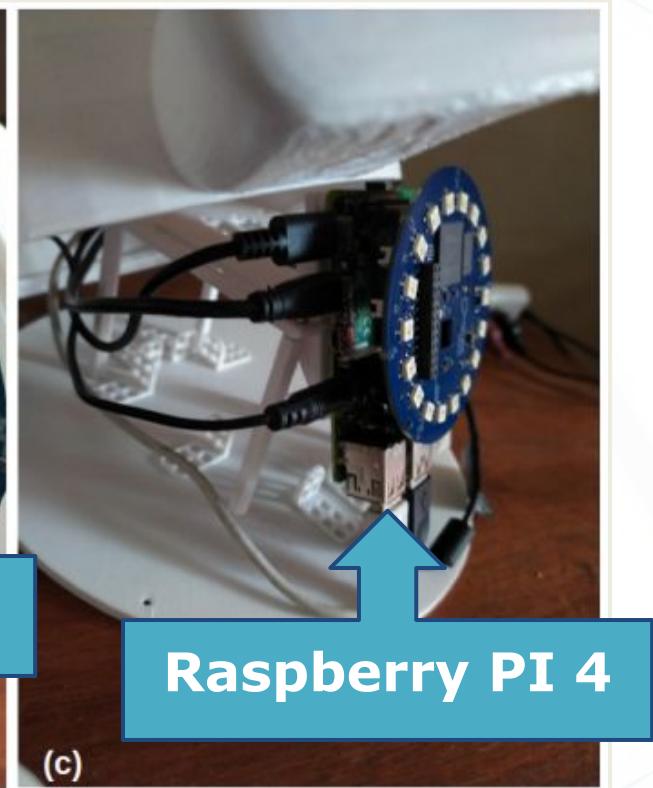
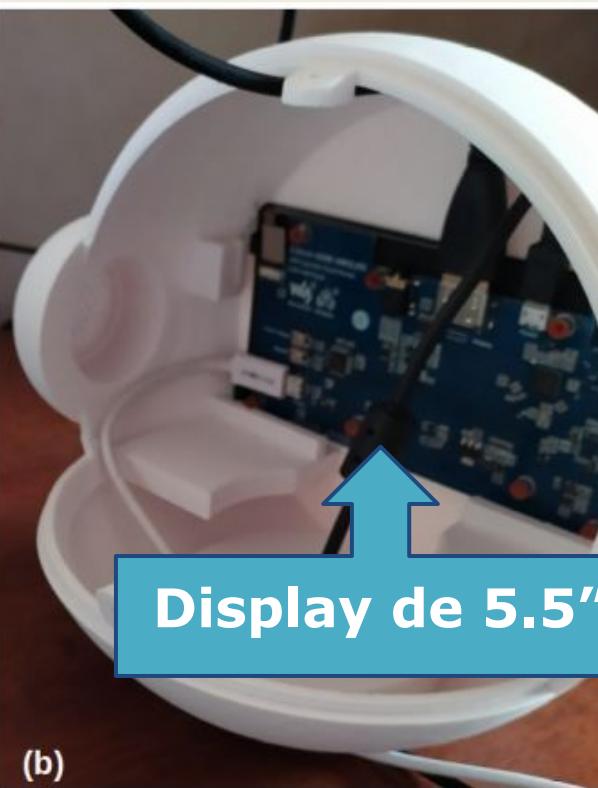
Cabeça (Tampo traseiro)



Base do corpo

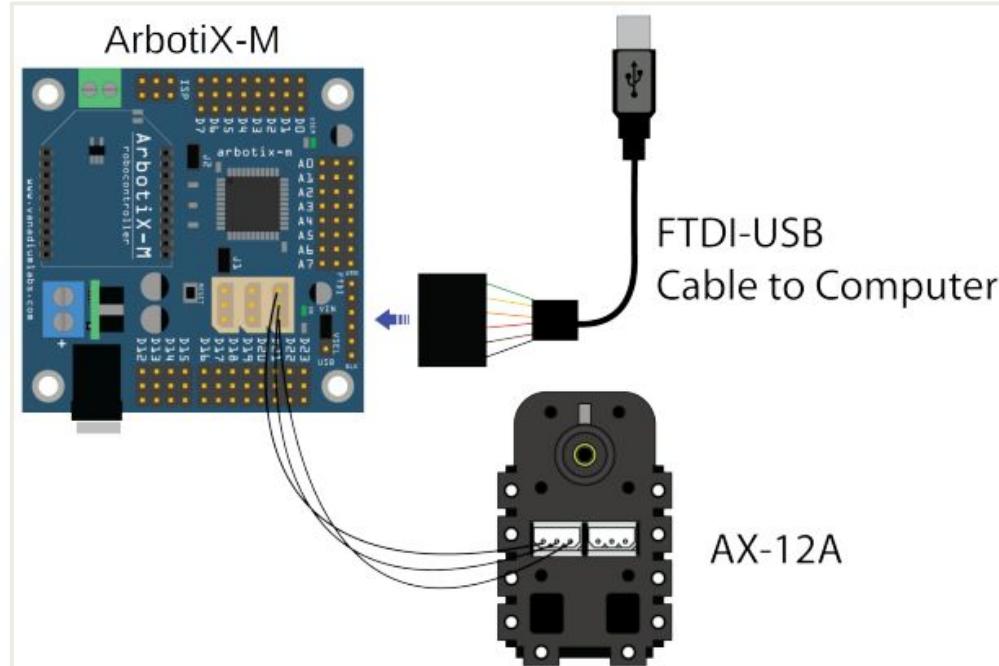


Robô EVA (Montagem física)



(a) Visão frontal da placa *Matrix Voice* com os 18 LEDs RGB, (b) Montagem do display de toque de 5.5" e (c) *Matrix Voice* acoplada ao *Raspberry*

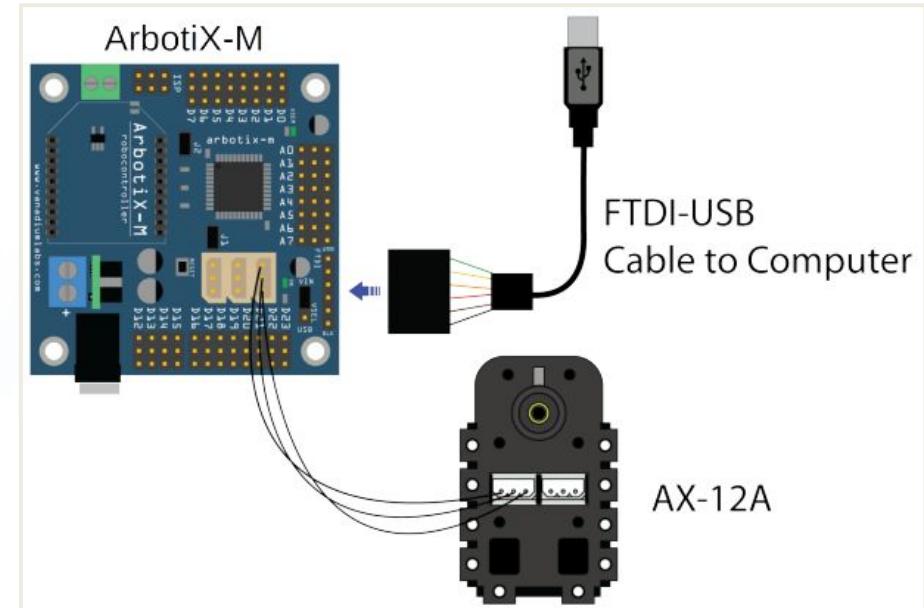
Robô EVA (Montagem física)



Placa **ArbotiX-M**, Servomotor **AX-12A** e um cabo FTDI-USB

Robô EVA (Montagem física)

- A placa **ArbotiX-M** controla os servomotores
- Se conecta ao Raspberry PI através do cabo **FTDI-USB**
- Pode controlar até 3 servomotores
- Tipo de conexão *DaisyChain*



Robô EVA (Alterando os IDs dos Servomotores)

```
#include <ax12.h>
#include <BioloidController.h>

void setup()
{
    // param1 = ID do servo
    // param2 = n. do registrador
    // param3 = novo ID

    ax12SetRegister(1, 3, 2);
}

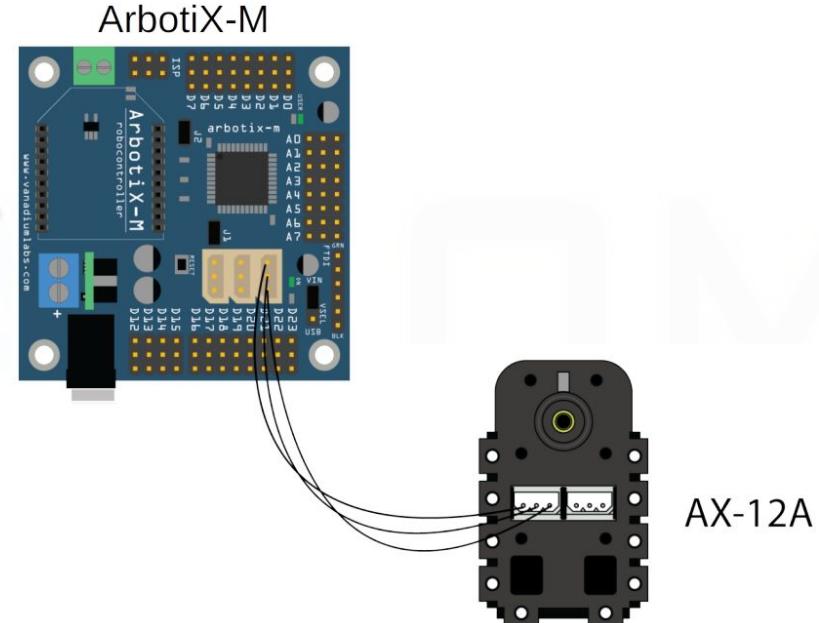
void loop() {}
```

Código para alterar o ID do servomotor
AX-12A

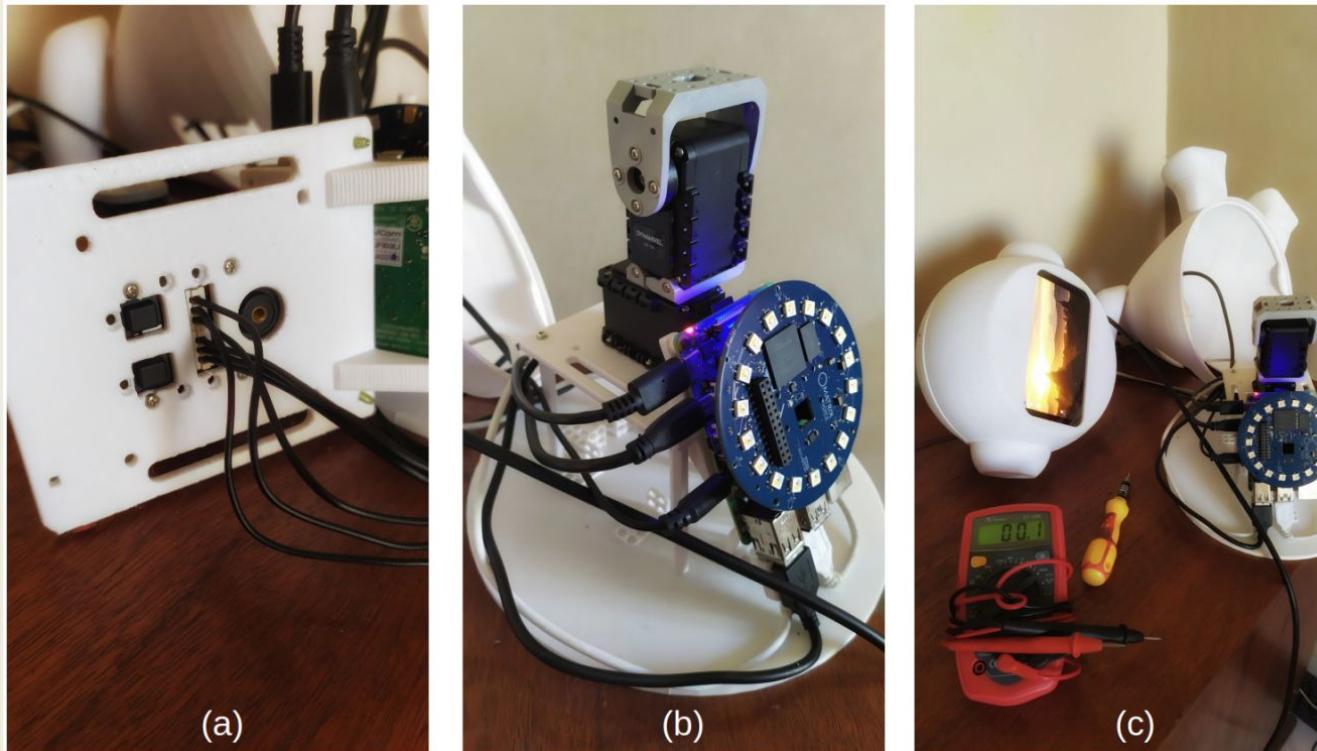


001100111100100111001

Transferência de Firmware



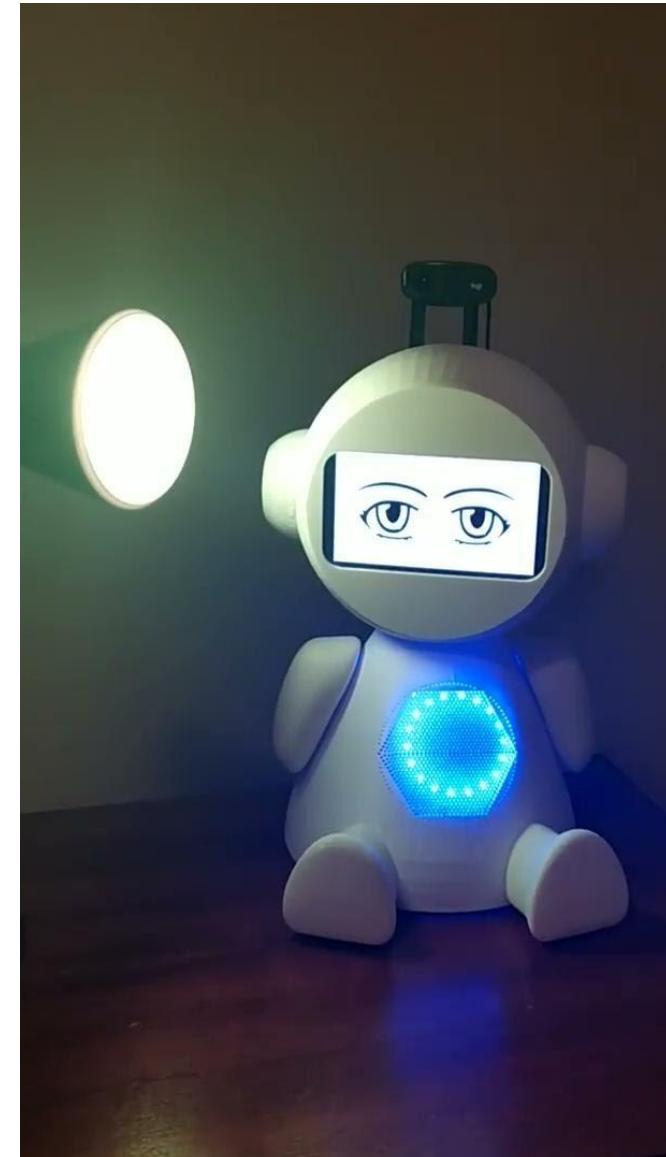
Robô EVA (Montagem física)



EVA

Um Robô Socialmente Assistivo

<https://drive.google.com/file/d/1kk31YRFc9WiPKY9fcnnZ8pyHqiKdWW4E/view?usp=sharing>



Perguntas?

Robô EVA

(Funcionalidades Principais)

Robô EVA (Funcionalidades principais)

- Instalação a partir do terminal do *Raspbian* no Raspberry
- Baixar o software do repositório
- Instalar as dependências
- Conta no IBM **Watson** e no **Google Cloud Speech API**

Robô EVA (Funcionalidades principais)

CAPACIDADES

COMUNICAÇÃO VERBAL

- Reconhecimento de VOZ
- Fala

COMUNICAÇÃO NÃO VERBAL

- Expressão através do olhar
- Animação com as luzes dos LEDs
- Movimentação da cabeça

Robô EVA (Comunicação verbal)

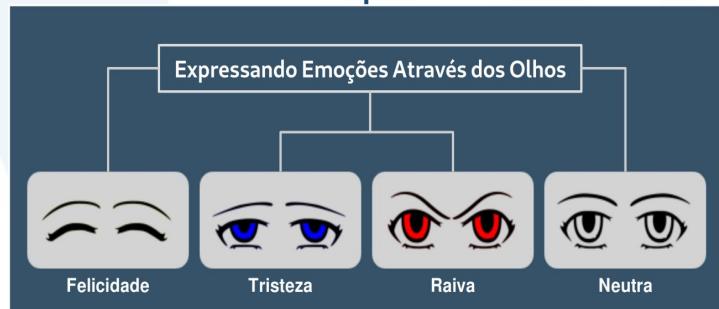


Robô EVA (Comunicação não verbal)



Robô EVA (Comunicação não verbal)

COMUNICAÇÃO NÃO VERBAL



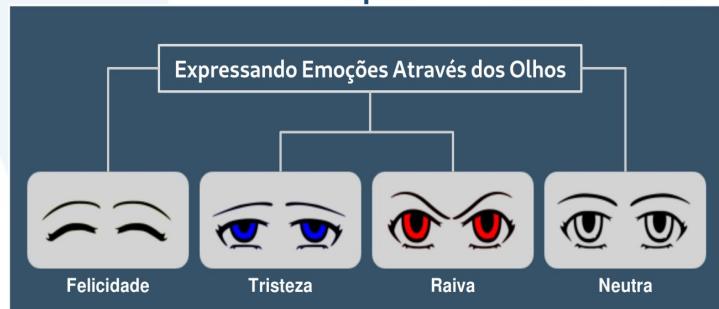
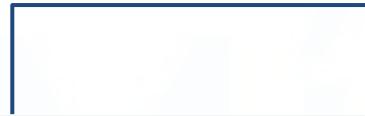
Expressões sintetizadas pelo robô **EVA**



Animação com os **LEDs** RGB

Robô EVA (Comunicação não verbal)

COMUNICAÇÃO NÃO VERBAL



Expressões sintetizadas pelo robô **EVA**



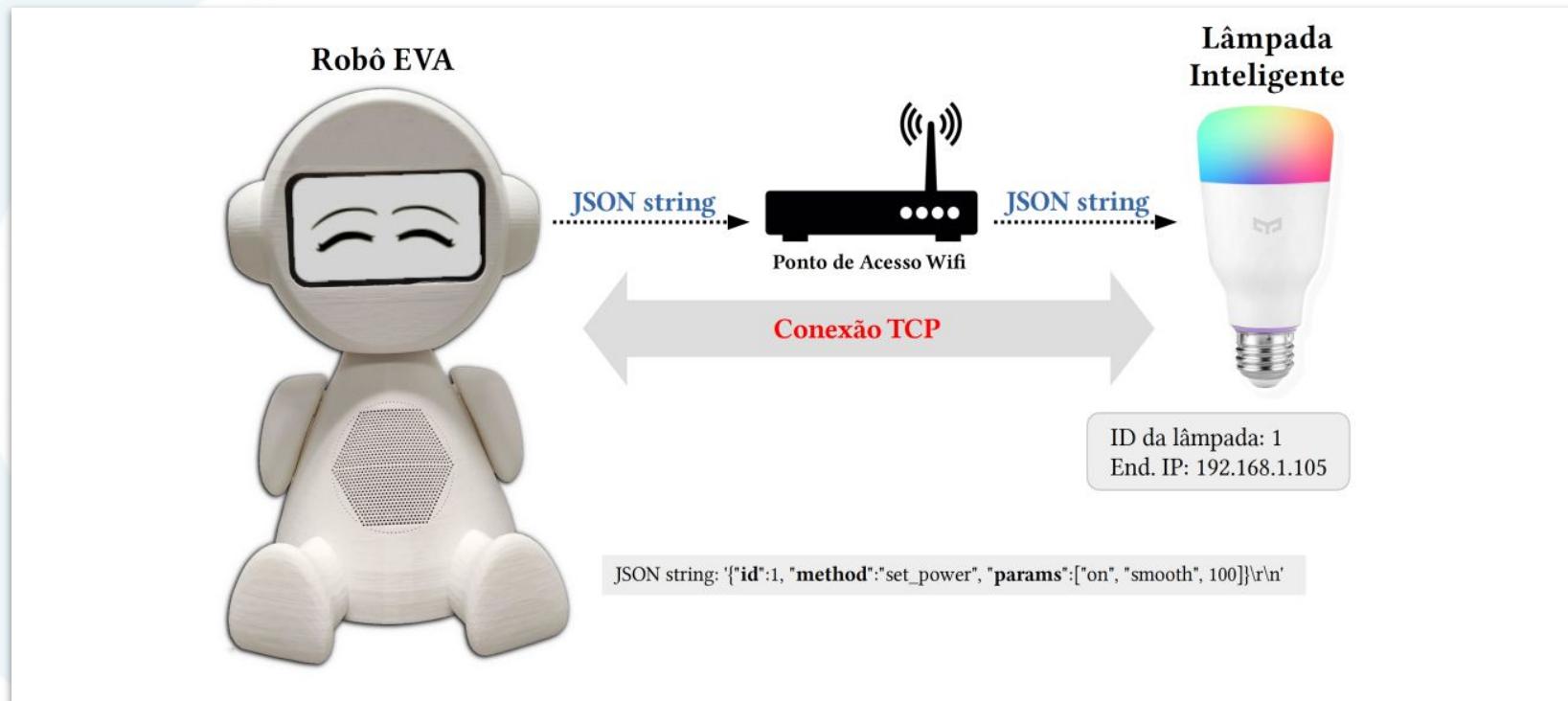
Animação com os **LEDs** RGB

Movimentação
da
Cabeça
(2 graus de liberdade)

Robô EVA (Execução de arquivos de áudio)

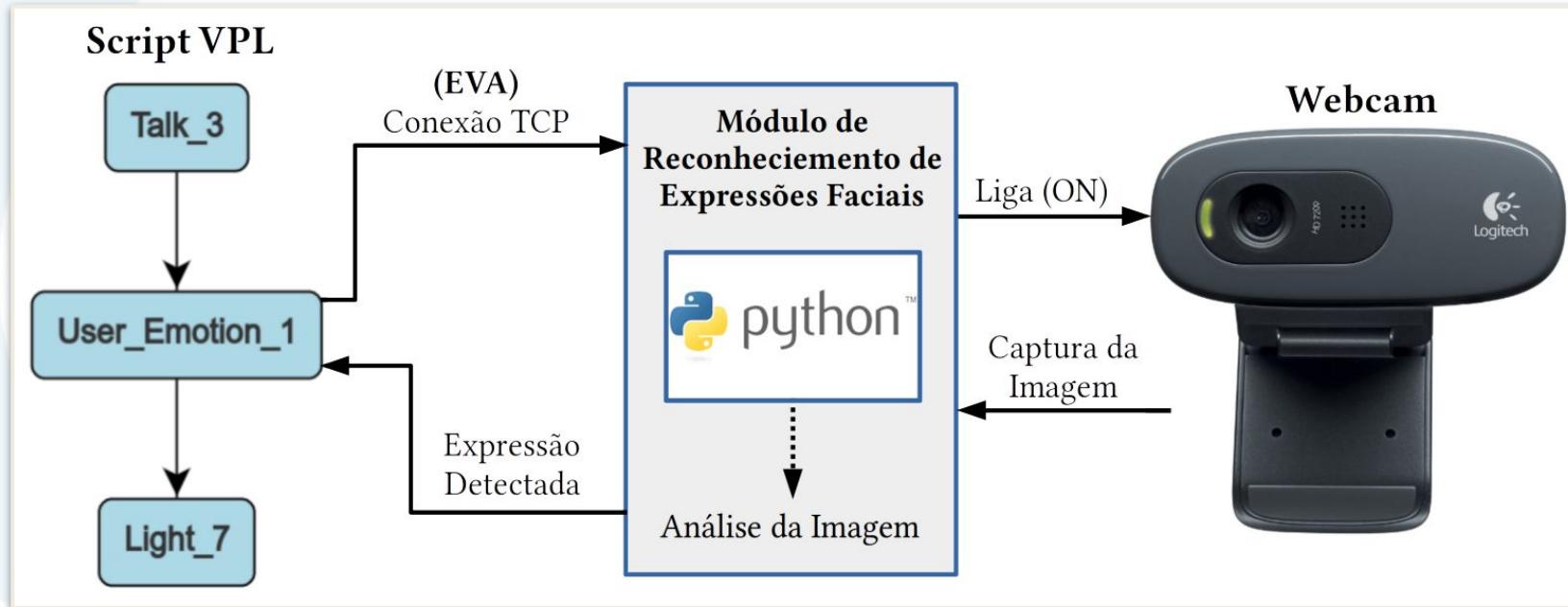


Robô EVA (Efeitos Sensoriais de Luz)



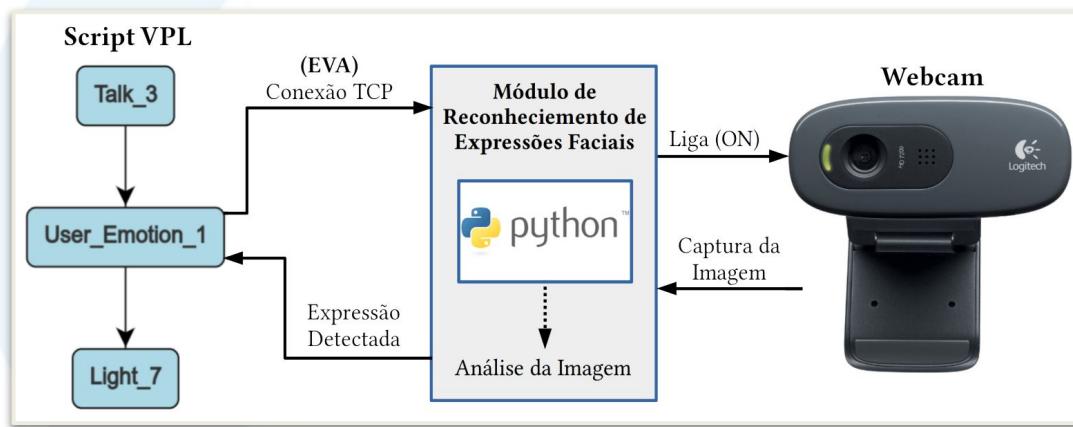
Componente **Light** - Controlando Efeitos Sensoriais de Luz [Rocha et al. 2021]

Robô EVA (Reconhecimento de Expressões Faciais)



Componente ***userEmotion*** - Reconhecimento de Expressões Faciais [Rocha et al. 2021]

Robô EVA (Reconhecimento de Expressões Faciais)

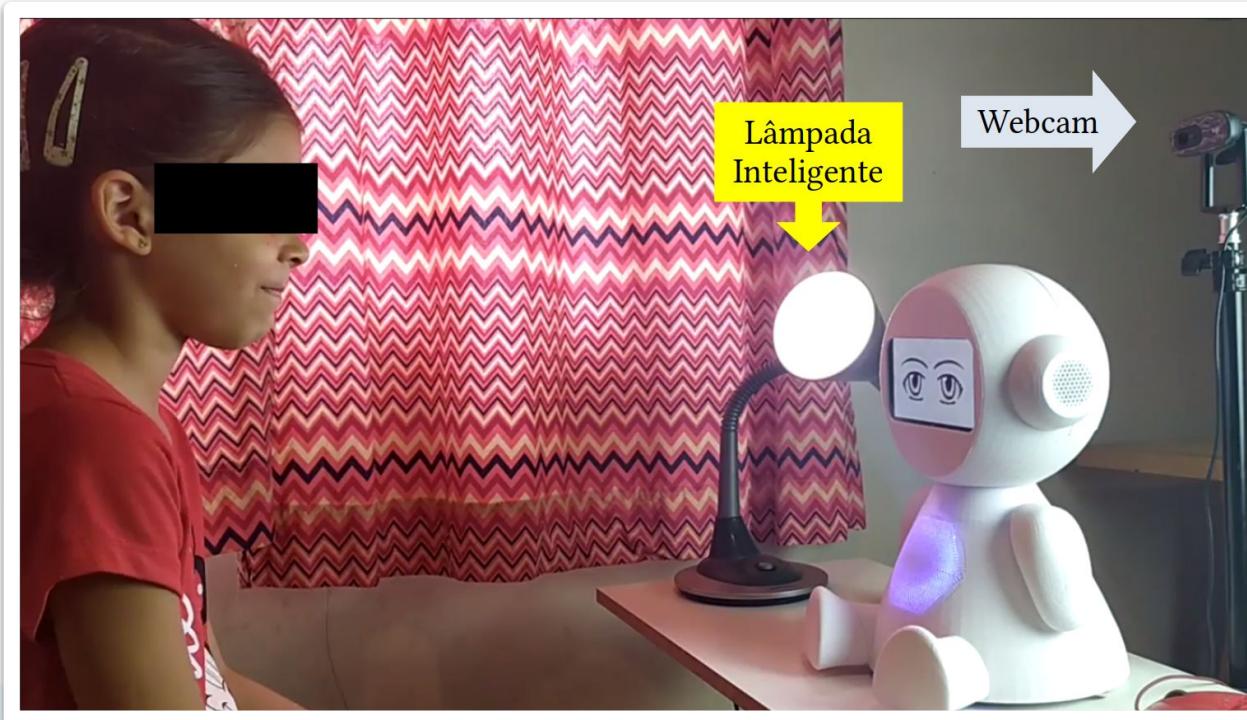


Componente ***userEmotion*** - Reconhecimento de Expressões Faciais [Rocha et al. 2021]

Expressões Reconhecidas:

- Neutra
- Raiva
- Desgosto (nojo)
- Medo
- Surpresa
- Alegria (Felicidade)
- Tristeza

Robô EVA (Interação com o robô EVA)



Criança jogando um Jogo Sério com o robô EVA

EVA

Um Robô Socialmente Assistivo (Movimento da Cabeça)

<https://drive.google.com/file/d/1LDf2zlcuPkHOhn9kIZ8Q3qxgSuPa0al4/view?usp=sharing>

Perguntas?

Roteiro do Minicurso

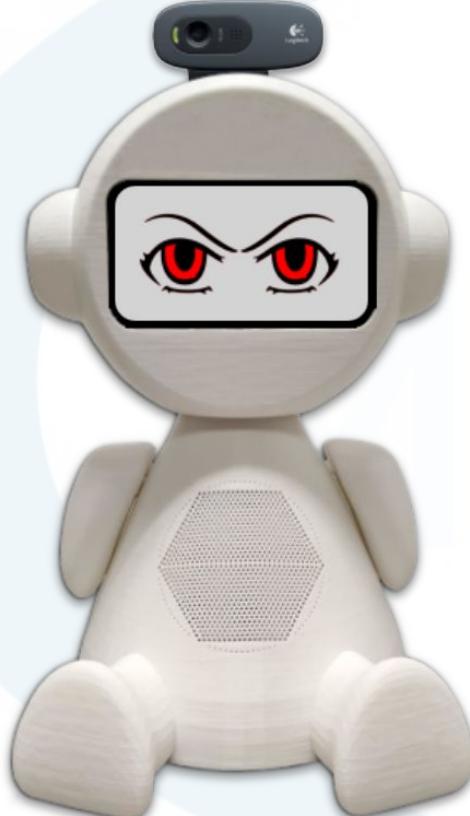
- **Introdução**

- Robôs Socialmente Assistivos (SARs)
- Aplicações em Saúde

- **Propostas de SARs**

- Comerciais
- Acadêmicas

Roteiro do Minicurso



Robô **EVA**

- **Um breve Histórico (suas versões)**
- **Arquitetura de *hardware* e *software* do Robô**
- **Montagem física do EVA**
 - Modelos 3d, placas, conectores e firmware
- **Principais Funcionalidades**
 - Comunicação *verbal* e *não verbal*
 - Interação multimodal (*efeitos sensoriais de luz, reconhecimento de expressões faciais*)

Roteiro do Minicurso

- **Desenvolvimento de Sessões de Terapia para o Robô EVA**
 - **EvaML** - Uma linguagem baseada em *XML*
 - Simulador **EvaSIM**
- **Atividade Prática**

Robôs Socialmente Assistivos

Desenvolvendo Sessões de Terapia Multissensorial com o Robô **EVA**

Marcelo Marques da Rocha (*marcelo_rocha@midiacom.uff.br*)

Sara Luzia de Melo (*sara.melo@midiacom.uff.br*)

Jesús Favela

Débora Christina Muchaluat Saade



Robôs Socialmente Assistivos

Desenvolvendo Sessões de Terapia Multissensorial com o Robô **EVA**

Marcelo Marques da Rocha

Sara Luzia de Melo

Jesús Favela

Débora Christina Muchaluat Saade



Roteiro do Minicurso

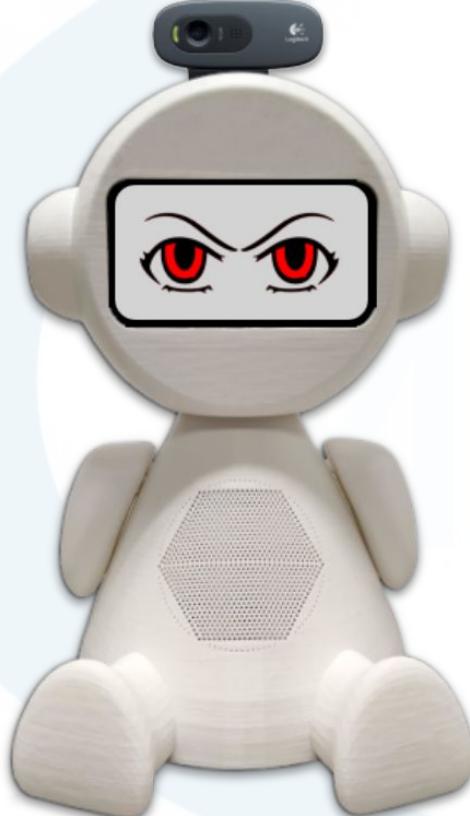
- **Introdução**

- Robôs Socialmente Assistivos (SARs)
- Aplicações em Saúde

- **Propostas de SARs**

- Comerciais
- Acadêmicas

Roteiro do Minicurso



Robô **EVA**

- **Um breve Histórico (suas versões)**
- **Arquitetura de *hardware* e *software* do Robô**
- **Montagem física do EVA**
 - Modelos 3d, placas, conectores e firmware
- **Principais Funcionalidades**
 - Comunicação *verbal* e *não verbal*
 - Interação multimodal (*efeitos sensoriais de luz, reconhecimento de expressões faciais*)

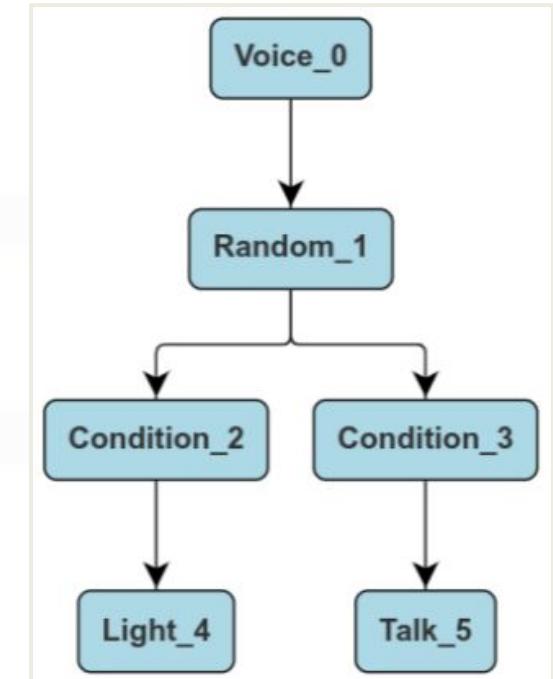
Roteiro do Minicurso

- **Desenvolvimento de Sessões de Terapia para o Robô EVA**
 - **EvaML** - Uma linguagem baseada em *XML*
 - Simulador **EvaSIM**
- **Atividade Prática**

Desenvolvimento de Sessões de Terapia para o Robô EVA

Desenvolvimento de Sessões de Terapia para o Robô EVA

- Linguagem de Programação Visual (**VPL**)
- Facilitando o desenvolvimento para usuários não técnicos
- Usa o framework web **GoJS**



Um Script na **VPL**

Desenvolvimento de Sessões de Terapia para o Robô EVA

- Proposta de um **Jogo Sério** em 3 estágios [Rocha et al. 2021]
- Dificuldades encontradas na VPL:
 - Difícil visualização dos elementos
 - Elementos distintos com o mesmo identificador
 - Dificuldade na alteração dos parâmetros dos elementos



Jogo Sério em três Estágios



(O Jogo da Imitação, com o robô EVA)

<https://drive.google.com/file/d/1gmNITSYX4Liy-OJa7vwLg7MOe2rhTtlU/view?usp=sharing>

EvaML

(Uma Linguagem Baseada em XML para a Especificação de Sessões Interativas com o Robô EVA)

EvaML (Objetivos)

“Embora o uso de ferramentas gráficas torne um usuário com alguma experiência produtivo, um usuário com habilidades avançadas no domínio da aplicação pode ter sua eficiência reduzida” [Novák 2010]

EvaML (Objetivos)

- Maior controle na entrada e edição dos comandos e seus parâmetros
- Abstração de elementos de linguagens de programação (Macros)
- Possibilitar a criação de scripts independente da interface de controle do robô

- Comandos que controlam os elementos de interação multimodal do robô estão presentes: componentes **Light**, e **userEmotion**
- Elementos para *criação e manipulação* de **variáveis**
- Geração de números aleatórios
- Controles condicionais usando elementos **switch** e **case**
- Outros...

EvaML (Por que XML?)

- O XML foi projetado para ser legível por humanos e por máquinas
- Mais **legível** para não-programadores do que as linguagens de propósito geral (C, C++, Python, Java)
- Sua gramática pode ser descrita usando uma DTD (*Document Type Definition*) ou um documento do tipo **XML Schema**
- É simples analisar a estrutura XML usando o DOM (*Document Object Model*)

EvaML (Noções básicas sobre XML)

- *XML* - eXtensible Markup Language
- Documentos *XML* são formados como “árvore” de elementos
- Uma árvore *XML* começa em um elemento raiz e se ramifica da raiz para os elementos filhos

<https://www.w3schools.com/>

EvaML (Noções básicas sobre XML)

- Todos os elementos *XML* devem ter uma tag de fechamento
- As tags *XML* diferenciam maiúsculas de minúsculas
- A tag `<Light>` é diferente da tag `<light>`
- As tags de abertura e fechamento devem ser escritas da mesma maneira
- Elementos *XML* podem ter atributos em pares nome/valor, assim como em *HTML*.
- Em *XML*, os valores dos atributos devem estar entre aspas (simples ou duplas)

<https://www.w3schools.com/>

EvaML (Noções básicas sobre XML)

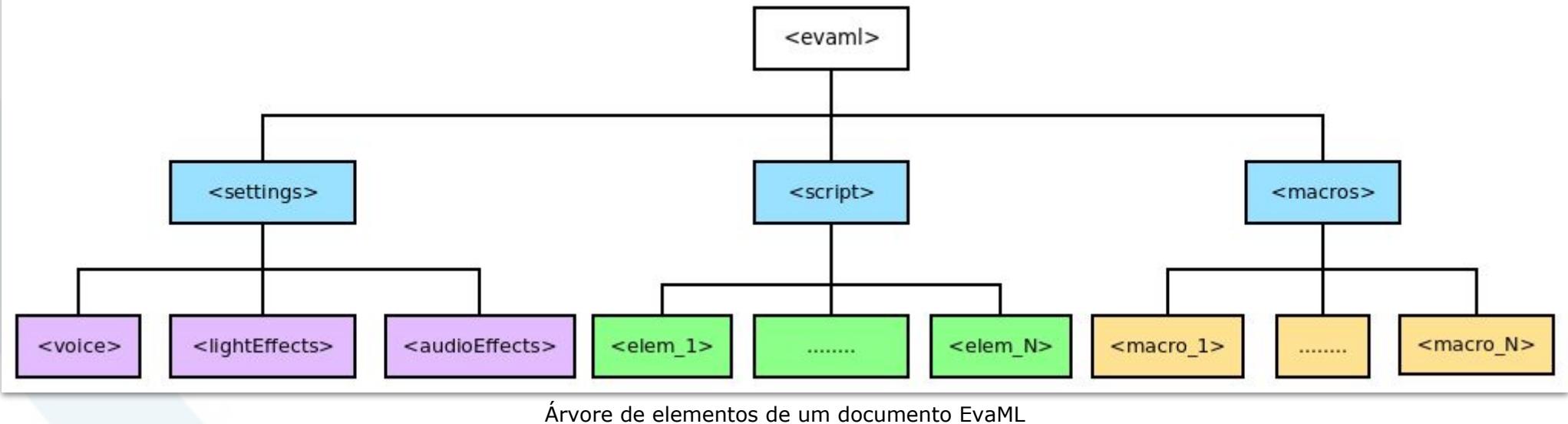
```
(1) <talk>Olá eu sou o robô EVA</talk>  
  
(2) <listen />  
  
(3) <light state="ON" color="RED" />  
  
(4) <macros>  
(5)   <macro>  
(6)     <wait duration="1000" />  
        <led animation="HAPPY" />  
(7)   </macro>  
(8) </macros>
```

Exemplo: Tags (abertura e fechamento) e Atributos em *XML*

EvaML (Elementos de um Documento EvaML)



EvaML (Elementos de um Documento EvaML)



EvaML (Elementos de um Documento EvaML)

Elemento settings - Define algumas características globais do script. É possível definir como será o timbre da voz e o idioma em que o robô irá se comunicar. É possível definir se o código gerado, ao ser executado, irá reproduzir comandos de efeitos de luz, efeitos sonoros ou até mesmo tocar música.

```
<settings>
  <voice tone="en-US_AllisonV3Voice" />
  <lightEffects mode="ON" />
  <audioEffects mode="ON"/>
</settings>
```

Exemplo: Seção **settings**

EvaML (Elementos de um Documento EvaML)

Elemento script - Contém a sequência de comandos que o robô deve executar.

```
<script>
<light state="ON" color="BLUE" />
<talk>Hi, I am robot EVA</talk>
<wait duration="2000" />
<audio source="mario-start" block="TRUE" />
<talk>Bye</talk>
<light state="OFF" />
</script>
```

Exemplo: Seção **script**

EvaML (Elementos de um Documento EvaML)

Elemento macros - É uma das abstrações criadas na linguagem EvaML. Como pode ser visto no trecho de código, é possível criar **macros** que podem ser referenciadas dentro do elemento `<script>`.

```
<script>
  <useMacro macro="START" />
</script>
<macros>
  <macro id="START">
    <talk>Hello, I'm robot Eva.</talk>
    <talk>What is your name?</talk>
    <talk>Let us play one more time?</talk>
  </macro>
</macros>
```

Exemplo: Seção **macros**

EvaML (Elementos de um doc. EvaML - Indicadores de ocorrência)

Atributo obrigatório

A ordem importa!

Ocorrências: 0 ou 1

Elemento	Atributo	Conteúdo
evaml	<u>name</u>	(settings, script, macros?)
settings		(voice lightEffects? audioEffects?)
script		(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* useMacro* listen* audio* led* counter* switch*)
macros		(macro+)

Elementos de um documento **EvaML** (Modelo de conteúdo dos Elementos)

Ocorrências: 1 ou mais

A ordem NÃO importa!

Ocorrências: 0 ou mais

EvaML (Elementos de um Documento EvaML - Comandos)

Elemento	Atributo	Conteúdo
voice	<u>tone</u>	empty
lightEffects	<u>mode</u>	empty
audioEffects	<u>mode</u>	empty
random	<u>id</u> , <u>min</u> , <u>max</u> ,	empty
wait	<u>id</u> , <u>duration</u>	empty
talk	<u>id</u>	text
stop		empty
light	<u>id</u> , <u>state</u> , <u>color</u>	empty
goto	<u>target</u>	empty
motion	<u>id</u> , <u>type</u>	empty
userEmotion	<u>id</u>	empty
evaEmotion	<u>id</u> , <u>emotion</u>	empty
useMacro	<u>macro</u>	empty
listen	<u>id</u>	empty
audio	<u>id</u> , <u>source</u> , <u>block</u>	empty
led	<u>id</u> , <u>animation</u>	empty
counter	<u>id</u> , <u>var</u> , <u>op</u> , <u>value</u>	empty

Modelo de conteúdo dos Elementos

EvaML (Elementos de um Documento EvaML - Comandos)

switch	<u>id</u> , <u>var</u>	(case+, default?)
macro	<u>id</u>	(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* listen* audio* led* counter* switch*)
case	<u>op</u> , <u>value</u>	(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* useMacro* listen* audio* led* counter* switch*)
default		(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* useMacro* listen* audio* led* counter* switch*)

Modelo de conteúdo dos Elementos

Perguntas?

EvaML (<voice>)

Elemento de configuração da linguagem que define, ao mesmo tempo, o timbre de voz (seu gênero) a ser usado pelo robô, e o idioma que será utilizado durante o processo de conversão de texto para fala do serviço IBM Watson.

Elemento	Atributo	Conteúdo
voice	<u>tone</u>	empty

```
<settings>
  <voice tone="en-US_AllisonV3Voice" />
  <lightEffects mode="ON" />
  <audioEffects mode="ON"/>
</settings>
```

Exemplo: Comando **voice**

Código	Gênero	Idioma
pt-BR_IsabelaV3Voice	feminino	português do Brasil
en-US_AllisonV3Voice	feminino	inglês dos EUA
en-US_EmilyV3Voice	feminino	inglês dos EUA
en-US_HenryV3Voice	masculino	inglês dos EUA
es-LA_SofiaV3Voice	feminino	espanhol latinoamericano
es-ES_EnriqueV3Voice	masculino	espanhol

Vozes para o serviço de texto para fala do **IBM Watson**

EvaML (<random>)

Gera um número inteiro aleatório no intervalo fechado **[min, max]**. O seu atributo **min**, define o limite inferior, e o seu atributo **max**, define o limite superior do número aleatório a ser gerado. O valor gerado pela função aleatória é armazenado em uma região especial da memória do robô, que funciona como um vetor. O caractere **\$** acessa o elemento no final do vetor.

Elemento	Atributo	Conteúdo
random	id, <u>min</u> , <u>max</u>	empty

```
<script>
  <talk>I will randomly choose a number from 1 to 3</talk>
  <random min="1" max="3" />
  <talk>The number drawn was $</talk>
</script>
```

Exemplo: Comando **random**

EvaML (<wait>)

Este comando pausa a execução do script pelo intervalo de tempo definido no seu atributo **duration**. A unidade de tempo usada é o milissegundo.

Elemento	Atributo	Conteúdo
wait	id, <u>duration</u>	empty

```
<script>
  <talk>I will wait for one second</talk>
  <wait duration="1000" />
  <talk>Ready!</talk>
</script>
```

Exemplo: Comando **wait**

EvaML (<talk>)

Através do uso do comando **<talk>** o robô pode falar um texto especificado. Ao se definir o texto a ser falado é possível utilizar o conteúdo da memória do robô como parte do texto, utilizando-se, no corpo do texto, o caractere **\$**, que referencia uma área especial da memória do robô ou a notação **#var** que referencia o conteúdo de uma variável definida pelo usuário.

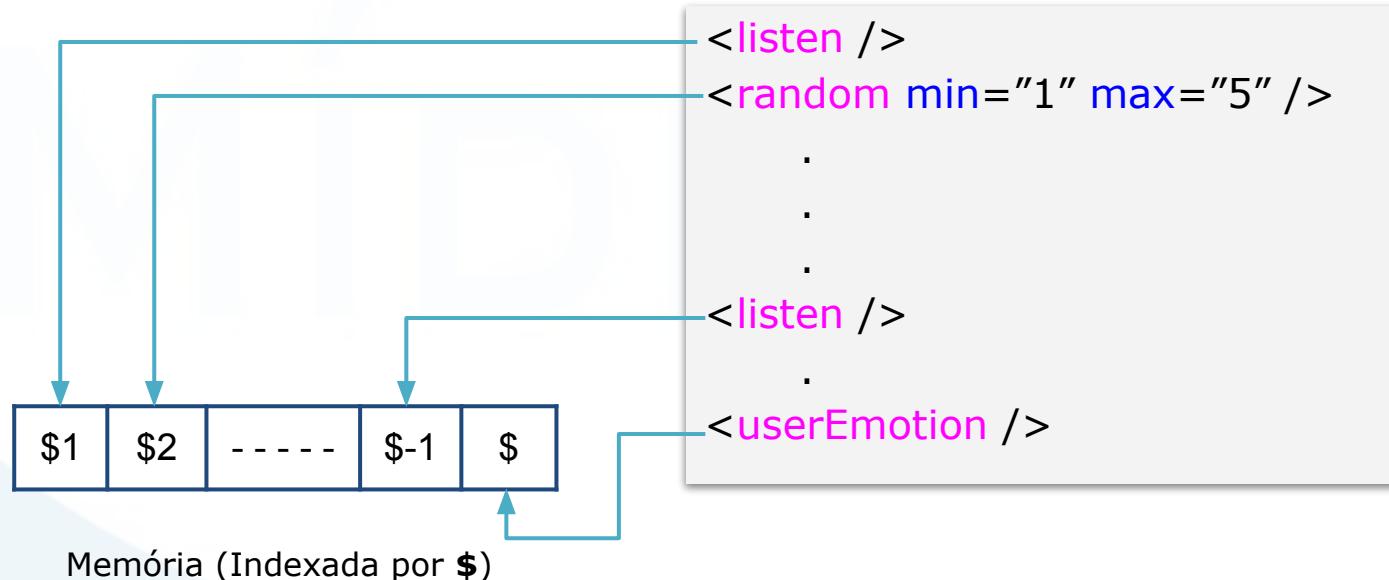
Elemento	Atributo	Conteúdo
talk	id	text

```
1 <script>
2   <talk>What is your name?</talk>
3   <listen />
4   <talk>Hello $ / Hi $ / Hi $, how is it going</talk>
5   <random min="1" max="3" />
6   <talk>Hello $1, I sorted number $</talk>
7 </script>
```

Exemplo: Comando **talk**

EvaML (Acessando a memória do robô usando o caractere \$)

O caractere \$, por convenção, faz referência ao valor no final do array na memória que armazena as respostas retornadas pelos comandos de interação com o usuário, são eles: <listen>, <userEmotion> e, os valores gerados pelo comando <random>.



EvaML (<stop>)

Este comando é muito simples e como o nome sugere, ele interrompe a execução do script.

Elemento	Atributo	Conteúdo
stop		empty

EvaML (<light>)

Controla a lâmpada inteligente. O seu atributo **state** pode assumir os valores "**ON**" e "**OFF**" e o atributo **color** define a cor da lâmpada. Essa cor pode ser indicada usando-se a representação hexadecimal RGB "#00ff00" ou alguma das cores da lista predefinida: "WHITE", "BLACK", "RED", "PINK", "GREEN", "YELLOW", "BLUE".

Elemento	Atributo	Conteúdo
light	id, <u>state</u> , color	empty

```
1 <script>
2   <light state="ON" color="RED" />
3   <wait duration="1000" />
4   <light state="ON" color="#00ff00" />
5   <wait duration="1000" />
6   <light state="ON" color="BLUE" />
7   <wait duration="1000" />
8   <light state="OFF" />
9 </script>
```

Exemplo: Comando **light**

EvaML (<goto>)

Altera o fluxo de execução do script para o comando com **id** referenciado em seu atributo **target**. O atributo **id** define o rótulo que será usado como valor no atributo **target** do comando **<goto>**.

Elemento	Atributo	Conteúdo
goto	<u>target</u>	empty

```
1 <script>
2   <light id="BEGIN" state="ON" color="RED" />
3   <wait duration="1000" />
4   <light state="OFF" />
5   <wait duration="1000" />
6   <goto target="BEGIN" />
7 </script>
```

Exemplo: Comando **goto**

EvaML (<motion>)

O robô pode mover sua cabeça e o comando `<motion>` é responsável por controlar este movimento. Ele possui o atributo `type` que pode assumir os seguintes valores: "YES", "NO", "CENTER", "LEFT", "RIGHT", "UP", "DOWN", "ANGRY", "2UP", "2DOWN", "2LEFT" e "2RIGHT".

Elemento	Atributo	Conteúdo
motion	id, <u>type</u>	empty

```
1 <script>
2   <talk>I will move my head to the right</talk>
3   <motion type="RIGHT" />
4   <talk>I am very angry</talk>
5   <light state="ON" color="RED" />
6   <evaEmotion emotion="ANGRY" />
7   <motion type="ANGRY" />
8 </script>
```

Exemplo: Comando `motion`

EvaML (<userEmotion>)

O robô é capaz de reconhecer expressões faciais através de uma webcam. Ele faz isso usando um módulo externo escrito em Python. O módulo recebe a solicitação, ativa a webcam e retorna as seguintes expressões como uma string: "**NEUTRAL**", "**ANGRY**", "**DISGUST**", "**FEAR**", "**SURPRISE**", "**HAPPY**" e "**SAD**". Essa resposta pode ser usada no restante do script sendo acessada através do caractere \$.

Elemento	Atributo	Conteúdo
userEmotion	id	empty

```
1 <script>
2   <light state="ON" color="GREEN" />
3   <talk>Hi! How are you feeling?</talk>
4   <userEmotion />
5   <audio source="mario-fundo" block="TRUE" />
6   <talk>I see that you feel $</talk>
7 </script>
```

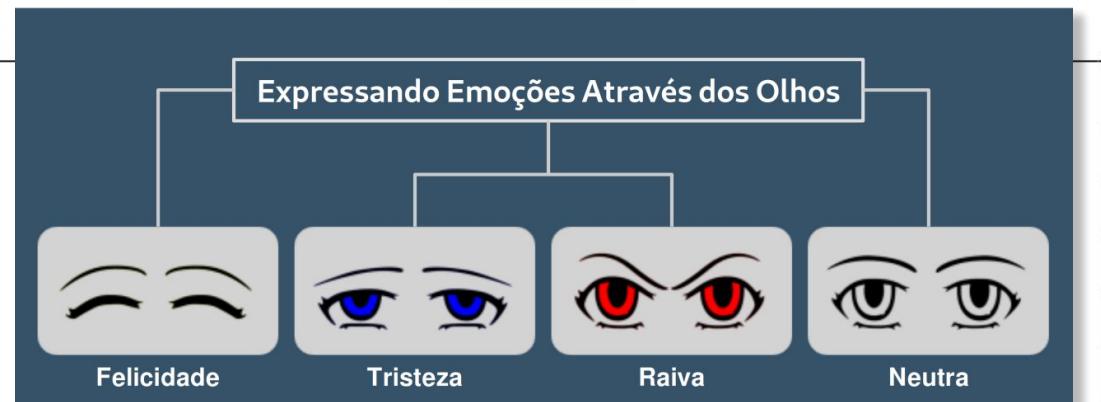
Exemplo: Comando userEmotion

EvaML (<evaEmotion>)

Controla a apresentação das expressões do olhar no display de 5.5". O seu atributo **emotion** pode ter os seguintes valores: "**HAPPY**", "**SAD**", "**ANGRY**" e "**NEUTRAL**".

Elemento	Atributo	Conteúdo
evaEmotion	id, <u>emotion</u>	empty

```
1 <script>
2   <evaEmotion emotion="HAPPY" />
3   <talk>I am feeling happy</talk>
4   <wait duration="1000" />
5   <evaEmotion emotion="SAD" />
6   <talk>I am feeling sad</talk>
7   <wait duration="1000" />
8 </script>
```



Expressões sintetizadas pelo robô **EVA**

Exemplo: Comando **evaEmotion**

EvaML (<macro>)

Uma macro define uma sequência de comandos que podem ser referenciados dentro do elemento <script> usando o comando <useMacro>. O elemento <macro> tem apenas o atributo **id** que é usado para identificá-lo. É importante saber que um elemento <macro> não pode conter a definição de outra macro e não pode, por meio de um comando <useMacro>, referenciar outra macro.

macro	<u>id</u>	(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* listen* audio* led* counter* switch*)
-------	-----------	--

```
1 <macros>
2   <macro id="START">
3     <talk>Hello, I'm robot Eva. I'll pick a number from one to four</talk>
4     <talk>For each one, I'll turn the bulb on in a different color</talk>
5     <talk>I will only terminate when the number drawn is three</talk>
6   </macro>
7   <macro id="END">
8     <talk>I'm leaving now / It's getting late</talk>
9   </macro>
10 </macros>
```

Exemplo: Comando **macro**

EvaML (<useMacro>)

O comando <useMacro> faz com que o código da macro, referenciado em seu atributo **macro**, seja expandido pelo código onde o comando <useMacro> está declarado.

Elemento	Atributo	Conteúdo
useMacro	<u>macro</u>	empty

```
1 <script>
2   <useMacro macro="START" />
3   <useMacro macro="END" />
4 </script>
5 <macros>
6   <macro id="START">
7     <talk>Hello, I'm robot Eva</talk>
8     <talk>I'm a socially assistive robot</talk>
9     <talk>I can speak</talk>
10    <talk>I can recognize facial expressions</talk>
11  </macro>
12  <macro id="END">
13    <talk>I'm leaving now / It's getting late</talk>
14  </macro>
15 </macros>
```

Exemplo: Comando **useMacro**

EvaML (<listen>)

O robô pode reconhecer a voz humana e para isso utiliza o serviço de conversão de fala para texto (STT) da API na nuvem do *Google*. O áudio capturado é enviado para a nuvem, processado, e então, o texto resultante do processo de STT é retornado para ser utilizado pelo software do robô.

Elemento	Atributo	Conteúdo
listen	id	empty

```
<script>
  <talk>Say what you are thinking</talk>
  <listen />
  <talk>Wow! You are thinking of $</talk>
</script>
```

Exemplo: Comando **listen**

EvaML (<audio>)

Reproduz um arquivo de áudio contido na pasta "sonidos". O comando possui os atributos **source** e **block**. O atributo **source** indica o nome do arquivo a ser tocado. O EVA só reproduz arquivos no formato .wav. O atributo **block** pode ter os seguintes valores, "**TRUE**" ou "**FALSE**" e define se a execução do arquivo deve bloquear a execução do script, ou seja, o comando que vem após o comando **<audio>** só será executado após o término da reprodução do áudio.

Elemento	Atributo	Conteúdo
audio	id, <u>source</u> , <u>block</u>	empty

```
1 <script>
2   <light state="ON" color="RED" />
3   <audio source="mario-start" block="TRUE" />
4   <light state="OFF" />
5 </script>
```

Exemplo: Comando **audio**

EvaML (<led>)

Controla a animação com os LEDs no peito do robô EVA. O seu atributo **animation** pode assumir os seguintes valores: "**HAPPY**" (verde), "**SAD**" (azul), "**ANGRY**" (vermelho), "**STOP**" (sem cor/desligado), "**SPEAK**" (azul), "**LISTEN**" (verde) e "**SURPRISE**" (amarelo).

Elemento	Atributo	Conteúdo
led	id, <u>animation</u>	empty

```
1 <script>
2   <led animation="HAPPY" />
3   <wait duration="1000" />
4   <led animation="SAD" />
5   <wait duration="1000" />
6   <led animation="ANGRY" />
7   <wait duration="1000" />
8   <led animation="SPEAK" />
9   <wait duration="1000" />
10  <led animation="LISTEN" />
11 </script>
```

Exemplo: Comando led

EvaML (<counter>)

Cria, inicializa e realiza operações matemáticas sobre variáveis. O atributo **var** define o nome variável, enquanto o atributo **op** define o tipo de operação, podendo assumir os seguintes valores: "=" (atribuição), "+" (adição), "*" (multiplicação), "/" (divisão) e "%" (módulo). Para referenciar o valor de uma variável criada pelo usuário, diferentemente do caractere "\$", é necessário utilizar o caractere "#" antes do nome da variável. O atributo **value** determina o valor usado na operação especificada.

Elemento	Atributo	Conteúdo
counter	id, <u>var</u> , <u>op</u> , <u>value</u>	empty

```
1 <script>
2   <counter var="x" op="=" value="10" />
3   <counter var="x" op)+" value="10" />
4   <counter var="x" op="*" value="10" />
5   <counter var="x" op="/" value="10" />
6   <counter var="x" op="%" value="3" />
7   <talk>The value of x is #x</talk>
8 </script>
```

Exemplo: Comando **counter**

EvaML (<switch>)

Define a variável que será comparada com os valores definidos nos comandos <case>. Para isso, seu atributo **var** deve conter o nome da variável a ser comparada, por exemplo: <switch var="\$"> ou <switch var="x">. O atributo **var** do comando <switch> determina com qual variável as comparações serão feitas nos comandos <case> e pode assumir os valores "\$" ou qualquer outro nome de variável que tenha sido declarada anteriormente.

Elemento	Atributo	Conteúdo
switch	id, <u>var</u>	(case+, default?)

EvaML (<case>)

O comando <case> especifica uma sequência de comandos que serão executados se a condição definida em seus atributos for verdadeira. O comando <case> possui o atributo **op**, que define o tipo de comparação ou operador lógico que será processado, e o atributo **value**, que contém o valor a ser comparado com a variável definida no atributo **var** do comando <switch>.

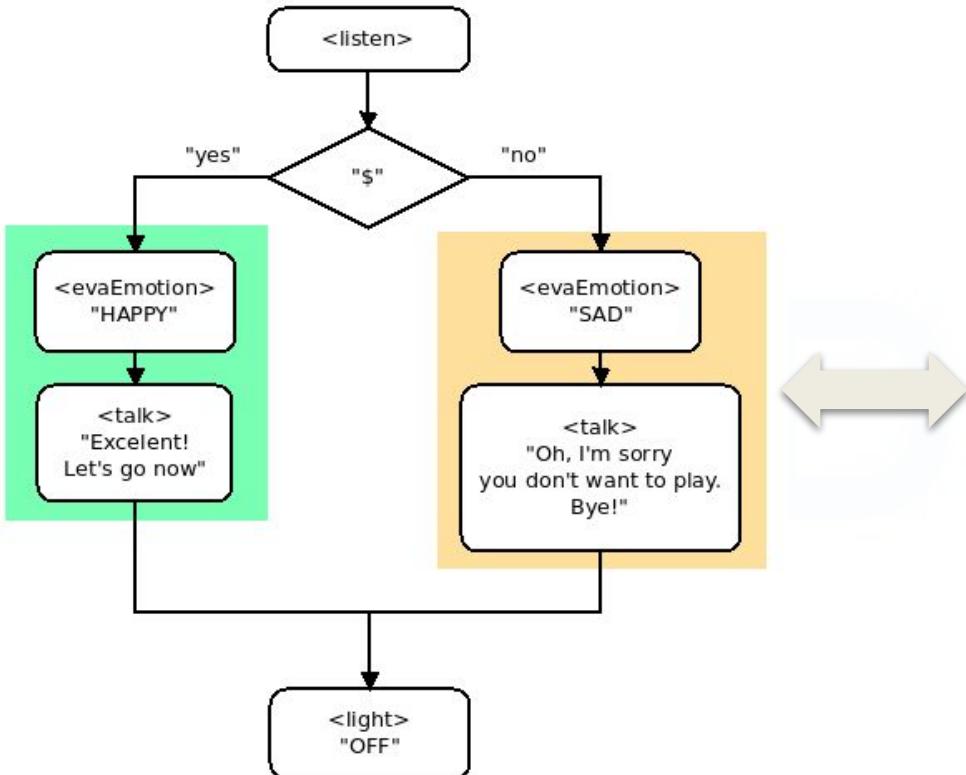
case	<u>op, value</u>	(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* useMacro* listen* audio* led* counter* switch*)
------	------------------	--

EvaML (<case>)

Atributo var do <switch>	Tipo de comparação (Atributo op do <case>)	Tipo do valor no atributo value do <case>	Descrição
\$	exact	string	Comparação exata entre duas strings. <u>Não é case sensitive.</u>
	contain	string	Verifica se “\$” contém a string em “value”. <u>Não é case sensitive.</u>
\$ ou “variável”	math (“eq”, “lt”, “gt”, “lte”, “gte”, “ne”)	Uma constante numérica, \$ ou #var .	Comparação numérica entre “\$” ou outra variável, com um valor “constante”, o \$, ou outra variável contida no atributo value de <case>. <u>Somente trabalha com números inteiros.</u>

Tipos de comparação e operações lógicas do comando <case>

EvaML (<case>)

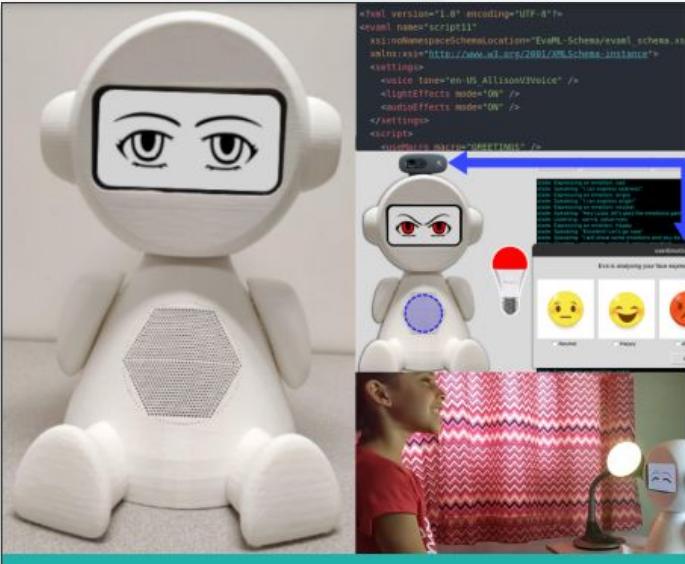


Fluxograma de exemplo

```
<script>
<listen />
<switch var="$">
  <case op="exact" value="yes">
    <evaEmotion emotion="HAPPY" />
    <talk>"Excellent! Let's go now"</talk>
  </case>
  <case op="exact" value="no">
    <evaEmotion emotion="SAD" />
    <talk>"Oh, I'm sorry you want to play. Bye!"</talk>
  </case>
</switch>
<light state="OFF" />
</script>
```

Exemplo: Comandos **switch** e **case**

EvaML (Manual de Referência)



EvaML and EvaSIM

An XML-Based Language for Specification of Interactive Sessions

and a Simulator for Robot EVA

Author: Marcelo Manques da Rocha

Email: marcelo_rocha@midiacom.uff.br

Institute: Fluminense Federal University - MidiaCom Lab - UFF

Version: 1.0



EvaML e EvaSIM - Manual de Referência

<https://github.com/midiacom/eva-robot/blob/master/EvaML-Reference-Manual/EvaML-Reference-Manual.pdf>

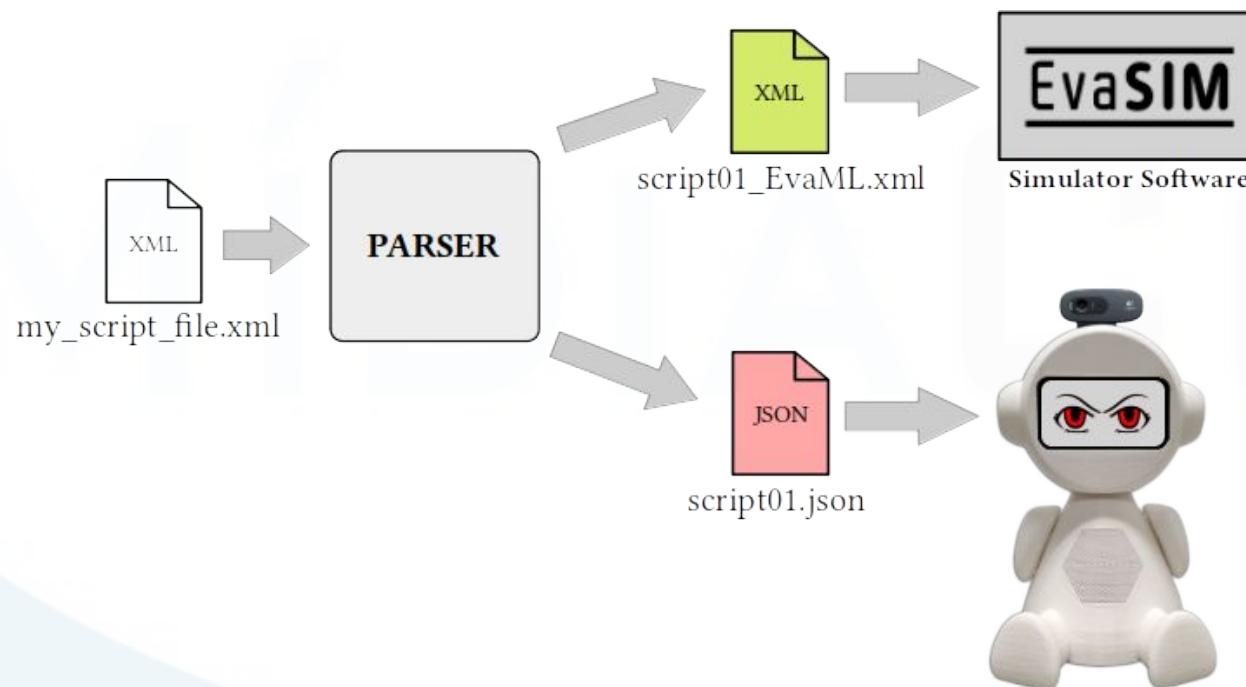
Perguntas?

EvaML

(Processo de Parsing de um Script EvaML)

EvaML (Processo de Parsing)

O processo de parsing de um documento EvaML ocorre em quatro etapas:

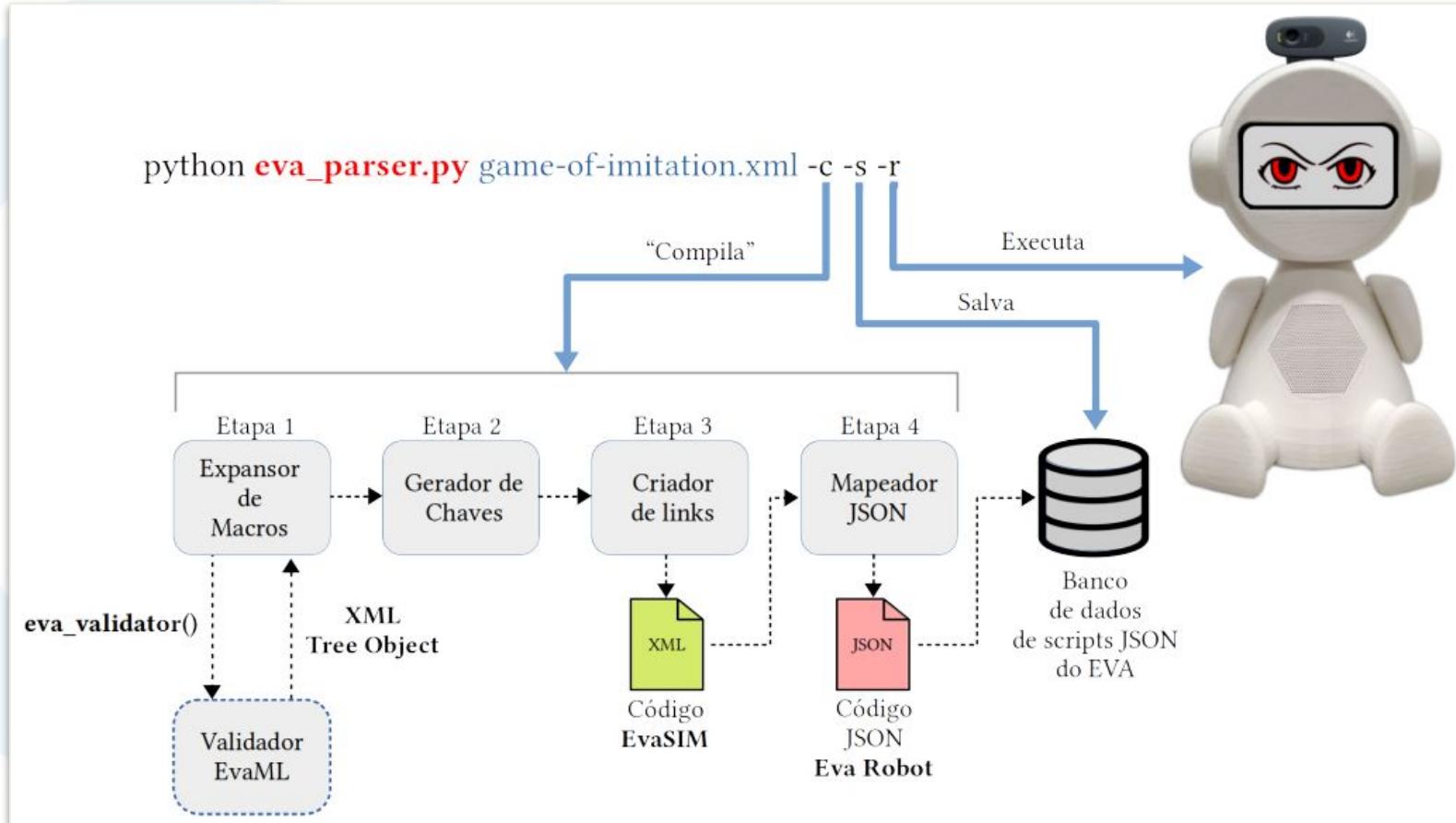


EvaML (Processo de Parsing)

O processo de parsing de um documento EvaML ocorre em quatro etapas:

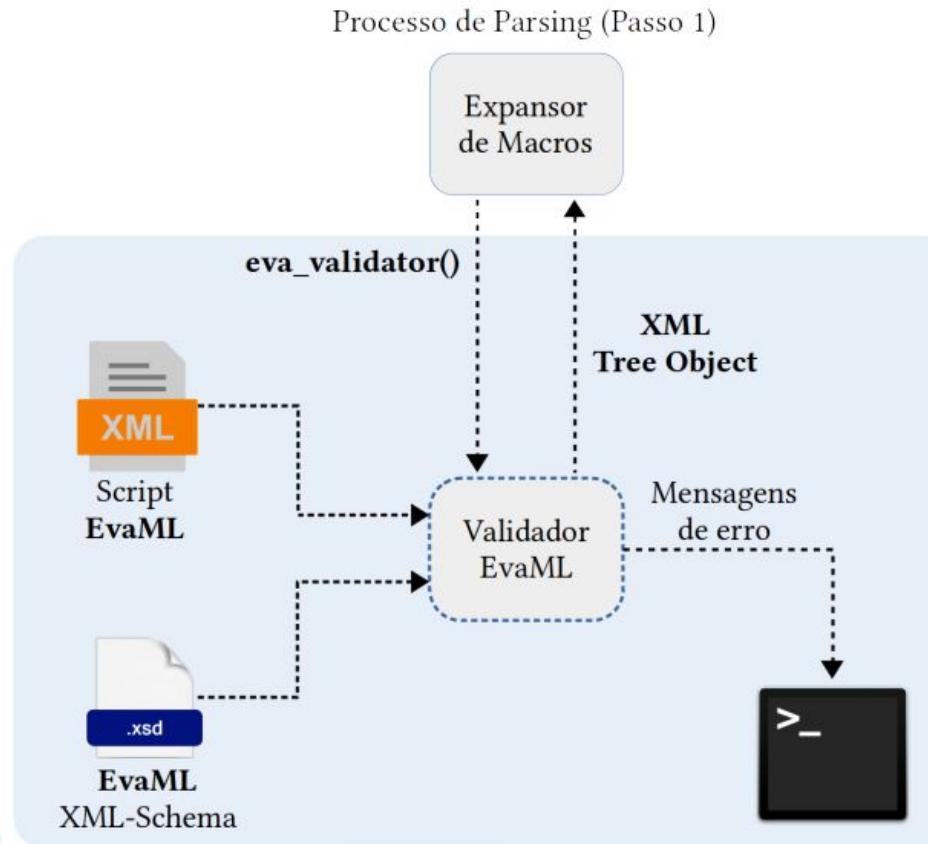
- **Etapa 1:** Expansor de Macros
 - **Validador XML (XML Schema)**
- **Etapa 2:** Gerador de Chaves
- **Etapa 3:** Criador de Links
- **Etapa 4:** Mapeador JSON

EvaML (Processo de Parsing - As 4 etapas)



Processo de *Parsing* e exemplo de linha de comando

EvaML (Processo de Validação de um Script EvaML)



Processo de validação de um documento EvaML usando *XML Schema*

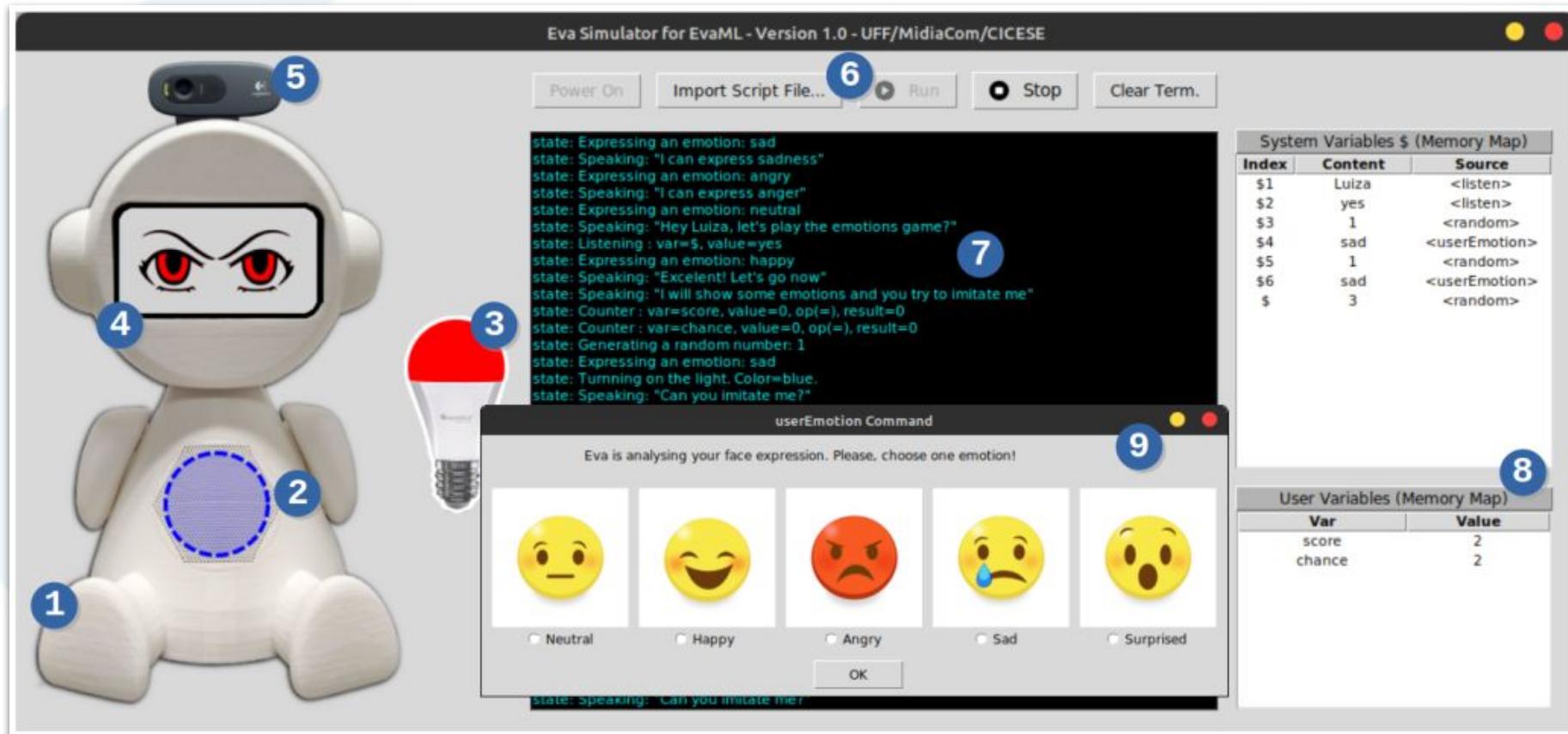
EvaSIM

(O Software Simulador do Robô EVA)

EvaSIM (O Software Simulador do Robô EVA)

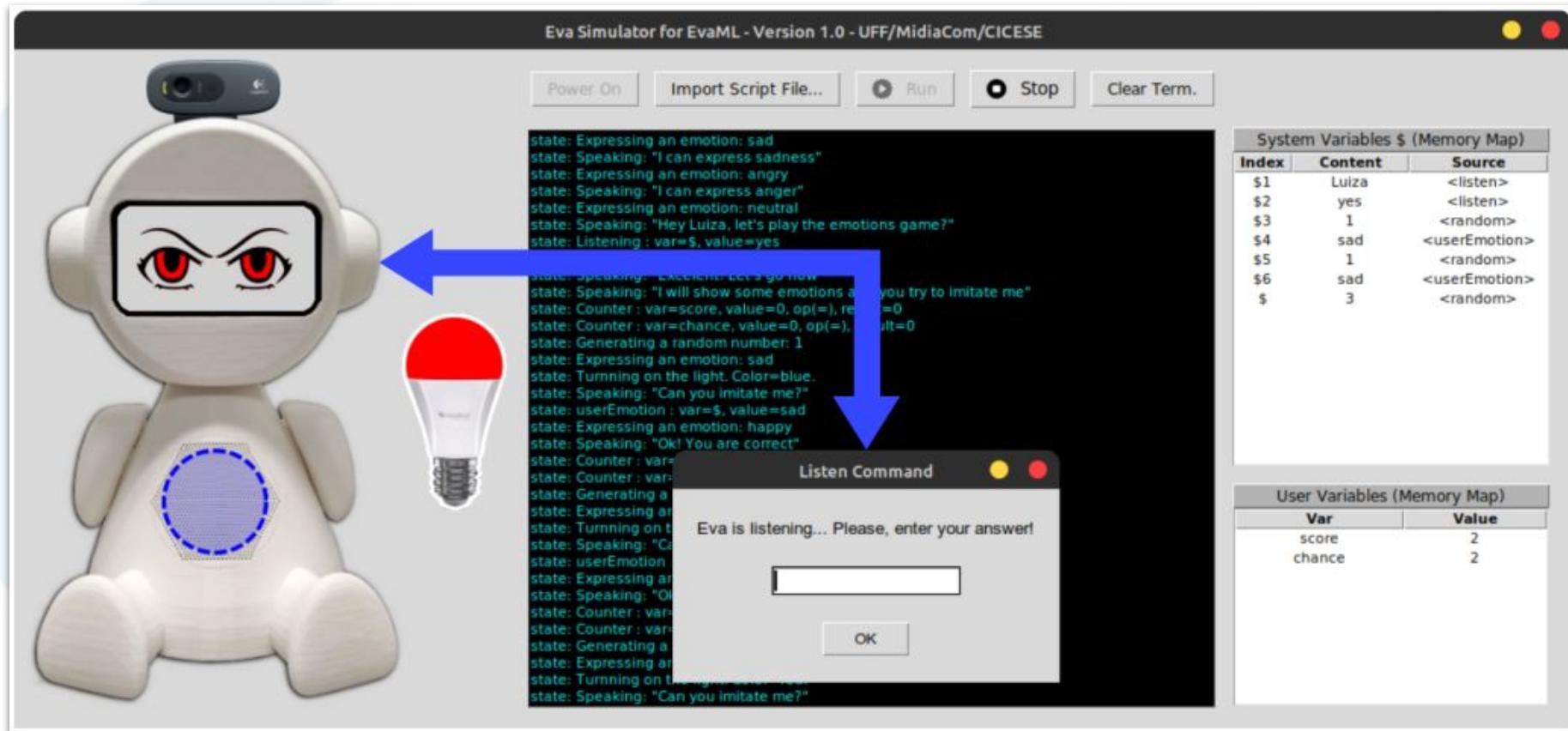
- Ambiente de teste para os scripts **EvaML**
- É portável (*Linux Ubuntu* e *Windows 10*)
- Linguagem *Python*
- Interface Gráfica de Usuário (*Tkinter*)
- Simples e leve

EvaSIM (Interface Gráfica de Usuário)



Elementos da Interface Gráfica de Usuário do **EvaSIM**

EvaSIM (Interface Gráfica de Usuário)



Simulando o reconhecimento de voz

EvaSIM (Interface Gráfica de Usuário)

```
state: Matrix Leds. Animation=STOP  
state: Moving the head! Movement type: CENTER  
state: Pausing. Duration=1000 ms  
state: Matrix Leds. Animation=SURPRISE  
state: Moving the head! Movement type: 2RIGHT  
state: End of script.
```

Indicação no terminal, o movimento da cabeça do robô

Perguntas?

Atividade Prática (parte 1)

(Fazendo o *Parsing* de um Script)

Atividade Prática (Preparando o ambiente)

- Baixando as ferramentas auxiliares
 - IDE Visual Studio Code (<https://code.visualstudio.com/>)
 - Instalar no VScode a extensão: *XML Language Support by Red Hat*
 - Use o atalho (Ctrl+shift+x)
 - digite *XML Language Support by Red Hat*
 - Clique em instalar a extensão

Atividade Prática (Preparando o ambiente)

- Instalar o **Python** e os módulos a seguir:
 - No Windows (Instalar os módulos “requests” e “xmlschema”)
 - pip install requests
 - pip install xmlschema
 - No Linux (Instalar o módulo “xmlschema”)
 - pip install xmlschema
- Baixar o *Parser* e o **EvaSIM** ([Link](#))
 - Versão para Linux Ubuntu 20.04.4 ou Linux Mint 20.3
 - Versão para Windows 10

Atividade Prática (Fazendo o *Parsing* de um Script)

- Com o **Visual Studio Code** já aberto
- No menu “Arquivo”, abra a pasta do “eva_sim”
- Abrir o arquivo (template) *my_script_file.xml*
- Entrar com o código **EvaML**
- Salvar o código

Atividade Prática (Fazendo o *Parsing* de um Script)

Template para a Atividade Prática

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <evaml name="script01" xsi:noNamespaceSchemaLocation="EvaML-Schema/
  evaml_schema.xsd"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4     <settings>
5     </settings>
6     <script>
7     </script>
8   </evaml>
```

Nome do arquivo: *my_script_file.xml*

Atividade Prática (Fazendo o *Parsing* de um Script)

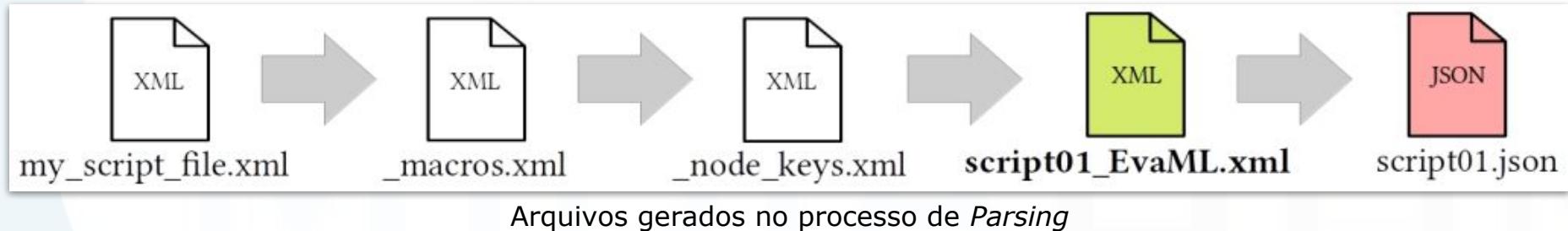
- Abra um terminal no VSCode através do menu “Terminal->Novo Terminal”
- Execute o *Parser* sobre o código EvaML:

```
python3 eva_parser.py my_script_file.xml -c
```

- O terminal deve exibir a seguinte mensagem:

```
marcelo@note-mint:~/Dropbox/Estudo XML$ python3 eva_parser.py my_script_file.xml -c
Step 01 - Processing Macros...
Step 02 - Generating Elements keys...
step 03 - Creating the Elements <link>...
step 04 - Mapping XML nodes and links to a JSON file...
marcelo@note-mint:~/Dropbox/Estudo XML$ █
```

Atividade Prática (Fazendo o *Parsing* de um Script)



Atividade Prática (parte 2)

(Importando e Executando um Script EvaML)

Atividade Prática (Importando e Executando um Script EvaML)

- Para executar o EvaSIM, abra a pasta “eva_sim”
- Clique no executável ***eva_sim.exe*** (no Windows) ou ***eva_sim*** (no Linux)

Atividade Prática (Importando e Executando um Script EvaML)



Botões de controle do **EvaSIM**

1. Ativa o simulador
2. Importa o arquivo de Script (**_EvaML** ou **JSON**)
3. Inicia a execução do Script
4. Interrompe a execução do Script
5. Limpa o terminal de mensagens

Atividade Prática (Importando e Executando um Script EvaML)

- Importar o arquivo *script01_EvaML.xml*
- Executar o script
- Observar os valores gerados nas *tabelas de memória* e no terminal

Atividade Prática (Hands On)

HANDS ON

Perguntas?

Considerações Finais

- Robôs Socialmente Assistivos (**SARs**)
 - Aparência, modalidade de interação, modo de operação e etc
- Trabalhos *acadêmicos* e produtos *comerciais*
 - Terapias para idosos (doenças degenerativas como *Alzheimer*)
 - Crianças com *Transtorno do Espectro Autista (TEA)*

Considerações Finais

- O Robô **EVA** (plataforma *open-source*)
 - Componentes de *hardware* e *software*
 - Interação via voz, lâmpada inteligente, integração de efeitos de luz, reconhecimento de expressões faciais do usuário
- **EvaML**
 - Baseada em *XML*
 - Desenvolvimento de terapias multissensoriais
- O simulador **EvaSIM**
- *Parser EvaML* e o simulador **EvaSIM** (Disponíveis para download)

Referências

- [Cruz Sandoval 2020]** Cruz Sandoval, D. (2020). Robot conversacional como apoyo a intervenciones no farmacológicas para adultos mayores con demencia Conversational robot to support non-pharmacological interventions for people with dementia. Tesis de doctorado en ciencias, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California. 125pp.
- [Mitjans 2020]** Mitjans, A. A. (2020). Affective computation in human-robot interaction. Master thesis, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California.
- [Novák 2010]** Novák, M. (2010). Easy implementation of domain specific language using xml. In Proceedings of the 10th Scientific Conference of Young Researchers (SCYR 2010), Košice, Slovakia, volume 19.
- [Rocha et al. 2021]** Rocha, M., Valentim, P., Barreto, F., Mitjans, A., Cruz-Sandoval, D., Favela, J., and C., M.-S. D. (2021). Towards enhancing the multimodal interaction of a social robot to assist children with autism in emotion regulation. In Proceedings of the 15th EAI International Conference on Pervasive Computing Technologies for Healthcare.

Obrigado!

Robôs Socialmente Assistivos

Desenvolvendo Sessões de Terapia Multissensorial com o Robô **EVA**

Marcelo Marques da Rocha (*marcelo_rocha@midiacom.uff.br*)

Sara Luzia de Melo (*sara.melo@midiacom.uff.br*)

Jesús Favela

Débora Christina Muchaluat Saade

