

Tópico	IA DE RECONHECIMENTO DE GESTOS	
Descrição da Aula	Passando do uso de modelos de rede neural pré-construídos com métodos padrão para o uso de redes neurais para controlar objetos na tela usando gestos manuais, o aluno vai aprender a estender a funcionalidade dos modelos de aprendizado de máquina e moldá-la para se adequar às suas soluções de aplicativos da web.	
Aula	ADV-C124	
Duração da Aula	55 min	
Objetivos	<ul style="list-style-type: none"> • Adicionar código para acessar a webcam • Adicionar código para criar tela e dar posição a ela • Adicionar código para inicializar o modelo Posenet • Adicionar código para a função modelLoaded() • Adicionar código para executar Posenet • Adicionar código para a função gotPoses() 	
Recursos Necessários	<ul style="list-style-type: none"> • Recursos da Professora <ul style="list-style-type: none"> ○ Utilize suas credenciais de login do gmail ○ Fone de ouvido com microfone ○ Webcam ○ Bloco de notas e caneta • Recursos do Aluno <ul style="list-style-type: none"> ○ Utilize suas credenciais de login do gmail ○ Fone de ouvido com microfone (opcional) ○ Webcam ○ Bloco de notas e caneta 	
Estrutura da Aula	Introdução Atividade Dirigida pela Professora Atividade Dirigida pelo Aluno Encerramento Dicas e Sugestões do Projeto	5 min 10 min 30 min 5 min 5 min
SESSÃO DE INTRODUÇÃO - 5 min		
Dizer 		Fazer 

*Oi, fico feliz em te ver novamente! Tudo bem com você?
Empolgado para aprender algo novo?*

REA: Obrigado, sim. Estou ansioso para isso.



A professora inicia a apresentação de slides do slide 1 ao 15.
Consulte as notas da oradora e siga as instruções em cada slide.



ATIVIDADE DA PROFESSORA - 10 min

Ask the student to get into Fullscreen.

A PROFESSORA INICIA O COMPARTILHAMENTO DE TELA

Dizer



Fazer



Por favor, siga o fluxo da aula:

Explique o código JS.

OBSERVAÇÃO: Não explique o código JS todo, apenas o que aparece no documento.

Deixe o aluno fazer o código JS.

Depois, se tiver tempo, faça as Atividades Adicionais.

Na última aula, concluímos HTML e CSS para o aplicativo da web.
Na aula de hoje vamos começar com o código JS no arquivo **main.js** que você baixou na aula passada.

Em todo código JS do p5.js sempre criamos primeiro a canvas (tela), e depois escrevemos o código de acesso à webcam.

Mas hoje vamos começar escrevendo o código JS para acessar a webcam.

1. Primeiro vamos criar a função **setup()** e dentro dela vamos escrever o código JS para acessar a webcam.

```
function setup() {  
  video = createCapture(VIDEO);  
  video.size(550, 500);  
}
```

- Código para acessar a webcam: `createCapture(VIDEO);`
 - **createCapture()** é a função que ajuda a acessar a webcam.
 - E temos que passar **VIDEO** na função **createCapture()**.

Como sabemos, quando acessamos a webcam por padrão em p5.js, ela cria outro **componente** para ela, e toda vez que colocamos a visualização da webcam na tela e ocultamos o **componente** extra. Mas desta vez a tela está ocupada, o que significa que na tela já estamos desenhando o quadrado, então não podemos colocar essa visualização da webcam na tela.

É por isso que precisamos manter esse **componente** extra e exibi-lo na página da web.

Qualquer coisa que seja exibida em uma página web precisa ter um tamanho adequado.

Portanto, precisamos dar o tamanho adequado a esse **componente** extra.

É por isso que vamos armazenar esse **componente** extra dentro de uma variável, e depois podemos usar essa variável para manipular seu tamanho.

- Armazenando a visualização da webcam em uma variável:

```
video = createCapture(VIDEO);
```

- Agora vamos dar tamanho à visualização da webcam:

```
video.size(550, 500);
```

- Dei **largura** como **550px** e **altura** como **500px** para a tela.
- Não é obrigatório fornecer os mesmos valores, mas sugiro que o faça.
- Saída:



Visualização da webcam

Desenvolvido por: BYJU'S Team

Agora vamos criar a tela.

```
function setup() {  
  video = createCapture(VIDEO);  
  video.size(550, 500);  
  
  canvas = createCanvas(550, 550);  
  canvas.position(560, 150);  
}
```

- Primeiro vamos criar a tela usando `createCanvas`, e armazená-la dentro de uma variável. Lembre-se de que Canvas é um elemento GUI e estamos armazenando-o dentro de uma variável, pois assim podemos usar a notação de ponto e manipular a posição do elemento Canvas.

```
canvas = createCanvas(550, 550);
```

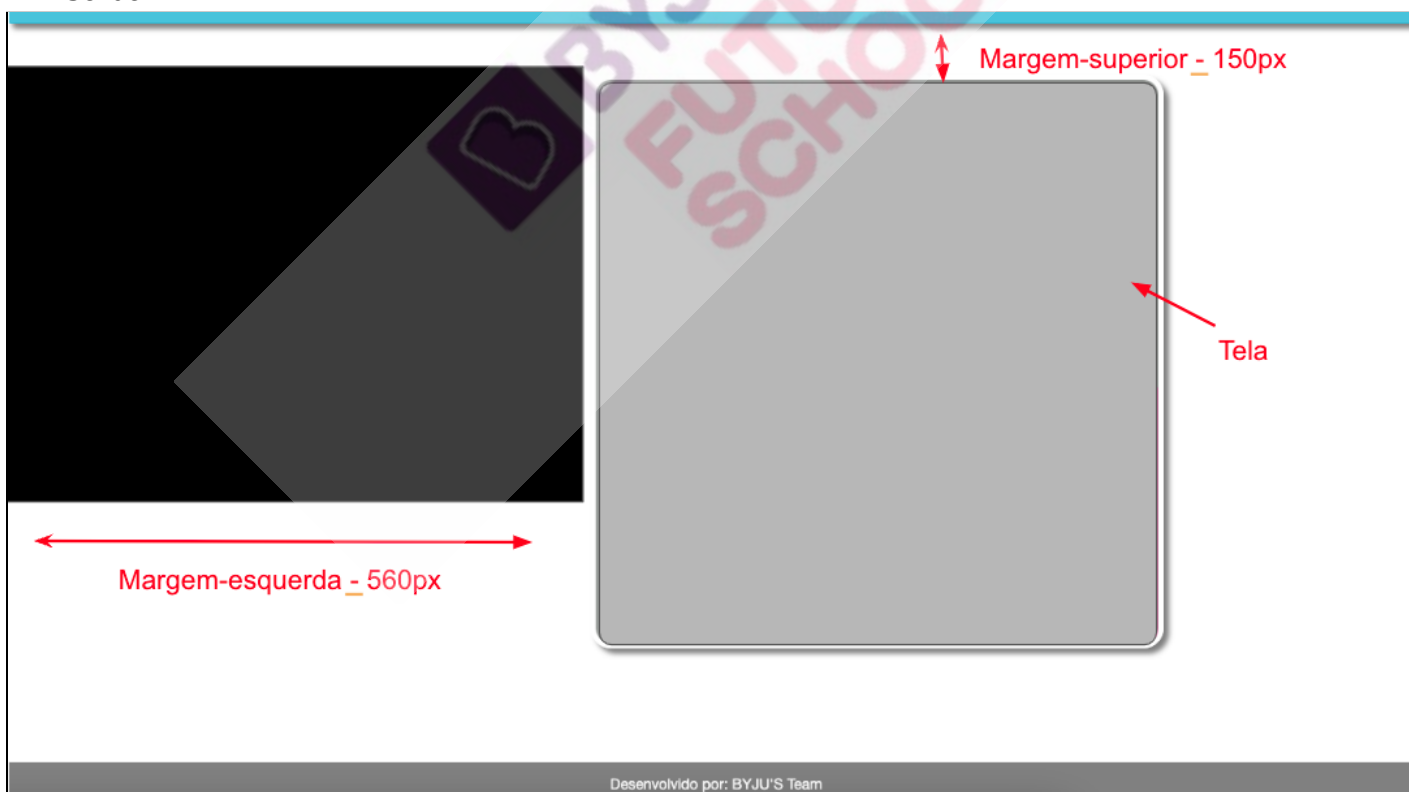
- Dei **largura** como **550px** e **altura** como **550px** para a tela.

- Não é obrigatório fornecer os mesmos valores, mas sugiro que os dê.
- Agora posicione esta tela de forma que seja colocada ao lado da visualização da webcam. Definimos a largura da **visualização da webcam como 550px**. Portanto, a tela deve ter 550px ou mais de margem esquerda. Isso ajudará a deixar a tela ao lado da visualização ao vivo da webcam.
- Usaremos a função **position()** para dar posição à tela. A função **position()** é uma função predefinida de p5.js, usada para dar posição a qualquer elemento como tela, vídeo e muitos outros elementos.

Sintaxe da função **position()**:

position(margem-esquerda, margem-superior)

- Código para dar posição à canvas: `canvas.position(560,150);`
- Demos **margem-esquerda** como 560px, porque o tamanho da visualização da webcam é de 550px.
- Demos **margem-superior** como 150px, porque deve haver alguma lacuna entre o cabeçalho do site e a tela.
- Saída:



2. Agora vamos definir uma função **draw()** e dentro dela vamos dar cor de fundo para a tela.

```
function setup() {
  video = createCapture(VIDEO);
  video.size(550, 500);

  canvas = createCanvas(550, 550);
  canvas.position(560, 150);
}

function draw() {
  background('#969A97');
}
```

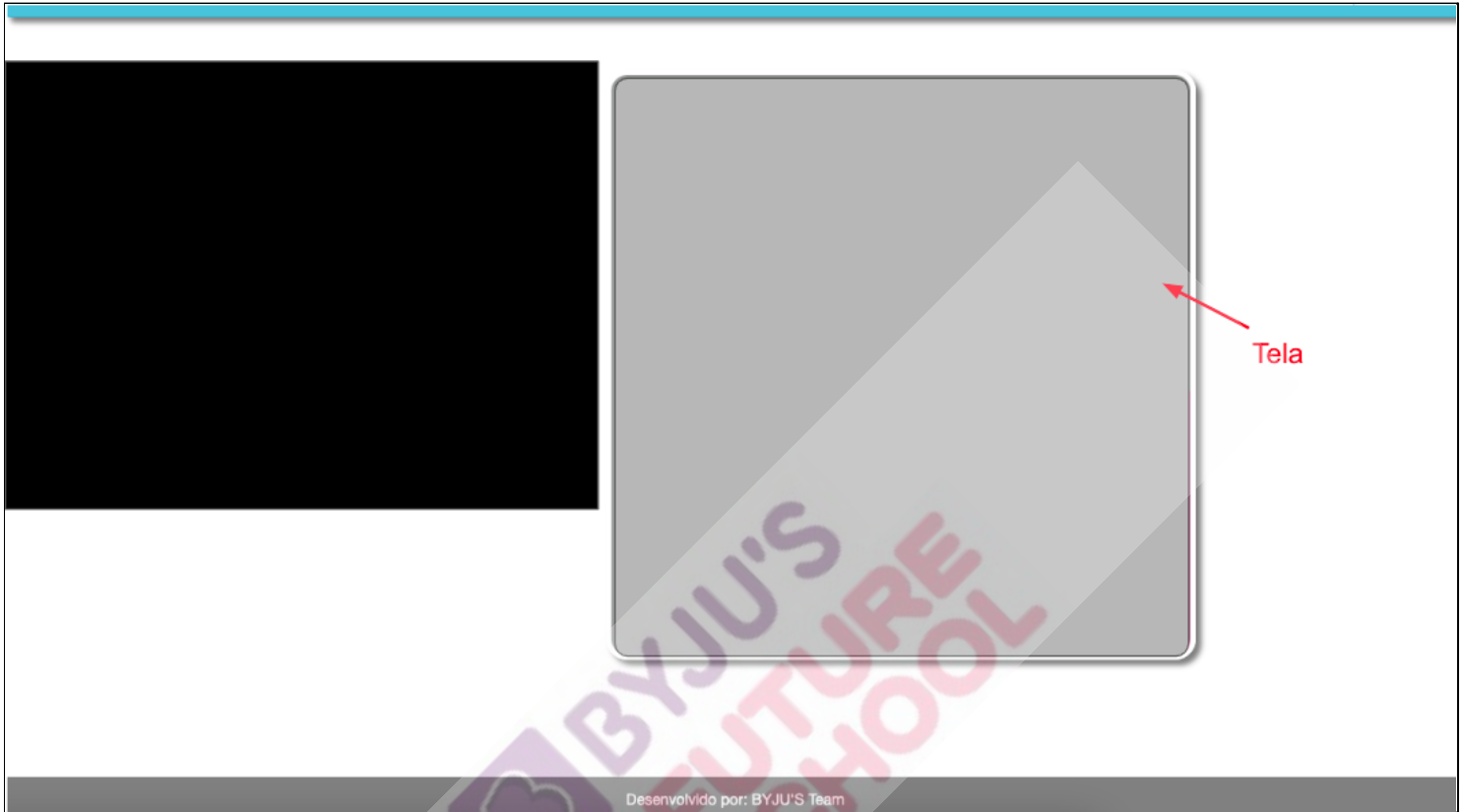
- **background()** é uma função predefinida de p5.js, que é usada para dar cor à tela.
 - Dentro desta função você pode passar código RGB, código HEX, código RGBA.
 - Usamos um código HEX para dar cor à tela. `background('#969A97');`
 - `'#969A97'` - este é um código HEX da cor cinza-escuro. Não é obrigatório usar esta cor, você pode usar a cor de sua escolha.

Para obter o código HEX de qualquer cor, basta pesquisar no google: “código HEX para **nome da cor**”, por exemplo:

A screenshot of a Google search for "código hex para preto". The search results show a table titled "Confira a tabela completa de cores HTML em Hexadecimal, RGB e nome em inglês." The table has three columns: "Cor", "Código RGB", and "Código Hexadecimal". It lists several colors including White, Black, DarkSlateGray, and DimGrey. At the bottom, it says "Mais 125 linhas • 19 de jan. de 2019".

Cor	Código RGB	Código Hexadecimal
White	255 255 255	#FFFFFF
Black	0 0 0	#000000
DarkSlateGray	47 79 79	#2F4F4F
DimGrey	105 105 105	#696969

- Saída:



Excelente! Configuramos a visualização da Webcam, tela e demos cor de fundo a ela.

3. Agora vamos escrever o código para configurar o Posenet.

P: Onde vamos escrever o código para configurar o Posenet? Será na função `setup()`, na função `draw()` ou fora delas e por quê?

R: Vamos escrever o código para configurar Posenet na função `setup()`, porque na função `setup()` todo o código de configuração é escrito.


```
function setup() {
  video = createCapture(VIDEO);
  video.size(550, 500);

  canvas = createCanvas(550, 550);
  canvas.position(560, 150);

  poseNet = ml5.poseNet(video, modelLoaded);
}
```

- Primeiro vamos definir uma variável `poseNet =`
- Então vamos escrever o nome da biblioteca `ml5.js` `= ml5.`
- Em seguida, escrever `poseNet`, assim: `= ml5.poseNet()`;
 - `poseNet` é uma função predefinida de `ml5.js` usada para inicializar o modelo Posenet. A linha de código acima inicializará a função `poseNet()`.
 - Normalmente, para chamar/inicializar qualquer função `ml5.js`, usamos apenas desta forma: `ml5.js.functionName()`.
 - Agora dentro desta função **Posenet()** precisamos passar dois parâmetros:
 - 1º é a entrada (significa a imagem OU o vídeo onde queremos que a Posenet execute ações) no nosso caso, a visualização da webcam.

```
poseNet = ml5.poseNet(video);
```

- 2º é a função para confirmar que o PoseNet foi inicializado.

```
poseNet = ml5.poseNet(video, modelLoaded);
```


4. Escreva o código da função `modelLoaded()` fora da função `setup()`.

```
function setup() {  
  video = createCapture(VIDEO);  
  video.size(550, 500);  
  
  canvas = createCanvas(550, 550);  
  canvas.position(560, 150);  
  
  poseNet = ml5.poseNet(video, modelLoaded);  
}  
  
function modelLoaded() {  
  console.log('PoseNet Is Initialized!');  
}
```

- A função **`modelLoaded()`** será usada para mostrar o status do PoseNet (ou seja, o PoseNet is initialized (é inicializado)) no console.
- Saída na tela do console:

PoseNet Is Initialized

main.js:17

- Agora poseNet é inicializado, também demos a entrada (passamos a visualização da webcam na qual queremos que a função `Posenet()` execute ações) para a função `Posenet()`.

Agora vamos escrever o código para executar o Posenet e obter os resultados.

5. Então agora vamos escrever o código para executar o Posenet, e este código virá na função **setup()**.

```
function setup() {
  video = createCapture(VIDEO);
  video.size(550, 500);

  canvas = createCanvas(550, 550);
  canvas.position(560, 150);

  poseNet = ml5.poseNet(video, modelLoaded);
  poseNet.on('pose', gotPoses);
}

function modelLoaded() {
  console.log('PoseNet Is Initialized!');
}
```

- Primeiro vamos escrever a variável que está segurando a inicialização do posnet que é: **poseNet**.
- Depois vamos escrever a função **on()**, que é uma função predefinida de ml5.js usada para iniciar a execução do Posenet: **poseNet.on()**;
- Então dentro deles precisamos passar 2 parâmetros.
 - Precisamos obter a **pose** (coordenadas x e y das 17 partes do corpo, que discutimos na aula passada).
 - Portanto, no 1º parâmetro, por padrão, escreveremos **pose** entre aspas simples:

```
poseNet.on('pose');
```

- No 2º parâmetro passaremos uma função "gotPoses" que obterá todas as poses (coordenadas x e y das 17 partes do corpo, que discutimos na aula passada) do modelo.

```
poseNet.on('pose', gotPoses);
```

- Vamos definir a função `gotPoses` no próximo ponto.
6. Escreva o código JS para a função `gotPoses()`:

```
function gotPoses(results)
{
  if(results.length > 0)
  {
    console.log(results);
  }
}
```

- Primeiro defina a função `gotPoses()`:

```
function gotPoses(results)
```

- E dentro da função `gotPoses()` escreva os resultados.
 - Os resultados contêm as coordenadas x e y de todas as 17 partes do corpo.

Vamos escrever todo o código dentro de uma condição `if`.

- Então, primeiro defina uma condição `if` na função `gotPoses()`, que verificará se o comprimento dos resultados é maior que 0 e então entrará na condição “if”, significa: **Se os resultados estiverem vazios, nada acontecerá.**

```
if(results.length > 0)
{
```

- O **objetivo** de escrever esta condição 'if' é que, se ocorrer alguma das situações mencionadas abaixo, nosso código não fará nada, isso impedirá nosso aplicativo de gerar erros e interromper o processo de Posenet. Situações:
 - A webcam não inicia por algum motivo;

- Não houver ninguém na frente da webcam, o que significa que o Posenet não poderá detectar ninguém;
- Ou o Posenet começa antes da webcam, o que também resultará no mesmo que o Posenet não conseguir detectar ninguém;
- Ou outros erros.

Se não usarmos esta condição 'if' e se ocorrer alguma das situações acima, o Posenet será interrompido e nosso aplicativo apresentará um erro.

- Agora na condição if vamos consolar os resultados. Os resultados têm todos os dados provenientes do Posenet, que são as coordenadas x e y de todas as 17 partes do corpo.

```
console.log(results);
```

- Saída na tela do console:



E então podemos ler esta matriz de resultados e buscar as coordenadas das partes do corpo conforme nossa necessidade.

Na próxima aula vamos aprender como buscar as coordenadas x e y do nariz. Também aprenderemos como buscar as coordenadas x do pulso direito e esquerdo.

A professora encerra o compartilhamento de tela

ATIVIDADE DO ALUNO  - 30 min

Agora é sua vez.

- Peça ao aluno para pressionar a tecla ESC para retornar ao painel
- Auxilie o aluno a iniciar o compartilhamento de tela
- A professora entra em modo tela cheia

O aluno inicia o compartilhamento de tela

Dizer



Fazer



***Observação: Por favor, siga os seguintes passos.**

Inicie a programação JS no arquivo **main.js** que iremos programar nas próximas aulas.

Mantenha os arquivos de código guardados, pois enviaremos esses arquivos para o github quando concluirmos a criação do site.

Atividade do Aluno 1 - DIAGRAMA DO CÓDIGO

Referência para Atividade do Aluno 1- REFERÊNCIA DA FUNÇÃO BACKGROUND() DE p5.js

Referência para Atividade do Aluno 2- REFERÊNCIA IF ELSE

O aluno deve iniciar a programação JS no arquivo **main.js**, baixado na aula anterior.

Se o tempo permitir, encoraje o aluno a adicionar, livremente, mais objetos.

A professora auxilia o aluno a encerrar o compartilhamento de tela

SESSÃO DE ENCERRAMENTO - 5 min

A professora inicia a apresentação de slides do slide 16 ao 18.
Consulte as notas da oradora e siga as instruções em cada slide.

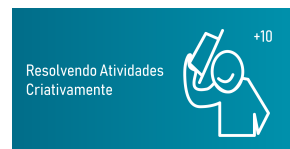


Você receberá tiradas de chapéu pelo excelente trabalho!

Aplauda o aluno e dê tiradas de chapéu.

(Dê, no mínimo, 2 tiradas de chapéu)

Pressione o ícone de Tirada de Chapéu para **Resolvendo Atividades Criativamente.**



Pressione o ícone de Tirada de Chapéu para **Ótima Pergunta.**



Pressione o ícone de Tirada de Chapéu

para “**Você se Concentrou**”.



DICAS E SUGESTÕES DO PROJETO - 5 min

A professora inicia a apresentação de slides dos slides 19 e 20.
Consulte as notas da oradora e siga as instruções em cada slide.



Dizer



Fazer



MANIPULADOR DE FONTE - 2

Objetivo do Projeto:

Na aula de hoje, completamos o código para inicializar a Posenet e executar o modelo.

Neste projeto você tem que escrever código para acessar a webcam, criar uma tela, inicializar e executar o modelo Posenet.

História:

Na escola Santa Fada, os alunos não gostavam de ler o quadro de avisos e geralmente perdiam as mensagens importantes que os professores/diretores tinham que transmitir. Então o diretor decidiu fazer um divertido quadro de avisos para os alunos onde um aluno pode aumentar ou diminuir o tamanho da fonte. Se eles não puderem ver o aviso escrito em letras pequenas, eles podem usar os pulsos para aumentar a fonte e ler o que está escrito.

O diretor gostou da interface do usuário que você criou no projeto 123, agora ele está animado para ver como a visualização da webcam e os dados do modelo Posenet podem ser vistos no resultado.

Observação: Você pode atribuir o projeto ao aluno na própria aula, clicando no botão Atribuir Projeto que está disponível na guia Projetos.


Abra o link da **Solução do Projeto** e demonstre o projeto ao aluno.

Eu sei que você vai se sair muito bem.

Até breve!

A Professora Para o Compartilhamento de Tela

ATIVIDADE ADICIONAL

A professora inicia a apresentação de slides  do slide 21 ao 25.

ATIVIDADE ADICIONAL DO ALUNO

O Aluno Inicia o Compartilhamento de Tela

Para a solução de todas as Atividades Adicionais, abra a **Atividade da Professora - 4** e navegue até o número de aula **C124**.

Atividade Adicional 1 -

Execute a **Atividade do Aluno-2** do **painel**
AS **TAREFAS** e **DICAS** são mencionadas no próprio website.

Atividade Adicional 2 -

Execute a **Atividade do Aluno-3** do **painel**
AS **TAREFAS** e **DICAS** são mencionadas no próprio website.

Atividade Adicional 3 -

Execute a **Atividade do Aluno-4** do **painel**
AS **TAREFAS** e **DICAS** são mencionadas no próprio website.

Atividade Adicional 4 -

Execute a **Atividade do Aluno-5** do **painel**
AS **TAREFAS** e **DICAS** são mencionadas no próprio website.

Atividade Adicional 5 -

Execute a **Atividade do Aluno-6** do **painel**
AS **TAREFAS** e **DICAS** são mencionadas no próprio website.

A Professora Orienta o Aluno a Parar o Compartilhamento de Tela



A professora termina a apresentação de slides no slide 26.
Consulte as notas da oradora e siga as instruções em cada slide.

<Dê tchau ao aluno>

“Isso nos traz ao final da nossa aula!
Obrigado por ser um aluno incrível!
Nos vemos na próxima aula!
Continue programando!”

✕ Terminar Aula

A professora clica em

Atividade	Nome da Atividade	Links
Atividade da Professora 1	LINK DO SITE	https://c-amilarorigues.github.io/ADV-V2-Desenho-Tempo-Real/
Atividade da Professora 2	DIAGRAMA DO CÓDIGO	https://s3-whjr-curriculum-uploads.w-hjr.online/7ef759cb-dce7-41a9-9854-8f930438e969.pdf
Atividade da Professora 3	CÓDIGO FONTE	https://drive.google.com/drive/folders/192lv-RkMO2dpkm2TjNg8SdzFecQn93Jj?usp=sharing
Atividade da Professora 4	SOLUÇÃO DAS ATIVIDADES ADICIONAIS	https://docs.google.com/spreadsheets/d/e/2PACX-1vSG0V0IQyGx7Zo7VJgaWAHYDpalldCZ-RepABc_slelct65DaM-R9YF0vCbS5JFdsF3MEFQ-WLslsRc/pubhtml
Atividade do Aluno 1	DIAGRAMA DO CÓDIGO	https://s3-whjr-curriculum-uploads.w-hjr.online/7ef759cb-dce7-41a9-9854-8f930438e969.pdf
Referência para Atividade do Aluno 1	REFERÊNCIA DA FUNÇÃO BACKGROUND() DE	https://p5js.org/reference/#/p5/background

	p5.js	
Referência para Atividade do Aluno 2	REFERÊNCIA IF ELSE	https://www.w3schools.com/js/js_if_else.asp
Solução do Projeto	MANIPULADOR DE FONTE - 2	https://s3-whjr-curriculum-uploads.w-hjr.online/54f0073a-6bd0-4729-8cc5-c35133029f34.pdf Apenas mostre a funcionalidade de acessar os resultados da webcam e Posenet na tela do console.
Link de Recurso Visual Referência da Professora	Recurso Visual COM NOTAS	https://s3-whjr-curriculum-uploads.w-hjr.online/10bdf85d-1627-4456-b2ee-be6ee737b011.html
Link de Recurso Visual Referência da Professora	Recurso Visual SEM NOTAS	https://s3-whjr-curriculum-uploads.w-hjr.online/eeb84ea4-c99a-481a-9229-5fc5b017099d.html
Link Questionário de Aula Referência da Professora	Questionário de Aula	https://s3-whjr-curriculum-uploads.w-hjr.online/75196561-ff54-43a1-ba00-b27f35b9f974.pdf