ANÁLISE PELA WEBCAM

1. CRIAR DETECTOR

Descrição: Criação dos nodes de Vídeo e de Renderização no DOM.

Obs.: Os tópicos desta seção estão apresentados na ordem em que devem ser programados.

- ***Elemento no DOM onde o vídeo da webcam será renderizado***

```
let divRoot = document.getElementById('main')
```

- ***Tamanho em pixels do vídeo***

```
let width = 640
let height = 480
```

- ***Zoom da face no frame que será renderizado***

  obs.: LARGE_FACE = Zoom alto | SMALL_FACE = Zoom baixo

```
let faceMode = affdex.FaceDetectorMode.LARGE_FACES
```

- ***Criação do objeto para detecção, baseado nos parâmetros especificado anteriormente***

```
let detector = new affdex.CameraDetector(
      divRoot, width, height, faceMode
)
```

2. CONFIGURAR CALLBACKS

Descrição: Callbacks são como âncoras ancoradas no ciclo de execução, elas executam funções estabelecidas no momento de sua chamada.

2.1. onInitialize

Descrição: *É* executada após a inicialização do detector

- ***Sucesso***

```
detector.addEventListener("onInitializeSuccess", function() {
      console.log('O detector foi inicializado com sucesso')
})
```

- ***Falha***

```
detector.addEventListener("onInitializeFailure", function() {
      console.log('Houve falha na inicialização')
})
```

2.2. onImageResults

Descrição: É executado quando um frame do vídeo fornecido pela webcam é processado

- ***Sucesso***
    - *Parâmetros recebidos:*

      faces: Dicionário das faces identificadas. Para cada face um dicionário com as emoções e expressões detectadas, além das coordenadas dos pontos de detecção
      image: Objeto com dados sobre o quadro processado
      timestamp: Tempo em segundos do momento em que a imagem foi capturada

```
detector.addEventListener("onImageResultsSuccess",
      function(faces, image, timestamp) {
            console.log(`${faces}, ${image}, ${timestamp}`)
      }
)
```

- ***Falha***
    - *Parâmetros recebidos:*

      image: Objeto com dados sobre o quadro processado
      timestamp: Tempo em segundos do momento em que a imagem foi capturada
      errDetail: String contendo o erro encontrado

```
detector.addEventListener("onImageResultsFailure",
      function(image, timestamp, errDetail) {
            console.log(`${image}, ${timestamp}, ${errDetail}`)
      }
)
```

2.3. onReset

Descrição: É executada após a execução da chamada de reset "detector.reset()"

- ***Sucesso***

```
detector.addEventListener("onResetSuccess", function() {
      console.log('O reset foi executado com sucesso')
})
```

- ***Falha***

```
detector.addEventListener("onResetFailure", function() {
      console.log('Houve falha na execução do reset')
})
```

2.4. onStop

Descrição: É executada após a execução da chamada de stop "detector.stop()"

- ***Sucesso***

```
detector.addEventListener("onStopSuccess", function() {
      console.log('O stop foi executado com sucesso')
})
```

- ***Falha***

```
detector.addEventListener("onStopFailure", function() {
      console.log('Houve falha na execução do stop')
})
```

2.5. onWebcamConnect

Descrição: É executada quando o detector tenta se conectar com a webcam

- ***Sucesso***

```
detector.addEventListener("onWebcamConnectSuccess", function() {
      console.log("Conexão com webcam executada com sucesso")
})
```

- ***Falha***

```
detector.addEventListener("onWebcamConnectFailure", function() {
      console.log("Houve falha na conexão com a webcam")
})
```

3. ESCOLHER CLASSIFICADORES

Descrição: Classificadores são as informações devolvidas pelo Affectiva no callback "onImageResults" a cerca das faces identificadas

- **Ativar ou desativar classificadores específicos:**
    - ***Desativar detecção da expressão sorriso***
      `detector.detectExpressions.smile = false`
    - ***Ativar detecção da emoção felicidade***
      `detector.detectEmotions.joy = true`
    - ***Desativar detecção da aparência gênero***
      `detector.detectExpressions.smile = false`
- **Ativar ou desativar categorias de classificores inteiras**
  obs.: Estes métodos se comportam como toggle “bool = !bool”
    - `detector.detectAllExpressions()`
    - `detector.detectAllEmotions()`
    - `detector.detectAllEmojis()`
    - `detector.detectAllAppearance()`

[Link para lista completa de classificadores]

4. MÉTODOS DE EXECUÇÃO

Descrição: Após a configuração dos callbacks e dos classificadores desejados deve-se executar os métodos de execução

- **start**

  Descrição: Inicializa a detecção

  ```
  detector.start()
  ```

- **stop**

  Descrição: Finaliza a detecção

  ```
  detector.stop()
  ```

- **reset**

  Descrição: reseta a detecção

  ```
  detector.reset()
  ```