

Camera Oximeter

- 16/0032458 – José Luiz Gomes Nogueira
 - 16/0008166 – Hugo Nascimento Fonseca
 - 16/0019311 – Victor André Gris Costa
-

Ferramentas

02

Softwares utilizados



- React Native:
 - React 16.x
 - Android SDK 28
- TypeScript
- Java
- OpenCV 4 em Java

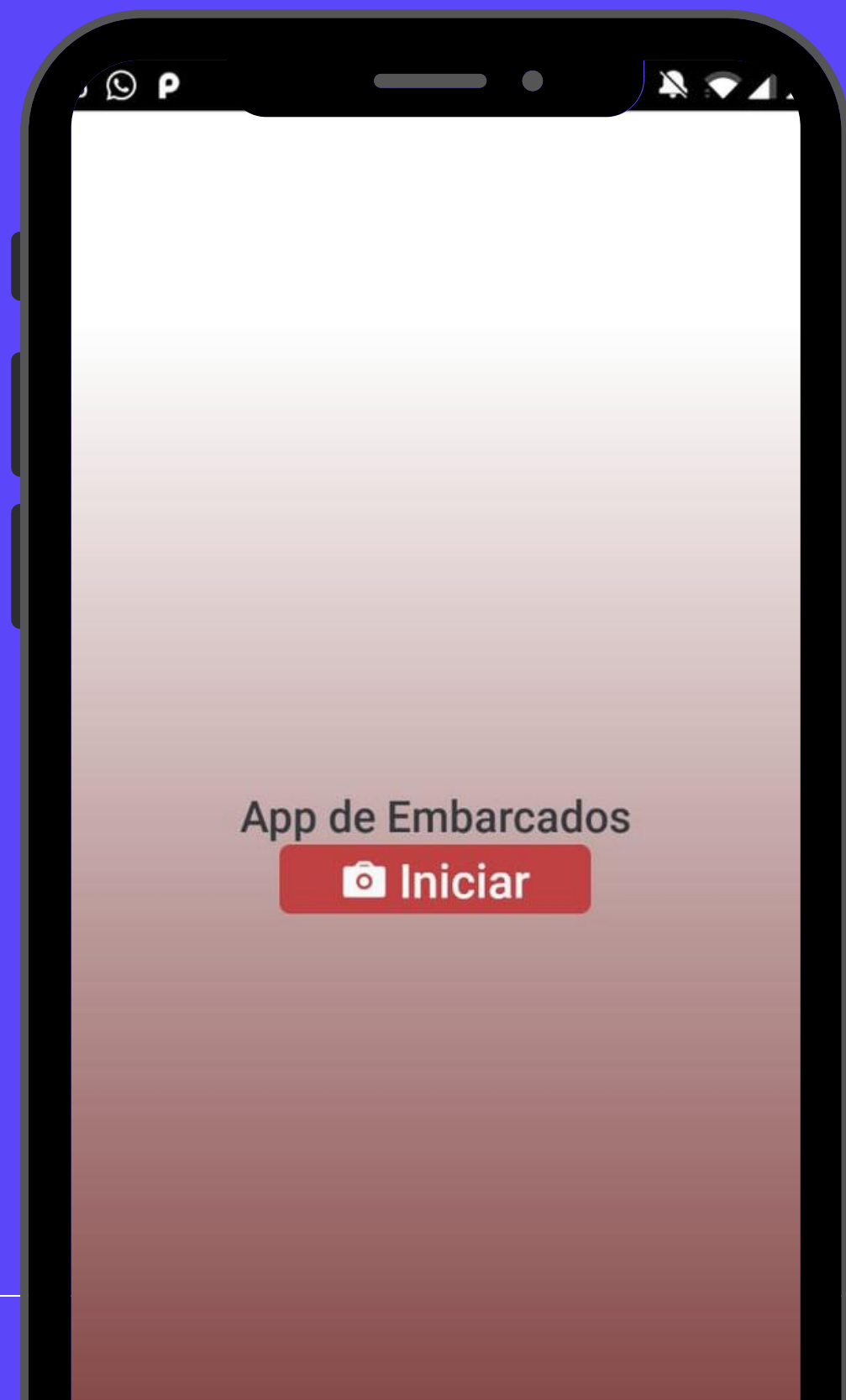
Etapas concluídas do projeto

- 1 – Criar um aplicativo react-native
- 2 – Abrir a câmera pelo aplicativo e capturar imagem
- 3 – Realizar detecção de face
- 4 – Estimar posição da região de interesse (testa)
- 5 – Instalar e configurar a openCV no projeto como modulo nativo em java
- 6 – Recortar a região de interesse e salvar



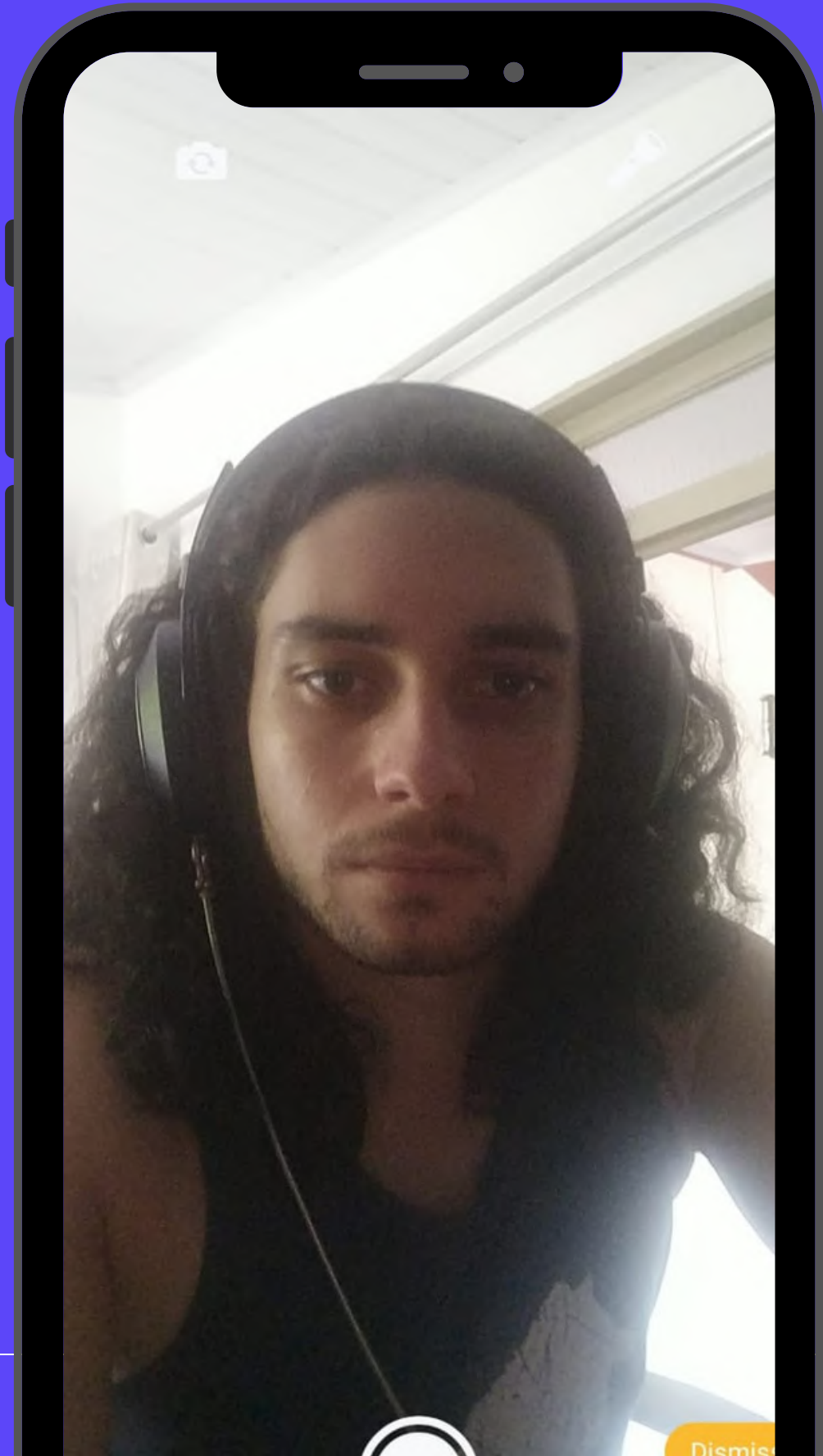
1 – Criar um aplicativo react-native

04



Criado um aplicativo react-native modificado para typescript e com integração ao expo.

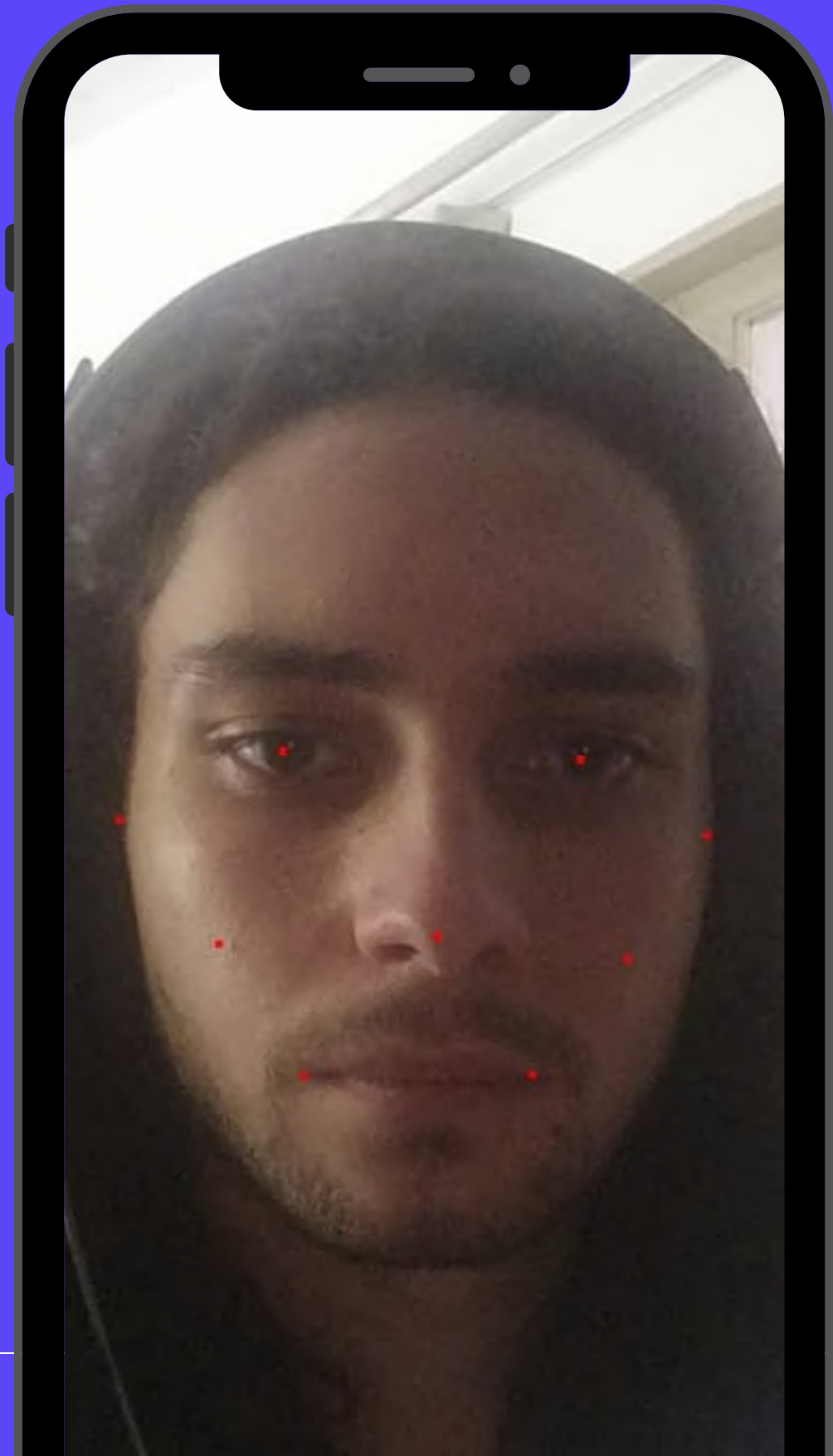
2 – Abrir a câmera pelo aplicativo e capturar imagem



Adicionado ao app uma segunda tela que abre a câmera utilizando o módulos do **expo**

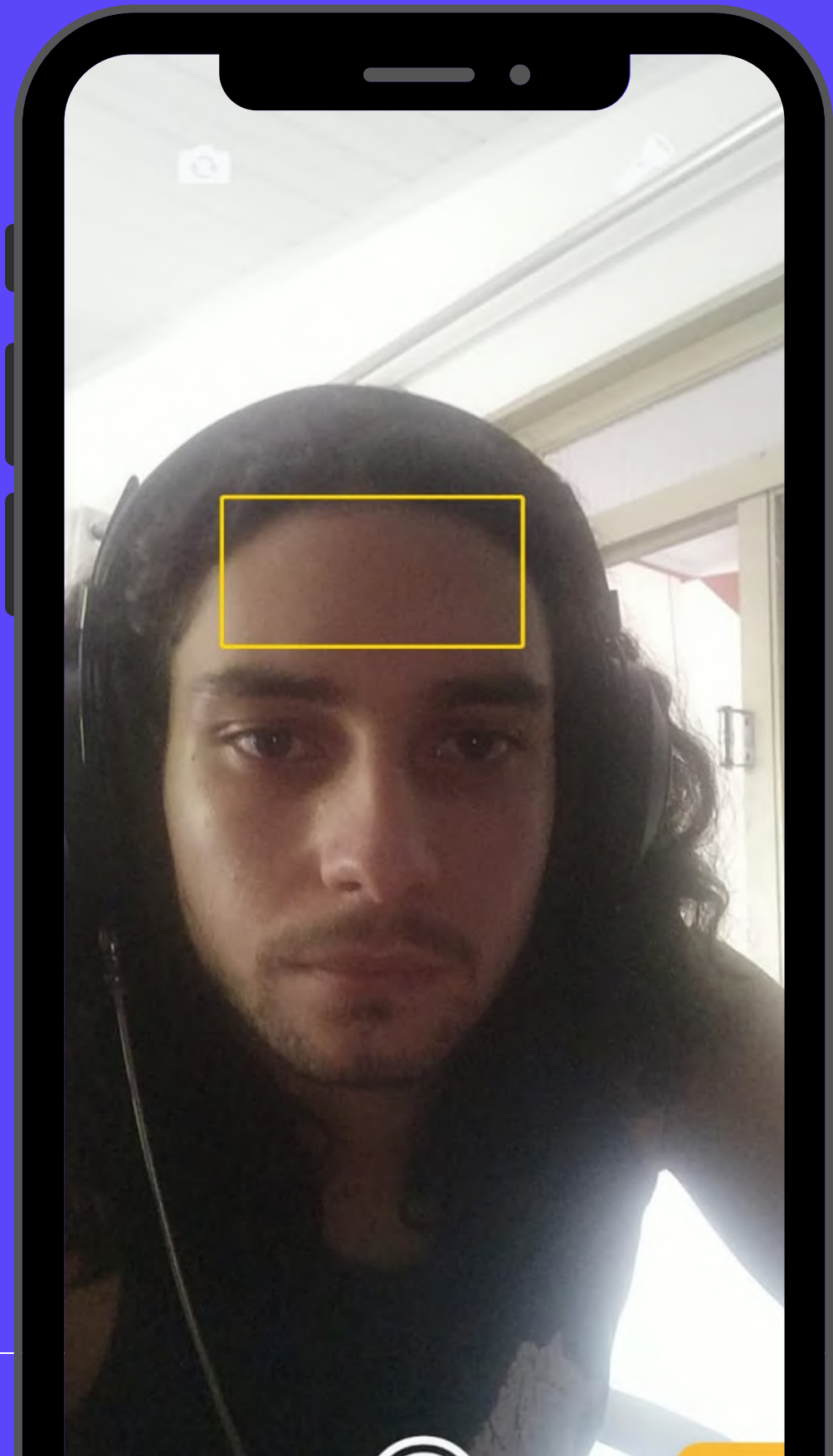
3 – Realizar detecção de face

06



Utilizando de um modulo **expo**, foi realizado a detecção dos pontos que formam o retângulo da face e pontos do rosto, como olhos, boca.

4 – Estimar posição da região de interesse (testa)



Utilizando dos pontos da face detectada, foi criado um retângulo na região da testa e exibido por cima do frame da câmera

5 – Instalar e configurar a openCV no projeto como modulo nativo em java

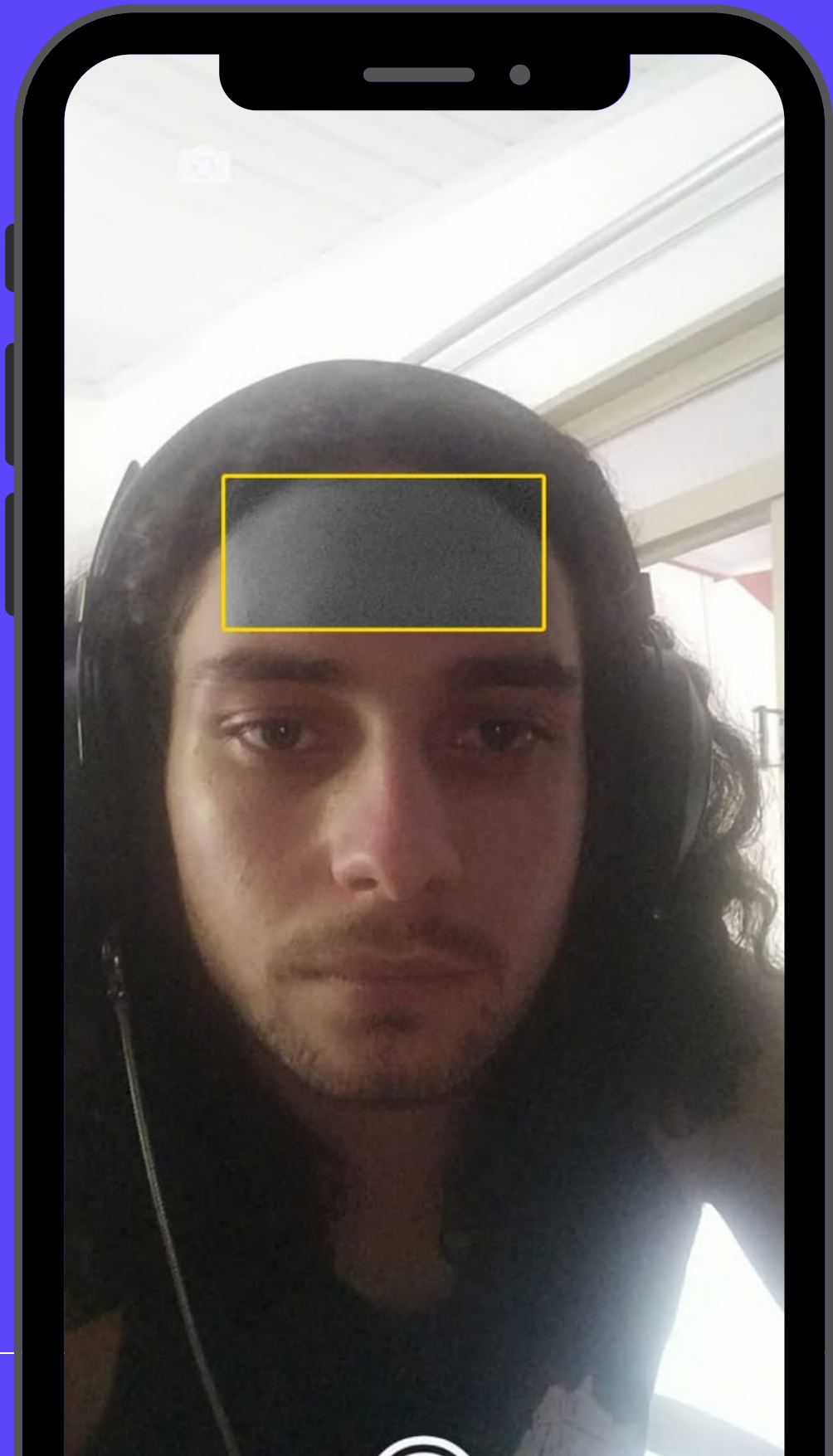
08



Utilizando alguns tutoriais da internet, foi inserido o módulo nativo da openCV 4 em java no projeto react-native e criado as classes para interface com os módulos em react.

6 – Recortar a região de interesse e salvar

09



Após algumas voltas e testes de arquitetura, o recorte e armazenamento dos frames recortados foram feitos inteiramente em uma nova classe java, de forma nativa.

Para isso foi necessário passar a imagem em string base64 para o modulo nativo com os dados dos pontos do rosto detectados usando o expo e um fator de escala da imagem, uma vez que a resolução da imagem e os valores de pixels dos pontos retornando não correspondem e variam de celular para celular

Arquitetura do projeto

10

Classes principais e suas funções

Home (react class)	Oximeter (react class)	RNOpenCvLibraryModule (native java class)	OpenCV (react class)
Tela	Tela	Módulo	Módulo
Tela inicial com o botão que leva para a tela com a câmera	Tela com a câmera aberta, nela é realizado a detecção de face com o expo e a chamada do modulo nativo para realização dos cálculos	Importa o modulo da openCV para manipulações. Implementa funções para: <ul style="list-style-type: none">– Conversões string base64– recorte da região– savamento dos frames E expõe uma função que realiza todo o fluxo de recorte e calculo do app para o react-native, com dados e imagens	Modulo que encapsula a função nativa criada para as classes em react-native no formato de promisse e define a interface typescript da função

Avanços pendentes

Remover ruídos da imagem

Realizar o cálculo

Acelerar a captura de frames, removendo a detecção do expo pela
detecção da openCV
