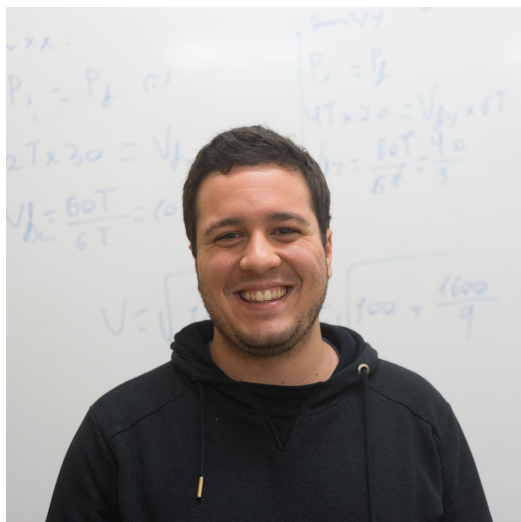


# Pizza Time

Computação Móvel - Módulo Flutter



Tomás Freitas - 84957



Francisco Petronilho - 89241

# Motivação

A motivação deste trabalho vem da vontade de criação de uma aplicação “simples” que demonstrasse ao mesmo tempo os conhecimentos adquiridos. Esta aplicação usa ligações bluetooth entre dispositivos, sensores corporais, sensores do telemóvel, utilização de API's externas e alguma persistência de dados.

## Requisitos e Funcionalidades

PizzaTime consiste numa aplicação que tem uma função *straightforward*, orientar o utilizador para a pizzeria mais próxima, contém também um sistema de descontos aplicados proporcionais ao batimento cardíaco máximo do utilizador. Por último, contém também um sistema de favoritos.

A aplicação desenvolvida tinha como requisito e desafio principal a recolha e envio de informação relativa ao batimento cardíaco do utilizador em tempo real para um telemóvel emparelhado.

Também decidimos em implementar certas features:

- Utilização de serviços externos para obtenção de Pizzarias (Nome, Localização, etc.)
- Utilização de QR Codes para utilização do cupão de desconto
- Utilização de uma app companheira, também desenvolvida em Flutter, para a captura e envio de dados relativos ao batimento cardíaco
- Implementar um sistema de favoritos locais recorrendo a shared preferences para a sua persistência
- Dar feedback contínuo ao utilizador acerca da distância restante usando Text-to-speech

# Arquitetura

## Aplicação wearable:

A aplicação wearable consiste em apenas um widget que captura e envia os dados relativos ao batimento cardíaco.

### Captura:

Devido à dificuldade de implementar uma abordagem que utilizasse os serviços da Google Fit, decidimos uma abordagem de leitura direta dos valores do sensor. Depois de uma pesquisa exaustiva dos módulos que permitissem esta feature, nenhum deles funcionava a 100%.

Para resolução deste problema utilizou-se um MethodChannel que nos permite, através da camada do flutter, chamar funções nativas do Android e obter assim os dados do batimento cardíaco.

### Envio:

Para o envio dos dados capturados foi utilizada a [Wearable Data Layer](#) que permite enviar dados entre telemóvel e relógio emparelhado.

## Aplicação Android:

A aplicação Android contém 3 páginas:

- MyApp: Raíz da aplicação
- BuildCompass: Widget que contém informações do restaurante quando temos permissões de localização
- BuildPermissionSheet: Widget que pede permissões ao utilizador
- GenerateQrPage: Widget que mostra o QR Code do desconto
- FavoritesPage: Widget que mostra os restaurantes favoritos

Estes widgets estão todos implementados no main.dart que usa Navigator e MaterialPageRoute para a navegação.

Inicialmente é obtida a localização com o objetivo de procurar pizzarias na área do utilizador, esta localização é obtida através do pacote Geolocator e passada para o pacote mapbox\_search que utiliza a API Mapbox.

Ao longo do tempo a UI é atualizada consoante as mudanças de direção e movimentos do utilizador, isto é conseguido através de dois nested StreamBuilders que utilizam streams de eventos da bússola e GPS do

telemóvel. Assim, a qualquer movimento é suavemente atualizada a bússola e a distância para o restaurante.

Para gerência de dados, visto que são dados locais, optou-se pela utilização do módulo `shared_preferences` pela sua facilidade de uso.

Posto isto, apesar da nossa aplicação Android teoricamente suportar também iOS, a mesma não terá as mesmas funcionalidades com um Apple Watch devido à captura do batimento cardíaco com código Android Nativo

## Resultados e Dificuldades

A implementação dos objetivos propostos foi concluída com sucesso apesar de algumas dificuldades complexas pelo caminho, nomeadamente:

- Dificuldade de interação com o Google Fit, não sendo possível no final
- Migração de packages para NullSafe devido à idade dos packages
- Desenvolvimento de aplicações para WearOS envolve muito tempo de espera para a compilação e transferência da aplicação

## Contribuições

Ambos os membros contribuíram com 50% do trabalho realizado, Tomás Freitas desenvolveu a aplicação WearOS, QR Codes e os favoritos; Francisco Petronilho desenvolveu o restante da aplicação Android: API's, UI/UX, Text-to-speech, navegação entre widgets.

## Manual

A aplicação foi concebida para ser simples, a única coisa que o utilizador tem de fazer é ter um relógio wearOS emparelhado com o telemóvel e iniciar a aplicação. Encontra no fundo da app dois botões que permitem navegar pelas restantes features.