FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO




# Life Bracelet

**Tiago Simões**

DISSERTAÇÃO

Orientador: Prof. Doutor Miguel Velhote Correia

Co-orientador: Eng. João Paredes

26 de Julho de 2019

© Tiago Simões, 2019

# Life Bracelet

**Tiago Simões**

DISSERTAÇÃO

26 de Julho de 2019

# Resumo

A qualidade de vida tem aumentado significativamente devido à tecnologia e aos grandes avanços, não só da engenharia mas também da medicina. A monitorização é a chave para a prevenção de uma parte significativa dos problemas de saúde mais comuns. O objetivo do projecto Life Bracelet é que se esteja um passo mais perto de atingir a grande finalidade da monitorização: um dispositivo que consiga medir e permitir a análise em tempo real de algumas das estatísticas do corpo humano mais importantes para a prevenção e diagnóstico.

Nesta Dissertação recolheu-se os fundamentos teóricos necessários ao desenvolvimento do circuito de sensorização da Life Bracelet. Pesquisou-se sobre métodos de obtenção de frequência cardíaca, oximetria e, por último, medição não-invasiva de glicose no sangue usando a fotopletismografia. Para os dois primeiros tópicos aborda-se uma metodologia de integração e modelação de um produto que seja eficaz e prático para o utilizador, enquanto que, para o terceiro tópico, por não estar bem explorado e investigado nos dias de hoje, tenta-se explorar um pouco mais dos métodos já conhecidos de forma a perceber qual destes será mais adequado a um futuro próximo. Depois de recolher o estado da arte das tecnologias actuais, numa fase inicial mostra-se a iteração anterior do produto e propõe-se uma nova solução para a sensorização do mesmo. De seguida mostra-se quais os materiais disponíveis ao projecto e decide-se quais as opções a tomar em termos de microprocessador e protocolos a usar, entre estes: o I2C e Bluetooth Low Power.

Depois de se explicar o método e desenho do circuito final explica-se o algoritmo que foi utilizado pelo microprocessador para calcular a frequência cardíaca e a saturação de oxigénio e envio dos dados por BLE. Por último fala-se de como foi desenvolvida a plataforma iOS que comunica com o sistema supramencionado e como são apresentados os dados ao utilizador final.

# Abstract

The quality of life has increased significantly due to technology and the great advances, not only of engineering but also of medicine. Monitoring is the key to preventing a significant portion of the most common health problems. The aim of the Life Bracelet project is to be one step closer to achieving the great purpose of monitoring: a device that can measure and allow real-time analysis of some of the most important human body statistics for prevention and diagnosis.

In this dissertation was collected the theoretical foundations necessary for the development of the sensor circuit of Life Bracelet. Research was done on methods for obtaining heart rate, oximetry and, finally, noninvasive measurement of blood glucose using photoplethysmography. The first two topics deal with a methodology for integrating and modeling a product that is effective and practical for the user, while for the third topic, because it is not well explored and investigated these days, it is tried to explore a little more of the methods already known in order to understand which of these will be more appropriate in the near future. After collecting the state-of-the-art of current technologies, an initial stage shows the previous iteration of the product and proposes a new solution for the sensor. The following is a list of the materials available to the project and the options to be taken in terms of microprocessor and protocols to be used, including I2C and Bluetooth Low Power.

After explaining the method and design of the final circuit was explained the algorithm that was used by the microprocessor to calculate the heart rate and oxygen saturation and tranfering the data by BLE. Finally, it was talked about how the iOS platform that communicates with the aforementioned system was developed and how the data is presented to the end user.

# Agradecimentos

Gostaria de agradecer em primeiro lugar aos meus pais, Celeste e Miguel , duas pessoas que independentemente dos obstáculos encontrados me mostraram que nada é insuperável com a motivação certa e que nenhum problema é tão grande como parece. Pela inspiração e valores humanos que me transmitiram que fizeram de mim o que sou hoje.

Ao meu irmão Pedro e primas Sara e Carolina pela amizade paciência e companhia a todas as horas. Aos meus tios e restante família pela preocupação e constante ajuda.

À Tuna de Engenharia da Universidade do Porto pela grande ajuda a encontrar-me a mim próprio, por me juntar a pessoas como eu, pela música, pela irmandade, pelos bons momentos e pelo porto de abrigo. Acima de tudo por me fazer reparar em coisas minhas que eu próprio desconhecia.

Aos padrinhos e afilhados pelo companheirismo, amizade e acima de tudo transmissão de conhecimento não só praxístico mas humano.

Um cumprimento especial para o Engenheiro João Paredes da Associação TECPorto pela paciência, conhecimento e por todas as horas fora-de-horas em que me ajudou.

Agradecer também a todas as pessoas da Faculdade de Engenharia pela excelência e auxílio durante todo o meu percurso académico.

Por último a todas as pessoas que passaram pelo meu caminho, pessoas que me marcaram cada uma à sua maneira e que fizeram de mim o que sou hoje. Um agradecimento muito especial também às pessoas que me deram força nas horas em que menos a tinha.

Tiago Simões

# Conteúdo


**1 Introdução** **1**
- 1.1 Enquadramento . . . 1
- 1.2 Motivação . . . 2
- 1.3 Objetivos . . . 2
- 1.4 Estrutura desta dissertação . . . 3

**2 Fundamentação Teórica** **5**
- 2.1 Sistema Cardiovascular . . . 5
  - 2.1.1 O Coração . . . 5
  - 2.1.2 Pulsação . . . 6
- 2.2 Método de obtenção do *ritmo cardíaco* . . . 7
  - 2.2.1 Eletrocardiografia (ECG) . . . 7
  - 2.2.2 Fotopletismografia (PPG) . . . 8
- 2.3 Oximetria . . . 10
  - 2.3.1 Hemoglobina . . . 10
  - 2.3.2 Fotopletismografia na Oximetria . . . 10
- 2.4 Medição de glicose . . . 12
  - 2.4.1 Glicose . . . 12
  - 2.4.2 Fotopletismografia na Medição de Glicose . . . 13
  - 2.4.3 Conformação de Calor Metabólico (MHC) . . . 13
- 2.5 Onde medir no corpo . . . 14

**3 Estado da Arte** **17**
- 3.1 Medição da frequência cardíaca e oximetria . . . 17
- 3.2 Medição da glicose no sangue . . . 20

**4 Materiais e Métodos** **23**
- 4.1 Monitor analógico de ritmo cardíaco . . . 23
- 4.2 Análise de requisitos . . . 25
- 4.3 Materiais e componentes disponíveis . . . 25
  - 4.3.1 Arduino Nano . . . 25
  - 4.3.2 ESP8266 . . . 26
  - 4.3.3 HC-05 . . . 27
  - 4.3.4 MAX30101 vs MAX30102 . . . 30
  - 4.3.5 ESP32 . . . 31
- 4.4 Protocolo I2C . . . 33
- 4.5 Bluetooth . . . 34
  - 4.5.1 BLE vs Bluetooth 4.0 . . . 34

4.5.2 Serviços GATT . . . . . 34
4.6 Software . . . . . 36

**5 Conceção e Desenvolvimento do Dispositivo** **37**
5.1 Sensor MAX30102 . . . . . 37
5.2 Desenho e Impressão do Circuito . . . . . 38
5.3 Arquitetura final do sistema . . . . . 41

**6 Desenvolvimento do Software** **43**
6.1 Programação . . . . . 43
6.2 Programa do microcontrolador . . . . . 44
6.3 Interface do utilizador . . . . . 46
6.3.1 Programação para iOS . . . . . 46
6.3.2 Testes de Conectividade . . . . . 46
6.3.3 Aplicação iOS . . . . . 47

**7 Resultados e Conclusão** **53**
7.1 Resultados dos testes . . . . . 53
7.2 Conclusão . . . . . 55
7.3 Trabalho Futuro . . . . . 56

**A Código do microcontrolador** **59**

**B Código aplicação** **63**

**Referências** **69**

# Lista de Figuras


2.1 Fluxo de sangue no coração . . . 6
2.2 Onda normal de um eletrocardiograma . . . 7
2.3 Fotopletismografia de reflexão . . . 9
2.4 Reflexão vs transmissão . . . 9
2.5 Sinal de PPG . . . 10
2.6 Espectro de absorção de $HbO_2$ e Hb. Adaptado de Martins (2010). . . . 11
2.7 3 sinais diferentes de um oxímetro, de acordo com a idade. Adaptado de Martins (2010) . . . 12
2.8 Influência da glicose na propagação da luz (a) com menor concentração de glicose e (b) com maior concentração de glicose no sangue. Adaptado de Santos et al. (2014) . . . 13
2.9 Esquema do método MHC. Retirado de Cho et al. (2004) . . . 14
2.10 Medição em diferentes sítios do do corpo. Verde: sítios de fácil leitura; Amarelo: médio; Vermelho: sítios de difícil leitura. A verde: Orelha e testa, Amarelo: pulso e braço e Vermelho: Gémeos e tornozelos. . . . 15

3.1 Métodos de medição Eletrocardiográfico vs Óptico . . . 17
3.2 Aplicação de *ritmo cardíaco* em *smartphone* . . . 18
3.3 Headphones com monitor de ritmo cardíaco da Jabra . . . 18
3.4 Sensor de tornozelo PPG . . . 19
3.5 Sensor de pé para recém nascidos . . . 19
3.6 Sensor de lactose e glicose a partir da transpiração. Retirado de Gao et al. (2016) 20
3.7 Sensor de MHC. Retirado de Cho et al. (2004) . . . 20
3.8 Representação do sensor da GlicoWise . . . 21

4.1 Circuito anterior de monitorização do ritmo cardíaco . . . 23
4.2 Arquitetura inicial . . . 24
4.3 Arduino Nano . . . 26
4.4 ESP8266 . . . 26
4.5 Módulo HC-05 . . . 27
4.6 Conexão entre Arduino Nano e HC-05 . . . 28
4.7 ESP32 DevKit v4 da ESPressif . . . 31
4.8 Esquema de protocolo I2C . . . 33
4.9 Hierarquia do GATT. Encontrada em Bluetooth-SIG. . . . 35
4.10 UUID dos serviços . . . 35

5.1 Max30102 . . . 37
5.2 Esquema de pins do MAX30102. Encotrado em MAX. . . . 38
5.3 Esquema da PCB do MAX30102 . . . 39

5.4 Versão final dos MAX30101 e 102 . . . . . 40
5.5 Arquitetura do Sistema . . . . . 41

6.1 Lógica geral de programação . . . . . 44
6.2 Lógica da configuração . . . . . 44
6.3 Lógica do loop . . . . . 45
6.4 Aplicação LightBlue . . . . . 47
6.5 Modelo da aplicação . . . . . 48
6.6 Ecrã inicial . . . . . 49
6.7 Medidor de ritmo cardíaco . . . . . 50
6.8 Ecrã de erro quando não deteta o dedo . . . . . 50
6.9 Informações sobre a aplicação . . . . . 51

7.1 Sanitas vs LifeBracelet . . . . . 54

B.1 Esquema do circuito do MAX30102 . . . . . 67
B.2 Esquema da PCB do MAX30102 . . . . . 68

# Lista de Tabelas


2.1 $HR_{rest}$ de cada faixa etária em bpm . . . . . 6

4.1 Características do Arduino Nano . . . . . 27
4.2 Comparação superficial entre MAX30101 e MAX30102 . . . . . 30
4.3 Componentes do DevKit . . . . . 32
4.4 Características do ESP32 . . . . . 32
4.5 Eficiência do BLE vs outros protocolos . . . . . 34

7.1 Comparação de resultados . . . . . 53

# Abreviaturas e Símbolos

| | |
|---|---|
| $\mu C$ | Microcontrolador |
| AC | Corrente Alternada |
| ACK | Acknowledgement |
| API | Application Programming Interface |
| BLE | Bluetooth Low Energy |
| BPM | Batimentos Por Minuto |
| DC | Corrente contínua |
| ECG | Eletrocardiograma |
| GATT | Generic Attribute Profile |
| Hb | Hemoglobina |
| $HR_{rest}$ | Resting Heart Rate |
| $HR_{max}$ | Maximum Heart Rate |
| HRV | Heart Rate Variability |
| I2C | Inter-Integrated Circuit |
| IoT | Internet Of Things |
| KPI | Key Performance Indicators |
| LED | Light Emiting Díode |
| MHC | Metabolic Heat Conformation |
| PCB | Printed Circuit Board |
| PPG | Fotopletismografia |
| SCL | Sinal de relógio (I2C) |
| SDA | Linda de dados (I2C) |
| SMD | Surface Mount Device |
| USB | Universal Serial Bus |
| WWW | *World Wide Web* |

# Capítulo 1

# Introdução

## 1.1 Enquadramento

O *Life Bracelet* é um projeto da Associação TecPorto que é uma organização dedicada ao Ensino, Formação, Pesquisa, Inovação e ao Desenvolvimento. Tem os objetivos de criar e ajudar no desenvolvimento de soluções inovadoras para problemas atuais em diversas áreas e apoiar o ensino e formação em áreas tecnológicas.

A TecPorto, no seu formato original, nasceu em Janeiro de 2018 e herdou os projetos, iniciativas e recursos do agora extinto *Instituto Onda Technology*. Alguns dos antigos membros do *Instituto Onda Technology* transitaram também para a TecPorto. A TecPorto foi fundada por uma equipa de pessoas com uma forte paixão por tecnologia que a estabeleceram na magnífica Cidade do Porto e pretende ser um marco positivo na qualidade e no reconhecimento do Porto como um polo de renome nas áreas de tecnologia.

A nível de investigação, esta instituição também funciona como think-tank tecnológico e científico e uma incubadora de projetos e ideias.

"O projeto *Life Bracelet* é a realização de uma ideia que surgiu de uma tragédia ocorrida em 2011, quando o irmão (engenheiro de 25 anos e com treino de natação de competição) de um dos fundadores da Associação TecPorto faleceu em circunstâncias ainda não totalmente justificadas. Na sequência deste evento, surgiu a ideia para a construção de um dispositivo que fosse capaz de monitorizar, em tempo real, os sinais vitais de pessoas em situações de risco como, por exemplo, utentes de recintos semelhantes, reportando esses dados a um sistema central que conseguisse notificar o pessoal de emergência de forma automática e, como homenagem, tentar ajudar a evitar que mais vidas fossem perdidas para eventos similares. A ideia original evoluiu para poder incluir versões que pudessem ser utilizadas em hospitais, lares e outras instituições de cuidados de saúde ou continuados e modelos auto-suficientes para indivíduos com problemas de saúde específicos. O projeto encontra-se em desenvolvimento desde 2011, tendo já produzido alguns protótipos para teste. Este projeto tem como objetivo continuar esse trabalho, estudando mais uma alternativa de sensorização. " TECPorto

## 1.2 Motivação

Vive-se uma era em que a informação está na ponta dos dedos, em que sempre que se abre o computador se pode observar um novo avanço tecnológico impressionante. Desde carros elétricos autónomos ou movidos a hidrogénio a *drones* e robôs inteligentes e computadores quânticos , já seria de esperar um avanço maior na área do bem-estar das pessoas. Também a área da Medicina já sofreu grandes avanços e cada vez aposta mais na prevenção. O projeto *Life Bracelet* tem em consideração colmatar algumas lacunas que ainda existem e oferecer uma monitorização mais constante do estado de saúde das pessoas, tornando-o uma experiência confortável e acima de tudo o menos intrusiva possível para que se torne um hábito constante das suas vidas.

## 1.3 Objetivos

Como objetivos tem-se a conceção, desenvolvimento, teste, validação e posterior construção de um dispositivo capaz de medir eficientemente a frequência cardíaca (*heart rate*) e a saturação de oxigénio. Estudar-se-á também alguns métodos de medição de glicose não-invasiva como objetivo extra.

Propostos os objetivos tratar-se-á de perceber o estado da arte presente dos três objetivos supramencionados e desenvolver um produto mais próximo de atingir os requisitos para integrar o mercado dos dispositivos *wearables*. Tendo isto em conta, esta dissertação tem como objetivo as seguintes metas:

- Estudar o enquadramento do tema no contexto atual;
- Estudar os métodos existentes para medição da frequência cardíaca e oximetria e decidir qual deles integrar;
- Elaboração ou escolha de um sensor para medir a frequência cardíaca oximetria com o método da fotopletismografia;
- Projeto de um circuito de dispositivo montado em superfície (SMD) e consequente desenvolvimento do circuito final em placa de circuito impresso (PCB);
- Elaboração de um algoritmo para processar o sinal proveniente do sensor;
- Desenvolvimento de software responsável pela comunicação com o módulo Bluetooth;
- Criação de um aplicativo *iOS* que deve funcionar como uma interface com o usuário;

Visto já ter havido também uma iteração anteriormente desenvolvida deste produto em 2016 ter-se-á também como meta:

- Melhorar eficiência energética do produto;
- Diminuir tamanho do circuito;

## 1.4 Estrutura desta dissertação

Uma introdução ao projeto é dada no Capítulo 1. No Capítulo 2, apresenta-se o sistema cardiovascular, que é necessário para entender o projeto, bem como os métodos de obtenção do ritmo cardíaco, oximetria e glicose, permitindo-nos entender o método mais eficiente para o projeto em questão. Posteriormente, o Capítulo 3 demonstra um estudo do estado da arte das tecnologias em questão. O Capítulo 4 apresenta os materiais e métodos já existentes utilizados para a idealização do produto final. O capítulo 5 apresenta a concretização da solução desenvolvida para a sensorização do produto e a sua arquitetura. O Capítulo 6 apresenta o algoritmo e a interface de utilizador desenvolvida e sua importância. Finalmente, o Capítulo 7 apresenta as conclusões extraídas ao longo do processo de desenvolvimento do projeto, bem como possíveis procedimentos para aprimora-lo no futuro.

# Capítulo 2

# Fundamentação Teórica

Neste capítulo apresenta-se os conceitos e as definições para a compreensão do sistema cardiovascular e tudo o que o mesmo implica no projeto.

## 2.1 Sistema Cardiovascular

O sistema circulatório, também chamado de sistema cardiovascular, é um sistema que permite que o sangue circule e transporte nutrientes dentro do corpo humano. Este sistema contribui de várias formas para a manutenção do equilíbrio do organismo, designado homeostase (dos termos gregos *homeo*, "similar" ou "igual", e *stasis*, "estático"), nomeadamente através das suas funções de:

- Transporte do oxigénio desde os pulmões para todo o organismo e do dióxido de carbono em sentido contrário para ser eliminado;
- Transporte de nutrientes absorvidos para os locais onde são necessários no organismo;
- Transporte de resíduos do metabolismo das células para os locais onde são eliminados;
- Transporte de substâncias produzidas em determinados locais do organismo (p. ex. hormonas, anticorpos) para outros locais onde são necessárias;
- Participação na regulação da temperatura corporal Gale Encyclopedia of Medicine (2008).

### 2.1.1 O Coração

O coração é um órgão de quatro câmaras, que recebe sangue venoso da circulação sistémica para a aurícula direita, passando-o para o ventrículo direito a ser bombeado pelos pulmões na circulação pulmonar; o sangue oxigenado nas veias pulmonares entra na aurícula esquerda e, daí, para o ventrículo esquerdo, para ser bombeado para fora na aorta para todas as artérias sistémicas. As válvulas controlam a direção do fluxo. O miocárdio (músculo cardíaco) forma as paredes das câmaras que são revestidas pelo endocárdio. O coração está contido no pericárdio membranoso Livingstone (2008).

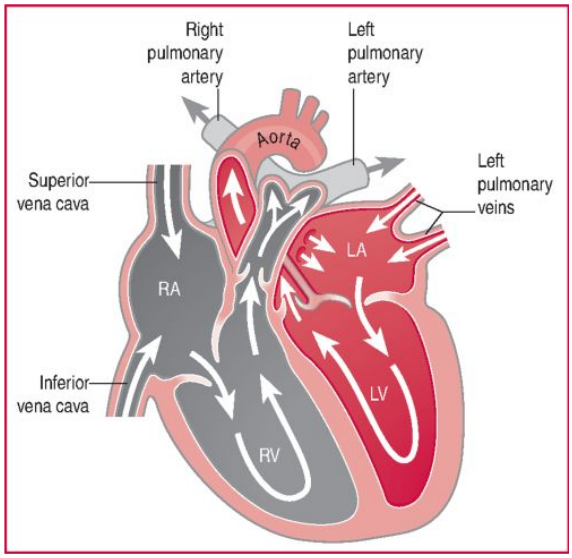



Figura 2.1: Fluxo de sangue no coração

### 2.1.2 Pulsação

A pulsação advém de uma contração e expansão das artérias que acontecem devido à diferença de pressão sanguínea em cada ciclo do coração. Durante a sístole, o coração encontra-se contraído e expele o sangue arterial carregado de oxigénio e nutrientes. Já durante a diástole, o coração expande e enche-se com sangue venoso que será expelido para os órgãos (ilustrado na figura 2.1).

Pode-se, então, inferir que o ritmo cardíaco (*ritmo cardíaco*) é uma contagem de quantas destas compressões/expansões existem num minuto. Este é, por isso, medido em Batimentos Por Minuto (bpm).

Existem algumas métricas para medição da pulsação e saber se ela é adequada à pessoa tendo em conta a idade e o exercício físico. A frequência cardíaca de repouso ($HR_{rest}$) é definida como a frequência cardíaca quando uma pessoa está acordada, num ambiente temperado e não foi sujeita a nenhum esforço ou estímulo recente, como stress ou surpresa. Uma grande base de provas indica que o intervalo normal é de 60 a 100 batimentos por minuto. American College of Cardiology (1995)

| recém-nascidos (0–3 meses) | bebés (3–6 meses) | bebés (6–12 meses) | crianças (1–10 anos) | > 10 anos até seniores | atletas de alta competição |
|---|---|---|---|---|---|
| 99-149 | 89–119 | 79-119 | 69–129 | 59–99 | 39–59 |

Tabela 2.1: $HR_{rest}$ de cada faixa etária em bpm

Outra dessas métricas é a pulsação máxima ($HR_{max}$) e existem várias fórmulas para a calcular:

$$Haskell\&Fox: \quad HR_{max} = 220 - idade \tag{2.1}$$

$$OaklandUniversityStudy: \quad HR_{max} = 192 - 0.007 * idade^2 \tag{2.2}$$

A fórmula 2.1 é reconhecidamente bastante variável, estimando-se ter um desvio padrão de 10 a 12 bpm. (Gellish et al., 2007, p.01) Contudo, continua a ser a "Regra Prática" em muitos estudos e aproximações no que toca a esta métrica. Por essa mesma razão foi efetuado um novo estudo pela Universidade de Oakland, onde se chegou à fórmula 2.2. Esta já se aproximou com uma equação quadrática, ao contrário da anterior que é linear, e com um erro máximo de $\pm$ 2 até 5 bpm. Estas fórmulas serão importantes para avaliar e comparar resultados posteriores.

## 2.2 Método de obtenção do *ritmo cardíaco*

### 2.2.1 Eletrocardiografia (ECG)

A eletrocardiografia é um procedimento não invasivo e comummente usado para registar alterações elétricas no coração. O registo, que é chamado de eletrocardiograma (ECG ou EKG), mostra uma série de ondas relacionadas com os impulsos elétricos que ocorrem durante cada batida do coração. Os resultados são impressos em papel ou exibidos num monitor. As ondas num registo normal são denominadas P, Q, R, S e T e seguem em ordem alfabética. O número de ondas pode variar e outras ondas podem estar presentes Gale Encyclopedia of Medicine (2008).

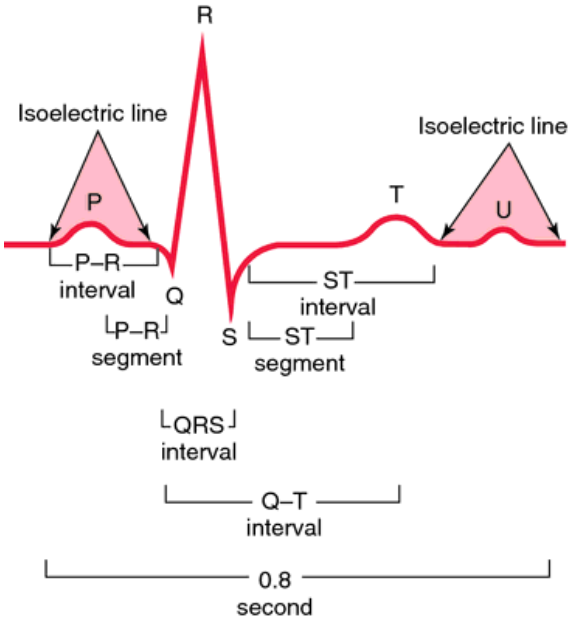



Figura 2.2: Onda normal de um eletrocardiograma

1. Ação do coração durante o intervalo P-R:
   (a) A contração auricular começa no pico da onda P.
   (b) Intervalo P-R - contração auricular.

(c) Ventrículos relaxados.

2. Ação do coração durante o complexo QRS:

   (a) contração ventricular começa no pico da R.

   (b) Válvulas A-V (mitral e tricúspide) fecham-se, causado som S1.

   (c) Contração de ventrículos.

   (d) O relaxamento auricular começa.

3. Acção do coração durante o segmento S-T:

   (a) Válvulas semilunares abertas (aórtica e pulmonar).

   (b) Ejeção de sangue de ventrículos - sístole.

4. Ação do coração durante a onda T:

   (a) Atraso da ejeção nos ventrículos.

   (b) Fecho de válvulas semilunares (aórtica e pulmonar) causando som S2.

5. Ação do coração durante o intervalo T-P:

   (a) Relaxamento dos ventrículos.

   (b) Válvulas A-V abertas.

   (c) Enchimento de ventrículos, causando som S3.

### 2.2.2 Fotopletismografia (PPG)

A frequência cardíaca pode ser estimada tendo por base o principio de fotopletismografia, ou seja, a medição de alterações do volume sanguíneo utilizando um método ótico. Existem duas técnicas de medição com este método:

- Na fotopletismografia de reflexão, uma fonte de luz e um detetor de luz são colocados no mesmo lado de uma parte do corpo, no dedo por exemplo Martins (2010). A fonte de luz geralmente usada é um díodo emissor de luz infravermelho. O detetor geralmente usado é um fototransístor. Quando a ponta do dedo é iluminada pela fonte, três coisas acontecem com uma intensidade que depende do volume de sangue na ponta do dedo: uma certa quantidade de luz será absorvida, outra quantidade vai ser transmitida e certa quantidade de luz será refletida. A intensidade da luz refletida varia com o volume de sangue na ponta do dedo que, por sua vez, varia de acordo com o batimento cardíaco. Especificamente, menor intensidade de luz refletida indica maior volume de sangue e vice-versa Saquib et al. (2015).

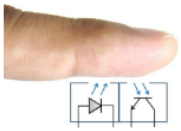


Figura 2.3: Fotopletismografia de reflexão

- Na fotopletismografia de transmissão os LEDs transmitem luz de comprimentos de onda específicos através do tecido, que é absorvido por fotodetetores no outro lado. A mudança na absorção de cada comprimento de onda determina os níveis de saturação de oxigénio. Esta técnica requer o uso de um local de teste fino com perfusão sanguínea adequada, como um dedo ou um lóbulo de forma a maximizar a transmissão de luz.

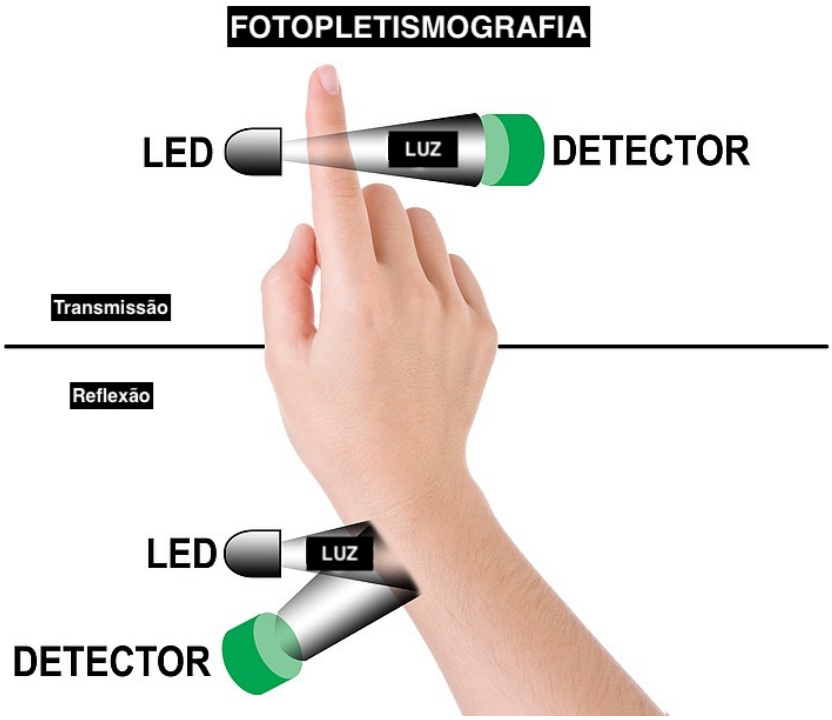



Figura 2.4: Reflexão vs transmissão

A forma de onda PPG consiste em dois componentes, contínua (DC) e alternada (AC), Figura 2.5, mas como confirma o estudo exposto no *Journal of Biomedical Optics*, é muito difícil encontrar uma relação direta entre os componentes da fotopletismografia Lee et al. (2011). Contudo pensa-se que componente DC é atribuído à absorção de luz do tecido não pulsátil: capilar venoso e sangue arterial. O componente AC, por outro lado, é devido à natureza pulsátil do sangue arterial Michael W. Wukitsch e Jonas A. Pologe (1992). Como as artérias têm uma conexão direta com o coração, o sangue arterial pulsa quando o coração pulsa. A frequência cardíaca instantânea pode ser calculada medindo o tempo entre os picos sistólicos (pico observado durante cada batimento

do coração) consecutivos. É importante notar aqui que a frequência cardíaca pode ser medida usando apenas um LED, vermelho ou infravermelho, por exemplo, já que o componente AC é o único sinal necessário.

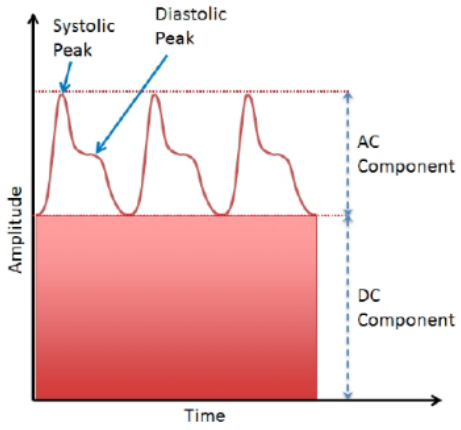



Figura 2.5: Sinal de PPG

## 2.3 Oximetria

### 2.3.1 Hemoglobina

A hemoglobina é uma proteína encontrada nos eritrócitos (glóbulos vermelhos) que transporta oxigénio molecular ($O_2$) no sangue e é representada pelo símbolo ($Hb$). A hemoglobina oxigenada (oxihemoglobina) tem um tom vermelho vivo; hemoglobina não ligada ao oxigénio (desoxihemoglobina) é mais escura. Isso explica a cor vermelha mais viva do sangue arterial, em que a hemoglobina é cerca de 97% saturada de oxigénio. O sangue venoso é mais escuro porque tem cerca de 20 a 70% de saturação, dependendo de quanto oxigénio está a ser usado pelos tecidos. Mosby (2012)

### 2.3.2 Fotopletismografia na Oximetria

Com base neste facto, Takuo Aoyagi descobriu, no inicio da década de 70, que é possível estimar o nível de saturação de oxigénio no sangue através da relação entre a diferença de luz infravermelha e vermelha absorvida. Este princípio é utilizado, correntemente, nos oxímetros comerciais Haahr (2006).

A figura 2.6 descreve as diferentes absorções da parte da oxihemoglobina ($HbO_2$) e desoxihemoglobina ($Hb$) para os diferentes comprimentos de onda supramencionados.

É possível verificar a existência de alguns pontos onde os espectros se intersectam - pontos isobésticos (pontos onde a absorção de oxihemoglobina e desoxihemoglobina é o mesmo). Pela

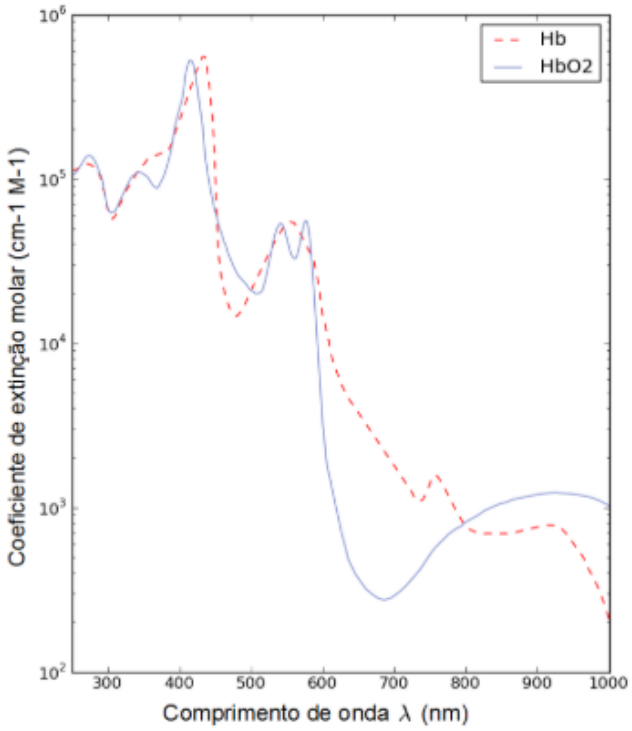



Figura 2.6: Espectro de absorção de $HbO_2$ e Hb. Adaptado de Martins (2010).

análise do intervalo de comprimentos de onda onde esta mesma absorção é predominante, é possível encontrar um ponto isobéstico a aproximadamente 805nm em que a oxihemoglobina apresenta uma absorção maior para comprimentos de onda na região do infravermelho. Por outro lado a desoxihemoglobina apresenta uma absorção maior para comprimentos de onda no espectro da região visível. [1]

A lei de Beer descreve os elementos que contribuem para a forma de onda do oxímetro:

$$A_{total} = E_1C_1L_1 + E_2C_2L_2 ... E_nC_nL_n \tag{2.3}$$

Onde: $A_{total}$ = Absorção total num dado comprimento de onda
$E_N$ = Coeficiente de extinção (absorvência)
$C_n$ = Concentração
$L_n$ = Distância

[1] Nota: Comprimentos de onda: Vermelho - 620–750 nm | Infravermelho - 700 nm a 1 mm

Na figura 2.7 pode-se observar, de acordo com a fórmula demonstrada, como mudam as formas de onda de oximetria tendo a idade como variável.

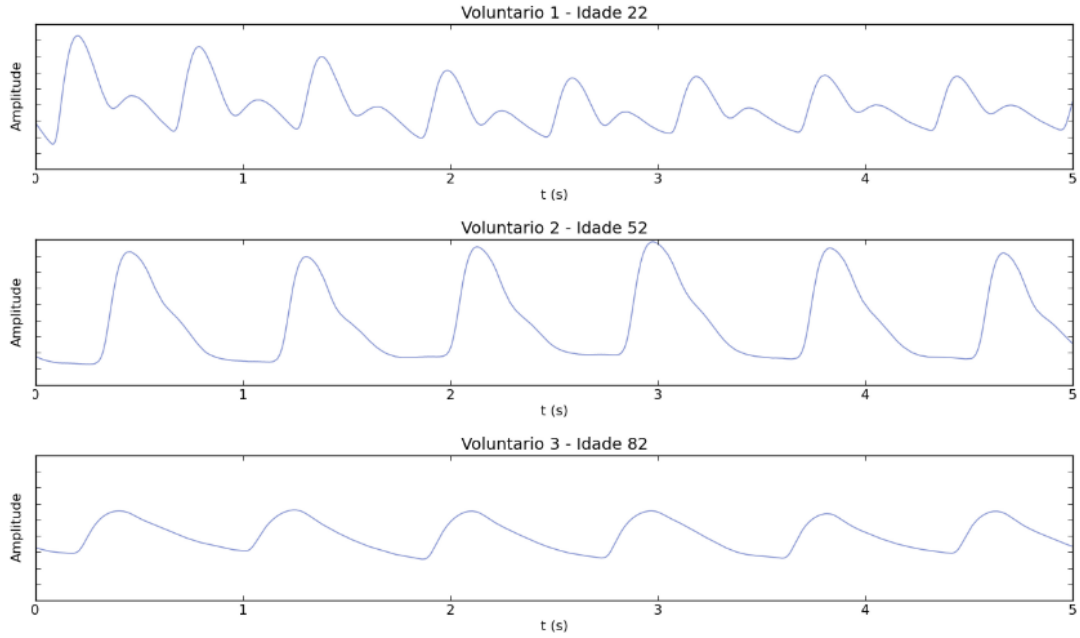



Figura 2.7: 3 sinais diferentes de um oxímetro, de acordo com a idade. Adaptado de Martins (2010)

Tem-se então todas as ferramentas para inferir como exprimir o nível de saturação do oxigénio no sangue:

$$SO_2 = \frac{C_{HbO_2}}{C_{HbO_2} + C_{Hb}} \tag{2.4}$$

Onde: $SO_2$ = Saturação do oxigénio no sangue
$C_{HbO_2}$ = Concentração de oxihemoglobina
$C_{Hb}$ = Concentração de desoxihemoglobina

## 2.4 Medição de glicose

### 2.4.1 Glicose

A glicose (ou dextrose) é o principal açúcar que o corpo produz a partir dos alimentos ingeridos. A glicose é transportada através da corrente sanguínea para fornecer energia a todas as células do corpo que, por sua vez, não podem usar a glicose sem a ajuda de insulina. É um açúcar simples (um monossacarídeo) que o corpo produz a partir de proteínas, gorduras e, em grande

parte, hidratos de carbono metabolizados (como, por exemplo, o amido pela amílase presente na saliva). A glicose ingerida é absorvida diretamente no sangue pelo intestino e resulta num rápido aumento da glicose no sangue.

### 2.4.2 Fotopletismografia na Medição de Glicose

A Fotopletismografia na medição da glicose tem como base o método utilizado pela oximetria de pulso para determinação da saturação de oxigénio no sangue arterial. Neste método, a medição não invasiva de glicose pode ser executada focando-se num feixe de luz no dedo e analisando a luz transmitida através do mesmo. A concentração de glicose pode ser determinada analisando a mudança no comprimento de onda, utilizando-se técnicas de espectroscopia, ou intensidade de luz transmitida em diferentes comprimentos de onda, geralmente de 700 a 2500 nm. Este método baseia-se nas propriedades de dispersão da luz causada pela glicose presente no sangue. O aumento da glicose diminui o desalinhamento do feixe que atravessa o tecido, pois o índice de refração é diminuído pela sua presença, proporcionando assim um caminho melhor para a passagem da luz, conforme a Figura 2.8. Como resultado, menos luz é absorvida e a intensidade da luz que atravessa o tecido é maior Santos et al. (2014).

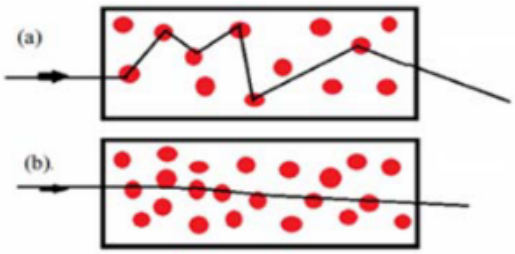



Figura 2.8: Influência da glicose na propagação da luz (a) com menor concentração de glicose e (b) com maior concentração de glicose no sangue. Adaptado de Santos et al. (2014)

### 2.4.3 Conformação de Calor Metabólico (MHC)

O método MHC foi derivado da observação que o ritmo circadiano homeostático do corpo humano depende da inter-relação entre calor metabólico, produção local de oxigénio e concentração de glicose, sendo baseado na seguinte equação conceptual:

$$[GLU] = F(calorgerado, taxadefluxosanguneo, Hb, HbO_2) \quad (2.5)$$

onde [GLU] representa a concentração de glicose, e F é uma função de um conjunto de variáveis metabólicas interdependentes e correlacionadas, que podem ser medidas na ponta do dedo de uma pessoa. As variáveis deste modelo consistem em geração de calor, concentração de hemoglobina ($Hb$), concentração de oxihemoglobina ($HbO_2$) e taxa de fluxo sanguíneo. Todas as variáveis

devem ser estimadas após a compensação pelas condições ambientais apropriadas. O dispositivo MHC usa sensores térmicos e ópticos para medir esses valores simultaneamente num indivíduo e, por meio de manipulações estatísticas, quantifica-se a concentração de glicose no sangue. Os vários sensores utilizados devem permitir a correção dos efeitos de diferentes interferências endógenas.Cho et al. (2004)

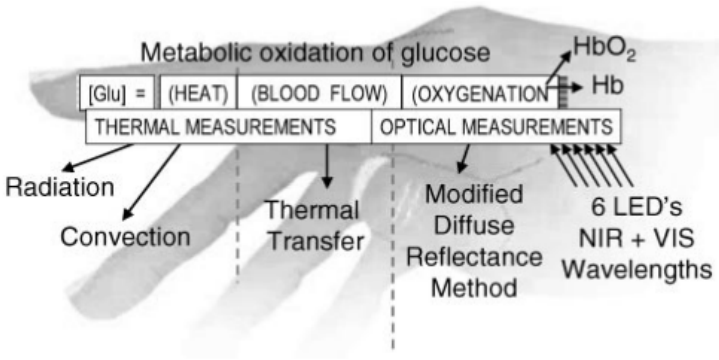



Figura 2.9: Esquema do método MHC. Retirado de Cho et al. (2004)

## 2.5 Onde medir no corpo

Depois de se abordar a variedade de todos os métodos mencionados anteriormente, é preciso salientar que estes têm margens de erro dependentes de vários fatores e disposições. Dependendo de onde são efetuados no corpo humano, pode obter-se resultados diferentes devido não só a função e composição das células mas também espessura do tecido ou mesmo dos órgãos, tendões ou músculos circundantes, como mostrado na figura 2.10.

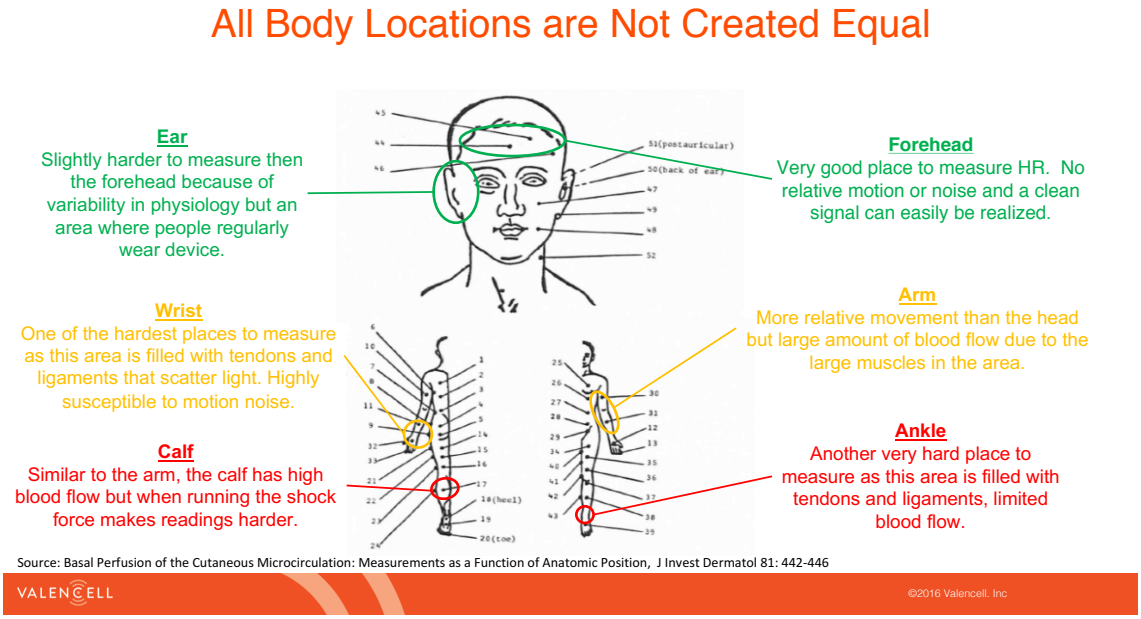



Figura 2.10: Medição em diferentes sítios do do corpo. Verde: sítios de fácil leitura; Amarelo: médio; Vermelho: sítios de difícil leitura. A verde: Orelha e testa, Amarelo: pulso e braço e Vermelho: Gémeos e tornozelos.

# Capítulo 3

# Estado da Arte

Neste capítulo apresenta-se o Estado da Arte dos métodos de medição mencionados no Capítulo 2. Serão apresentadas as tecnologias atualmente em uso, bem como a sua fiabilidade e, por último, as que se encontram ainda sobre investigação.

## 3.1 Medição da frequência cardíaca e oximetria

A grande explosão de desportos ao ar livre levou a uma enorme procura de dispositivos de medição de frequência cardíaca e oximetria. Isto veio criar uma larga oferta de produtos de custo reduzido para o uso no desporto.

A grande competição neste momento será entre as chamadas bandas torácicas observadas em 3.1a para colocar no peito e os medidores de pulso 3.1b. O primeiro usa a tecnologia discutida em 2.2.1(Eletrocardiografia) e o segundo em 2.2.2 (Fotopletismografia). Cada um destes métodos tem vantagens e desvantagens. O primeiro tem a vantagem de ser um método mais preciso, enquanto o segundo tem a seu favor a comodidade por ser menos invasivo. É também preciso ter em consideração que, para o consumidor final, a parte estética tem um grande peso na compra do produto, algo que se deve ter em conta quando se desenvolve o mesmo. Por este motivo também se nota uma predominância tão acentuada dos medidores de pulso.




(a) Banda de Peito (b) Medidor de pulso

Figura 3.1: Métodos de medição Eletrocardiográfico vs Óptico

Também é possível obter, embora sem grande precisão, a frequência cardíaca usando uma simples aplicação num *smartphone*. Tudo isto recorrendo apenas ao auxílio de tecnologias bastante comuns já presentes nos dispositivos de hoje em dia, neste caso a câmara (receptor) e o *flash* (emissor). Pode ter várias vantagens em relação à facilidade e confortabilidade de medição, porém, como referido anteriormente, não deve ser considerada uma técnica fiável para fins médicos por estar associada a uma incerteza muito grande devido à forte influência da luz ambiente.

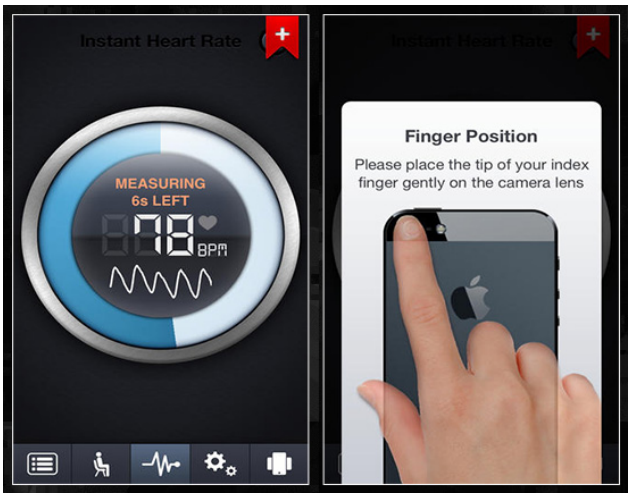



Figura 3.2: Aplicação de *ritmo cardíaco* em *smartphone*

Como falado no capítulo anterior é já possível medir estes sinais vitais nos mais diversos sítios do corpo humano. Os produtos seguintes mostram algumas das mais diversas formas de o fazer.



Figura 3.3: Headphones com monitor de ritmo cardíaco da Jabra

Os auriculares da figura 3.3 medem vários tipos de sinais vitais usando a fotopletismografia no lóbulo auricular.

Após uma pesquisa aprofundada de alguns fóruns ligados a este tipo de produtos pôde-se concluir que grande parte dos utilizadores admite ter uma má experiência com os mesmos devido à sua

imprecisão. Assim sendo essas empresas recorrem a algoritmos de compensação de movimento para tirar partido de acelerómetros parra minimizar o erro das medições que são apresentadas aos utilizadores.

Tem-se o exemplo de algumas tecnologias que ainda se encontram em investigação. Estas baseiam-se em novos métodos para estimar a frequência cardíaca média a partir de um sensor de fotopletismografia (PPG) usado no pé / tornozelo durante a atividade rápida de ciclismo. Colocando o sensor PPG na metade inferior do corpo permite que mais energia seja armazenada a partir de métodos de colheita de energia, a fim de fornecer um sensor autónomo, mas que vem com o custo de introduzir interferência de movimento significativa no traço PPG Jarchi e Casson (2016).



Figura 3.4: Sensor de tornozelo PPG

No caso de o sensor ser aplicado numa parte do corpo que não seja suscetível a rotações ou aceleração causadas por movimento já se pode encontrar uma vasta gama de produtos como ilustra a figura 3.5, um monitor para recém-nascidos.




Figura 3.5: Sensor de pé para recém nascidos

## 3.2 Medição da glicose no sangue

Na área da glicose existem outras tecnologias que estão em investigação. No exemplo da figura 3.7 tem-se um conjunto de sensores amperométricos de glicose e lactato (com saída de corrente) que são baseados em glicose oxidase e lactato oxidase imobilizada dentro de um filme permeável de um polissacarídeo linear de quitosana. Este sensor é baseado na transpiração do utilizador. Recolhe-a e, de acordo com a reação química, retorna um valor de corrente diferente.

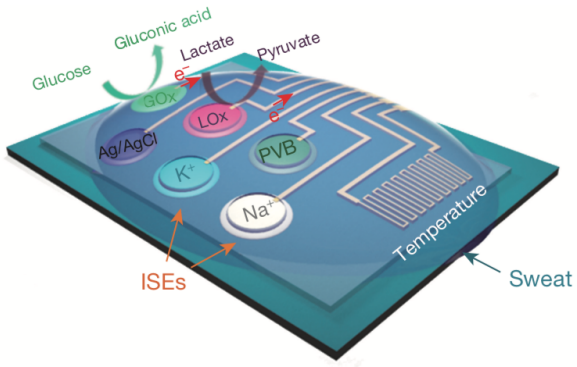



Figura 3.6: Sensor de lactose e glicose a partir da transpiração. Retirado de Gao et al. (2016)

O sensor da figura 3.7 é baseado no método endereçado na subsecção 2.4.3 (MHC).



Figura 3.7: Sensor de MHC. Retirado de Cho et al. (2004)

Muitas empresas estão a apostar no mercado da medição não invasivas da glicose. Segundo a *International Diabetes Federation* aproximadamente 425 milhões de pessoas no Mundo padeciam de diabetes em 2017 e prevê-se um aumento de quase 48% até 2045. Depois de alguma pesquisa, é notório que é um mercado interessante para as grandes empresas tendo em conta a sua prevalência, proporcionando uma vasta pesquisa atrás de métodos não invasivos.

Como exemplo, uma dessas empresas é a *GlucoWise*, que afirma ter um protótipo pronto a usar em 2021. Este produto utiliza um método comparável à fotopletismografia apresentada na

secção 2.3.2. Contrariamente a esta, onde é mais comum trabalhar em frequências mais próximas do visível ou dos infravermelhos, o método deste novo produto inova por usar antes radiação eletromagnética na gama das alta-frequências (HF), a rondar os 65 GHz.




Figura 3.8: Representação do sensor da GlicoWise

# Capítulo 4

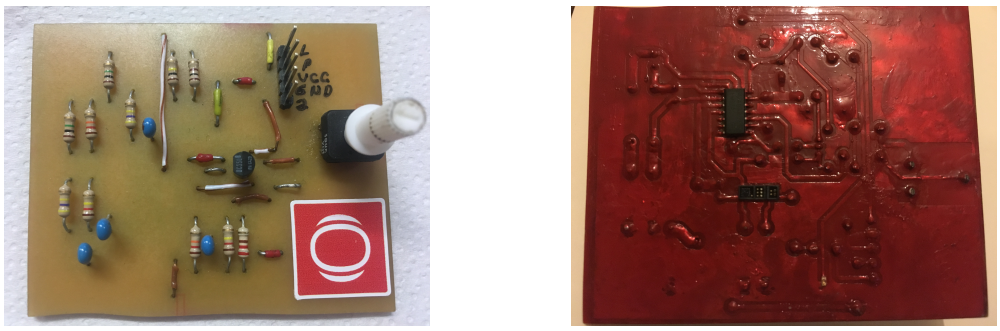
# Materiais e Métodos

Neste capítulo contextualiza-se o estado anterior do projeto até ao momento. Começa-se por demonstrar e explicar a iteração anterior do circuito, iniciada em 2016, que é capaz de provar o conceito de monitorizar sinais vitais como o ritmo cardíaco. Seguidamente explora-se e explica-se os materiais e métodos disponíveis para a realização do projeto atual e, por último, o porquê da escolha dos mesmos.

## 4.1 Monitor analógico de ritmo cardíaco

O circuito seguinte foi desenvolvido no início do projeto LifeBracelet com objetivo de se obter uma leitura analógica do ritmo cardíaco.

O mesmo é composto por 3 partes importantes: o Díodo Emissor de Luz (LED), o Sensor SFH7060 e o circuito condicionador de sinal. O sensor SFH7060 é um sensor desenvolvido pela OSRAM especialmente para aplicações que envolvem a medição de pulsação e oximetria através de fotopletismografia. É um sensor destinado a reflexão, como pode ser verificado pela disposição dos elementos que o constituem. Este inclui três LEDs verdes, um vermelho e um infravermelho.



(a) Frente da PCB (b) Parte traseira da PCB

Figura 4.1: Circuito anterior de monitorização do ritmo cardíaco

O sinal produzido pelo SFH7060 era tratado através de um circuito de condicionamento de sinal. Depois de tratado, o sinaĺ era lido pelo microcontrolador, que era responsável por interpreta-lo, processa-lo e por possibilitar a comunicação com dispositivos externos, com a sua programação feita em linguagem C. O utilizador podia interagir com o sistema através de um dispositivo *Android*. Para comunicar com esta aplicação, utilizou-se um módulo *Bluetooth*, o HC-06, que estaria ligado a um *Arduino Uno*, ou então diretamente com um computador pessoal através da porta de série virtual por USB, sendo os dados apresentados em bruto ou numa aplicação desenvolvida para o efeito.

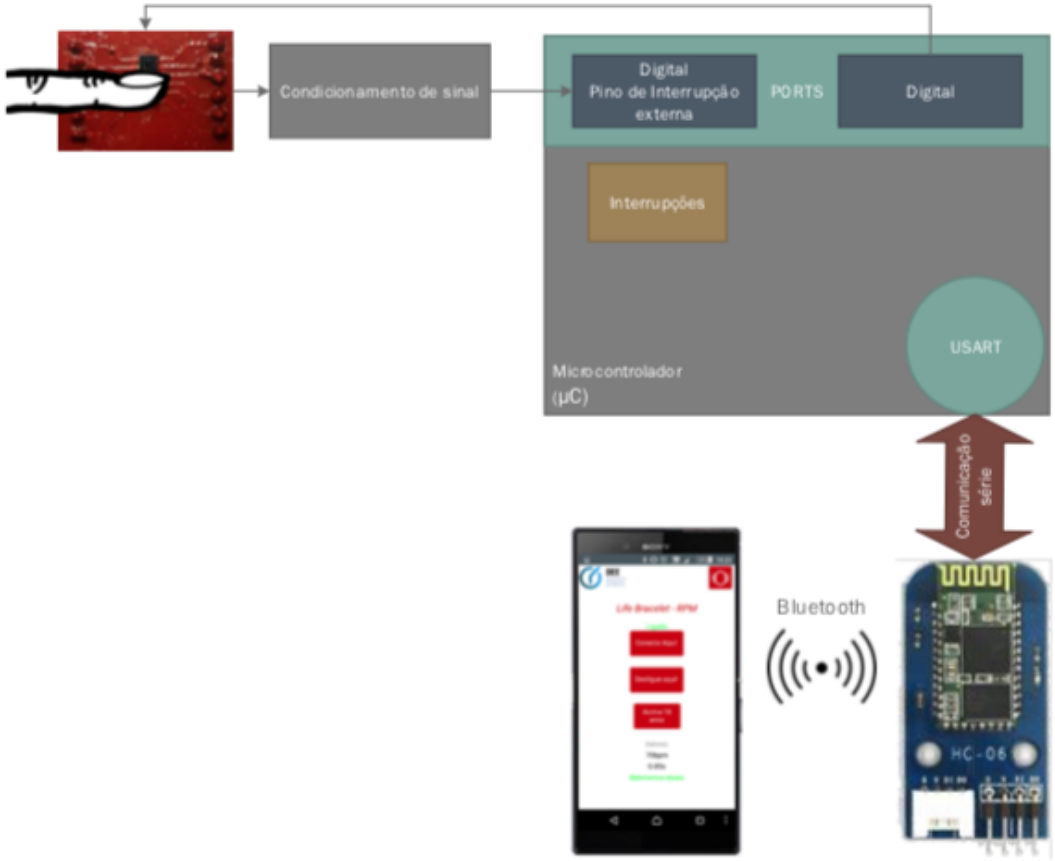



Figura 4.2: Arquitetura inicial

## 4.2 Análise de requisitos

Antes de se proceder à escolha do sensor, microcontrolador ou qualquer outro componente, deverá ter-se em conta o estado do projeto anterior bem como todos os requisitos no início projeto. Como referido anteriormente, esta pulseira não será usada para fins desportivos mas sim em contexto de saúde. O objetivo é não ser uma *fitbit* e poder ser usada em contexto hospitalar e em contexto desportivo, não na área *fitness* mas sim para deteção de situações anómalas, funcionando com monitorização constante. Terá então que ter uma capacidade de processamento maior e de enviar valores para um servidor ou aplicação de forma constante e eficiente a nível energético para que seja possível manter uma monitorização constante de um paciente por exemplo que tenha uma anomalia nos seus sinais vitais tal como uma fibrilação auricular. Os requisitos energéticos devem também ser cumpridos para que a pulseira necessite do mínimo possível de sessões de manutenção (carregamento de bateria ou substituição de pilha), reduzindo o tempo em que o utilizador se encontra sem monitorização.

Resumidamente há que ter em conta:

- Tamanho do sensor ;
- Eficiência energética;
- Rapidez de transferência de dados;
- Fácil integração num sistema maior.

## 4.3 Materiais e componentes disponíveis

### 4.3.1 Arduino Nano

O Arduino Nano é um microcontrolador onde vai ser processada toda a informação que chega do sensor escolhido. Além de ser *Breadboard friendly* e ter um chip *ATMEGA328*, extremamente compatível com quase todas as plataformas este microprocessador vem num tamanho extremamente compacto, algo que pode vir a ser muito benéfico para o produto final. Por outro lado este também possui menos memória que os seus pares como o Arduino Uno. Contudo para o processamento dos dados inerentes ao projeto pensa-se que será o suficiente. Na figura 4.3 pode observar-se a PCB em que vem integrado, com ligação micro USB para fácil conexão com o computador. Na tabela 4.1 também é possível ver as suas características.



Figura 4.3: Arduino Nano

### 4.3.2 ESP8266

O ESP8266 é um microcontrolador da *Espressif* que inclui capacidade de comunicação por *Wi-Fi*. Este módulo pode ser utilizado como microcontrolador ou apenas interface de comunicação pelo Arduino e uma das suas grandes vantagens é o seu preço que ronda os três a seis Euros. A comunicação do módulo com o Arduino pode ser feita via porta-série utilizando os pinos *RX* e *TX*, podendo ser configurada através de comandos *AT*.

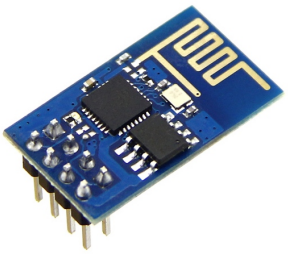


Figura 4.4: ESP8266

Após a conclusão da configuração da porta, reparou-se que a comunicação entre o Arduino e o controlador *Wi-Fi* era inexistente. O envio do comando "AT"não produzia qualquer resposta quando se deveria receber a mensagem "OK"(que sinalizaria que a placa estava a funcionar corretamente). Depois de alguma pesquisa chegou-se à conclusão que poderia ser um problema do *firmware* da placa que poderia estar desatualizado. Mudou-se então alguns pinos da placa para que esta entrasse em modo *flash* e tentou-se atualizar o *firmware* não se tendo obtido qualquer resposta. Como tal, e na possibilidade de esta estar inoperacional optou-se por adotar outra solução que será referida no capítulo 4.3.5 . Como o objetivo do projeto era também a eficiência energética ponderou-se fazer a transmissão por Bluetooth usando o módulo HC-05.

| Arduíno Nano | |
|---|---|
| Arquitetura | AVR |
| Tensão de operação | 5 V |
| Memória Flash | 32 KB menos 2 para o bootloader |
| SRAM | 2 KB |
| Velocidade de relógio | 16 MHz |
| Pins analógicos | 8 |
| EEPROM | 1 KB |
| Corrente DC por pin | 40 mA (I/O Pins) |
| Tensão de entrada | 7-12 V |
| Pins I/O digitais | 22 (6 PWM) |
| PWM Output | 6 |
| Consumo de energia | 19 mA |
| Tamanho da PCB | 18 x 45 mm |
| Peso | 7g |

Tabela 4.1: Características do Arduino Nano

### 4.3.3 HC-05

O módulo HC-05 permite a comunicação sem-fios entre o *Arduino* e o dispositivo móvel através de *Bluetooth*, um protocolo de comunicação por radio-frequência na banda ISM que opera a uma frequência de 2.4Ghz e se encontra otimizado para a comunicação a curta de distância, normalmente entre 5 a 10 metros.

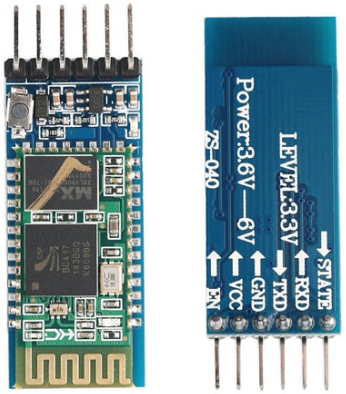



Figura 4.5: Módulo HC-05

Este módulo do tipo *slave* ou *master* pode tanto aguardar por uma comunicação com outro dispositivo ou mesmo inicia-la. Após estabelecerem a conexão, para trocar dados entre ambos, é necessário que o dispositivo *master* suporte a versão igual ou superior à do *slave*. Este módulo é de fácil configuração, possuindo possui apenas 6 pinos:

- EN - Pino *enable key*

- VCC - Pino de alimentação a 3,3V
- RX - Pino de recepção de dados
- TX - Pino de emissão de dados
- GND - Ground
- STATE - Pino de estado

O pino *Enable key* é usado para alternar entre o modo de Transferência de Dados e o modo Comandos AT que é usado para configurar a placa, não tendo sido usado já que, para o efeito, pode-se apenas sinalizar a placa com o botão premido que automaticamente entrará em modo AT. O pino de alimentação pode ser conectado diretamente à fonte de alimentação 5V da placa do Nano, o pino de receção de dados ao pino de transmissão de dados do microcontrolador, o pino de envio de dados ao pino de receção de dados, e finalmente, o pino GND (ground) definido no sistema ou um terreno disponível na placa do microcontrolador. Por último também é possível usar o pino State para se ligar, por exemplo, a um LED para se ter um *feedback* da comunicação. Este também não foi usado.

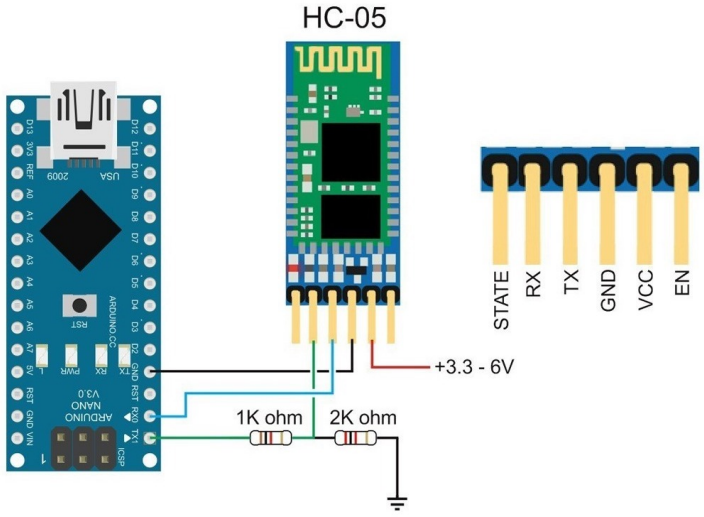



Figura 4.6: Conexão entre Arduino Nano e HC-05

Com os comandos AT foi possível obter o estado da placa entre outras informações necessárias à comunicação. Depois de várias tentativas de comunicação com um *iPhone*, descobriu-se que o HC-05 simplesmente não funciona com o sistema operativo móvel iOS, porque este sistema suporta apenas alguns perfis Bluetooth que se deve às limitações do programa de licenciamento escolhido pelo fabricante do telemóvel. Numa tentativa de contornar a situação, e após uma pesquisa intensa, determinou-se que, embora alguns dispositivos Bluetooth não sejam suportados iOS, todos os dispositivos BLE (Bluetooth Low Energy) o são. Como base neste facto, investigou-se,

para efeitos de prototipagem, uma alternativa que poderá satisfazer todos os requisitos do sistema e que será abordada no capítulo seguinte.

### 4.3.4 MAX30101 vs MAX30102

Os sensores MAX30102 e MAX3101 são circuitos integrados da *Maxim Integrated* que facilitam a medição de oximetria de pulso e monitor de frequência cardíaca pelo método refletivo. Incluem LEDs internos, fotodíodos, elementos óticos e eletrónica com baixa introdução de ruído e, por último, rejeição de luz ambiente que, por sua vez, elimina uma boa parte dos erros de medições. Este sensor é uma solução de que facilita consideravelmente o processo de *design* para dispositivos portáteis e, por esse motivo, foi o selecionado de entre todas as opções disponíveis.

| Dispositivo | LEDs | Modos |
|---|---|---|
| MAX30101 | Vermelho e Infravermelho e Verde | HR e SPO2 |
| MAX30102 | Vermelho, Infravermelho | HR e SPO2 |

Tabela 4.2: Comparação superficial entre MAX30101 e MAX30102

Como ambos os sensores foram enviados como amostras de conceito para a Associação TecPorto pela *Maxim Integrated*, podem ser considerados como apropriados para desenvolvimento e teste de protótipos de produto. Por outro lado, é possível que não correspondam ao produto final e sejam suscetíveis a problemas que ainda não foram detetados ou corrigidos. Foi o caso do MAX30101 que, em seis amostras, três não aguentaram o calor da soldadura SMD e partiram imediatamente, apesar de terem sido sujeitas à temperatura correta conforme a *datasheet*. Nos outros três, com a ajuda do algoritmo que será mencionado no capítulo 6, foi possível saber que dois deles nem sequer eram reconhecidos como um sensor e o outro apenas tinha o LED verde funcional. Por outro lado não foi o caso das amostras do MAX30102, que se mostraram perfeitamente funcionais e, assim sendo, foram as utilizadas no protótipo deste projeto.

### 4.3.5 ESP32

O ESP32 é um chip único que combina Wi-Fi e Bluetooth de 2,4 GHz projetado para ter ultra-baixo consumo de energia pela TSMC (*Taiwan Semiconductor Manufacturing Company)*. Espressif Systems (2018) O ESP32 foi projetado especialmente para aplicações móveis, *wearables* (roupa inteligente) e *Internet-of-Things* (IoT). Neste projeto, utilizou-se uma versão do mesmo: o ESP32-WROOM-32. Neste, existem dois núcleos de CPU que podem ser controlados individualmente e a frequência de relógio da CPU é ajustável de 80 MHz a 240 MHz. O utilizador também pode desligar a CPU e fazer uso do co-processador de baixa potência para monitorizar constantemente os periféricos. O ESP32 pode integrar um vasto conjunto de periféricos, desde sensores de toque capacitivos, sensores Hall, interface de cartão SD, Ethernet, SPI de alta velocidade, UART, I2S e I2C.

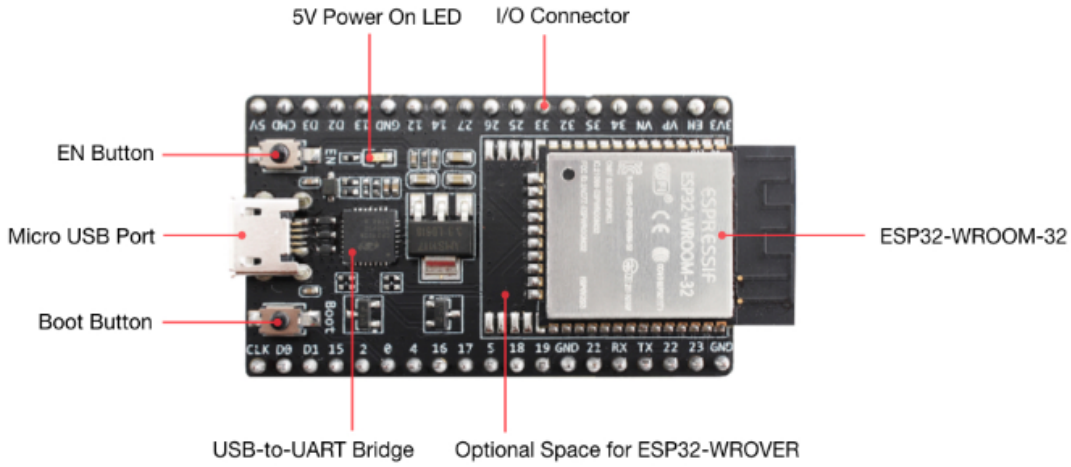



Figura 4.7: ESP32 DevKit v4 da ESPressif

Para o âmbito do desenvolvimento do projecto LifeBracelet trabalhou-se com um Kit de Desenvolvimento da ESPressif que pode ser visto na figura 4.7.

| Componente | Descrição |
|---|---|
| ESP32-WROOM-32 | Um módulo com o ESP32 no seu núcleo. |
| EN | Botão de reset. |
| Boot | Manter pressionado o botão Boot e, em seguida, pressionar EN inicia o modo Download de firmware para descarregar o firmware através da porta série. |
| Bridge USB-to-UART | O bridge USB-UART único fornece taxas de transferência de dados com velocidade até 3 Mbps. |
| Porta Micro USB | Fonte de alimentação para a placa, bem como a interface de comunicação entre o computador e o módulo ESP32. |
| 5V Power On LED | Acende quando a fonte de alimentação USB ou 5V externa é conectada à placa. |
| I/O | A maioria dos pinos no módulo ESP é dividida nos cabeçalhos dos pinos na placa. Pode-se programar o ESP32 para habilitar múltiplas funções como PWM, ADC, DAC, I2C, I2S, SPI, etc. |

Tabela 4.3: Componentes do DevKit

| Características | |
|---|---|
| Processador Principal | LX6 32-bit Dual-core, operando 2-240 MHz |
| Processador secundário | ULP (Ultra Low Power coprocessor) 8MHz e consome 150uA |
| Flash | 4Mb |
| RAM | 520Kb |
| GPIO | 34, com 3.3V e 12mA. |
| ADC | 8, com resolução de 12-bit |
| DAC | 2, com resolução 8-bit |
| WiFi | 2,4 GHz, 802.11 b/g/n. |
| Bluetooth | Bluetooth Low Energy v4.2 (BLE) |

Tabela 4.4: Características do ESP32

## 4.4 Protocolo I2C

O *Inter-Integrated Circuit* ou I2C é um protocolo muito popular e muito usado por muitos dispositivos eletrónicos de curto alcance dos dias de hoje. Inventado pela *Philips Semiconductors* em 1989, este tipo de implementação tem como principal vantagem necessitar de apenas duas linhas para comunicar até 128 ou 1024 dispositivos, dependendo do modo de endereçamento (usando 7 ou 10 bits - $2^7$ = 128, $2^10$ = 1024). Isto é possível porque cada um dos dispositivos tem um endereço diferente que o master utiliza para sinalizar o dispositivo com o qual quer comunicar. Isto pode trazer grandes benefícios para projetos de larga escala e traz uma implementação muito mais modular e facilitada.

As duas linhas chamam-se *Serial Clock* (SCL) e *Serial Data* (SDA). O primeiro transmite o sinal de *Clock* que sincroniza a comunicação entre os dispositivos no barramento I2C que é gerado pelo dispositivo "master". O segundo transporta a informação.

A sequência do protocolo de dados é transmitido na seguinte ordem:

- Bit de início de comunicação (Start);
- Endereço do "slave" (7bits);
- 1 bit de R/W (read ou write) que indica se o master que ler (bit a 1) ou escrever para (bit a 0) o "slave";
- 1 Bit de Acknowledge (ACK) do "slave";
- 8 bits de dados seguidos de 1 bit the ACK do "slave" por cada 8 bits de informação em loop;
- Bit de fim de comunicação (Stop).

Podemos ver uma pequena síntese do mesmo no esquema da figura 4.8:

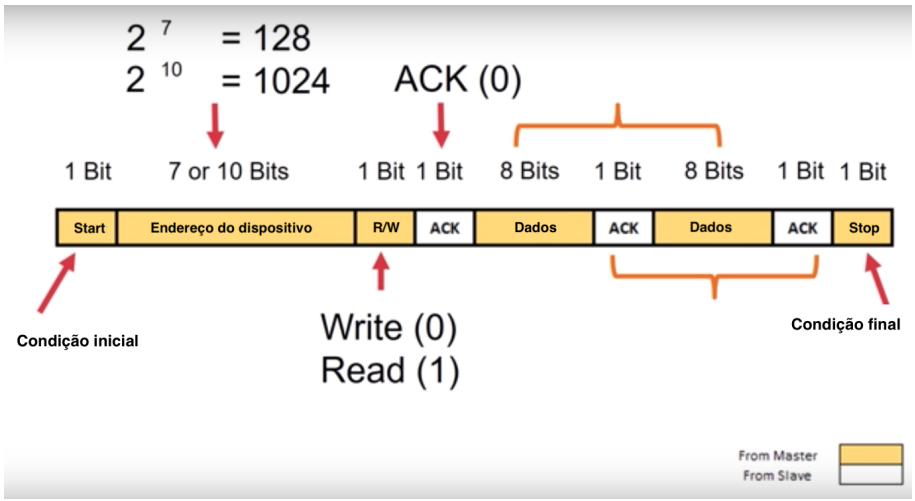



Figura 4.8: Esquema de protocolo I2C

## 4.5 Bluetooth

### 4.5.1 BLE vs Bluetooth 4.0

Como mencionado na tabela 4.4 uma das opções de conectividade que o ESP32 oferece é Bluetooth Low Energy (BLE).

Em resumo, o *Bluetooth* e o *Bluetooth Low Energy* são usados para propósitos muito diferentes. O segundo não foi lançado como uma versão do Bluetooth 4.0 mas sim como uma alternativa à forte demanda que começava a aparecer com o surgimento da *Internet of Things*. Enquanto o Bluetooth pode ser melhor para lidar com muitos dados, consome bateria muito mais rapidamente, além de ser bastante mais caro de implementar num projeto. O BLE é usado para aplicações que não precisam de transferir dados em grandes quantidades e que podem depender de uma bateria durante anos a fio a um custo mais baixo. Na tabela seguinte pode-se observar as diferenças entre o BLE e o Bluetooth, bem como outros protocolos muito comummente usados no dias de hoje como o *Wi-Fi* e o *ZigBee*.

| | Bluetooth Normal | BLE | ZigBee | Wi-Fi |
|---|---|---|---|---|
| Padrão de camada física | IEEE 802.15.1 | GFSK | IEEE 802.15.4 | IEEE 802.11abg |
| Frequências | 2.4 GHz | 2.4 GHz | 868 MHz - 2.4 GHz | 2.4 GHz - 5 GHz |
| *Bitrate* (Mbps) | 1 até 3 | 1 | 0.25 | 11(b), 54(g), 600(n) |
| Distância Máxima (m) | 10-100 | 50 | 10-100 | 100-250 |
| Consumo | Alto | Muito Baixo | Muito Baixo | Muito Alto |
| Vida Útil da Bateria | Dias | Meses a anos | Meses a anos | Horas |
| Tamanho da Rede | 7 | Indefinido | 64,000+ | 255 |

Tabela 4.5: Eficiência do BLE vs outros protocolos

### 4.5.2 Serviços GATT

Os Serviços do GATT são conjuntos de características e relacionamentos com outros serviços que englobam o comportamento de uma parte de um dispositivo Bluetooth-SIG. Este estabelece em detalhe como funciona a troca de dados de algum perfil ou utilizador por uma conexão BLE.

Todos os perfis BLE padrão são, portanto, baseados no GATT e devem obedecer às suas regras para operar corretamente. Isso torna o GATT uma secção chave deste protocolo, já que cada item de dados relevante para as aplicações e utilizadores deve ser formatado, encriptado e enviado de acordo com as suas regras. O GATT usa o Protocolo de Atributos (detalhado no ATT (*Attribute Protocol*) como o seu protocolo de transporte para trocar dados entre dispositivos. Esses dados são organizados hierarquicamente como observado na figura 4.9.

O nível superior da hierarquia é um **perfil**, que é composto por um ou mais **serviços**. Um dispositivo BLE pode conter mais do que um serviço.

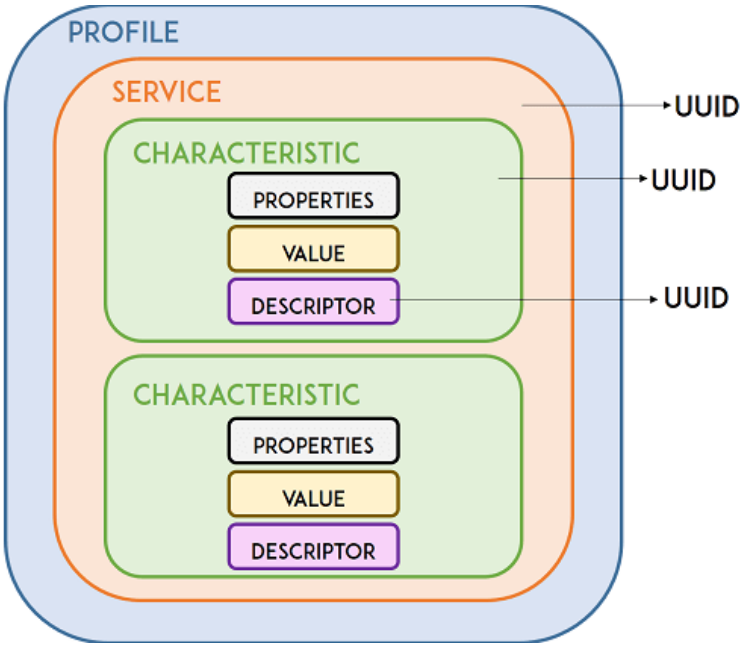



Figura 4.9: Hierarquia do GATT. Encontrada em Bluetooth-SIG.

Cada **serviço** contém pelo menos uma **característica** ou também pode referenciar outros serviços. Um serviço é simplesmente uma coleção de informações, como leituras de sensores, por exemplo.

Existem serviços pré-definidos para vários tipos de dados definidos pelo *SIG* como: Nível da Bateria, Pressão Sanguínea, Frequência Cardíaca, Escala de Peso, etc.

Cada serviço, característica e descritor tem um UUID (*Universal Unique Identifier*). Um *UUID* é um número exclusivo de 128 bits (16 bytes). Por exemplo: para o caso da *Life Bracelet* pode-se observar na figura 4.10, retirado do *website* da SIG, que o UUID do serviço é "0x180D".

| | | | |
|---|---|---|---|
| Health Thermometer | org.bluetooth.service.health_thermometer | 0x1809 | GSS |
| Heart Rate | org.bluetooth.service.heart_rate | 0x180D | GSS |
| HTTP Proxy | org.bluetooth.service.http_proxy | 0x1823 | GSS |
| Human Interface Device | org.bluetooth.service.human_interface_device | 0x1812 | GSS |
| Immediate Alert | org.bluetooth.service.immediate_alert | 0x1802 | GSS |
| Indoor Positioning | org.bluetooth.service.indoor_positioning | 0x1821 | GSS |

Figura 4.10: UUID dos serviços

## 4.6 Software

Durante todo o processo foram disponibilizados vários *softwares* não só para edificar e desenhar as placas, mas também para programar os componentes e escrever os resultados como:

- XCode versão 10.2.1 (*software* de desenvolvimento para iOS);
- KiCad versão 5.0.1 (desenho de circuitos e PCBs);
- Arduino IDE versão 1.8.9 (programação do microcontrolador);
- Light Blue versão 3.0.0 (Conexão de dispositivos BLE);

Todos os anteriores são de utilização livre de pagamento.

# Capítulo 5

# Conceção e Desenvolvimento do Dispositivo

Como tentativa inicial de idealizar o produto em questão, optou-se por um sensor digital para se obter medições precisas de ritmo cardíaco e oximetria e, simultaneamente, reduzir o tamanho do circuito. O sensor MAX30102 foi o disponibilizado pela TecPorto. Optou-se pelo uso do BLE visto que, no projeto em questão, se pretendia uma transferência de dados esporádica, comparada por exemplo com um *streaming* mais contínuo como os colunas *Bluetooth*. Além disso, um dos objetivos fulcrais do trabalho era precisamente melhorar a eficiência energética para se chegar a um produto final com dimensões aceitáveis para o mercado em questão.

No capítulo presente pode-se concluir que opções foram escolhidas de todas as ponderadas no capítulo anterior. Seguidamente é explicado como foi desenhado e impresso o circuito e, por último, como foram conectadas as partes para chegar à arquitetura final do sistema.

## 5.1 Sensor MAX30102

O MAX30102 funciona com uma única fonte de alimentação de 1,8V e uma fonte de alimentação separada de 3,3V para os LEDs internos. A comunicação faz-se através de um interface compatível com I2C, um protocolo que permite que vários circuitos integrados digitais "slave" comuniquem com um ou mais chips "master", tornando a implementação mais modular e acessível.



Figura 5.1: Max30102

Pode-se observar algumas características que levaram à escolha deste sensor:

- Monitor de Frequência Cardíaca e Bio-sensor de Oxímetro de Pulso em Solução Refletora LED
- Módulo ótico de 14 pinos minúsculo de 5,6 mm x 3,3 mm x 1,55 mm
  - Vidro de Cobertura Integrado para Desempenho Ótima e Robusto
- Operação de energia ultra-baixa para dispositivos móveis
  - Taxa de amostragem programável e corrente de LED para economia de energia
  - Monitor de frequência cardíaca de baixa potência (<1mW)
  - Corrente de Desligamento Ultra-Baixa (0.7 $\mu A$, typ)
- Capacidade de output de dados rápida
  - Taxas de Amostragem Altas
- Resiliência de artefatos de movimento robusto
  - SNR alto
- -40 º C a + 85 º C Faixa de temperatura operacional

## 5.2 Desenho e Impressão do Circuito

Foi então desenhada uma PCB (Printed Circuit Board) e, com recurso a soldadura *Surface Mount Device* (SMD), colocou-se todos os componentes do circuito.

Em primeiro lugar, começou-se por desenhar o circuito no KiCad, um programa de código aberto para projetos de circuitos integrados, com o objetivo de facilitar a conceção de *layouts* e suas conversões para placas de circuito impresso.

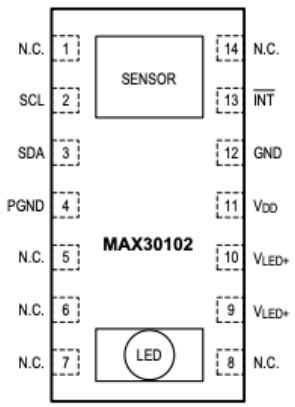



Figura 5.2: Esquema de pins do MAX30102. Encotrado em MAX.

Seguindo a *datasheet* do MAX30102, foi possível fazer um circuito mais eficiente em termos energéticos e muito mais pequeno que os anteriores.[1]

Os componentes são os seguintes:

- 3 Condensadores de 10$\mu$ F (Farad) e 1 de 4.7$\mu$ F;
- LM3671 - conversor DC-DC optimizado para alimentar circuitos de baixa tensão;
- 5 conectores para *breadboard*;
- Uma bobine de 2,2 $\mu$H;

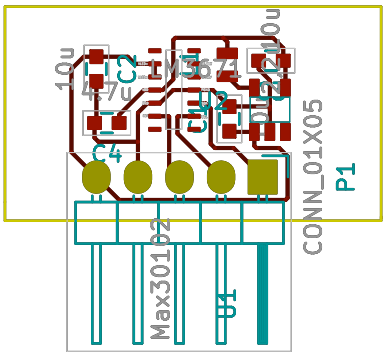



Figura 5.3: Esquema da PCB do MAX30102

O passo seguinte seria a impressão e soldadura da PCB. Este passo foi extremamente demorado já que era um processo completamente novo e que requer muita precisão devido ao tamanho dos componentes que o integram a placa de circuito impresso. Pode ser assim dividido em cinco passos distintos:

1. Inicialmente foi impresso o circuito da figura 5.3 três vezes numa folha de acetato com o auxílio de uma impressora a laser, como é possível verificar na segunda página do Anexo B. Seguidamente, este acetato é preso numa PCB fotossensível e é imediatamente colocado numa caixa fechada exposta a luz ultravioleta. No caso, o recurso disponibilizado pela Associação TECPorto.

2. Depois de alguns minutos retirou-se a PCB e colocou-se em revelador apropriado para a película foto-sensível da PCB. Passado alguns minutos reparou-se que o circuito impresso

[1] Nem a *datasheet* nem excertos do seu conteúdo podem ser apresentados por ter sido disponibilizada sob acordo de confidencialidade à TecPorto.

se começa revelar na placa. Nesse momento apenas a tinta que não recebeu luz ultravioleta vai ficar na placa e a que ficou exposta vai ser dissolvida. A zona de película que ficou na placa é chamada de máscara, ficando o restante cobre exposto.

3. Posteriormente, a placa foi mergulhada em cloreto de ferro onde ocorreu o processo de gravura química: o ataque seletivo pelo cloreto em diferentes áreas da superfície da peça é controlado pela "máscara". Após a remoção do cobre exposto, é necessário lavar a placa.

4. Por último, foi necessário soldar os componentes no sítio, sendo que a técnica aplicável depende do componente: alguns componentes são soldados através de técnicas SMD, em que se coloca o componente e, com a ajuda de uma fonte de ar quente, a solda derrete e por tensão superficial alinha o próprio componente; outros componentes são soldados pela técnica clássica, usando ferro de soldar.

5. Como passo extra pode-se também cobrir a PCB, depois de soldada, com verniz para evitar corrosão e exposição a situações que possam facilitar a ocorrência de curto-circuitos.

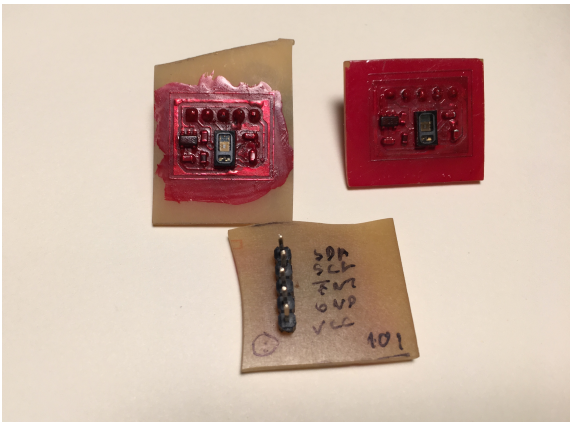


Figura 5.4: Versão final dos MAX30101 e 102

A figura 5.4 demonstra uma versão final do MAX30101(em baixo) e duas do MAX30102 (em cima). Todas os pins estão devidamente identificadas. De cima para baixo podemos observar:

- SDA e SCL (pins de transferência I2C mencionados na secção 4.4),
- INT (que serve para provocar interrupts externos),
- GND (ground) ,
- VCC (5V).

## 5.3 Arquitetura final do sistema

Na figura 5.5 observa-se a arquitetura do sistema. O utilizador pode interagir com o sistema através de um dispositivo iOS que está ligado e que comunica com o ESP32 por BLE (*Bluetooth Low Energy*). O circuito do MAX30102 faz todas as medições necessárias e envia os valores em forma digital.

O ESP32 está pré-carregado com o programa que gere e realiza todas as medições. O utilizador apenas tem que ligar a placa a qualquer fonte de alimentação para funcionar, até numa porta USB no computador. Tendo a aplicação *Life Bracelet* instalada no telemóvel, está pronto a funcionar.

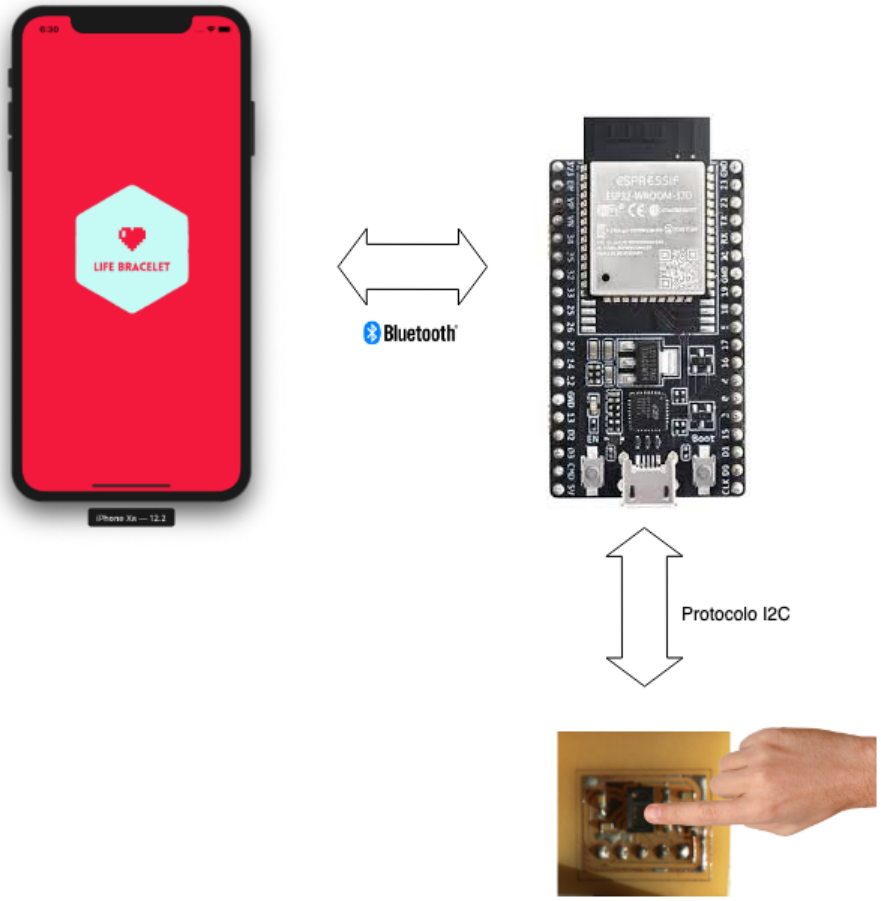



Figura 5.5: Arquitetura do Sistema

# Capítulo 6

# Desenvolvimento do Software

Este capítulo aborda a implementação de software solução para tratamento de dados provenientes do MAX30102, sendo descritas as estratégias utilizadas em função do circuito desenvolvido no capítulo anterior. Este algoritmo será processado pelo microcontrolador, e trata não só de processar o sinal enviado por I2C pelo sensor mas de o transformar em batimentos por minuto. Tem também de transformar os dados para serem enviados para o iPhone por BLE. Em último lugar é apresentado a solução pensada para apresentar os dados ao utilizador final através de uma interface desenvolvida para iOS.

## 6.1 Programação

A programação do microprocessador foi feita utilizando o IDE do Arduino. Este IDE, por ser uma referência entre microprocessadores já inclui, desde a versão 1.6.4, uma forma de incluir facilmente a partir da opção *Board Manager* o suporte para outras placas como é o caso da ESP32. Isto torna extremamente simples a compilação e *upload* do código para a placa.

A linguagem utilizada é C++ com algumas bibliotecas específicas que facilitam o acesso a vários recursos, como funções que podem ser chamadas para controlar o hardware. Se não existissem essas funções seria preciso programar para os registos de cada uma das placas. Ora isto garantiu uma maior escalabilidade do projeto, embora implique a utilização de mais memória. Para, por exemplo, passar do código do Arduino Nano para a ESP32, bastou importar uma nova *Board* para o IDE, ficando imediatamente disponíveis as funções (como, por exemplo, serial.print()) adaptadas aos registos específicos desta placa. Numa fase posterior, no caso de um lançamento de um produto final para o mercado, será apenas necessário adaptar algumas partes do código para a placa específica de maneira a utilizar de forma mais eficiente a memória. Qualquer novo projeto deste IDE começa sempre com a seguinte estrutura:

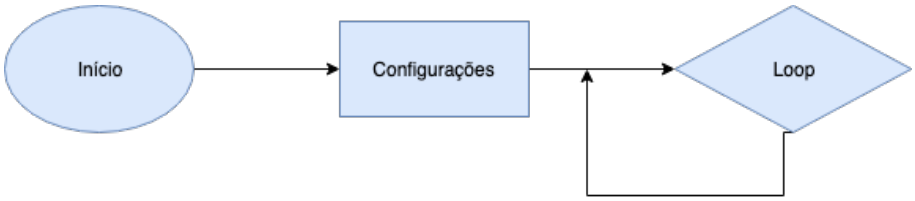



Figura 6.1: Lógica geral de programação

## 6.2 Programa do microcontrolador

Estes sensores de fotopletismografia são muito comuns no mercado e, por isso, a *Maxim* disponibiliza algumas bibliotecas de funções que podem ser importadas para o *IDE* do *Arduino* e facilitar a programação do mesmo. Muitos destes algoritmos já foram otimizados para o sensor havendo pouca necessidade de alteração, reduzindo-se frequentemente a ajustes necessários para melhor satisfazer os requisitos do produto - como, por exemplo, leituras por segundo, tipos de LED a usar de acordo com a energia que se quer consumir ou a frequência com que se efetua as médias das medições. Como tal, a programação do microprocessador pode ser dividida em duas partes: configuração do hardware (sensor e módulo BLE da ESP32) e processamento dos dados (cálculo do da frequência cardíaca e transmissão dos dados).

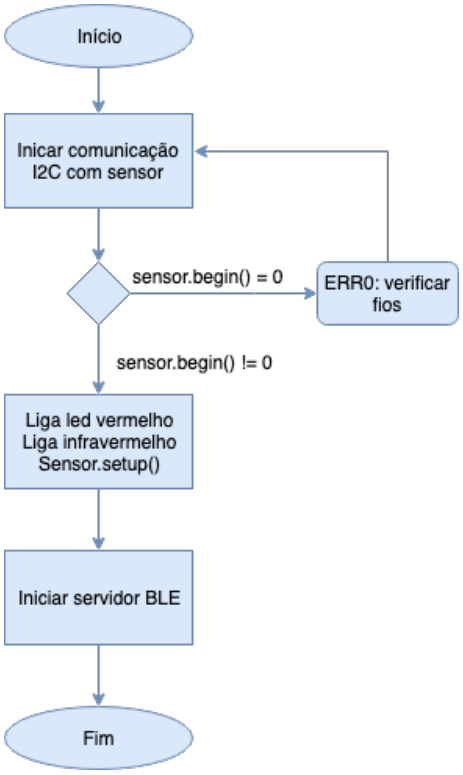



Figura 6.2: Lógica da configuração

Nesta primeira função void setup(), são feitas as configurações necessárias não só ao funcionamento do sensor mas também a inicialização do servidor BLE. Tenta-se iniciar uma comunicação por I2C com o sensor com a função particleSensor.begin(). No caso de o resultado ser igual a zero significa que não está a conseguir estabelecer comunicação e envia uma mensagem ao utilizador para verificar a cablagem. Caso contrário liga os LEDs para sinalizar que está funcional e executa a função Init.BLE() que vai criar o serviço e as suas características e termina a função setup().

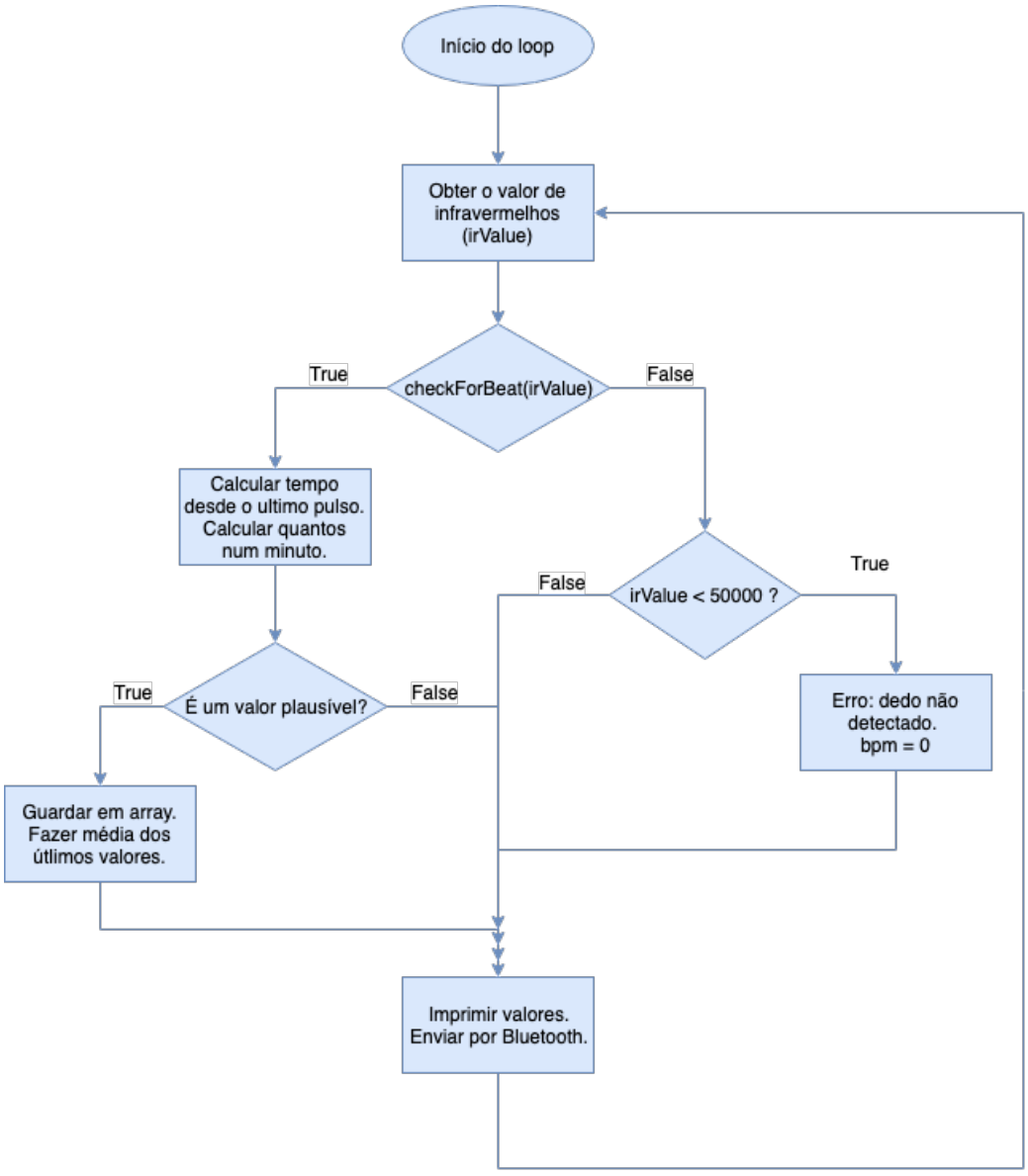



Figura 6.3: Lógica do loop

Como é possível ver pelo diagrama de atividade da figura 6.3, primeiro verifica-se o valor de infravermelhos lido pelo sensor. Se este valor corresponder a um batimento é calculado o tempo entre este e o último valor detetado e é feita uma média dos batimentos por minuto. No caso de

não ser detetado, verifica-se se o valor de infravermelhos está muito baixo, o que significa que nenhum dedo está encostado no sensor. Neste caso, o valor de batimentos por minuto será reposto a zero para depois, como se poderá consultar no capítulo seguinte, esse valor seja processado pela aplicação e se mostre uma mensagem de erro ao utilizador.

A implementação de referência da saturação de oxigénio, contrariamente ao algoritmo de medição de ritmo cardíaco que foi desenvolvido é baseado em informação disponível online, não está disponível publicamente mas apenas sobre acordo de confidencialidade que não permite à TEC-Porto a divulgação a terceiros. Por este motivo o seu código fonte não pode ser incluído neste relatório. Por outro poderão ser divulgados todos os resultados obtidos.

## 6.3 Interface do utilizador

A interface do utilizador é uma vertente muito importante do projeto visto ser a única parte do mesmo a que o utilizador tem acesso. Embora seja necessário criar um sistema eficiente e complexo, não pode ser descurada a parte da interação com o utilizador, já que a apresentação de resultados pode ser tão importante como os resultados em si. Não havendo uma interação compreensível, responsiva e em tempo real com uma plataforma que seja de fácil acesso, todos os processos que estejam a correr em segundo plano, por mais complicados que sejam, serão completamente inúteis. Hoje em dia existem mais de 1.3 mil milhões de dispositivos iOS ativos no mundo e, como tal, optou-se por se desenvolver uma aplicação para o mesmo sistema operativo. Por outro lado era também a plataforma mais conhecida durante a programação e à que se tinha melhor acesso.

### 6.3.1 Programação para iOS

Como referido na secção 4.6, esta interface foi desenhada em Xcode. Este é o ambiente de desenvolvimento integrado (IDE) da Apple usado para criar aplicações para produtos da mesma, incluindo o iPad, iPhone, Apple Watch e Mac. O Xcode fornece ferramentas para gerir todo o *workflow* da criação, *debugging* até ao envio para a AppStore.

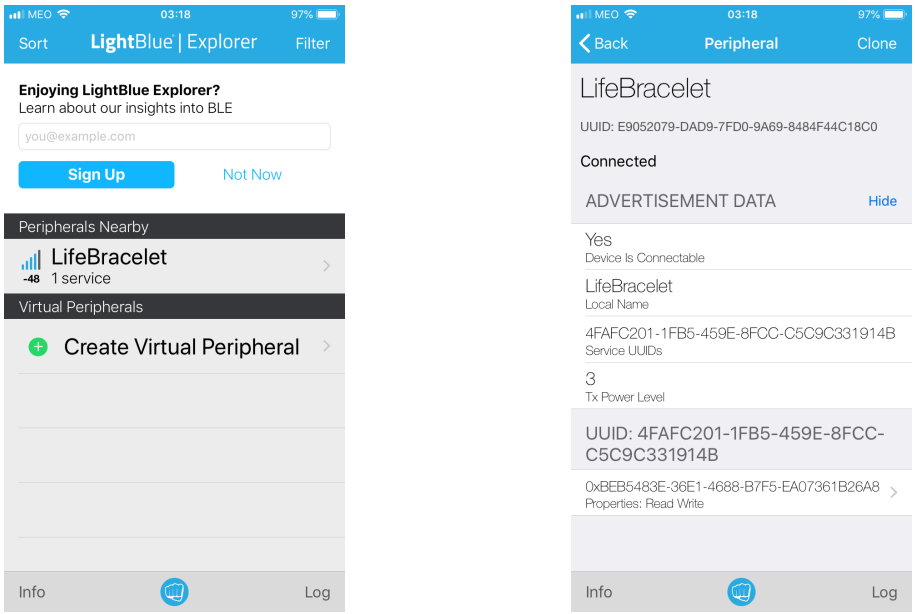
A programação foi feita em Swift 5.0, uma linguagem alternativa à linguagem Objective-C que emprega conceitos modernos da teoria de linguagem de programação e se empenha em oferecer uma sintaxe mais simples. Esta foi lançada em 2014 pela Apple para desenvolvimento no iOS, macOS, watchOS, tvOS e Linux.

Para tal, teve-se em consideração um livro disponibilizado gratuitamente pela Apple chamado "The Swift Proggraming Language" que explica como funciona exatamente a linguagem e que serviu como base para desenvolver o interface.

### 6.3.2 Testes de Conectividade

Antes de mais, procedeu-se à realização de testes para determinar se o servidor BLE do ESP32 era reconhecido pelo iPhone. Para isso usou-se a aplicação LightBlue que pesquisa dispositivos

BLE, já que o *scanner* padrão dos dispositivos só suporta Bluetooth nativo.




(a) Dispositivo LifeBracelet (b) Características do serviço

Figura 6.4: Aplicação LightBlue

Do lado esquerdo, na figura 6.4a, pode-se observar que o iPhone consegue encontrar o serviço BLE que o nosso microcontrolador está a emitir. Já na figura 6.4b, pode-se observar algumas das características do mesmo, bem como o seu UUID (como descrito na subsecção 4.5.2) e das suas características, valor que será necessário para programar a conexão bem sucedida entre a aplicação e o serviço.

Tendo-se confirmado que a parte do algoritmo da ESP32 estava operacional e que estava a ser detetado o serviço, passou-se, então, ao desenvolvimento da aplicação para iOS, pensando-se primeiro num modelo de apresentação dos dados e como estes seriam apresentados ao utilizador.

### 6.3.3 Aplicação iOS

No modelo seguinte pode-se observar quatro instâncias importantes na aplicação: o ecrã inicial - apresenta a aplicação e torna a marca mais reconhecível ao utilizador através uma imagem ou logótipo, o Medidor de Ritmo Cardíaco/Saturação de Oxigénio - a instância principal da aplicação, é onde este pode observar as medições, um *popup* de erro - caso o dedo não seja detetado e por último uma página “about” - onde o utilizador pode conhecer alguns factos sobre a aplicação. Estas instâncias relacionam-se entre si e podem ser mais facilmente observadas com a ajuda do esquema 6.5.

Iniciou-se o processo com o desenho de um logótipo para a aplicação para que fosse mais facilmente reconhecida na AppStore e para que a mesma fosse familiar. Apresenta-se então o ecrã inicial da aplicação com o mesmo logótipo.

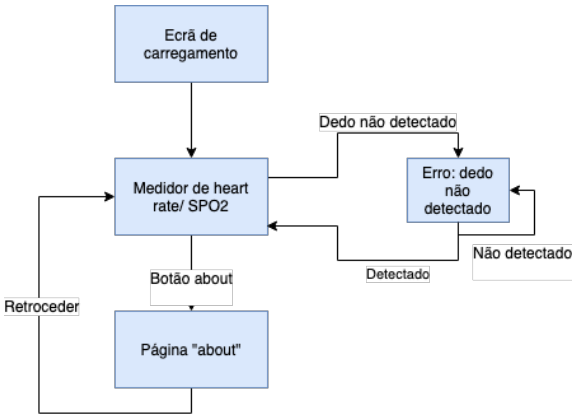



Figura 6.5: Modelo da aplicação

De seguida, o procedimento normal neste tipo de aplicações seria então conectar o dispositivo por Bluetooth ao dispositivo que fornece o serviço de medição de ritmo cardíaco. Contudo, por se tratar de uma comunicação por BLE em que os dois dispositivos conhecem o UUID do outro e reconhecem que é um serviço de medição de HR, sem ser preciso qualquer tipo de emparelhamento o dispositivo começa automaticamente a receber os dados.

Na figura 6.7 pode-se observar à esquerda, na figura 6.7a o estado inicial da aplicação. Ela permanecerá no modo de espera até encontrar algum serviço e começar a receber dados. No caso de tal acontecer passará automaticamente para o estado ativo (figura 6.7b). No anexo B é possível ver todo o código Swift que trata de todos estes dados.

Como temos definidos os UUID dos serviços e características (como mostrado no capítulo anterior) no microcontrolador, também aqui é necessário defini-los para que haja uma comunicação bem sucedida. Podemos ver um pequeno trecho desse mesmo código em baixo, agora em Swift.

```
let heartRateServiceCBUUID = CBUUID(string: "0x180D")
let heartRateMeasurementCharacteristicCBUUID = CBUUID(string: "2A37")
let bodySensorLocationCharacteristicCBUUID = CBUUID(string: "2A38")
```

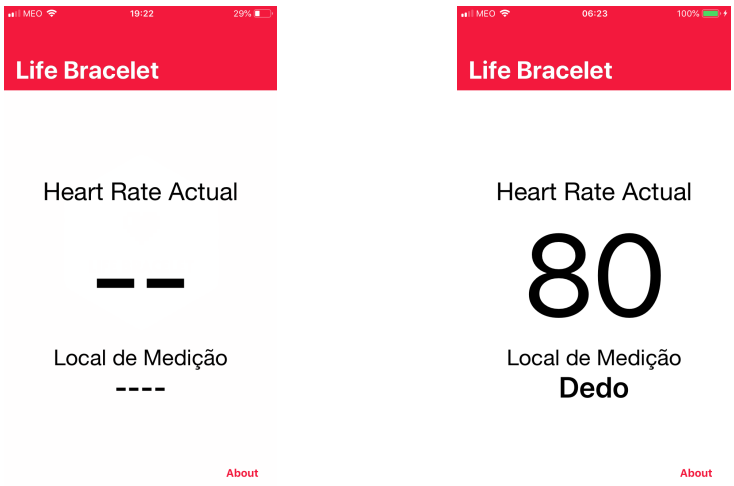



Figura 6.6: Ecrã inicial

Já do lado do microcontrolador é feita uma *flag* no caso do valor da variável *irValue* ser menor que 50000, como se pode verificar no esquema 6.3. No caso de tal acontecer significa que os valores que o MAX está a receber na zona de infravermelhos são muito baixos e, portanto, é razoável presumir que não há nenhum dedo no sensor. Caso tal aconteça, o valor de bpm é posto a 0, o que irá indicar à aplicação, depois de ser transmitido, que não existe nenhum dedo no sensor e que por isso deverá apresentar uma mensagem ao utilizador.

No erro aparece a mensagem: "Aviso: Nenhum dedo detetado (figura 6.8). Encoste o centro do dedo indicador no sensor e pressione OK". Caso o utilizador pressione o botão e não tiver o dedo no sensor, a aplicação voltará imediatamente ao mesmo estado. Quando este puser o dedo no sensor e pressionar "OK" poderá então voltar a ler valores.

Por último foi também desenvolvida uma *About Page* que abre quando o botão é pressionado. Este ecrã mostra apenas algumas informações acerca da aplicação (figura 6.9).




(a) Estado pronto a receber (b) A receber dados

Figura 6.7: Medidor de ritmo cardíaco




Figura 6.8: Ecrã de erro quando não deteta o dedo




Figura 6.9: Informações sobre a aplicação

# Capítulo 7

# Resultados e Conclusão

Resultados de teste de validação e comparação com dispositivo comercial Discussão desses resultados.

## 7.1 Resultados dos testes

Este dispositivo foi testado em vários sujeitos de idades e sexos diferentes já que, como demonstrado no capítulo 2, estes valores diferem de acordo com os mesmos. Na tabela seguinte podemos observar os resultados comparados com dois dispositivos:

- Medidor de dedo de SPO2 e ritmo cardíaco da Sanitas, para uso profissional em situações de emergência. Este dispositivo usa fotopletismografia de transmissão.
- Medidor de ritmo cardíaco da *GeoNaute*. Este dispositivo é usado para desporto e faz medições por eletrocardiografia.

| Indivíduo | MAX30102 | Sanitas SPO25 (PPG) | Geonaute (ECG) |
|---|---|---|---|
| 25 anos, Masculino | 77 bpm<br>97% SpO2 | 77 bpm<br>98% SpO2 | 76 bpm<br>— |
| 20 anos, Feminino | 68 bpm<br>99% SpO2 | 69 bpm<br>99%SpO2 | 69 bpm<br>— |
| 20 anos, Masculino, Desportista | 65 bpm<br>99%SpO2 | 66 bpm<br>99%SpO2 | 66 bpm<br>— |
| 50 anos, Masculino | 75 bpm<br>97%SpO2 | 74 bpm<br>98%SpO2 | 75 bpm<br>— |

Tabela 7.1: Comparação de resultados

Observando a tabela acima, é possível verificar que os valores obtidos pelos sistemas desenvolvidos se aproximam todos aos medidos por equipamentos comercialmente disponíveis, o que poderá viabilizar utilização dos sistemas baseados no protótipo desenvolvido. Na figura 7.1 é possível observar uma comparação entre um valor obtido por um dos equipamentos e a aplicação LifeBracelet.

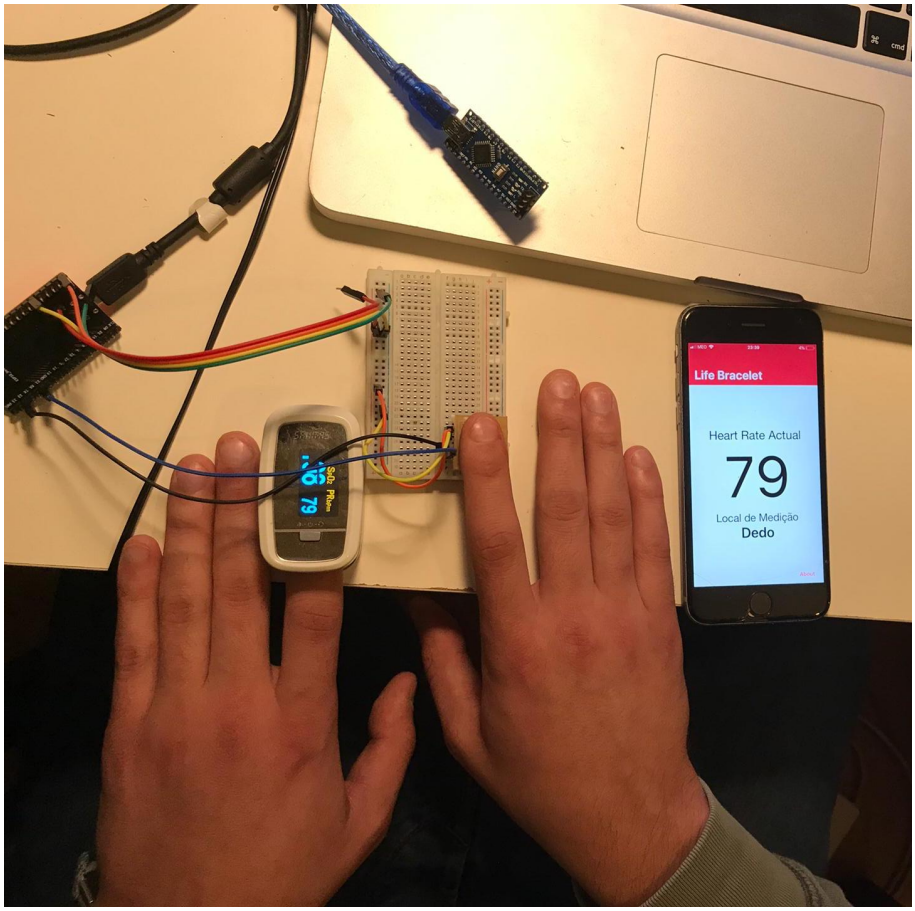



Figura 7.1: Sanitas vs LifeBracelet

Por último tentou-se fazer uma pequena estimativa do gasto energético do sistema. Segundo alguns testes pesquisados sobre o ESP32 numa duração de 48h no modo *Ultra Low Power* e com conexão *Bluetooth* ou *Wi-Fi* é expectável um consumo de 115mA. No caso de estar no modo *Deep Sleep*, quando a conexão é desligada, é estimado que consuma apenas à volta de 5.6mA.

No caso da pulseira enviar dados a cada 30 segundos pode-se estimar que:

$$Consumo = (eventos_{hora} * tempo_{evento}/3600s) * corrente) \quad (7.1)$$

$$Consumo = 120(eventos/hora) * 1s(envio)/3600s * 115mA \quad (7.2)$$

$$Consumo \quad = \quad 3,87mAh \tag{7.3}$$

$$3000/3.87*24h \quad \equiv \quad 32dias \tag{7.4}$$

Sendo o MAX30102 também de Ultra Baixo Consumo (1mWh) é possível inferir, que à tensão de 5V tenhs uma consume de aproximadamente 0.2mAh.

Usando uma bateria comum de 3000mAh pode-se estimar que dure então à volta de :

$$3000/(3.87[ESP32]+0.2[MAX30102])*24h \quad \equiv \quad 30,45dias \tag{7.5}$$

Nestas condições pode-se inferir que usando estas configurações se poderia fazer durar a pulseira durante aproximadamente um mês.

## 7.2 Conclusão

O objetivo principal do trabalho proposto consistia no melhoramento dos circuitos até então desenvolvidos de maneira a torna-los mais eficientes em termos energéticos e ao mesmo tempo mais pequenos para que se aproximassem mais de um produto final. Nesse aspeto sentiu-se que o trabalho desenvolvido foi bem sucedido e que se chegou um pouco mais perto de desenvolver um produto que poderá, quem sabe, proporcionar um estilo de vida mais confortável aos seus utilizadores e, por outro lado, criar algo a um preço muito mais acessível que possa vir a ser um substituto de um sistema hospitalar que se encontra desatualizado em comparação com o mundo tecnológico que o acompanha. Os equipamentos que podemos encontrar ainda são extremamente pouco eficientes em termos de conectividade e o tratamento de dados que é feito entre pacientes ainda se encontra pouco automatizado. A implementação de um produto como a *LifeBracelet* poderá disponibilizar uma monitorização muito mais simples e confortável para um paciente mas mais do que isso: contínuo e conectado. Isto poderá levar a novas descobertas científicas através do processamento desses mesmos dados.

O protótipo prometeu reforçar grande parte das competências adquiridas ao longo do Mestrado em Engenharia Eletrotécnica e de Computadores. A integração numa associação como a TecPorto permitiu a aquisição de novos conhecimentos em diversas áreas da engenharia bem como uma amostra de como lidar com o mercado de trabalho.

Depois de se perceber os objetivos, deu-se então ao início ao desenvolvimento do protótipo. Numa primeira fase procurou-se as soluções que constituem o mercado atual no que diz respeito a sistemas de obtenção de ritmo cardíaco. Depois de escolhido o sensor tendo em conta os requisitos da Associação TecPorto e os seus recursos, começou-se a pensar no dimensionamento do circuito.

Para implementação do sensor MAX30102 foi necessário entender protocolos como o I2C que, embora simplifique bastante a comunicação depois de perceber como funciona, relevou apresentar uma curva de aprendizagem bastante longa.

Ainda no processo do dimensionamento, foi necessário estudar a vários níveis o processo de criação "amador" de uma PCB. O processo apresentado no capítulo 5 é muito demorado, desde o desenho da placa até à sua impressão e soldadura. A interface gráfica foi criada com base numa aplicação para iOS, utilizando a comunicação BLE entre os dispositivos, apresentado-se, mais uma vez, novos conhecimentos adquiridos durante o projeto. Após a criação do interface de utilizador, foi possível observar os valores obtidos para ambas as soluções e ver quão próximos estão daqueles encontrados em produtos já existentes no mercado, demonstrando assim a possibilidade de integrar os protótipos desenvolvidos com o projeto *Life Bracelet*. A linguagem usada para a desenvolver, o *Swift*, também se revelou nova e de longa aprendizagem. Embora seja uma linguagem moderna e bem estruturada difere muito de todas as cobertas durante o curso de Engenharia Eletrotécnica. Contudo, revelou-se também no final por ser uma ótima experiência de aprendizagem e, quem sabe, mais uma valência para o mercado de trabalho.

Como conclusão final, tira-se deste projeto uma grande capacidade de entendimento como do planeamento, projeção e conceção de um novo produto. Não só a nível de gestão - saber adaptar os materiais disponíveis e gerir o tempo a gastar em cada parte do projeto - mas principalmente a nível de engenharia. Desde a conceptualização e desenho de um circuito, à sua impressão e processo de montagem, desde a escolha de protocolos de comunicação ao seu entendimento. Não só tentar perceber o mais indicado mas quais as vantagens e desvantagens que este pode trazer ao produto.A programação de cada parte e como esta vai afetar o desempenho final a nível de memória, largura de banda e outros recursos. Como tal, considera-se que foi um projeto completo a todos níveis e que foi uma grande preparação para um futuro inserido na área da engenharia.

## 7.3 Trabalho Futuro

Em termos de planeamento futuro do projeto, poderá melhorar-se o produto a vários níveis. Embora se tenha reduzido em muito o tamanho do mesmo e melhorado a sua eficiência energética, o passo a seguir será abandonar os kits de desenvolvimento e soldar apenas o chip do microcontrolador de maneira a ocupar o mínimo de espaço possível. Dever-se-á também, como dito anteriormente, melhorar a eficiência do programa a nível de memória. Começar por usar menos bibliotecas e remover os *prints* que estão a ser usados para *debugging*. Deve-se também fundir as funcionalidades de medição de ritmo cardíaco e saturação de oxigénio, de forma a permitir que ambos os valores possam ser apresentados em simultâneo.

Depois de maximizar todos os pontos supramencionados deve-se tentar adicionar novas funcionalidades como: exibição dos valores num mostrador, gerar um gráfico que exponha a forma

de um batimento cardíaco e, por último, a medição não invasiva da glicose. Embora esta funcionalidade tenha sido referida como um dos objetivos sente-se que é um assunto que, por uma questão temporal e gestão de recursos não seria possível desenvolver com os recursos atualmente disponíveis online.

# Anexo A

# Código do microcontrolador

```
1
2  #include <BLEDevice.h>
3  #include <BLEUtils.h>
4  #include <BLEServer.h>
5  #include <BLE2902.h>
6  #include <Wire.h> //I2C
7  #include "MAX30105.h" //Max310102
8  #include "heartRate.h"
9
10
11 MAX30105 particleSensor;
12
13 const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
14 byte rates[RATE_SIZE]; //Array of heart rates
15 byte rateSpot = 0;
16 long lastBeat = 0; //Time at which the last beat occurred
17
18 float beatsPerMinute;
19 int beatAvg;
20
21
22 byte flags = 0b00111110;
23 byte bpm;
24 byte heart[8] = { 0b00001110, 60, 0, 0, 0 , 0, 0, 0};
25 byte hrmPos[1] = {2};
26
27 bool _BLEClientConnected = false;
28
29 #define heartRateService BLEUUID((uint16_t)0x180D)
30 BLECharacteristic heartRateMeasurementCharacteristics(BLEUUID((uint16_t)0x2A37)
       , BLECharacteristic::PROPERTY_NOTIFY);
31 BLECharacteristic sensorPositionCharacteristic(BLEUUID((uint16_t)0x2A38),
       BLECharacteristic::PROPERTY_READ);
32
33 BLEDescriptor heartRateDescriptor(BLEUUID((uint16_t)0x2901));
```

```
34 BLEDescriptor sensorPositionDescriptor(BLEUUID((uint16_t)0x2901));
35
36 class MyServerCallbacks : public BLEServerCallbacks {
37     void onConnect(BLEServer* pServer) {
38       _BLEClientConnected = true;
39     };
40
41     void onDisconnect(BLEServer* pServer) {
42       _BLEClientConnected = false;
43     }
44 };
45
46 void InitBLE() {
47   BLEDevice::init("LifeBracelet");
48   // Create the BLE Server
49   BLEServer *pServer = BLEDevice::createServer();
50   pServer->setCallbacks(new MyServerCallbacks());
51
52   // Create the BLE Service
53   BLEService *pHeart = pServer->createService(heartRateService);
54
55   pHeart->addCharacteristic(&heartRateMeasurementCharacteristics);
56   heartRateDescriptor.setValue("Frequencia cardiaca");
57   heartRateMeasurementCharacteristics.addDescriptor(&heartRateDescriptor);
58   heartRateMeasurementCharacteristics.addDescriptor(new BLE2902());
59
60   pHeart->addCharacteristic(&sensorPositionCharacteristic);
61   sensorPositionDescriptor.setValue("Posicao 0 - 6");
62   sensorPositionCharacteristic.addDescriptor(&sensorPositionDescriptor);
63
64   pServer->getAdvertising()->addServiceUUID(heartRateService);
65
66   pHeart->start();
67   // Start advertising
68   pServer->getAdvertising()->start();
69 }
70
71
72
73 void setup()
74 {
75   Serial.begin(115200);
76   Serial.println("Initializing MAX30102...");
77
78   // Initialize sensor MAX30102
79   if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400
       kHz speed
80   {
81     Serial.println("MAX30105 was not found. Please check wiring/power. ");
```

```
82      while (1);
83    }
84    Serial.println("Place your index finger on the sensor with steady pressure.")
        ;
85
86    particleSensor.setup(); //Configure sensor with default settings
87    particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate
        sensor is running
88    particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
89
90
91    // Initialize sensor MAX30102
92    InitBLE();
93    bpm = 0;
94
95
96  }
97
98  void loop()
99  {
100   //MAX30102
101   long irValue = particleSensor.getIR();
102
103   if (checkForBeat(irValue) == true)
104   {
105     //We sensed a beat!
106     long delta = millis() - lastBeat;
107     lastBeat = millis();
108
109     beatsPerMinute = 60 / (delta / 1000.0);
110
111     if (beatsPerMinute < 255 && beatsPerMinute > 20)
112     {
113       rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the
        array
114       rateSpot %= RATE_SIZE; //Wrap variable
115
116       //Take average of readings
117       beatAvg = 0;
118       for (byte x = 0 ; x < RATE_SIZE ; x++)
119         beatAvg += rates[x];
120       beatAvg /= RATE_SIZE;
121     }
122   }
123
124   Serial.print("IR=");
125   Serial.print(irValue);
126   Serial.print(", BPM=");
127   Serial.print(beatsPerMinute);
```

```
128   Serial.print(", Avg BPM=");
129   Serial.print(beatAvg);
130
131   if (irValue < 50000){
132     Serial.print(" No finger?");
133     bpm = 0;    //add lines apagar
134   }
135   Serial.println();
136
137
138   //BLE
139
140
141    heart[1] = (byte)bpm;
142   int energyUsed = 3000;
143   heart[3] = energyUsed / 256;
144   heart[2] = energyUsed - (heart[2] * 256);
145   Serial.print("BPM in BLE=");
146   Serial.println(bpm);
147
148   heartRateMeasurementCharacteristics.setValue(heart, 8);
149   heartRateMeasurementCharacteristics.notify();
150
151   sensorPositionCharacteristic.setValue(hrmPos, 1);
152
153   bpm = beatAvg;
154
155
156 }
```

Listing A.1: Programa em C da ESP32

# Anexo B

# Código aplicação

```
1
2
3  //  LifeBracelet
4  //
5  //  Created by Tiago Simoes on 25/05/2019.
6  //  Copyright 2019 Tiago Simoes. All rights reserved.
7  //
8  import UIKit
9  import CoreBluetooth
10
11 let heartRateServiceCBUUID = CBUUID(string: "0x180D")
12 let heartRateMeasurementCharacteristicCBUUID = CBUUID(string: "2A37")
13 let bodySensorLocationCharacteristicCBUUID = CBUUID(string: "2A38")
14
15 class HRMViewController: UIViewController {
16
17   @IBOutlet weak var heartRateLabel: UILabel!
18   @IBOutlet weak var bodySensorLocationLabel: UILabel!
19
20   var centralManager: CBCentralManager!
21   var heartRatePeripheral: CBPeripheral!
22
23   override func viewDidLoad() {
24     super.viewDidLoad()
25
26     centralManager = CBCentralManager(delegate: self, queue: nil)
27
28     heartRateLabel.font = UIFont.monospacedDigitSystemFont(ofSize:
       heartRateLabel.font!.pointSize, weight: .regular)
29   }
30
31   func onHeartRateReceived(_ heartRate: Int) {
32     heartRateLabel.text = String(heartRate)
33     print("BPM: \(heartRate)")
34   }
```

```
35 }
36
37 extension HRMViewController: CBCentralManagerDelegate {
38   func centralManagerDidUpdateState(_ central: CBCentralManager) {
39     switch central.state {
40     case .unknown:
41       print("central.state is .unknown")
42     case .resetting:
43       print("central.state is .resetting")
44     case .unsupported:
45       print("central.state is .unsupported")
46     case .unauthorized:
47       print("central.state is .unauthorized")
48     case .poweredOff:
49       print("central.state is .poweredOff")
50     case .poweredOn:
51       print("central.state is .poweredOn")
52       centralManager.scanForPeripherals(withServices: [heartRateServiceCBUUID])
53     }
54   }
55
56   func centralManager(_ central: CBCentralManager, didDiscover peripheral:
       CBPeripheral,
57                       advertisementData: [String : Any], rssi RSSI: NSNumber) {
58     print(peripheral)
59     heartRatePeripheral = peripheral
60     heartRatePeripheral.delegate = self
61     centralManager.stopScan()
62     centralManager.connect(heartRatePeripheral)
63   }
64
65   func centralManager(_ central: CBCentralManager, didConnect peripheral:
       CBPeripheral) {
66     print("Connected!")
67     heartRatePeripheral.discoverServices([heartRateServiceCBUUID])
68   }
69 }
70
71 extension HRMViewController: CBPeripheralDelegate {
72   func peripheral(_ peripheral: CBPeripheral, didDiscoverServices error: Error
       ?) {
73     guard let services = peripheral.services else { return }
74     for service in services {
75       print(service)
76       peripheral.discoverCharacteristics(nil, for: service)
77     }
78   }
79
```

```
80  func peripheral(_ peripheral: CBPeripheral, didDiscoverCharacteristicsFor
      service: CBService, error: Error?) {
81    guard let characteristics = service.characteristics else { return }
82
83    for characteristic in characteristics {
84      print(characteristic)
85
86      if characteristic.properties.contains(.read) {
87        print("\(characteristic.uuid): properties contains .read")
88        peripheral.readValue(for: characteristic)
89      }
90      if characteristic.properties.contains(.notify) {
91        print("\(characteristic.uuid): properties contains .notify")
92        peripheral.setNotifyValue(true, for: characteristic)
93      }
94    }
95  }
96
97  func peripheral(_ peripheral: CBPeripheral, didUpdateValueFor characteristic:
       CBCharacteristic, error: Error?) {
98    switch characteristic.uuid {
99    case bodySensorLocationCharacteristicCBUUID:
100     let bodySensorLocation = bodyLocation(from: characteristic)
101     bodySensorLocationLabel.text = bodySensorLocation
102   case heartRateMeasurementCharacteristicCBUUID:
103     let bpm = heartRate(from: characteristic)
104     onHeartRateReceived(bpm)
105     if (bpm == 0) {                                  //if bpm = 0 show alert
106       Alert.showIncompleteFormAlert(on: self)
107       }
108   default:
109     print("Unhandled Characteristic UUID: \(characteristic.uuid)")
110   }
111 }
112
113 private func bodyLocation(from characteristic: CBCharacteristic) -> String {
114   guard let characteristicData = characteristic.value,
115     let byte = characteristicData.first else { return "Error" }
116
117   switch byte {
118   case 0: return "Outro"
119   case 1: return "Peito"
120   case 2: return "Dedo" //trocado com pulso
121   case 3: return "Pulso"
122   case 4: return "Mao"
123   case 5: return "Orelha"
124   case 6: return "Pe"
125   default:
126     return "Nao foi possivel detectar"
```

```
127     }
128   }
129
130   private func heartRate(from characteristic: CBCharacteristic) -> Int {
131     guard let characteristicData = characteristic.value else { return -1 }
132     let byteArray = [UInt8](characteristicData)
133
134
135     let firstBitValue = byteArray[0] & 0x01
136     if firstBitValue == 0 {
137
138       return Int(byteArray[1])
139     } else {
140
141       return (Int(byteArray[1]) << 8) + Int(byteArray[2])
142     }
143   }
144 }
```

Listing B.1: Aplicação em Swift

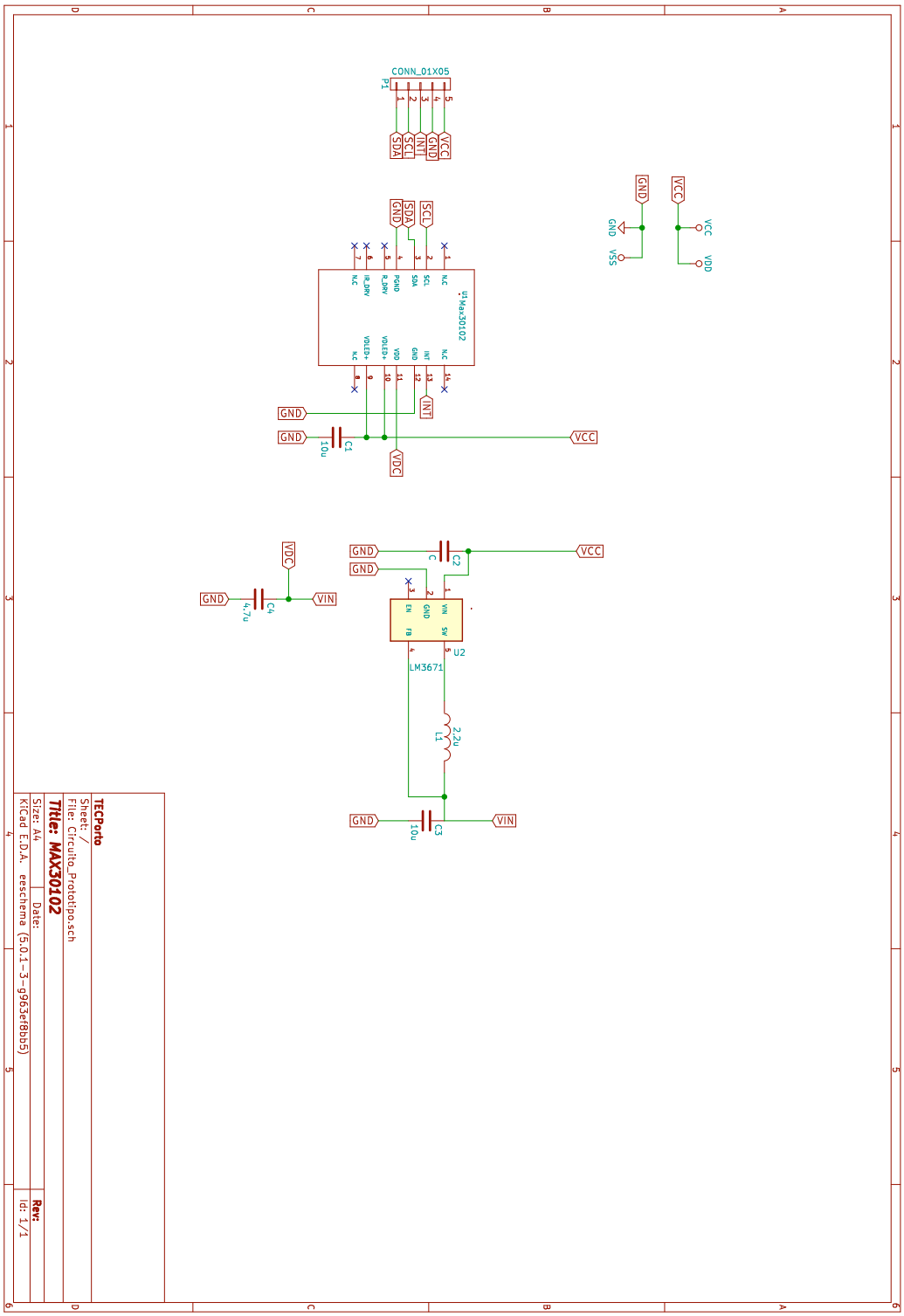



Figura B.1: Esquema do circuito do MAX30102

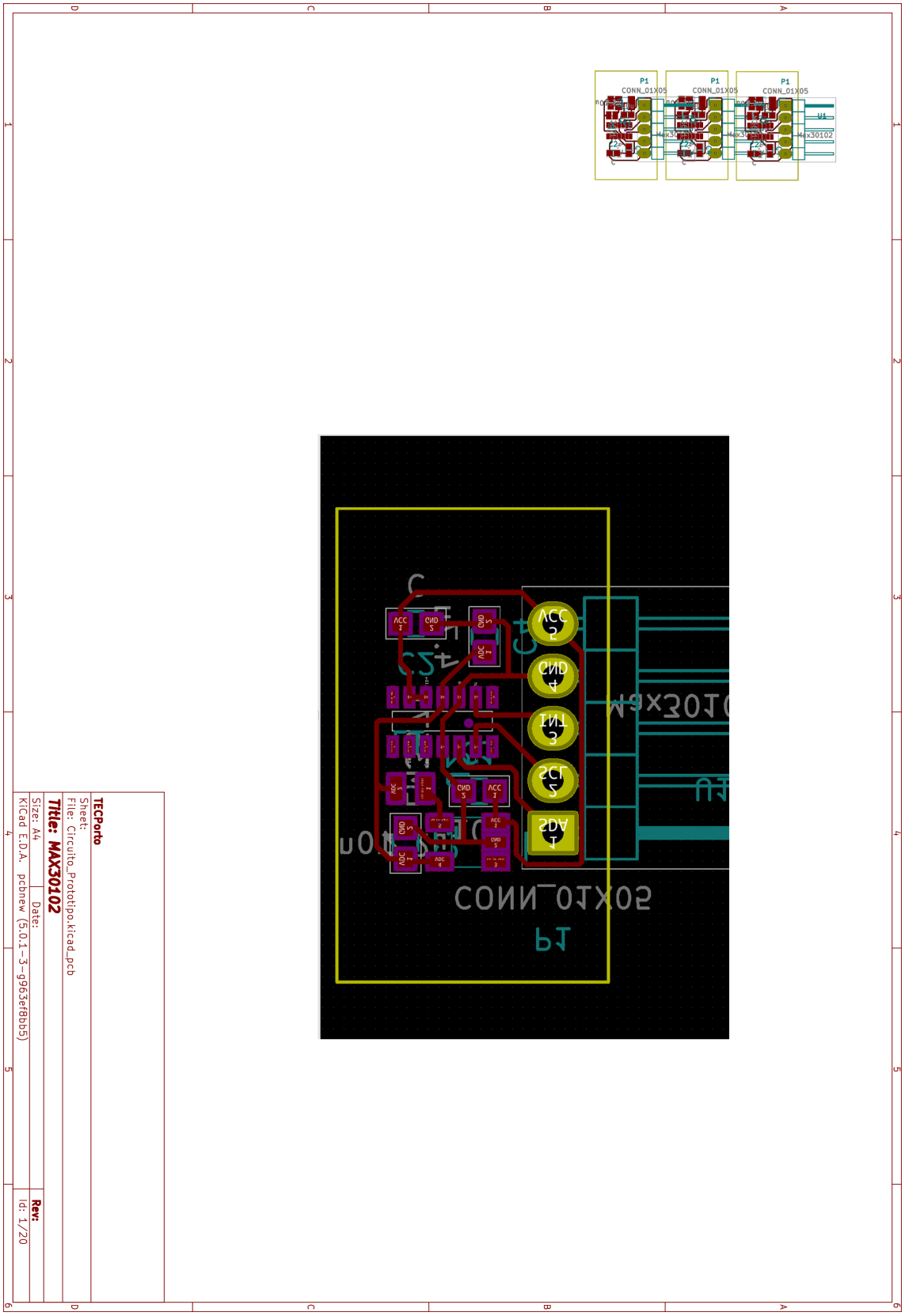



Figura B.2: Esquema da PCB do MAX30102

# Referências


Max30102 high-sensitivity pulse oximeter and heart-rate sensor for wearable health - maxim. https://www.maximintegrated.com/en/products/sensors/MAX30102.html. (Accessed on 07/26/2019).

ACC American College of Cardiology. The american journal of cardiology official journal of the american college of cardiology., 1995. URL http://www.sciencedirect.com/science/journal/00029149.

Bluetooth-SIG. Bluetooth technology website. https://www.bluetooth.com/specifications/gatt/services/. (Accessed on 06/10/2019).

Ok Kyung Cho, Yoon Ok Kim, Hiroshi Mitsumaki e Katsuhiko Kuwa. Noninvasive measurement of glucose by metabolic heat conformation method. *Clinical Chemistry*, 50(10):1894–1898, 2004. ISSN 00099147. doi: 10.1373/clinchem.2004.036954.

Espressif Systems. ESP32 Series Datasheet. 2018. ISSN 00070610.

Gale Encyclopedia of Medicine. *eletrocardiography. (n.d.) Gale Encyclopedia of Medicine.* 2008. URL https://medical-dictionary.thefreedictionary.com/electrocardiography. Último acesso em 2019/01/11.

Wei Gao, Sam Emaminejad, Hnin Yin Yin Nyein, Samyuktha Challa, Kevin Chen, Austin Peck, Hossain M Fahad, Hiroki Ota, Hiroshi Shiraki, Daisuke Kiriya, Der Hsien Lien, George A Brooks, Ronald W. Davis e Ali Javey. Fully integrated wearable sensor arrays for multiplexed in situ perspiration analysis. *Nature*, 529(7587):509–514, 2016. ISSN 14764687. doi: 10.1038/nature16521. URL http://dx.doi.org/10.1038/nature16521.

Ronald L. Gellish, Brian R. Goslin, Ronald E. Olson, Audry McDonald, Gary D. Russi e Virinder K. Moudgil. Longitudinal modeling of the relationship between age and maximal heart rate. *Medicine and Science in Sports and Exercise*, 39(5):822–829, 2007. ISSN 01959131. doi: 10.1097/mss.0b013e31803349c6.

Rasmus G Haahr. Reflectance pulse oximetry sensor for the electronic patch. *Master of Science dissertation, Dept. of Micro and Nanotechnology, Technical University of Denmark*, 2006.

Delaram Jarchi e Alexander J. Casson. Estimation of heart rate from foot worn photoplethysmography sensors during fast bike exercise. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2016-October:3155–3158, 2016. ISSN 1557170X. doi: 10.1109/EMBC.2016.7591398.

Chungkeun Lee, Hang Sik Shin e Myoungho Lee. Relations between ac-dc components and optical path length in photoplethysmography. *Journal of Biomedical Optics*, 16(7):077012, 2011. ISSN 10833668. doi: 10.1117/1.3600769.

Churchill Livingstone. *Hearth (anatomy). (n.d) Dictionary of Sport and Exercise Science and Medicine*. 2008. URL https://medical-dictionary.thefreedictionary.com/Heart+(anatomy). Último acesso em 2019/01/11.

Rui Miguel Silva Martins. Desenvolvimento de um Sensor de Fotopletismografia para Monitorização Cardíaca para aplicação no Pulso. página 90, 2010.

Michael T. Petterson BA RRT CPFT David R. Tobler BSEE Michael W. Wukitsch, BA e BSEE Jonas A. Pologe. Some Definitions in Radiometry. *Biomedical optics express*, 2(3):1452–1457, 1992. ISSN 2156-7085. doi: 10.1364/BOE.5.004053.

Mosby. *Hemoglobin (n.d) Mosby's Dental Dictionary*. ElSevier, 5 edição, 2012. ISBN 0815146310. URL https://medical-dictionary.thefreedictionary.com/Hemoglobin. Último acesso em 2019/01/20.

P H Oliveira Santos, D Veiga Teodoro, F Augusto De Andrade e J F Batista Guimarães. Proposta De Método Não Invasivo Para Detectar Variação Do Nível De Glicemia. *XXIV Congresso Brasileiro de Engenharia Biomédica*, (June):2091–2094, 2014.

Nazmus Saquib, Md Tarikul Islam Papon, Ishtiyaque Ahmad e Ashikur Rahman. Measurement of heart rate using photoplethysmography. *Proceedings of 2015 International Conference on Networking Systems and Security, NSysS 2015*, páginas 1–6, 2015. doi: 10.1109/NSysS.2015.7043525.

TECPorto. Tecporto. https://tecporto.pt/. (Accessed on 07/25/2019).