

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO
ENGENHARIA DE COMPUTAÇÃO

MATHIAS CUSTÓDIO MOREIRA
DANIEL MARTINS LEAL

**DESENVOLVIMENTO DE UM SMARTWATCH PARA PESSOAS COM
DEFICIÊNCIA VISUAL**

Goiânia, 13 de dezembro de 2019

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR
VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE
GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG**

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC nº 1204/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG):

Nome completo do autor: *Mathias Custódio Moreira, Daniel Martins Leal*

Título do trabalho: *Desenvolvimento de um Smartwatch para pessoas com deficiência visual*

2. Informações de acesso ao documento:

Concorda com a liberação total do documento SIM NÃO¹

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF do TCCG.

Mathias C. Moreira, Daniel Martins Leal
(Nome completo do autor)²

Prof. Carlos Galvão Pinheiro Júnior
Coordenador do Curso de Engenharia
de Computação - UFG
Mat nº 1329250

Ciente e de acordo:

Carlos Galvão Pinheiro J
(Nome completo do orientador)²

Data: 23 / 12 / 19

¹ Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Versão abril de 2018

² As assinaturas devem ser originais sendo assinadas no próprio documento, imagens coladas não serão aceitas.

MATHIAS CUSTÓDIO MOREIRA

DANIEL MARTINS LEAL

**DESENVOLVIMENTO DE UM SMARTWATCH PARA PESSOAS COM
DEFICIÊNCIA VISUAL**

Projeto Final de Curso submetido à Universidade Federal de Goiás como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Engenharia de Computação. Sob orientação do Professor Dr. Carlos Galvão Pinheiro Junior.

Goiânia, 13 de dezembro de 2019

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Custódio Moreira, Mathias

Desenvolvimento de um smartwatch para pessoas com deficiência visual [manuscrito] / Mathias Custódio Moreira, Daniel Martins Leal. - 2019.

45 f.

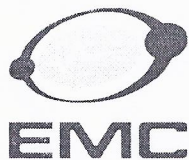
Orientador: Prof. Dr. Carlos Galvão Pinheiro Junior.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), Engenharia da Computação, Goiânia, 2019.

Inclui tabelas, lista de figuras, lista de tabelas.

1. Smartwatch. 2. Deficiência visual. 3. Arduino. 4. Acessível. I. Martins Leal, Daniel. II. Galvão Pinheiro Junior, Carlos, orient. III. Título.

CDU 621.3



ATA DE AVALIAÇÃO DE PROJETO FINAL

CURSO

() Eng. Elétrica () Eng. Mecânica (X) Eng. de Computação
() Projeto Final 1 (X) Projeto Final II

AVALIAÇÃO DE PROJETO FINAL

Título do projeto: DESENVOLVIMENTO DE UM SMARTWATCH PARA PESSOAS COM DEFICIÊNCIA VISUAL

BANCA AVALIADORA

Membro 1: CARLOS GALVÃO PINHEIRO JÚNIOR

Membro 2: FERNANDO FERREIRA DE MELO

Membro 3: GEYVERSON TEIXEIRA DE PAULA

ESTUDANTES

Matrícula	Nome
201303891	MATHIAS CUSTÓDIO MOREIRA
201303871	DANIEL MARTINS LEAL

NOTAS

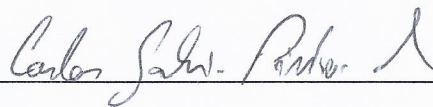
Matrícula	Membro 1				Membro 2				Membro 3				Média
	NPT	NTE	NAA	NF	NPT	NTE	NAA	NF	NPT	NTE	NAA	NF	
201303891	10,0	8,0	8,0	8,2	10,0	8,0	8,0	8,2	10,0	8,0	8,0	8,2	8,2
201303871	10,0	8,0	8,0	8,2	10,0	8,0	8,0	8,2	10,0	8,0	8,0	8,2	8,2

NPT – Nota plano de trabalho; NTE – Nota do trabalho escrito; NAA – Nota de apresentação e arguição

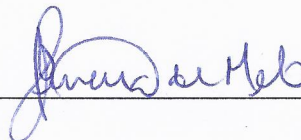
Para Eng. Elétrica, Mecânica e PFC2 da Eng. Da Computação: $NF = 0,1 \times NPT + 0,45 \times NTE + 0,45 \times NAA$

Para PFC1 da Eng. Da Computação: $NF = 0,3 \times NPT + 0,7 \times NAA$

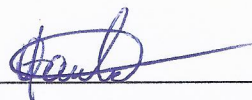
Goiânia, 13 de Dezembro de 2019.



Membro 1



Membro 2



Membro 3

AGRADECIMENTOS

Agradecemos às nossas famílias por estarem sempre do nosso lado durante toda a graduação. Agradecemos por todo o incentivo e todo o apoio que recebemos.

Agradecemos a todos os amigos que conhecemos ao longo do curso, por nos acompanharem durante nossa jornada, compartilhando as suas experiências e nos ajudando nas nossas dificuldades.

Agradecemos a todos os docentes da faculdade pelo conhecimento passado ao longo desses anos. Certamente esse conhecimento irá nos nortear por toda a nossa vida profissional.

Agradecemos ao nosso orientador, Prof. Dr. Carlos Galvão Pinheiro Junior, e também a Profa. Dra. Karina Rocha Gomes da Silva por terem estado ao nosso lado durante todo esse projeto, nos orientando e aconselhando pacientemente qual o melhor caminho a seguir.

E por fim agradecemos a todos que nos ajudaram direta ou indiretamente na realização deste trabalho.

“Quer você acredite que consiga fazer alguma coisa ou não, você está certo.”

Henry Ford

RESUMO

Este trabalho tem como objetivo o desenvolvimento de um Smartwatch voltado para pessoas com deficiência visual. O dispositivo foi implementado de forma que possa ser usado sem a necessidade da visão. Desta forma, é possível acessar seus recursos através de comandos de voz. O projeto consiste em um relógio equipado com sensores para medir a temperatura e a taxa de batimentos cardíacos e também um aplicativo para smartphone que se conecta ao relógio através de uma conexão bluetooth. A interação do usuário com o dispositivo é feita através do uso do aplicativo do smartphone usando comandos de voz. O protótipo foi construído baseado na plataforma Arduino e alguns outros dispositivos, como sensores, e o baixo custo dos componentes foi um dos requisitos. O Smartwatch desenvolvido neste projeto nos mostra que, com algumas pequenas mudanças, esta tecnologia pode se tornar acessível para um número cada vez maior de pessoas.

Palavras-chave: Smartwatch. Deficiência visual. Arduino. Acessível.

ABSTRACT

This work aims the development of a Smartwatch for people with visual impairment. The device was implemented in a way that it could be used without the need of the sight. In this way, its possible to access its resources through voice commands. The project consists of a watch fitted with sensors to measure body temperature and heart rate along with a smartphone app that connects with the watch through a bluetooth connection. The user interaction with the device is made using the phone app through voice commands. The prototype was built based on the Arduino platform and some other devices, like sensors, and the low cost of the components was another requirement. The Smartwatch developed in this project shows us that, with little changes, this technology could be accessible for an increasing number of people.

Keywords: Smartwatch. visual impairment. Arduino. Accesible.

LISTA DE ILUSTRAÇÕES

Figura 1: Pulseira que alerta sobre obstáculos.....	15
Figura 2: Relógio Dot.....	16
Figura 3: Breakout board do MLX90614.....	18
Figura 4: Sensor MAX30100 com e sem Breakout board.....	19
Figura 5: Módulo bluetooth HM10.....	20
Figura 6: Arduino Pro Mini.....	21
Figura 7: Logo da Xamarin no canto superior esquerdo.....	22
Figura 8: Aplicativo não conectado ao relógio.....	23
Figura 9: Aplicativo conectado ao relógio.....	24
Figura 10: Aplicativo aguardando um comando de voz.....	24
Figura 11: Estrutura de um projeto Xamarin.....	26
Figura 12: Arquitetura do NuGet.....	27
Figura 13: Conexão da bateria e do módulo de cargas no Arduino.....	30
Figura 14: Conexão do MAX30100 no Arduino.....	30
Figura 15: Conexão do MLX90614 no Arduino.....	31
Figura 16: Conexão do HM10 no Arduino.....	32
Figura 17: Arduino UNO.....	33
Figura 18: Esquema de ligação entre o Arduino Pro Mini e o Arduino Uno.....	33
Figura 19: Modelagem 3D da case do Smartwatch.....	35
Figura 20: Imagem do smartwatch com suas dimensões.....	35
Figura 21: Imagem do <i>smartwatch</i> pronto no pulso do usuário.....	36
Figura 22: Fluxograma da <i>Sketch</i> do Arduino.....	38
Figura 23: Mi Band.....	41
Figura 24: Termômetro digital.....	42

LISTA DE TABELAS

Tabela 1: Pessoas com deficiência visual no Brasil por região.....	12
Tabela 2: Teste das medições dos batimentos cardíacos.....	40
Tabela 3: Teste das medições da Temperatura.....	41
Tabela 4: Cotação Itens Smartwatch em dez/2019.....	44

1. INTRODUÇÃO	12
2. FUNDAMENTAÇÃO TEÓRICA	17
2.1 Termômetro MLX90614	17
2.2 Sensor de batimentos cardíacos MAX30100	18
2.3 Módulo bluetooth 4.0 HM10	19
2.4 Arduino	20
2.5 Xamarin	21
3. IMPLEMENTAÇÃO	23
3.1 Implementação do Aplicativo para Smartphone	23
3.1.1 Definição do aplicativo	23
3.1.2 Implementação	25
3.2 Implementação do relógio	28
3.2.1 Funcionamento	28
3.2.2 Montagem do Relógio	29
3.2.2.1 Montagem do Hardware	29
3.2.2.2 Desenvolvimento do software	32
3.2.2.3 Desenvolvimento da case 3D	34
3.2.2.4 Desenvolvimento do código	36
4. RESULTADOS E DISCUSSÕES	39
4.1 Comunicação entre o smartwatch e o smartphone	39
4.2 Reconhecimento dos comandos de voz	39
4.3 Medição dos batimentos cardíacos	40
4.4 Medição da temperatura corporal	41
4.5 Teste do botão de localizar celular	42
4.6 Teste da bateria e do módulo de cargas	42
4.7 Teste do leitor de textos	43
4.8 Custo total do projeto	43
5. CONCLUSÕES	45
5.1 Principais Dificuldades Encontradas	45
5.2 Trabalhos Futuros	45
6. REFERÊNCIAS	46

1. INTRODUÇÃO

Nos últimos anos, a tecnologia tem avançado quase exponencialmente, em especial as tecnologias da informação. Temos visto o desenvolvimento de computadores, smartphones e outros dispositivos com desempenho e capacidade de processamento inimagináveis há pouco tempo atrás. Porém, existe um considerável número de pessoas que possuem algum tipo de deficiência física que as impedem de usufruir dessas tecnologias.

Estima-se que a cada 5 segundos uma pessoa se torna cega no mundo (World Report on Disability 2010 e Vision 2020) e, no Brasil, existem mais de 6,5 milhões de pessoas com deficiência visual, sendo 582 mil cegas e seis milhões com baixa visão, segundo censo 2010 feito pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Só no estado de Goiás, segundo o próprio IBGE, o número de pessoas totalmente cegas chega a 14 mil. Na Tabela 1, podemos ver a quantidade de pessoas com deficiência por região do Brasil.

Pessoas com deficiência visual por região	Total	% população local
Norte	574.823	3,6
Nordeste	2.192.455	4,1
Sudeste	2.508.587	3,1
Sul	856.086	3,2
Centro-Oeste	443.357	3,2

Tabela 1. Pessoas com deficiência visual no Brasil por região

Devido a esses dados, existem leis para garantir mais igualdade e acessibilidade para pessoas com, não apenas deficiência visual, mas qualquer outro tipo de deficiência. De acordo com a Lei Brasileira de Inclusão da Pessoa com Deficiência (Lei Nº 13.146, de 6 de Julho de 2015.) temos que:

Art. 2º - Considera-se pessoa com deficiência aquela que tem impedimento de longo prazo de natureza física, mental, intelectual ou sensorial, o qual, em interação com uma ou mais

barreiras, pode obstruir sua participação plena e efetiva na sociedade em igualdade de condições com as demais pessoas.

Art. 4º - Toda pessoa com deficiência tem direito à igualdade de oportunidades com as demais pessoas e não sofrerá nenhuma espécie de discriminação.

Este trabalho tem como foco a inclusão de pessoas com deficiência visual no uso da tecnologia com o objetivo de facilitar suas vidas pois sabemos o quão difícil pode ser o dia a dia dessas pessoas. Sua dificuldade está até mesmo em atividades bastante simples, como localizar objetos pessoais ou ler um texto qualquer. Por isso é tão importante a criação de projetos que visem a acessibilidade dessas pessoas.

Com o objetivo de dar uma pequena contribuição nesse processo de inclusão social, neste trabalho é desenvolvido um *smartwatch* voltado para pessoas com deficiência visual. Os *smartwatches* são dispositivos que tem ganhado bastante popularidade nos últimos anos e consistem, basicamente, em um relógio com diversas funções úteis para o usuário, tais como monitoramento dos batimentos cardíacos e da temperatura corporal. Geralmente, esses dispositivos podem se conectar com um smartphone via conexão *bluetooth*.

Por ser voltado para pessoas com dificuldades para enxergar ou completamente cegas, o *smartwatch* desenvolvido neste projeto não contém um *display* gráfico e toda a interação é feita através de comandos de voz. Para usá-lo, basta instalar o aplicativo no *smartphone* e ligar o *smartwatch*. O aplicativo irá reconhecer o dispositivo e se conectar automaticamente. Uma vez conectado, o dispositivo ficará em modo *standby* aguardando comandos vindos do *smartphone*. Quando o usuário diz algum comando, o aplicativo irá mandar uma solicitação para o dispositivo, via conexão *bluetooth*, que por sua vez irá ativar o respectivo sensor, fazer a medição requisitada, mandar a informação de volta para o *smartphone* e depois desativar o sensor novamente. Uma vez que o aplicativo recebe a informação requisitada, essa informação será informada ao usuário através de uma voz virtual. O aplicativo também possui algumas funções extras para auxiliar seus

usuários, como o reconhecimento de texto através da câmera e um botão que, quando apertado, faz com que o *smartphone* toque e facilite a sua localização.

O projeto foi desenvolvido seguindo a metodologia *DIY (Do It Yourself)* ou Faça Você Mesmo, em tradução literal. Esta metodologia é basicamente uma forma de construção, modificação ou reparação das coisas sem a ajuda direta de especialistas ou profissionais (Wikipédia, 2019). Desta forma, qualquer pessoa poderá replicar este projeto, bastando adquirir os componentes e montá-los da forma como será descrito nos próximos capítulos. Outro requisito que foi levado em consideração foi o baixo custo. Portanto, todos os componentes utilizados podem ser facilmente encontrados à venda na internet por um preço bastante acessível, aumentando ainda mais a acessibilidade do projeto.

Durante o desenvolvimento, foram analisados outros trabalhos similares com a finalidade de se ter um melhor embasamento. Dentre todos os trabalhos analisados, serão citados dois que possuem propostas parecidas com o nosso projeto.

O primeiro trabalho a ser analisado será o do estudante Luan Henrique de Oliveira, aluno da Escola Técnica Estadual (Etec), de Rio Claro no interior paulista, orientado pelo Prof. Armando Bayeux da Silva. Em seu projeto de conclusão de curso (TCC) Luan Henrique desenvolveu uma pulseira que identifica obstáculos em ruas e calçadas e que pode ajudar pessoas com deficiência visual a ter mais autonomia na mobilidade urbana. O aparelho, que pode ser observado na Figura 1, possui um sensor ultrassônico e foi programado para alertar os usuários da presença de placas, telefones públicos, portões automáticos em movimento a até dois metros de distância, e possui quatro tipos de vibração para alertar o usuário de que está se aproximando de algum obstáculo.



Figura 1. Pulseira que alerta sobre obstáculos.

Este trabalho tem como foco principal facilitar a locomoção de usuários com pouca ou nenhuma visão e, apesar de ser uma ideia bastante interessante e útil, possui uma proposta um pouco diferente do nosso projeto. Alguns diferenciais do nosso projeto que podem ser destacados é a possibilidade de conexão com um *smartphone*, uma maior quantidade de recursos e uma maior interação com o usuário através dos comandos de voz.

Outro projeto analisado foi o relógio Dot, visto na Figura 2. Este projeto, que já está sendo comercializado, consiste em um relógio, voltado para pessoas com deficiência visual, que não possui um display para exibir suas informações. Ao invés disso, ele possui gravuras na linguagem Braille e o usuário lê as informações do relógio passando o dedo sobre a superfície do dispositivo. Essas escritas em Braille são feitas de forma dinâmica e o dispositivo possui várias funcionalidades, tais como informar desde as horas e a data até notificações do celular, como mensagens de aplicativos e quem está ligando. É possível ainda rejeitar chamadas telefônicas apenas apertando um botão do relógio.



Figura 2. Relógio Dot.

Os diferenciais do nosso projeto em relação ao projeto Dot está na maior quantidade de recursos que o nosso *smartwatch* possui e na interação com o usuário que é feita por voz, o que torna a nosso projeto acessível para qualquer pessoa sem a necessidade de conhecer a linguagem braille. No entanto, o projeto deles permite a uma pessoa, que entende a linguagem Braille, se sentir mais confortável com a leitura.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Termômetro MLX90614

Um dos recursos presentes no smartwatch é um sensor de temperatura, que serve para medir a temperatura corporal do usuário e enviar esta informação ao *smartphone* quando solicitado.

Dentre todos os sensores analisados, o que mais atendeu aos requisitos do projeto foi o MLX90614 (Figura 3) pois possui um custo relativamente baixo, mede a temperatura sem a necessidade de contato, é um instrumento de fácil utilização e que já vem calibrado de fábrica com uma boa precisão.

O MLX90614 é um termômetro infravermelho para medidas de temperatura sem contato. Tanto o *chip* detector contendo uma termopilha sensível à infravermelho quanto o condicionador de sinal *ASIC* estão integrados no mesmo encapsulamento TO-39. Integrado no MLX90614 estão um amplificador de baixo ruído, um ADC de 17 bits e uma poderosa unidade *DSP* portanto atingindo uma alta acurácia e resolução do termômetro (MELEXIS).

O sensor vem calibrado de fábrica com uma faixa de temperatura de -40°C à 125°C para medição de temperatura ambiente e -70°C à 380°C para medição de temperatura em objetos. Possui uma alta acurácia de $0,5^{\circ}\text{C}$ e uma resolução de $0,02^{\circ}\text{C}$ (MELEXIS).

A escolha do MLX90614 para o projeto se deu devido a seu baixo custo e a sua facilidade de utilização. A comunicação com o sensor pode ser feita via interface I²C e não há necessidade de se calibrar o sensor pois o mesmo já vem calibrado de fábrica.

No protótipo, o sensor está contido em uma *breakout board* que, basicamente, consiste em uma placa de circuito impresso que contém, além do sensor, alguns componentes necessários para a sua utilização, como resistores de *pull-up* por exemplo.



Figura 3. *Breakout board* do MLX90614.

2.2 Sensor de batimentos cardíacos MAX30100

Outro recurso presente no smartwatch é a capacidade de se medir os batimentos cardíacos do usuário. Alguns modelos de sensores para este propósito foram analisados, e o que mais atendeu aos requisitos do projeto foi o módulo MAX30100 (Figura 4) devido a sua facilidade de uso, boa precisão e custo relativamente baixo. Existem opções de sensores com menor custo, porém com precisão muito inferior.

O MAX30100 é um sensor de taxa cardíaca e oxímetro. Ele combina dois leds, um fotodetector e um processador de sinal analógico de baixo ruído para detectar a taxa de oxigenação do sangue e a taxa de batimentos cardíacos. O sensor opera de 1.8VDC até 3.3VDC e pode ser colocado em *standby*, consumindo baixa corrente, via *software* (Maxim Integrated).

A comunicação com o sensor, assim como no MLX90614, pode ser feita via interface I²C o que facilita muito o seu uso e economiza portas do microcontrolador já que se pode conectar vários sensores ao mesmo tempo nas mesmas portas I²C.

Assim como no MLX90614, o MAX30100 também está contido em uma *breakout board*.



Figura 4. Sensor MAX30100 com e sem Breakout board.

2.3 Módulo *bluetooth* 4.0 HM10

A comunicação entre o relógio e o *smartphone* será feita via conexão *bluetooth*. A tecnologia *bluetooth* já é relativamente antiga e já foram lançadas algumas versões desde a sua criação. Ao procurar um dispositivo para fazer a comunicação, um dos requisitos foi a de que este dispositivo implementasse a versão 4.0 da tecnologia *bluetooth* pois se trata de uma versão mais recente e otimizada para um menor consumo de energia. Dentre as opções analisadas, o módulo escolhido para o projeto foi o HM10 (Figura 5) devido ao seu baixo custo e fácil utilização.

O HM10 é um módulo *bluetooth* 4.0 (*Low Energy*) baseado no chipset CC2541. Ele suporta operar como *MASTER* ou como *SLAVE*, bastando apenas configurá-lo por comandos AT. É possível fazer uma comunicação entre dois módulos *bluetooth* (sendo um *MASTER* e o outro *SLAVE*) e também se comunicar entre um módulo e um notebook, PDA, celular, dentre outros. Sua tensão de operação varia entre 2.7VDC à 3.3VDC e possui um alcance de até 10 metros (Eletrogate).

O *bluetooth* 4.0, dentre várias vantagens, possui um maior alcance e um menor consumo de bateria. A comunicação com o HM10 pode ser feita através das portas seriais do microcontrolador (Rx e Tx)



Figura 5. Módulo *bluetooth* HM10.

2.4 Arduino

O relógio será desenvolvido baseado na plataforma Arduino, sendo assim, todos os sensores e módulos do dispositivo estão conectados e são controlados por essa plataforma. A escolha do Arduino se deu pela sua facilidade de utilização, pelo fato de possuir muita documentação na internet e o seu baixo custo.

O Arduino é uma plataforma eletrônica *open-source* baseada em *hardware* e *software* de fácil utilização. Placas Arduino são capazes de ler entradas - luz em um sensor, o dedo em um botão ou uma mensagem no Twitter - e transforma isso em uma saída - ativando um motor, ligando um *LED* ou publicando algo online. Você pode “dizer” a sua placa o que fazer através do envio de um conjunto de instruções para o microcontrolador da placa. Para fazer isso você usa a linguagem de programação do Arduino e o *Software* do Arduino (*IDE*) (Arduino).

Existem várias versões de Arduino que podem ser usados nos mais diferentes projetos e, neste projeto, é utilizado a versão Arduino Pro Mini (Figura 6). O Arduino Pro Mini é uma versão do Arduino com dimensões reduzidas o que é perfeito para o projeto de um relógio. Esta versão da placa possui uma maior limitação em relação aos seus recursos se comparada a outras versões de Arduino, mas é suficiente para suprir as necessidades do projeto.



Figura 6. Arduino Pro Mini.

2.5 Xamarin

Para desenvolver o aplicativo do smartphone, foi escolhida a plataforma de desenvolvimento Xamarin. A Xamarin é uma plataforma de desenvolvimento de *software* constituída de ferramentas, linguagens de programação e bibliotecas para desenvolver variados tipos de aplicações. Foi desenvolvida baseada na plataforma .NET e possui ferramentas e bibliotecas para desenvolver aplicativos para Android, iOS, tvOS, watchOS, macOS e Windows (Xamarin).

A plataforma Xamarin é bastante utilizada para o desenvolvimento de aplicativos mobile devido a sua arquitetura multi-plataforma. Com apenas uma base de código (geralmente escrita em C#) é possível desenvolver aplicativos para vários sistemas operacionais diferentes, em especial Android e iOS.

A escolha da Xamarin para o projeto se deve ao fato de sua fácil utilização e pequena curva de aprendizagem pois é muito semelhante ao desenvolvimento de aplicações Desktop com Java ou C# por exemplo. Além disso, com a utilização da plataforma, é possível desenvolver aplicativos tanto pra Android quanto para iOS com um único código devido a sua característica multi-plataforma. O logo da Xamarin pode ser visto na figura 7.

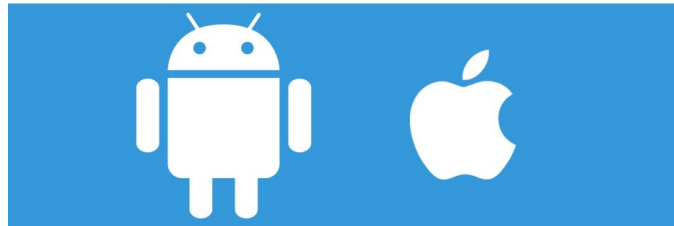


Figura 7. Logo da Xamarin no canto superior esquerdo.

3. IMPLEMENTAÇÃO

Neste capítulo serão apresentadas todas as etapas de desenvolvimento do *smartwatch* e do aplicativo para *smartphone* e o processo de comunicação entre eles. O protótipo desenvolvido consiste em um relógio provido com alguns sensores para monitorar os sinais vitais do usuário. Os sensores em questão são um sensor de temperatura e um sensor de batimentos cardíacos. Devido ao fato do projeto ser voltado para pessoas com deficiência visual, o protótipo não possui um *display* gráfico, sendo assim, toda a interação do usuário com o dispositivo é feita por um botão contido no relógio e através de comandos de voz no aplicativo.

3.1 Implementação do Aplicativo para Smartphone

3.1.1 Definição do aplicativo

O aplicativo para *smartphone* desenvolvido neste projeto foi criado pensando nos usuários que possuem deficiência visual, portanto, não possui uma interface gráfica para navegação e a sua utilização é feita exclusivamente através de comandos de voz. Ao abrirmos o aplicativo nos deparamos com uma tela branca com apenas uma notificação indicando o status de conexão com o relógio. A conexão com o relógio é feita de forma automática quando o aplicativo o localiza. A interface do aplicativo pode observada nas figuras 8 e 9.

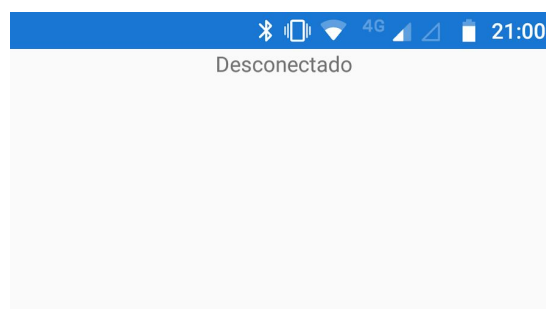


Figura 8. Aplicativo não conectado ao relógio.

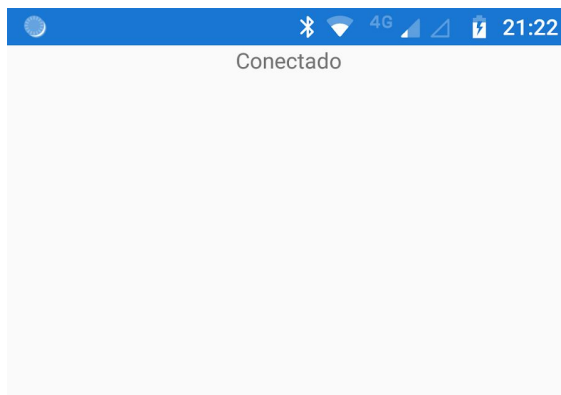


Figura 9. Aplicativo conectado ao relógio.

Para iniciar o uso é necessário tocar na tela e, ao fazer isso, ocorrerá um aviso sonoro e o aplicativo ficará esperando por um comando de voz.

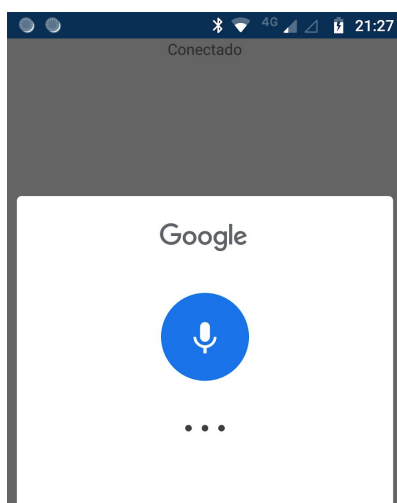


Figura 10. Aplicativo aguardando um comando de voz.

Para fazer o reconhecimento da voz do usuário, e convertê-la em texto que será interpretado pelo aplicativo, é utilizado a *API* padrão do google para reconhecimento de voz. Essa *API* processa o áudio com a voz do usuário na nuvem, portanto, é necessário conexão com a internet para realizar o processo.

Após falar o comando, irá ocorrer um segundo aviso sonoro indicando que a voz do usuário foi reconhecida e que será agora interpretada. Os comandos disponíveis no aplicativo são:

- Qual a minha temperatura?
- Qual o meu batimento cardíaco?
- Quantas horas?
- Qual a data de hoje?
- Ler texto.

Os dois primeiros comandos dependem da conexão com o relógio, portanto, se o aplicativo não estiver conectado ao relógio, a informação solicitada não poderá ser informada e o usuário será notificado sobre a falha na conexão.

Ao utilizar o comando “Ler texto”, será aberta a câmera do celular onde o usuário poderá capturar uma imagem. Após a captura, o aplicativo irá fazer o reconhecimento de qualquer texto contido na imagem e falará, em voz, para o usuário.

3.1.2 Implementação

Para a implementação do aplicativo é utilizada a plataforma de desenvolvimento Xamarin e, para utilizá-la, é necessário a instalação da IDE Visual Studio. A versão do Visual Studio utilizada é a *Community* que é gratuita e pode ser obtida na página oficial da Microsoft.

A versão *Community* do Visual Studio, apesar de gratuita, possui todos os recursos necessários para o desenvolvimento deste aplicativo, incluindo a plataforma Xamarin.

Após a instalação do Visual Studio e do Xamarin, cria-se um novo projeto de aplicativo mobile. A estrutura inicial do projeto, que pode ser observada na Figura 11, é bem definida, incluindo um projeto separado para Android e iOS, com configurações específicas de cada plataforma, e um projeto principal com códigos e configurações comuns a ambas.

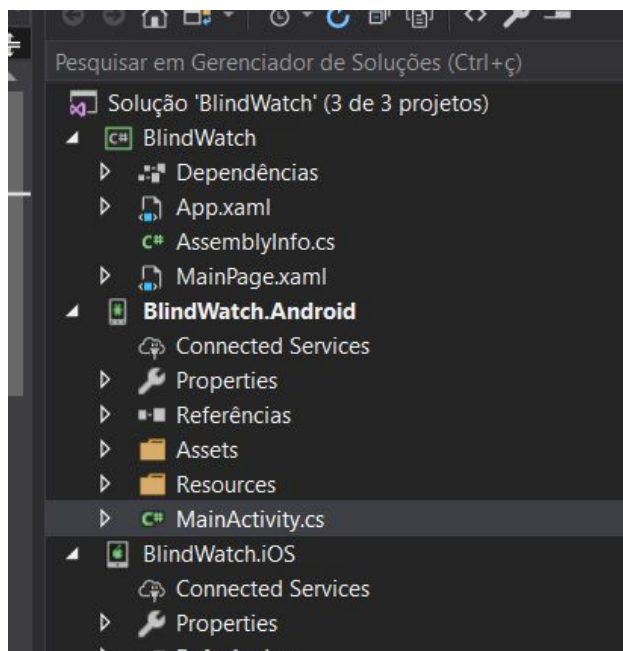


Figura 11. Estrutura de um projeto Xamarin.

Todas as telas de uma aplicação Xamarin ficam em arquivos .xaml e a tela principal do aplicativo está contida no arquivo MainPage.xaml. A nossa aplicação não contém interface gráfica, portanto só será necessária uma tela onde ocorrerá a interação com os comandos de voz.

Uma ferramenta essencial para qualquer plataforma de desenvolvimento moderna é um mecanismo por meio do qual os desenvolvedores podem criar, compartilhar e consumir código útil. Geralmente, esse código é fornecido em "pacotes" que contêm código compilado (como DLLs).

No Visual Studio, o mecanismo com suporte da Microsoft para compartilhamento de código é o NuGet, que define como os pacotes são criados, hospedados e consumidos. A arquitetura do NuGet pode ser vista na Figura 12.

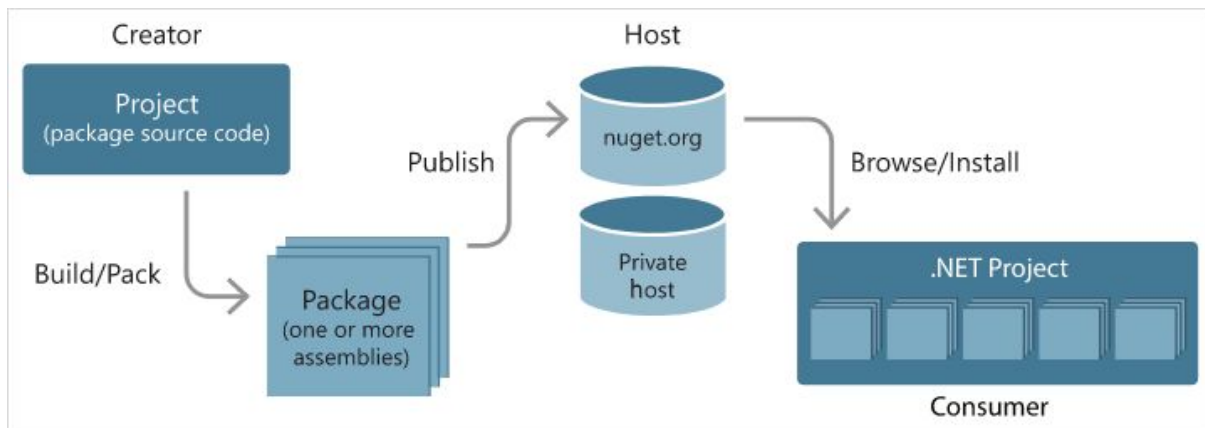


Figura 12. Arquitetura do NuGet.

fonte: docs.microsoft

No nosso projeto são utilizados alguns pacotes do Nuget, com isso podemos aproveitar algumas funcionalidades que já foram implementadas por outros desenvolvedores. Um desses pacotes é o Plugin.BLE que contém uma série de métodos já implementados para o gerenciamento de conexões bluetooth e será utilizado para manipular o bluetooth do smartphone e fazer a comunicação com o relógio.

Outro pacote utilizado é o GooglePlayServices.Vision que contém uma série de métodos implementados pela Google para manipulação de imagens como reconhecimento de texto, detecção de objetos, reconhecimento de rosto dentre outras funcionalidades e será utilizado para implementar o reconhecimento de texto em imagem que é um dos recursos do relógio.

Para se utilizar os recursos do *smartphone*, como a câmera e o *bluetooth*, é necessário que o aplicativo tenha as permissões necessárias. O código para solicitar essas permissões para o usuário pode ser encontrado em um pacote NuGet chamado Plugin.Permissions. Neste pacote temos métodos já implementados para verificar se uma determinada permissão está concedida ou solicitar a permissão ao usuário.

3.2 Implementação do relógio

3.2.1 Funcionamento

O relógio desenvolvido neste projeto consiste em um dispositivo eletrônico baseado na plataforma Arduino. A versão do Arduino utilizada foi a Pro Micro que possui dimensões bastante reduzidas e pode, inclusive, ser utilizada em projetos finais e não apenas em protótipos.

Além do Arduino, foram utilizados um sensor de temperatura (MLX90614), um sensor de batimentos cardíacos (MAX30100), um módulo *bluetooth* 4.0 (HM10), uma bateria LiPo de 100mAh e o módulo carregador de bateria TP4056.

O dispositivo foi projetado para funcionar sob demanda, o que significa que os sensores ficam desativados e apenas são ativados no momento em que o usuário solicita algum comando, como obter temperatura corporal ou batimentos cardíacos. Esta abordagem foi necessária para a economia da bateria pois, devido ao baixo custo, a mesma não possui uma boa autonomia.

O dispositivo, na maior parte do tempo, fica em modo *standby* na espera de comandos vindos do celular. Durante esta espera apenas o módulo *bluetooth* e o próprio Arduino ficam em funcionamento. Os comandos que são recebidos do celular podem ser os seguintes:

- Obter batimentos cardíacos
- Obter temperatura corporal
- Obter nível da bateria

Ao receber um comando solicitando os batimentos cardíacos ou temperatura corporal, o relógio ativa o respectivo sensor, faz a medição, retorna o valor e desativa o sensor novamente. O relógio também pode enviar a porcentagem da bateria para que o usuário saiba o momento de recarregar.

Um outro recurso que está presente no relógio é um botão que, quando acionado, envia um sinal para o *smartphone* fazendo com que o mesmo emita um som de alerta. O objetivo deste recurso é que o usuário, com problemas de visão,

possa localizar com mais facilidade o seu celular. O recurso está limitado ao alcance da conexão *bluetooth* do módulo utilizado que é de aproximadamente 10 metros.

3.2.2 Montagem do Relógio

3.2.2.1 Montagem do Hardware

A montagem do *hardware* do relógio começa pelo Arduino Pro Micro que será o cérebro do dispositivo. Nele, todos os demais módulos e sensores estão conectados.

A alimentação do Arduino será feita através do uso de uma bateria LiPo de 100 mAh. A faixa de operação da bateria utilizada está entre 3.6V à 4.2V não podendo ser carregada acima ou abaixo desta faixa sob risco de danos. Quando a tensão da bateria estiver em 4.2V a mesma estará completamente carregada e, portanto, a sua porcentagem será 100% e quando sua tensão atinge os 3.6V a porcentagem será considerada 0% sendo necessária a recarga. Para calcular a porcentagem da bateria para valores entre 0 e 100% será feita uma simples regra de três associando 4.2V a 100% e 3.6 a 0%, desta forma podemos obter qualquer valor dentro deste intervalo.

A carga da bateria será feita com o uso de um módulo de carregamento de baterias. O módulo utilizado será o TP4056 que basicamente se trata de um dispositivo que contém um CI para controle de cargas. O módulo, convenientemente, possui uma entrada micro-usb que pode ser utilizada para conectar uma fonte externa de maneira mais prática. A bateria e o módulo TP4056 podem ser vistos na Figura 13.

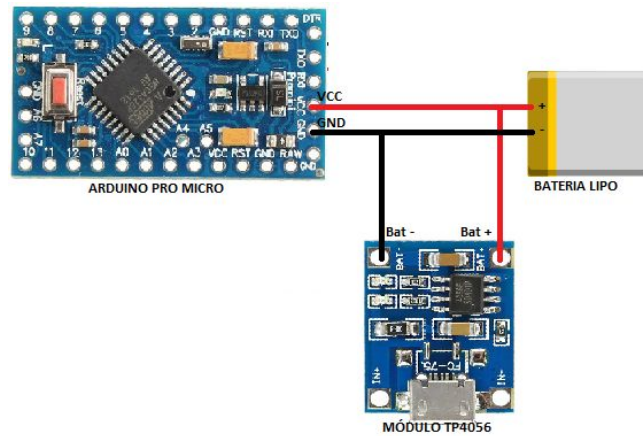


Figura 13. Conexão da bateria e do módulo de cargas no Arduino.

O sensor de batimentos cardíacos utilizado no projeto é o MAX30100. Este sensor possui um *led* infravermelho que detecta a variação da oxigenação do sangue para então calcular a taxa de batimentos cardíacos.

A comunicação do MAX30100 com o Arduino será feita pela interface I²C. Uma das vantagens do protocolo I²C é a possibilidade de se conectar vários dispositivos (*slaves*) em um microcontrolador (*master*) utilizando apenas duas portas. No Arduino essas portas são a A4 (*SDA*) e a A5 (*SCL*) onde *SDA* (*Serial Data*) é a porta de dados e *SCL* (*Serial Clock*) é a porta do sinal de *clock* que é gerado pelo *master*. Na Figura 14 podemos ver a conexão do MAX30100 com o Arduino.

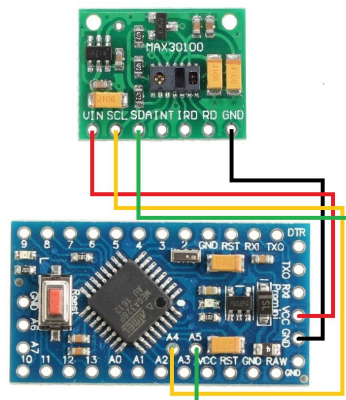


Figura 14. Conexão do MAX30100 no Arduino.

Já o sensor de temperatura utilizado no projeto é o MLX90614. Ele possui um sensor infravermelho capaz de medir a temperatura de objetos que estão à sua frente sem a necessidade de contato.

A comunicação do MLX90614 com o Arduino, assim como no sensor de batimentos cardíacos, também será feita pela interface I²C. No protocolo I²C cada dispositivo *slave* recebe um endereço distinto que serve para distingui-lo dos demais. Ambos os sensores do relógio, estão conectados nas portas A4 e A5 do Arduino e cada um possui um endereço distinto dentro do protocolo. Na Figura 15 podemos ver a conexão entre o MLX90614 e o Arduino.

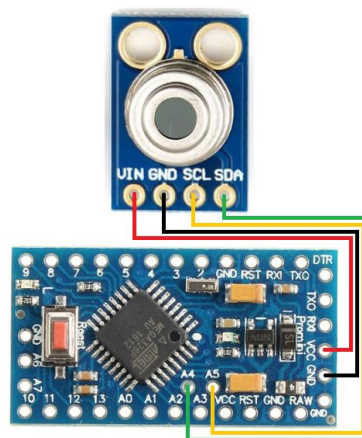


Figura 15. Conexão do MLX90614 no Arduino.

A comunicação entre o relógio e o *smartphone* é feita via conexão *bluetooth* com a utilização do módulo HM10. Este módulo é compatível com a versão 4.0 do *bluetooth* que possui algumas vantagens em relação às versões anteriores como baixo consumo de energia e maior alcance de comunicação.

O pareamento entre o módulo HM10 e o *smartphone* é feito de forma independente do Arduino, portanto, basta o módulo estar ligado que automaticamente irá aparecer na lista de dispositivos *bluetooth* do celular.

O aplicativo do *smartwatch* fica monitorando a conexão com o módulo *bluetooth* constantemente e, ao encontrar o dispositivo, inicia o pareamento de forma automática. O aplicativo também detecta automaticamente quando o módulo é desconectado, seja por ter sido desligado ou estar fora de alcance.

A conexão do módulo com o Arduino é feita através da comunicação serial. No Arduino, a comunicação serial é feita pelas portas digitais 0(Rx) e 1(Tx) que se conectam nas portas seriais do módulo de forma alternada, ou seja, a porta Tx do módulo é conectada na porta Rx do Arduino e a porta Rx do módulo é conectada na porta Tx do Arduino. Na Figura 16, podemos ver a conexão entre o HM10 e o Arduino.

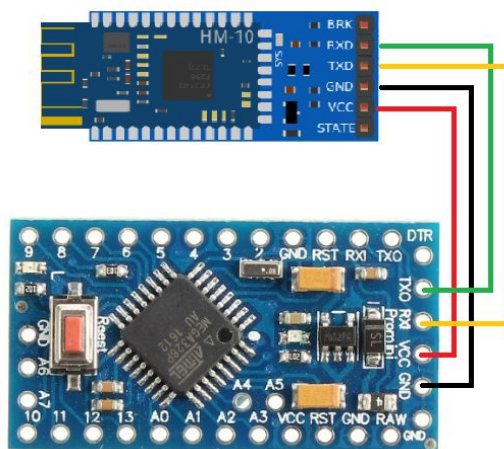


Figura 16. Conexão do HM10 no Arduino.

3.2.2.2 Desenvolvimento do *software*

Neste tópico é explicado o processo de desenvolvimento do código que roda no dispositivo. O código foi desenvolvido utilizando a IDE do Arduino e os testes foram feitos em um Arduino UNO (Figura 17). A utilização do Arduino UNO foi necessária pois, durante o desenvolvimento, o código foi compilado e gravado diversas vezes e este processo seria extremamente trabalhoso em um Arduino Pro Mini já que este não possui uma interface de gravação embutida na placa.

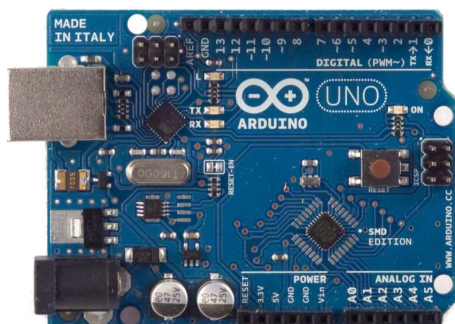


Figura 17. Arduino UNO.

Para gravar o código em um Arduino Pro Mini é necessário a utilização de um gravador externo ou até mesmo a utilização de outro Arduino. Neste projeto a gravação do Arduino Pro Mini é feita com o auxílio de um Arduino UNO seguindo o processo a seguir.

Retira-se o microcontrolador Atmega328P da placa Arduino UNO e conecta-se os dois Arduinos seguindo o esquema da Figura 18:

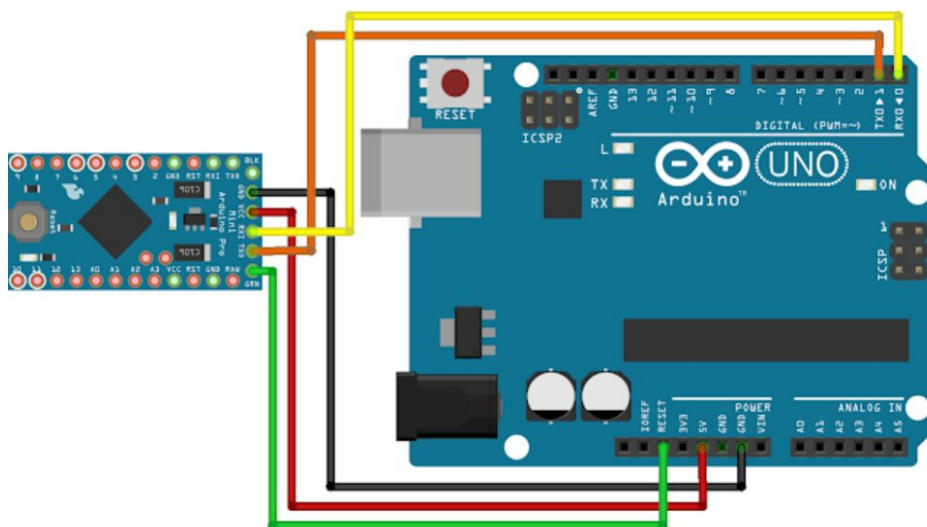


Figura 18. Esquema de ligação entre o Arduino Pro Mini e o Arduino Uno.

De acordo com a Figura 18, a ligação entre os dois Arduinos deve ser feita seguindo o seguinte esquema:

Arduino UNO	Arduino Pro Mini
5V -----	VCC
GND -----	GND
RESET -----	DTR
TX -----	TX
RX -----	RX

Após feita a ligação basta conectar o Arduino UNO ao computador pelo cabo USB AB, seleccionar a porta COM na IDE e seleccionar o modelo do Arduino, que neste caso será o Arduino Pro Mini.

Por ser um processo bastante trabalhoso, a gravação no Arduino Pro Mini só ocorreu em testes finais depois de vários testes já terem sido realizados no Arduino UNO.

3.2.2.3 Desenvolvimento da case 3D

Os componentes do *smartwatch* foram alojados em uma *case* impressa em uma impressora 3D. O projeto da *case* foi feito utilizando o *software* SketchUp 2019. O SketchUp é um *software* próprio para a criação de modelos em 3D no computador. Foi originalmente desenvolvido pela *At Last Software*, uma empresa estadunidense com sede em Boulder, Colorado, a qual foi adquirida pela Google, como anunciado a 14 de Março de 2006 (Wikipédia).

A *case* foi projetada de forma a acomodar bem todos os componentes e ter a menor dimensão possível. Depois da modelagem 3D, o projeto foi exportado para o formato .stl para enfim ser impresso na impressora. O projeto da *case* no SketchUp pode ser visto no Figura 19.

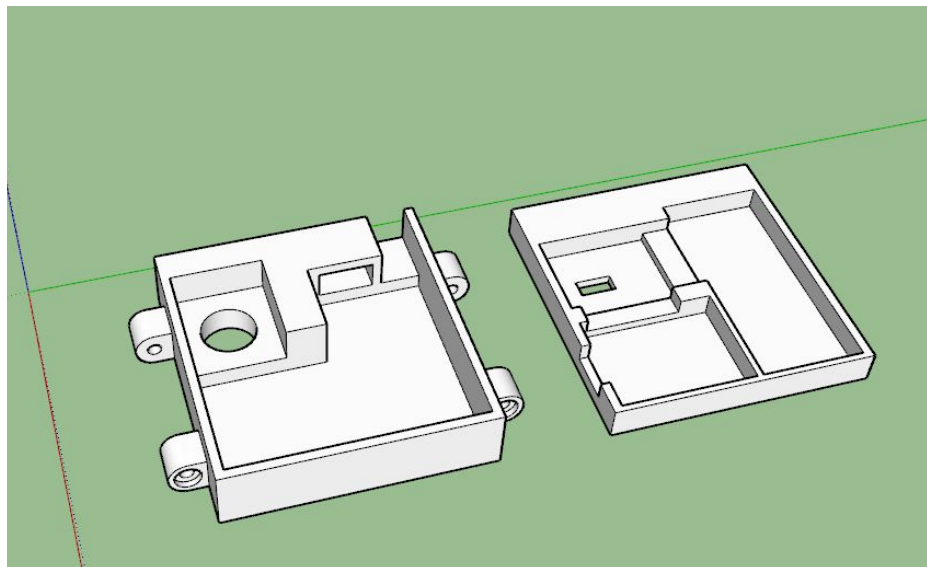


Figura 19. Modelagem 3D da case do Smartwatch.

No projeto, pode ser observado a presença de um encaixe para a fixação de uma alça para o relógio. Após concluída a impressão, fixadas as alças e alojado o circuito na case, o resultado final da montagem e as dimensões do projeto podem ser vistos nas figuras 20 e 21.

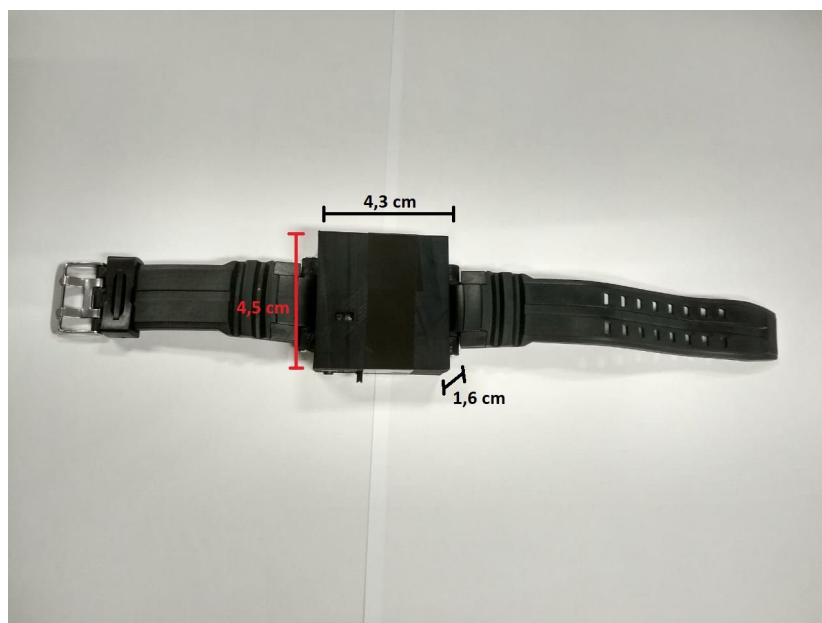


Figura 20. Imagem do *smartwatch* com suas dimensões.

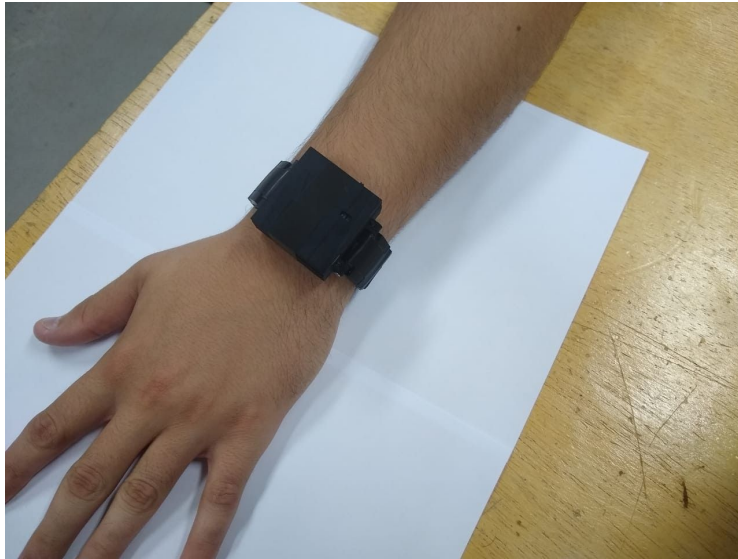


Figura 21. Imagem do *smartwatch* pronto no pulso do usuário.

3.2.2.4 Desenvolvimento do código

Um código desenvolvido para o Arduino é denominado de Sketch e basicamente é dividido em 4 partes:

- Declaração de bibliotecas
- Declaração de variáveis globais e constantes
- Setup
- Loop

Uma biblioteca é um conjunto de funcionalidades já implementadas por terceiros que podem ser reutilizadas em nossos projetos. Por exemplo, para utilizar o sensor de batimentos cardíacos seria necessário a implementação de um código bastante complexo para a comunicação com o sensor, porém, felizmente, já existe uma biblioteca com toda essa complexidade implementada sendo necessário apenas declará-la e utilizar seus métodos.

Em nosso projeto são declaradas duas bibliotecas, uma para o sensor de batimentos cardíacos e outra para o sensor de temperatura e o nosso código se inicia com essas declarações.

Após a declaração das bibliotecas, é feita a declaração das variáveis e das constantes. No projeto, são declaradas duas variáveis representando os sensores e uma constante representando a porta que será utilizada para conectar um botão. Este botão, quando acionado, enviará um sinal para o *smartphone* que fará com que o mesmo emita um sinal sonoro com o objetivo de facilitar a sua localização.

Depois de declarar as bibliotecas, variáveis e constantes é declarado o *Setup*. O *Setup* é um método padrão de uma *Sketch* para Arduino e é o primeiro método a ser executado quando o Arduino inicia seu funcionamento e serve para fazer as configurações iniciais do programa.

No nosso projeto, o *Setup* é utilizado para iniciar a comunicação serial e para configurar a porta que será utilizado para ler o estado do botão como *INPUT*.

Por fim é declarado o *Loop*. O *Loop* é outro método padrão de uma *Sketch* para Arduino e ele é executado logo após a execução do *Setup*. Porém, enquanto o *Setup* é executado apenas uma vez, o *Loop* é executado infinitas vezes em sequência durante todo o funcionamento do dispositivo.

No nosso projeto, o *Loop* é utilizado para constantemente ler as informações vindas do módulo *bluetooth*, que seriam os comandos do *smartphone*, e também verificar o estado do botão para localizar o celular. Quando é encontrado um dado na entrada serial, o dispositivo inicia o comando solicitado, que pode ser ativar um dos sensores e fazer a respectiva leitura para mandar de volta à informação para o *smartphone* ou enviar a informação do nível da bateria.

Quando o dispositivo recebe um comando para ler a temperatura ou batimentos cardíacos, o respectivo sensor é ativado, realiza a leitura, envia a informação para o *smartphone* e o sensor é desativado novamente.

Todo o funcionamento do código da *Sketch* pode ser observado no fluxograma da Figura 22 abaixo.

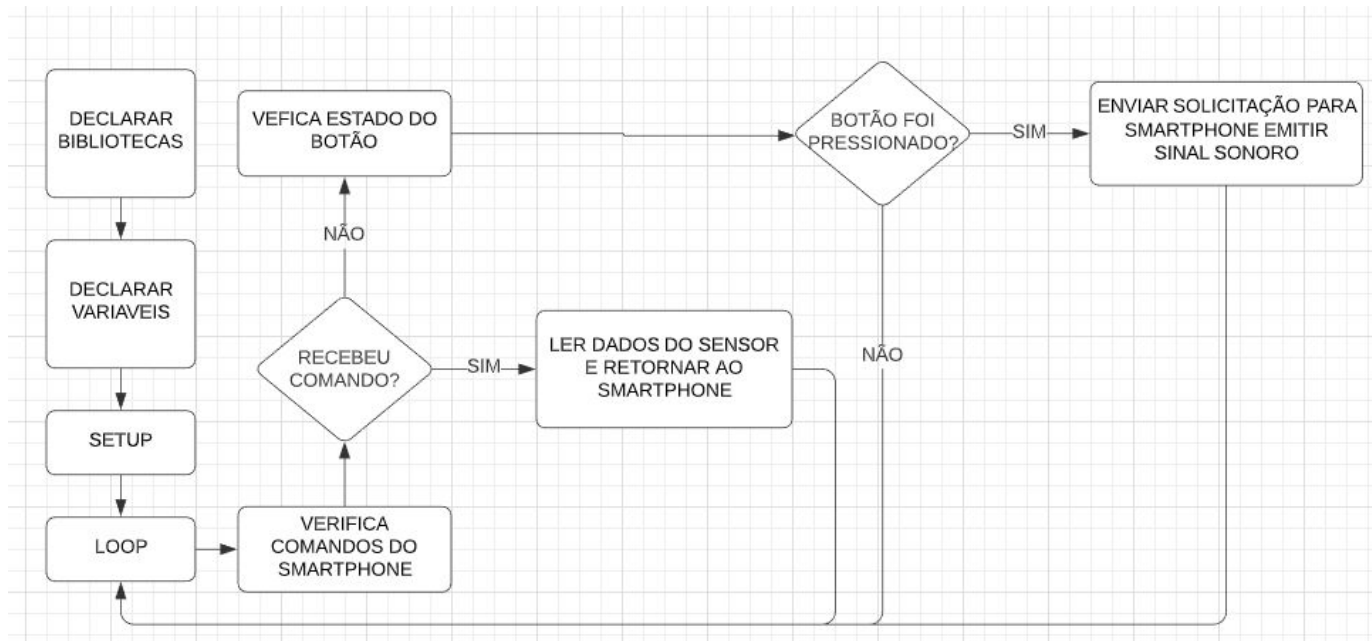


Figura 22. Fluxograma da *Sketch* do Arduino.

4. RESULTADOS E DISCUSSÕES

Após o término do desenvolvimento do protótipo, foi realizado uma série de testes para identificar possíveis falhas e também definir algumas melhorias a serem implementadas.

Foram testadas todas as funcionalidades do projeto desde o *smartwatch* até o aplicativo do *smartphone* e também a comunicação entre os dois. Os testes foram feitos em várias etapas, e serão descritos nas seções a seguir.

4.1 Comunicação entre o smartwatch e o smartphone

O primeiro teste a ser feito foi para verificar a conexão entre o *smartwatch* e o *smartphone*. De acordo com a definição do projeto, esta conexão deve ser feita de forma automática. O aplicativo, constantemente, monitora a presença do *smartwatch* e, ao localizá-lo, inicia a conexão. Quando o *smartwatch* é desligado ou fica fora de alcance, o aplicativo também é capaz de identificar isso automaticamente.

Para fazer o teste, primeiro foi aberto o aplicativo e o mesmo estava com status “Desconectado”. Em seguida o *smartwatch* foi ligado e colocado próximo ao *smartphone*. Instantes depois o aplicativo reconheceu o *smartwatch* e iniciou a conexão automaticamente, como estava previsto, e o *status* mudou para “Conectado”. Depois, o *smartwatch* foi desligado e, conseqüentemente, o *status* do aplicativo mudou para “Desconectado” novamente.

O mesmo teste foi repetido com o *smartwatch* em distâncias cada vez maiores e foi constatado que a conexão com o *smartphone* pode ocorrer a uma distância máxima de aproximadamente 12 metros.

4.2 Reconhecimento dos comandos de voz

Após o teste de conexão entre o *smartwatch* e o *smartphone*, o próximo passo foi testar os comandos de voz do aplicativo. De acordo com o projeto, para que os comandos de voz possam ser interpretados, é necessária uma conexão com

a internet pois o processamento do áudio é feito no servidor da Google, que é quem fornece a API utilizada no aplicativo.

Quando o *smartphone* está sem conexão com a internet e é dito algum comando, a aplicativo informa ao usuário, por voz, da falta de conexão. Já com o *smartphone* conectado à internet, todos os comandos foram reconhecidos pela API sem maiores problemas.

4.3 Medição dos batimentos cardíacos

Para testar os batimentos cardíacos, os resultados foram comparados com os de um outro *smartwatch* já comercializado e muito conhecido: O Mi Band (Figura 23). O Mi Band é um *smartwatch* fabricado pela chinesa Xiaomi e que, dentre outros recursos, também possui um medidor de batimentos cardíacos.

Para iniciar o teste, primeiro foi feito a conexão entre o *smartphone* e o *smartwatch* e, depois, foi verificada a conexão do *smartphone* com a internet. Logo após foi dito o comando: “Batimentos Cardíacos”. O *smartwatch* prontamente retornou o resultado que foi informado, em voz, pelo aplicativo.

O teste foi repetido diversas vezes em duas pessoas diferentes e os resultados foram colocados na Tabela 2 a seguir.

Mi Band	Smartwatch	Mi Band	Smartwatch
90 Bpm	87 Bpm	95 Bpm	93 Bpm
88 Bpm	84 Bpm	92 Bpm	96 Bpm
84 Bpm	89 Bpm	94 Bpm	92 Bpm
92 Bpm	96 Bpm	91 Bpm	87 Bpm
87 Bpm	83 Bpm	89 Bpm	93 Bpm
93 Bpm	89 Bpm	92 Bpm	94 Bpm
Pessoa 1		Pessoa 2	

Tabela 2. Teste das medições dos batimentos cardíacos.



Figura 23. Mi Band.

4.4 Medição da temperatura corporal

Para testar a medição da temperatura, os resultados foram comparados com as medições de um termômetro digital comercial. O termômetro em questão é semelhante ao visto na Figura 24.

Para iniciar o teste, primeiro foi feita a conexão entre o *smartphone* e o *smartwatch* e, depois, foi verificada a conexão do *smartphone* com a internet. Logo após foi dito o comando: “Temperatura Corporal”. O *smartwatch* retornou o resultado que foi informado, em voz, pelo aplicativo.

O teste foi repetido diversas vezes em duas pessoas diferentes e os resultados foram colocados na Tabela 3 a seguir.

Termômetro	Smartwatch	Termômetro	Smartwatch
36,4 °C	36,6 °C	36,6 °C	36,4 °C
36,3 °C	36,6 °C	36,6 °C	36,3 °C
36,4 °C	36,7 °C	36,7 °C	36,5 °C
36,5 °C	36,6 °C	36,6 °C	36,5 °C
36,6 °C	36,5 °C	36,5 °C	36,5 °C
36,5 °C	36,3 °C	36,7 °C	36,6 °C
Pessoa 1		Pessoa 2	

Tabela 3. Teste das medições da Temperatura.



Figura 24. Termômetro digital.

4.5 Teste do botão de localizar celular

Um dos recursos que constam no projeto é um botão presente no relógio que, quando apertado, manda um sinal para o *smartphone* solicitando sua localização. Quando o aplicativo recebe esta solicitação, o mesmo emite um sinal sonoro e uma vibração para que possa ser mais facilmente localizado pelo usuário.

No teste, primeiro foi feita a conexão entre o *smartphone* e o *smartwatch*, e logo em seguida foi apertado o botão. Imediatamente o *smartphone* emitiu um som de alerta e uma vibração como era esperado. O teste foi repetido para várias distâncias diferentes e funcionou em uma distância máxima de aproximadamente 12 metros.

4.6 Teste da bateria e do módulo de cargas

Um dos grandes problemas encontrados no projeto foi em relação a bateria. A bateria ideal para o projeto deveria ter pequenas dimensões, ter baixo custo e uma boa autonomia. Seguindo esses requisitos, foi utilizada uma bateria de 100 mAh com dimensões de aproximadamente 15x23 mm. No entanto, a mesma não demonstrou uma boa autonomia durante os testes. Mesmo com todos os ajustes no projeto para economia de energia, a duração de uma carga da bateria foi de aproximadamente dois dias, o que é muito pouco se comparado com outros dispositivos disponíveis no mercado.

O módulo de cargas, por outro lado, funcionou como esperado. Durante os testes ele foi capaz de carregar perfeitamente a bateria, inclusive acendendo um led indicando que a carga estava completa.

4.7 Teste do leitor de textos

O último teste realizado no projeto foi o do recurso de reconhecimento de texto através da câmera. Com este recurso, o usuário pode capturar uma imagem com a câmera, e o aplicativo irá reconhecer qualquer texto que esteja contido na imagem. Todo o processo é feito através de comandos de voz. Ao solicitar o comando, o aplicativo irá pedir para que usuário posicione câmera próximo ao texto, e o próprio aplicativo irá inicializar a câmera e fazer a captura da imagem. Após a captura, a imagem será processada e o texto reconhecido será lido ao usuário pelo aplicativo.

Durante os testes, o reconhecimento do texto foi bastante preciso, reconhecendo inclusive textos manuscritos. No entanto, dependendo da qualidade do texto e também da qualidade da imagem, o processamento pode não ser feito corretamente e o texto vir com alguns erros. Mas, de modo geral, a transcrição da imagem em texto foi feita de forma bastante satisfatória.

4.8 Custo total do projeto

Todo o projeto foi construído levando-se em consideração o baixo custo. Porém, o valor final para a montagem de apenas um *smartwatch* ficou um pouco acima do preço de outros *smartwatches* presentes no mercado. Esse valor, no entanto, poderia ser reduzido se a produção for feita em grande escala.

Foi feita uma cotação para todos os componentes do projeto em alguns fornecedores nacionais e os melhores preços para os componentes e o valor final do dispositivo, podem ser conferidos na Tabela 4.

Arduino Mini Pro	R\$ 12,18
MLX90614	R\$ 40,90
Bluetooth 4.0 Hm-10	R\$ 24,40
MAX30100	R\$ 32,45
Case 3D (Impressão)	R\$ 30,00
Pulseira Relógio	R\$ 20,00
Bateria LiPO 100mah	R\$ 25,00
TOTAL:	R\$ 184,93

Tabela 4. Cotação Itens Smartwatch em dez/2019

5. CONCLUSÕES

A abordagem deste capítulo trata em analisar e discorrer acerca dos itens proposto neste projeto, como as principais dificuldades, problemas ocorridos durante a elaboração bem como propostas de melhorias para futuras versões.

5.1 Principais Dificuldades Encontradas

Dentre as principais dificuldades encontradas, podemos destacar o alto custo de implementação do projeto sem aporte financeiro de terceiros, sendo todo o projeto custeado pelos próprios alunos desenvolvedores, o que inviabilizou o uso de diversas ferramentas mais robustas durante o desenvolvimento. Outra dificuldade encontrada otimização do uso da bateria. Mesmo com todos os ajustes para economia de energia, o consumo da bateria continuou bastante elevado se comparado com outros smartwatches do mercado.

5.2 Trabalhos Futuros

Uma possível melhoria que poderia ser implementada seria o projeto de uma placa de circuito impresso onde todos os componentes eletrônicos do smartwatch poderiam ser migrados. Desta forma, todo o circuito eletrônico estaria disposto em uma placa única, ocupando muito menos espaço e portanto possibilitando a redução no tamanho do dispositivo.

Com o projeto da placa de circuito impresso e conseqüentemente a redução nas dimensões do smartwatch, seria possível a agregação de outros sensores, como, por exemplo, um acelerômetro para detecção de passos e gestos.

6. REFERÊNCIAS

Digital plug & play infrared thermometer in a O-can, Melexis, 2019 Disponível em:

<<https://www.melexis.com/en/product/mlx90614/digital-plug-play-infrared-thermometer-to-can/>> Acesso em: 25 de novembro de 2019.

Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health, Maxim Integrated, 2019 Disponível em:

<<https://www.maximintegrated.com/en/products/sensors/MAX30100.html/>> Acesso em: 25 de novembro de 2019.

What is Arduino?, Arduino, 2019 Disponível em:

<<https://www.arduino.cc/en/Guide/Introduction/>> Acesso em: 25 de novembro de 2019.

What is Xamarin?, Microsoft .Net Disponível em:

<<https://dotnet.microsoft.com/apps/xamarin/>> Acesso em: 25 de novembro de 2019.

Xiaomi Mi Band, Wikipedia, Disponível em:

<https://pt.wikipedia.org/wiki/Xiaomi_Mi_Band/> Acesso em: 25 de novembro de 2019.

SketchUp, Wikipedia, Disponível em: <<https://pt.wikipedia.org/wiki/SketchUp/>> Acesso em: 25 de novembro de 2019.

Estatísticas da deficiência visual, Fundação Dorina Nowill para Cegos, 2019 Disponível em:

<<https://www.fundacaodorina.org.br/a-fundacao/deficiencia-visual/estatisticas-da-deficiencia-visual/>> Acesso em: 25 de novembro de 2019.

BRASIL, 2015, **Lei n. 13.146, de 6 de jul. de 2015**. Lei Brasileira de Inclusão da pessoa com Deficiência. Disponível em:

<http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2015/Lei/L13146.htm>. Acesso em: 25 de novembro de 2019.

Estudante de Rio Claro cria pulseira para deficientes visuais, Portal Diário do Rio Claro, Disponível em:

<<http://j1diario.com.br/estudante-de-rio-claro-cria-pulseira-para-deficientes-visuais/>> Acesso em: 25 de novembro de 2019.

Conheça o relógio sul-coreano que exibe notificações para braille, Portal Olhar Digital, Disponível em:

<<https://olhardigital.com.br/noticia/conheca-o-relogio-sul-coreano-que-exibe-notificacoes-para-braille/88100>> Acesso em: 25 de novembro de 2019.