

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

KLEISSON ROQUE TEDESCO

**SISTEMA PARA MONITORAMENTO REMOTO DOS SINAIS VITAIS
DE UM BEBÊ**

TRABALHO DE CONCLUSÃO DE CURSO 2

PATO BRANCO

2016

KLEISSON ROQUE TEDESCO

**SISTEMA PARA MONITORAMENTO REMOTO DOS SINAIS VITAIS
DE UM BEBÊ**

Trabalho de Conclusão de Curso como requisito parcial à obtenção do título de Bacharel em Engenharia de Computação, do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná.

Orientador: MSc. Robison Cris Brito

PATO BRANCO

2016



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Engenharia de Computação



TERMO DE APROVAÇÃO

Às 8 horas e 20 minutos do dia 7 de dezembro de 2016, na sala V008, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, reuniu-se a banca examinadora composta pelos professores Robison Cris Brito (orientador), Vinicius Pegorini e Kathya Silvia Collazos Linares para avaliar o trabalho de conclusão de curso com o título **Sistema para monitoramento remoto dos sinais vitais de um bebê**, do aluno **Kleisson Roque Tedesco**, matrícula 01114964, do curso de Engenharia de Computação. Após a apresentação o candidato foi arguido pela banca examinadora. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.

Robison Cris Brito
Orientador (UTFPR)

Vinicius Pegorini
(UTFPR)

Kathya Silvia Collazos Linares
(UTFPR)

Beatriz Terezinha Borsoi
Coordenador de TCC

Pablo Gauterio Cavalcanti
Coordenador do Curso de
Engenharia de Computação

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

TEDESCO, Kleisson Roque. Sistema para monitoramento remoto dos sinais vitais de um bebê. 2016. 58f. Trabalho de Conclusão de Curso de bacharelado em Engenharia de Computação - Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

O presente trabalho aborda o projeto e a implementação de um sistema de monitoramento dos sinais vitais de um bebê, para apoio ao diagnóstico e uso pediátrico. O texto apresenta uma abordagem sobre os sinais vitais monitorados, padrões de normalidade e dispositivos de mensuração. Também aborda as ferramentas utilizadas para a implementação do projeto, e o método como elas foram associadas. O sistema é composto por um sensor de temperatura, um sensor de movimentos e um sensor de batimentos cardíacos conectados ao microcontrolador que envia estes dados através da rede Wi-Fi, para um servidor. Os dados são recebidos por um aplicativo, desenvolvido para *devices* Android, que realiza a interface entre o usuário e o sistema. Ao final do desenvolvimento, foi verificado que os objetivos propostos foram atendidos, a transmissão dos dados lidos nos sensores atendeu as expectativas.

Palavras-chave: Monitoramento. Sinais vitais. Microcontrolador. Android.

ABSTRACT

TEDESCO, Kleisson Roque. System for remote monitoring of a baby's vital signs. 2016. 58f Trabalho de Conclusão de Curso de Bacharelado em Engenharia de Computação - Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

This paper presents the design and implementation of a system to monitor the vital signs of a baby, to support the diagnosis and pediatric use. The text presents an approach on monitoring signals, normality standards and measurement devices. It also represent how tools used to implement the project, and how they are associated. The system consists of a temperature sensor, a motion sensor and a heart rate sensor connected to the microcontroller that sends this data over the Wi-Fi network, to a server. The data is served by an application, developed for Android devices, which performs the interface between the user and the system. At the end of the development, it was verified that the proposed objectives were achieved, the transmission of the data read in the sensors achieved the expectations.

Keywords: Monitoring. Vital signs. Mircocontroller. Android.

LISTA DE FIGURAS

Figura 1 – Causas de morte em crianças menores que 5 anos de idade em 2015.	14
Figura 2 - Número de óbitos infantis (masculinos e femininos) notificados, no ano selecionado, segundo capítulo da CID10.	15
Figura 3 - Número de óbitos infantis (masculinos e femininos) notificados, no ano selecionado e últimos quatro anos precedentes, segundo grupo etário.	15
Figura 4 – Termômetro Digital	22
Figura 5 – Estetoscópio Digital.....	23
Figura 6 – Oxímetro Digital.....	23
Figura 7 - Sensor de temperatura Dallas DS18B20	29
Figura 8 - Pulse Sensor (Sensor Batimentos)	30
Figura 9 - Módulo MPU6050.....	31
Figura 10 - Power Bank Sony	31
Figura 11 - Microcontrolador Spark Core.....	32
Figura 12 - Dashboard Ubidots	33
Figura 13 - Escopo do Sistema.....	38
Figura 14 - Diagrama Sensor Temperatura.....	38
Figura 15 - Diagrama Sensor Batimentos	39
Figura 16 - Diagrama Módulo MPU6050	40
Figura 17 - Diagrama de Blocos Ubidots	40
Figura 18 - Diagrama Aplicativo	41
Figura 19 - Diagrama Servidor <i>push</i>	41
Figura 20 - Faixa Abdominal Desenvolvida.....	42
Figura 21 - Tela Aplicativo	43
Figura 22 - Notificação sendo recebida	44
Figura 23 - Testes.....	53

LISTA DE QUADROS

Quadro 1 - Iterações definidas	35
---	-----------

LISTA DE TABELAS

Tabela 1 – Temperatura corporal normal	20
Tabela 2 – Frequência cardíaca normal.....	22

LISTAGENS DE CÓDIGO

Listagem 1 – Codigos Biblioteca spark-dallas-temperature e onewireunofficial	44
Listagem 2 – Funções Sensor Batimentos.	45
Listagem 3 - Codigo módulo MPU6050	46
Listagem 4 - Código Ubidots	46
Listagem 5 - Classe Ubidots Client.....	49
Listagem 6 - Código Aplicativo	50
Listagem 7 - Classe FIREbaseMessagingService.....	51
Listagem 8 - Classe FIREbaseInstanceIDService	51
Listagem 9 - Servidor Verificação <i>Push</i>	52
Listagem 10 - Post no FCM	52

LISTA DE SIGLAS E ABREVIATURAS

SMSL	Síndrome da Morte Súbita do Lactente
SIDS	<i>Sudden Infant Death Syndrome</i>
IoT	<i>Internet of things</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
UTIP	Unidade de Terapia Intensiva Pediátrica
UTIN	Unidade de Terapia Intensiva Neonatal
BPM	Batimentos Por Minuto
FCM	<i>Firebase Cloud Messaging</i>

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 CONSIDERAÇÕES INICIAIS	11
1.2 PROBLEMA.....	13
1.3 OBJETIVOS.....	16
1.3.1 Objetivo Geral.....	16
1.3.2 Objetivos Específicos.....	16
1.4 JUSTIFICATIVA	17
1.5 ESTRUTURA DO TRABALHO	18
2 REFERENCIAL TEÓRICO	20
2.1 SINAIS VITAIS, FORMAS, DISPOSITIVOS DE MENSURAÇÃO E SEUS PADRÕES DE NORMALIDADE.....	20
2.2 FATORES DE RISCO PARA A MORTE SUBITA INFANTIL E POSSÍVEIS FORMAS DE PREVENÇÃO.....	24
2.3 ESTADO DA ARTE.....	25
3 MATERIAIS E MÉTODO	28
3.1 MATERIAIS.....	28
3.1.1 Faixa Abdominal.....	28
3.1.2 Sensores.....	28
3.1.3 Bateria	31
3.1.4 Microcontrolador e Módulo de Comunicação.....	32
3.1.5 Servidor Ubidots	33
3.1.6 Plataforma Android.....	34
3.2 MÉTODO	34
4 RESULTADO	37
4.1 ESCOPO DO SISTEMA.....	37
4.2 MODELAGEM DO SISTEMA.....	38
4.3 APRESENTAÇÃO	41
4.4 IMPLEMENTAÇÃO	44
4.5 TESTES	52
5 CONCLUSÃO E TRABALHOS FUTUROS.....	54
REFERÊNCIAS.....	56

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais do presente trabalho. Aspectos conceituais que envolvem o problema, os objetivos, as justificativas e os argumentos que defendem a escolha do tema e da solução proposta.

1.1 CONSIDERAÇÕES INICIAIS

Com a evolução da informática, em especial, dos dispositivos eletrônicos que estão cada vez mais rápidos, pequenos, baratos e com mais recursos, é possível projetar soluções envolvendo hardware e software para automatizar muitas das tarefas do cotidiano.

Este avanço inclui as técnicas comunicação móvel, a computação embarcada e a miniaturização dos dispositivos e sensores eletrônicos, que está gerando progressos significativos para o desenvolvimento de aplicações na área médica, permitindo a otimização da prestação de serviços pelos profissionais de saúde.

O aprimoramento da comunicação sem fio (*wireless*) expandiu as possibilidades de monitoramento e controle dos dispositivos eletro-médicos em meio remoto, ampliando a locomobilidade dos profissionais de saúde. A miniaturização dos dispositivos eletrônicos móveis, a maior eficiência de baterias e a redução de consumo de energia dos semicondutores potencializaram o desenvolvimento de inúmeras soluções inovadoras através da computação ubíqua (RENATO et al., 2013).

Computação Ubíqua é a ideia de que a computação se move para fora das estações de trabalho e computadores pessoais e torna-se pervasiva na nossa vida cotidiana, aonde quer que estejamos. É uma tendência que está crescendo exponencialmente nos últimos anos (KAHL; FLORIANO, 2011).

Tendo em vista que a portabilidade e a funcionalidade dos equipamentos eletrônicos estão cada vez mais aperfeiçoadas, os dispositivos “vestíveis” estão revolucionando o mercado como uma nova maneira de cuidar da saúde. Estes dispositivos são chamados de *wearable computer* (VAN LAERHOVEN; SCHMIDT; GELLERSEN, 2002).

O termo *wearable computer* está definido com o um computador vestível, portátil que auxilie o ser humano em tarefas do dia a dia, ou ainda, permitindo o monitoramento de parâmetros, disponibilizando estes dados para o indivíduo posteriormente.

Wearable computer pode ser classificado como um computador que está sempre ligado e sempre acessível (MELOROSE; PERROY; CAREAS, 2015), podendo este ser altamente usual em crianças recém nascidas para controlar sua temperatura, movimentos no berço e até em idosos para controlar sua pressão sanguínea, batimentos cardíacos entre outros.

Este trabalho irá abordar o uso de *wearable computer* em crianças recém nascidas, sendo que no início da vida humana, um monitoramento contínuo, principalmente durante o sono, pode auxiliar os pais, em especial os com pouca experiência, a lidar com situações rotineiras, como o monitoramento das febres, o movimento no berço dos bebês recém nascidos, pois qualquer anomalia detectada, combinada de uma ação rápida, pode contribuir para diminuir o risco da morte súbita.

Morte súbita e inexplicável continua sendo uma das principais causas da mortalidade infantil. A síndrome da morte súbita do lactente (SMSL), derivada do inglês *Sudden Infant Death Syndrome* (SIDS) é definida como a morte súbita de um bebê com menos de 1 ano de idade, normalmente ocorre durante o sono, e permanece não explicada após uma investigação completa, incluindo uma autópsia completa e uma avaliação das circunstâncias do ambiente do sono. Uma das maneiras possíveis de se descobrir a causa da morte súbita é o monitoramento dos sinais vitais, que permite a coleta de dados para análise posterior, facilitando ao médico a elaboração de um laudo mais preciso e mais confiável (RANDALL et al., 2013).

Ainda, o monitoramento pode gerar estatísticas que quando analisadas podem detectar alguns tipos de anomalias, como cardiopatias congênitas, que se monitoradas podem gerar prognósticos mais eficientes, melhorando a forma do tratamento.

Para realizar o monitoramento remoto, conceitos de Internet das Coisas foram utilizadas para projetar o sistema. Internet das Coisas, do inglês IoT, (*internet of things*) é o termo utilizado para classificar os dispositivos comuns de uso no dia a dia que possuem acesso a Internet. Estes dispositivos são capazes de mensurar dados, controlar sistemas e até mesmo auxiliar nas tarefas domésticas. Por meio de sensores ou dispositivos vestíveis pode-se, intermitente ou continuamente, monitorar e gravar em um servidor de Internet dados relevantes. Esta forma de coleta e armazenamento de dados é uma importante solução para os limites impostos pelo tradicional monitoramento destes sinais. Possibilitando inclusive disparar notificações que auxiliem na prevenção de possíveis pioras na saúde do usuário do sistema.

O presente trabalho apresenta a elaboração de um dispositivo *Wearable Computer* para mensurar sinais vitais de bebês, assim como monitorar comportamento dos mesmos, e notificar um dispositivo Android quando algum dos sinais mensurados não esteja dentro dos padrões pré-definidos. Para gerenciamento destes dados, bem como sua disponibilização ao usuário, é utilizada uma estrutura baseada na Internet das Coisas, com estrutura de persistência e comunicação compatíveis para este.

1.2 PROBLEMA

Segundo a Organização Mundial da Saúde, no ano de 2015, 6,5 milhões de crianças morreram antes de completarem 5 anos de idade. O risco de uma criança morrer antes de completar esta idade é mais alto na Região Africana, com cerca de 81 mortes para cada 1000 nascidos vivos.

Muitos países ainda têm um elevado índice de mortalidade infantil, particularmente aqueles na Região Africanas. As desigualdades na mortalidade infantil entre os países de alta e baixa renda permanecem grandes. Em 2015, a taxa de mortalidade de menores de cinco ano em países de baixa renda foi de 76 mortes por 1000 nascidos vivos, cerca de 11 vezes a taxa média nos países de alta renda (7 mortes por 1.000 nascidos vivos) (WORLD HEALTH ORGANIZATION, 2016).

Além da desigualdade social, diversos fatores contribuem para a composição destes números, dentre eles diarreia, pneumonia, HIV, entre outros. Na Figura 1, estes números estão representados em percentual, divididos pelas causas registradas. A síndrome da morte súbita do lactente esta classificada no grupo outros, que se caracteriza por mortes de causa desconhecida.

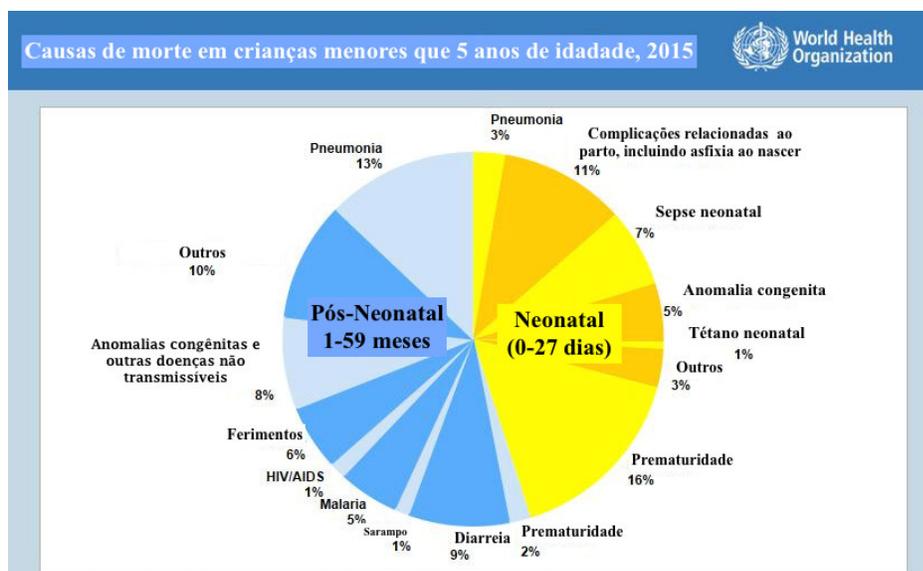


Figura 1 – Causas de morte em crianças menores que 5 anos de idade em 2015.

Fonte: Adaptado de (WORLD HEALTH ORGANIZATION, 2016).

De acordo com o Centro de Controle de Doenças e Prevenções Americano (*Center for Diseases Control and Preventions*) cerca de 3500 bebês menores de um ano morrem por ano de causas desconhecidas nos Estados Unidos. Deste numero 44% dos casos foram classificados como síndrome da morte súbita do lactente, 31% causa desconhecida, e 25% asfixia acidental ou estrangulamento na cama.

Nos Estados Unidos, estes números diminuiram consideravelmente após o lançamento pela Academia Americana de Pediatria a campanha recomendações *Safe Sleep*, em 1992, o início da campanha *Back to Sleep* em 1994, e o lançamento do *Sudden Unexplained Infant Death Investigation Reporting Form* em 1996, que são campanhas de incentivo ao sono seguro, orientando os pais sobre as melhores formas de colocar o bebe para dormir (CENTER FOR DISEASE CONTROL AND PREVENTION, 2015).

No Brasil, este índice é calculado pelo Sistema Único de Saúde (SUS), e consta na lista de causas evitáveis de morte reduzíveis por ações adequadas de promoção a saúde, vinculadas a ações de atenção a saúde. A organização Mundial da Saúde classifica a Síndrome da Morte Súbita do Lactente em um grupo particular, denominado de Sintomas, sinais e achados anormais de exames clínicos e de laboratório, e o SUS segue hierarquicamente esta classificação (MALTA et al., 2011).

No ano de 2015 foram registradas 31.441 mortes de crianças menores que 4 anos segundo o Painel de Monitoramento Infantil e Fetal. Dessas mortes, 953 (2%) foram classificadas nas que se referem a morte súbita, representado pelo capítulo XVIII como representa a Figura 2 (SISTEMA UNICO DE SAUDE, 2016).

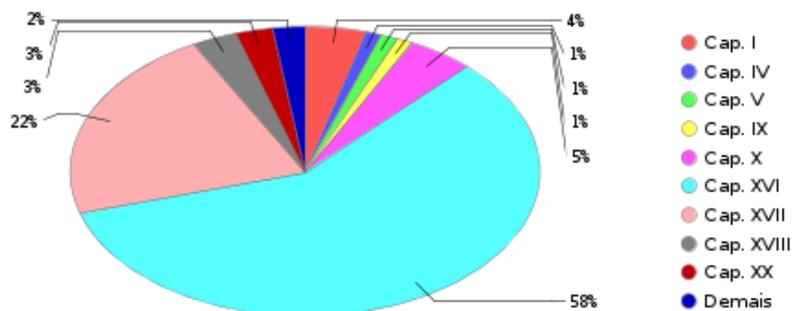


Figura 2 - Número de óbitos infantis (masculinos e femininos) notificados, no ano selecionado, segundo capítulo da CID10.

Fonte: (SISTEMA UNICO DE SAUDE, 2016)

A Figura 3 classifica as mortes por seus tipos de grupo, e apresenta como foram distribuídas estas mortes nos anos anteriores. Estes casos estão divididos em 3 grupos: Neonatal precoce (primeira semana completa de vida) 167 em 2015, Neonatal Tardia (três semanas seguintes) 66 em 2015 e pós-neonatal e meses seguintes 720 em 2015. Os números somam um total de 953 mortes em 2015, e estão diminuindo devido a campanhas governamentais para o acompanhamento e aconselhamento de prevenção a síndrome da morte súbita do lactente (SISTEMA UNICO DE SAUDE, 2016).

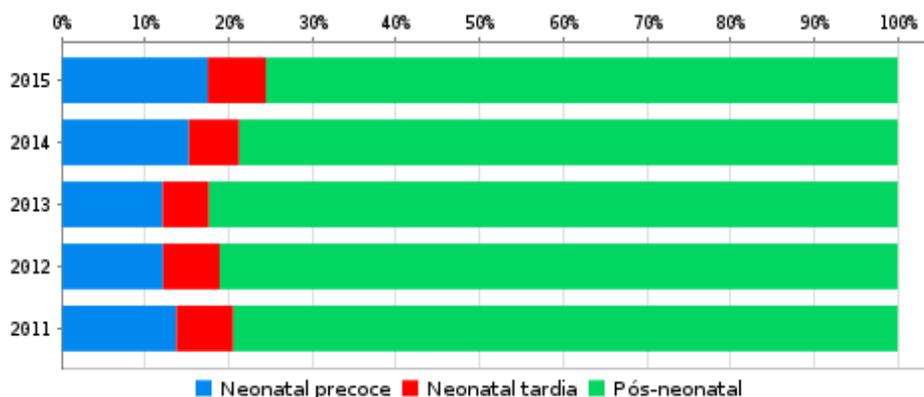


Figura 3 - Número de óbitos infantis (masculinos e femininos) notificados, no ano selecionado e últimos quatro anos precedentes, segundo grupo etário.

Fonte: (SISTEMA UNICO DE SAUDE, 2016)

Vários estudos epidemiológicos realizados desde a década de 60 evidenciam alguns fatores que contribuem para um perfil de risco do lactente. Dentre estes: mãe jovem, múltipara, intervalo curto entre as gestações, ausência de pré-natal, prematuridade, baixo peso de nascimento, irmão que morreu da SMSL (Síndrome da Morte Súbita do Lactente), baixo nível sócio econômico, além de um predomínio no sexo masculino, etnias e regiões geográficas. Fortes indícios evidenciam que há um risco até 14% maior quando os lactentes dormem na posição peito para baixo.

A partir destes indícios foi iniciada a campanha “Reduza o risco de SMSL” na década de 90, e logo foi observada uma redução na mortalidade pós-natal em até 50%. Um estudo realizado no Rio Grande do Sul, nas cidades de Pelotas e Porto Alegre evidenciou que os casos de SMSL são sub diagnosticados e não constam nos dados oficiais.

Este mesmo estudo atribui um índice de 4% para a região de Pelotas como consequência da SMSL, e um índice de 0.08% na região de Porto Alegre. Reanalizando a Figura 2, e comparando com os números deste estudo atribui-se uma estimativa de aproximadamente 6.120 casos por ano no Brasil (MAGDA; NUNES, 2005).

1.3 OBJETIVOS

Os objetivos do trabalho dividem-se em geral e específicos, que são apresentados a seguir.

1.3.1 Objetivo Geral

Desenvolver um protótipo de monitoramento de sinais vitais de um bebê por meio de um *wearable computer* que colete sinais vitais e os envie para um dispositivo Android. Com auxílio da arquitetura de IoT, informe estes dados e notifique via smartphone se algum parâmetro pré-estabelecido esta anormal.

1.3.2 Objetivos Específicos

Destacam-se como objetivos específicos:

- Implementar a aquisição de dados utilizando sensores.
- Implementar e testar o protótipo do aplicativo desenvolvido para a transmissão de dados.

- Desenvolver um aplicativo para smartphones que receba os dados dos sensores e os disponibilize para visualização e emita alertas de acordo com parâmetros pré-estabelecidos.

1.4 JUSTIFICATIVA

Morte súbita não apresenta na literatura uma definição uniforme, pode ser descrita como quando se excluem todas as causas violentas de óbito (homicídio, suicídio, envenenamento ...). No ano de 1970, um comitê internacional composto por patologistas clínicos e epidemiologistas concluiu que a morte súbita e inesperada é definida como morte instantânea ou dentro de 24 horas após o início dos sinais e sintomas. Caso a morte súbita tenha ocorrido em lugares não-assistidos por médico ou se esse tem alguma dúvida sobre o que levou o paciente ao óbito, um exame necroscópico deve ser solicitado ao serviço de verificação de óbito da região. As principais causas para a morte súbita são as doenças cardiovasculares, é mais comum nas populações urbanas do que nas rurais e nas classes sociais mais baixas (REIS; CORDEIRO; CURY, 2006).

A morte súbita também é comum em bebês e ninguém sabe a sua origem na maioria dos casos. A morte súbita tem várias causas que em algumas circunstâncias pode ser evitada. Sempre existem novos desafios para poder oferecer uma maior e melhor qualidade de vida para pais e bebês. Assim, os pais podem sentir-se mais seguros em relação a seus bebês que também serão mais protegidos de certa forma. Em um cenário real seria possível prever quando algo dá errado com um bebê, a fim de evitar males maiores ou até mesmo a morte (FONSECA et al., 2015).

Não só a morte súbita, mas também outros problemas preocupam os pais logo nos primeiros meses de vida da criança. A febre por exemplo, ocorre frequentemente e sem razão aparente.

Segundo Alves et al., (2004), mais de um terço dos atendimentos pediátricos são motivados pela febre que, na maioria das vezes, tem como causas infecções benignas e de evolução autolimitada. A febre representa a principal razão para o emprego de drogas na prática pediátrica; mais de 80% das crianças já utilizaram algum tipo de antipirético (remédios para baixar a febre) antes de completar o primeiro ano de vida. A febre é também a razão mais frequente para o emprego de antibióticos, na maioria das vezes de forma inadequada.

Embora a febre não seja uma doença, e sim, o sintoma de algum problema que esteja ocorrendo (ex. inflamação de garganta, problemas intestinais, bronquite, etc.), este sintoma é

muito desconfortável, especialmente para os bebês, deixando-as irritadas, com dificuldade em descansar e/ou pegar no sono. Também, quando não controlada, a febre pode atingir temperaturas altas, podendo provocar outros sintomas, como convulsões e vômitos.

Neste cenário, um dispositivo que monitore estas variáveis pode trazer um maior conforto e tranquilidade aos pais. Pois como as causas não podem ser identificadas na hora em que o problema surge, um monitoramento pode auxiliar em um melhor prognóstico.

O presente trabalho busca agregar conhecimento sobre as tecnologias da área de Internet das Coisas, abrindo um leque de estudos na monitoração de sinais vitais por meio de sensores já existentes.

A computação ubíqua tem como objetivo incorporar os dispositivos computacionais no cotidiano das pessoas, de tal forma que a interação entre os mesmos seja feita da maneira mais natural possível. Para isto estes dispositivos precisam interpretar as formas usuais de comunicação dos usuários e fazer a captura dos seus contextos de interesse.

Um dos grandes benefícios possibilitados pela computação ubíqua, para os profissionais de saúde é a redução dos esforços na execução das tarefas, monitorar dados vitais, trazendo maior satisfação pelo trabalho e melhor qualidade de vida. Os pacientes, por sua vez, ganham com a redução dos erros médicos e melhores resultados no tratamento da saúde (RENATO et al., 2013)

Utilizando sensores, microcontroladores, transmissores e um banco de dados, é possível construir um sistema no qual sinais vitais de um ser humano podem ser obtidos e disponibilizados a pacientes, médicos, enfermeiras, cuidadores, familiares e outros. Com base nisso, considerando um cenário real, utilizando tecnologia móvel, este trabalho propõe um sistema que tenta monitora e detecta situações de risco em bebês e alertar aqueles que são responsáveis por ela / ele ou até mesmo autoridades competentes.

1.5 ESTRUTURA DO TRABALHO

No primeiro capítulo são apresentadas as considerações iniciais, onde se estuda a abordagem inicial para o tema, logo apos os objetivos estão divididos em gerais e específicos. É apresentada uma justificativa sobre a importância do desenvolvimento do sistema de monitoramento. No segundo capítulo tem-se o referencial teórico, onde são apresentados os sinais vitais, seus padrões de normalidade e as formas como estes sinais podem ser mensurados, também contem o estado da arte que informa os trabalhos semelhantes

encontrados. No capítulo Materiais e Métodos são apresentados os componentes utilizados no trabalho, assim como a forma em que estes se intercomunicam. Por fim, a conclusão que ressalta a importância do desenvolvimento do projeto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta conceitos sobre sinais vitais e seus padrões de normalidade. Na sequência são apresentados trabalhos semelhantes já desenvolvidos.

2.1 SINAIS VITAIS, FORMAS, DISPOSITIVOS DE MENSURAÇÃO E SEUS PADRÕES DE NORMALIDADE.

Os indicadores do estado de saúde de um paciente são temperatura, pulso, respiração e pressão arterial, esses indicadores permanecem mais ou menos constantes e sua variação pode indicar alguma enfermidade. (MURTA; RUSSI, 2006).

Os indicadores do funcionamento fisiológico básico, ou seja, do estado de equilíbrio térmico, circulatório e respiratório. São denominados sinais vitais. Recebem esse nome em razão de sua importância e por seus parâmetros serem regulados por órgãos vitais, revelando o funcionamento deles. A variação dos valores pode indicar problemas relacionados com insuficiência ou excesso de consumo de oxigênio, depleção sanguínea, desequilíbrio eletrolítico, invasão bacteriana, entre outros. (MURTA; RUSSI, 2006).

O equilíbrio entre o calor produzido pelo organismo e o calor dissipado para o ambiente é chamado de temperatura corporal. Pode ser verificado na região axilar, inguinal, bucal ou retal. A axilar é a mais utilizada pela praticidade, podendo o termômetro ser compartilhado entre diferentes pacientes, embora menos fidedigna, e seu valor normal varia no adulto entre 36,5°C e 37°C. O mesmo padrão de normalidade é atribuído para a verificação inguinal. Já para verificação oral fica entre 36,8°C a 37,2°C enquanto a retal entre 37°C e 38°C. Os limites letais para a temperatura corporal oscilam entre 22°C e 42°C, geralmente a hipotermia é menos tolerada que a hipertermia. (MURTA; RUSSI, 2006).

A temperatura central do corpo pouco se altera, mas a superficial varia de acordo com a vascularização e o ambiente. Existe uma temperatura que se mantém relativamente constante no organismo que é a temperatura central, dos tecidos profundos.

A temperatura superficial pode variar conforme o fluxo sanguíneo para os tecidos e a quantidade de calor perdido para o ambiente externo (POTTER, 2004). As terminologias relativas à temperatura corporal são apresentadas na Tabela 1.

Tabela 1 – Temperatura corporal normal

Afebril	36 a 37°C
Estado Febril	37,5 a 37,8°C
Febre	38 a 38,9°C
Pirexia	39 a 40°C
Hiperpirexia	Acima de 40°C
Hipotermia	Temperatura abaixo de 35°C
Hipertermia	38 a 40°C

Fonte: (MURTA; RUSSI, 2006).

Em bebês com menos de três meses, temperatura acima de 37,5°C ou abaixo de 35,5°C são consideradas anormais, e em bebês com mais de três meses, temperaturas superiores a 39°C, ou se a febre vier acompanhada de choro persistente e irritabilidade extrema (POTTER, 2004).

A temperatura pode ser verificada com termômetros de vidro (com bulbo alongado, para uso oral e axilar; bulbo reduzido, utilizado em qualquer local; e bulbo vermelho, indicado para uso retal), termômetros descartáveis (fitas de plástico com sensor) ou termômetros eletrônicos timpânico, axilar e oral.

A mensuração da temperatura é uma parte fundamental na avaliação física da maioria de pacientes em situação de emergência, ou eventuais enfermidades (MURTA; RUSSI, 2006).

A maioria dos semicondutores possuem uma resistividade elétrica que muda conforme a variação de temperatura. Assim, um resistor concebido a partir deste material apresenta uma resistência que varia com a temperatura. O resistor sensível à temperatura que é usado para esta medição é conhecido como um detector de temperatura de resistência. Geralmente têm resistências nominais da ordem de centenas de ohms com um coeficiente de temperatura positivo inferior a 0,5%/°C. Sua resistência versus sua característica de temperatura é linear, sendo este comumente usado na medicina como é o caso do termômetro digital (NEUMAN, 2010).

Este termômetro permite a verificação de um sinal vital do paciente e a partir dos dados aferidos, o médico pode tomar decisões que serão relativamente fundamentais no tratamento da enfermidade. A Figura 4 representa um termômetro digital usado na maioria dos hospitais.



Figura 4 – Termômetro Digital
Fonte: Adaptado da Internet

Estes tipos de sensores são encontrados em vários tamanhos e formas, dependendo de sua aplicação. A resistência, como uma função da característica de temperatura, é dada pela Eq. 1.

$$R(T) = R_0[1 + \alpha(T - T_0)] \quad (1)$$

onde R_0 é a Resistencia do sensor em uma conhecida temperatura T_0 , α é o coeficiente resistivo de temperatura, e T é a temperatura (NEUMAN, 2010).

Um segundo sinal vital é a pulsação (frequência cardíaca) que é a elevação palpável do fluxo sanguíneo percebida em qualquer lugar do corpo onde possa ser detectada pulsação arterial, nesses locais existem artérias que transmitem os pulsos para a superfície da pele. Para aferir a pulsação deve-se contar os batimentos cardíacos durante um minuto. Esta será a frequência cardíaca, classificada conforme a Tabela 2.

Tabela 2 – Frequência cardíaca normal

Idade	Frequência normal
Bebês de menos de um ano.	70 a 190 batimentos por minuto.
Crianças de 1 a 10 anos.	80 a 110 batimentos por minuto.
Crianças de mais de 10 anos e adultos.	70 a 100 batimentos por minuto.
Atletas bem treinados.	50 a 100 batimentos por minuto.

Fonte: (MURTA; RUSSI, 2006).

Um dispositivo utilizado para medir os batimentos cardíacos é conhecido com estetoscópio que amplifica diversos sinais sonoros, inclusive o sinal dos batimentos cardíacos. Utilizado por profissionais de saúde, o mesmo possibilita diagnosticar analogicamente se o paciente esta com os batimentos na frequência normal. Paralelamente existem dispositivos digitais que a partir de sensores captam esta frequência em diversas partes do corpo, como o representado pela Figura 5. O mesmo possibilita armazenar os dados em um computador e analisa-los de forma visual.



Figura 5 – Estetoscópio Digital
Fonte – Adaptado da Internet

Outro dispositivo é o Oxímetro Digital, representado pela Figura 6. Este dispositivo trabalha com um sensor fotopletismógrafo. Além de representar os batimentos cardíacos este dispositivo é capaz de mensurar a taxa de oxigenação sanguínea.



Figura 6 – Oxímetro Digital
Fonte – Adaptado da Internet

A partir de dispositivos digitais é possível captar estes dados e condicioná-los utilizando-os em sistemas destinados a aplicações voltadas a medicina. Estes dispositivos

digitais possuem uma vantagem sobre os analógicos, como os dados são apresentados de forma fixa, o indivíduo que os observa interpreta-os exatamente como são.

Unidos à informática, estas aplicações possibilitam um vasto uso em momentos nos quais os sinais vitais são a única forma de se obter diagnósticos ou uma interpretação do estado de saúde do ser humano.

2.2 FATORES DE RISCO PARA A MORTE SUBITA INFANTIL E POSSÍVEIS FORMAS DE PREVENÇÃO.

Na maioria dos casos a morte súbita acontece enquanto o bebê está em estado de sono profundo, pois como ainda não desenvolveu mecanismos de defesa contra diversos fatores envolvidos no ambiente de sono, o mesmo encontra-se vulnerável.

Os fatores podem ser descritos como a posição barriga para baixo de dormir e o superaquecimento do ambiente de sono que caracterizam importantes influências de risco. A posição barriga para baixo é considerada a principal causa para a Síndrome da SMSL. A avaliação da influência da posição de dormir sobre o mecanismo do despertar de lactentes saudáveis nascidos a termo, tanto do sono quieto quanto ativo, com maior impacto entre os dois e três meses de idade, caracterizando uma fase de maior risco para a SMSL. Outro fator a ser citado é o coo-leito, definido como o hábito de a criança dormir na mesma cama que o cuidador, também o tabagismo, uso de drogas, e a prática materna de cobrir a cabeça da criança durante o sono.

A cama compartilhada agrava o risco de superaquecimento da criança devido ao calor proveniente do corpo dos pais sob as cobertas, contribuindo também para uma possível asfixia. A temperatura corporal caracteriza um fator chave na prevenção da SMSL, em virtude de sua interação com os mecanismos do sono, do controle respiratório e do despertar, dada a incapacidade do lactente para acordar em resposta a uma temperatura ambiental acima de 28 Graus Celsius (GEIB; NUNES, 2006).

Com o intuito de monitorar a posição de sono do lactente dispositivos de monitoração da posição de um objeto podem ser utilizados a fim de avaliar quando uma mudança de estado aconteceu. Estes sensores são de extrema importância para evitar a SMSL, principalmente no caso da morte por asfixia, caracterizada pela falta de oxigenação sistêmica. Com o intuito de

evitar esse problema, o sensor de posição pode impedir a morte por asfixia, porque uma vez que uma mudança de posição espontânea aconteça, o sensor pode detectá-la e enviar para o dispositivo que possibilite emitir uma notificação do ocorrido. O sensor de posicionamento é um acelerômetro de três eixos, que em um dos eixos representa a posição do bebê.

2.3 ESTADO DA ARTE

O monitoramento remoto de sinais vitais torna-se possível com a disponibilidade de sensores sem fio e de uma rede para acessar estes sensores. Dispositivos de temperatura corporal e medição da frequência cardíaca são exemplos de sensores sem fio usados em aparelhos médicos.

A expansão de serviços de Internet e redes de sensores tem aberto oportunidades na área de saúde. Estes possuem muitos benefícios, onde a coleta de dados e serviço de monitoramento são viáveis.

No monitoramento saúde *in home*, a tecnologia permite que os pacientes monitorem seus próprios sinais vitais e comuniquem os resultados para um profissional do hospital. Assim, o médico pode antecipar ao problema, evitando que este chegue ao estado de cuidados extremos. Com esta ideia, Mansor et al. (2013) propôs uma aplicação na qual os dados referentes aos sinais vitais de um paciente são transmitidos para um servidor, e o médico poderia acessá-los remotamente.

Dentro da população de recém-nascidos internados em hospitais, uma porcentagem significativa poderia obter alta mais cedo se houvesse um sistema que permitisse o monitoramento contínuo de seus sinais vitais pela equipe do hospital, sem a necessidade do bebê internado.

Atualmente, os únicos lugares onde esse nível de monitoramento contínuo pode ocorrer é dentro das Unidade de Terapia Intensiva Pediátrica (UTIP) e Unidade de Terapia Intensiva Neonatal (UTIN). Nestes, os monitoramentos podem ocorrer pela utilização de camas para fins de monitorização e não exclusivamente para o tratamento em curso. (GREER

et al., 2014) propôs um sistema de monitoramento utilizando as redes de comunicação 3G e Wi-Fi, disponibilizando os dados obtidos de crianças hospitalizadas em unidades de tratamento intensivo. Estes dados ficam armazenados em um servidor onde médicos de qualquer lugar podem verificá-los, caso autorizados, ajudando a obter um melhor prognóstico.

FONSECA et al., (2015) desenvolveu um berço que suporta sensores para recolher vários dados inclusive sinais vitais. Este berço se conecta a um smartphone ou Tablet via Bluetooth ou Wi-Fi, enviando estes dados e permitindo o acompanhamento remoto em tempo real.

No início da vida, os bebês passam a maior parte do tempo dormindo, e várias funções vitais podem ser mensuradas com sensores, porém nada pode substituir a presença de um cuidador. O aplicativo móvel é usado como um recurso de monitoramento a mais e fornece uma forma poderosa para apoiar e prevenir a maioria das situações de perigo, tal como uma parada cardíaca associada a uma mudança de posição do bebê.

Um artigo publicado por ACHARYA et al., (2015) mostram que uma parada cardíaca apresenta variações nos batimentos cardíacos algum tempo antes de acontecer. É possível prever uma parada cardíaca quatro minutos antes do seu início, com uma precisão média de 86,8%. Isto agrega importância ao trabalho pois o monitoramento cardíaco contínuo em situações de risco pode prover relatórios que se devidamente analisados por um profissional de saúde contribuem a prover um prognóstico mais adequado.

Muitos bebês prematuros apresentam um perfil de risco maior do que bebês nascidos em tempo normal. Os motivos estão relacionados com um incompleto desenvolvimento dos sistemas fisiológicos que mantem o bebê vivo. Uma análise da variação dos batimentos cardíacos pode ajudar na identificação de condições anormais e ter uma interpretação do comportamento do sistema nervoso central.

O trabalho proposto por LUCCHINI et al., (2014), apresenta os resultados das análises de 35 bebês nascidos prematuramente em sono profundo, nas posições de barriga para baixo e barriga para cima. A taxa de batimentos cardíacos foi analisada tanto no domínio do tempo como no domínio da frequência com parâmetros não lineares. Estes parâmetros foram comparados com os de adultos normais, assim como de bebês em condições normais. O objetivo final foi identificar fatores de risco que possam contribuir para uma possível morte súbita. O monitoramento cardíaco pode identificar varias patologias, logo este estudo pode contribuir para o desenvolvimento mais preciso e personalizado nos processos de diagnósticos.

Um dos produtos existentes no mercado é o Pacif-i, uma chupeta inteligente que monitora a temperatura do bebê e, em seguida, transmite os dados para um aplicativo no smartphone ou Tablet. O dispositivo tem um termômetro incorporado dentro da chupeta de silicone e um chip Bluetooth que envia as leituras de temperatura.

Esses dados também podem ser facilmente compartilhados com outras pessoas, como por exemplo um médico. Além disso, no caso de a criança estar em um tratamento, o aplicativo permite a criação de lembretes para tomar o medicamento.

O dispositivo também possui um sensor de proximidade embutido que permite aos pais monitorar a localização da chupeta e, assim, ser alertado quando a criança esta fora de um limite pré-estabelecido. Os pais podem definir uma distância de até 20 metros, dentro do aplicativo. Um alarme é acionado, assim, quando a chupeta cruza esse limite (MALLIK, 2015)

Criado por ex-funcionários de Google e Apple, o *Sproutling* tem como objetivo principal tornar mais simples o monitoramento de recém-nascidos, já que os dados obtidos pelo sensor conseguem informar quando o bebê está dormindo ou agitado e prestes a acordar, entre outros alertas mostrados na tela do celular por meio de notificações animadas. Esta tornozeleira inteligente envia informações sobre bebês para pais e mães. O Sproutling, como se chama o dispositivo, é capaz de medir batimentos cardíacos, temperatura, umidade e uma série de outros fator (HEATHER, 2014).

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais pesquisados e definidos para a elaboração do projeto, assim como o método de desenvolvimento do protótipo.

Para o desenvolvimento do sistema proposto, várias questões devem ser abordadas, como, por exemplo, o correto funcionamento do sensor, o melhor local onde o dispositivo pode ser colocado afim de se obter uma melhor aferição dos dados, interferências que podem ocorrer na transmissão de dados em geral. Na sequência será feito uma apresentação breve de cada um dos componentes que constituirá o sistema como um todo.

3.1 MATERIAIS

Nesta seção serão descritos os materiais utilizados no desenvolvimento do projeto.

3.1.1 Faixa Abdominal

Segundo Dias (2013), o Neoprene é uma borracha sintética de alta qualidade revestida com tecidos dos dois lados. É um material altamente resistente e que possui grande maleabilidade. Ademais, oferece elasticidade em todas as direções e proporciona uma perfeita modelagem ao corpo, além de não absorver água, o Neoprene recupera sua forma original mesmo depois de esticado ou comprimido. Por esse motivo e por ser considerado um material hipoalérgico e atóxico este material pode ser utilizado na confecção de uma faixa confortável e segura.

Com este material foi desenvolvido uma faixa abdominal onde estão contidos os sensores, o microcontrolador juntamente com o módulo de comunicação e a bateria para alimentação. Esta faixa como um todo possibilita a aquisição dos sinais vitais, o condicionamento dos mesmos e a transmissão para um servidor web, o qual armazenara os dados lidos nos sensores.

Entretanto, o tecido de Neoprene tem como característica ser isolante térmico por ser um composto de borracha sintética, e isto pode interferir na leitura dos sensores. Contudo, um orifício aberto no local onde os sensores estão posicionados permite que esta leitura seja efetuada sem maiores problemas.

3.1.2 Sensores

Sensor é o termo empregado para designar dispositivos sensíveis a alguma forma de

energia do ambiente que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza física que precisa ser mensurada, como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc. (WENDLING, 2010).

Para mensurar a temperatura é utilizado o termômetro digital DS18B20, que é um sensor capaz de mensurar temperaturas na faixa de -55°C até $+125^{\circ}\text{C}$. O mesmo apresenta uma resolução ajustável entre 9 e 12 bits, o que dependendo da aplicação, onde se requer, mas ou menos precisão pode-se escolher o valor e ajustá-lo.

O sensor se comunica com o microcontrolador através do protocolo Wire, desenvolvido pela empresa Texas Instruments, cujo funcionamento consiste em um sinal de dados unidirecional e uma referência, ou seja, envia uma palavra que é lida e interpretada pelo microcontrolador possibilitando a utilização do valor pela aplicação. O consumo de energia do mesmo é baixo. A Figura 7 apresenta um sensor como o utilizado no desenvolvimento do protótipo.

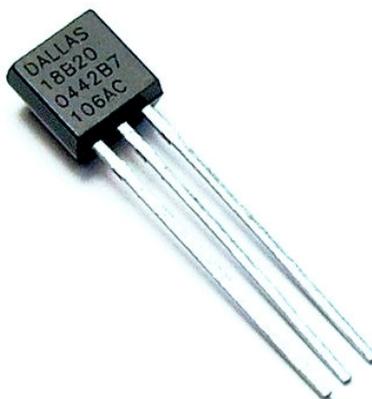


Figura 7 - Sensor de temperatura Dallas DS18B20
Fonte – Dallas Inc.

Outro sinal vital mensurado é a frequência cardíaca, o dispositivo utilizado para este fim é o Pulse Sensor - Figura 8. Este é do tipo fotopletismógrafo não invasivo utilizado para mensurações de taxas cardíacas. Estes sensores são capazes de mensurar o nível de oxigenação no sangue dependendo da aplicação que forem destinados. Ao detectar um pulso no coração este sensor é capaz de reproduzir em forma de onda este pulso e enviar esta onda como um sinal elétrico. Esta onda é recebida pelo microcontrolador que faz o processamento utilizando uma lógica que será apresentada no capítulo quatro.

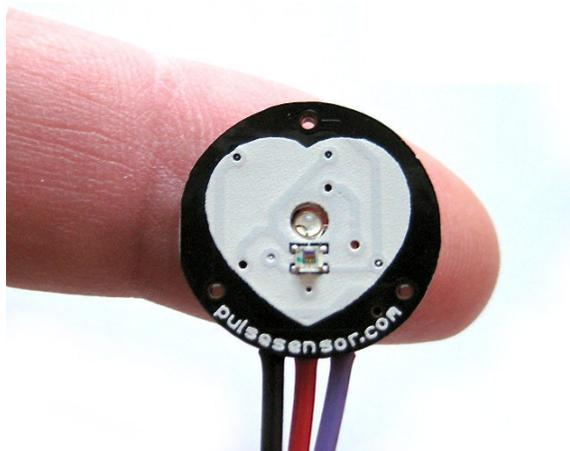


Figura 8 - Pulse Sensor (Sensor Batimentos)
Fonte – Pulse Sensor Inc.

O terceiro sensor utilizado é em um módulo integrado que contém um acelerômetro, um giroscópio e um sensor de temperatura. O acelerômetro é utilizado para identificar quando um movimento acontece, e seu princípio de funcionamento está baseado em medir a aceleração própria do objeto, por meio de cálculos internos, este possibilita saber o posicionamento do mesmo em relação a um referencial pré-estabelecido.

O módulo prove também um sinal de temperatura que é utilizado para mensurar a temperatura do ambiente, e um sinal de giroscópio, o qual não é utilizado no trabalho. O módulo utilizado é o 6050, apresentado pela Figura 9, que possui bloco de memória e armazena as informações coletadas. A troca de informação entre o microcontrolador e o sensor acontece por meio do protocolo I2C bidirecional, contendo um sinal de *clock* para a sincronia e um sinal de dados para a troca de informações.

O módulo possui ainda outras conexões que possibilitam, por exemplo, a ativação do mesmo quando uma interrupção acontece e também o acoplamento de mais módulos para uma melhor precisão.

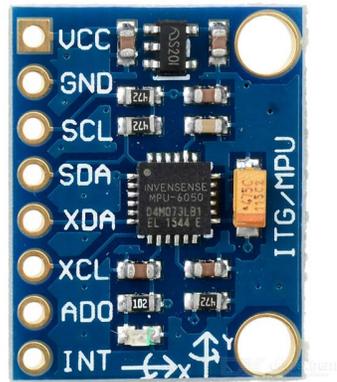


Figura 9 - Módulo MPU6050
Fonte: Adaptado da Internet.

3.1.3 Bateria

Para o protótipo foi utilizada uma bateria recarregável de íon de lítio, semelhante ao apresentado na Figura 10, o qual não possui um custo elevado. Chamado de Power Bank da empresa Sony, com capacidade energética é de aproximadamente 3000mAh. Esta bateria tem um tamanho maior do que as encontradas no mercado, porém para efeitos de prototipação responde bem ao desejado. Provem uma autonomia de aproximadamente 20 horas, ao projeto.



Figura 10 - Power Bank Sony
Fonte – Adaptado da Internet

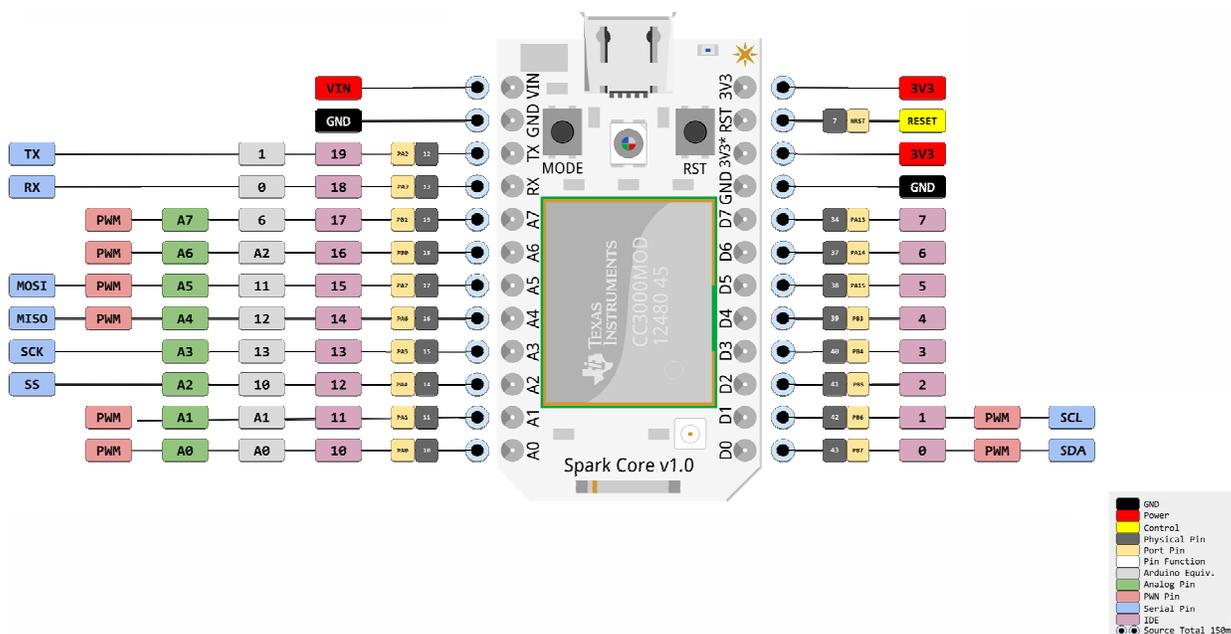
3.1.4 Microcontrolador e Módulo de Comunicação

A empresa Particle IO dispõe de uma ferramenta nomeada Spark core, cuja sua utilidade é a prototipação na área da Internet das Coisas. Este kit é constituído por um microcontrolador ARM Cortex M3 e um módulo Wi-fi SimpleLink™ CC3000 que possibilitam a aquisição de dados e a transmissão pela Internet através do protocolo Wi-Fi 802.11 b/g.

Dentre as principais características do microcontrolador destacam-se:

- ARM 32-bit Cortex™-M3 CPU Core;
- 72Mhz *clock*, 1.25 DMIPS/MHz (Dhrystone 2.1);
- 128KB de memória Flash;
- 20KB de memória SRAM;
- 12 bits conversor ADC;
- USB 2.0 interfaces;
- USART, SPI e I2C interfaces;
- JTAG debug interface.

A Figura 11 apresenta o microcontrolador e seu diagrama de pinos, percebe-se que o mesmo tem uma porta analógica e uma digital. Na Figura também estão representadas as conexões disponíveis em cada pino.



3.1.5 Servidor Ubidots

Para a troca de informações entre o microcontrolador e um dispositivo remoto de monitoramento, como um Smartphone ou Tablet, foi escolhida um serviço chamado de Ubidots, que em sua versão gratuita atende as especificações de projeto. Seu Dashboard, espaço de trabalho, esta representado pela Figura 12.

Quando se fala de Internet das Coisas, a simplicidade deve ser garantida para que o acesso a este servidor seja fácil e rápido. No Ubidots, cada cliente tem uma chave de acesso, compartilhando no ambiente todas variáveis lidas a partir de sensores. A comunicação é feita através do protocolo HTTP onde a *String* com as informações é definida e enviada ou requisitada ao servidor, contudo a versão gratuita possui algumas limitações:

- Não se pode personalizar o tipo de dado a ser armazenado na variável (somente numérico)
- As consultas de eventos passados não podem ser efetuadas, pois as conexões com o servidor, no modo gratuito, retornam somente os últimos 50 valores.

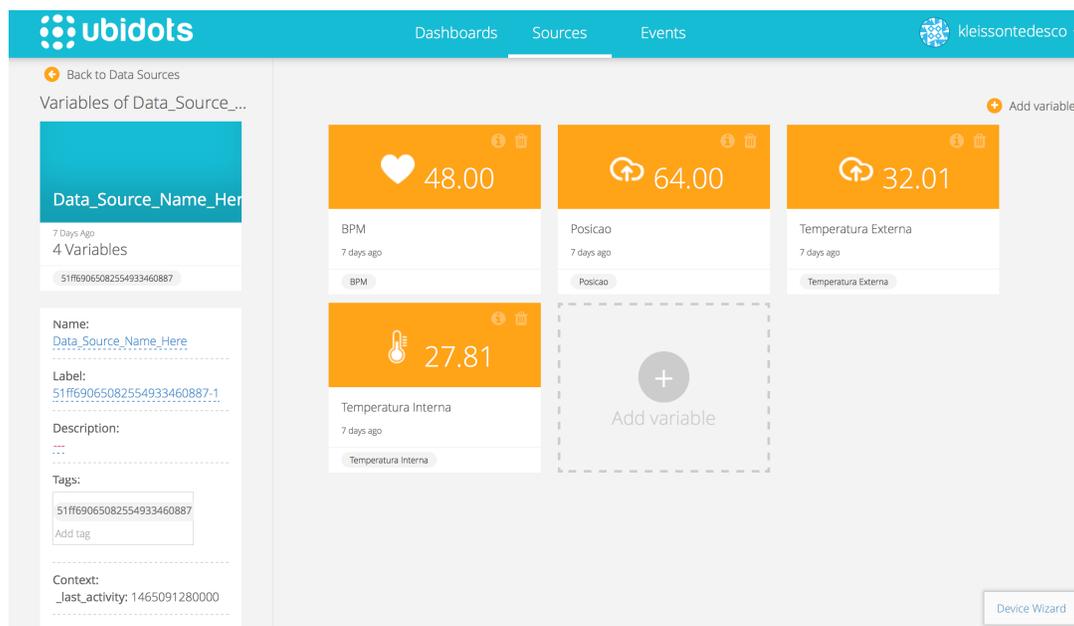


Figura 12 - Dashboard Ubidots
Fonte – Adaptado da Internet.

3.1.6 Plataforma Android

Por ser de código livre, a plataforma Android provê ferramentas destinadas ao desenvolvimento de aplicativos para smartphones e Tablets, sendo estes o foco do presente trabalho. O Android é mantida pela empresa Google, visando permitir aos desenvolvedores criar aplicações que possam tirar total proveito dos recursos que um aparelho portátil oferece.

Comunidades de desenvolvedores trabalham juntas para construir aplicações móveis inovadoras, o que possibilita uma evolução contínua da plataforma. Esta é desenvolvida com base no sistema operacional Linux é composta por um conjunto de ferramentas que atuam em todas as fases do desenvolvimento do projeto (PEREIRA; SILVA, 2009).

Por meio da IDE de desenvolvimento Android Studio é possível programar aplicativos utilizando-se da linguagem Android, sendo disponibilizado pela IDE um ambiente de simulação para testar os aplicativos, este chamado de AVD (*Android Virtual Device*). A programação para a plataforma é baseada em JAVA e XML.

Para a troca de mensagens, é integrado ao Android uma ferramenta chamada Firebase Cloud Messaging. Esta é utilizada para implementar um sistema de notificação do tipo *push*. Este permite que o aplicativo não fique fazendo verificações constantes por mensagens ao servidor, o que poderia consumir toda a bateria e/ou a banda de Internet. Para isso, um servidor seria responsável por fazer estas verificações e enviar as notificações quando necessário.

3.2 MÉTODO

A aquisição de dados é efetuada pelos sensores que estão acoplados a faixa abdominal, onde os sinais são condicionados pelo microcontrolador que por meio de um módulo de comunicação transmite, via protocolo http, os mesmos por *wireless* ao servidor Ubidots. Assim o dispositivo Android busca estes dados no servidor e apresenta ao usuário final em forma de variáveis, apresentando gráficos e ou tabelas dos dados obtidos.

Com base no modelo sequencial linear descrito em Pressman (2009), foi efetuada a modelagem e a implementação do sistema. O modelo sequencial é utilizado para denominar os processos (ou fases). Essas fases foram organizadas em ciclos iterativos de modelagem e implementação.

O Quadro 1 apresenta os processos (fluxos de trabalho) e as iterações que foram desenvolvidas.

Iterações	1ª iteração	2ª iteração	3ª iteração
Processos			
Requisitos	Definição dos componentes básicos.	Complemento dos Materiais.	Revisão dos componentes.
Análise	Análise do funcionamento dos componentes.	Análise e ajuste dos materiais necessários.	Revisão do funcionamento dos componentes
Projeto	Definição de como os componentes serão utilizados.	Ajustes da interface do aplicativo.	Revisão da modelagem.
Implementação	Programação dos componentes.	Implementação do firmware do dispositivo.	Implementação do aplicativo.
Testes	Testes da Programação.	Realização de testes do código do aplicativo.	Realizado testes do código e de interação do sistema.

Quadro 1 - Iterações definidas

Fonte – (PRESSMAN, 2009)

A seguir estão descritas as etapas (identificadas como processos no Quadro 1) definidas para o desenvolvimento do sistema e as principais atividades de cada uma dessas etapas.

a) Pesquisa dos Materiais

O levantamento dos materiais necessários inicia tendo como objetivo desenvolver um dispositivo confortável e usual possibilitando a mensuração dos sinais vitais, identificando os dados a serem mensurados e os sensores necessários assim como microcontroladores e módulos de transmissão. Paralelamente foram definidos os requisitos do aplicativo, que devem ser simples e funcionais.

b) Análise do sistema

Com base nos requisitos estabelecidos foram definidos os casos de uso do sistema. Esses casos possibilitaram a estruturação do mesmo, analisando a viabilidade e a usabilidade do sistema por quem e para qual o mesmo seria desenvolvido.

c) Projeto

Com base nos componentes analisados o projeto do sistema possibilitou um estudo avançado da melhor forma de desenvolver o mesmo. O projeto leva em consideração análises efetuadas de várias ferramentas disponíveis no mercado, definindo quais são as mais apropriadas e que melhor auxiliarão na execução do projeto.

d) Implementação

A implementação consiste basicamente na programação do microcontrolador e do aplicativo Android, utilizando-se dos materiais estudados na fase de projeto. Ocorreu em duas etapas paralelamente. São as etapas o desenvolvendo da faixa abdominal, e o desenvolvimento do aplicativo que tendo em vista os estudos dos requisitos e casos de uso, possibilitando uma forma aceitável da implementação do aplicativo.

e) Testes

Os testes do sistema foram efetuados a representar condições reais de uso, onde foram simuladas variações de temperatura, e de batimentos cardíacos assim como a posição no objeto acoplado a faixa abdominal.

4 RESULTADO

Esta seção detalha a metodologia utilizada no desenvolvimento do projeto, assim como seus aspectos teóricos e práticos.

4.1 ESCOPO DO SISTEMA

O sistema está dividido em duas linhas que trabalham paralelamente: aplicativo para a plataforma Android e o aplicativo para o Microcontrolador, a conexão entre elas é provida pela plataforma Ubidots, que permite a gravação dos dados por parte do microcontrolador, assim como uma leitura por parte do aplicativo Android.

A Figura 13 apresenta uma visão geral do sistema. É possível observar a faixa abdominal fixada no bebê que contém o microcontrolador assim como os sensores. Este recebe os dados, processa e os envia para o Servidor Ubidots, através de um roteador Wireless conectado a Internet.

Esses dados são requisitados pelo aplicativo Android que os recebe e mostra na tela em tempo real os dados numéricos da última leitura, assim como os gráficos com o histórico. Um terceiro elemento no sistema é o servidor JAVA para o push, que verifica estes dados e caso alguma anormalidade seja detectada, envia notificações *push* através da ferramenta Firebase Cloud Messaging.

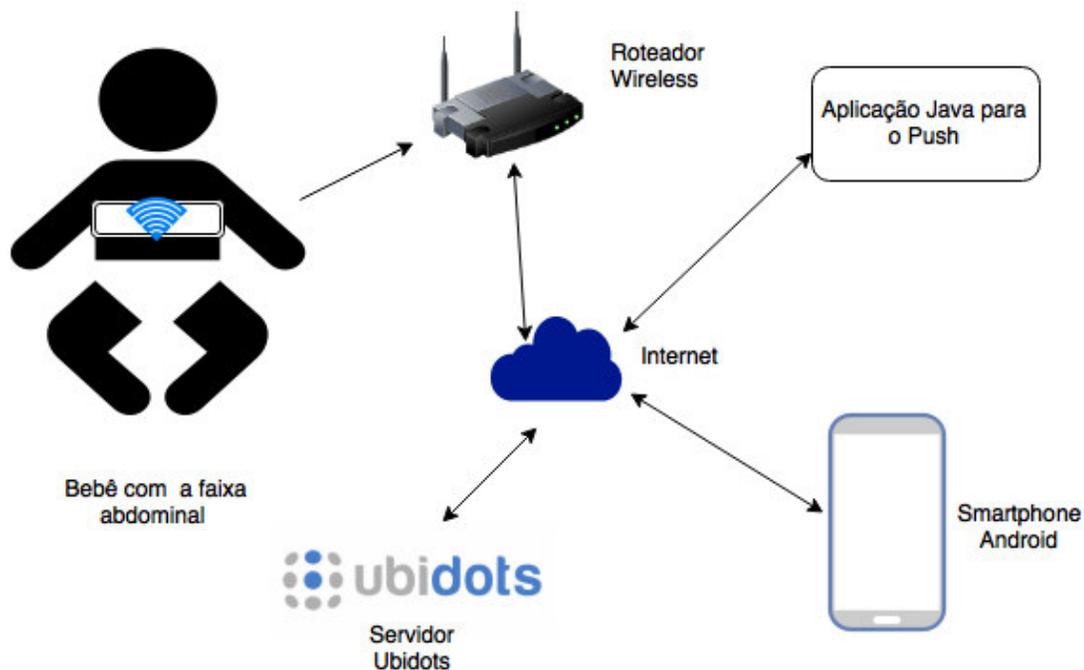


Figura 13 - Escopo do Sistema
Fonte – Autoria Própria.

4.2 MODELAGEM DO SISTEMA

Nesta seção são apresentados como os sensores, assim como o sistema em geral funciona. Diagramas são utilizados para apresentar como cada componente trabalha, e a sua contribuição para o projeto.

O sensor de temperatura é ativado quando o sistema entra em funcionamento, e permanece enviando o valor de temperatura recuperado por meio do protocolo One Wire. No microcontrolador, as leituras destes valores são efetuadas com uma verificação de consistência do dado. Se passar por esta verificação, o dado é preparado para o envio ao Ubidots. A Figura 14 apresenta como este sensor se comunica com o microcontrolador.

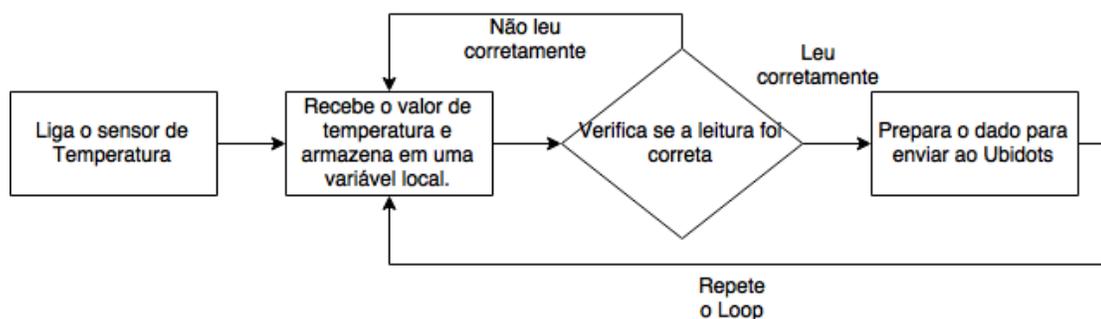


Figura 14 - Diagrama Sensor Temperatura
Fonte – Autoria Própria.

O segundo sensor utilizado é o sensor de pulso, como descrito na seção 3 , este é um sensor fotopletismografico não invasivo responsável por identificar os batimentos cardíacos por meio de sinais infravermelhos e circuitos de instrumentação.

O código no microcontrolador tem o objetivo de medir o tempo entre um batimento e outro, portanto é importante ter uma taxa de amostragem regular com resolução suficiente para obter este tempo entre as batidas. Para isso é utilizado o timer_2 do microcontrolador, que garante uma interrupção a cada 2 milissegundos, permitindo a procura por uma batida e se caso encontrar armazena o momento em que ela ocorreu.

Uma segunda função é ativada a partir desta primeira batida e busca a segunda batida para armazenar seu tempo. Assim, é diminuído o segundo valor do primeiro e se obtém um tempo entre um batimento e outro. Este tempo é utilizado para calcular o valor de batimentos por minuto. Esta lógica é apresentada no Fluxo da Figura 15.

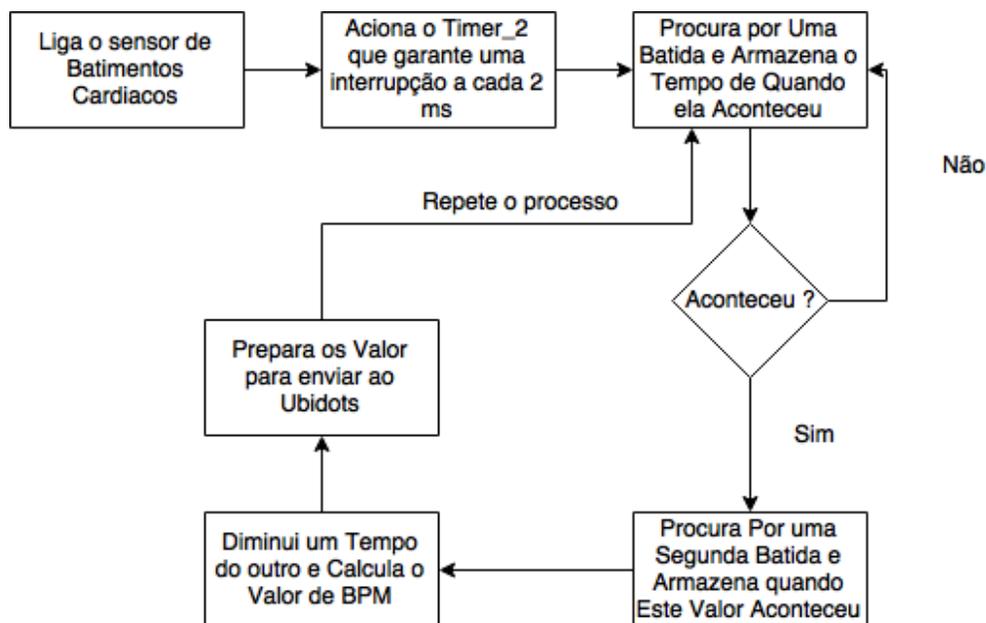


Figura 15 - Diagrama Sensor Batimentos
Fonte – Autoria Própria.

O terceiro sensor é o módulo integrado MPU-6050, utilizado para a aquisição dos sinais de temperatura externa e movimentação. Este módulo funciona com registradores internos que armazenam os dados dos sensores e por meio de uma interface de comunicação disponibilizam estes dados a outros sistemas.

No código do microcontrolador são implementados os protocolos para a leitura destes registradores, assim como os cálculos da temperatura e o ângulo da posição física do módulo. Primeiramente é iniciado o módulo enviando um sinal através do protocolo I2C, e durante o loop uma requisição dos valores é enviada constantemente e o módulo retorna estes

valores que são armazenados em variáveis do microcontrolador. A Figura 16 representa este processo.

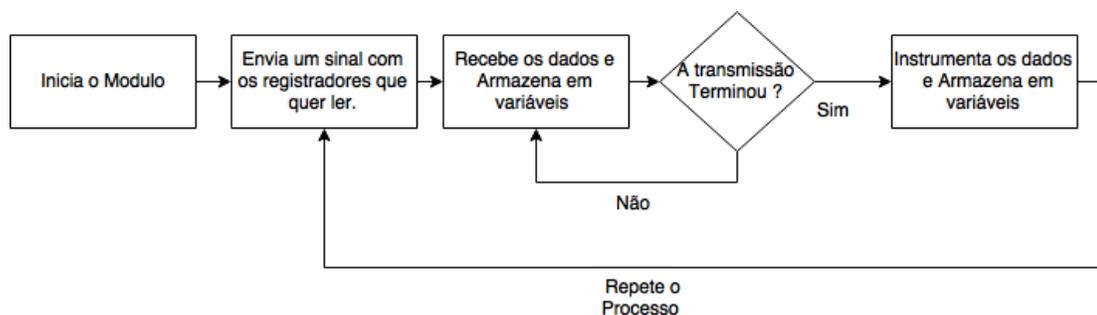


Figura 16 - Diagrama Módulo MPU6050
Fonte – Autoria Própria

De posse dos dados lidos nos sensores, os mesmos são preparados em um texto no formato JSON com suas respectivas ids de variáveis e transmitidos ao servidor Ubidots utilizando-se do protocolo HTTP. A Figura 17 representa este processo em forma de diagrama de Blocos.

JSON é um formato leve para intercambio de dados computacionais, a simplicidade de seu uso é o popularizou este formato no uso e implantação de diversas aplicações.

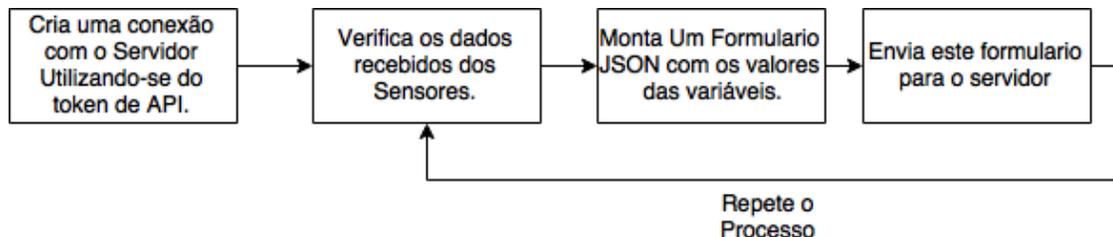


Figura 17 - Diagrama de Blocos Ubidots
Fonte – Autoria Própria

O passo seguinte foi o desenvolvimento do aplicativo Android, utilizando-se da IDE de desenvolvimento Android Studio, e da linguagem de programação Java.

Foi concebido o programa de tal forma que, ao aplicativo ser iniciado, este passa a efetuar requisições ao servidor Ubidots que retorna os valores contidos nas variáveis alimentadas pelo microcontrolador. Estes dados são processados e apresentados na tela em forma de componentes de apresentação de conteúdo e gráficos criados, estes alimentados dinamicamente com os valores buscados.

Ainda, outra aplicação desenvolvida é responsável por enviar requisições *push* caso algum dado esteja fora do padrão pré-estabelecido (temperatura acima da cadastrada como normal, por exemplo). A Figura 18 apresenta o diagrama de blocos de como o aplicativo

funciona, assim como o que acontece quando uma notificação *push* é recebida.

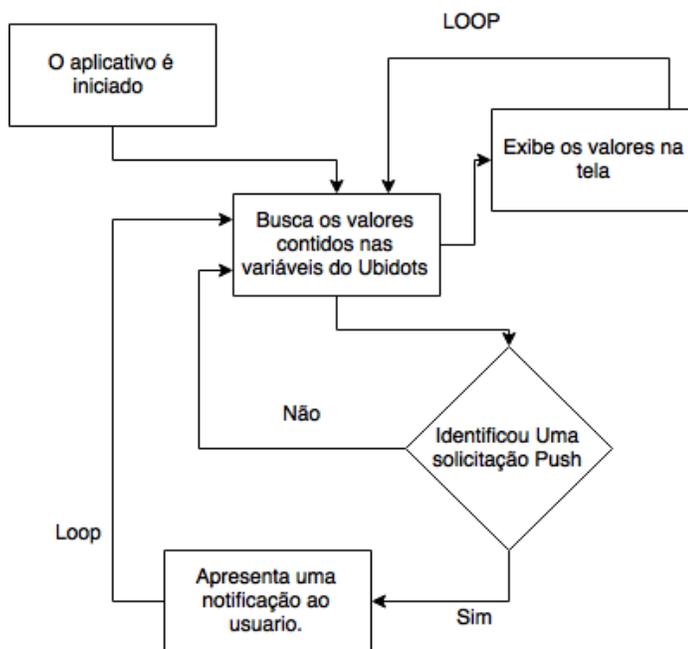


Figura 18 - Diagrama Aplicativo
Fonte – Autoria Própria

A aplicação JAVA para o servidor de push foi desenvolvida utilizando-se a IDE de desenvolvimento Eclipse, e é responsável por fazer verificações constantes no servidor Ubidots, caso algum valor encontrado na variável não esteja de acordo com o parâmetro pré-cadastrado pelo usuário, enviando assim uma notificação por meio da plataforma Firebase Cloud Messaging ao aplicativo Android que interpreta esta informação e exibe ao usuário. A Figura 19 representa um diagrama de como este aplicativo funciona.

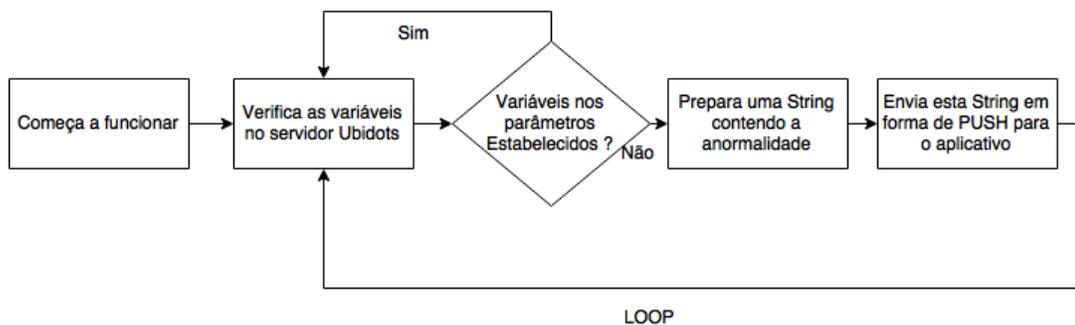


Figura 19 - Diagrama Servidor push
Fonte – Autoria Própria

4.3 APRESENTAÇÃO

Esta seção apresenta a abraçadeira desenvolvida e as telas do sistema Android desenvolvido, assim como a ordem em que as mesmas são apresentadas ao usuário.

A Figura 20 representa a abraçadeira desenvolvida em material de Neoprene e a localização dos sensores na mesma. Esta localização leva em consideração os objetivos do projeto, onde o melhor lugar para os sensores é: Para o de posição no meio do peito, pois permite uma melhor captura de um movimento brusco. Para o de batimentos do lado esquerdo, próximo de onde o coração está localizado. E para o de temperatura, logo abaixo da axila, pois é um bom local para se medir a temperatura corporal. É importante ressaltar que para fins de projeto a abraçadeira foi desenvolvida em um tamanho maior.

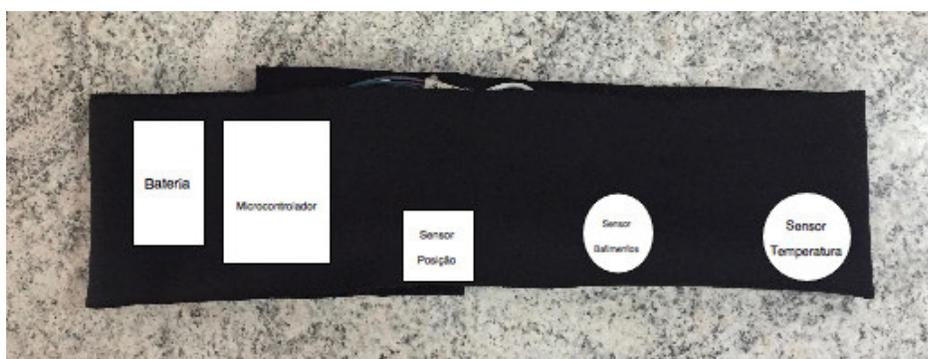


Figura 20 - Faixa Abdominal Desenvolvida
Fonte – Autoria Própria.

A Figura 21 apresenta a tela inicial do aplicativo, esta, já contém todas as informações importantes, para evitar a navegação entre telas. Nesta, são contidos os valores instantâneos de temperatura interna, que representa a temperatura do bebê, a temperatura externa que representa a temperatura do ambiente, os batimentos por minuto e a posição do bebê.

Também em forma de gráficos é apresentado o histórico da leitura. Este gráfico é dinâmico e interativo, podendo o usuário clicar e arrastar para ver valores mais antigos.

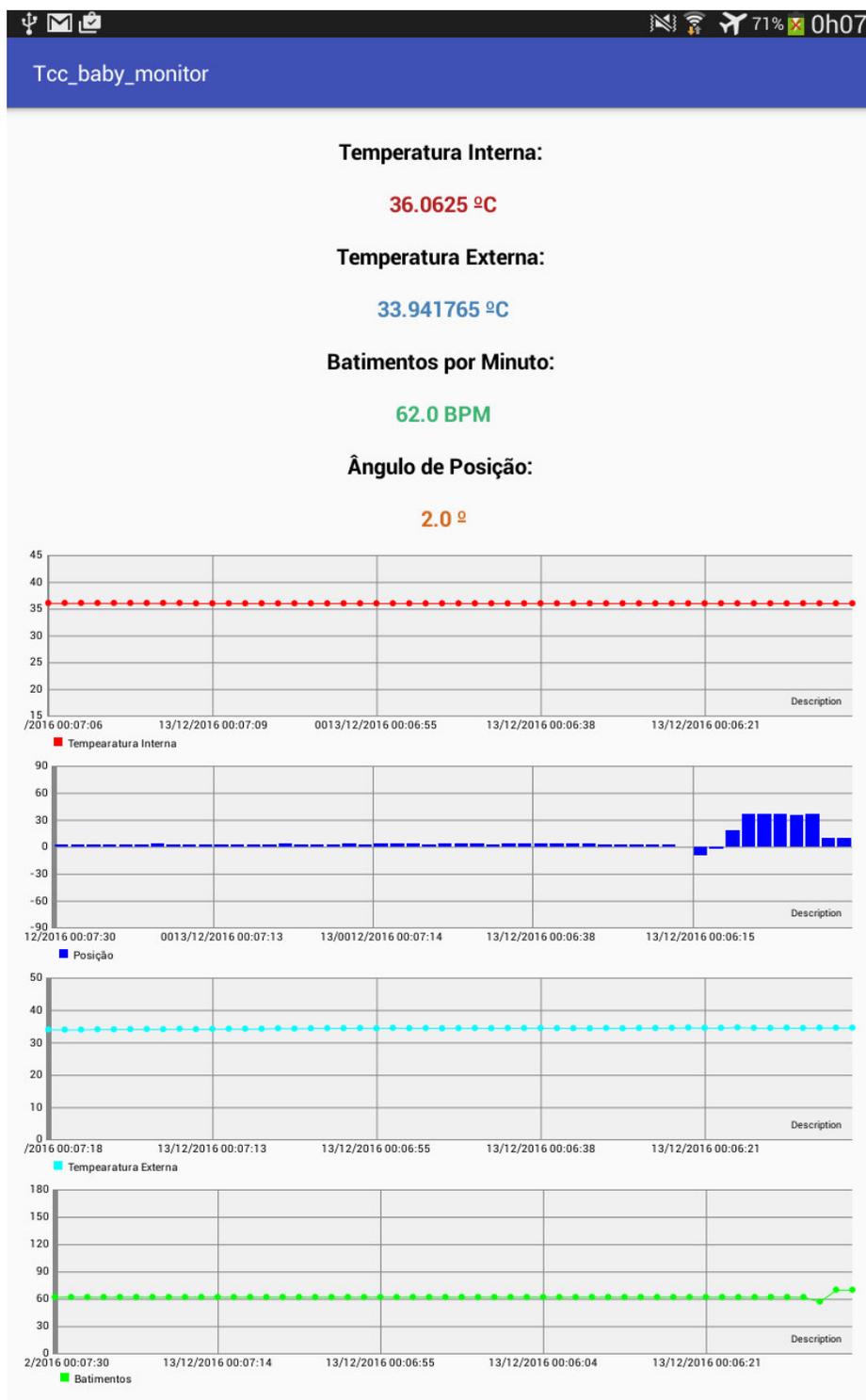


Figura 21 - Tela Aplicativo
Fonte – Autoria Própria

A Figura 22 representa uma notificação sendo recebida. Esta mensagem identifica o remetente com o nome da aplicação, assim como o texto do problema (no exemplo, batimentos fora do padrão cadastrado). Ao clicar sobre a mensagem, o usuário é redirecionado

para a tela principal do aplicativo. Ao receber a notificação, além do efeito sonoro, é realizado também um efeito de vibração do dispositivo. As configurações dos parâmetros de normalidade são realizadas dentro da aplicação que efetua as verificações no servidor Ubidots.



Figura 22 - Notificação sendo recebida
Fonte – Autoria Própria

4.4 IMPLEMENTAÇÃO

Na primeira parte da implementação foi desenvolvido o código do microcontrolador, onde todos os sensores descritos na seção Materiais e Métodos foram estudados e acoplados ao projeto

Para o sensor de temperatura Dallas DS18B20, foram utilizadas duas bibliotecas: spark-dallas-temperature e onewireunofficial, que trabalham juntas e as funções nelas contidas estão descritas na listagem de código 1.

```
#define ONE_WIRE_BUS D2 // Definição De Um Endereço De Memória Para O Sensor
OneWire oneWire(ONE_WIRE_BUS); // Criando O Protocolo De Comunicação Para Aquele Endereço.
DallasTemperature sensor(&oneWire); // Associando O Endereço Ao Sensor.

sensor.begin(); // Iniciando Os Sensor De Temperatura
sensor.requestTemperatures(); // Requisitando A Temperatura Ao Sensor.

temperature=sensor.getTempCByIndex( 0 ); // Armazenando O Valor Recebido Em Uma Variável Local.
```

Listagem 1 – Codigos Biblioteca spark-dallas-temperature e onewireunofficial

Por meio das funções contidas nas duas bibliotecas, o sensor de temperatura consegue se comunicar com o microcontrolador provendo o valor de temperatura lido no ambiente. Este

valor retornado já é em Celsius, e nenhuma conversão é necessária.

Para o sensor de batimentos cardíacos a biblioteca pulseSensor foi utilizada juntamente com a biblioteca SparkIntervalTimer. A primeira permite um cálculo dos valores de batimentos cardíacos por minuto e utiliza-se de funções contidas na segunda para garantir que interrupções acontecerão em certos intervalos de tempo. Todas as funções destas bibliotecas são utilizadas, e o valor de batimentos cardíacos é armazenado em uma variável local, acessada pelo código principal. A mesma se encontra na seção de anexos deste trabalho.

A Listagem 2 contém o código responsável por encontrar o valor de BPM, utilizando-se de funções contidas na própria biblioteca e de uma lógica já apresentada na seção 4.2. A função pulseISR() é a responsável por encontrar a diferença de tempo entre dois batimentos e retornar em uma variável o valor em batimentos por minuto. A mesma é instalada no microcontrolador para acontecer a cada 2 milissegundos garantindo uma precisão ao código.

```
void interruptSetup(void){
  // Aloca A Função No Microcontrolador Para Acontecer A Cada 2ms
  pulseTimer.begin(pulseISR, 2000, uSec); // Passando Os Parâmetros Ao Microcontrolador Para Instalar A
  Função.
}
```

Listagem 2 – Funções Sensor Batimentos.

O terceiro sensor possui bibliotecas de código também, porém transmite seus dados através da interface I2C, e a biblioteca Wire, já nativa no microcontrolador, implementa este protocolo por meio de algumas funções pré-definidas. A Listagem 2 apresenta as funções utilizadas no código providas por esta biblioteca.

```
const int I2C_ADDR = 0x68; //Endereço I2C Do Módulo, Definido No Datasheet

Wire.begin(); // Inicia O Protocolo Wire
Wire.beginTransmission(I2C_ADDR); //Indica Que O Sensor Vai Transmitir Pelo Endereço Passado
Wire.write(0x6B); //Gerenciamento De Energia Do Sensor
Wire.endTransmission(true); // Termina A Transmissão

//código em loop
Wire.beginTransmission(I2C_ADDR); // Indica Que O Sensor Vai Começar Transmitir
Wire.write(0x3B); //Faz Uma Requisição Do Primeiro Registrador De Posição
Wire.endTransmission(false); //Indica Que A Transmissão Não Acabou
Wire.requestFrom(I2C_ADDR, 8, true); //Faz A Requisição De 8 Registradores
AcX = Wire.read() << 8 | Wire.read(); // Primeiros 2 Registradores
AcY = Wire.read() << 8 | Wire.read(); // Terceiro E Quarto Registradores
AcZ = Wire.read() << 8 | Wire.read(); // Quinto E Sexto Registradores
temp = Wire.read() << 8 | Wire.read(); // Sétimo E Oitavo Registradores

Pitch = FunctionsPitchRoll(AcY, AcX, AcZ); // Calcula A Posição No Eixo X, Com Base Nos Valores Do Nos
```

```

Registadores, E Armazena Em Uma Variável Local.

temperaturaExterna = ((float)temp/340.00)+36.53;//Armazena Os Valores Da Temperatura Em Uma Variável
Local.

//Função Utilizada Para Calcular A Posição.

double FunctionsPitchRoll(double A, double B, double C){
double DatoA, DatoB, Value;

//Formula Contida No Datasheet
DadoA = A;
DadoB = (B*B) + (C*C);
DadoB = sqrt(DadoB);

Value = atan2(DadoA, DadoB);
Value = Value * 180/3.14;

return (int)Value;// Retorna O Ângulo Calculado.
}

```

Listagem 3 - Código módulo MPU6050

As requisições feitas ao módulo retornam valores armazenados nos registradores do mesmo, e com base nas fórmulas encontradas no datasheet os cálculos de ângulo e temperatura externa são efetuados e armazenados em variáveis no microcontrolador.

Com os valores armazenados nas variáveis, o passo seguinte é a transmissão para o servidor Ubidos, que armazena os mesmos na Internet. Esta transmissão é feita utilizando-se da biblioteca Ubidots, e as funções contidas nela que são utilizadas neste trabalho são descritas na Listagem 3.

Para armazenar estes valores a aplicação possui uma chave, e cada variável um token, por meio deste código é possível sincronizar estes dados lidos em suas respectivas variáveis no servidor, representadas por estes tokens.

```

#define TOKEN "fEHilGyGocq6kP9A7BD9GzbyddpIxm" // Token Da Api, Obtido No Ubidots.
Ubidots ubidots(TOKEN); // Cria Um Espaço Para Armazenar Os Valores Das Variáveis.

//Aqui Será Montado Um Formulário Com Os Dados Das Variáveis

ubidots.add("Posicao ", angulo); // Valor Do Angulo
ubidots.add("Temperatura Externa", temperaturaExterna);//Valor Da Temperatura Externa
ubidots.add("Temperatura Interna", tempCalculada);//Valor Da Temperatura Interna
ubidots.add("BPM", BPM);//Valor Dos Batimentos Cardíacos

ubdots.sendAll; //Envia Este Formulário Para O Servidor.

```

Listagem 4 - Código Ubidots

Esta verificação dos valores e o armazenamento no Ubidots acontece a cada minuto, e utiliza-se de um formulário do tipo JSON, que monta em forma de *strings*, os dados serem enviados. O envio acontece pelo protocolo HTTP que esta implementado no firmware do microcontrolador. Este protocolo, ainda que não tão seguro quanto o HTTPS, cumpre com os objetivos propostos durante o desenvolvimento do trabalho. Para futuras implementações uma troca pelo protocolo HTTPS foi considerada.

O desenvolvimento do aplicativo Android utilizou consultas do servidor do Ubidots por meio da classe UbidotsClient, que implementa uma conexão com o servidor Ubidots através da biblioteca okhttp. Esta classe monta um formulário contendo as informações de token de api e id da variável do Ubidots, e este formulário é enviado como forma de requisição ao servidor que retorna o valor contido nesta variável assim como um timestamp representando a hora em que aquela variável foi gravada.

A Listagem 5 apresenta esta classe e o retorno da mesma é um vetor com 50 valores, contidos nas variáveis assim com 50 valores de timestamp.

```
* Copyright (C) 2015, francesco Azzola
*
*(http://www.survivingwithandroid.com)
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*
* 05/12/15
*/

import android.util.Log;

import com.squareup.okhttp.Callback;
import com.squareup.okhttp.OkHttpClient;
import com.squareup.okhttp.Request;
import com.squareup.okhttp.Response;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class UbidotsClient {
```

```

private UbiListener listener;

public UbiListener getListener() {
    return listener;
}

public void setListener(UbiListener listener) {
    this.listener = listener;
}

public void handleUbidots(String varId, String apiKey, final UbiListener listener) {

    final List<Value> results = new ArrayList<>();

    OkHttpClient client = new OkHttpClient();
    Request req = new Request.Builder().addHeader("X-Auth-Token", apiKey)
        .url("http://things.ubidots.com/api/v1.6/variables/" + varId + "/values")
        .build();

    client.newCall(req).enqueue(new Callback() {
        @Override
        public void onFailure(Request request, IOException e) {
            Log.d("Chart", "Network error");
            e.printStackTrace();
        }

        @Override
        public void onResponse(Response response) throws IOException {
            String body = response.body().string();
            Log.d("Chart", body);

            try {
                JSONObject jsonObj = new JSONObject(body);
                JSONArray jRes = jsonObj.getJSONArray("results");
                for (int i=0; i < jRes.length(); i++) {
                    JSONObject obj = jsonObj.getJSONObject(i);
                    Value val = new Value();
                    val.timestamp = obj.getLong("timestamp");
                    val.value = (float) obj.getDouble("value");
                    results.add(val);
                }
                listener.onDataReady(results);
            }
            catch (JSONException jse) {
                jse.printStackTrace();
            }
        }
    });
}

protected static class Value {
    float value;
    long timestamp;
}

protected interface UbiListener {
    public void onDataReady(List<Value> result);
}

```

```

    }
}

```

Listagem 5 - Classe Ubidots Client

Esta classe foi implementada e a função UbidotsClient() apresentada na Listagem 5 é responsável por verificar estes dados na classe principal da aplicação Android. O código descrito apresenta os batimentos cardíacos que são recuperados do banco de dados e apresentados no aplicativo, primeiro em forma de componente de Text, e depois um gráfico utilizando a biblioteca MPAndroidChart.

```

( new UbidotsClient() ).handleUbidots(bpmk, api_key, new UbidotsClient.UbiListener() { // instancia a conexão
com o Ubidots Utilizando a classe Uidots Client
@Override
public void onDataReady(List<UbidotsClient.Value> result) { // enquanto a conexao esta estapelescida

List<Entry> entries = new ArrayList();// vetor para armazenar os dados
List<String> labels = new ArrayList<String>();// vetor para armazenar os timestamps
for (int i=0; i < result.size(); i++) { // for para preencher estes vetores

Entry be = new Entry(result.get(i).value, i);// adicionando o valor em um auxiliar chamado be
entries.add(be);// adiciona o valor no vedor

// auxiliar para converter o timestamp em data
Date d = new Date(result.get(i).timestamp);
// adicionando as datas no vetor
labels.add(sdf.format(d));

//aqui é onde se busca o valor para atualizar o textView
bpm = String.valueOf(result.get(result.size()-48).value);

// Handler para atualizar o textview

Handler handler = new Handler(FragmentosGraficos.this.getActivity().getMainLooper());
handler.post(new Runnable() {

@Override
public void run() {
txt_bpm.setText(bpm+" BPM");

}
});

}

// Monta o gráfico utilizando de funções contidas na biblioteca MPAndroidChart

LineDataSet lse = new LineDataSet(entries, "Batimentos");

lse.setDrawHighlightIndicators(false);
lse.setDrawValues(false);
lse.setColor(Color.GREEN);
lse.setCircleColor(Color.GREEN);
lse.setLineWidth(1f);
lse.setCircleSize(3f);
lse.setDrawCircleHole(false);
lse.setFillAlpha(65);

```

```

lse.setFillColor(Color.GREEN);

LineData ld = new LineData(labels, lse);

BPM_Chart.setData(ld);
//handler para plotar o gráfico

Handler handler = new Handler(FragmentosGraficos.this.getActivity().getMainLooper());
handler.post(new Runnable() {

    @Override
    public void run() {
        BPM_Chart.invalidate();
    }
});
});
});

```

Listagem 6 - Código Aplicativo

Para o desenvolvimento do programa que envia notificações de push, foi utilizada a plataforma Firebase Messaging. Primeiramente foi efetuado um cadastro na ferramenta e um arquivo com o formato JSON foi gerado e adicionado ao projeto. Este arquivo contém dados da conta e uma chave responsável por gerenciar as notificações. Logo a classe FirebaseMessagingService apresentada na Listagem 7 foi adicionada ao projeto. Esta classe é responsável por gerenciar as mensagens recebidas através da plataforma do Google, e emitir um alerta no dispositivo móvel.

```

public class FirebaseMessagingService extends FirebaseMessagingService {

    private static final String TAG = "MyFirebaseMsgService";

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        Log.d(TAG, "De: " + remoteMessage.getFrom());
        //Log.d(TAG, "Mensagem: " + remoteMessage.getNotification().getBody());
        Log.d(TAG, "Mensagem: " + remoteMessage.getData());

        sendNotification(remoteMessage.getData().toString());

        //sendNotification(remoteMessage.getData().put("", "message"));
    }

    private void sendNotification(String messageBody) {
        Intent intent = new Intent(this, MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
            PendingIntent.FLAG_ONE_SHOT);

        Uri defaultSoundUri= RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
    }
}

```

```

NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this)
    .setSmallIcon(R.mipmap.ic_launcher)
    .setContentTitle("Tcc_Baby_Monitor")
    .setContentText(messageBody)
    .setAutoCancel(true)
    .setSound(defaultSoundUri)
    .setVibrate(new long[] { 1000, 1000, 1000, 1000, 1000 })
    .setContentIntent(pendingIntent);

NotificationManager notificationManager =
    (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

notificationManager.notify(0, notificationBuilder.build());
}
}

```

Listagem 7 - Classe FirebaseMessagingService

A classe FirebaseInstanceIdService apresentada na Listagem 8 foi adicionada e ela é responsável por gerenciar os tokens de cada dispositivo, assim para enviar uma notificação por *push*, através da plataforma, é necessário se obter este token do dispositivo assim como a chave da api.

```

public class FirebaseInstanceIdService extends FirebaseInstanceIdService {

    private static final String TAG = "MyFirebaseIIDService";

    @Override
    public void onTokenRefresh() {
        // Get updated InstanceID token.
        String refreshedToken = FirebaseInstanceId.getInstance().getToken();
        Log.d(TAG, "Refreshed token: " + refreshedToken);

        // TODO: Caso queira armazenar o Token em um servidor próprio, utilize o método abaixo.
        sendRegistrationToServer(refreshedToken);
    }

    private void sendRegistrationToServer(String token) {
        //Crie aqui a implementação de envio do Token ao servidor

        // System.out.println("token: "+token);
    }
}

```

Listagem 8 - Classe FirebaseInstanceIdService

Estas funções foram registradas no arquivo android_manifest.xml e rodam em background quando o aplicativo é iniciado. Surge assim a necessidade de uma aplicação que fique verificando os dados no ubidots e envie alertas por meio do FCM ao aplicativo instalado em um dispositivo.

Utilizando-se da plataforma Eclipse, foi desenvolvido um programa que

constantemente faz estas verificações. Foi adicionado ao projeto a biblioteca `ubidots-master` que possibilita a criação de uma conexão direto com o servidor Ubidots. A Listagem 9 apresenta a chamada da biblioteca, assim como a obtenção das variáveis.

```
ApiClient api = new ApiClient("673950841e99fcf83b9437835636808040288487");// Instancia Uma Conexão
Com O Servidor Passando O Código De Api.

Variable BPM = api.getVariable("57296d3c7625421d154bc2be");//Busca No Servidor A Variável Pelo
Código.

Value[] values_BPM = BPM.getValues();// Salva em um vetor os valores retornados do Ubitods.
```

Listagem 9 - Servidor Verificação *Push*

De posse dos valores retornados pelo servidor é feita uma verificação se os mesmos estão nos parâmetros estabelecidos, citados na sessão 2.1 e caso algum valor não atenda a esta condição, o programa envia uma notificação utilizando-se dos métodos contidos no código representado pela Listagem 10. O código completo esta dentro da classe `EnviaPush` e foi obtido do livro *Google Android – Aprenda a criar aplicações para dispositivos moveis* que inicialmente foi escrito para a plataforma *Google Cloud Messaging* a qual esta em processo de desativação pela empresa. O mesmo foi atualizado e as modificações funcionaram na nova plataforma (LECHETA, 2015).

```
Map<String,String> params = new HashMap<String,String>();// Cria Uma String Dupla
params.put("msg","temperatura interna anormal");// Adiciona A Mensagem
String result = post(API_KEY, DEVICE_REGISTRATION_ID, params);//Posta No Servidor Google
```

Listagem 10 - Post no FCM

4.5 TESTES

Os testes em bebês não foram possíveis pois há um fator burocrático que poderia atrasar desenvolvimento do projeto. Este fator retrata-se ao comitê de ética da universidade, que pode levar um tempo maior do que o tempo hábil para analisar e permitir que estes testes sejam efetuados.

Contudo, os testes foram realizados durante a implementação em ambiente controlado no autor deste trabalho, cada sensor foi adicionado ao microcontrolador e testado. De posse do sistema completamente funcional, o mesmo foi ligado e utilizado por algumas horas. Logo foi possível verificar, pela Figura 23 que o sistema armazenou os dados e os mostrou em forma de gráficos no aplicativo.

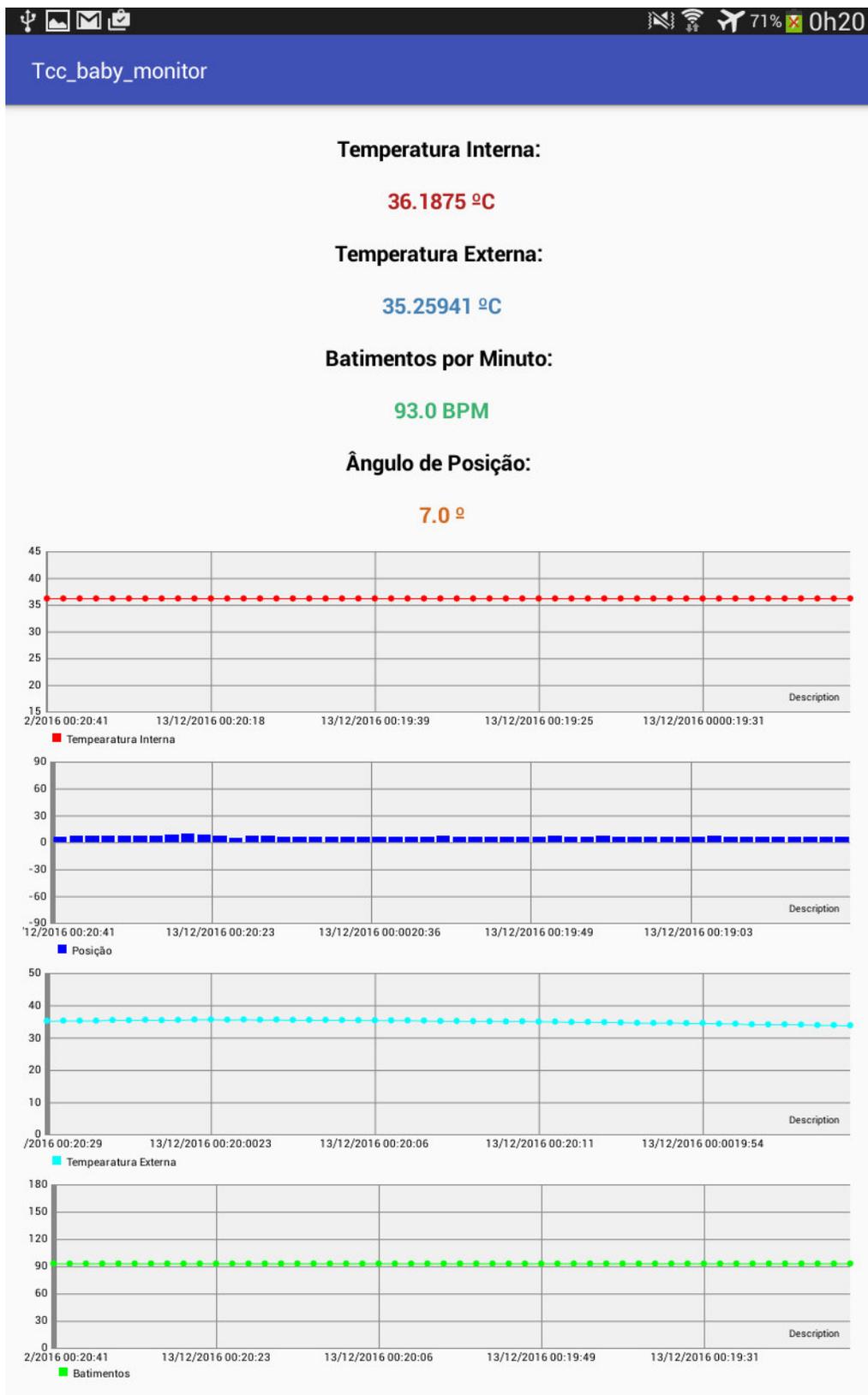


Figura 23 - Testes
Fonte - Autoria Própria

Futuramente como ampliação do projeto, estes testes podem ser efetuados em bebês, o que permitira uma maior aplicabilidade e um melhor relato sobre o comportamento do dispositivo.

5 CONCLUSÃO E TRABALHOS FUTUROS

Este projeto realizou a integração de hardware e software afim de auxiliar na prevenção de um problema que afeta milhares de crianças todos os anos: a morte súbita, assim como auxiliar os pais no monitoramento da temperatura e dos batimentos cardíacos dos recém-nascidos, assim como a intensidade de suas movimentações. Este pode ser utilizado pelos pais preocupados com o estado de saúde dos filhos ou ainda pelos profissionais de saúde que desejam manter um registro de parâmetros fisiológicos sobre seus pacientes.

O sistema implementado apresenta os dados lidos dos sensores que representam sinais vitais de um bebê por meio da plataforma Android, e notifica este caso algum dado não esteja nos padrões de normalidade.

Durante o desenvolvimento, algumas limitações foram encontradas, como por exemplo as limitações presentes na versão gratuita do servidor Ubidots, o qual não possibilita que esta solução seja aplicada no mercado, pois ao se expandir o número de usuários do sistema não é possível utilizá-lo (apenas para o monitoramento de um microcontrolador), já que o mesmo é destinado aplicações pequenas.

Contudo, estas limitações não impediram o cumprimento dos objetivos tanto geral como específicos. A aquisição de dados utilizando os sensores foi implementada através do microcontrolador, que por sua vez permitiu a transmissão de dados ao servidor. O desenvolvimento do aplicativo levou em consideração fatores como usabilidade, dinamicidade e possibilitou a visualização dos sinais vitais lidos nos sensores. Também para o desenvolvimento da faixa abdominal foram utilizados materiais com dimensões reduzidas e que consomem pouca energia, voltados para a Internet das Coisas. Esta solução utilizou sensores, um microcontrolador e ferramentas de programação gratuitas.

Como trabalhos futuros, deseja-se adicionar mais sensores, pressão sanguínea por exemplo, ampliando a usabilidade do dispositivo e abrindo um leque maior na monitoração dos sinais vitais com base nos padrões de normalidade. Também encontrar uma solução para autenticação de usuários, permitindo uma maior privacidade nos dados gravados no servidor e também uma disponibilização destes dados em plataformas diferentes, como um site de internet por exemplo.

Ainda, deseja-se ampliar o numero de utilizadores, pois o sistema pode auxiliar

indivíduos de qualquer idade que sofram de problemas de saúde e um monitoramento contínuo melhora o diagnóstico/tratamento de doenças cardiovasculares.

A inovação proposta neste trabalho pode mudar o mercado de equipamentos de saúde, já que integra novas tecnologias na monitoração de sinais fisiológicos com smartphones e tablets. Esta solução deve tornar mais comum o cuidado com a saúde até mesmo por pessoas saudáveis. Além disso, a implementação bem-sucedida deste projeto possibilita que futuramente outros métodos e sensores sejam incorporados.

Por fim o desenvolvimento do projeto aplicou os conhecimentos obtidos durante o curso, fixando e ampliando os conceitos de computação nos dias atuais.

REFERÊNCIAS

ACHARYA, U. R. et al. Automated Prediction of Sudden Cardiac Death Risk Using Kolmogorov Complexity and Recurrence Quantification Analysis Features Extracted from HRV Signals. **Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015**, p. 1110–1115, 2016.

ALVES, J. G. B. A. et al. **Pediatria**. 3. ed. Rio de Janeiro - RJ: Guanabara, 2004.

CENTER FOR DISEASE CONTROL AND PREVENTION. **Sudden Unexpected Infant Death and Sudden Infant Death Syndrome**. Disponível em: <<http://www.cdc.gov/art/>>. Acesso em: 15 abr. 2016.

DIAS, J. A. B. D. S. **Desenvolvimento de aparato de suporte à locomoção aquática de bebês e de crianças de 3 a 24 meses**, 2013. (Nota técnica).

FONSECA, A. M. et al. A sudden infant death prevention system for babies. **2014 IEEE 16th International Conference on e-Health Networking, Applications and Services, Healthcom 2014**, p. 525–530, 2015.

GEIB, L. T. C.; NUNES, M. L. Hábitos de sono relacionados à síndrome da morte súbita do lactente: Estudo populacional. **Cadernos de Saude Publica**, v. 22, n. 2, p. 415–423, 2006.

GREER, R. et al. Remote, real-time monitoring and analysis of vital signs of neonatal graduate infants. **Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE**, p. 1382–1385, 2014.

HEATHER, K. **Is this the future of parenting?** Disponível em: <<http://edition.cnn.com/2014/08/12/tech/innovation/wearable-baby-monitors/>>.

KAHL, M.; FLORIANO, D. **COMPUTAÇÃO UBÍQUA, TECNOLOGIA SEM LIMITES**. 2011.

LECHETA, R. R. **Google Android - Aprenda a criar aplicações para dispositivos móveis**. [s.l: s.n.].

LUCCHINI, M. et al. Multi - parametric Heart Rate Analysis in Premature Babies exposed to Sudden Infant Death Syndrome risk. **Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE**, p. 6389–6392, 2014.

MAGDA, D.; NUNES, L. Síndrome da Morte Súbita do Lactente Epidemiológicos , Fisiopatologia e Prevenção. **Medicina**, 2005.

MALLIK, N. 3 Lesser-Known uses of Beacons – Breaking the Internet of Things (IoT) Barrier. 2015.

MALTA, D. C. et al. Atualização da lista de causas de mortes evitáveis (5 a 74 anos de idade) por intervenções do Sistema Único de Saúde do Brasil. **Epidemiologia e Serviços de Saúde**, v. 20, n. 3, p. 409–412, 2011.

MANSOR, H.; SHUKOR, M. Body temperature measurement for remote health monitoring system. **Smart Instrumentation, Measurement and Applications (ICSIMA), 2013 IEEE International Conference on**, n. November, p. 26–27, 2013.

MELOROSE, J.; PERROY, R.; CAREAS, S. Definition of “Wearable Computer”. **Statewide Agricultural Land Use Baseline 2015**, v. 1, 2015.

MURTA, G. F.; RUSSI, J. N. **Procedimentos Básicos de Enfermagem no Cuidar**. São Caetano do Sul: Difusao Editora, 2006.

NEUMAN, M. R. Measurement of vital signs: Temperature. **IEEE Pulse**, v. 1, n. October, 2010.

PEREIRA, L. C. O. P.; SILVA, M. L. **Android Para Desenvolvedores**. Rio de Janeiro - RJ: BRASPORT, 2009.

PRESSMAN, R. S. **Software Engineering A Practitioner’s Approach**. 7. ed. New York - NY: McGraw-Hill, 2009.

RANDALL, B. B. et al. Potential Asphyxia and Brainstem Abnormalities in Sudden and Unexpected Death in Infants. **Pediatrics**, v. 132, n. 6, p. e1616–e1625, 2013.

REIS, L. M. DOS; CORDEIRO, J. A.; CURY, P. M. Análise da prevalência de morte súbita e os fatores de riscos associados: estudo em 2.056 pacientes submetidos a necropsia. **Jornal Brasileiro de Patologia e Medicina Laboratorial**, v. 42, n. 4, p. 299–303, 2006.

RENATO, A. et al. Monitoramento de Sinais Vitais. v. 2, n. 1, p. 1–32, 2013.

SISTEMA UNICO DE SAUDE. Painel de Monitoramento da Mortalidade Infantil e Fetal. 2016.

VAN LAERHOVEN, K.; SCHMIDT, A.; GELLERSEN, H.-W. Multi-sensor context aware clothing. **Proceedings. Sixth International Symposium on Wearable Computers**, p. 49–56, 2002.

WENDLING, M. **Sensores**, 2010. Disponível em: <<http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>>. Acesso em: 10 out. 2016

WORLD HEALTH ORGANIZATION. **Under-five mortality**. Disponível em: <http://www.who.int/gho/child_health/mortality/mortality_under_five_text/en/>.