

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE CIÊNCIAS - CAMPUS BAURU

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PAULO EDUARDO MANZONE MAIA

APLICATIVO PARA MONITORAMENTO DO SONO

BAURU

2019

PAULO EDUARDO MANZONE MAIA

APLICATIVO PARA MONITORAMENTO DO SONO

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.

Orientador: Prof. Dr. Wilson Massashiro Yonezawa

Paulo Eduardo Manzone Maia Aplicativo para monitoramento do sono/
Paulo Eduardo Manzone Maia. – Bauru, 2019- 40 p. : il. (algumas color.) ;
30 cm.

Orientador: Prof. Dr. Wilson Massashiro Yonezawa

Trabalho de Conclusão de Curso – Universidade Estadual Paulista “Júlio de
Mesquita Filho”

Faculdade de Ciências

Bacharelado em Ciência da Computação, 2019.

1. Desenvolvimento híbrido 2. API Rest 3. Auxílio ao sono 4. Machine Learning
5. No SQL

Paulo Eduardo Manzone Maia

Aplicativo para monitoramento do sono

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. Wilson Massashiro Yonezawa

Orientador

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

Profa. Dra. Simone Domingues Prado

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

Prof. Associado João Pedro Albino

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

Bauru, _____ de _____ de _____.

Dedico este trabalho aos pequenos e gigantes sobre os quais me apoio.

Agradecimentos

Agradeço primeiramente à minha família, por ser minha base, por ter me criado e ter me apoiado nas minhas decisões. Especialmente à minha mãe e à minha irmã, Maurinda Manzone Maia e Camila Manzone Maia, que sempre estiveram ao meu lado e contribuíram diretamente para a realização deste trabalho e para toda minha formação.

Agradeço aos meus professores, por toda a sua dedicação, por terem me passado todo o conhecimento que adquiri durante todos estes anos e por terem me aconselhado nos momentos necessários.

Agradeço aos meus amigos que sempre estiveram ao meu lado no decorrer de toda minha formação, e que juntos desfrutamos de bons momentos e nos apoiamos nos momentos ruins.

Agradeço ao meu orientador Prof. Dr. Wilson Massashiro Yonezawa por abraçar a ideia deste trabalho, por me guiar durante o desenvolvimento e por todo o esforço despendido nesse processo.

Agradeço a todas as pessoas que, mesmo que indiretamente, afetaram positivamente a minha vida, em todos os sentidos. Pelo que se vê, e pelo que não se vê.

*"O Passado pode doer.
Mas do jeito que eu vejo, você pode fugir dele,
ou aprender com ele."
(Rafiki)*

Resumo

Os hábitos cotidianos modernos vêm cada vez mais influenciando a saúde das pessoas. A qualidade geral do sono é um fator muito importante na qualidade de vida de um ser humano e tem impacto direto em sua produtividade e na própria sociedade. É grande a fatia da população que sofre com problemas relacionados à baixa qualidade do sono e esse tema vem se tornando cada vez mais relevante. Nesse sentido, surgiram diversas aplicações móveis que vêm tentando auxiliar o sono, visando a melhora dessa qualidade, seja monitorando o sono, melhorando o ambiente ou melhorando os hábitos de seus usuários. Este trabalho propõe o desenvolvimento de um aplicativo de monitoramento do sono através de sensor de smartphone, aliado a um sistema de recomendação que disponibiliza conteúdos científicos sobre o sono com base em informações obtidas do usuário como regularidade e quantidade de sono, de forma mais clara para a população, utilizando tecnologias de desenvolvimento de aplicações híbridas.

Palavras-chave: Sono, Desenvolvimento Híbrido, Aplicativos móveis.

Abstract

The modern daily habits are increasingly influencing the people health. The overall sleep quality is a very important factor in the human being life quality and has direct influence on his productivity and on the society itself. The proportion of the population that suffers from problems related to poor sleep quality is already large and this theme is becoming increasingly relevant. In this sense, Many mobile applications have emerged that try to help with it, aiming to improve the sleep quality, whether monitoring the sleep, improving the environment or improving the users habits. This work proposes the development of a smartphone sensor sleep monitoring application, combined with a recommendation system, that provides scientific content about sleep based on user informations like sleep regularity and quantity, in a clearer way to the population, using hybrid application development technologies.

Keywords: Sleep, Hybrid Development, Mobile Applications.

Lista de figuras

Figura 1 – Sleep as Android.	14
Figura 2 – Acelerômetro MEMS.	18
Figura 3 – Giroscópio MEMS.	18
Figura 4 – Crescimento das lojas de aplicativo.	21
Figura 5 – <i>Dashboard</i> Firebase.	24
Figura 6 – <i>Dashboard</i> ElasticBeanstalk.	25
Figura 7 – Arquitetura da aplicação.	27
Figura 8 – Módulos do aplicativo.	28
Figura 9 – Mecanismo de tratamento do sensor.	28
Figura 10 – Mecanismo de tratamento do sensor ao ser ativado.	29
Figura 11 – Visualização de dados após disponibilização.	29
Figura 12 – Tela de <i>login</i>	30
Figura 13 – Tela principal.	31
Figura 14 – Tela qualidade geral.	31
Figura 15 – Tela de gráfico.	32
Figura 16 – Tela de configuração.	33
Figura 17 – Tela de dica do dia.	33
Figura 18 – Exemplo de saída.	35
Figura 19 – Saída esperada.	36
Figura 20 – Saída com erros.	36

Lista de quadros

Quadro 1 – Categorias de algoritmos de <i>Machine Learning</i>	20
--	----

Lista de abreviaturas e siglas

API	Application Programming Interface
ARM	Advanced RISC Machine
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
HTTP	Hypertext Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
ICSD	International Classification of Sleep Disorders
LSTM	Long Short Term Memory
MEMS	Micro Electrical Mechanical Systems
PCA	Principal Component Analysis
RAM	Random Access Memory
REST	Representational State Transfer
RISC	Reduced Instruction Set Computer
S3	Simple Storage Service
SRI	Sleep Regularity Index
URI	Uniform Resource Identifier
WSGI	Web Server Gateway Interface

Sumário

1	INTRODUÇÃO	13
1.1	Aplicativos	13
1.2	Problema	14
1.3	Justificativa	15
1.4	Objetivos	15
2	FUNDAMENTAÇÃO	16
2.1	Sono	16
2.1.1	Regularidade	16
2.2	Smartphones	17
2.2.1	Sensor	17
2.3	Aprendizado de Máquina	19
3	FERRAMENTAS E TECNOLOGIAS	21
3.1	Flutter e Desenvolvimento <i>Mobile</i>	21
3.2	REST	22
3.3	Python e Flask	23
3.4	Firebase	23
3.5	AWS	24
4	MÉTODOS	26
4.1	Análise de Distúrbios	26
4.2	Design da Aplicação	26
4.3	Transferência e Visualização de Dados	34
4.4	Recomendador	34
5	ANÁLISES E RESULTADOS	35
5.1	Captação	35
5.2	Visualização	36
5.3	Recomendação	37
6	CONSIDERAÇÕES FINAIS	38
6.1	Trabalhos Futuros	38
	REFERÊNCIAS	39

1 Introdução

O estilo de vida moderno vem cada vez mais influenciando o comportamento humano e isto vem afetando diretamente a qualidade do sono. Distúrbios relacionados ao sono são um problema comum entre os brasileiros. Segundo o IBGE (Instituto Brasileiro de Geografia e Estatística) (2013), 7,6% da população se utiliza de medicamentos para dormir e esse problema vem afetando até os indivíduos mais jovens (NUNES, 2002).

O sono não pode ser ignorado como fator importante da saúde das pessoas. Segundo Müller e Guimarães (2007), “(...) as perturbações do sono podem acarretar alterações significativas no funcionamento físico, ocupacional, cognitivo e social do indivíduo, além de comprometer substancialmente a qualidade de vida”.

Weber e Montovani (2002) expõem diversos estudos que apontam fortes relações dos hábitos e qualidade de sono com a ocorrência de acidentes no trânsito, o que torna a discussão um assunto de segurança também. Com todo esse contexto, percebe-se que o sono é um grande problema tanto para o indivíduo quanto para a sociedade.

Recentemente, com o advento da telefonia celular e da criação dos *smartphones*, a criação de soluções baseadas em aplicativo cresceram de forma vertiginosa. Diversas áreas se beneficiam destas tecnologias, que vão desde as redes de transporte (Uber/Moovit) até aplicações médicas (Medscape/Google Fit). A área do sono não ficou de fora dessa tendência. Diversas aplicações foram criadas neste sentido abrangendo desde despertadores inteligentes até auxílios sonoros, como músicas ou sons relaxantes, o que vem auxiliando diversas pessoas, tendo como pré-requisito apenas a posse de um *smartphone*.

No Brasil, de acordo com o IBGE, 92,6% dos domicílios possuem algum celular, o que indica oportunidades de criação de soluções com alta abrangência e capacidade de absorção por parte da população. No caso dos aplicativos voltados ao auxílio do sono, o uso por parte das pessoas vem crescendo recentemente e alguns aplicativos já possuem mais de um milhão de *downloads* na loja virtual Google Play. Esses aplicativos tem foco principalmente em disponibilizar ferramentas para auxílio do sono disponibilizando também ao usuário estatísticas sobre o mesmo.

1.1 Aplicativos

A Figura 1 apresenta uma interface exemplo desses aplicativos, um padrão que se repete na maioria dos aplicativos semelhantes, onde a movimentação da pessoa é traduzida em um gráfico utilizando o acelerômetro do celular, que deve ser deixado na cama próximo ao usuário. Através desse gráfico é possível deduzir os estágios de sono ao decorrer do repouso, já

que os mesmos têm fortes relações com o nível de atividade muscular ([FERNANDES, 2006](#)).

Figura 1 – Sleep as Android.



Fonte: *print screen* do aplicativo Sleep as Android.

Estas estatísticas são muito importantes, tanto para a percepção de anomalias no ciclo do sono, quanto para o despertar mais suave, em momentos em que a pessoa apresenta um sono mais leve.

1.2 Problema

Porém, tudo isso pode ser pouco efetivo ao se tratar de distúrbios do sono. Como muitos deles precisam ser tratados por médicos especialistas, não cabe aos aplicativos esta função. Uma possibilidade a ser explorada por este trabalho é a exibição de conteúdo de conscientização sobre estes distúrbios ao notar anomalias no comportamento dos gráficos, trazendo informações que possam ser relevantes, como as patologias mais comuns, as recomendações de higiene do sono e informações gerais sobre o sono em si por exemplo.

Os problemas relacionados ao sono são um grande empecilho, tanto para a sociedade quanto para os indivíduos e é um mal se apresenta em boa parte das pessoas, como mostram estudos como [Zanuto et al. \(2015\)](#). Devido aos diversos malefícios já expostos que os distúrbios do sono podem trazer, acredita-se que se os mesmos fossem rapidamente detectados de forma ampla, haveriam grandes benefícios.

Os aplicativos móveis são meios de amplo acesso e baixo custo que podem auxiliar nesse processo, através principalmente da detecção de anomalias e de conscientização sobre as mesmas e sobre recomendações para um sono mais efetivo.

Portanto espera-se que a criação de um aplicativo que reúna tanto as funcionalidades de auxílio para pessoas saudáveis quanto as funcionalidades de detecção de anomalias para pessoas com distúrbios seja uma forma viável de detectar mais rapidamente na população

indivíduos que apresentem sintomas que devessem ser melhor investigados e assim aumentar a qualidade do sono da população em geral.

1.3 Justificativa

O bem-estar do indivíduo é de fato uma das grandes preocupações da nossa sociedade. Indivíduos saudáveis têm maior grau de satisfação com suas vidas e apresentam melhores resultados em suas áreas de atuação.

Portanto, é de grande valia que qualquer esforço no sentido de amenizar ou remover os efeitos negativos que alguma patologia ou comportamento possa trazer às pessoas seja avaliado e colocado em prática.

Os problemas com o sono se encaixam nesse contexto. Um número muito grande de pessoas apresenta algum tipo de efeito negativo relacionado ao sono, o que só reforça a necessidade de medidas para tratá-los. Eles colaboram com prejuízos econômicos e sociais e devem ser amplamente combatidos.

Um fato que colabora com a solução apresentada nessa perspectiva é que aplicativos podem ser amplamente utilizados e são economicamente viáveis no contexto desta aplicação, tendo assim um grande potencial de melhorar o bem-estar geral dos humanos.

1.4 Objetivos

Este trabalho teve como objetivo a aplicação de conceitos de visualização de dados, aprendizagem de máquina, aliado a tecnologias de desenvolvimento *mobile* e *web*, buscando construir uma ferramenta para levar conhecimento às pessoas de forma inteligente, intuitiva e personalizada, ao mesmo tempo em que coleta dados para diversos usos, dos quais alguns serão explorados durante o trabalho.

Como é um problema muito amplo, este trabalho se focará principalmente no distúrbio da insônia, procurando detectá-la e levar informações sobre a mesma, ao mesmo tempo que cria um conjunto de dados que poderá ser utilizado para aplicações futuras.

2 Fundamentação

Nesta seção serão apresentados alguns conceitos relacionados à natureza do sono humano e também às tecnologias adotadas, visando embasar as escolhas de projeto.

2.1 Sono

O sono é uma parte fundamental no bom funcionamento do corpo. Associados a ele estão diversos distúrbios que podem ser muito prejudiciais à saúde humana, tanto física quanto psicológica, e também à sociedade (MÜLLER; GUIMARÃES, 2007).

Muitos destes distúrbios apresentam mudanças no comportamento do ciclo do sono, que possui um certo padrão em pessoas saudáveis(FERNANDES, 2006). Este ciclo é composto de alguns estágios que estão diretamente associados ao nível de agitação e movimentação do corpo durante o sono, como descrito em Jansen et al. (2007) e Fernandes (2006).

Levando isso em consideração, é possível detectar aproximadamente o estágio de sono que um indivíduo se encontra ao se monitorar o nível de movimentação. Este trabalho busca explorar essa relação, registrando movimentos durante o sono para tentar relacionar anomalias do ciclo do sono ao comportamento dos dados captados, visando detectar distúrbios do sono.

Alguns distúrbios possuem maior correlação com anomalias no ciclo do sono, como é o caso da insônia, no qual se apresenta um atraso para a saída do estado de vigília, e também em alguns casos um encurtamento do período em que o corpo permanece nesses estágios (NUNES, 2002), estando portanto em um estado mais agitado por mais tempo em comparação com um indivíduo saudável.

2.1.1 Regularidade

Para este trabalho, foi utilizado o SRI (*Sleep Regularity Index*) como índice de regularidade do sono. "O SRI foi originalmente descrito como a probabilidade que dois pontos no tempo, separados por 24 horas, estejam no mesmo estado de sono-vigília, durante todos os dias"(LUNSFORD-AVERY et al., 2018, tradução nossa). Este índice possui correlação negativa com a sonolência diurna, e com doenças cardiovasculares, metabólicas e psiquiátricas (LUNSFORD-AVERY et al., 2018), sendo um bom indicador para melhorar a saúde do usuário. Este índice é calculado como mostrado na Equação 2.1.

$$SRI = -100 + \frac{200}{M(N-1)} \sum_{j=1}^M \sum_{i=1}^{N-1} \delta(S_{i,j}, S_{i+1,j}) \quad (2.1)$$

onde M é o número intervalos em que é dividido o dia, N é o número de dias, $\delta(S_{i,j}, S_{i+1,j}) = 1$ se $S_{i,j} = S_{i+1,j}$ e $\delta(S_{i,j}, S_{i+1,j}) = 0$ se $S_{i,j} \neq S_{i+1,j}$.

Em termos de categorização, índices SRI abaixo de 60,8 pode, ser considerados irregulares, enquanto índices maiores que 84,0 podem ser considerados regulares, levando em consideração o estudo de [Lunsford-Avery et al. \(2018\)](#). Esse trabalho mostrou também um valor mediano aferido numa amostra de 6814 adultos de aproximadamente 75,0 e média de aproximadamente 72,0.

2.2 Smartphones

Smartphones são telefones móveis com características avançadas, como alta conectividade, suporte multimídia, alto poder computacional, alta capacidade de armazenamento e uma variedade de sensores, como acelerômetros, giroscópio e telas sensíveis ao toque ([TECHOPEDIA, 2019](#)).

A telefonia móvel já está muito disseminada no país ([IBGE, 2017](#)), o que se torna um meio interessante para a criação de soluções que busquem atingir uma grande parcela das pessoas. Os *smartphones*, por disponibilizarem uma grande quantidade de sensores, criaram uma grande oportunidade para soluções que utilizem dados desses sensores, possibilitando por exemplo a criação dos aplicativos de monitoramento do sono.

Exemplificando uma configuração considerada normal atualmente, o Samsung Galaxy M10 possui 3 gigabytes de memória RAM, câmeras de 13 e 5 megapixels, sistema operacional Android, 32 gigabytes de memória interna, processador com 8 núcleos de 1.6 gigahertz e conta com acelerômetro, sensor de luminosidade e proximidade ([SAMMOBILE, 2019](#)). Apesar de ser um celular recente, ele não possui giroscópio, o que ainda é muito comum em celulares mais acessíveis.

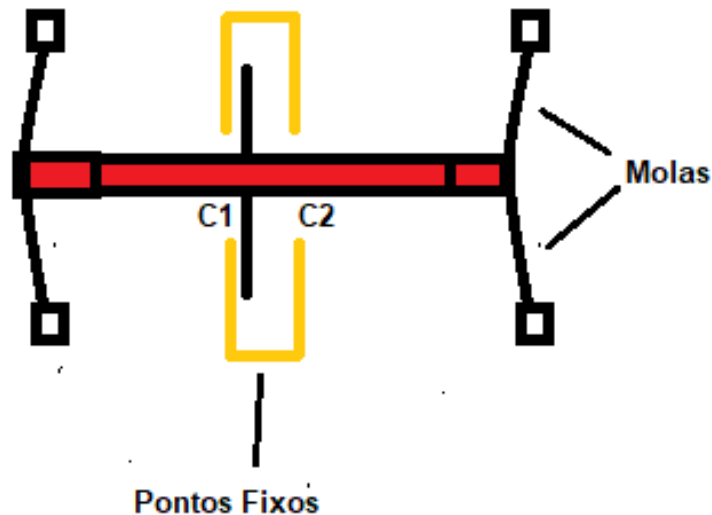
2.2.1 Sensor

De acordo com [Collins \(2019, tradução nossa\)](#), "Um sensor é um instrumento que reage a certas condições físicas ou impressões como calor e luz, e que é usado para dar informação". Para a captação da movimentação do usuário neste trabalho, foi necessário escolher um sensor que fosse capaz de captar essa movimentação, de forma simples e que essa solução pudesse ser disponibilizada. Dois sensores ficaram em evidência ao levar em conta esses pontos, sendo eles o acelerômetro e o giroscópio.

O acelerômetro mede a aceleração de um objeto, no caso o *smartphone*, nos três eixos de rotação, ou seja, independente da direção do movimento, o sensor é capaz de fazer a detecção. Existem diversos modelos de acelerômetros, tendo como exemplo os Capacitivos, os Piezoelétricos, os Piezoresistivos e os Eletromecânicos ([FIGUEIREDO et al., 2007](#)).

A Figura 2 apresenta o esquema de um acelerômetro MEMS (*Micro Electrical Mechanical Systems*), no qual a aceleração é medida utilizando as capacitâncias medidas nos pontos C1 e C2, que variam com a distância entre a massa e o ponto fixo.

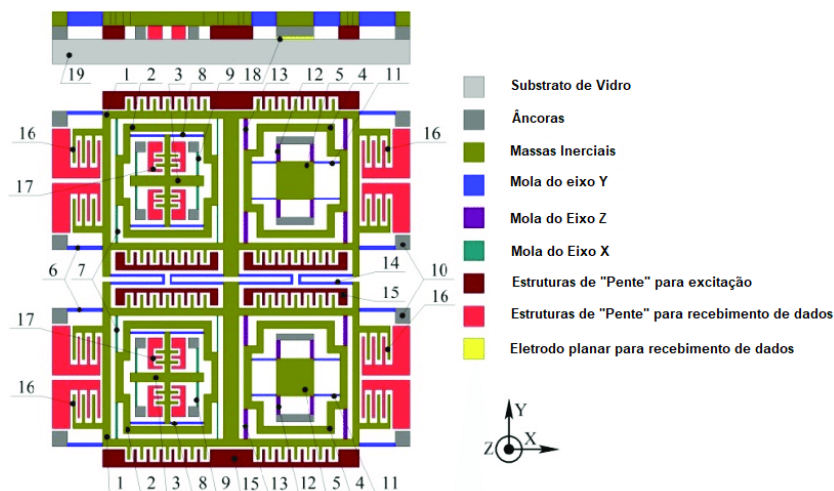
Figura 2 – Acelerômetro MEMS.



Fonte: Elaborada pelo autor.

Já o giroscópio é um sensor "capaz de detectar velocidade angular" (PASSARO et al., 2017), podendo ser usado em aplicações que necessitem de uma precisão maior da orientação do smartphone. A Figura 3 mostra o esquema de um giroscópio MEMS, no qual o silicone converte o sinal de entrada (velocidade angular) num sinal elétrico que é processado e retorna as informações de orientação do objeto. (NESTERENKO; KOLEDA; BARBIN, 2018)

Figura 3 – Giroscópio MEMS.



Fonte: Adaptado de Nesterenko, Koleda e Barbin (2018).

Porém, essa precisão vem com um custo. Nem todo aparelho possui este sensor ([MEDIUM, 2017](#)), ficando muitas vezes restrito aos smartphones de topo de linha, e por isso optou-se pelo acelerômetro, que além de atender os outros requisitos está disponível na maioria dos *smartphones*.

O parâmetro de desempenho de um acelerômetro varia conforme a utilização. Pode-se medir amplitude máxima, sensibilidade, frequência das medições, número de eixos, construção e massa. Porém este trabalho não leva em conta o desempenho do sensor, supondo assim que haja um sensor com desempenho aceitável no aparelho utilizado.

2.3 Aprendizado de Máquina

Segundo [Bishop \(2006, tradução nossa\)](#)

O problema de procurar padrões nos dados é fundamental e tem uma longa e bem-sucedida história. Por exemplo, as extensas observações astronômicas de Tycho Brahe no século XVI permitiu a Johannes Kepler descobrir as leis empíricas da movimento planetário, que por sua vez forneceu uma base para o desenvolvimento da mecânica clássica.

Diversos problemas atuais envolvem a procura por padrões, e os problemas médicos relacionados ao sono não fogem dessa realidade.

"O aprendizado de máquina é um ramo da inteligência artificial que aplica algoritmos sistematicamente para sintetizar as relações subjacentes entre dados e informação."([AWAD; KHANNA, 2015, tradução nossa](#)). Com isso, por exemplo, é possível criar sistemas computacionais que busquem relações entre dados de um paciente e sua condição médica, possibilitando ao sistema aprender com essas correlações e fazer inferências com outros dados de entrada.

Esses algoritmos podem ser divididos em três categorias principais: supervisionado, não-supervisionado e por reforço.

Os algoritmos de aprendizado supervisionada são conduzidos de forma que o programa já tenha as respostas corretas para os dados de treinamento, sendo trabalho do sistema replicar essas respostas corretamente ([DANGETI, 2017](#)).

Nos algoritmos de aprendizado não-supervisionado, o programa não tem as respostas corretas, sendo papel do sistema agrupar os dados conforme características do próprio conjunto de dados de treino, utilizando técnicas como k-means ou PCA (*Principal Component Analysis*) por exemplo ([DANGETI, 2017](#)).

Ja nos algoritmos de aprendizado por reforço, não existe uma variável alvo de análise, mas sim um sistema de recompensa. Neles, um agente é responsável por planejar seu próprio modo de resolver o problema, de forma que atinja os objetivos e receba as recompensas. Assim,

quanto mais o agente atingir os objetivos explicitados pelo sistema, melhor a sua solução do problema ([DANGETI, 2017](#)).

Alguns exemplos de algoritmos de cada categoria podem ser visualizados no Quadro 1.

Quadro 1 – Categorias de algoritmos de *Machine Learning*.

Categoria	Exemplos de Algoritmos
Supervisionado	SVM; N��ive Bayes
N��o-Supervisionado	k-means; PCA
Por Refor��o	Q-Learning

Fonte: Elaborado pelo autor.

3 Ferramentas e Tecnologias

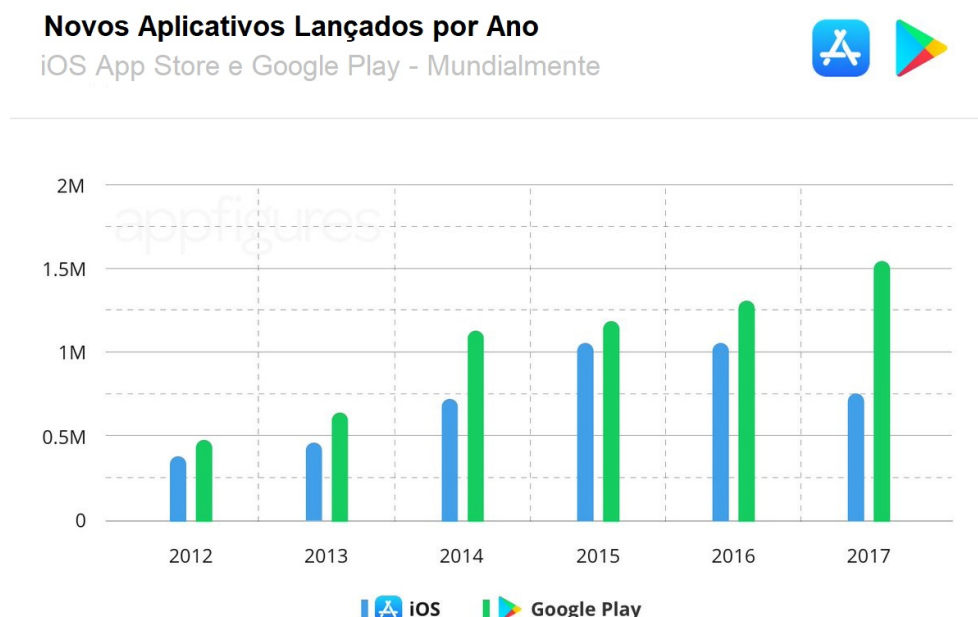
Nesta seção serão tratadas algumas ferramentas e tecnologias utilizadas no desenvolvimento deste projeto, visando mostrar vantagens de se utilizá-las.

3.1 Flutter e Desenvolvimento *Mobile*

Com a consolidação da utilização de *smartphones* pela população, abriu-se um grande leque de oportunidades para o desenvolvimento de aplicações. O poder de processamento, geração de dados e comunicação que está distribuída pelos diversos dispositivos pode ser aproveitado e direcionado para as mais diversas aplicações, e é o que vem ocorrendo.

Como mostrado na Figura 4, o número de aplicativos lançados ano a ano com a popularização dos *smartphones* vem crescendo e isso torna o próprio desenvolvimento de apps algo a ser melhorado.

Figura 4 – Crescimento das lojas de aplicativo.



Fonte: Adaptado de Página da Appfigure no Twitter¹.

Com esse crescimento surgiram também diversas ferramentas para auxiliar o desenvolvimento de aplicativos, principalmente os *frameworks* híbridos. Estes *frameworks* possibilitam, através de um código único e de um fluxo de desenvolvimento melhorado, a criação de aplicativos para os principais sistemas operacionais para *smartphones*, o Android e o iOS.

¹ Disponível em <<https://twitter.com/appfigures/status/983440681334116352>>. Acesso em: 24 out. 2019.

Exemplos destes *frameworks* são o Ionic, o React Native e o Flutter. Para o projeto, buscou-se um *framework* que ao mesmo tempo que entregasse desempenho, proporcionasse um fluxo de desenvolvimento mais ágil e simples.

O Flutter é o framework de código aberto da Google para desenvolvimento híbrido, que constrói aplicativos para *smartphones*, *desktop*, *web*, embarcados e para o Fuchsia, que é o novo Sistema Operacional que a Google vem desenvolvendo (GOOGLE, 2019a). Ele apresenta um fluxo de desenvolvimento muito ágil, baseado em *widgets* e de fácil depuração, na medida que é possível utilizar o *Hot Reload*, uma ferramenta que permite a apresentação das mudanças do código em tempo real no aplicativo em si, através de uma máquina virtual Dart, que é a linguagem de programação do *framework* (GOOGLE, 2019b).

No quesito desempenho ele também atinge os requisitos, já que constrói código nativo para ARM, não necessitando de *WebViews* ou Máquinas Virtuais em suas versões de *release*. Assim, quando o programador termina de desenvolver a aplicação ou quer ter uma versão otimizada, o *framework* disponibiliza um modo de *build* em que ele transforma o fonte em código nativo, perdendo a função de *Hot Reload*, mas ganhando desempenho próximo ao nativo.

Apesar disso, o Flutter possui motor gráfico próprio, a Skia. Assim o sistema nativo é responsável apenas por proporcionar uma superfície para que a Skia desenhe os componentes gráficos. Ganha-se com isso padronização de desenvolvimento, já que os componentes gráficos nativos variam e sistema para sistema e isso impacta diretamente o desenvolvimento utilizando os mesmos (GOOGLE, 2019a).

Com o *framework* vem também diversas bibliotecas, que permitem a interação com diversos serviços nativos. Um deles é o Sensors, *plugin* oficial do *framework* para lidar com as saídas do sensor dos dispositivos. Ele disponibiliza uma interface simples de código no qual se pode obter os valores do sensor através de uma *Stream* (PUB.DEV, 2019).

3.2 REST

REST (*Representational State Transfer*) é um modelo para criação de arquiteturas de *software* distribuído, baseado na comunicação via rede (FERREIRA, 2017).

Este modelo é muito utilizado para a criação de *Web Services*, no qual os recursos que são utilizados numa aplicação são administrados. Exemplo disso é a gestão do banco de dados, que pode ser disposto numa camada que só interaja com a aplicação por meio de uma camada que faça o controle de requisições, uma API (*Application Programming Interface*).

Cada item gerenciado num modelo REST é denominado "recurso", e deve ser identificado por uma URI (*Uniform Resource Identifier*).

Quando uma requisição é disparada pelo cliente, junto com o recurso é enviado o

método de manipulação do recurso, dentre eles ([FERREIRA, 2017](#)):

- *GET*: Obter os dados de um recurso
- *POST*: Criar um novo recurso.
- *PUT*: Substituir os dados de um determinado recurso.
- *PATCH*: Atualizar parcialmente um determinado recurso.
- *DELETE*: Excluir um determinado recurso.
- *HEAD*: Similar ao GET, mas utilizado apenas para se obter os cabeçalhos de resposta, sem os dados em si.
- *OPTIONS*: Obter quais manipulações podem ser realizadas em um determinado recurso.

Dessa forma a aplicação cliente pode enviar requisições criando as mais diversas funções, adaptando os métodos e os recursos para a sua solução.

3.3 Python e Flask

Flask é um *microframework* python voltado para o desenvolvimento *web*. Foi desenvolvido com base no motor de *templates* Jinja e no Werkzeug WSGI toolkit. Busca a simplicidade e velocidade de desenvolvimento, não força dependências nem estruturas de projeto, ao mesmo tempo que é escalável e possui diversos adicionais caso sejam necessários ([THE PALLETS PROJECTS, 2019a](#)).

O objetivo com a utilização do Flask é criar um servidor que responda a requisições HTTP (*Hypertext Transfer Protocol*) e faça a interação tanto com os algoritmos de recomendação de conteúdo, quanto com o banco de dados, mapeando as requisições para os recursos disponibilizados pela API, como guiado por [The Pallets Projects \(2019b\)](#). Uma facilidade a mais é a linguagem, que é a mesma utilizada nos algoritmos de *machine learning*, facilitando a comunicação entre essas duas partes do projeto.

3.4 Firebase

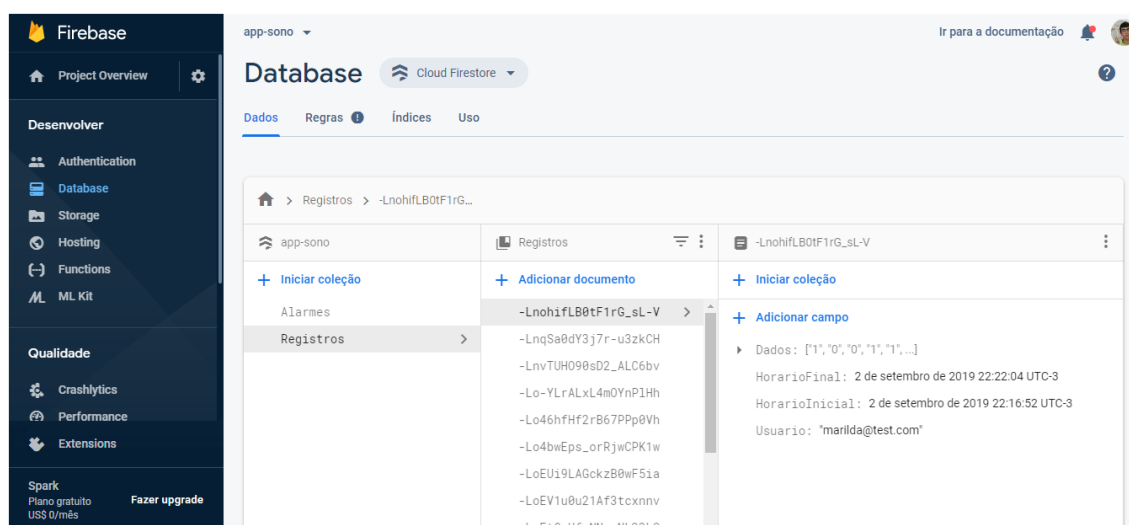
A persistência dos dados é fundamental no desenvolvimento de diversos aplicativos. Essa camada da aplicação pode consumir em muitos casos uma grande parcela do tempo de desenvolvimento, o que acaba prejudicando a agilidade do desenvolvimento de protótipos.

Buscou-se nesse projeto uma solução de persistência que permitisse a prototipação de forma mais maleável, o que foi proporcionado pelo Firebase. É uma solução da Google para persistência "No SQL" que é integrada a diversas ambientes de desenvolvimento, incluindo o

Flutter. A Figura 5 mostra o *dashboard* da ferramenta, através da qual é possível visualizar os dados, fazer operações de inserção, remoção e modificação e também interagir com outras ferramentas que o Firebase disponibiliza, como armazenamento, hospedagem e autenticação.

Além disso, o Firebase possui diversas funcionalidades voltadas para *Apps* e *WebApps*, como as Análises de Uso, a Análise de erros e o Banco de Dados na nuvem em tempo real (FIREBASE, 2019).

Figura 5 – *Dashboard* Firebase.



Fonte: *print screen* da página do Firebase.

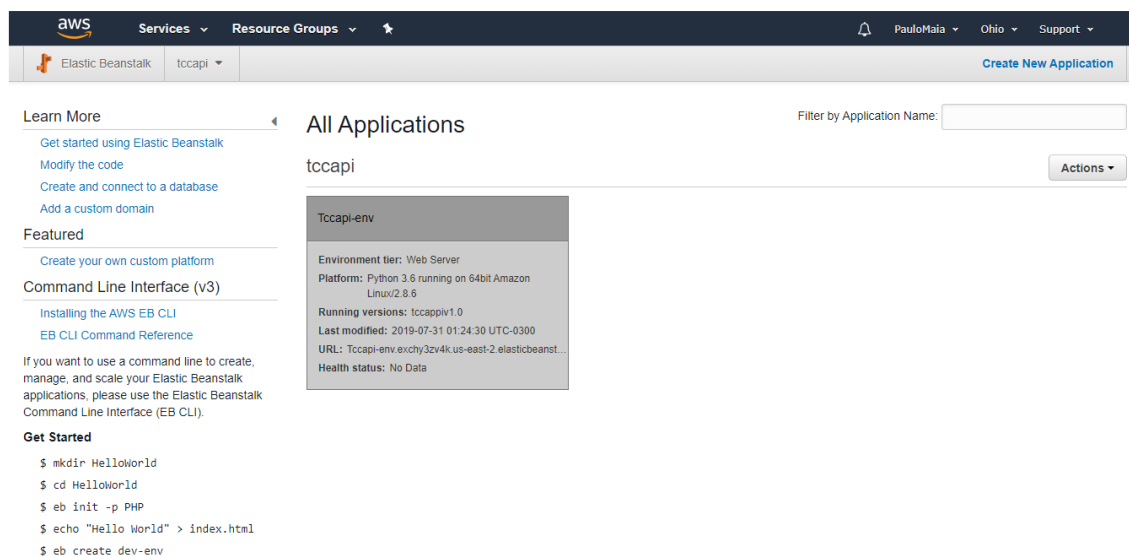
3.5 AWS

O desenvolvimento de aplicações é muito incerto no que diz respeito à capacidade de infraestrutura dos servidores. As aplicações podem ter grandes picos de uso, ou podem acabar nem tendo utilização. Portanto, serviços de computação em nuvem sob demanda podem ser uma solução muito benéfica, já que além de agilizar o processo de disponibilização dessa estrutura, tem a capacidade de adequar dinamicamente à demanda da aplicação (BREBNER, 2012).

Como a aplicação dispõe de um modelo REST de arquitetura distribuída, ou seja, possui uma API via protocolo HTTP, é necessário que um dispositivo faça o papel de servidor e possua disponibilidade e desempenho adaptativo. Por isso decidiu-se que a API seria hospedada numa plataforma de *Web Services*. A AWS (*Amazon Web Services*) é uma provedora de plataformas de computação sob demanda, que proporciona mais infraestrutura na medida que a demanda pelos serviços sobe, através do EC2 (*Elastic Compute Cloud*). Além disso já possui diversos outros serviços que dão suporte a outras funções que podem ser necessárias a este tipo de aplicação, como por exemplo armazenamento em nuvem, que é disponibilizado pelo S3 (*Simple Storage Service*) na AWS (AMAZON, 2019).

A Figura 6 mostra o *dashboard* do ElasticBeanstalk, através da qual pode-se configurar o funcionamento das instâncias de computação em nuvem, configurando suas capacidade, ambiente, o código que vai ser executado, entre outras configurações.

Figura 6 – *Dashboard* ElasticBeanstalk.



The screenshot displays the AWS Elastic Beanstalk console. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (PauloMaia, Ohio, Support). Below the navigation bar, the 'Elastic Beanstalk' section is active, showing 'tccapi' as the selected application. The main content area is titled 'All Applications' and includes a filter by application name. The application 'tccapi' is listed with an 'Actions' button. A detailed view of the 'Tccapi-env' environment is shown, including its tier (Web Server), platform (Python 3.6), running versions (tccappiv1.0), last modified date (2019-07-31), URL, and health status (No Data). On the left sidebar, there are links for 'Learn More', 'Featured', and 'Command Line Interface (v3)', along with a 'Get Started' section containing terminal commands for creating and running an application.

```
$ mkdir HelloWorld
$ cd HelloWorld
$ eb init -p PHP
$ echo "Hello World" > index.html
$ eb create dev-env
```

Fonte: *print screen* da página do ElasticBeanstalk.

4 Métodos

Nesta seção serão apresentadas as etapas do projeto e como foram desenvolvidas.

4.1 Análise de Distúrbios

Primeiramente buscou-se entender melhor sobre os distúrbios e seus sintomas. Nesta fase buscou-se selecionar qual deles apresentaria melhor relação com os dados obtidos pelo sensor. Para isso, procurou-se um distúrbio facilmente perceptível através do gráfico de sono gerado. Tomou-se como base o documento ICSD (*International Classification of Sleep Disorders*), que categoriza e descreve os distúrbios do sono, facilitando a tarefa de se optar por um dos distúrbios para a aplicação.

O ICSD aponta algumas categorias, dentre elas a Insônia, Distúrbios de Ciclo Circadiano do Sono-Vigília e Distúrbios de Movimento Relacionados ao Sono. Estas categorias agrupam distúrbios que têm influência na movimentação durante o sono, seja pela movimentação anormal, ou pela distorção do ciclo circadiano, com o encurtamento, alongamento ou atraso de fases do mesmo.

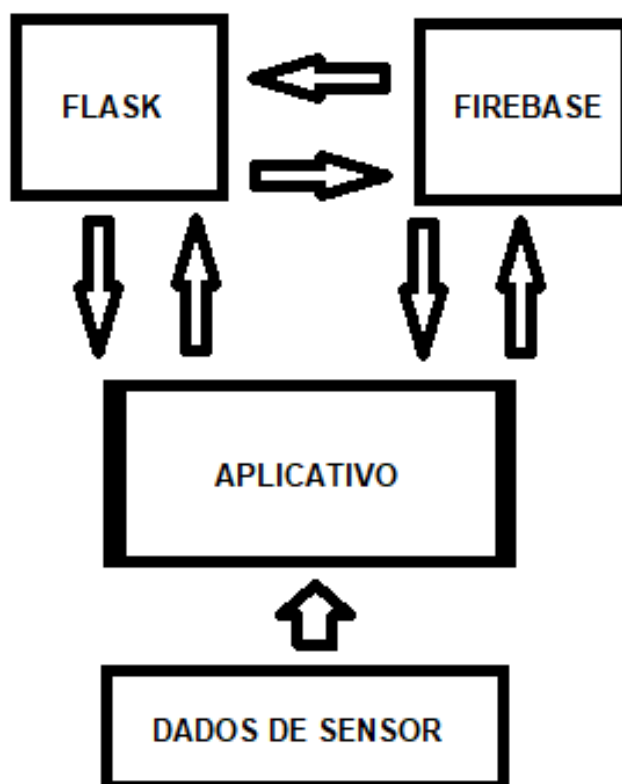
Pelo fato das características das Insônias serem mais relacionadas à duração e posicionamento temporal das fases do ciclo, ela é mais perceptível através de análise gráfica e espera-se portanto que o treinamento de um classificador para a mesma seja mais fácil, sendo portanto a categoria de foco deste trabalho.

Foi realizado também um levantamento das aplicações que lidam com esse distúrbio, o que mostrou que grande parte das aplicações estão focadas em ajudar o usuário a dormir, ou apenas monitorar seu sono, indicando uma falta de aplicações voltadas a levar informação às pessoas. Exemplos de aplicações são o Pzizz, que possui um sistema patenteado de emissão de sons para induzir o sono, utilizando princípios psicoacusticos, e o SleepCycle, que monitora o seu sono e te ajuda a acordar melhor, lendo os dados dos sensores do celular.

4.2 Design da Aplicação

As atividades chave no desenvolvimento do aplicativo foram a sua integração com o Firebase, com a API, e com as saídas dos sensores. A Figura 7 mostra o esquema geral da aplicação.

Figura 7 – Arquitetura da aplicação.



Fonte: Elaborada pelo autor.

Para a integração com o firebase foi utilizada a biblioteca oficial para o Cloud Firestore. Utilizou-se também a biblioteca de integração com a autenticação de usuário, que permitiu a individualização do uso do aplicativo e tratamento de permissões de uso do banco de dados.

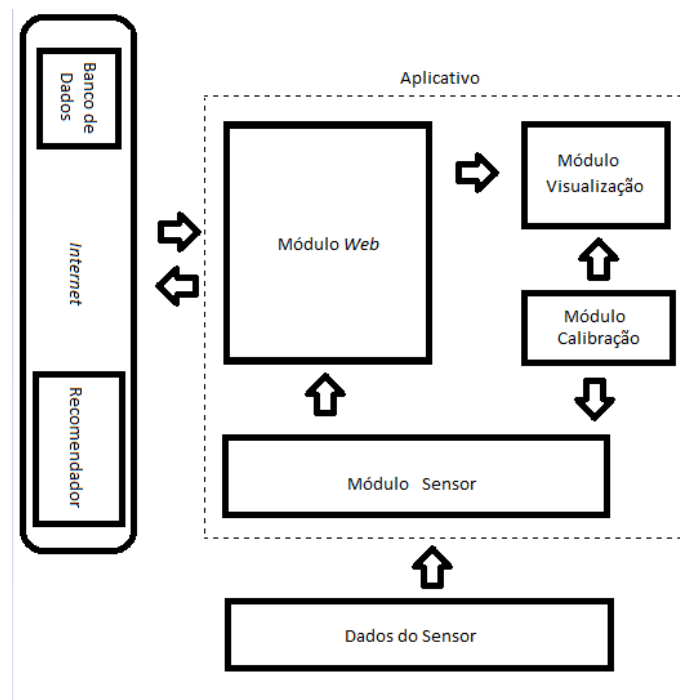
Na parte de integração com a API, como foi desenhada com uma arquitetura REST, a única biblioteca necessária foi a do protocolo HTTP, que permitiu comunicação via *internet* com o API, que ficou hospedada numa instância EC2 da AWS. Isso permitiu uma fácil integração entre um app desenvolvido em Dart, com uma API desenvolvida em Python.

Desta forma, o aplicativo envia solicitações parametrizadas à API, que as recebe, processa e retorna apenas os dados necessários ao aplicativo, exigindo menos esforço computacional da aplicação cliente.

Já para o uso dos dados do sensor, foi utilizada a biblioteca Sensors, que disponibiliza uma *Stream* de dados vindos dos sensores.

A divisão em módulos do aplicativo e como elas interagem está ilustrado na Figura 8.

Figura 8 – Módulos do aplicativo.



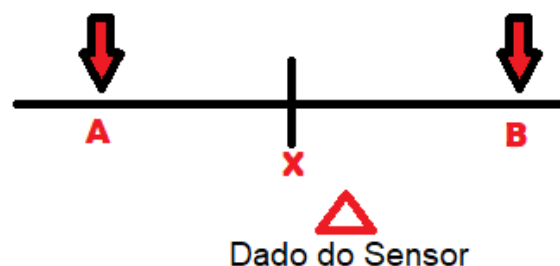
Fonte: Elaborada pelo autor.

Tendo os dados, buscou-se uma forma de criar um vetor de dados que representasse a movimentação do usuário durante o sono. Por padronização, foi retirado a dimensão da intensidade da movimentação, pois além dessa variável ser muito instável por causa da variação da rigidez dos colchões e posicionamento do celular, ela não interferiria na aplicação.

Os dados recebidos de sensores de *smartphones* podem ser um tanto instáveis, portanto foi criado um algoritmo que mede se a intensidade da variação do dado do sensor ultrapassa um limite.

Ilustrando esse mecanismo, a Figura 9 indica os limites calculados A e B, um valor X que é gravado como sendo a posição em repouso do aparelho (que muda dinamicamente com o tempo) e a saída do sensor. Lembrando que este tratamento é feito nos três eixos de rotação

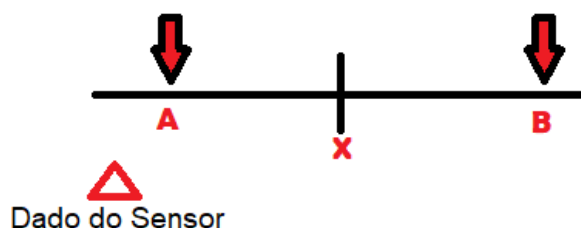
Figura 9 – Mecanismo de tratamento do sensor.



Fonte: Elaborada pelo autor.

Para ilustrar uma situação em que o algoritmo consideraria que houve movimentação do aparelho, a Figura 10 mostra uma saída do sensor que ultrapassa o limite inferior de valor. O valor aferido deixa o intervalo entre A e B e indica assim que ouve uma variação da saída do sensor em um eixo de rotação que não é considerado uma variação de imprecisão, mas sim de uma real movimentação do aparelho.

Figura 10 – Mecanismo de tratamento do sensor ao ser ativado.

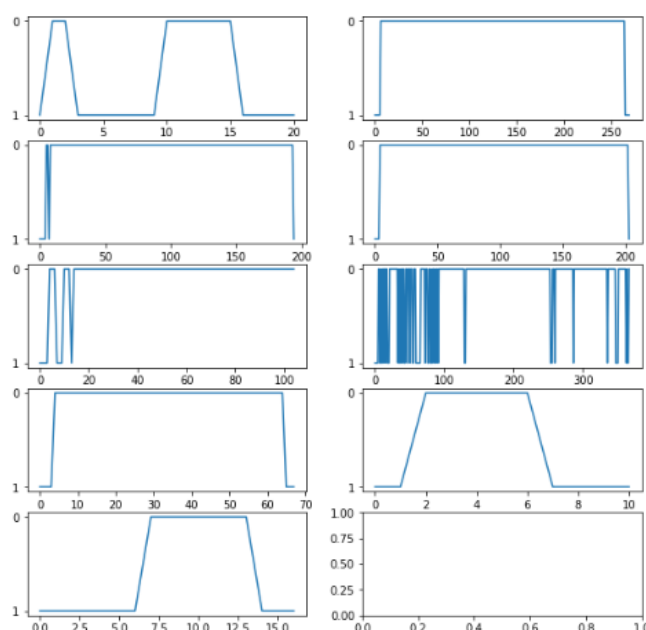


Fonte: Elaborada pelo autor.

A princípio, utilizou-se uma distância predefinida entre A e X, e entre X e B, obtida através de utilização em aparelho próprio. Porém, após a disponibilização do aplicativo, percebeu-se a necessidade da adequação desse intervalo para cada aparelho.

Esta seção dos dados, apresentado nos gráficos da Figura 11 indicou a discrepância entre as saídas obtidas nos aparelhos. Enquanto alguns obtiveram a saída desejada, alguns deixaram de detectar a movimentação noturna, por estarem pouco sensíveis, e outros detectavam movimentações que deveriam ser consideradas erros de precisão do sensor.

Figura 11 – Visualização de dados após disponibilização.



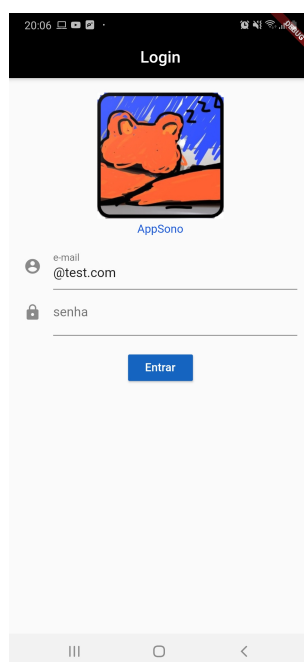
Fonte: Elaborada pelo autor.

Para resolução deste problema, criou-se um mecanismo de calibração. Ele funciona lendo os dados do sensor com o aparelho em repouso, assim ele consegue determinar qual a variação normal de erro de precisão do aparelho e assim determinar um intervalo que aumente a probabilidade de se detectar uma movimentação mas que se tenha uma probabilidade ainda menor de se detectar um "falso-positivo".

Para a persistência dos dados de calibração no aparelho dos usuários, foi utilizada a biblioteca SharedPreferences, que dá acesso a uma interface mais direta de interação com o armazenamento interno, o que facilitou o desenvolvimento dos mecanismos de gravação e recuperação destes dados.

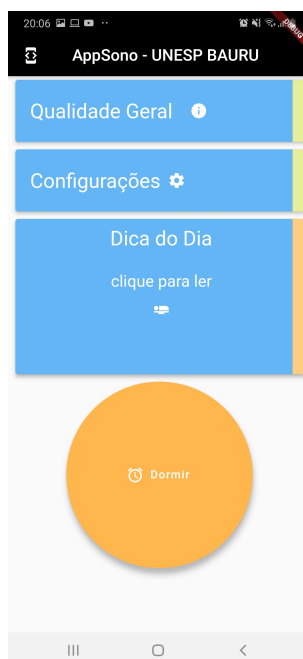
As telas foram divididas em 3 seções: Qualidade do sono, Configurações e dicas, que aparecerem na tela principal (Figura 13), que vêm após o usuário logar no sistema (Figura 12).

Figura 12 – Tela de *login*.



Fonte: Elaborada pelo autor.

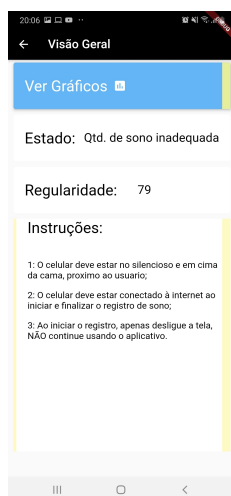
Figura 13 – Tela principal.



Fonte: Elaborada pelo autor.

Quando o usuário entrar na seção de qualidade do sono, encontrará informações como Estado Geral, que disponibiliza informações de fácil entendimento para as pessoas, e Regularidade, como mostrado na Figura 14, que são informações dadas com base nas entradas de sono do usuário. O texto exibido no Estado geral varia conforme os dados que o sistema possui sobre o usuário, podendo conter observações quanto ao horário de dormir, quantidade de horas dormidas e também relacionado à própria regularidade, facilitando o entendimento deste índice pelas pessoas.

Figura 14 – Tela qualidade geral.



Fonte: Elaborada pelo autor.

Além dessas informações, existe também a tela de visualização de gráfico, exibida na Figura 15. Esta tela disponibiliza para o usuário uma forma visual de analisar seu sono, apresentando um gráfico de linhas com os dados da última noite de sono.

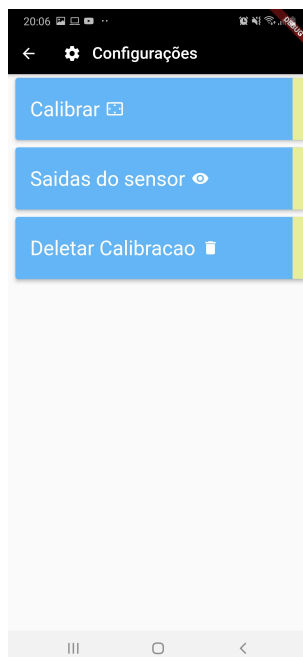
Figura 15 – Tela de gráfico.



Fonte: Elaborada pelo autor.

Na tela de configurações, mostrada na Figura 16, estão algumas opções relacionadas à calibração do celular, necessária para o funcionamento correto do aplicativo. Esta tela possui também uma opção de visualização dos dados do sensor, que disponibiliza ao usuário uma forma de ver as saídas que seriam gravadas nos registros de sono.

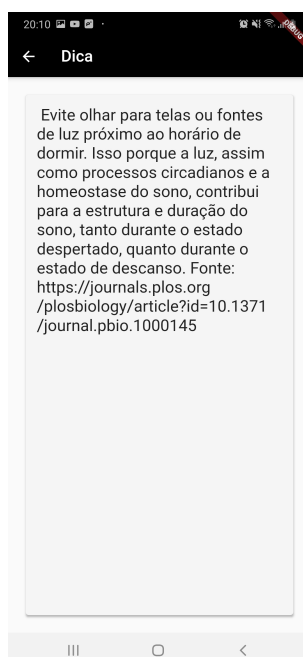
Figura 16 – Tela de configuração.



Fonte: Elaborada pelo autor.

Por último, a seção de Dica do Dia apresenta apenas um texto, que é escolhido pela API, e trazido do banco de dados, apresentando informações sobre o sono ao usuário, que pode ser visualizada na Figura 17.

Figura 17 – Tela de dica do dia.



Fonte: Elaborada pelo autor.

4.3 Transferência e Visualização de Dados

Para a obtenção e visualização dos dados foi utilizada a biblioteca Firebase Admin e também as bibliotecas Numpy e Matplotlib.

Foi feito a leitura dos documentos dos registros de sono e as entradas foram gravadas em um vetor, que continha vetores com os valores obtidos do banco.

Com esses valores, foi possível tanto a utilização destes dados para o treinamento das redes quanto para realizar a visualização dos dados. No caso da visualização de dados, como os vetores já possuíam o formato necessário para representação em duas dimensões, não foi necessário nenhum tratamento especial, tendo assim somente o problema de representar múltiplos gráficos, que foi resolvido utilizando as *subplots*, disponibilizadas na biblioteca Matplotlib.

4.4 Recomendador

A ideia inicial era a criação de recomendador baseado em uma rede neural, utilizando o Keras, para classificação dos dados de sono recebidos. Porém, como o número de dados obtidos foi bem reduzido, o treinamento do modelo ficou prejudicado, o que implicaria numa baixa taxa de acertos da aplicação ao usar as classificações da API, restando à aplicação, no tempo de desenvolvimento deste trabalho, estar apta a funcionar corretamente ao receber uma rede treinada, no que diz respeito a recomendações baseadas nos distúrbios.

Assim, o recomendador ficou restrito a analisar os dados de quantidade de horas dormidas, regularidade do sono e também os conteúdos já visualizados pelo usuário, retornando assim um texto da base de dados após requisição http.

5 Análises e Resultados

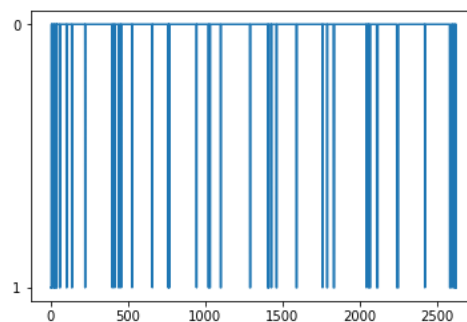
Nesta seção serão feitas as análises dos resultados obtidos neste trabalho

5.1 Captação

A captação da movimentação do usuário é parte essencial deste trabalho. Por meio da leitura do sensor do acelerômetro e realizando algumas calibrações e tratamento de ruídos, foi possível a criação de um vetor binário, no qual as posições com 0 indicam falta de movimento durante um intervalo e as posições com 1 indicam intervalos de tempo no qual o usuário se movimentou. Para a geração dos dados, buscou-se a variação nos dispositivos, afim de possibilitar a análise das diferenças de captação entre os *smartphones*, e também de ambientes, tentando analisar o funcionamento correto em diversas superfícies. Então, foi disponibilizada a aplicação para 5 dispositivos diferentes, variando marcas e modelos, sendo cada um empregado em ambiente diferente.

A Figura 18 ilustra uma saída obtida em um uso real da aplicação, sob a qual poderão ser feitas as análises. Primeiramente percebe-se um certo intervalo entre os períodos de maior movimentação, o que corrobora com a ideia de sono apresentar ciclos.

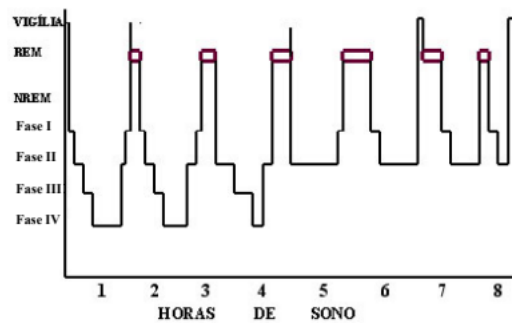
Figura 18 – Exemplo de saída.



Fonte: Elaborada pelo autor.

A Figura 19 exemplifica com um gráfico a estrutura de sono esperada teoricamente, com a qual podemos comparar a saída obtida na aplicação e perceber como ambas apresentam ciclicamente períodos de sono mais leve e períodos de sono mais profundo.

Figura 19 – Saída esperada.



Fonte: [Fernandes \(2006\)](#).

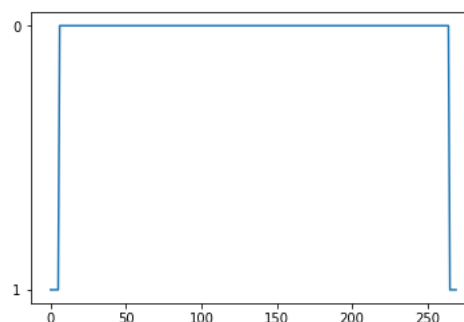
É possível também perceber um período de agitação maior no início do sono em ambos os gráficos, o que é normal devido à latência entre a tentativa de se iniciar um sono e o tempo real em que o usuário sai do estado de vigília e vai para um sono mais profundo. A análise desse trecho é essencial na tentativa de diagnosticar o distúrbio da insônia.

5.2 Visualização

A visualização dos dados é essencial neste trabalho, pela quantidade de dados gerados, que dificulta sua análise e por consequência o desenvolvimento do trabalho em si.

Utilizando as bibliotecas de visualização de dados do Python foi possível a criação de múltiplos gráficos que permitiram a leitura mais rápida das saídas da aplicação, permitindo verificar a compatibilidade entre dados e teoria e assim encontrar erros, como ilustrado na Figura 20.

Figura 20 – Saída com erros.



Fonte: Elaborada pelo autor.

Percebe-se que a quantidade de dados gerados não corresponde à quantidade esperada para uma noite de sono, e nem corresponde aproximadamente a um formato esperado, sugerindo assim que algo no algoritmo de captação não está funcionando corretamente.

Estes algoritmos podem também, de forma mais bruta, dar uma ideia do que está

acontecendo no sono em geral. Claro que como apenas uma variável está sendo monitorada, não se pode basear decisões nestes gráficos, porém eles podem auxiliar de alguma forma.

5.3 Recomendação

O treinamento de algoritmos de classificação possibilitaria ao sistema de recomendação de conteúdo escolher conteúdo mais pertinente ao usuário, porém, no tempo de desenvolvimento deste trabalho, os resultados não foram muito precisos.

O tamanho do conjunto de dados de treinamento é um fator que impacta diretamente na taxa de acerto dos algoritmos de classificação. Levando isso em conta, os resultados esperados com esses algoritmos neste trabalho não seriam muito bons.

Um fator decisivo para a quantidade de entradas ser baixa é que o conjunto de dados foi gerado pela própria aplicação em um estágio anterior de desenvolvimento, e também a natureza dos dados, que só podem ser captados uma vez por dia, impossibilitando uma expansão rápida dessa quantidade.

Testes realizados com a rede indicaram baixo número de acertos no conjunto de dados de teste, o que indicou que a rede não conseguiu generalizar o aprendizado, mas sim deu *overfit*, e em outros treinos não conseguiu chegar em uma acurácia aceitável.

Porém, mesmo com a parte de classificação desativada, o sistema de recomendação já é capaz de sugerir conteúdo relacionado às horas dormidas e a regularidade do sono, além de textos sobre o sono e sua higiene, adaptando qual conteúdo é sugerido ao usuário.

O SRI é parte fundamental no recomendador, pois faz com que o usuário receba conteúdo relacionado à regularidade do sono quando percebe que o usuário possui um índice baixo, ou seja, menor que 60,8.

6 Considerações Finais

Conforme descrito e seguindo os objetivos propostos, o aplicativo desenvolvido foi capaz de captar dados do sono do usuário, permitindo a análise desses dados pelo aplicativo e com isso fazer recomendação de textos informativos sobre o sono.

Para a prototipação da aplicação, as ferramentas utilizadas foram adequadas, sendo sua utilização viável em projetos de outras aplicações de leitura e análise de dados de sensores de *smartphone*.

A aplicação é de fácil uso e disponibiliza facilmente para o usuário informações úteis, dando ao usuário um panorama geral sobre o seu sono sem ter sido necessário aumentar a complexidade de uso da mesma.

O desenvolvimento do projeto proporcionou o aprendizado de diversas tecnologias novas não ensinadas durante o curso, como as tecnologias de desenvolvimento *mobile* híbrido, e isso permitiu a exploração de diversos conceitos que são exigidos no mercado de trabalho, agregando conhecimento ao processo de formação.

As dificuldades neste trabalho foram, de forma geral, mais relacionadas à programação assíncrona, que ocasionou diversos erros não perceptíveis ao compilador, criou a necessidade de estudo de conceitos e de APIs em todas as linguagens utilizadas no trabalho, até mesmo nas ferramentas no caso do Firebase.

6.1 Trabalhos Futuros

Para trabalhos futuros propõe-se a adição de novos sensores à aplicação.

A detecção dos distúrbios não ocorre utilizando exclusivamente dados de movimentação do paciente. Espera-se que com o tempo, dispositivos com mais sensores tornem-se mais populares e possibilitem o desenvolvimento de aplicações utilizando uma gama maior de dados.

Exemplificando, o trabalho poderia se expandir utilizando sensores de temperatura corporal, batimentos cardíacos, luminosidade de ambiente e também microfones, dando aos algoritmos de diagnóstico mais componentes a serem analisados, tornando a solução mais precisa.

Propõe-se também a utilização da aplicação em grupos mais selecionados de usuários, para que os resultados obtidos sejam mais relevantes, o que não foi possível durante esse trabalho, e também a criação de mais conteúdo educacional, agregando uma quantidade maior e mais atualizada de informações aos usuários da aplicação.

Referências

AMAZON. *AWS Documentation*. 2019. Disponível em: <https://docs.aws.amazon.com/index.html?nc2=h_ql_doc_do>. Acesso em: 12 out. 2019.

AWAD, M.; KHANNA, R. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. [S.l.]: Apress, 2015.

BISHOP, C. M. *Pattern recognition and machine learning*. New York, NY: Springer, 2006. (Information science and statistics). Softcover published in 2016. Disponível em: <<https://cds.cern.ch/record/998831>>.

BREBNER, P. C. Is your cloud elastic enough?: Performance modelling the elasticity of infrastructure as a service (iaas) cloud applications. In: *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*. New York, NY, USA: ACM, 2012. (ICPE '12), p. 263–266. ISBN 978-1-4503-1202-8. Disponível em: <<http://doi.acm.org/10.1145/2188286.2188334>>.

COLLINS. *Definition of Sensor*. 2019. Disponível em: <<https://www.collinsdictionary.com/dictionary/english/sensor>>. Acesso em: 30 out. 2019.

DANGETI, P. *Statistics for Machine Learning: Techniques for Exploring Supervised, Unsupervised, and Reinforcement Learning Models with Python and R*. [S.l.]: Packt Publishing, 2017. ISBN 1788295757, 9781788295758.

FERNANDES, R. M. F. O sono normal. *Medicina (Ribeirao Preto. Online)*, v. 39, n. 2, p. 157–168, 2006.

FERREIRA, R. *REST: Princípios e boas práticas*. 2017. Disponível em: <<https://blog.caelum.com.br/rest-principios-e-boas-praticas/>>. Acesso em: 13 out. 2019.

FIGUEIREDO, L. J.; GAFANIZ, A. R.; LOPES, G. S.; PEREIRA, R. Aplicações de acelerômetros. 2007. Disponível em: <<https://docplayer.com.br/17264561-Aplicacoes-de-acelerometros.html>>.

FIREBASE. *Firebase Guides*. 2019. Disponível em: <<https://firebase.google.com/docs/guides>>. Acesso em: 12 out. 2019.

GOOGLE. *Flutter FAQ*. 2019. Disponível em: <<https://flutter.dev/docs/resources/faq>>. Acesso em: 12 out. 2019a.

GOOGLE. *Flutter Technical Overview*. 2019. Disponível em: <<https://flutter.dev/docs/resources/technical-overview>>. Acesso em: 12 out. 2019b.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Pesquisa Nacional de Saúde (PNS)*. 2013. Disponível em: <<http://www.ibge.gov.br/home/estatistica/populacao/pns/2013/>>. Acesso em: 17 mar. 2019.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Pesquisa Nacional por Amostra de Domicílios (PNAD)*. 2017. Disponível em: <<https://www.ibge.gov.br/estatisticas/sociais/populacao/9171-pesquisa-nacional-por-amostra-de-domicilios-continua-mensal.html?=&t=downloads>>. Acesso em: 17 mar. 2019.

JANSEN, J. M.; LOPES, A. J.; JANSEN, U.; CAPONE, D.; MAEDA, T. Y.; NORONHA, A.; MAGALHÃES, G. *Medicina da noite: da cronobiologia à prática clínica*. [S.l.]: SciELO-Editora FIOCRUZ, 2007. 103-120 p. ISBN 978-85-7541-336-4.

LUNSFORD-AVERY, J. R.; ENGELHARD, M. M.; NAVAR, A. M.; KOLLINS, S. H. Validation of the sleep regularity index in older adults and associations with cardiometabolic risk. *Scientific reports*, Nature Publishing Group, v. 8, n. 1, p. 14158, 2018.

MEDIUM. *How to enjoy virtual reality without gyroscope sensor*. 2017. Disponível em: <<https://medium.com/@buckydroid/how-to-enjoy-virtual-reality-without-gyroscope-sensor-b1a5b051c765>>. Acesso em: 13 out. 2019.

MÜLLER, M. R.; GUIMARÃES, S. S. Impacto dos transtornos do sono sobre o funcionamento diário e a qualidade de vida. *Estudos de psicologia*, Pontifícia Universidade Católica de Campinas, v. 24, n. 4, p. 519–528, 2007.

NESTERENKO, T. G.; KOLEDA, A. N.; BARBIN, E. S. Integrated microelectromechanical gyroscope under shock loads. In: IOP PUBLISHING. *IOP Conference Series: Materials Science and Engineering*. [S.l.], 2018. v. 289, n. 1, p. 012003.

NUNES, M. L. Distúrbios do sono. *Jornal de Pediatria*, v. 78, n. 1, p. 63–72, 2002.

PASSARO, V. M. N.; CUCCOVILLO, A.; VAIANI, L.; CARLO, M. D.; CAMPANELLA, C. E. Gyroscope technology and applications: A review in the industrial perspective. 2017. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5677445/>>.

PUB.DEV. *Sensors*. 2019. Disponível em: <<https://pub.dev/packages/sensors>>. Acesso em: 12 out. 2019.

SAMMOBILE. *Samsung Galaxy M10 Specs*. 2019. Disponível em: <<https://www.sammobile.com/samsung/galaxy-m10/specs/SM-M105F/>>. Acesso em: 30 out. 2019.

TECHOPEDIA. *What is a Smartphone? - Definition*. 2019. Disponível em: <<https://www.techopedia.com/definition/2977/smartphone>>. Acesso em: 26 out. 2019.

THE PALLETS PROJECTS. *Flask*. 2019. Disponível em: <<https://palletsprojects.com/p/flask/>>. Acesso em: 12 out. 2019.

THE PALLETS PROJECTS. *Flask Documentation*. 2019. Disponível em: <<https://flask.palletsprojects.com/en/1.1.x/>>. Acesso em: 12 out. 2019.

WEBER, S. A. T.; MONTOVANI, V. C. Doenças do sono associadas a acidentes com veículos automotores: revisão das leis e. *Rev Bras Otorrinolaringol*, SciELO Brasil, v. 68, n. 3, p. 412–5, 2002.

ZANUTO, E. A. C.; LIMA, M. C. S. d.; ARAÚJO, R. G. d.; SILVA, E. P. d.; ANZOLIN, C. C.; ARAUJO, M. Y. C.; CODOGNO, J. S.; CHRISTOFARO, D. G. D.; FERNANDES, R. A. Distúrbios do sono em adultos de uma cidade do estado de são paulo. *Revista Brasileira de Epidemiologia*, SciELO Public Health, v. 18, p. 42–53, 2015.