

# iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA

---

Monitorização não-invasiva com sensores Balistocardiográficos para a gestão do sono

Bernardo Marques Correia da Silva

Mestrado em Engenharia de Telecomunicações e Informática

Orientador:

Prof. Doutor Rui Miguel Neto Marinheiro, Professor

Auxiliar,

ISCTE - Instituto Universitário de Lisboa

Outubro, 2020

Monitorização não-invasiva com sensores Balistocardiográficos para a gestão do sono

Bernardo Marques Correia da Silva

Mestrado em Engenharia de Telecomunicações e Informática

Orientador:  
Prof. Doutor Rui Miguel Neto Marinheiro, Professor  
Auxiliar,  
ISCTE - Instituto Universitário de Lisboa

Outubro, 2020



*Dedico esta dissertação à minha mãe, MARIA DULCE MARQUES MADEIRA  
CORREIA.*



## **Agradecimento**

Quero agradecer a todos os que me ajudaram, de alguma forma, a concluir esta etapa importante na minha vida.

Ao Professor Doutor Rui Neto Marinheiro, agradeço toda a disponibilidade e orientação prestada.

À minha família, por todo o apoio prestado e por estarem sempre lá, tanto nos bons momentos como nos maus.

À minha namorada, por me ajudar a nunca desistir e a seguir em frente na conclusão desta dissertação.



## Resumo

A monitorização do sono tem um impacto importante no quotidiano das pessoas. Nesta monitorização, a primeira metodologia adotada consiste na Polysomnography (PSG). Atualmente, a via intrusiva é a mais precisa e fidedigna (PSG). Logo, existe uma necessidade, nesta área do sono, pela via não-invasiva, Ballistocardiography (BCG).

A metodologia PSG usa sensores caros e a sua realização prende-se apenas com apoio hospitalar. Por outro lado, a metodologia BCG usa sensores mais baratos e apresenta uma utilização doméstica. E, atua sobre o sistema através do som do dispositivo móvel, neste caso, ou de outros tipos de atuadores.

A aplicação "GoToBed" desenvolvida é um meio para se atingir a monitorização do sono com o apoio de um sensor não-intrusivo e atuação. Por forma, a ajudar a pessoa a acordar na fase do sono leve.

A monitorização do sono é realizada com o apoio de um sensor externo denominado SCA11H. Este sensor respeita a metodologia BCG. No âmbito desta dissertação, o Random Forest é o melhor algoritmo de classificação, ao nível de desempenho e de aprendizagem (Waikato Environment for Knowledge Analysis (Weka)). Com este modelo definido, realiza-se a predição das várias fases do sono. Entre elas, o sono acordado, Rapid Eye Movement (REM), leve e profundo. Consequentemente, a estratégia definida para um acordar na fase do sono leve consiste numa análise 15 minutos antes do alarme tocar. Nesta análise, se a pessoa tiver no sono profundo toca um som relaxante. Se não, o alarme volta a tocar, na hora prevista, com um som normal.

**Palavras-chave:** Sono, Balistocardiografia, Polissonografia, Som, Algoritmo, Modelo, Predição, Monitorização.





## Abstract

Sleep monitoring has an important impact on people's daily lives. In this monitoring, the first methodology adopted consists of Polysomnography (PSG). Currently, the intrusive route is the most accurate and reliable (PSG). Therefore, there is a need, in this area of sleep, for the non-invasive route, Ballistocardiography (BCG).

The PSG methodology uses expensive sensors and its realization only relies on hospital support. On the other hand, the BCG methodology uses cheaper sensors and presents a domestic use. And, act on the system through the sound of the mobile device, in this case, or other types of actuators.

The application "GoToBed" developed is a means to achieve sleep monitoring with the support of a non-intrusive sensor and performance. In order to help the person to wake up in the phase of light sleep.

Sleep monitoring is performed with the support of an external sensor called SCA11H. This sensor respects the methodology BCG. Within the scope of this dissertation, Random Forest is the best classification algorithm, in terms of performance and learning (Waikato Environment for Knowledge Analysis (Weka)). With this defined model, the various sleep phases are predicted. Among them, awake sleep, Rapid Eye Movement (REM), light and deep. Consequently, the defined strategy for waking up in the light sleep phase consists of an analysis 15 minutes before the alarm sounds. In this analysis, if the person is in deep sleep it plays a relaxing sound. If not, the alarm sounds again, at the scheduled time, with a normal sound.

**Keywords:** Sleep, Ballistocardiography, Polysomnography, Sound, Algorithm, Model, Prediction, Monitoring.



# Índice

Agradecimento	iii
Resumo	v
Abstract	vii
Lista de Acrónimos	xi
Lista de Tabelas	xiii
Lista de Figuras	xv
Lista de Fórmulas	xvii
Capítulo 1. Introdução	1
1.1. Enquadramento e Motivação	1
1.2. Objetivos	2
1.3. Questões de Investigação	2
1.4. Método de Investigação	3
1.5. Estrutura da Dissertação	4
Capítulo 2. Revisão da Literatura	7
2.1. Fases do Sono	7
2.2. PSG versus BCG	8
2.3. Algoritmos de Classificação	12
2.3.1. Random Forest	13
2.3.2. PART	14
2.3.3. J48	16
2.3.4. IBk	18
2.4. Aplicações	19
2.4.1. Beddit	19
2.4.2. DREEM 2	20
2.4.3. Emfit	20
2.4.4. Pillow	21
2.4.5. ZEEQ	22
2.4.6. Sleep As Android	23
2.4.7. Comparação entre Aplicações	23
Capítulo 3. Desenho e Implementação da Proposta	27

3.1. Proposta de monitorização e atuação	27
3.2. Implementação da Proposta	30
3.2.1. Sensor	32
3.2.2. Filtragem dos dados	35
3.2.3. Aprendizagem de modelos	40
Capítulo 4. Testes e Resultados	43
4.1. Constante de filtragem (K)	43
4.2. Avaliação dos Algoritmos de classificação	45
4.3. Aprendizagem dos Algoritmos de classificação	47
4.4. Análise Funcional	53
4.5. Atividades	55
Capítulo 5. Conclusões	59
5.1. Conclusões, Limitações e Trabalho Futuro	59
Referências Bibliográficas	63
Apêndice A. Anexos	67

## Lista de Acrónimos

**Acc:** Accelerometer.

**API:** Application programming interface.

**App:** Application.

**ARFF:** Attribute-Relation File Format.

**B2B:** Beat-to-Beat time.

**B2B’:** Beat-to-Beat time 1.

**B2B’’:b> Beat-to-Beat time 2.**

**BCG:** Ballistocardiography.

**bpm:** beat per minute.

**BSN:** Body Sensor Network.

**CSS:** Cascading Style Sheets.

**dHRV:** derived Heart Rate Variability.

**ECG:** Electrocardiography.

**EEG:** Electroencephalography.

**EMG:** Electromyography.

**EOG:** Electro-oculography.

**FP:** False Positives.

**HF:** High Frequency.

**HFHRV:** High Frequency Heart Rate Variability.

**HR:** Heart Rate.

**HRFILT:** Heart Rate Filtered.

**HRV:** Heart Rate Variability.

**HRVFILT:** Heart Rate Variability Filtered.

**HTML:** Hypertext Markup Language.

**IP:** Internet Protocol.

**KNN:** K-nearest neighbors.

**LF:** Low Frequency.

**LFHRV:** Low Frequency Heart Rate Variability.

**MEMS:** Micro-electromechanical Systems.

**ML:** Machine Learning.

**NREM:** Non-Rapid Eye Movement.

**PSG:** Polysomnography.

**QR:** Quick Response.

**Rdepth:** Respiration depth.

**REM:** Rapid Eye Movement.

**rms:** root mean square.

**RR:** Respiration Rate.

**RRFILT:** Respiration Rate Filtered.

**RRV:** Respiration Rate Variability.

**SS:** Signal Strength.

**SSDP:** Simple Service Discovery Protocol.

**SSFILT:** Signal Strength Filtered.

**Std:** Standard deviation.

**SV:** Stroke Volume.

**SVFILT:** Stroke Volume Filtered.

**SVV:** Stroke Volume Variability.

**TCP:** Transmission Control Protocol.

**TP:** True Positives.

**UART:** Universal Asynchronous Receiver/Transmitter.

**UDP:** User Datagram Protocol.

**Weka:** Waikato Environment for Knowledge Analysis.

## Lista de Tabelas

2.1	HR, HRV, RR, RRV e Rdepth típicos em cada fase do sono. [40]	12
2.2	Comparação entre atuadores presentes em todas as Aplicações.	24
2.3	Comparação entre todas as aplicações referenciadas e analisadas ao longo desta dissertação.	24
2.4	Comparação entre sensores presentes em todas as Aplicações.	25
3.1	Principais tecnologias usadas na implementação da aplicação GoToBed.	30
3.2	Parâmetros de saída do sensor BCG SCA11H. [13]	33
3.3	Constantes iniciais (instante $t = 0$ segundos) para os seguintes parâmetros: HRFILT, LFHRV, HFHRV, SVFILT, SVV, dHRV, HRVFILT, RRFILT, RRV e SSFILT.	40
4.1	Valores da constante de filtragem para cada expressão de filtragem.	45
4.2	Classificação dos diferentes algoritmos de classificação, ao nível, de performance.	47
4.3	Dados para exemplo referente ao cálculo da estatística do kappa. [31]	50
4.4	Instâncias classificadas corretamente nos diferentes algoritmos (%).	50
4.5	Estatística do kappa nos diferentes algoritmos.	51
4.6	Fases do sono.	53





## Lista de Figuras

2.1	Progressão das várias fases do sono num adulto normal e saudável, durante uma noite. [42]	8
2.2	Atividade EEG característica de cada um dos quatro estados do sono NREM ( $\mu\text{V}/\text{sec}$ ). NOTA: No estado 2, a seta indica um complexo K e o sublinhado mostra dois fusos adormecidos. [42]	8
2.3	Mecanismo envolvido na monitorização do sono no PSG. [48]	9
2.4	Processo de monitorização do sono no BCG. [10]	10
2.5	Sinal BCG típico (Accelerometer (Acc)[ $\mu\text{g}$ ], root mean square (rms)). [40]	10
2.6	Correlação entre a razão Heart Rate Variability (HRV) de BCG e ECG. [40]	11
2.7	Frequência cardíaca noturna e correlação - BCG versus PSG. [40]	11
2.8	Classificação do algoritmo Random Forest. [59]	14
2.9	Metodologia de trabalho do algoritmo Random Forest. [59]	14
2.10	Exemplo: forma como o algoritmo, PART, constrói uma árvore parcial. [61]	16
2.11	Funcionamento do algoritmo KNN. [55]	18
2.12	Monitor de sono usado pela aplicação Beddit.	19
2.13	Monitor de sono usado pela aplicação Dreem 2	20
2.14	Monitor de sono usado pela aplicação Emfit QS.	21
2.15	Monitor de sono usado pela aplicação ZEEQ Smart Pillow.	22
3.1	Sequência estrutural proposta para implementação da construção e utilização do modelo.	28
3.2	Proposta do diagrama de sequência funcional da aplicação, em estudo.	29
3.3	Implementação da proposta referente à aplicação GoToBed.	31
3.4	Estrutura do sensor SCA11H. [9]	32
3.5	Protocolo SSDP. [13]	33
3.6	Possíveis opções de posicionamento para o sensor SCA11H. [12]	34
3.7	Diagrama de blocos referente ao sensor SCA10H. [11]	34
3.8	Estrutura de dados de entrada e saída no processo de filtragem.	35
3.9	Dependências entre as expressões presentes na filtragem de dados.	39
3.10	Estrutura dos dados para efetuar a predição para uma dada noite.	41

4.1	Constantes de filtragem iguais ao inverso de potências de base 2 (6, 7, 8, 9, 10 e 11) do sinal HFHRV.	44
4.2	Diagrama de atividades para efetuar o teste de desempenho de cada Algoritmo de classificação, em estudo, no conjunto de dados de teste.	46
4.3	Estrutura dos dados para efectuar a aprendizagem dos algoritmos (Ficheiro ARFF).	47
4.4	Diagrama de atividades para efetuar a aprendizagem de cada Algoritmo de classificação no conjunto de dados de teste.	48
4.5	Gráficos entre os dados do estado do sono da noite PSG14 e a predição BCG para o algoritmo Random Forest.	52
4.6	Matrizes de confusão entre os dados do estado do sono da melhor noite e a predição BCG para os melhores algoritmos de classificação Weka, em estudo.	53
4.7	Análise funcional da aplicação, em estudo (Música relaxante é acionada).	54
4.8	Análise funcional da aplicação, em estudo (Música relaxante não é acionada).	55
4.9	Atividades referentes à aplicação Android - GoToBed.	56

## Lista de Fórmulas

2.1	Heart Rate Variability (HRV)	12
2.2	Teoria de informação da entropia ( $H(x)$ )	17
2.3	Coefficiente de correlação da classificação de Kendall ( $\tau$ )	17
3.1	Exponencial filtro passa-baixo ( $y(t)$ )	36
3.2	Heart Rate Filtered (HRFILT)	36
3.3	Low Frequency Heart Rate Variability (LFHRV)	36
3.4	High Frequency Heart Rate Variability (HFHRV)	37
3.5	Stroke Volume Filtered (SVFILT)	37
3.6	Stroke Volume Variability (SVV)	37
3.7	derived Heart Rate Variability (dHRV)	37
3.8	Heart Rate Variability Filtered (HRVFILT)	37
3.9	Respiration Rate Filtered (RRFILT)	38
3.10	Respiration Rate Variability (RRV)	38
3.11	Signal Strength Filtered (SSFILT)	38
4.1	Estatística do kappa ( $K$ )	49
4.2	Precisão Observada ( $P_r(a)$ )	49
4.3	Precisão Esperada ( $P_r(e)$ )	49



## CAPÍTULO 1

### Introdução

#### 1.1. Enquadramento e Motivação

O tema referente a esta dissertação concentra-se na gestão de descanso através da monitorização do sono. Esta ideia surgiu através de um problema centrado na monitorização do sono, a partir de casa. Isto é, sem o uso de métodos invasivos. A aplicação desenvolvida, em Android, é um meio para atingir o objetivo.

Por outro lado, esta aplicação tem um impacto positivo, no meio ambiente, ao tentar acordar sempre o utilizador na fase do sono mais leve. Esta solução ajuda a pessoa a não acordar sobressaltada, por se encontrar a dormir na fase do sono profundo.

O método PSG é invasivo, dispendioso e necessita de um técnico especialista para identificar as fases do sono. Logo, o método a usar é o BCG, visto que, não é intrusivo e pode ser utilizado em casa. Assim, não é necessário qualquer tipo de ajuda hospitalar por parte de médicos/especialistas na identificação das fases do sono, durante uma noite. [41] [40]

Esta aplicação vai trabalhar em conjunto com um sensor externo que, por sua vez, usa a metodologia BCG. Tanto a área dos sensores como a área da Machine Learning (ML) estão interligadas, ao nível do tratamento dos dados provenientes do sensor, por forma a construir um modelo definitivo. Neste caso, o modelo final prende-se com a predição das fases do sono, durante uma noite. Mas, poderíamos usar estes dados do sensor BCG para construir outro tipo de modelo.

ML é uma área em ascensão e representa uma aplicação de inteligência artificial que fornece aos sistemas a capacidade de aprender e melhorar automaticamente com a experiência, sem ser explicitamente programado. O objetivo principal de ML é permitir que os computadores aprendam automaticamente sem intervenção humana. [63] [56]

Todos estes dados estão sujeitos a um processo de filtragem. Este tipo de filtragem consiste num filtro passa-baixo (janela deslizante). O uso deste filtro incide sobre a eliminação do ruído nos sinais. Ou seja, quanto menor a constante de filtragem, menor o peso das amostras recentes e menor a sensibilidade às frequências elevadas, daí eliminando, mais facilmente, o ruído presente no sinal. Por outro lado, quanto menos peso tiver o histórico, maior é a suscetibilidade ao ruído.

Em suma, as metodologias mais usadas, nesta área do sono, correspondem ao PSG e BCG. O BCG é a mais recente e foi desenvolvida tendo como base o PSG. E, o seu principal objetivo prende-se com a obtenção de um nível de precisão idêntico ao de PSG. De forma, a realizar uma monitorização do sono fidedigna. Logo, para atingir este fim, vão decorrer estudos, tanto na área dos sensores como no tipo de materiais e equipamentos

utilizados. Os principais tipos de equipamentos baseiam-se em acelerómetros (sensores de posição) e giroscópios (sensores de movimento).

Foi submetido um artigo baseado nesta dissertação para a conferência "ConfTele 2021" e está disponível nos anexos.

## **1.2. Objetivos**

O objetivo principal prende-se com a monitorização do sono através de sensores externos não-invasivos e atuação no meio ambiente. A aplicação desenvolvida é um meio para se atingir o mesmo.

Consequentemente, o sensor externo (BCG) apoia a monitorização. Enquanto que, o som do dispositivo móvel apoia a atuação. Este sensor externo a definir, tem de apresentar um preço razoável e um fácil manuseamento por parte do utilizador, tanto a nível de configuração como de calibração. Mas, também um conteúdo simples que inclua os parâmetros de saída necessários para solucionar o objetivo.

Por fim, a aplicação desenvolvida ajuda a acordar a pessoa na fase do sono leve. Assim, é possível determinar mais tarde através de um estudo, se ao alterar o meio ambiente, se melhora a qualidade do sono.

## **1.3. Questões de Investigação**

As questões de investigação presentes, nesta secção, vão ser respondidas no capítulo referente às Conclusões. Todas elas estão interligadas e têm um propósito ao longo do desenvolvimento da dissertação. Consequentemente, os principais tópicos abordados recaem sobre as diferentes metodologias, os sensores internos ou externos, os atuadores, a monitorização e qualidade do sono e, por último, o algoritmo de classificação mais adequado.

- 1 – Quais as metodologias a utilizar na aplicação móvel?
- 2 – No desenvolvimento desta aplicação, quais os sensores (internos e/ou externos) e atuadores a usar?
- 3 – De que forma, é realizada toda a monitorização do descanso?
- 4 – Como avaliar, de forma credível, a qualidade do sono?
- 5 – Qual é o algoritmo de classificação mais adequado, neste estudo?
- 6 – Será que se consegue obter o mesmo nível de qualidade da monitorização do sono, com sensores mais baratos?

#### 1.4. Método de Investigação

O ponto de entrada referente a esta ideia surgiu através de um estudo centrado na monitorização do sono, a partir de casa. Isto é, sem o uso de métodos invasivos. Consequentemente, é necessário efetuar um levantamento de requisitos sobre as aplicações existentes com um maior número possível de diferenças entre elas, de forma a se concluir qual a que apresenta uma melhor metodologia. E, uma melhor monitorização referente à qualidade do sono, aos sensores utilizados e à atuação imposta por cada Application (App). A metodologia a adotar é o BCG.

De acordo, com os sensores internos e/ou externos utilizados nas mesmas, a próxima fase consiste nas escolhas, ao nível dos sensores a usar dependendo dos recursos disponíveis e do tempo para realizar esta App. Os sensores externos necessitam de efetuar um processo de configuração e calibração para cada tipo de cama. A escolha do sensor recaiu sobre o sensor externo SCA11H.

Depois de todo este processo de escolha, passamos para uma fase de tratamento dos dados provenientes do sensor, em estudo, ao nível de filtragem. De seguida, é necessário definir qual o melhor algoritmo de classificação a usar em toda a fase de aprendizagem do modelo final. A avaliação de desempenho é realizada com o apoio da ferramenta Weka Experiment Environment. Os algoritmos de classificação escolhidos para efetuar a avaliação do desempenho, sobre um conjunto de dados de teste, são os seguintes: rules.ZeroR, bayes.NaiveBayes, functions.Logistic, functions.SMO, lazy.IBk, rules.PART, trees.REPTree, trees.J48 e trees.RandomForest. De acordo com o resultado final da avaliação, obtém-se os quatro melhores algoritmos de classificação. Assim, para cada um destes algoritmos é efetuada uma aprendizagem do modelo através da ferramenta Weka Explorer. Com o fim de toda esta aprendizagem, retira-se o modelo final que obteve melhores resultados. E, por fim, concluímos o processo com a predição das várias fases do sono. [7] [6] [5] Neste caso, o Random Forest é o melhor algoritmo de classificação.

O desenvolvimento desta aplicação móvel consiste no uso de uma ferramenta crucial denominada Android Studio que, por sua vez, utiliza a linguagem JAVA. Este desenvolvimento está estruturado, ao nível, do alarme, da filtragem dos parâmetros de saída do sensor externo, da Application programming interface (API) referente ao algoritmo de classificação e respetiva predição da fase do sono e, por fim, da conexão entre o sensor externo e a aplicação.

Em suma, com os valores preditivos referentes às fases do sono, é possível analisar os ciclos do sono, controlar a fase do sono em que a pessoa acorda e atuar sobre a mesma. Este tipo de atuação ocorre através do som do dispositivo móvel. Por sua vez, a atuação perfeita consiste em acordar a pessoa na fase do sono leve. Desta forma, quinze minutos antes da hora prevista para tocar o alarme é realizada uma análise. ou seja, se a pessoa nesse instante se encontrar na fase do sono mais profundo, é despoletada uma música relaxante. Enquanto que, se encontrar noutra fase diferente da profunda, apenas é iniciada uma música normal na hora definida previamente no alarme.



## 1.5. Estrutura da Dissertação

O presente estudo está organizado em 4 capítulos que pretendem refletir todo o processo de aprendizagem desta dissertação até à sua conclusão.

O primeiro capítulo introduz o tema da metodologia de investigação, motivação e objetivos da mesma, bem como uma breve descrição da estrutura do trabalho.

O segundo capítulo reflete o enquadramento teórico, designado por Revisão da literatura. Este capítulo contém explicações sobre temas existentes, entre eles, as fases do sono, aplicações relacionadas com o tema da dissertação, BCG e PSG e, por último, algoritmos de classificação.

O terceiro capítulo apresenta a descrição referente ao desenho e implementação da proposta e, também, a análise dos resultados obtidos, de acordo com a metodologia que se entendeu mais apropriada. Esta metodologia está diretamente relacionada com o melhor algoritmo de classificação, em estudo, ao nível de performance. Toda esta descrição consiste nos seguintes subcapítulos: Proposta de monitorização e atuação, Implementação da Proposta (Sensor, Filtragem dos dados e Aprendizagem de modelos) e Testes e Resultados (Constante de filtragem (K), Avaliação e Aprendizagem dos Algoritmos de classificação, Análise Funcional e Atividades presentes na App).

No quarto e último capítulo apresentam-se as conclusões deste estudo, bem como as recomendações, limitações e trabalhos futuros.





## CAPÍTULO 2

### Revisão da Literatura

#### 2.1. Fases do Sono

A arquitetura do sono representa a organização das várias fases do sono, durante uma noite normal de descanso, como ilustrado na Figura 2.1. Os dois tipos de sono existentes correspondem ao sono de movimentos oculares não rápidos (Non-Rapid Eye Movement (NREM)) e ao sono de movimentos oculares rápidos (REM). De acordo com a figura 2.1, o sono NREM é constituído por quatro estados do sono (1, 2, 3 e 4) e alternam ciclicamente com o sono REM. Se esta alternância se tornar irregular, quer dizer que existem distúrbios no sono. Estes estados e ciclos do sono foram descobertos com o uso de registos de atividade cerebral, isto é, Electroencephalography (EEG). [24] [57] [42] Estes 4 estados do sono NREM podem-se transformar em 2 estados do sono. Ou seja, as fases do sono 1 e 2 correspondem ao estado do sono mais leve, enquanto que, as fases do sono 3 e 4 correspondem ao estado do sono mais profundo.

Um episódio de sono típico depende de várias etapas para se chegar ao sono REM, começando com um curto período na fase do sono NREM 1, progredindo para a fase 2 e, de seguida, pelas fases 3 e 4. Por sua vez, o sono NREM constitui, aproximadamente, 80% do tempo total gasto no sono e o sono REM constitui os 20% restantes. [25] Um ciclo do sono corresponde à passagem entre todas as fases do sono (NREM e REM). O primeiro ciclo do sono tem a duração média de 70 a 100 minutos, enquanto que os posteriores apresentam uma duração média entre 90 a 120 minutos, aproximadamente. [24] [57] [42]

A fase do sono NREM 1 desempenha um papel de transição entre cada ciclo do sono. Geralmente, este estado do sono dura 1 a 7 minutos no ciclo inicial do sono. O sono no estado 2 dura, aproximadamente, 10 a 25 minutos no ciclo inicial e nos ciclos sucessivos (representa entre 45 a 55 por cento do tempo total gasto no sono). Já, a fase do sono 3 dura apenas alguns minutos e constitui 3 a 8 por cento do tempo total gasto no sono. Por fim, a última fase 4 dura, aproximadamente, 20 a 40 minutos no primeiro ciclo (representa entre 10 a 15 por cento do tempo total gasto no sono). Por outro lado, a fase do sono REM pode durar apenas 1 a 5 minutos no ciclo inicial do sono e tende a prolongar-se à medida que o sono progride. [24] [57] [42]

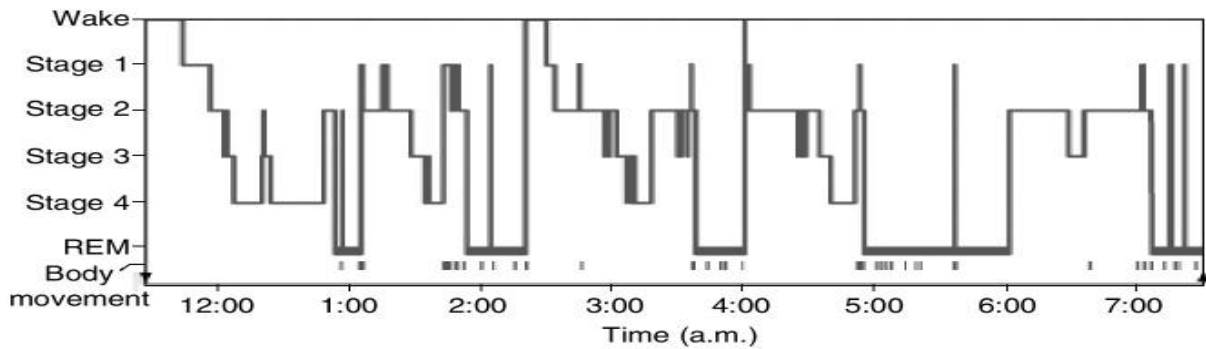


FIGURA 2.1. Progressão das várias fases do sono num adulto normal e saudável, durante uma noite. [42]

Consequentemente, a atividade cerebral no EEG na fase 1 transita do estado acordado para ondas de baixa voltagem e frequência mista (figura 2.2). A figura 2.2, indica, para a fase 2, uma atividade cerebral com uma voltagem relativamente baixa e uma frequência mista caracterizada pela presença de fusos do sono e complexos K. Na fase 3, o EEG indica alta voltagem e lenta atividade de onda (figura 2.2). Por último, na fase do sono NREM 4, a atividade cerebral no EEG caracteriza-se em quantidades aumentadas de alta voltagem e lenta atividade de onda (figura 2.2). Por outro lado, o sono REM apresenta atividades de ondas cerebrais dessincronizadas (baixa voltagem e frequência mista), atonia muscular e surtos rápidos de movimentos oculares. É nesta fase REM onde ocorrem a maior parte dos sonhos. [24] [57] [42]

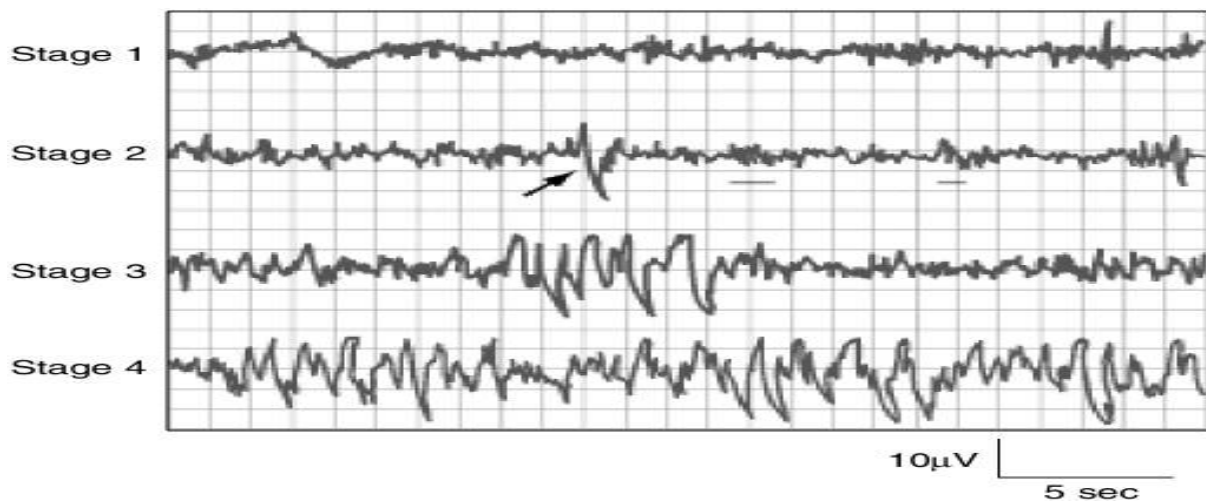


FIGURA 2.2. Atividade EEG característica de cada um dos quatro estados do sono NREM ( $\mu\text{V}/\text{sec}$ ). NOTA: No estado 2, a seta indica um complexo K e o sublinhado mostra dois fusos adormecidos. [42]

## 2.2. PSG versus BCG

O PSG (Figura 2.3) é um método intrusivo, caro e complexo na análise do sono. Apenas é adequado para medições contínuas de curto prazo. A monitorização e a análise do sono são realizadas por técnicos e médicos altamente especializados, em ambiente de

laboratório. Esta medição requer o uso de vários tipos de sensores. Assim, tornou-se importante encontrar novas formas de monitorizar o sono de uma forma mais eficaz, barata e contínua a longo prazo. Consequentemente, surgiram os métodos baseados na balistocardiografia (BCG) que apresentam um potencial crescente na análise do sono. O PSG apresenta dezoito canais para medir os vários tipos de sinais, como ilustrado na figura 2.3. Entre eles, seis eléctrodos para medir a atividade cerebral com o apoio do procedimento de EEG, dois eléctrodos para medir os movimentos oculares usando o método de (Electro-oculography (EOG)), dois eléctrodos para medir a tensão da musculatura mandibular através da Electromyography (EMG), três eléctrodos para medir a função cardíaca com a Electrocardiography (ECG) e, por fim, cintos de tórax e abdómen para medir o esforço respiratório. [64] [47]

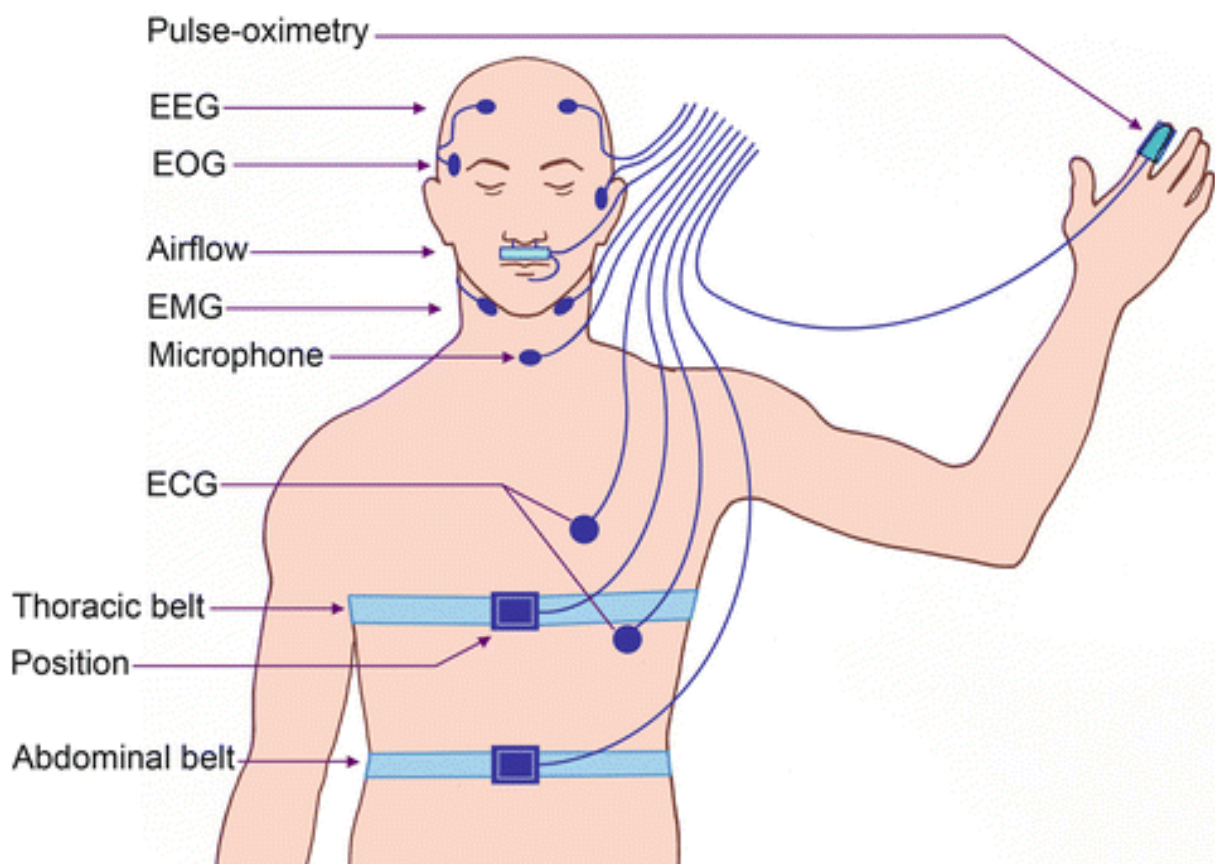


FIGURA 2.3. Mecanismo envolvido na monitorização do sono no PSG. [48]

O BCG (Figura 2.4) é um método que não possui qualquer tipo de mecanismo de carácter intrusivo. Tem a capacidade de medir o bombeamento mecânico do coração e fornecer o tempo e o volume sistólico relativo (Stroke Volume (SV)) de cada batimento cardíaco. É uma excelente ferramenta para medir a recuperação e a qualidade do sono. Uma boa noite de sono é importante no funcionamento do coração ou dos vasos sanguíneos, no bem-estar mental e na prevenção de tipos de cancro. Os atletas para atingirem bons resultados, ao nível do treino e competição, necessitam de uma boa qualidade de sono. Os parâmetros referentes ao batimento cardíaco e à respiração medem, também, a condição do

sistema nervoso autónomo e indicam muitas condições patológicas. Entre elas, a variação do volume sistólico como uma medida da condição do fluído corporal. [41] [40]

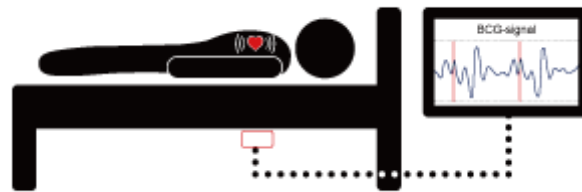


FIGURA 2.4. Processo de monitorização do sono no BCG. [10]

A figura 2.5, representa um sinal BCG típico. Este sinal, consiste no recuo causado pelo sangue fluindo para a aorta, para a rotação da aorta e para o pulso da pressão arterial, continuando pelas artérias. Este movimento ocorre, principalmente, ao longo do eixo longitudinal do corpo e a amplitude de aceleração é, tipicamente, da ordem de 1 mg. Como existem grandes variações nas constantes de mola e fatores de amortecimento em camas, obtém-se resultados com diferenças na modulação de amplitude de sinal e constantes de tempo de atenuação (Figura 2.5,  $d \rightarrow e \rightarrow b \rightarrow c \rightarrow a$ ). [41] [40]

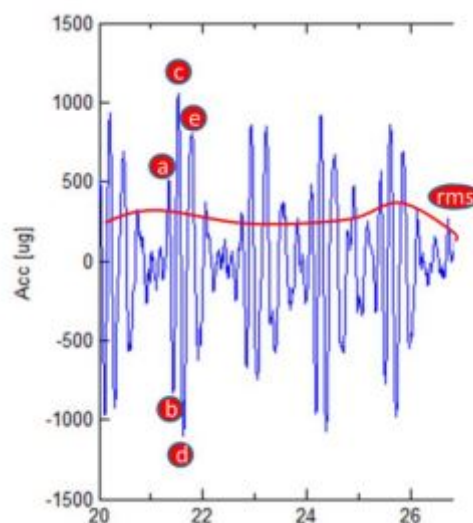


FIGURA 2.5. Sinal BCG típico (Accelerometer (Acc)[ $\mu\text{g}$ ], root mean square (rms)). [40]

Os níveis de stress e recuperação são medidos através da divisão entre a variabilidade da frequência cardíaca de alta e baixa frequência. Este cálculo é usado para avaliar os resultados do teste, ao nível do desempenho. Se as pessoas se movessem ou falassem, os resultados eram negativos. Assim, apenas se contabilizaram as gravações com sinal BCG relevante por mais de 80% do tempo. A correlação obtida entre o BCG e o ECG resultou em, aproximadamente, 96%, como ilustrado na figura 2.6. [41] [40]

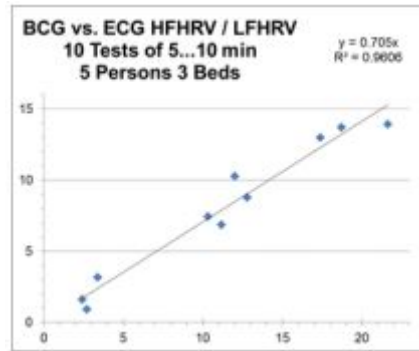


FIGURA 2.6. Correlação entre a razão Heart Rate Variability (HRV) de BCG e ECG. [40]

A figura 2.7 corresponde à comparação entre o BCG e o PSG, na análise do sono durante uma noite. No PSG, o padrão mais fidedigno para análise do sono, os batimentos cardíacos são medidos através de uma cinta torácica ECG. Para a Heart Rate (HR), a diferença, durante a noite, medida entre BCG e PSG foi de  $-0,5 \pm 3,2$  beat per minute (bpm) (Média  $\pm 2 \times$  Standard deviation (Std)) e a correlação 97% para médias de 30 minutos. As correlações para High Frequency Heart Rate Variability (HFHRV) e Low Frequency Heart Rate Variability (LFHRV) foram 67% e 71% respectivamente. [41] [40]

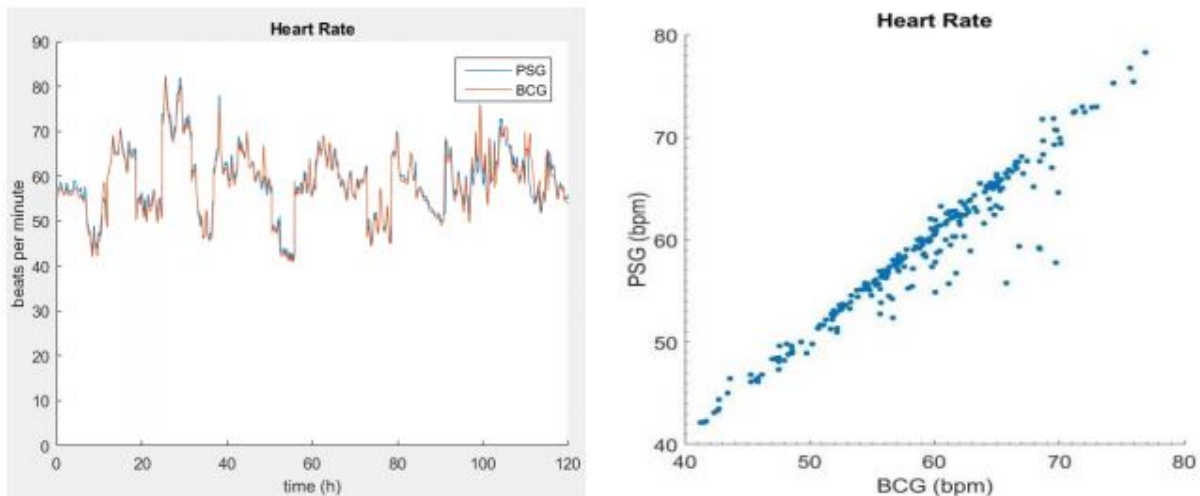


FIGURA 2.7. Frequência cardíaca noturna e correlação - BCG versus PSG. [40]

A frequência cardíaca (HR) e a variabilidade da frequência cardíaca (HRV) são calculadas através do sinal de ECG. Para High Frequency (HF), está definida uma variabilidade de frequência cardíaca (HRV) entre 0,15 - 0,4 Hz. Para Low Frequency (LF), está definida uma variabilidade de frequência cardíaca (HRV) entre 0,05 - 0,15 Hz. A Respiration Rate (RR) e a Respiration Rate Variability (RRV) são medidas a partir do sinal do cinto torácico. A Respiration depth (Rdepth) está relacionada com a diferença de amplitude entre os picos de expiração e inspiração do sinal. A variabilidade da frequência cardíaca (HRV) relativa foi calculada de acordo com a Equação 2.1. [64] [47]



### 2.1. Heart Rate Variability (HRV)

$$HRV = \frac{HFHRV}{LFHRV * Rdepth} \quad (2.1)$$

O Rdepth foi adicionado na Equação 2.1 para filtrar a modulação da variabilidade da frequência cardíaca (HRV) causada pela respiração. [47]

Os estados do sono foram avaliados, visualmente, a partir das gravações de EEG, EOG e EMG em períodos de 30 segundos por um técnico qualificado. Os critérios da Academia Americana de Medicina do Sono foram usados na respetiva pontuação. [64] [47]

De acordo com a tabela 2.1, observa-se que a frequência cardíaca (HR) aumenta durante o estado do sono REM e acordado, comparando com o sono profundo e leve. A variabilidade da frequência cardíaca apresenta um nível mais alto no sono profundo e um nível mais baixo no sono REM e acordado. Entre os estados do sono não foram registadas diferenças na taxa de respiração (RR), daí o valor médio. A taxa de respiração variável registou uma diminuição durante o sono profundo e um aumento no sono REM e acordado (valor mais alto). A profundidade respiratória (Rdepth) estava no nível mais baixo durante o sono profundo e no nível mais alto durante o sono REM e acordado. (Tabela 2.1). [47]

TABELA 2.1. HR, HRV, RR, RRV e Rdepth típicos em cada fase do sono. [40]

	Sono Leve	Sono Profundo	Sono REM	Acordado
HR	Diminuiu	Diminuiu	Aumentou	Aumentou
HRV	Médio	Aumentou	Diminuiu	Diminuiu
RR	Médio	Médio	Médio	Médio
RRV	Médio	Baixo	Aumentou	Alto
Rdepth	Médio	Diminuiu	Aumentou	Aumentou

A fase do sono REM é onde ocorre a maior parte dos sonhos e é importante para a recuperação mental, enquanto que o sono profundo é importante para a recuperação física. Quando a pessoa adormece, a frequência cardíaca (HR) diminui, aproximadamente, 10 bpm e a respiração torna-se constante (RRV diminui). De seguida, a variabilidade da frequência cardíaca (HRV) começa a aumentar, indicando uma transição do sono leve para o profundo. A fase do sono NREM corresponde ao estado do sono leve, sono profundo e acordado. [41] [40]

### 2.3. Algoritmos de Classificação

Os algoritmos de classificação, em estudo, que apresentam um melhor desempenho (usados na aprendizagem do modelo) são os seguintes: Random Forest, IBk (K-nearest neighbors (KNN)), J48 e PART. Será demonstrado, mais à frente, que o algoritmo Random Forest foi o que apresentou melhores resultados, no âmbito desta dissertação.

Ao comparar os resultados desta dissertação com outros trabalhos científicos [25] [62], existem algumas semelhanças com a linha de investigação e opções adotadas neste trabalho. Entre elas, a classificação das fases do sono de 30 em 30 segundos, o uso das metodologias BCG e PSG e, por fim, o uso do algoritmo de classificação KNN. O estudo apresentado nesta tese considera um número de noites de PSG inferior ao artigo [25] e superior ao artigo [62]. Contudo, inclui uma maior variedade de tipo de amostras, e.g. ao nível das idades. Nesta dissertação, o mecanismo de multi-classificação é mais elaborado, visto que, contém um processo de filtragem, antes de efetuar o processo de avaliação e aprendizagem do modelo, eliminando por isso ruídos não desejáveis, idealmente potenciando, por sua vez, a predição das fases do sono. Apesar de não ser fácil comparar os resultados finais, porque o ponto de partida é distinto, ou seja, o conjunto de dados de teste e aprendizagem é diferente, pode-se concluir que o algoritmo Random Forest poderá apresentar vantagem em relação ao IBk. Outra vantagem significativa, em relação aos trabalhos anteriores, está presente ao nível da atuação que, como se descreverá mais à frente, procura acordar o utilizador na fase mais leve do sono.

Seguidamente serão apresentados as principais características e potencialidades dos vários algoritmos considerados durante o estudo desta tese.

### 2.3.1. Random Forest

O algoritmo de classificação Random Forest é a abordagem de ML mais versátil do mundo. Representa uma técnica de modelagem preditiva que atua como uma abordagem de tomada de decisão. É um algoritmo semelhante à árvore de decisão, que se baseia numa árvore composta por várias árvores de decisão, juntando os resultados na saída, por forma, a melhorar o desempenho do modelo (figura 2.8). O Random Forest treina cada árvore num conjunto separado de observações e divide os nós em cada árvore, assumindo um número limitado de atributos. De acordo com a figura 2.8, a decisão final é garantida, através da média das previsões de cada árvore individual. [34] [32] [59]

O funcionamento metodológico do algoritmo Random Forest baseia-se numa seleção de amostras aleatórias provenientes do conjunto de dados disponível usado para efetuar a aprendizagem. Consequentemente, cria uma árvore de decisão para cada amostra e obtém os resultados estimados de cada árvore de decisão (figura 2.9). De seguida, é realizada a votação dos resultados estimados, como ilustrado na figura 2.9. Por fim, o resultado mais votado da predição é reconhecido como o resultado estimado do terminal. [34] [32] [59]

O algoritmo Random Forest é aplicável em problemas de regressão e classificação, com a diferença que a variável é contínua e categórica para regressão e classificação, respetivamente. Por sua vez, este algoritmo tem a capacidade de fornecer uma estimativa aproximada de variáveis importantes na classificação e propor procedimentos experimentais para descobrir as interações entre as variáveis. Também, apresenta a capacidade de manter a precisão, enquanto uma grande parte dos dados está em falta no conjunto de dados disponível. Em suma, quanto maior a pureza, maior a oportunidade de tomar decisões certas. [34] [32] [59]

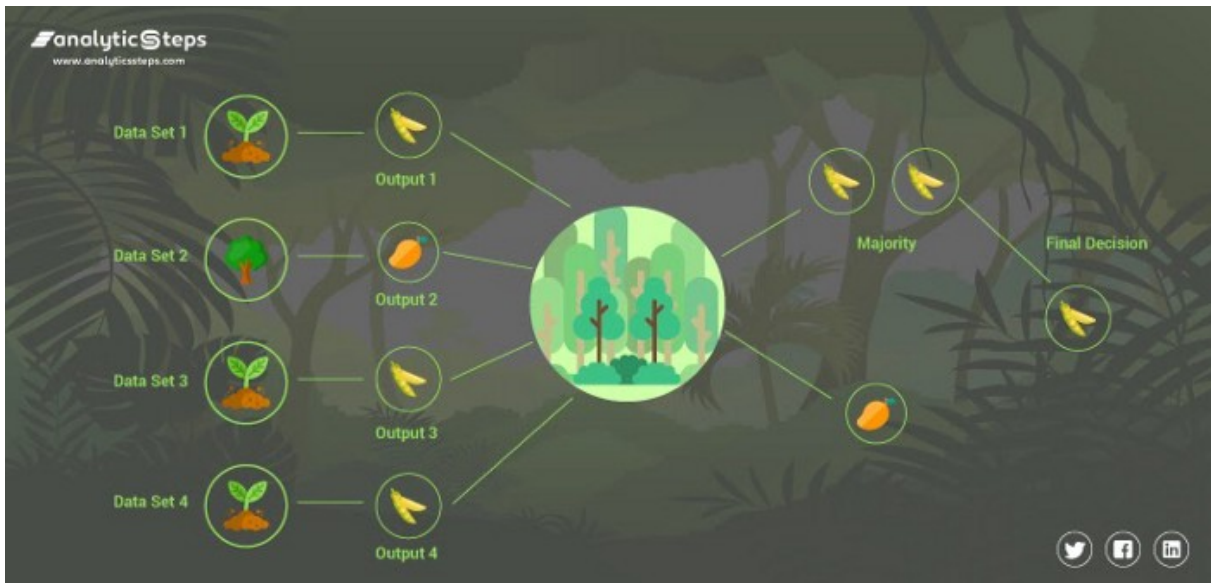


FIGURA 2.8. Classificação do algoritmo Random Forest. [59]

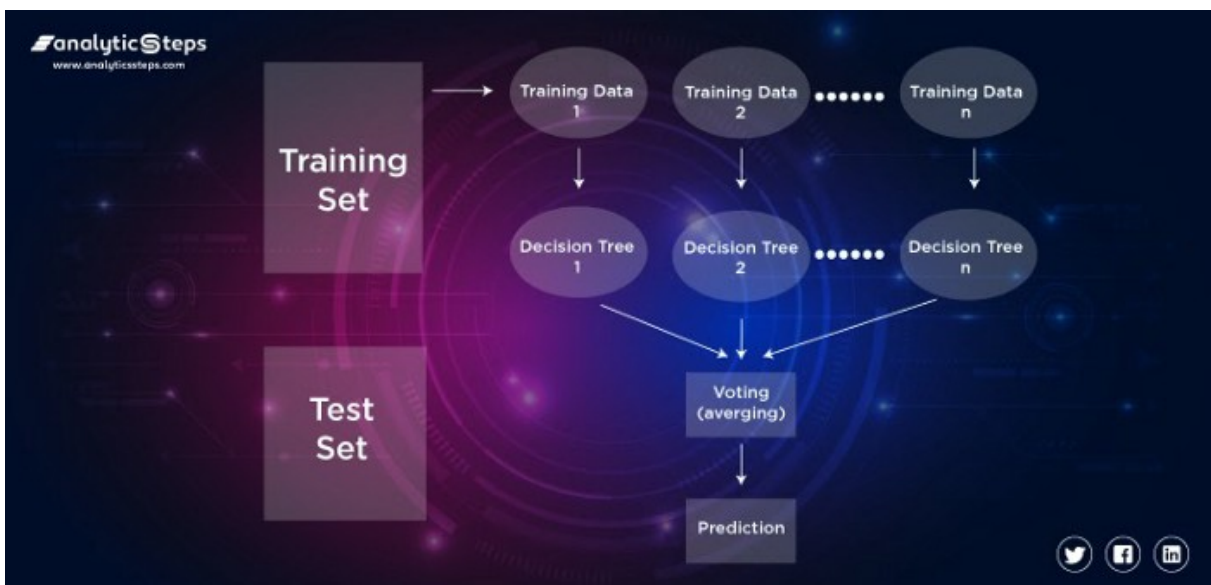


FIGURA 2.9. Metodologia de trabalho do algoritmo Random Forest. [59]

Um exemplo típico do algoritmo Random Forest, corresponde à procura de clientes leais e fraudulentos num banco. Mas, também, à descoberta de um tipo de doença, investigando os relatórios médicos dos pacientes (aplicado na medicina). [34] [32] [59]

### 2.3.2. PART

O algoritmo de classificação PART é o novo método para indução de regras. Consequentemente, este algoritmo para produzir conjuntos de regras precisos não necessita de realizar uma optimização global. Tem como principal vantagem a simplicidade adicionada. No modo de funcionamento, adota uma estratégia de separar e conquistar, na medida em que constrói uma regra. De modo que, remove as instâncias que a regra cobre e continua

criando regras, recursivamente, para as instâncias restantes. Por exemplo, para se criar uma única regra, uma árvore de decisão podada é construída para o conjunto atual de instâncias. De seguida, a folha com a maior cobertura é transformada numa regra e a árvore é descartada. Isto evita a generalização precipitada, generalizando apenas quando as implicações são conhecidas. Por sua vez, usar uma árvore podada para obter uma regra em vez de construí-la incrementalmente adicionando conjunções uma de cada vez, evita o problema de poda excessiva da separação básica. Contudo, ao usar a metodologia separar e conquistar em conjunto com árvores de decisão adiciona flexibilidade e maior velocidade. [20] [39] [61]

A ideia principal é construir uma árvore de decisão parcial, em vez de uma totalmente explorada. Por sua vez, uma árvore de decisão parcial é uma árvore de decisão comum que contém ramificações para subárvores indefinidas. Ao se encontrar uma subárvore estável, que não possa ser mais simplificada, a construção de árvores cessa e uma única regra é lida. [20] [39] [61]

A figura 2.10, representa um exemplo, passo a passo, da forma como o algoritmo, PART, constrói uma árvore parcial. Durante as fases 1 a 3, a construção de árvores continua, recursivamente, de forma normal. De seguida, o nó 3, que apresenta uma entropia, isto é, uma incerteza entre variáveis aleatórias num conjunto de dados, mais baixa, em relação, aos nós 2 e 4, é o escolhido para efetuar a expansão. Os nós pretos representam folhas. Nas fases 2 e 3, o nó preto tem uma entropia mais baixa, mas como é uma folha não pode ser expandido. Assim, ocorre um retrocesso e, por sua vez, o nó 5 é o escolhido para expansão. Uma vez alcançada a fase 3, como o nó 5 tem todos os seus filhos expandidos em folhas, é acionada a poda. Assim, a substituição da subárvore para o nó 5 é considerada e aceite. Uma vez alcançada a fase 4, o nó 3 é considerado para substituição da subárvore e esta operação é novamente aceite. Consequentemente, o retrocesso continua e o nó 4 com entropia mais baixa do que 2, é expandido em duas folhas. Por fim, obtemos a fase 5 onde já não existe mais nenhuma substituição e o retrocesso termina, efetivamente, com a árvore parcial de 3 folhas. [20] [39] [61]

É importante referir que um nó só pode ser podado se todos os seus sucessores forem folhas. Isso, só pode acontecer se todas as suas subárvores tiverem sido exploradas e forem consideradas folhas ou forem podadas de volta às folhas. Contudo, se um conjunto de dados não contém ruídos e contém instâncias suficientes para evitar que o algoritmo faça qualquer poda, apenas um caminho da árvore de decisão completa deve ser explorado. Assim, o ganho, ao nível de desempenho, diminui à medida que ocorre mais poda. No entanto, apenas uma única regra pode ser extraída de uma árvore parcial. Por fim, ao construir uma árvore parcial, os valores ausentes são tratados, exatamente, da mesma forma que o algoritmo J48 (estudado anteriormente). [20] [39] [61]

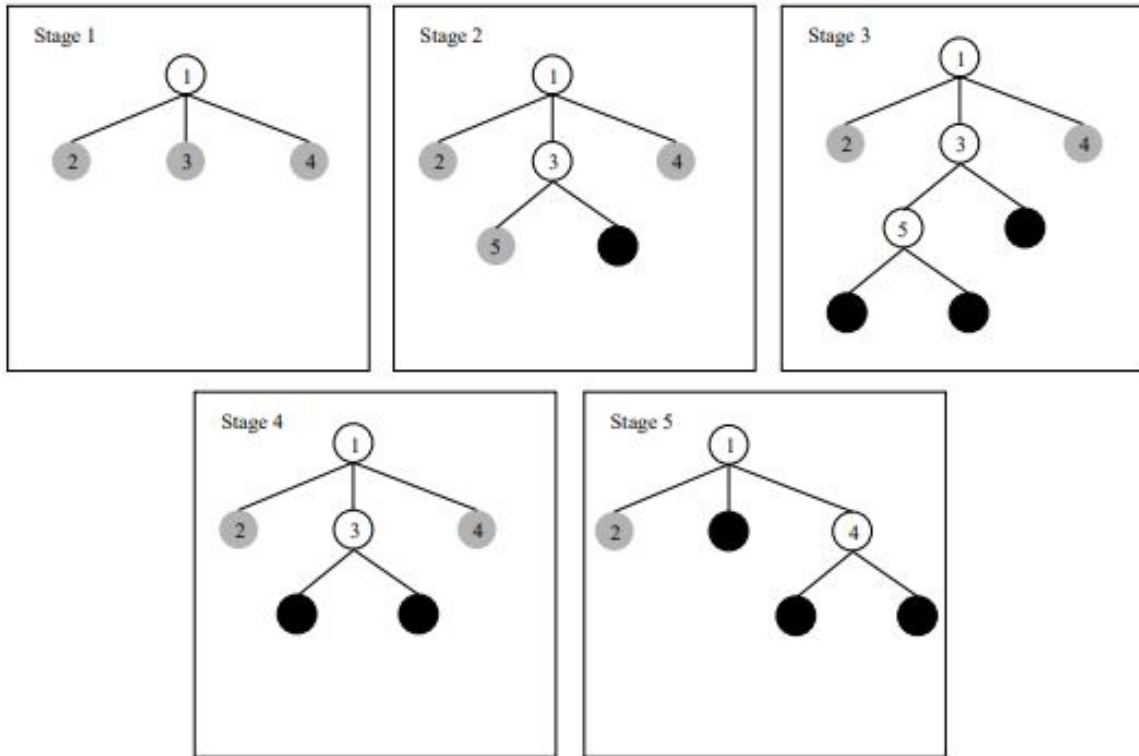


FIGURA 2.10. Exemplo: forma como o algoritmo, PART, constrói uma árvore parcial. [61]

### 2.3.3. J48

O algoritmo de classificação J48 corresponde ao algoritmo C4.5. Este algoritmo usa árvores de decisão para realizar a classificação. O algoritmo desenvolve um nó de decisão utilizando as estimativas esperadas da classe. A árvore de decisão J48 pode lidar com características particulares, estimativas de atributos perdidos ou ausentes dos dados e custos de atributos variáveis. Aqui, a precisão pode ser expandida por poda (um nó transforma-se em folha). [16] [28] [26]

O funcionamento deste algoritmo recai sobre as três etapas seguintes:

- **Etapa 1:** se as instâncias pertencerem a uma classe semelhante, a folha é rotulada com uma classe semelhante;
- **Etapa 2:** Para cada atributo, os dados, com um maior potencial, serão calculados e o ganho nos dados será obtido no teste do atributo;
- **Etapa 3:** Finalmente, o melhor atributo será escolhido dependendo do parâmetro de seleção atual. [16] [28] [26]

A construção de uma árvore de decisão com valor significativo é uma das etapas mais importantes para a geração de regras, através do algoritmo J48. Por outro lado, apesar deste algoritmo ser bem conhecido, existem três limitações inerentes ao mesmo. A primeira consiste na árvore com ramos vazios, ou seja, nós com valores zero que não contribuem para a criação da árvore ou nem ajudam a criar qualquer classe na fase de

classificação. Assim, a árvore torna-se mais larga e complicada. A segunda prende-se numa árvore com ramos insignificantes, isto é, ramos com atributos que não sejam significativos para a classificação, apenas diminuem a usabilidade das árvores de decisão que, por sua vez, leva ao aparecimento do problema de sobreajuste. Por último, a terceira limitação corresponde ao problema do sobreajuste e ocorre quando a exibição do algoritmo obtém informações com atributos excepcionais. Este tipo de problema causa muitas fragmentações na concepção da árvore de decisão. Normalmente, o algoritmo constrói árvores e faz crescer os ramos apenas o suficiente para classificar, na perfeição, os exemplos de aprendizagem. Por forma a solucionar este problema, existem dois tipos de ferramentas, entre elas, a entropia da teoria da informação e o coeficiente de correlação. [16] [28] [26]

O comportamento de processos aleatórios é um fator importante para implementar a codificação da teoria da informação. Consequentemente, a entropia é uma medida de incerteza moderada de coleta de informações quando não temos conhecimento do resultado de uma fonte de dados. Isso, também, indica a quantidade média de dados que obteremos do resultado de uma fonte de dados. A Equação 2.2 é usada para medir a teoria da informação da entropia. O  $X$  é um atributo, o  $p$  é cada elemento e o  $j$  é a posição de cada elemento de  $X$ . [16] [28] [26]

### 2.2. Teoria de informação da entropia ( $H(x)$ )

$$H(X) = \sum_{j=1}^k p_j \log_2 \frac{1}{p_j} = - \sum_{j=1}^k p_j \log_2 p_j \quad (2.2)$$

Em suma, quanto menor o valor de  $H(X)$ , menor a aleatoriedade do atributo  $X$ , ou seja, este atributo é mais significativo para a mineração de dados. O valor da entropia atinge o mínimo 0, quando todos os outros  $p_j$  são 0. Por outro lado, o valor atinge o máximo  $\log_2 k$ , quando todos os  $p_j$  são iguais a  $\frac{1}{k}$ . [16] [28] [26]

O coeficiente de correlação é uma das principais ferramentas estatísticas para examinar conjuntos de variáveis e descobrir a sua relação. Assim, o utilizador seria capaz de tomar decisões, com base nas informações disponíveis por coeficientes de correlação. O coeficiente de correlação de classificação usado é o de Kendall. A equação 2.3 representa a medição da média do coeficiente de correlação da classificação de Kendall. [16] [28] [26]

### 2.3. Coeficiente de correlação da classificação de Kendall ( $\tau$ )

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \quad (2.3)$$

O coeficiente de Kendall,  $\tau$ , é definido de seguinte forma:  $n_c \rightarrow$  Número de pares concordantes,  $n_d \rightarrow$  Número de pares discordantes e  $n \rightarrow$  Número total de pares. A faixa de valores referente a este coeficiente varia entre -1 e +1. [16] [28] [26]

As vantagens presentes neste algoritmo, em relação a outros sistemas de árvores de decisão são as seguintes: mitigar o sobreajuste, através do processo de poda numa única

passagem, trabalhar com dados discretos e contínuos e, por fim, este algoritmo lida muito bem com o problema de dados incompletos. [16] [28] [54]

#### 2.3.4. IBk

O algoritmo de classificação IBk corresponde ao algoritmo KNN que, por sua vez, é um dos algoritmos mais simples de classificação. [7] [5] Apesar da simplicidade, tem a capacidade de gerar resultados altamente competitivos. Um exemplo simples para perceber o funcionamento do algoritmo consiste em descobrir a classe da estrela azul, presente na figura 2.11. Como se pode verificar na figura 2.11, esta classe apenas pode corresponder ao círculo vermelho ou ao retângulo verde. O parâmetro K indica o número de vizinhos a usar para definir a classe da estrela azul e, neste caso, o K é igual a 3. Daí o círculo, presente na figura 2.11, incluir os 3 vizinhos mais próximos da estrela azul. Assim, podemos concluir que a classe da estrela azul pertence à classe do círculo vermelho (Figura 2.11). [33] [30] [55]

A escolha do parâmetro K (constante) é muito importante neste algoritmo. Contudo, para selecionar o K correto é necessário executar o algoritmo KNN várias vezes com diferentes valores de K. De seguida, define-se o K que apresenta um menor número de erros e que mantém a capacidade do algoritmo realizar previsões com precisão, na presença de novos dados. [33] [30] [27]

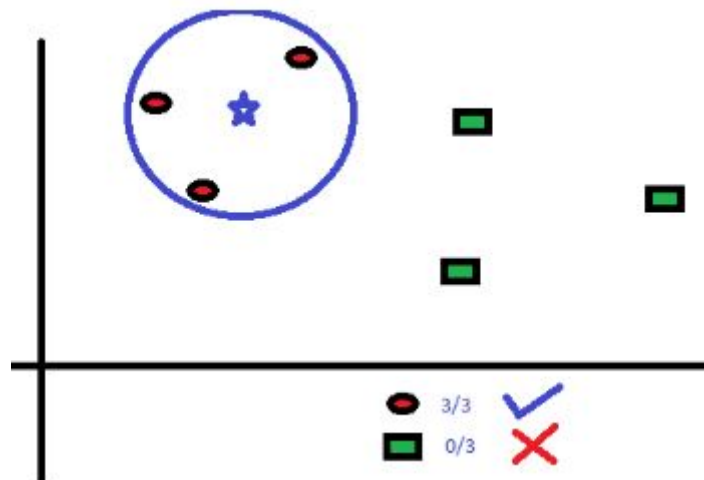


FIGURA 2.11. Funcionamento do algoritmo KNN. [55]

Existe uma maior instabilidade de previsões, à medida que se diminui o valor de K para 1. Consequentemente, existe uma maior estabilidade de previsões, à medida que se aumenta o valor de K, devido ao funcionamento do algoritmo. Este aumento é controlado pelo número de erros. No caso de classificação, isto é, de votação por maioria, o parâmetro K tem de ser ímpar para se ter um desempate. [33] [30] [27]

Este algoritmo apresenta inúmeras vantagens, entre elas, a facilidade de implementação, a versatilidade, o uso para fins de classificação, regressão e pesquisa e, por fim, a inexistência da necessidade de construir um modelo, ajustar vários parâmetros ou fazer

suposições adicionais. Por outro lado, também apresenta uma desvantagem que se prende com o número de exemplos e/ou preditores/variáveis independentes. Ou seja, quanto maior este número, mais lento se torna este algoritmo. [33] [30] [27]

## 2.4. Aplicações

Todas as aplicações, em estudo, presentes nesta dissertação não possuem qualquer tipo de mecanismo de caráter intrusivo.

### 2.4.1. Beddit

A aplicação Beddit apenas está disponível para IOS. Utiliza um monitor de sono Beddit (Figura 2.12), específico para monitorizar automaticamente o sono, de forma, a obter uma melhor qualidade de descanso. Este sensor externo tem um valor total de 149 euros.<sup>1</sup> Ao nível de estrutura, apresenta vários sensores finos, macios e flexíveis, tornando, assim, a sua presença na cama entre o colchão e o lençol impercetível. Estes sensores correspondem ao sensor de força piezoelétrico, de toque capacitivo, de humidade e de temperatura. Por outro lado, reúne e analisa todos os dados provenientes do sono, entre eles, a duração e eficiência, respiração, ritmo cardíaco, temperatura, movimento, ressonar, temperatura e humidade ambiente. O atuador interno usado, para acordar o utilizador, é o Som. E, a distância máxima entre o sensor e o utilizador corresponde a, aproximadamente, 30 cm (espessura de um colchão normal). [2]



FIGURA 2.12. Monitor de sono usado pela aplicação Beddit.<sup>2</sup>

De acordo com o modo de funcionamento, todo o processo envolvente nesta aplicação provém de uma metodologia conhecida como balistocardiografia (BCG). Isto é, cada vez que o coração bate, a aceleração do sangue gera um impulso mecânico que é medido através do sensor externo Beddit. [35]

Em suma, toda a transferência de dados proveniente do sensor externo (monitor de sono) é realizada, de forma inteligente, através do Bluetooth. Assim, estes resultados estão disponíveis para o utilizador, imediatamente, quando a pessoa acorda, sem a necessidade de qualquer interação prévia com o dispositivo. [2]

<sup>1</sup><https://www.apple.com/pt/shop/product/MVF02ZM/A/monitor-de-sono-beddit>

<sup>2</sup><https://www.apple.com/pt/shop/product/MVF02ZM/A/monitor-de-sono-beddit>



### 2.4.2. DREEM 2

A aplicação DREEM 2 está disponível para IOS e Android. Utiliza uma banda inteligente para colocar na cabeça (Figura 2.13) específica para efetuar uma monitorização automática referente ao sono, de forma, a obter uma melhor qualidade, ao nível, de descanso. Apresenta um valor total de, aproximadamente, 428,10 euros.<sup>3</sup> Esta banda, ao nível estrutural, apresenta vários sensores, entre eles, 6 sensores EEG, um acelerómetro, um oxímetro de pulso e um sonómetro. Por outro lado, quando a pessoa coloca esta banda, dá-se início a toda a monitorização, onde reúne e analisa todos os dados provenientes do sono, como a atividade do cérebro (EEG), o movimento, a frequência cardíaca, a taxa de respiração e as fases do sono. É um dispositivo leve e autónomo para não incomodar o utilizador durante o descanso. Contudo, sem usar o Bluetooth e Wifi, é possível analisar todos os dados em tempo real e, mais importante, atuar em momentos precisos para otimizar o sono do utilizador. Em suma, toda a transferência de dados proveniente do sensor externo é realizada, de forma automática, através do Bluetooth ou Wifi para a cloud. Assim, estes resultados estão disponíveis para o utilizador na aplicação, imediatamente, quando a pessoa acorda, sem a necessidade de qualquer interação prévia com o dispositivo. O atuador interno usado, para acordar o utilizador, é o som. [18][19]



FIGURA 2.13. Monitor de sono usado pela aplicação Dreem 2.<sup>4</sup>

### 2.4.3. Emfit

A aplicação Emfit QS+ACTIVE está disponível apenas para aplicação Web. Usa um monitor de sono com variabilidade da frequência cardíaca Emfit QS+ ACTIVE (Figura 2.14) específico para monitorizar automaticamente o sono, de forma a obter uma melhor qualidade de descanso. Apresenta um valor total de 299.98 euros.<sup>5</sup> Este monitor é colocado entre a cama e o colchão, tornando, assim, a sua presença impercetível. Por outro lado,

<sup>3</sup><https://shop.dreem.com/north-america/en/>

<sup>4</sup><https://shop.dreem.com/north-america/en/>

<sup>5</sup><https://shop-eu.emfit.com/products/emfit-qs>

ao ligar à corrente este monitor inicia automaticamente o envio de dados. Em tempo real, é possível verificar todos os dados no aplicativo web, onde reúne e analisa todos os dados provenientes do sono. Entre eles, a frequência cardíaca, a variabilidade da frequência cardíaca, a taxa de respiração, o movimento e as diferentes fases do sono. O atuador interno usado, para acordar o utilizador, é o Som. E, a distância máxima entre o sensor e o utilizador corresponde a, aproximadamente, 30 cm (espessura de um colchão normal). [36] [38]

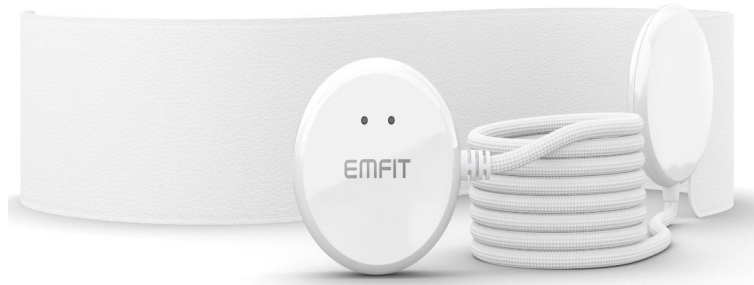


FIGURA 2.14. Monitor de sono usado pela aplicação Emfit QS.<sup>5</sup>

De acordo com o modo de funcionamento, toda a tecnologia envolvente nesta aplicação provém de duas métricas conhecidas como a balistocardiografia (BCG) e a variabilidade da frequência cardíaca (HRV). Esta tecnologia usa sensores de ferroelretros proprietários da Emfit. Este tipo de sensores são compostos por uma folha quase-piezoelétrica, sensores de banda e cabo. Os sensores Emfit convertem a força mecânica em energia elétrica proporcional. Assim, este sensor baseia-se numa carga elétrica permanente dentro da estrutura celular do núcleo do sensor. O próprio material eletroativo é tendencioso e não usa energia. É importante referir que o EMFIT QS+ACTIVE é o único monitor de saúde gratuito que pode fornecer uma análise para recuperação e prontidão, com base no HRV. Daí, ser, maioritariamente, usado pelos atletas profissionais. Em suma, toda a transferência de dados proveniente do sensor externo é realizada, de forma inteligente, através de Wifi ou dados móveis. Assim, estes resultados estão disponíveis para o utilizador, imediatamente, quando acorda, sem a necessidade de qualquer interação prévia com o dispositivo. Dispositivos com Wifi devem ser colocados longe da cama para evitar perturbações durante o sono. Todos os dados ficam guardados e seguros numa Cloud, o que torna o acesso mais fácil e rápido. [36] [38] [37]

#### 2.4.4. Pillow

A aplicação Pillow está disponível para IOS. E está incluída nas melhores aplicações móveis da Nova Saúde e Fitness na App Store. Tem o objetivo de melhorar a qualidade do sono efectuando a monitorização do sono com o apoio de um sensor acelerométrico embebido no dispositivo móvel. O algoritmo avançado da App Pillow monitoriza os movimentos e a frequência cardíaca para acordar o utilizador na fase do sono mais leve. É necessário colocar o dispositivo a menos de 1 metro da pessoa, por forma, a registar

corretamente os ciclos do sono. Conseqüentemente, o modo de funcionamento é simples e interativo, baseando-se num método não invasivo. O armazenamento de dados e áudio é realizado através da iCloud. Por fim, o atuador interno usado, para acordar o utilizador, é o som. [50][49]

#### 2.4.5. ZEEQ

A aplicação ZEEQ Smart Pillow está disponível para IOS e Android. Utiliza uma almofada inteligente ZEEQ (Figura 2.15) específica para monitorizar automaticamente o sono para obter uma melhor qualidade, ao nível, de descanso. Apresenta um valor total de, aproximadamente, 51,71 euros.<sup>6</sup> Este monitor, ao nível de estrutura, apresenta vários sensores e atuadores inseridos no núcleo da almofada, entre eles, um giroscópio de 3 eixos que acompanha todo o movimento corporal, um microfone que monitoriza o ressonar, oito altifalantes sem fio de alta precisão, uma tecnologia de ondas áudio para um sono repousante e dois motores de vibração ajustáveis que reagem ao ressonar. Por outro lado, quando a pessoa se deita este monitor inicia toda a monitorização, onde reúne e analisa todos os dados provenientes do sono, como a duração, o movimento e o ressonar. Em suma, toda a transferência de dados proveniente do sensor externo é realizada, de forma automática, através do Bluetooth. Assim, estes resultados estão disponíveis para o utilizador, imediatamente, quando a pessoa acorda, sem a necessidade de qualquer

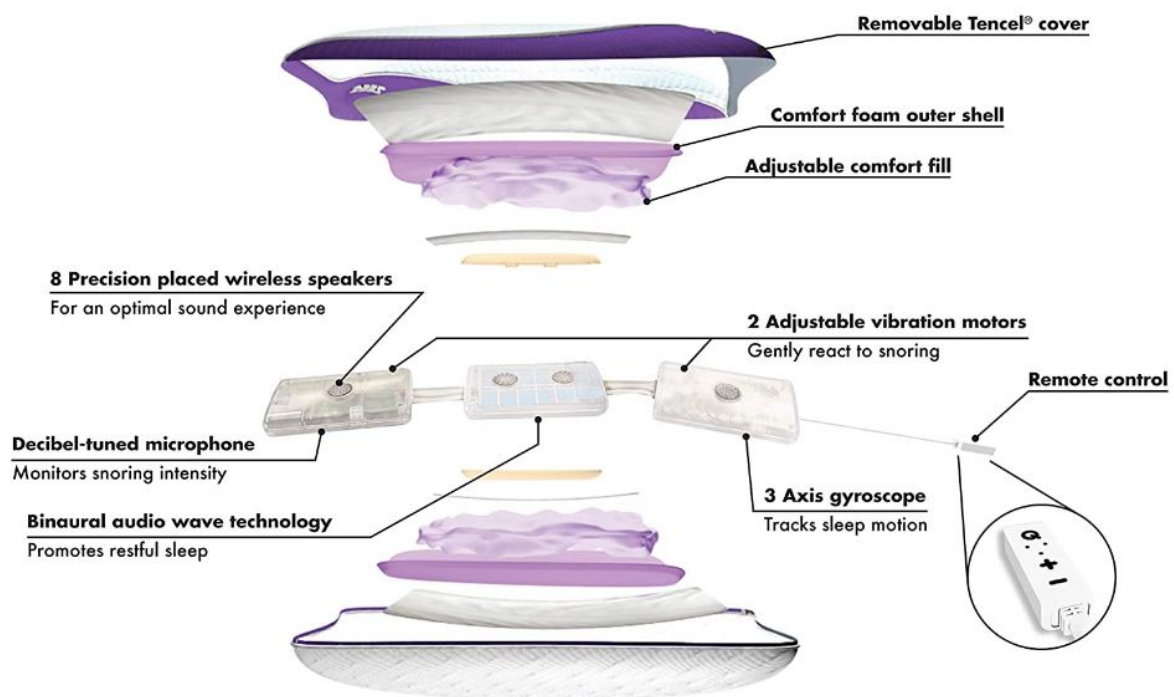


FIGURA 2.15. Monitor de sono usado pela aplicação ZEEQ Smart Pillow.<sup>7</sup>

<sup>6</sup>[https://www.amazon.com/dp/B06XG7G5SC/ref=as\\_li\\_ss\\_tl?&linkCode=s11&tag=remfit-20&linkId=83cb69387e059fcb9ff51d84adb741aa&language=en\\_US](https://www.amazon.com/dp/B06XG7G5SC/ref=as_li_ss_tl?&linkCode=s11&tag=remfit-20&linkId=83cb69387e059fcb9ff51d84adb741aa&language=en_US)

<sup>7</sup>[https://www.amazon.com/dp/B06XG7G5SC/ref=as\\_li\\_ss\\_tl?&linkCode=s11&tag=remfit-20&linkId=83cb69387e059fcb9ff51d84adb741aa&language=en\\_US](https://www.amazon.com/dp/B06XG7G5SC/ref=as_li_ss_tl?&linkCode=s11&tag=remfit-20&linkId=83cb69387e059fcb9ff51d84adb741aa&language=en_US)

interação prévia com o dispositivo. O atuador interno usado, para acordar o utilizador, é o Som e a vibração. E, por último, esta almofada acorda o utilizador, em estudo, sem perturbar a pessoa do lado. [51]

#### 2.4.6. Sleep As Android

A aplicação Sleep as Android está disponível para Android. Atualmente, por forma, a reconhecer todas as fases do sono podem ser utilizados 3 tipos de sensores, entre eles, o acelerómetro (inerente no dispositivo móvel), o sensor ultrassónico (SONAR) e o infravermelho. Com estes tipos de sensores, é possível usar um método denominado actigrafia do sono, que regista todo o movimento corporal. Não é tão preciso como o PSG, mas tem como vantagem uma configuração simples e é perfeito para o uso diário em casa, fornecendo resultados comparáveis. Com estes resultados, é possível acordar o utilizador numa hora ideal. E, assim, verificar se apresenta uma boa qualidade de sono, através da análise de 6 dimensões. Entre elas, a duração do sono, a irregularidade do sono, a percentagem do sono profundo, a eficiência do sono, a duração do ressonar e a avaliação subjetiva do sono. Por sua vez, existe a necessidade de colocar o telemóvel a menos de 1 metro de distância do corpo do utilizador. Consequentemente, o armazenamento de dados e áudio é realizado através da GoogleCloud. Por fim, a aplicação obriga o utilizador a levantar-se da cama com o apoio do captcha, onde existem 4 tipos diferentes, entre eles, o 'Morning Math', o 'Shake your phone', o 'Find the Sheep' e o mais eficiente, 'Scan Quick Response (QR) Code'. Além do captcha, existem mais 2 tipos de atuadores, entre eles, o som e a luz inteligente. [52][53]

#### 2.4.7. Comparação entre Aplicações

A tabela 2.2, compara todos os atuadores presentes em todas as aplicações. A forma mais comum de atuação é através do som proveniente do dispositivo móvel ou do sensor externo (ZEEQ), de forma, a acordar a pessoa. A aplicação Sleep as Android apresenta mais tipos de atuadores. Entre eles, o captcha e a luz inteligente. O captcha tem como objetivo, obrigar o utilizador a levantar-se da cama para conseguir desligar o alarme.

Analisando as tabelas 2.3 e 2.4, pode-se concluir que para as aplicações que apresentam sensores internos, o custo da mesma é gratuito (Pillow e Sleep As Android). Enquanto que, para as aplicações que utilizam sensores externos, já têm um custo associado. Ou seja, as aplicações Beddit, Emfit QS, ZEEQ Smart Pillow, DREEM 2 e "GoToBed" (proposta presente no capítulo 3) custam 149, 299.98, 51.71, 428.10 e 168.45 euros, respetivamente. Assim, o ZEEQ é a App mais barata e o DREEM 2 a mais cara. Isto faz todo o sentido, tendo em conta, as metodologias associadas a cada uma delas. A metodologia BCG está presente nas aplicações Beddit, Emfit QS e "GoToBed". Por sua vez, as metodologias SONAR e EEG correspondem à App Sleep as Android e ao DREEM 2, respetivamente. Também, existem diferenças, inerentes a cada aplicação, entre a forma como se transferem e armazenam os vários tipos de dados e áudio. Por fim, como se pode verificar na tabela 2.3, as aplicações representam sistemas operativos distintos.

TABELA 2.2. Comparação entre atuadores presentes em todas as Aplicações.

Aplicações	Tipo de Atuador	Atuadores Utilizados
Beddit	Interno	Som
Sleep as Android	Interno Externo	Luz inteligente, som e captcha
Pillow	Interno	Som
Emfit QS	Interno	Som
ZEEQ Smart Pillow by Rem-Fit	Externo	Som Vibração
DREEM 2	Interno	Som
<b>Esta Proposta "GoToBed"</b>	Interno	Som

TABELA 2.3. Comparação entre todas as aplicações referenciadas e analisadas ao longo desta dissertação.

Aplicações	Sistema Operativo	Custo [euros]	Metodologias	Transferência de dados e áudio	Armazenamento de dados e áudio
Beddit	IOS	149,00	BCG	Bluetooth	-
Sleep as Android	Android	0	SONAR	-	GoogleCloud
Pillow	IOS	0	-	-	iCloud
Emfit QS	Web	299,98	BCG	Wifi Dados móveis	Cloud
ZEEQ Smart Pillow by Rem-Fit	Android IOS	51,71	-	Bluetooth	-
DREEM 2	Android IOS	428,10	EEG	Bluetooth Wifi	GoogleCloud iCloud
<b>Esta Proposta "GoToBed"</b>	Android	168,45	BCG	Wifi	Local/Cloud

A tabela 2.4, analisa a distância máxima possível entre o sensor e o utilizador. As aplicações com sensores internos apresentam uma maior distância igual a 100 cm. Por outro lado, as aplicações, com metodologia BCG, que funcionam em conjunto com sensores externos incidem sobre uma distância máxima de 30 (Beddit e Emfit) e 60 ("GoToBed")

cm. Contudo, nas aplicações ZEEQ e DREEM 2 não existe uma distância máxima, visto que, se trata de uma almofada e de uma banda para a cabeça, respetivamente.

Ao nível de sensores utilizados, na tabela 2.4, tanto o Beddit como o Emfit QS ambos

TABELA 2.4. Comparação entre sensores presentes em todas as Aplicações.

Aplicações	Tipo de Sensor	Sensores Utilizados	Distância máxima/mínima entre o sensor e o utilizador [cm]
Beddit	Externo	Sensor de força piezoelétrico, de toque capacitivo, de humidade e de temperatura	30 (máxima)
Sleep as Android	Interno (Dispositivo)	Sensor ultrassónico (SONAR), sensor de movimento/posição (giroscópio e acelerómetro) e sensor infravermelho passivo	100 (máxima)
Pillow	Interno (Dispositivo)	Sensor de movimento/posição (giroscópio e acelerómetro)	100 (máxima)
Emfit QS	Externo	Sensores de ferroelétricos (Folha quase-piezoelétrica, sensores de banda e cabo)	30 (máxima)
ZEEQ Smart Pillow by Rem-Fit	Externo	Sensor de movimento (giroscópio de 3 eixos)	-
DREEM 2	Externo	6 sensores EEG (Eletroencefalograma), acelerómetro, oxímetro de pulso e sonómetro	-
<b>Esta Proposta "GoToBed"</b>	Externo	Sensor SCA10H (acelerómetro)	20 (mínima) 60 (máxima)

apresentam um sensor de estrutura fina e macia. Consequentemente, as aplicações Sleep as Android e Pillow recaem sobre os mesmos princípios, isto é, sobre sensores de movimento/posição (giroscópio e acelerómetro). No entanto, o que as distingue é um sensor ultrassónico (metodologia SONAR). Por outro lado, a aplicação DREEM 2 apresenta seis sensores de EEG, daí ser mais cara a sua aquisição. E, também, um acelerómetro, um

oxímetro de pulso e um sonómetro. Um oxímetro de pulso foi desenvolvido para medir a taxa de batimento cardíaco e a saturação de oxigénio do sangue. [3] Por fim, a proposta ("GoToBed") apresenta um sensor SCA10H (acelerómetro).

### Desenho e Implementação da Proposta

#### 3.1. Proposta de monitorização e atuação

Para atingir os objetivos a que se propõe este trabalho, é proposta uma nova aplicação, que iremos designar por "GoToBed" que apresenta uma sequência estrutural, como ilustrada na figura 3.1. Esta sequência estrutural representa duas fases distintas, onde a primeira consiste na construção do modelo final, enquanto que, a segunda apresenta a utilização do modelo resultante.

Numa primeira fase, para monitorizar o sono, usa-se um sensor externo baseado no BCG, porque a monitorização com outros sensores podem ser mais invasivas, como o PSG, ou menos precisas. De seguida, com os dados de saída do sensor é aplicado um processo de filtragem, onde é necessário eliminar o ruído presente nos sinais. De forma, a preservar e mostrar, com um maior detalhe, a informação relevante presente nesses mesmos sinais. Neste caso, optou-se por um filtro passa-baixo, devido a uma maior sensibilidade às frequências elevadas para uma constante de filtragem superior. Consequentemente, na aprendizagem do modelo, são injetados os dados de saída do sensor BCG e do PSG para se conseguir construir o modelo final. Como o PSG apresenta uma solução mais precisa e fidedigna, definiu-se esta abordagem, visto que, os resultados são obtidos por técnicos especialistas. E, apenas é usado o valor de referência do estado do sono. Assim, juntam-se estes mesmos dados para se efetuar a aprendizagem para cada algoritmo de classificação, em estudo.

Numa segunda fase, com o modelo definido ocorre a predição das várias fases do sono. Esta predição utiliza os seguintes algoritmos de classificação Weka: RandomForest, J48, REPTree, PART, IBk, SMO, Logistic, ZeroR, NaiveBayes. De seguida, constrói-se as matrizes de confusão, por forma, a se concluir qual o melhor algoritmo presente neste estudo, tanto a nível de desempenho como de aprendizagem. Assim, já se encontram todas as condições para definir as fases do sono para qualquer pessoa, durante uma noite. E, por fim, o atuador definido pode ser o som proveniente do dispositivo móvel, a luz inteligente, a vibração ou outros tipos de atuadores externos ou internos.

Em suma, nesta proposta, pode-se optar por duas vias. A primeira consiste num modelo pré-configurado e devidamente testado com o apoio da ferramenta Weka Explorer. O processo de aprendizagem do modelo é realizado com dados de pessoas diferentes, onde ocorre a junção dos dados de PSG e BCG, em simultâneo. Assim, no desenvolvimento da aplicação, já apresenta o modelo previamente definido usado para efetuar a predição, juntamente com o melhor algoritmo de classificação. A segunda dá a oportunidade ao utilizador de treinar o PSG e o BCG, em simultâneo. Isto é, de poder treinar o modelo



apenas com dados provenientes do utilizador. E, assim, poder efetuar a aprendizagem e, por sua vez, a construção do modelo, através da aplicação a desenvolver, para qualquer algoritmo de classificação Weka, em estudo. Ou seja, em vez de se efetuar a aprendizagem do modelo externamente referente à aplicação, realiza-se todo o processo intrinsecamente na aplicação, usando a biblioteca Weka disponível para JAVA.

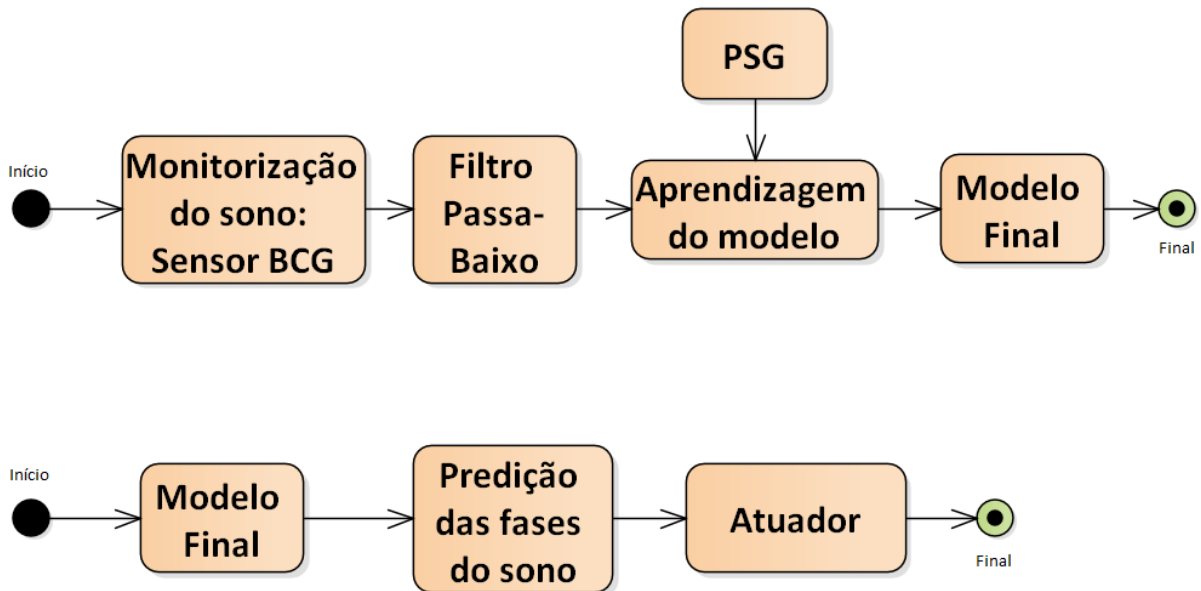


FIGURA 3.1. Sequência estrutural proposta para implementação da construção e utilização do modelo.

Na figura 3.2, está representada a proposta do diagrama de sequência funcional da aplicação proposta. Numa primeira instância, conecta-se o sensor à aplicação via wifi. Contudo, se o sensor não tiver sido, previamente, calibrado ou configurado, é necessário realizar a respetiva calibração e configuração do mesmo. De seguida, prossegue-se com a fase da definição da atuação pretendida para esta aplicação como, por exemplo, um alarme para um acordar mais suave. E, conseqüentemente, a aplicação começa a receber, a cada segundo, os dados de saída do sensor. Todos estes dados estão sujeitos a um processo de filtragem. Por sua vez, ocorre a criação de um ficheiro Attribute-Relation File Format (ARFF) com todos os dados, previamente, filtrados. Com este ficheiro, é possível efetuar a predição das fases do sono através do algoritmo de classificação. Posteriormente, é criado um gráfico com todas as fases do sono. Neste trabalho, com esta informação, se o utilizador tiver no sono profundo, 15 minutos antes do alarme tocar, é despoletada uma música relaxante. Este tipo de música vai durar apenas até à hora prevista de tocar o alarme ou até o utilizador estar num estado diferente do que o profundo. Senão, dá-se início a uma música normal, apenas na hora definida no alarme. Por fim, se a pessoa, em estudo, acordou no estado do sono profundo, a noite de descanso é classificada como uma má recuperação. Enquanto que, se não acordou no estado do sono profundo, a noite de descanso é classificada como uma boa recuperação.

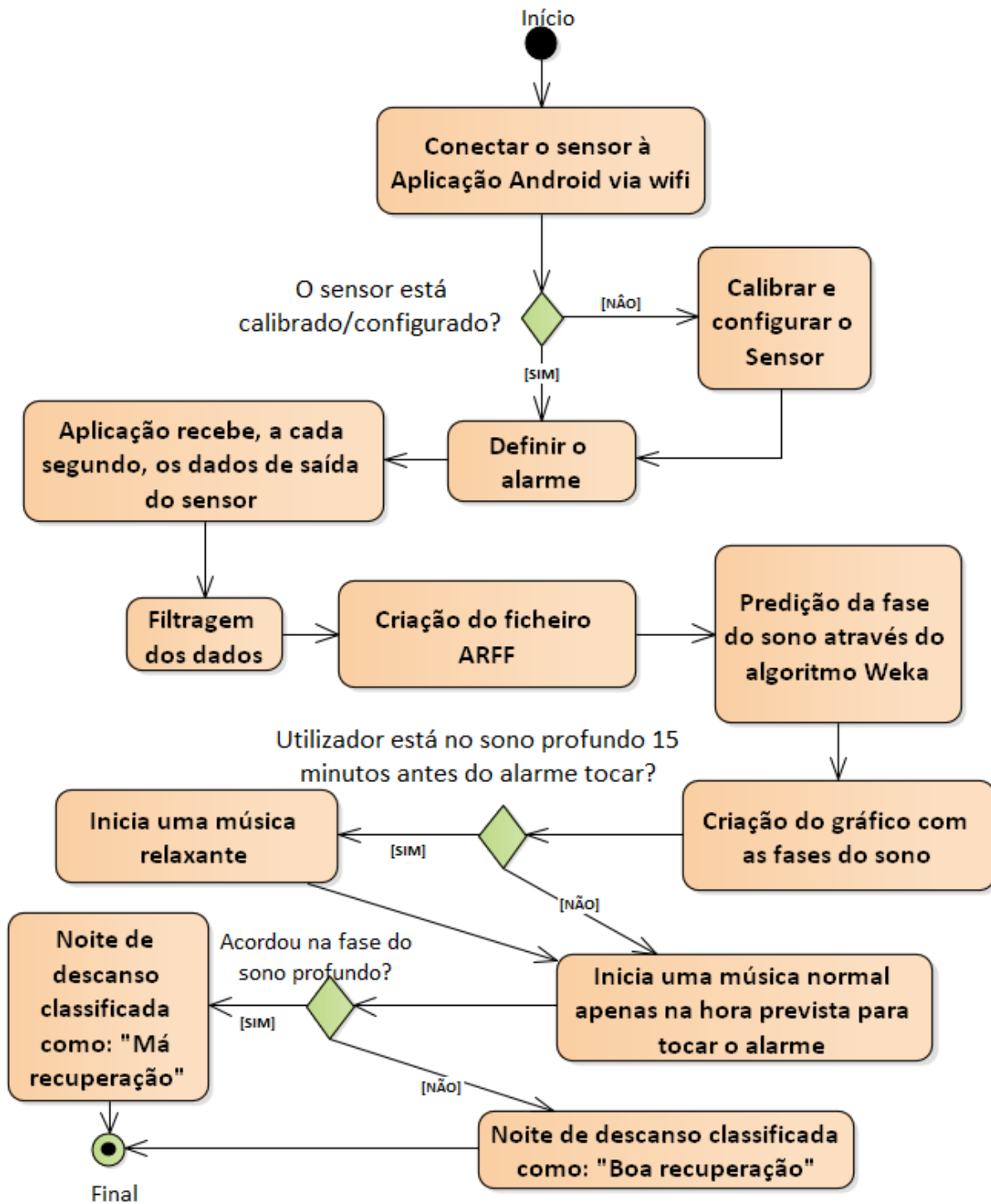


FIGURA 3.2. Proposta do diagrama de seqüência funcional da aplicação, em estudo.

Existem diferentes consequências para os diferentes estados. Isto é, para o estado profundo do sono, a atuação ocorre ao nível de um som mais relaxante, enquanto que, para todos os outros estados do sono (leve, REM e acordado), o som resultante é mais intenso. Em suma, este tipo de atuação tem como objetivo um acordar mais calmo na fase do sono profundo e um acordar mais repentino nas outras fases.

### 3.2. Implementação da Proposta

Na implementação da proposta, estão indicadas todas as opções definidas para o desenvolvimento da aplicação com o apoio de um sensor externo SCA11H (BCG). A aplicação "GoToBed"<sup>1</sup> desenvolvida, em Android, tem como ferramenta de desenvolvimento, o Android Studio. A versão da App de demonstração do sensor SCA11H (Android), foi usada para testar a ligação ao sensor. [8]

O sensor a definir tem de apresentar, obrigatoriamente, os seguintes requisitos: um baixo preço, acesso livre a dados e uma quantidade de métricas suficientes e relevantes para a concretização dos objetivos. Assim, a escolha do sensor SCA11H consistiu no uso da metodologia BCG, no tipo de parâmetros de saída do sensor e na facilidade imposta pela instalação (calibração, configuração e posição). O preço do sensor é de 168,45 euros.<sup>2</sup> [12] [13] [9]

A figura 3.1, representa os principais componentes envolvidos durante o desenvolvimento da aplicação GoToBed. O principal diz respeito à plataforma Android Studio que, por sua vez, oferece as ferramentas mais rápidas para criar aplicativos em todos os tipos de dispositivos Android. O código implementado no Eclipse pode ser migrado diretamente para o Android Studio. A linguagem utilizada no desenvolvimento da aplicação é JAVA. O Apache Cordova é usado na instalação da aplicação de demonstração do sensor SCA11H, visto que, inclui as linguagens de Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) e JavaScript. O conceito de Socket, inerente à linguagem JAVA, serve para efetuar a comunicação, neste caso, entre o sensor externo e a aplicação via Wifi. Por outro lado, o conceito de Thread, nesta aplicação, corresponde ao registo, segundo a segundo, de todos os parâmetros de saída provenientes do sensor SCA11H. Por fim, a inclusão da API do Weka surgiu com a questão referente à predição das fases do sono. [17] [23] [29] [14] [8]

TABELA 3.1. Principais tecnologias usadas na implementação da aplicação GoToBed.

Android Studio
Eclipse
JAVA
Apache Cordova
HTML
CSS
JavaScript
Thread
API Weka

<sup>1</sup><https://github.com/BernardoSilva-61191/Thesis-App-GoToBed.git>

<sup>2</sup><https://pt.mouser.com/ProductDetail/Murata-Electronics/SCA11H-A01-036?qs=31X5AGUhm2%2FiHkWXyX7oTw%3D%3D>

A figura 3.3 consiste na implementação da proposta referente à aplicação GoToBed. Consequentemente, a primeira fase recai sobre a recolha dos dados através do sensor

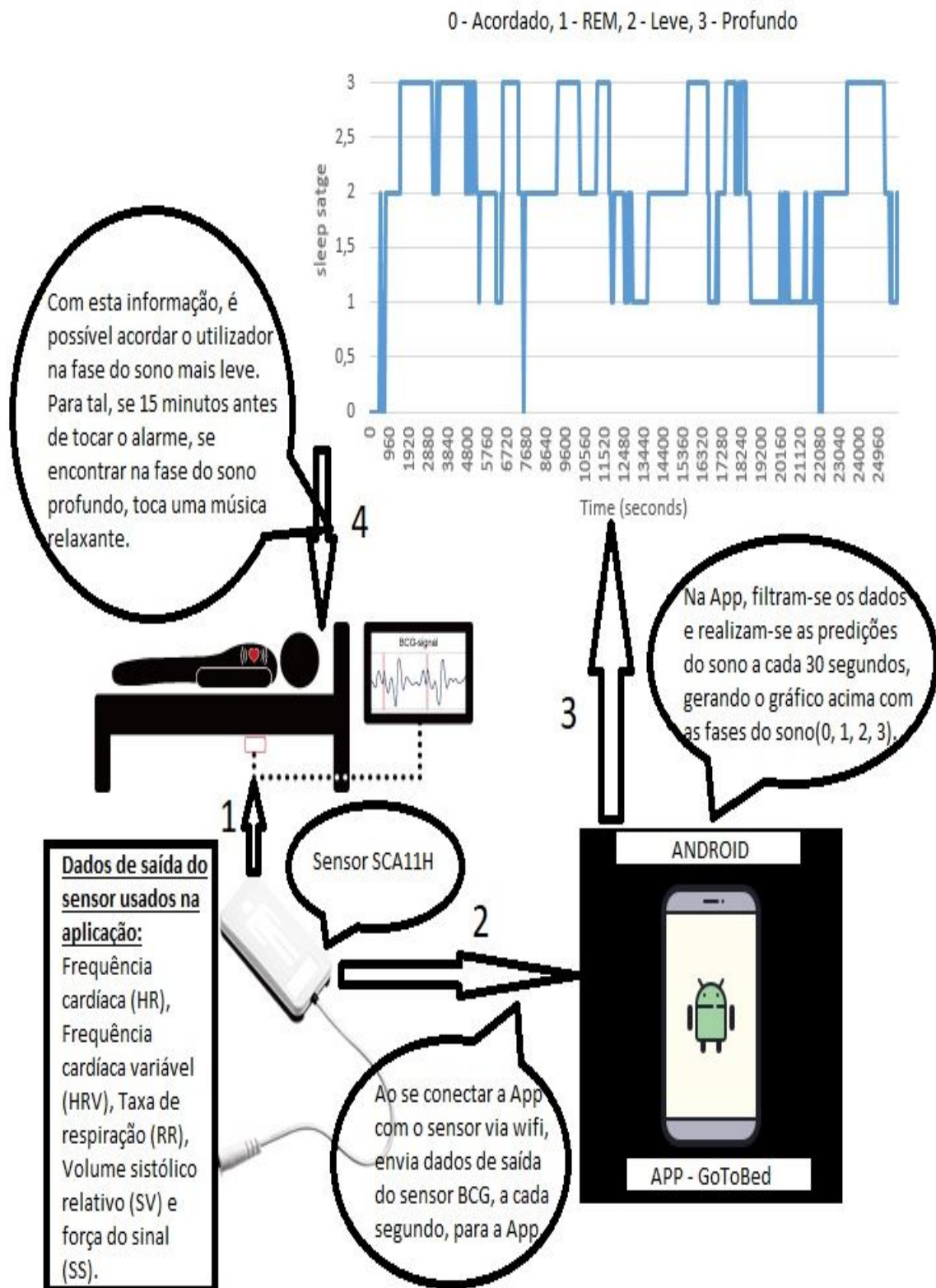


FIGURA 3.3. Implementação da proposta referente à aplicação GoToBed.

SCA11H. Ou seja, usando a metodologia BCG inserida no sensor obtemos todo este tipo de informação como, por exemplo, o HR, o HRV, o RR, o SV e a Signal Strength (SS), que serão explicados mais à frente. De seguida, a segunda fase prende-se com a conexão da aplicação com o sensor via Wifi, por forma, a obter todos os dados de saída do sensor BCG, a cada segundo. Por sua vez, na terceira fase, a aplicação filtra todos estes dados e realiza as predições do sono, de 30 em 30 segundos. Por último, como quarta etapa, está bem presente o objetivo principal da aplicação. Esta aplicação atua, de forma a acordar o utilizador na fase do sono mais leve. Para tal, é realizada uma análise, 15 minutos antes da hora prevista de tocar o alarme. Se esta análise obter como resultado a fase do sono profundo, dá-se início a uma música relaxante. Senão, o alarme toca a música normal na hora definida previamente.

### 3.2.1. Sensor

O nó do sensor usado, BCG SCA11H, é constituído pelo módulo sensor BCG SCA10H e pelo adaptador de rede sem fios (Figura 3.4). [9]

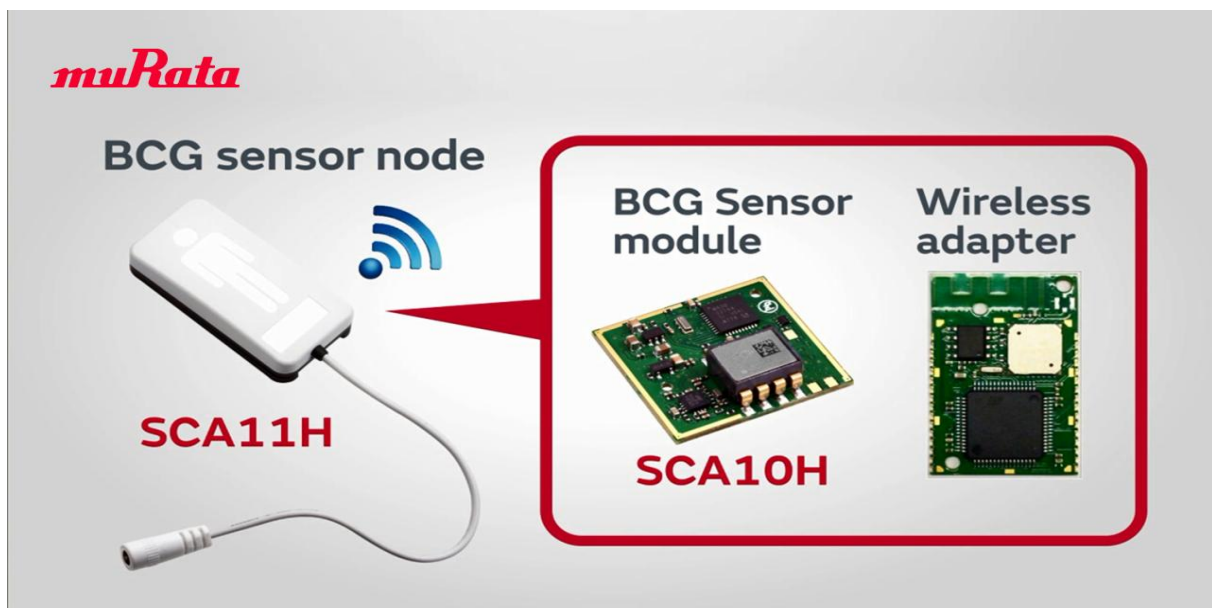


FIGURA 3.4. Estrutura do sensor SCA11H. [9]

O sensor permite dois modos de comunicação: comunicação via cloud e comunicação local. A comunicação de dados foi configurada no modo de comunicação local, porque assim é possível obter a informação de forma direta. Neste modo, o nó do sensor SCA11H funciona como servidor Transmission Control Protocol (TCP)/Internet Protocol (IP) e grava os dados de saída no formato ASCII na porta 8080 do endereço IP definido. A tabela 3.2 representa todos os parâmetros de saída do sensor SCA11H e a sua respetiva descrição. Estes parâmetros são reportados na frequência de amostragem de 1 Hz. [13]

TABELA 3.2. Parâmetros de saída do sensor BCG SCA11H. [13]

Variável	Unidade	Descrição
Time	-	Timestamp do sistema
HR	1/min	frequência cardíaca [beat per minute (bpm)]
RR	1/min	Taxa de respiração [bpm - breath per minute (respiração por minuto)]
SV	unidade arbitrária	Volume sistólico relativo
HRV	ms	Variabilidade simplificada da frequência cardíaca
SS	unidade arbitrária	Indicação de força do sinal medido
Status	-	0 = sinal baixo 1 = sinal ok 2 = sinal alto 3 = (perto da) sobrecarga 4 = (perto da) frequência cardíaca máxima
Beat-to-Beat time (B2B)	ms	Tempo de batimento a batimento
Beat-to-Beat time 1 (B2B')	ms	Tempo de batimento a batimento '
Beat-to-Beat time 2 (B2B'')	ms	Tempo de batimento a batimento ''

A descoberta do nó do sensor SCA11H baseia-se no protocolo Simple Service Discovery Protocol (SSDP). A figura 3.5, explica o funcionamento do protocolo em estudo. Em primeiro lugar, a aplicação envia o pacote "DISCOVER" para Broadcast:2000 (255.255.255.255:2000) usando o formato de pacote User Datagram Protocol (UDP). De seguida, o Body Sensor Network (BSN), que é representado pelo Sensor SCA11H, recebe o pedido "DISCOVER" e envia o pacote "NOTIFICATION" para a aplicação. Por fim, a aplicação extrai as informações de endereço IP da resposta de "NOTIFICATION" recebida. [13]

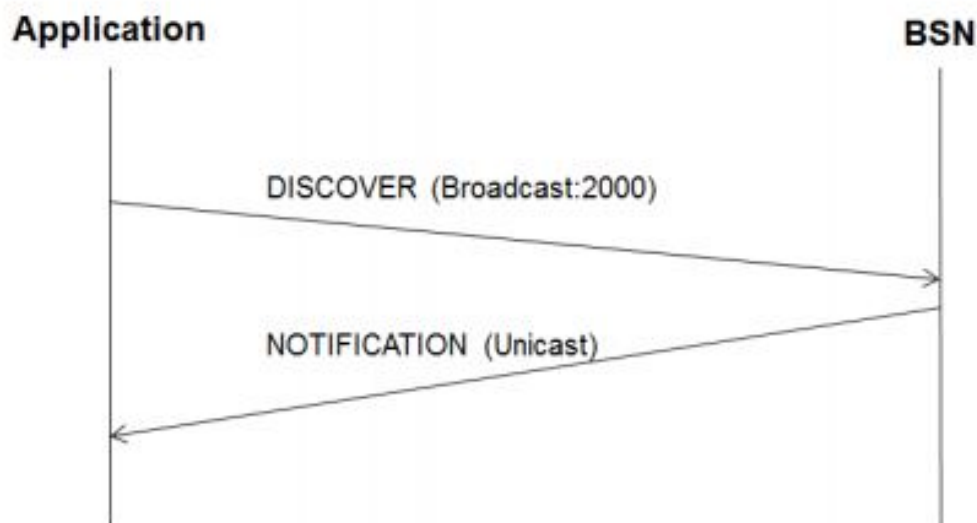


FIGURA 3.5. Protocolo SSDP. [13]

O posicionamento do SCA11H é muito importante. Consequentemente, a orientação do sensor SCA11H deve ser a mesma da impressa na parte superior do sensor (símbolo de uma pessoa). Este sensor tem de ser posicionado num local ideal para medir a vibração da cama causada pela circulação sanguínea. Se a vibração da cama estiver abafada atenuadamente, a localização ou estrutura da mesma devem ser reconsideradas. De acordo com a figura 3.6, existem várias opções de posição para o sensor SCA11H. Entre elas, sob o colchão (mas não por baixo da pessoa), na estrutura da cama ou ao lado cama. Por sua vez, a distância recomendada entre a pessoa e o sensor é de, no mínimo, 20 cm. De seguida, após a definição da posição do sensor é realizada a fase de calibração. Nesta fase, recomenda-se que a calibração seja realizada separadamente para cada pessoa na cama. Em primeiro lugar, inicia-se a calibração, durante 1 minuto, com a cama vazia e sem ruídos. E, por último, inicia-se a calibração, durante 1 minuto, com a cama ocupada e sem ruídos. A fase de calibração só está concluída, se obtiver sucesso com a cama ocupada e vazia, senão tem de voltar a repetir o processo. É importante referir que no processo de configuração, é necessário configurar a mesma rede local (Wifi) para o sensor se conectar e poder partilhar toda a informação disponível, presente na tabela 3.2. [12]

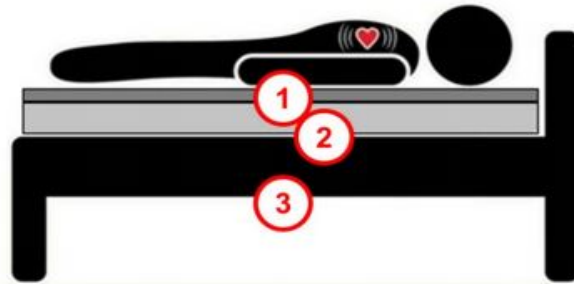


FIGURA 3.6. Possíveis opções de posicionamento para o sensor SCA11H. [12]

O módulo sensor BCG SCA10H está incluído na estruturação do sensor SCA11H. A figura 3.7, apresenta o diagrama de blocos do sensor SCA10H. O módulo mais importante é o que está assinalado a vermelho, o acelerómetro (SCA61T 3D Micro-electromechanical Systems (MEMS)). Este acelerómetro inclui um ruído ultrabaixo e uma largura de banda

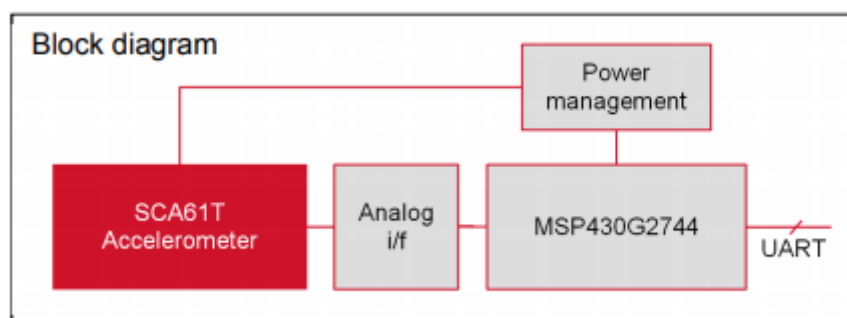


FIGURA 3.7. Diagrama de blocos referente ao sensor SCA10H. [11]



de ruído estreito. Conseqüentemente, os outros módulos correspondem ao interface Universal Asynchronous Receiver/Transmitter (UART), ao módulo MSP430G2744 (microcontrolador) e a um condicionamento de sinal analógico. O processamento de sinal de frequência respiratória e de coração está integrado com uma taxa de saída de 1 Hz. [11]

### 3.2.2. Filtragem dos dados

A necessidade de filtragem surge sobre o efeito da eliminação do ruído e do movimento da pessoa nos sinais provenientes dos dados de saída do sensor SCA11H BCG. Assim, consegue-se extrair informações relevantes e remover informações indesejadas. O filtro tanto pode apresentar um tempo de reação a alterações elevado como baixo. Para um tempo de resposta elevado, quanto maior a constante de filtragem (K), maior é o peso imposto às amostras recentes e tem mais sensibilidade às frequências elevadas. Por outro lado, k's menores dão um maior peso atribuído ao histórico e menor é a variação do sinal atribuído. E, assim, não se consegue reagir tão rapidamente às variâncias. Contudo, mais à frente serão comparados vários valores de k para poder fazer a melhor escolha para cada uma das Equações definidas.

Como ilustrado na figura 3.8, a estrutura de dados de entrada no processo de filtragem

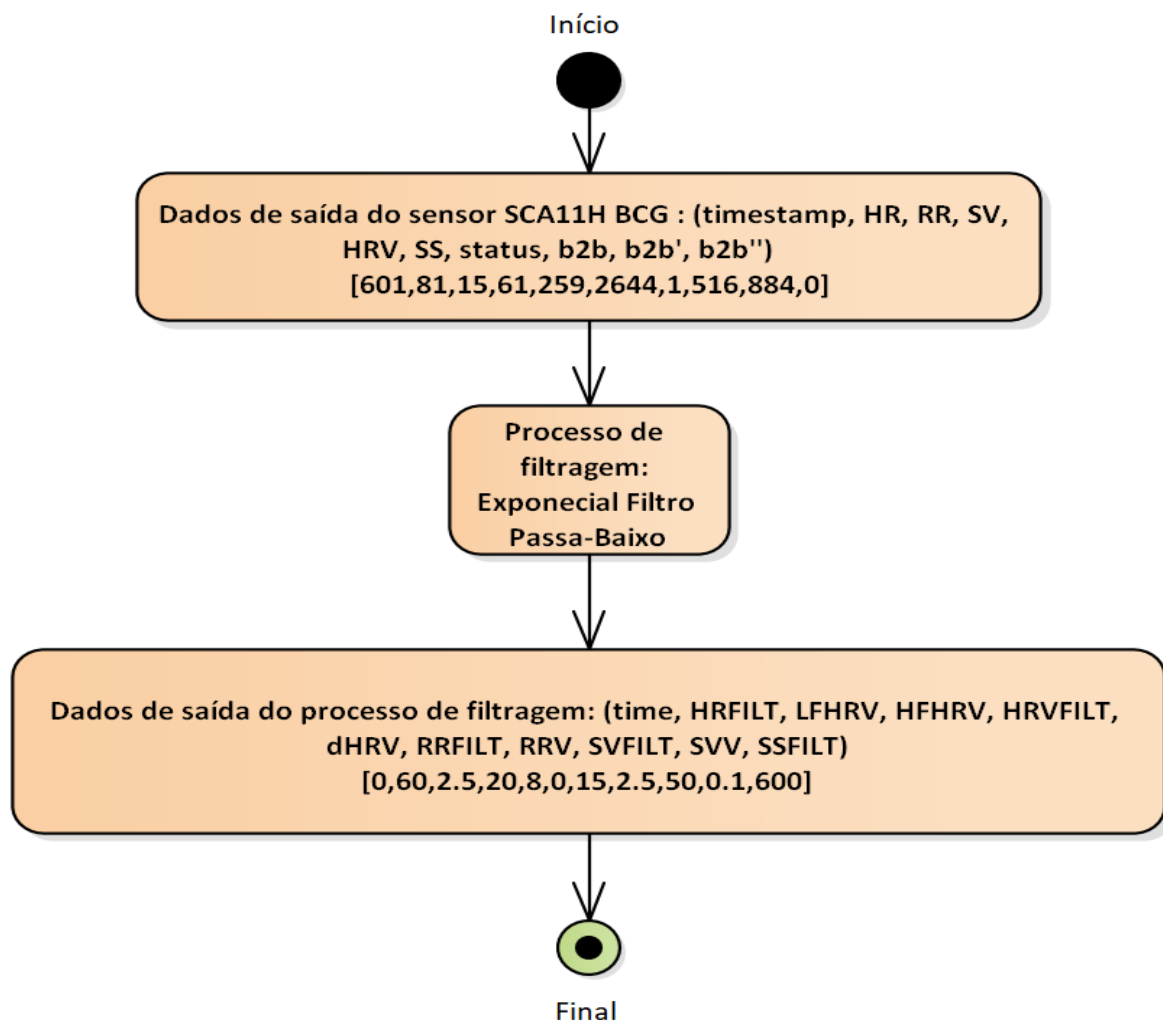


FIGURA 3.8. Estrutura de dados de entrada e saída no processo de filtragem.



(filtro passa-baixo) corresponde aos dados de saída do sensor SCA11H BCG. Os parâmetros provenientes do sensor são: o timestamp, o HR, o RR, o SV, o HRV, o SS, o status, o B2B, o B2B' e o B2B".

O filtro passa-baixo é um componente omnipresente em muitos tipos diferentes de sistemas de processamento de sinal. Assim, o processo de filtragem atenua todos os dados do sensor. Esta atenuação reflete-se acima de uma frequência especificada. Contudo, a quantidade de atenuação para cada frequência varia de filtro para filtro. E, a maioria dos filtros do sistema de controlo são filtros passa-baixo projetados para reduzir o ruído de alta frequência. [21] [58]

Por fim, os dados de saída no processo de filtragem apresentam o tempo e os valores relativos às expressões do Heart Rate Filtered (HRFILT), do LFHRV, do HFHRV, do Heart Rate Variability Filtered (HRVFILT), do derived Heart Rate Variability (dHRV), do Respiration Rate Filtered (RRFILT), do RRV, do Stroke Volume Filtered (SVFILT), do Stroke Volume Variability (SVV) e do Signal Strength Filtered (SSFILT).

Todos os dados provenientes do sensor BCG SCA11H são filtrados usando a Equação 3.1. Esta equação representa um filtro passa-baixo implementado, através de uma média exponencial deslizante. O parâmetro  $k$  corresponde à constante de filtragem (varia entre 0 e 1), o  $x(t)$  é o valor original, ou seja a amostra atual do sinal, o  $y(t-1)$  é o valor filtrado anteriormente e, por fim, o  $y(t)$  é o valor filtrado no instante  $t$ . Quanto maior o  $k$ , menor o esquecimento. [43]

### 3.1. Exponencial filtro passa-baixo ( $y(t)$ )

$$y(t) = y(t - 1) * (1 - k) + x(t) * k \quad (3.1)$$

O HRFILT é calculado com base na Equação 3.1 e mede a filtragem da frequência cardíaca (HR). O  $x(t)$  corresponde ao  $HR(t)$  como valor original e o  $y(t-1)$  corresponde ao HRFILT( $t-1$ ) como valor filtrado no instante anterior, de acordo com a Equação 3.2.

### 3.2. Heart Rate Filtered (HRFILT)

$$HRFILT(t) = HRFILT(t - 1) * (1 - k) + HR(t) * k \quad (3.2)$$

O LFHRV é calculado com base na Equação 3.1 e mede a filtragem de baixas frequências da variabilidade da frequência cardíaca (HRV). O  $x(t)$  corresponde ao valor absoluto da diferença entre o HRFILT( $t$ ) e o HR( $t$ ) como valor original e o  $y(t-1)$  corresponde ao LFHRV( $t-1$ ) como valor filtrado no instante anterior, como mostrado na Equação 3.3.

### 3.3. Low Frequency Heart Rate Variability (LFHRV)

$$LFHRV(t) = LFHRV(t - 1) * (1 - k) + |HRFILT(t) - HR(t)| * k \quad (3.3)$$

O HFHRV é calculado com base na Equação 3.1 e mede a filtragem de altas frequências da variabilidade da frequência cardíaca (HRV). O  $x(t)$  corresponde ao HRV( $t$ ) como valor

original e o  $y(t-1)$  corresponde ao  $HFHRV(t-1)$  como valor filtrado no instante anterior, de acordo com a Equação 3.4.

#### 3.4. High Frequency Heart Rate Variability (HFHRV)

$$HFHRV(t) = HFHRV(t-1) * (1 - k) + HRV(t) * k \quad (3.4)$$

O SVFILT é calculado com base na Equação 3.1 e mede a filtragem do volume sistólico relativo (SV). O  $x(t)$  corresponde ao  $SV(t)$  como valor original e o  $y(t-1)$  corresponde ao  $SVFILT(t-1)$  como valor filtrado no instante anterior, como mostrado na Equação 3.5.

#### 3.5. Stroke Volume Filtered (SVFILT)

$$SVFILT(t) = SVFILT(t-1) * (1 - k) + SV(t) * k \quad (3.5)$$

O SVV é calculado com base na Equação 3.1 e mede a filtragem da variabilidade do volume sistólico relativo (SV). O  $x(t)$  corresponde ao inverso do valor absoluto da diferença entre o  $SV(t)$  e o  $SVFILT(t)$  a dividir por  $SVFILT(t)$  como valor original e o  $y(t-1)$  corresponde ao  $SVV(t-1)$  como valor filtrado no instante anterior, de acordo com a Equação 3.6.

#### 3.6. Stroke Volume Variability (SVV)

$$SVV(t) = SVV(t-1) * (1 - k) + \frac{|SV(t) - SVFILT(t)|}{SVFILT(t) * k} \quad (3.6)$$

O dHRV é calculado com base na Equação 3.1 e mede a filtragem da derivada da variabilidade da frequência cardíaca (HRV). O  $x(t)$  corresponde ao valor da diferença entre o  $HRVFILT(t)$  e o  $HRVFILT(t-1)$  como valor original e o  $y(t-1)$  corresponde ao  $dHRV(t-1)$  como valor filtrado no instante anterior, como mostrado na Equação 3.7.

#### 3.7. derived Heart Rate Variability (dHRV)

$$dHRV(t) = dHRV(t-1) * (1 - k) + HRVFILT(t) - HRVFILT(t-1) * k \quad (3.7)$$

O HRVFILT é calculado através da relação entre os componentes HF e LF. Para reduzir a modulação causada pela respiração, é necessário ampliar o HRVFILT com o SVV. [43] Ao multiplicar o  $HFHRV(t)$  por 10, obteve-se valores a rondar o valor definido como constante inicial (igual a 8), de acordo com a Equação 3.8.

#### 3.8. Heart Rate Variability Filtered (HRVFILT)

$$HRVFILT(t) = \frac{HFHRV(t) * 10}{LFHRV(t) * SVV(t)} \quad (3.8)$$

O RRFILT é calculado com base na Equação 3.1 e mede a filtragem da taxa de respiração (RR). O  $x(t)$  corresponde ao  $RR(t)$  como valor original e o  $y(t-1)$  corresponde ao  $RRFILT(t-1)$  como valor filtrado no instante anterior, como mostrado na Equação 3.9.

### 3.9. Respiration Rate Filtered (RRFILT)

$$RRFILT(t) = RRFILT(t - 1) * (1 - k) + RR(t) * k \quad (3.9)$$

O RRV é calculado com base na Equação 3.1 e mede a filtragem da variabilidade da taxa de respiração (RR). O  $x(t)$  corresponde ao valor absoluto da diferença entre o  $RR(t)$  e o  $RRFILT(t)$  como valor original e o  $y(t-1)$  corresponde ao  $RRV(t-1)$  como valor filtrado no instante anterior, de acordo com a Equação 3.10.

### 3.10. Respiration Rate Variability (RRV)

$$RRV(t) = RRV(t - 1) * (1 - k) + |RR(t) - RRFILT(t)| * k \quad (3.10)$$

O SSFILT é calculado com base na Equação 3.1 e mede a filtragem da força do sinal (SS). O  $x(t)$  corresponde ao  $SS(t)$  como valor original e o  $y(t-1)$  corresponde ao  $SSFILT(t-1)$  como valor filtrado no instante anterior, de acordo com a Equação 3.11.

### 3.11. Signal Strength Filtered (SSFILT)

$$SSFILT(t) = SSFILT(t - 1) * (1 - k) + SS(t) * k \quad (3.11)$$

A escolha ideal de todas as constantes iniciais consistia num estudo para cada utilizador da aplicação. Ou seja, o processo de filtragem não era iniciado no instante temporal, igual a zero segundos ( $t=0s$ ). Consequentemente, o processo de monitorização aguarda um tempo pré-definido (5 minutos), para que seja possível definir os valores iniciais para a pessoa, em estudo. No entanto, a abordagem implementada na aplicação aqui referida correspondeu a um caso em particular. Neste caso, todas as constantes iniciais, presentes na tabela 3.3 foram calculadas, de acordo com o valor médio dos dados originais provenientes do sensor BCG SCA11H, entre eles, o HR, o HRV, o SV, o RR e o SS.

Para melhor classificar o processamento dos diversos sinais, a figura 3.9, representa as dependências entre as expressões presentes na filtragem de dados. As expressões assinaladas a verde correspondem aos valores finais usados no cálculo da fase do sono (HRFILT, HRVFILT, dHRV, RRV, SVV e SSFILT). Por sua vez, a expressão HRFILT depende do valor de HR, a expressão HFHRV depende do HRV, a expressão RRFILT depende do RR, a expressão SVFILT depende do SV, a expressão SSFILT depende do SS. A expressão LFHRV depende do HRFILT e do HR, a expressão RRV depende do RRFILT e do RR, a expressão SVV depende do SVFILT e do SV. Já, o HRVFILT depende do LFHRV, do HFHRV e do SVV e, por fim, o dHRV depende do HRVFILT.

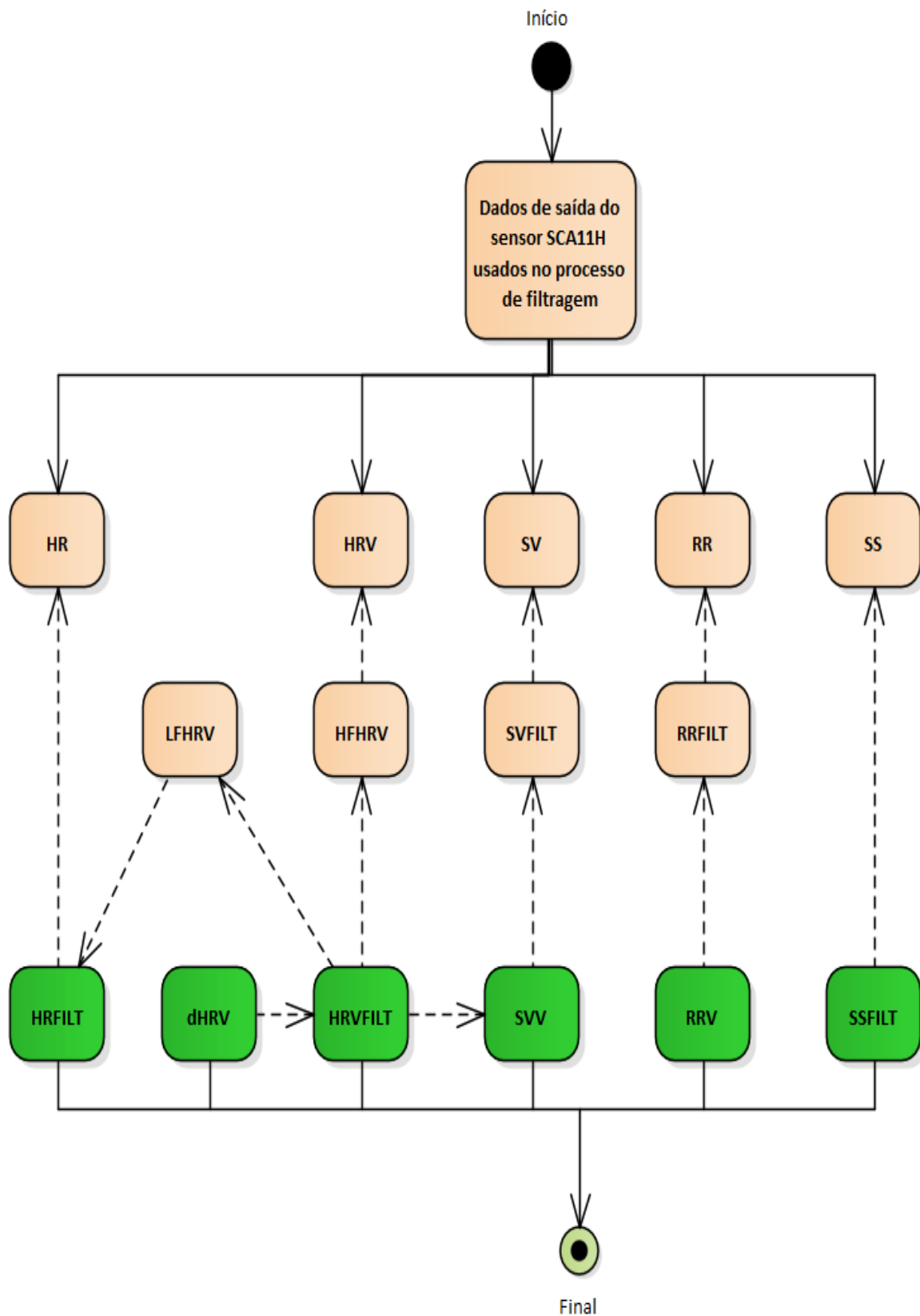


FIGURA 3.9. Dependências entre as expressões presentes na filtragem de dados.

TABELA 3.3. Constantes iniciais (instante  $t = 0$  segundos) para os seguintes parâmetros: HRFILT, LFHRV, HFHRV, SVFILT, SVV, dHRV, HRVFILT, RRFILT, RRV e SSFILT.

Parâmetros	Constante inicial ( $t = 0s$ )
HRFILT	60
LFHRV	2.5
HFHRV	20
SVFILT	50
SVV	0.1
dHRV	0
HRVFILT	8
RRFILT	15
RRV	2.5
SSFILT	600

### 3.2.3. Aprendizagem de modelos

Ao longo da implementação da proposta, existem duas fases bem definidas e dependentes. Ou seja, como primeira fase impõe-se a construção do modelo, através da aprendizagem do modelo e como segunda fase apresenta a aplicação desse mesmo modelo, onde irá realizar a predição do sono inerente. A primeira fase recai sobre uma avaliação de desempenho e uma aprendizagem de modelo para todos os algoritmos de classificação, analisadas mais à frente. Por sua vez, a segunda fase que se prende com a utilização do modelo corresponde à aplicação da API Weka diretamente na aplicação proposta.

A API Weka é usada na implementação da aplicação "GoToBed". Todo o processo de testes de desempenho e de aprendizagem dos algoritmos de classificação é realizado diretamente através da ferramenta Weka. Por sua vez, o uso da API tem como objetivo obter a fase do sono com o apoio do modelo criado externamente. A função que calcula o estado do sono (0, 1, 2 ou 3), de trinta em trinta segundos, apresenta como parâmetros de entrada, o tempo, o HRFILT, o HRVFILT, o dHRV, o SVV, o RRV e o SSFILT. A primeira parte da função tem como objetivo a construção de um ficheiro ARFF, de acordo com o formato presente na figura 3.10. [44] Esta função usa o modelo RandomForest, que apresentou uma melhor performance durante toda a aprendizagem dos algoritmos de classificação, para efetuar a predição do estado do sono, durante uma noite. Consequentemente, esta predição/classificação realiza-se para cada instância ('0,60,8,0.1,2.5,600,?' - Figura 3.10), substituindo o ponto de interrogação por um estado do sono usando a função intrínseca da API "RandomForest.classifyInstance('0,60,8,0.1,2.5,600,?')". [45] [46] [60]

```
@relation sleepStageFilter
@attribute time numeric
@attribute hrfilt numeric
@attribute hrvfilt numeric
@attribute dhrv numeric
@attribute svvfilt numeric
@attribute rrvfilt numeric
@attribute ssfilt numeric
@attribute sleepRef {0,1,2,3}

@data
0,60,8,0,0.1,2.5,600,?
```

FIGURA 3.10. Estrutura dos dados para efetuar a predição para uma dada noite.



## Testes e Resultados

### 4.1. Constante de filtragem (K)

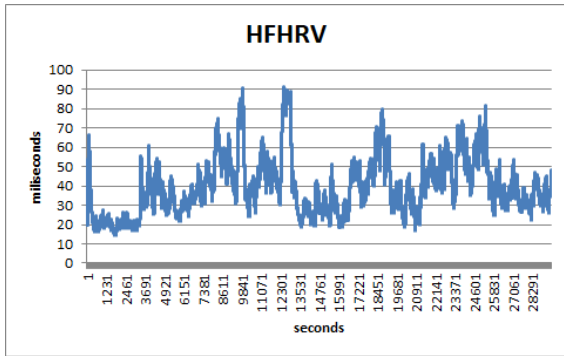
A escolha da constante de filtragem ( $k$ ) é um processo muito importante no decorrer deste estudo. Tem como principal objetivo a obtenção dos valores de  $k$  para o processo de filtragem. Esta constante admite um valor entre 0 e 1. Todas as expressões envolvidas no processo de filtragem passaram por esta fase de decisão. Ou seja, todas foram sujeitas a testes semelhantes com os mesmos valores de  $k$ . Neste caso, a expressão implícita diz respeito ao HFHRV. As potências de base 2 são usadas nas constantes de filtragem. Estas têm como vantagem uma maior rapidez de processamento e menor consumo de energia, devido a não serem usadas operações de divisão, mas, sim, shifts binários feitos em apenas um ciclo de relógio no processador. Neste caso, quanto maior o período do sinal HFHRV, menor a frequência de amostragem.

A necessidade de filtragem surge sobre o efeito da eliminação do ruído nos sinais. Assim, é possível extrair-se informações relevantes e remover informações indesejadas. Por sua vez, o objetivo principal consiste num sinal com o mínimo de ruído e o máximo de informação relevante.

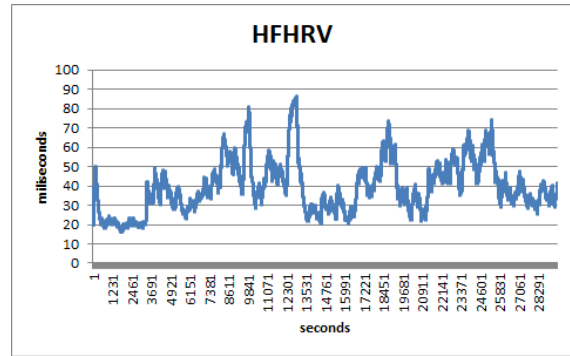
Conseqüentemente, as figuras 4.1a e 4.1b, apresentam sinais com bastante ruído. Logo, estas opções são imediatamente descartadas e existe a necessidade de testar novamente com um  $k$  inferior. Ao serem efetuados novos testes para constantes de filtragem iguais a  $\frac{1}{256}$  e  $\frac{1}{512}$ , resultaram as figuras 4.1c e 4.1d. Nesta fase, o nível de ruído diminuiu significativamente, sem a perda aparente de informação importante. Mas, mesmo assim, a figura 4.1d, ainda apresenta algum ruído. E, de seguida, prossegue-se com novos testes em busca do melhor sinal possível. Estes testes representam como constantes de filtragem os seguintes valores:  $\frac{1}{1024}$  (figura 4.1e) e  $\frac{1}{2048}$  (figura 4.1f). Posteriormente, analisando as figuras 4.1e e 4.1f, concluímos que a figura 4.1e é a que contém empiricamente o melhor resultado possível, ao nível de sinal e de filtragem. Visto que, a figura 4.1f, apesar de mostrar um nível de ruído semelhante, apresenta uma perda de informação relevante. Pelo contrário, a figura 4.1e, incide sobre um sinal com o mínimo de ruído possível e o máximo de informação relevante possível. Em suma, estes picos presentes no sinal pretendido representam os ciclos do sono. E, neste caso, pode-se concluir que ocorreram quatro ciclos do sono, durante a noite, em estudo.

A tabela 4.1, representa todos os valores escolhidos, após esta fase de testes, para todas as expressões sujeitas ao processo de filtragem.

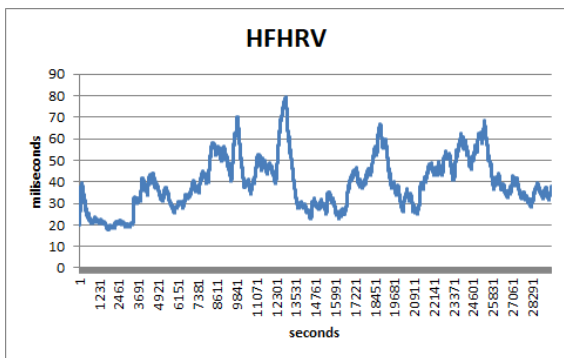




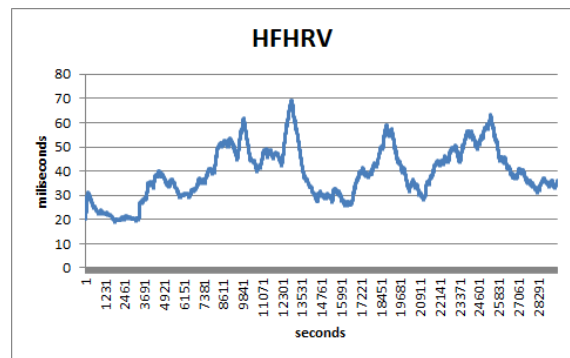
(A) Imagem referente à constante de filtra-  
gem igual a  $\frac{1}{64}$  do sinal HFHRV.



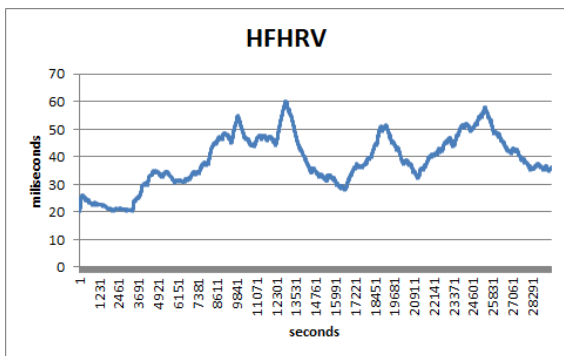
(B) Imagem referente à constante de filtra-  
gem igual a  $\frac{1}{128}$  do sinal HFHRV.



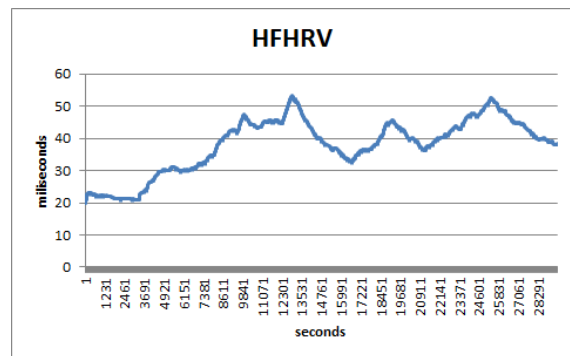
(C) Imagem referente à constante de filtra-  
gem igual a  $\frac{1}{256}$  do sinal HFHRV.



(D) Imagem referente à constante de filtra-  
gem igual a  $\frac{1}{512}$  do sinal HFHRV.



(E) Imagem referente à constante de filtra-  
gem igual a  $\frac{1}{1024}$  do sinal HFHRV.



(F) Imagem referente à constante de filtra-  
gem igual a  $\frac{1}{2048}$  do sinal HFHRV.

FIGURA 4.1. Constantes de filtragem iguais ao inverso de potências de base 2 (6, 7, 8, 9, 10 e 11) do sinal HFHRV.

TABELA 4.1. Valores da constante de filtragem para cada expressão de filtragem.

Variável	Constante de filtragem (K)
HRFILT	$\frac{1}{256}$
LFHRV	$\frac{1}{1024}$
HFHRV	$\frac{1}{1024}$
SVFILT	$\frac{1}{256}$
SVV	$\frac{1}{1024}$
dHRV	$\frac{1}{1024}$
RRFILT	$\frac{1}{1024}$
RRV	$\frac{1}{1024}$
SSFILT	$\frac{1}{1024}$

## 4.2. Avaliação dos Algoritmos de classificação

Nesta avaliação, pretende-se testar um dado conjunto de testes para cada algoritmo de classificação, em estudo. Para a sua resolução, vai ser usada a ferramenta Weka Experiment Environment. Ao nível da solução, espera-se obter os quatro melhores algoritmos de classificação para se efetuar, de seguida, a aprendizagem do modelo com os mesmos.

Os dados, referentes a doze pessoas (mulheres e homens) com idades compreendidas entre 24 e 46 anos (doze noites), usados para efetuar os testes, ao nível, de desempenho e de aprendizagem dos modelos, foram fornecidos pela empresa Murata Electronics (Finlândia). Estes dados são constituídos pelos dados de saída do Sensor SCA11H (BCG) e pelo valor de referência do estado do sono (PSG), retirados a cada trinta segundos, durante todas as 12 noites.

A figura 4.2, ilustra o processo de testes de desempenho realizados para cada Algoritmo de classificação, em estudo, num conjunto de dados de teste. Os algoritmos de classificação Weka, em estudo, são os seguintes: rules.ZeroR, bayes.NaiveBayes, functions.Logistic, functions.SMO, lazy.IBk, rules.PART, trees.REPTree, trees.J48, trees.RandomForest. A escolha destes algoritmos foi baseada nos mais comuns e usados, ao nível de problemas de classificação multi-classe, na ferramenta Weka. Toda esta experimentação foi executada com o apoio da ferramenta Weka Experiment Environment. No tipo de experiência definimos uma validação cruzada e incluímos 12 amostras (ficheiro ARFF) referentes a 12 noites. Todas as amostras apresentam a mesma estrutura tanto a nível de atributos como de classificação, como se pode verificar na figura 4.3. Ao nível de classificação, é sempre o último atributo que é usado, neste caso, o 'sleepRef' (Figura 4.3). Todos os outros atributos, entre eles, o tempo, o HRFILT, o HRVFILT, o dHRV, o SVV, o RRV e o SSFILT, são usados para a aprendizagem dos diferentes algoritmos em estudo e calculados, de acordo com o sub-capítulo referente à filtragem dos dados. [7] [6] Tipicamente, cada amostra apresenta, aproximadamente, mil instâncias de dados (depende da duração do sono).

Por fim, inicializa-se todo o processo de avaliação de desempenho para cada algoritmo de classificação. Esta análise é sequencial, ou seja, é executado um algoritmo de cada vez para o mesmo conjunto de dados de teste (12 noites) e, assim, sucessivamente. O resultado final irá ditar quais os 4 algoritmos com um melhor nível de desempenho, neste caso, uma maior percentagem. [7] [6]

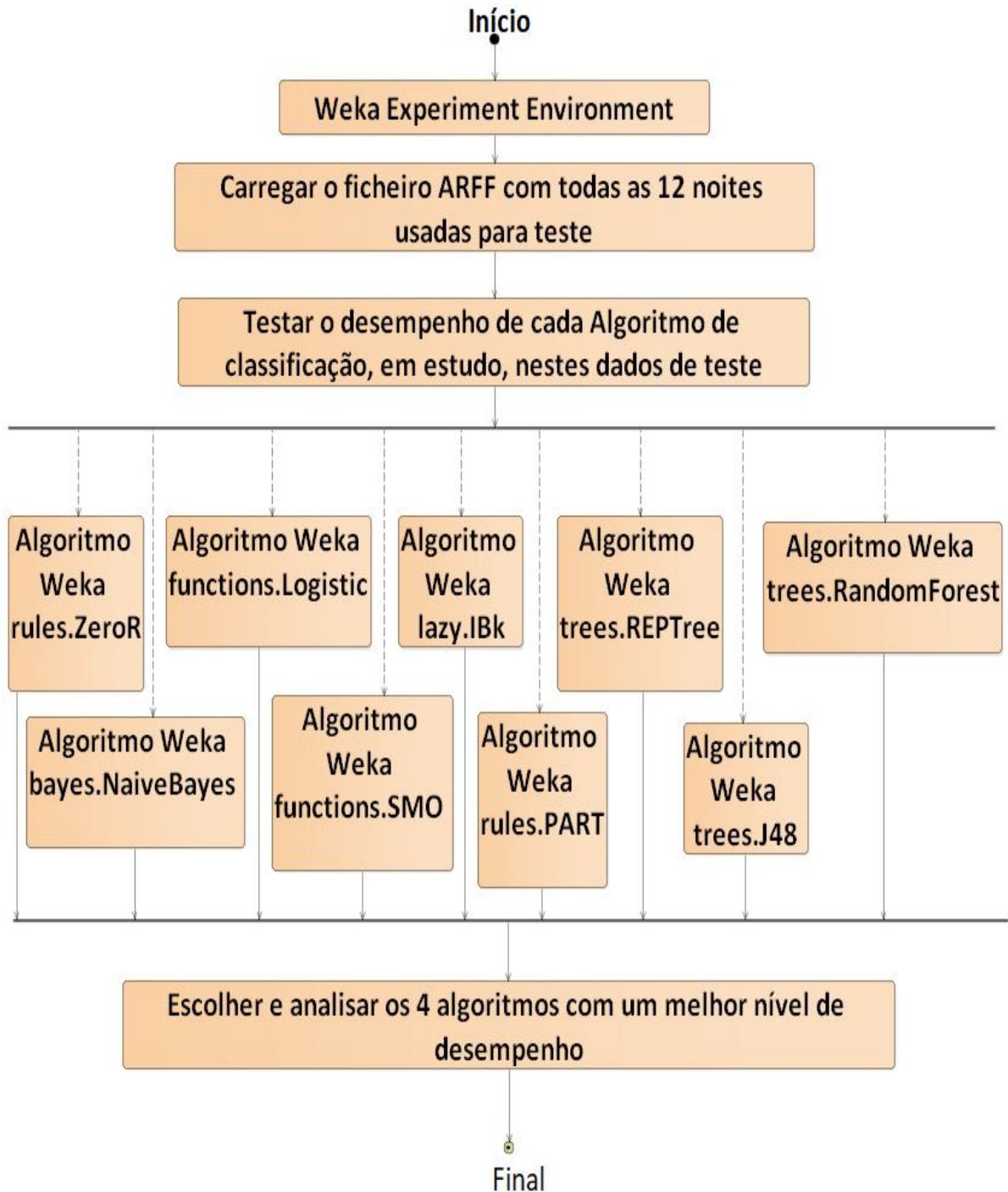


FIGURA 4.2. Diagrama de atividades para efetuar o teste de desempenho de cada Algoritmo de classificação, em estudo, no conjunto de dados de teste.

```

@relation sleepStageFilter

@attribute time numeric
@attribute hrfilt numeric
@attribute hrvfilt numeric
@attribute dhrv numeric
@attribute svv numeric
@attribute rrv numeric
@attribute ssfilt numeric
@attribute sleepRef {0,1,2,3}

@data
0,60,8,0,0.1,2.5,600,0

```

FIGURA 4.3. Estrutura dos dados para efectuar a aprendizagem dos algoritmos (Ficheiro ARFF).

Ao analisar a tabela 4.2, verifica-se que os melhores algoritmos de classificação, ao nível de performance, são os seguintes: `trees.RandomForest`, `lazy.IBk`, `trees.J48` e `rules.PART`.

TABELA 4.2. Classificação dos diferentes algoritmos de classificação, ao nível, de performance.

Algoritmos	Performance (%)
<code>rules.ZeroR</code>	44.35
<code>bayes.NaiveBayes</code>	51.02
<code>functions.Logistic</code>	55.57
<code>functions.SMO</code>	52.17
<b><code>lazy.IBk</code></b>	<b>90.29</b>
<b><code>rules.PART</code></b>	<b>85.47</b>
<code>trees.REPTree</code>	81.54
<b><code>trees.J48</code></b>	<b>86.42</b>
<b><code>trees.RandomForest</code></b>	<b>91.81</b>

### 4.3. Aprendizagem dos Algoritmos de classificação

Nesta aprendizagem, pretende-se treinar um dado conjunto de testes para os quatro melhores algoritmos de classificação, em estudo. Para a construção do modelo, vai ser usada a ferramenta Weka Explorer. Ao nível da solução, espera-se obter o melhor algoritmo de classificação para se efetuar, de seguida, a construção do modelo final. Com este modelo definido, serão efetuadas as predições das várias fases do sono.

A figura 4.4, ilustra o processo de testes de aprendizagem realizados para cada Algoritmo de classificação, em estudo, num conjunto de dados de teste. Neste estudo, juntaram-se sempre todos os dados referentes às doze noites. Nesta aprendizagem, usou-se um ficheiro ARFF com as onze noites para efetuar a aprendizagem do modelo e, para testar o mesmo, um ficheiro ARFF com a noite que não estava incluída na aprendizagem e, assim, sucessivamente para todas as noites e para todos os 4 melhores algoritmos de classificação, em estudo. Entre eles, o `trees.RandomForest`, o `trees.J48`, o `rules.PART` e

o lazy.IBk. O Weka Explorer é a ferramenta usada para efetuar a aprendizagem, por forma, a obter um modelo final com o melhor nível de desempenho possível. [7] [5] Com este modelo final, para os melhores algoritmos de classificação, efetua-se a predição de todas as fases do sono (de 30 em 30 segundos), durante uma noite. Esta fase de predição, está implementada diretamente na aplicação GoToBed, isto é, já é uma função interna da aplicação. Enquanto que, toda a aprendizagem do melhor modelo associado ao algoritmo de classificação, é realizado externamente, acedendo diretamente à ferramenta Weka.

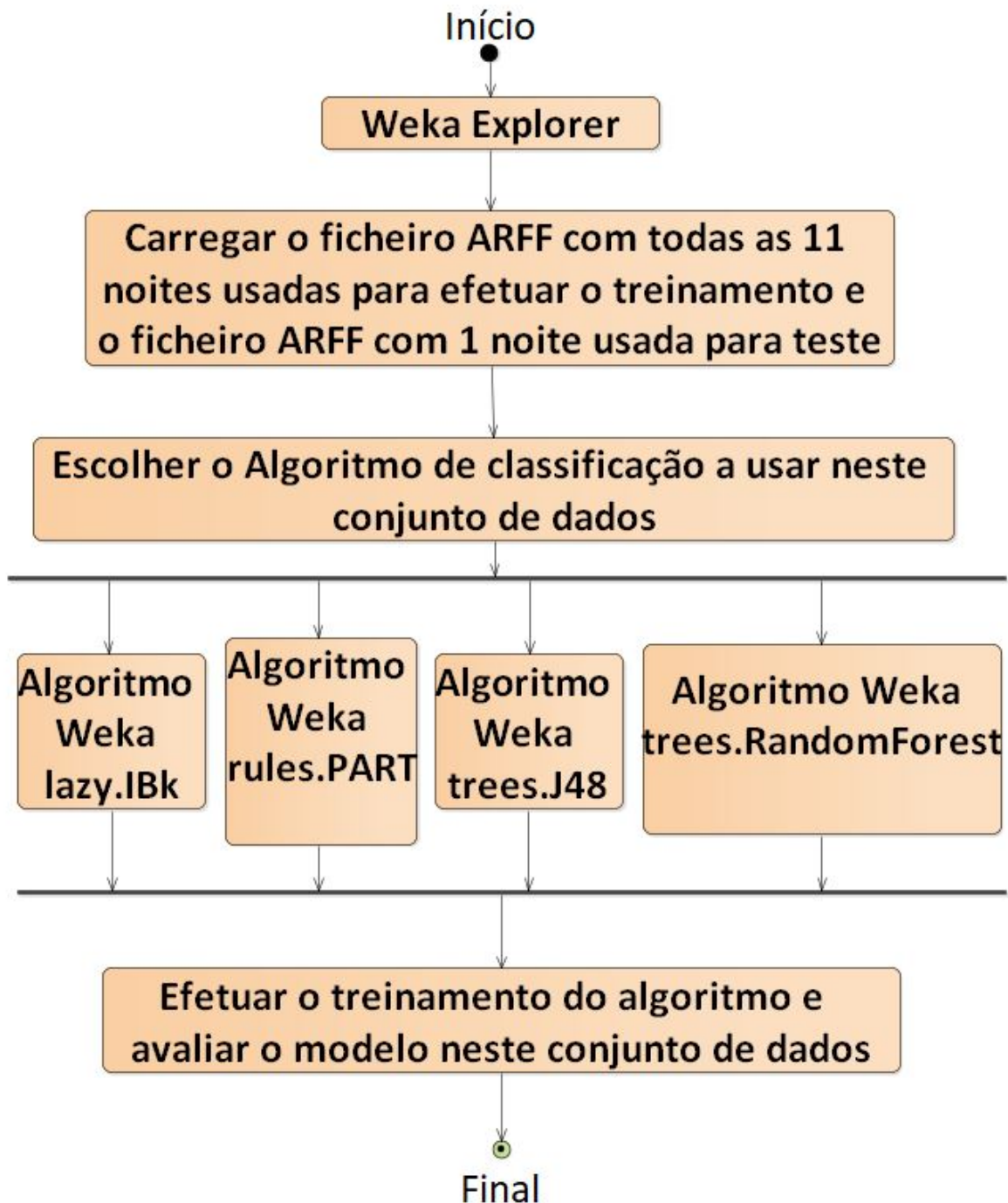


FIGURA 4.4. Diagrama de atividades para efetuar a aprendizagem de cada Algoritmo de classificação no conjunto de dados de teste.

Nesta ferramenta, está implementado todo o processo de ML. A avaliação do modelo, neste conjunto de dados, depende do sumário obtido a partir dos resultados de cada aprendizagem. [7] [1] [4] Os dados principais desses mesmos sumários, vão ser analisados de seguida, com recurso a tabelas e figuras. Entre eles, a percentagem de instâncias classificadas corretamente, a estatística do kappa e as matrizes de confusão.

A estatística do kappa é uma estatística robusta útil para testes de confiabilidade entre avaliadores ou intraexaminadores. E, serve, para interpretar o resultado como nenhuma concordância para valores inferiores a zero, como nenhuma a leve concordância para valores entre 0.01 e 0.20, como regular para valores entre 0.21 e 0.40, como moderado para valores entre 0.41 e 0.60, como substancial para valores entre 0.61 e 0.80 e, por fim, como acordo quase perfeito para valores entre 0.81 e 1. [31]

Neste caso, as matrizes de confusão servem para comparar um conjunto de amostras previamente definido (PSG) com as predições do sono definidas através do modelo (BCG). Assim, é possível concluir qual a percentagem de amostras BCG que correspondem à verdade (PSG).

A Equação 4.1 diz respeito ao cálculo da estatística do kappa (K). Como a maioria das estatísticas de correlação, o kappa pode variar entre -1 e +1. Este cálculo resulta diretamente de uma matriz de confusão (tabela 4.3). [31]

#### 4.1. Estatística do kappa (K)

$$K = \frac{P_r(a) - P_r(e)}{1 - P_r(e)} \quad (4.1)$$

onde  $P_r(a)$  representa a precisão observada e  $P_r(e)$  representa a precisão esperada.

O cálculo da precisão observada corresponde à Equação 4.2. [31] onde  $cell(normal:normal)$

#### 4.2. Precisão Observada ( $P_r(a)$ )

$$P_r(a) = \frac{cell(normal : normal) + cell(abnormal : abnormal)}{n} \quad (4.2)$$

representa o valor de 147,  $cell(abnormal:abnormal)$  representa o valor de 62 e  $n$  representa o número total de instâncias/observações (=222), como ilustrado na tabela 4.3.

O cálculo da precisão esperada corresponde à Equação 4.3. [31] onde  $cm^1$  representa

#### 4.3. Precisão Esperada ( $P_r(e)$ )

$$P_r(e) = \frac{\left(\frac{cm^1 * rm^1}{n}\right) + \left(\frac{cm^2 * rm^2}{n}\right)}{n} \quad (4.3)$$

a coluna 1 marginal (=157),  $rm^1$  representa a linha 1 marginal (=150),  $cm^2$  representa a coluna 2 marginal (=65),  $rm^2$  representa a linha 2 marginal (=72) e  $n$  representa o

número total de instâncias/observações (=222), de acordo com a tabela 4.3.

Em suma, quanto maior a precisão esperada, menor o valor resultante da estatística do kappa. [31]

TABELA 4.3. Dados para exemplo referente ao cálculo da estatística do kappa. [31]

		Rater 1		Row Marginals	
		normal	abnormal		
Rater 2	normal	147	3	150	rm <sup>1</sup>
	abnormal	10	62	72	rm <sup>2</sup>
Column Marginals		157	65	222	n
		cm <sup>1</sup>	cm <sup>2</sup>		

Analisando as tabelas 4.4 e 4.5, no caso dos algoritmos de classificação lazy.IBk e trees.RandomForest, a noite que obteve melhores resultados foi a 'PSG14'. Enquanto que, nos algoritmos de classificação trees.J48 e rules.PART, os melhores resultados recaem sobre a noite 'PSG10'. Posteriormente, a figura 4.6 apresenta as matrizes de confusão para as melhores noites e, por sua vez, para cada algoritmo de classificação.

No sumário dos quatro melhores algoritmos de classificação (lazy.IBk, trees.J48, rules.PART e trees.RandomForest), obtém-se informação para as seis melhores noites, ao nível, das instâncias classificadas corretamente (tabela 4.4) e da estatística do kappa (tabela 4.5). O algoritmo que apresenta melhores resultados é o trees.RandomForest e, conseqüentemente, a melhor noite recai sobre a 'PSG14' com um valor de, 59.0278% (instâncias classificadas corretamente) e 0.4042 (estatística do kappa). [7] [5]

TABELA 4.4. Instâncias classificadas corretamente nos diferentes algoritmos (%).

Noites	lazy.IBk (%)	trees.J48 (%)	rules.PART (%)	trees.RandomForest (%)
PSG08	44.3378	42.3225	42.5144	54.7985
PSG10	42.6488	45.4435	47.6306	51.8834
PSG11	27.7584	41.6957	37.7468	51.9164
PSG13	33.0454	33.80	31.6415	43.4125
PSG14	47.6852	38.6574	41.3194	59.0278
PSG16	47.1885	28.6659	33.5171	42.9989

TABELA 4.5. Estatística do kappa nos diferentes algoritmos.

Noites	lazy.IBk	trees.J48	rules.PART	trees.RandomForest
PSG08	0.2219	0.1683	0.1565	0.3025
PSG10	0.1696	0.216	0.2414	0.271
PSG11	-0.0361	0.1016	0.0711	0.13
PSG13	0.0787	0.0997	0.0624	0.2156
PSG14	0.2242	0.1368	0.1538	0.4042
PSG16	0.172	0.0531	0.1122	0.1332

No sumário referido anteriormente, também estão incluídas as taxas de True Positives (TP) e False Positives (FP). Um verdadeiro positivo é um resultado em que o modelo prediz corretamente a classe positiva. Conseqüentemente, um falso positivo é um resultado em que o modelo prevê incorretamente a classe positiva. Por exemplo, analisando a figura 4.6a, obtemos este tipo de valores. Os valores (assinalados a laranja) 27, 85, 206, 192 correspondem a verdadeiros positivos. Enquanto que, os restantes valores correspondem a falsos positivos. [15]

Analisando figuras 4.6a, 4.6b, 4.6c e 4.6d, dizem respeito à melhor noite resultante do estudo para os melhores quatro algoritmos/modelos, entre eles, o trees.RandomForest, o lazy.IBk, o trees.J48 e o rules.PART. Estas matrizes de confusão são compostas pela verdade fundamental, isto é, pelos estados do sono referentes ao PSG (colunas) e pelo ML, ou seja, pelos valores de predição das fases do sono referente ao BCG (linhas). E, assim, compara-se os valores de BCG com PSG através de uma matriz de confusão, como ilustrado na figura 4.6. A matriz de confusão perfeita terá valores superiores a zero apenas nas células que formem uma diagonal (figura 4.6a). Analisando a figura 4.6a, os maiores valores entre colunas e linhas estão nas posições desejadas. Ao comparar as figuras 4.6a, 4.6b, 4.6c e 4.6d, concluí-se que o resultado da matriz de confusão referente ao algoritmo trees.RandomForest (Figura 4.6a) é o mais próximo do esperado. Isto é, os maiores valores para cada coluna estão nas células corretas (0:0 = 27; 1:1 = 85, 2:2 = 206, 3:3 = 192) formando uma diagonal como está assinalado a laranja.

A figura 4.5, mostra, na forma de gráfico, a comparação da indicação da fase do sono obtida com o PSG e com o obtido pelo BCG. Este gráfico apresenta os dados do sono provenientes do PSG referente à noite 'PSG14' e a predição BCG proveniente do modelo final criado com o apoio do algoritmo Weka trees.RandomForest. Ao analisar esta figura, verifica-se que existe uma grande semelhança nas transições entre o sinal PSG e o sinal da predição BCG. O tempo gasto durante o estado do sono profundo (=3) na predição BCG está próximo do original (PSG). Apenas está dividido, maioritariamente, com a fase do sono leve. Enquanto que, o estado do sono leve (=2) na predição BCG, não é muito semelhante ao original (PSG). Está dividido, significativamente, entre as fases do sono profundo e REM (3 e 1, respetivamente). Por sua vez, o estado do sono REM está confuso na predição BCG com a fase do sono leve (=2). E, também, os valores preditos



não estão muito próximos do original (PSG). Por fim, o estado do sono acordado é o que apresenta uma maior discrepância entre o gráfico original (PSG) e o preditivo (BCG). Todas estas conclusões foram baseadas com os resultados da figura 4.6a. Contudo, a confusão predomina sobre as fases do sono profundo (=3), leve (=2) e REM (=1). Assim, o algoritmo escolhido para ser implementado na aplicação é o Random Forest. Por fim, conclui-se que esta predição é eficientemente eficaz para aplicações não médicas, com a atuação no meio ambiente para melhorar a qualidade do sono.

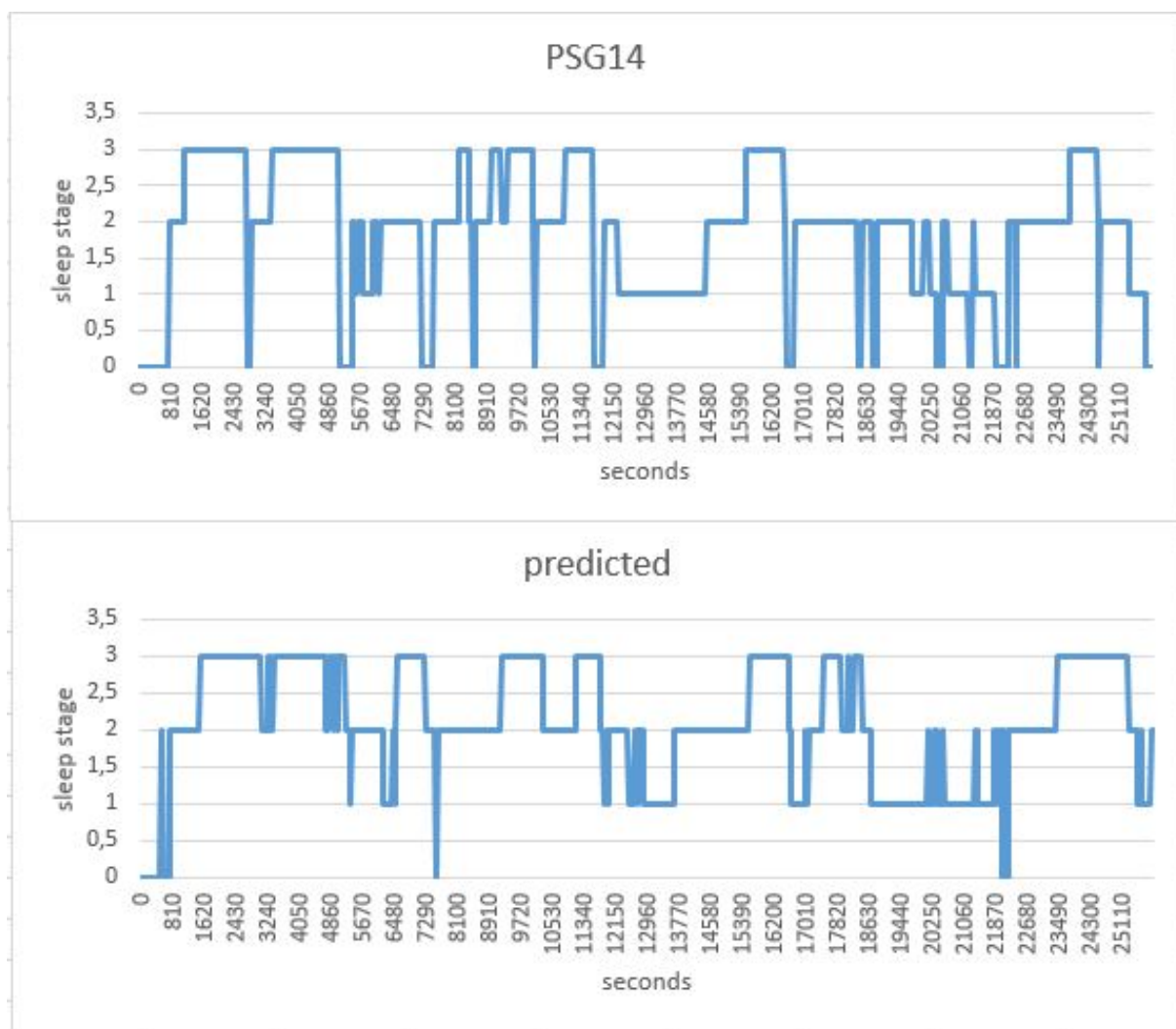


FIGURA 4.5. Gráficos entre os dados do estado do sono da noite PSG14 e a predição BCG para o algoritmo Random Forest.

	0	1	2	3
0	27	22	37	19
1	0	85	69	0
2	2	63	206	102
3	0	0	40	192

(A) Matriz de confusão entre os dados do estado do sono da noite PSG14 e a predição BCG para o algoritmo Weka trees.RandomForest.

	0	1	2	3
0	44	14	27	3
1	29	54	85	3
2	59	30	195	59
3	8	0	114	99

(B) Matriz de confusão entre os dados do estado do sono da noite PSG10 e a predição BCG para o algoritmo Weka rules.PART.

	0	1	2	3
0	38	5	31	14
1	43	22	88	18
2	58	25	188	72
3	15	0	80	126

(C) Matriz de confusão entre os dados do estado do sono da noite PSG10 e a predição BCG para o algoritmo Weka trees.J48.

	0	1	2	3
0	33	18	48	6
1	18	69	67	0
2	44	65	229	35
3	3	11	137	81

(D) Matriz de confusão entre os dados do estado do sono da noite PSG14 e a predição BCG para o algoritmo Weka lazy.IBk.

FIGURA 4.6. Matrizes de confusão entre os dados do estado do sono da melhor noite e a predição BCG para os melhores algoritmos de classificação Weka, em estudo.

#### 4.4. Análise Funcional

Nesta fase da análise funcional, espera-se concluir que o objetivo principal deste estudo foi cumprido com sucesso. É necessário demonstrar um caso em que é despoletado o alarme 15 minutos, antes da hora prevista e outro caso onde esse alarme não é despoletado, antes da hora previamente definida do alarme.

As fases do sono estão identificadas através de um valor inteiro de referência (0, 1, 2 ou 3) que, por sua vez, correspondem a uma fase do sono pré-definida na tabela 4.6.

TABELA 4.6. Fases do sono.

Referência	Estado do sono
0	Acordado
1	REM
2	Sono Leve
3	Sono Profundo

As figuras 4.7 e 4.8, representam o objetivo funcional, ao nível de atuação, da aplicação "GoToBed". Consequentemente, nas mesmas figuras, estão presentes duas linhas, uma

assinalada a verde e outra assinalada a laranja ('ALARM'). Por sua vez, a marca a verde corresponde ao som relaxante. Enquanto que, a marca a laranja incide sobre a hora pré-definida para tocar o alarme, através de um som normal.

Consequentemente, para o estado do sono profundo, a atuação ocorre ao nível de um som mais relaxante, enquanto que, para todos os outros estados do sono (leve, REM e acordado), o som resultante é mais intenso. Em suma, este tipo de atuação tem como objetivo um acordar mais calmo na fase do sono profundo e um acordar mais repentino nas outras fases. É importante referir que os resultados referentes à fase do sono são calculados de trinta em trinta segundos. Neste caso, num caso real (figura 4.7), a pessoa, em estudo, está na fase do sono profundo, 15 minutos antes de tocar o alarme. E, assim, a aplicação inicia uma música mais calma. Daí, o sinal a verde apresentar um impulso, onde é ativada a música relaxante. E, após os 15 minutos, inicia uma música mais intensa para um acordar mais rápido, na hora prevista para tocar o alarme ('ALARM'). O objetivo desta música relaxante consiste, principalmente, numa mudança da fase do sono profundo para a fase do sono leve. Em suma, os dois tipos de música, presentes na forma de atuação no meio ambiente, são despoletados.

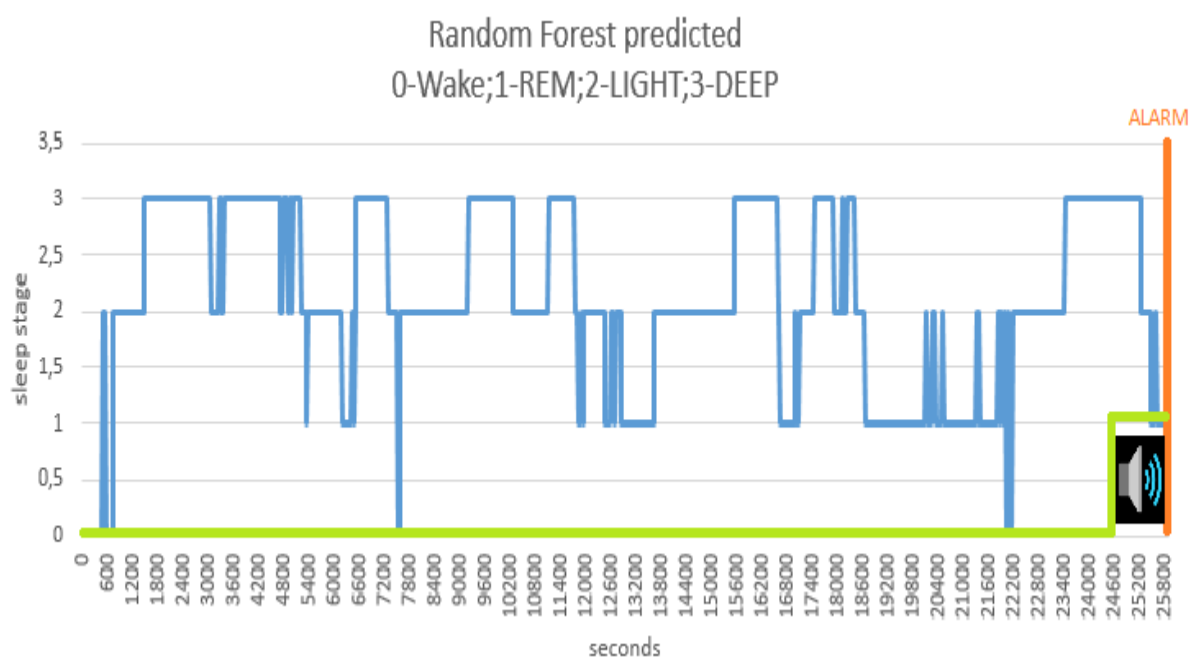


FIGURA 4.7. Análise funcional da aplicação, em estudo (Música relaxante é acionada).

Por outro lado, num caso real (figura 4.8), a pessoa, em estudo, encontra-se na fase do sono leve, 15 minutos antes de tocar o alarme. Logo, a aplicação já não aciona uma música mais relaxante (ausência de impulso do sinal a verde). Em suma, só inicia uma música mais intensa para um acordar mais rápido, na hora prevista para tocar o alarme ('ALARM').

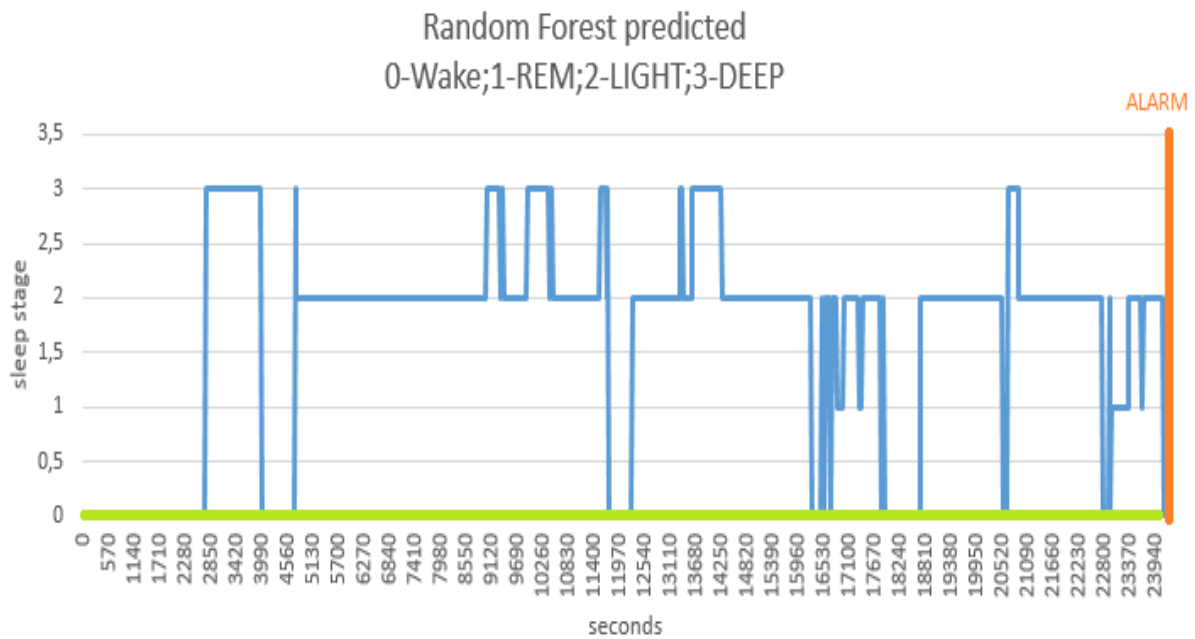


FIGURA 4.8. Análise funcional da aplicação, em estudo (Música relaxante não é acionada).

#### 4.5. Atividades

O desenvolvimento da aplicação, consiste na implementação das várias atividades, ao nível de código que não é visível e do código da interface que já é visível. Aqui, vão ser testadas todas as funções inerentes de cada atividade imposta na aplicação proposta.

A aplicação GoToBed representa um conjunto de 5 atividades. Na Figura 4.9a, o utilizador verifica os dados principais resultantes do sensor SCA11H BCG em tempo real a cada segundo, entre eles, o HR, o HRV, o RR e o SV. E, por fim, tem acesso ao estado da cama, se está ocupada ou vazia.<sup>1</sup>

De acordo, com a figura 4.9b, o utilizador para se conectar via Wifi necessita de fazer uma pesquisa do sensor na rede em uso. Para a conexão ser bem sucedida, é necessário que tanto o sensor como o telemóvel estejam ligados na mesma rede Wifi. Quanto maior a força do sinal proveniente do sensor, maior a qualidade de dados vindos do mesmo. Cada vez que se testa o sensor numa cama diferente e, conseqüentemente, com uma pessoa diferente, tem que se efetuar uma nova calibração tanto com a cama vazia como ocupada, sem qualquer tipo de ruído.<sup>1</sup> Nesta atividade 4.9b, ainda existe a possibilidade de se definir o alarme (Atividade 4.9c).<sup>2</sup>

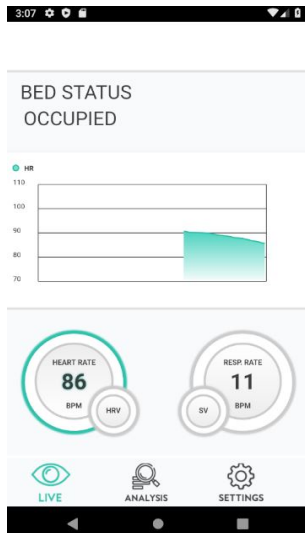
Como ilustrado na figura 4.9d, o gráfico em tempo real representa o estado do sono de 30 em 30 segundos.

Por último, o utilizador tem acesso ao gráfico final do estado do sono durante uma noite representado na figura 4.9e, de acordo com o algoritmo Random Forest implementado no

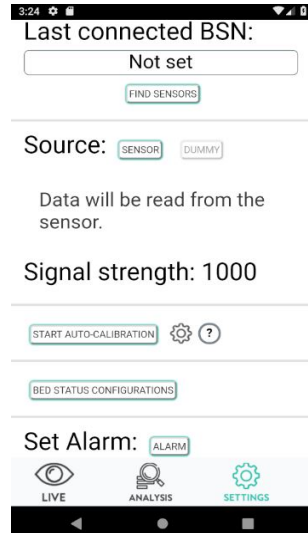
<sup>1</sup>[https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h\\_11h/new-bcg-demo-app-v2-for-android.ashx?la=en-us&cvid=20180329024939647600](https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h_11h/new-bcg-demo-app-v2-for-android.ashx?la=en-us&cvid=20180329024939647600)

<sup>2</sup><https://github.com/annathehybrid/RichardDawkinsAlarmClock>

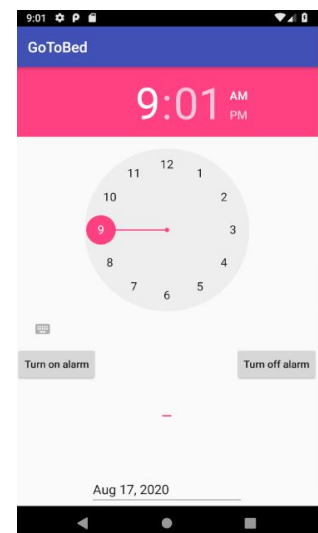
modelo e, por sua vez, usado na predição. Assim, é possível concluir que a aplicação ajuda na qualidade do sono e a acordar a pessoa sempre na fase do sono leve, através da atuação proveniente do dispositivo móvel. Por sua vez, esta aplicação cumpre com todos os requisitos presentes na proposta.



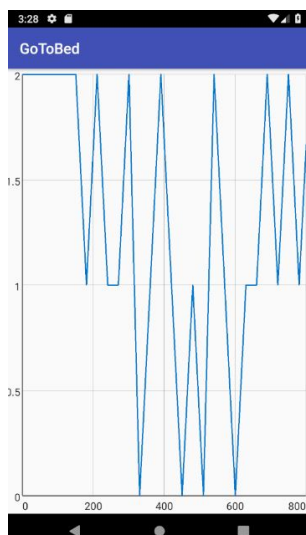
(A) Atividade referente à visualização dos dados do sensor SCA11H BCG em tempo real.



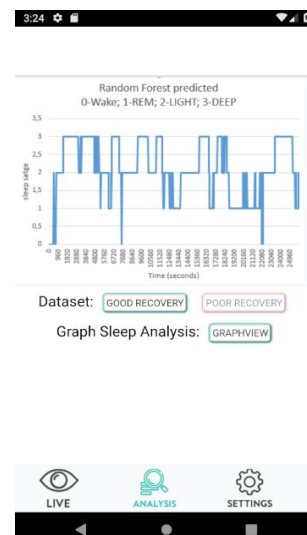
(B) Atividade referente às definições do sensor SCA11H BCG, entre elas, a conexão Wifi, os parâmetros de calibração e o alarme.



(C) Atividade referente à configuração do alarme.<sup>3</sup>



(D) Atividade referente à visualização do estado do sono em tempo real.



(E) Atividade referente à visualização do gráfico final da predição do estado do sono numa noite.

FIGURA 4.9. Atividades referentes à aplicação Android - GoToBed.

<sup>3</sup><https://github.com/annathebrid/RichardDawkinsAlarmClock>





## CAPÍTULO 5

### Conclusões

#### 5.1. Conclusões, Limitações e Trabalho Futuro

Esta dissertação assumiu como objetivo principal a monitorização do sono para obter uma boa qualidade de sono. Contudo, a aplicação desenvolvida, em Android, é um meio para atingir o mesmo. Por sua vez, a monitorização inclui uma estratégia para tentar acordar sempre o utilizador na fase do sono leve.

No capítulo da revisão da Literatura, conclui-se que o PSG é um método intrusivo, caro e complexo na análise do sono. Enquanto que, o BCG é um método que não possui qualquer tipo de mecanismo de carácter invasivo e é a melhor solução para uso doméstico. Também, é possível compreender todas as fases e ciclos do sono REM e NREM (sono leve e profundo). Ao nível de duração, os sonos NREM e REM constituem 80% e 20%, repetidamente, do tempo total gasto no sono. Contudo, o algoritmo de classificação Random Forest e J48 correspondem a abordagens com árvores de decisão. Por outro lado, o algoritmo IBk (KNN) recai no número de vizinhos a usar para definir a classe e o algoritmo PART é o novo método para indução de regras. Por fim, realizou-se um levantamento de requisitos sobre aplicações existentes com tipos de sensores e atuadores internos e externos.

O objetivo concretiza-se usando a plataforma Android Studio e o sensor externo SCA11H. Já a estratégia, é implementada com o apoio da atuação (Som).

Na questão dos testes e resultados, concluiu-se que para cada equação implementada no processo de filtragem existe um valor de constante de filtragem específico. Por sua vez, foram testados e analisados todos os algoritmos de classificação, em estudo, num dado conjunto de testes. Desta avaliação, resultaram os melhores algoritmos de classificação, ao nível de desempenho, entre eles, o Random Forest, o J48, o IBk (KNN) e o PART. Consequentemente, ao nível dos testes de aprendizagem dos algoritmos de classificação referidos anteriormente, obteve-se o Random Forest como melhor algoritmo de classificação para o mesmo conjunto de testes. Devido, a apresentar o melhor valor de instâncias classificadas corretamente e de kappa. De seguida, cria-se o modelo final com base neste algoritmo. Na presença deste modelo, é possível efetuar as predições das várias fases do sono para o BCG e comparar as mesmas com as obtidas pelo PSG, através de matrizes de confusão. Assim, pode-se concluir que o Random Forest é, de facto, o algoritmo de classificação que apresenta um melhor nível de resultados para a realização da monitorização do sono. Consequentemente, a análise funcional e as atividades presentes na aplicação vieram provar, em parte, que a aplicação funciona e cumpre com o objetivo proposto, ao longo desta dissertação.



Existe uma limitação referente a uma quantidade pequena de amostras BCG e PSG (12 noites). A empresa que criou o sensor usado forneceu todos estes dados para conseguir efetuar o estudo e recolher toda esta aprendizagem até se construir o modelo final. Devido ao estado de pandemia que aconteceu durante a parte experimental da tese, não foi possível recolher mais amostras de referência, em meio hospitalar, usando sensores PSG de maior precisão.

Futuras investigações poderiam registar um maior número de dados referentes ao BCG (sensor SCA11H) e PSG. De forma, a efetuar novamente a aprendizagem dos algoritmos de classificação para se obter um nível de desempenho superior. Também, poderiam inserir mais tipos de variáveis (por exemplo, idade, peso, género), por forma, a criar um modelo final mais preciso e fidedigno. Consequentemente, na ótica da aplicação, podia-se implementar um registo histórico referente a cada noite e obter uma análise desse mesmo registo, a 30, 60 ou 90 dias. Seguindo esta linha de raciocínio, além do gráfico com as fases do sono, também se podiam registar os movimentos e o ressonar, durante a noite. Posteriormente, para guardar todos estes registos é necessário a criação de uma base de dados. Por exemplo, poderiam usar o Firebase Realtime Database para este efeito e o Firebase Authentication para questões de autenticação. Também, se podem usar e testar outro tipo de estratégias para acordar a pessoa, em questão. [22]

O objetivo deste estudo consistiu num conhecimento sobre a predição das várias fases do sono, usando os algoritmos de classificação presentes na ferramenta Weka, e na implementação das mesmas na aplicação "GoToBed". Nesta implementação, está disponível toda a análise funcional e, por sua vez, a manipulação dos estados do sono, por forma, ao utilizador acordar sempre na fase do sono mais leve, com o apoio de um som relaxante ou, então, de um som normal do alarme.

Por último, conclui-se que esta aplicação é útil para acordar a pessoa na fase do sono leve e apresenta, como resultado, um acordar mais calmo e relaxante por parte do utilizador.





## Referências Bibliográficas

- [1] J. M. Alonso and A. Bugarín. Expliclas: Automatic generation of explanations in natural language for weka classifiers. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, June 2019.
- [2] Apple. Beddit sleep monitor. Available: "<https://www.beddit.com/>". Accessed: 22.09.2020 [Online].
- [3] A. Azhari, S. Yoshimoto, T. Nezu, H. Iida, H. Ota, Y. Noda, T. Araki, T. Uemura, T. Sekitani, and K. Morii. A patch-type wireless forehead pulse oximeter for spo2 measurement. In *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4, Oct 2017.
- [4] J. Brownlee. How to save your machine learning model and make predictions in weka. Available: "<https://machinelearningmastery.com/save-machine-learning-model-make-predictions-weka/>". Accessed: 30.07.2020 [Online].
- [5] J. Brownlee. How to use classification machine learning algorithms in weka. Available: "<https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/>". Accessed: 30.07.2020 [Online].
- [6] J. Brownlee. How to work through a multi-class classification project in weka. Available: "<https://machinelearningmastery.com/multi-class-classification-tutorial-weka/>". Accessed: 30.07.2020 [Online].
- [7] J. Brownlee. *Machine Learning Mastery With Weka: Analyze Data, Develop Models, and Work Through Projects*. Machine Learning Mastery, 2016.
- [8] Murata Manufacturing Company. Bcg demo app v2 for android platform. Available: "[https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h\\_11h/new-bcg-demo-app-v2-for-android.ashx?la=en-us&cvid=20180329024939647600](https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h_11h/new-bcg-demo-app-v2-for-android.ashx?la=en-us&cvid=20180329024939647600)". Accessed: 1.03.2020 [Online].
- [9] Murata Manufacturing Company. Bed sensor: Basic knowledge and application image of ballistocardiology (bcg). Available: "<https://video.murata.com/en-global/detail/videos/accelerometers/video/5045769962001>". Accessed: 27.08.2020 [Online].
- [10] Murata Manufacturing Company. Contactless bed sensor. Available: "[https://www.murata.com/en-eu/products/sensor/accel/sca10h\\_11h](https://www.murata.com/en-eu/products/sensor/accel/sca10h_11h)". Accessed: 27.08.2020 [Online].
- [11] Murata Manufacturing Company. Product datasheet sca10h. Available: "[https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h\\_11h/product-specification-1322-rev3-sca10h-product-datasheet-eng.ashx?la=en-us&cvid=20161121232536316400](https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h_11h/product-specification-1322-rev3-sca10h-product-datasheet-eng.ashx?la=en-us&cvid=20161121232536316400)". Accessed: 5.10.2020 [Online].
- [12] Murata Manufacturing Company. Sca11h bed sensor installation guide. Available: "[https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h\\_11h/app-2077-rev3-sca11h-installation-guide-eng-id-18685.ashx?la=en-us&cvid=20170413232732051400](https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h_11h/app-2077-rev3-sca11h-installation-guide-eng-id-18685.ashx?la=en-us&cvid=20170413232732051400)". Accessed: 24.03.2020 [Online].
- [13] Murata Manufacturing Company. Sca11h operation modes. Available: "[https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h\\_11h/product-specification-1324-rev1-sca11h-operation-modes-eng-id-15741.ashx?la=en-us&cvid=20160209090049206400](https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h_11h/product-specification-1324-rev1-sca11h-operation-modes-eng-id-15741.ashx?la=en-us&cvid=20160209090049206400)". Accessed: 27.08.2020 [Online].

- [14] Apache Cordova. Apache cordova. Available: "<https://cordova.apache.org/>". Accessed: 30.09.2020 [Online].
- [15] Machine Learning Crash Course. Classification: True vs. false and positive vs. negative. Available: "<https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative>". Accessed: 29.09.2020 [Online].
- [16] S. Das, S. Dahiya, and A. Bharadwaj. An online software for decision tree classification and visualization using c4.5 algorithm (odtc). In *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 962–965, March 2014.
- [17] Android Developers. Download android studio and sdk tools. Available: "<https://developer.android.com/studio>". Accessed: 29.09.2020 [Online].
- [18] Dreem. Dreem 2 - sleep, finally. Available: "<https://dreem.com/>". Accessed: 23.09.2020 [Online].
- [19] Dreem. Dreem 2 - understand your sleep with the best tracker. Available: "<https://dreem.com/understand>". Accessed: 23.09.2020 [Online].
- [20] K. Elekar, M. M. Waghmare, and A. Priyadarshi. Use of rule base data mining algorithm for intrusion detection. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–5, 2015.
- [21] George Ellis. Chapter 9 - filters in control systems. In *Control System Design Guide (Fourth Edition)*, pages 165 – 183. Butterworth-Heinemann, Boston, fourth edition, 2012.
- [22] Firebase. Firebase helps mobile and web app teams succeed. Available: "<https://firebase.google.com/>". Accessed: 30.09.2020 [Online].
- [23] Eclipse Foundation. The community for open innovation and collaboration. Available: "<https://www.eclipse.org/>". Accessed: 30.09.2020 [Online].
- [24] G. Garcia-Molina, S. Vissapragada, A. Mahadevan, R. Goodpaster, B. Riedner, M. Bellesi, and G. Tononi. Probabilistic characterization of sleep architecture: Home based study on healthy volunteers. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2834–2838, 2016.
- [25] A. Gasmi, V. Augusto, P. A. Beaudet, J. Faucheu, C. Morin, X. Serpaggi, and F. VASSEL. Sleep stages classification using cardio-respiratory variables. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1031–1036, Aug 2020.
- [26] N. Saravanan, V. Gayathri. Performance and classification evaluation of j48 algorithm and kendall’s based j48 algorithm (knj48). Available: "[https://www.periyaruniversity.ac.in/ijcii/issue/marnew/2\\_mar\\_18.pdf](https://www.periyaruniversity.ac.in/ijcii/issue/marnew/2_mar_18.pdf)". Accessed: 16.09.2020 [Online].
- [27] Onel Harrison. Machine learning basics with the k-nearest neighbors algorithm. Available: "<https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>". Accessed: 15.09.2020 [Online].
- [28] Hehui Qian and Zhiwei Qiu. Feature selection using c4.5 algorithm for electricity price prediction. In *2014 International Conference on Machine Learning and Cybernetics*, volume 1, pages 175–180, July 2014.
- [29] Java. Java — oracle. Available: "<https://www.java.com/en/>". Accessed: 30.09.2020 [Online].
- [30] Jiang Fuli and Chu Chu. Application of knn improved algorithm in automatic classification of network public proposal cases. In *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 82–86, 2017.
- [31] McHugh M. L. Interrater reliability: the kappa statistic. In *Biochemia medica*, page 276–282, October 2012.
- [32] H. Lan and Y. Pan. A crowdsourcing quality prediction model based on random forests. In *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, pages 315–319, 2019.

- [33] G. Li and J. Zhang. Music personalized recommendation system based on improved knn algorithm. In *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 777–781, 2018.
- [34] G. Louppe. “*Understanding Random Forest From Theory to Practice*”. PhD dissertation, University of Liège - Faculty of Applied Sciences Department of Electrical Engineering & Computer Science, Belgium, 2015.
- [35] Beddit Ltd. Beddit sleep monitor and the science behind it. Available: ”[https://www.beddit.com/static/Beddit\\_Science\\_Leaflet\\_A4\\_HR.pdf](https://www.beddit.com/static/Beddit_Science_Leaflet_A4_HR.pdf)”. Accessed: 22.09.2020 [Online].
- [36] Emfit Ltd. Emfit - sleep tracking with hrv. Available: ”<https://www.emfit.com/why-choose-emfit-for-sleep-analysis>”. Accessed: 22.09.2020 [Online].
- [37] Emfit Ltd. Ferro-electret sensors for oem customers. Available: ”<https://www.emfit.com/copy-of-company>”. Accessed: 21.10.2020 [Online].
- [38] Emfit Ltd. Qs+active - sleep and recovery monitor for the active life. Available: ”<https://www.emfit.com/active-life-sleep-and-recovery>”. Accessed: 22.09.2020 [Online].
- [39] M. M. Mazid, A. B. M. S. Ali, and K. S. Tickle. A comparison between rule based and association rule mining algorithms. In *2009 Third International Conference on Network and System Security*, pages 452–455, 2009.
- [40] Ulf Meriheinä. Bcg measurements in beds. Available: ”[https://www.murata.com/-/media/webrenewal/products/sensor/pdf/flyer/bcg-measurements-in-beds\\_12.11.2019.ashx?la=en-us&cvid=20191129025013000000](https://www.murata.com/-/media/webrenewal/products/sensor/pdf/flyer/bcg-measurements-in-beds_12.11.2019.ashx?la=en-us&cvid=20191129025013000000)”. Accessed: 27.08.2020 [Online].
- [41] M. Nagura, Y. Mitsukura, T. Kishimoto, and M. Mimura. A practical bcg measuring system with bed sensors and algorithm for heartbeat detection. In *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, pages 317–321, 2018.
- [42] U.S. National Library of Medicine National Center for Biotechnology Information. Sleep physiology - sleep disorders and sleep deprivation. Available: ”<https://www.ncbi.nlm.nih.gov/books/NBK19956/>”. Accessed: 13.09.2020 [Online].
- [43] S. Nurmi. “*Nocturnal sleep quality and quantity analysis with ballistocardiography*”. Master’s thesis, School of Electrical Engineering, Espoo, Finland, 2016.
- [44] University of Waikato. Creating arff file - weka wiki. Available: ”[https://waikato.github.io/weka-wiki/formats\\_and\\_processing/creating\\_arff\\_file/](https://waikato.github.io/weka-wiki/formats_and_processing/creating_arff_file/)”. Accessed: 29.07.2020 [Online].
- [45] University of Waikato. Serialization - weka wiki. Available: ”<https://waikato.github.io/weka-wiki/serialization/>”. Accessed: 29.07.2020 [Online].
- [46] University of Waikato. Use weka in your java code - weka wiki. Available: ”[https://waikato.github.io/weka-wiki/use\\_weka\\_in\\_your\\_java\\_code/](https://waikato.github.io/weka-wiki/use_weka_in_your_java_code/)”. Accessed: 29.07.2020 [Online].
- [47] Sami Nurmi, Tarja Saaresranta, Tero Koivisto, Ulf Meriheinä, Lauri Palva. Validation of an accelerometer based bcg method for sleep analysis. Available: ”<https://aaltodoc.aalto.fi/bitstream/handle/123456789/21176/isbn9789526068428.pdf?sequence=1&isAllowed=y>”. Accessed: 28.08.2020 [Online].
- [48] S. R. Pandi-Perumal, D. Warren Spence, and Ahmed S. BaHammam. *Polysomnography: An Overview*, pages 29–42. Springer New York, New York, NY, 2014.
- [49] Pillow. Pillow automatic sleep tracker for the apple watch, iphone and ipad. Available: ”<https://neybox.com/pillow-sleep-tracker-en>”. Accessed: 23.09.2020 [Online].
- [50] Pillow. Pillow automatic sleep tracker on the app store. Available: ”<https://apps.apple.com/app/pillow-sleep-cycle-alarm-clock/id878691772>”. Accessed: 23.09.2020 [Online].
- [51] REM-Fit. Zeeq - rem-fit. Available: ”<https://remfit.com/pages/zeeq>”. Accessed: 23.09.2020 [Online].

- [52] Jiri Richter. Sleep - sleep as android. Available: "<https://docs.sleep.urbandroid.org/>". Accessed: 24.09.2020 [Online].
- [53] Jiri Richter. Sleep as android - sleep cycle smart alarm - apps on google play. Available: "<https://play.google.com/store/apps/details?id=com.urbandroid.sleep>". Accessed: 24.09.2020 [Online].
- [54] Sumit Saha. What is the c4.5 algorithm and how does it work? Available: "<https://towardsdatascience.com/what-is-the-c4-5-algorithm-and-how-does-it-work-2b971a9e7db0>". Accessed: 16.09.2020 [Online].
- [55] Tavish Srivastava. Introduction to k-nearest neighbors: A powerful machine learning algorithm (with implementation in python & r). Available: "<https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>". Accessed: 15.09.2020 [Online].
- [56] Expert System. What is machine learning? a definition. Available: "<https://expertsystem.com/machine-learning-definition/>". Accessed: 12.09.2020 [Online].
- [57] A. Tataraidze, L. Korostovtseva, L. Anishchenko, M. Bochkarev, and Y. Sviryaev. Sleep architecture measurement based on cardiorespiratory parameters. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3478–3481, 2016.
- [58] Marc T. Thompson. Chapter 14 - analog low-pass filters. In *Intuitive Analog Circuit Design (Second Edition)*, pages 531 – 583. Newnes, Boston, second edition, 2014.
- [59] Neelam Tyagi. Random forest algorithms in machine learning: A comprehensive study. Available: "<https://medium.com/analytics-steps/random-forest-algorithms-in-machine-learning-a-comprehensive-study-de5168b285ba>". Accessed: 14.09.2020 [Online].
- [60] Xiaoran Wang. Java code examples for weka.classifiers.trees.j48. Available: "<https://www.programcreek.com/java-api-examples/?api=weka.classifiers.trees.J48>". Accessed: 29.07.2020 [Online].
- [61] Eibe Frank, Ian H Witten. Generating accurate rule sets without global optimization. Available: "<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.143.8073&rep=rep1&type=pdf>". Accessed: 16.09.2020 [Online].
- [62] R. Yi, M. Enayati, J. M. Keller, M. Popescu, and M. Skubic. Non-invasive in-home sleep stage classification using a ballistocardiography bed sensor. In *2019 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pages 1–4, May 2019.
- [63] Yi Tan and Guo-Ji Zhang. The application of machine learning algorithm in underwriting process. In *2005 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3523–3527 Vol. 6, 2005.
- [64] A. Yilmaz, M. A. Mutlu, M. Günay, and A. U. Demir. Analysis of portable apnea devices: Comparison with polysomnographic (psg) records. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2019.

APÊNDICE A

**Anexos**



# Non-invasive monitoring with Ballistocardiographic sensors for sleep management

Bernardo Silva

*Instituto de Telecomunicações*  
*Instituto Universitário de Lisboa (ISCTE-IUL)*  
Lisbon, Portugal  
bernardo\_silva@iscte-iul.pt

Rui Neto Marinheiro

*Instituto de Telecomunicações*  
*Instituto Universitário de Lisboa (ISCTE-IUL)*  
Lisbon, Portugal  
rui.marinheiro@iscte-iul.pt

**Abstract**—Sleep has an important impact on people’s daily lives. A successful methodology for monitoring sleep is Polysomnography (PSG). This is an accurate and reliable approach but, unfortunately, very invasive. PSG uses expensive sensors that must be positioned by experts, what, in practice, makes its adoption only viable in hospital setups. Therefore, there is a demand for better non-invasive alternatives, such as Ballistocardiography (BCG). BCG uses cheaper sensors, easy to install and ideal for domestic use. This allows its integration in solutions that manage sleep, using mobile apps not only for presenting valuable information to users but may also for acting on the environment, through actuators, such as sound.

This work uses this principle to help users to wake up smoothly. Sleep monitoring is performed with Murata SCA11H BCG external sensors. Low-pass filters have been implemented, using a sliding exponential average, for all metrics. The Random Forest algorithm was then selected for sleep phase classification, that presented the best performance when using the Weka exploration tool for learning methods. With the implemented model, it has been proved that four sleep phases are predicted. It was then possible to define a strategy for avoiding waking up alarms to be fired during deep sleep. It consists on the analysis 15 minutes prior to the alarm and, when deep sleep is detected, a relaxing sound is played.

This work demonstrated that non-invasive sleep monitoring can be used to actuate on, and improve, the user environment, in a home setup with cheap sensors.

**Index Terms**—Ballistocardiography, Sleep Monitoring, Sleep prediction.

## I. INTRODUCTION

The work explored here focuses on rest management, through sleep monitoring. In particular it contributes to solve the problem of reliable sleep monitoring and actuation at home, without using invasive methods.

In short, the most used methodologies, in this area, are PSG and BCG. The PSG monitoring method is the most reliable but it is invasive, expensive and requires a specialist to identify the stages of sleep. Therefore, BCG should be the method to be used, for non invasive setups that can be easily deployed at home, with no need for hospital support or specialists in the identification of sleep phases. [20] [19]

BCG is the most recently developed method and it is somehow supported on PSG. It reads different metrics, by using sensors such as accelerometers and gyroscopes. In order to obtain enough precision for a reliable sleep monitoring, this

metrics must be compared with PSG classification carried out by specialists.

The main objective for the work presented here is to monitor sleep through external non-invasive sensors and to act accordingly on the environment.

An integration proposal and a mobile application “Go-ToBed”, presented here, has been developed as a means to achieve this goal.

Section II surveys diferentes sleep monitoring approaches and some known applications in the area.

Section III presents our integration proposal and application. This uses an external BCG sensor, selected because of its price and easy handling by end users, both in terms of configuration and calibration. Low pass filters are proposed for the data obtained from those sensors. Only then a classification of sleep stages and actuation on the environment is possible. In order to avoid a startle response, when on deep sleep, the actuation aims to wake up users during the light sleep phase. For this, a smooth sound is played prior to the alarm.

Section IV presents configuration options for the filters and a performance comparison of different classification learning methods and their use on sleep prediction on our proposal.

Conclusions are presented in Section V

## II. LITERATURE REVISION

### A. PSG versus BCG

PSG is an intrusive, expensive and complex method for analyzing sleep. It is only suitable for continuous short-term measurements. Sleep monitoring and analysis are performed by highly specialized technicians and doctors in a laboratory setting. This measurement requires the use of several types of sensors. Thus, it has become important to find new ways to monitor sleep in a more effective, inexpensive and continuous way in the long run. Consequently, methods based on Ballistocardiography (BCG) have emerged, which have an increasing potential in the analysis of sleep. The PSG features 18 channels for measuring various types of signal. Among them, 6 electrodes to measure brain activity with the support of a Electroencephalography (EEG) procedure, 2 electrodes to measure eye movements using the Electro-oculography

(EOG) method, 2 electrodes to measure the tension of the mandibular musculature through the Electromyography (EMG), 3 electrodes to measure cardiac function with the Electrocardiography (ECG) and, finally, chest and abdomen belts to measure respiratory effort. [26] [22]

BCG (Figura 1) is a method that does not have any type of intrusive mechanism. It has the ability to measure the mechanical pumping of the heart and provide the time and relative stroke volume (SV) of each heartbeat. It is an excellent tool to measure recovery and sleep quality. A good night's sleep is important in the functioning of the heart or blood vessels, in mental well-being and in preventing some types of cancer. It help athletes to achieve good results, in terms of training and competition. need a good quality of sleep. Heart rate and breathing parameters also measure the condition of the autonomous nervous system and indicate many pathological conditions. Among them, the change in stroke volume as a measure of the condition of body fluid. [20] [19]

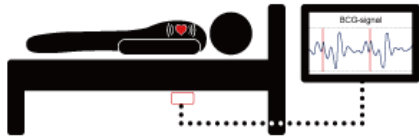


Fig. 1. BCG sleep monitoring process. [6]

### B. Non-invasive monitoring & sleep management applications

We have identified related some applications in this area: Beddit [1], Sleep as Android [25], Pillow [23], Emfit QS [17], ZEEQ Smart Pillow by Rem-Fit [24], DREEM 2 [10] and compared the with this proposal ("GoToBed").

The most common way of acting is through the sound coming from the mobile device or the external sensor (ZEEQ), in order to wake the person. The Sleep as Android application features more types of actuators. Among them, captcha and intelligent light. The captcha aims to force the user to get out of bed in order to turn off the alarm.

Analyzing Tables I and II, it can be concluded that for applications that have internal sensors, the cost is free (Pillow and Sleep As Android). Where as, for applications that use external sensors, they already have an associated cost. That is, the applications Beddit, Emfit QS, ZEEQ Smart Pillow, DREEM 2 and "GoToBed" (proposal present in chapter 3) have a cost of 149, 299.98, 51.71, 428.10 and 168.45 euros, respectively. Thus, ZEEQ is the cheapest App and DREEM 2 the most expensive. This makes perfect sense, taking into account the methodologies associated with each of them. The BCG methodology is present in the Beddit, Emfit QS and "GoToBed" applications, but only the later supports WiFi. In turn, the SONAR and EEG methodologies are used respectively by App Sleep as Android and DREEM 2. Also,

there are differences, inherent to each application, between the way in which the different types of data and audio are transferred and stored. Finally, as can be seen in the table I, applications represent different operating systems.

TABLE I  
COMPARISON BETWEEN ALL APPLICATIONS REFERENCED AND ANALYZED THROUGHOUT THIS ARTICLE.

Applications	Operating system	Cost [euros]	Methodologies	Data and audio transfer
Beddit	IOS	149,00	BCG	Bluetooth
Sleep as Android	Android	0	SONAR	-
Pillow	IOS	0	-	-
Emfit QS	Web	299,98	BCG	Wifi Mobile data
ZEEQ Smart Pillow by Rem-Fit	Android IOS	51,71	-	Bluetooth
DREEM 2	Android IOS	428,10	EEG	Bluetooth Wifi
This Proposal "GoToBed"	Android	168,45	BCG	Wifi

TABLE II  
COMPARISON BETWEEN SENSORS PRESENT IN ALL APPLICATIONS.

Applications	Sensor type	Sensors used
Beddit	External	Piezoelectric force sensor , capacitive touch, humidity and temperature
Sleep as Android	Internal (Device)	Ultrasonic sensor (SONAR), motion/position sensor (gyroscope and accelerometer) and passive infrared sensor
Pillow	Internal (Device)	Motion/position sensor (gyroscope and accelerometer)
Emfit QS	External	Ferroelectric sensors (Sheet quasi-piezoelectric, sensors band and cable)
ZEEQ Smart Pillow by Rem-Fit	External	Motion sensor (3-axis gyroscope)
DREEM 2	External	6 EEG sensors, accelerometer, pulse oximeter and sound level meter
This Proposal "GoToBed"	External	SCA10H sensor (accelerometer, gyroscope)

Table II analyzes the maximum possible distance between the sensor and the user. Applications with internal sensors have a working distance up to 100cm. On the other hand, the applications with BCG, that use external sensors and feature more precision, focus on smaller distances: 30cm for Beddit and Emfit, and 20 for "GoToBed" cm. The distance for the ZEEQ and DREEM 2 applications is not critical, since they are using respectively cushion and headband integrated sensor.

At the level of used sensors, in the Table II, both Beddit and Emfit QS have a sensor with a thin and soft structure. The Sleep as Android and Pillow applications fall under the same principles, that is, motion/position sensors (gyroscope and accelerometer). However, what distinguishes them is an ultrasonic sensor. On the other hand, the DREEM 2 application features six EEG sensors, which is why it is more expensive, and also includes an accelerometer, a pulse oximeter and a sound level meter. A pulse oximeter was developed to measure the heart rate and oxygen saturation of the blood. [2] Finally, the proposal "GoToBed" features a Murata SCA10H chip with an accelerometer and a gyroscope sensor.

### III. MONITORING AND ACTUATION PROPOSAL

In this section the proposal integration and implementation are detailed and all options are justified for the application development including the external sensor to be used.

In Figure 2, the functional sequence diagram of the proposed integration and application is presented. In a first instance, the sensor is connected to the application via wifi. However, if the sensor has not been previously calibrated or configured, it is necessary to perform its calibration and configuration. Then, the phase of defining the desired action for this application is continued, such as, for example, an alarm for a smoother wake up. And, consequently, the application starts receiving, every second, the output data from the sensor. All of this data is subject to a filtering process. In turn, a ARFF file is created with all the data previously filtered. With this file, it is possible to make the prediction of the phases of sleep through the classification algorithm.

Subsequently, a graph is created with all phases of sleep. In this work, with this information, if the user is in deep sleep, 15 minutes before the alarm triggeres, a relaxing music is played in advance. This type of music will last until the expected time to wake up or until the user is in a different state other than deep sleep. Otherwise, a normal alarm music starts, only at the time set in the alarm. Finally, if the person under study woke up in a state of deep sleep, the night of rest is classified as a bad recovery. Whereas, if you have not woken up in the state of deep sleep, the night of rest is classified as a good recovery.

This is done because there are different consequences at different sleep states. That is, for the deep sleep state a more relaxing sound is used, while, for all other states of sleep (light, REM and awake), the used sound is more intense. In short, this type of action aims to a calmer wake up in the deep sleep phase and to wake up more firmly in the other phases.

Figure 3 depicts the implementation of the proposal for the GoToBed application. The first phase is the monitoring, where the collection of data is performed through sensors.

The selection of the sensors to be used is critical and must present, necessarily, the following requirements: a low price, free access to data, a number of sufficient and relevant metrics to achieve the objectives, and ease of use. Thus, we

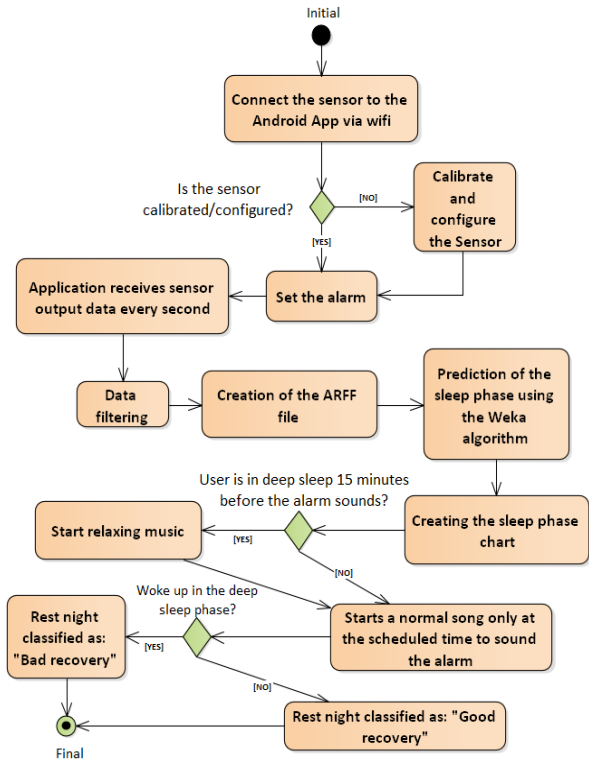


Fig. 2. Proposed functional sequence diagram for the application under study.

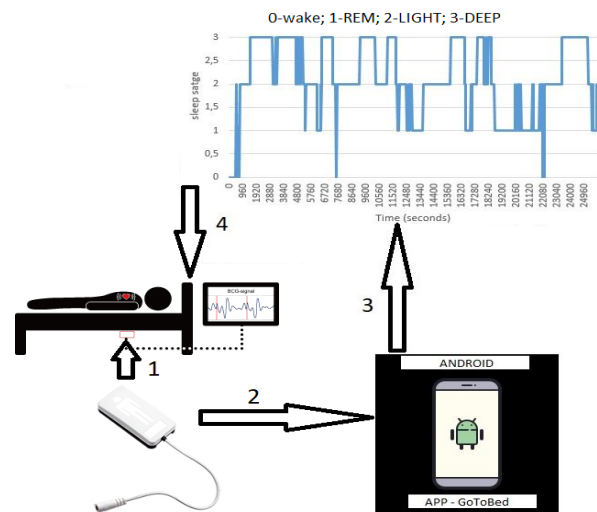


Fig. 3. Implementation of the proposal for the GoToBed application.

have opted for the Murata SCA11H chip that have sensors, that implement the BCG methodology, outputting several metrics for accelerometer and gyroscope sensors, and presents easy procedures that facilitate its installation, in particular its calibration, configuration and positioning. [7] [8] [5] With this sensor, we obtain all this type of information, for example, Heart rate (HR), Heart rate variability (HRV), Respiration rate (RR), Stroke volume (SV) and Signal strength (SS), which will be explained later.

Then, the second phase involves connecting the application with the sensors via WiFi, in order to obtain the output data from the BCG sensors, every second. The implemented application "GoToBed", installed on an Android mobile phone, was implemented using the Android Studio development tool.

In turn, in the third phase, the application filters all these data and makes sleep predictions every 30 seconds.

Finally, in the fourth step, the main objective of the application is achieved. This application acts in order to wake the user up in the lightest sleep phase. For this, an analysis is carried out 15 minutes before the scheduled time for the alarm. If this analysis results in the deep sleep phase, a relaxing music begins.

The use of data from the BCG SCA11H sensors is particularly critical for this proposal. Raw data is very noisy, particularly for higher frequencies, and a proper data processing is required. This has been achieved by using several filters, supported by the Equation 1. This equation represents a low-pass filter implementation, using a sliding exponential average. The  $k$  parameter is a filtering constant (varies between 0 and 1),  $x(t)$  is the original value, that is, the current signal sample,  $y(t-1)$  is the previously filtered value and, finally,  $y(t)$  is the filtered value at time  $t$ . The higher the  $k$ , the more forgetfulness the filter is, that is, higher frequencies are more relevant or the sensor is more reactive. [21]

$$y(t) = y(t-1) * (1 - k) + x(t) * k \quad (1)$$

HRFILT is calculated based on Equation 1 and measures heart rate (HR) filtering. The  $x(t)$  matches HR(t) as the original value and  $y(t-1)$  matches HRFILT(t-1) as the filtered value in the previous instant, as depicted to Equation 2. All remaining metrics have a similar treatment.

$$HRFILT(t) = HRFILT(t-1) * (1 - k) + HR(t) * k \quad (2)$$

HRVFILT is calculated using the relationship between the HF and LF components. To reduce the modulation caused by breathing, it is necessary to extend the HRVFILT with the SVV. [21] When multiplying the HFHRV(t) by 10, values were obtained around the value defined as the initial constant (equal to 8), according to Equation 3.

$$HRVFILT(t) = \frac{HFHRV(t) * 10}{LFHRV(t) * SVV(t)} \quad (3)$$

The choice for all initial values must be done automatically for each user of the application. That is, the filtering process

doesn't start immediately at boot time, that is at instance zero seconds ( $t = 0s$ ). The monitoring process waits for a configured time (pre-defined for 5 minutes), so that it is possible to define the initial values for the person under study.

To better describe the processing of the various signals, figure 4 shows the dependencies between the expressions present in the data filtering. The expressions marked in green correspond to the final values used in the calculation of the sleep phase (HRFILT, HRVFILT, dHRV, RRV, SVV and SSFILT). In turn, the expression HRFILT depends on the value of HR, the expression HFHRV depends on HRV, the expression RRFILT depends on the RR, the expression SVFILT depends on the SV, the expression SSFILT depends on the SS. The expression LFHRV depends on HRFILT and HR, the expression RRV depends on RRFILT and RR, the expression SVV depends on SVFILT and SV. HRVFILT, on the other hand, depends on LFHRV, HFHRV and SVV and, finally, dHRV depends on HRVFILT.

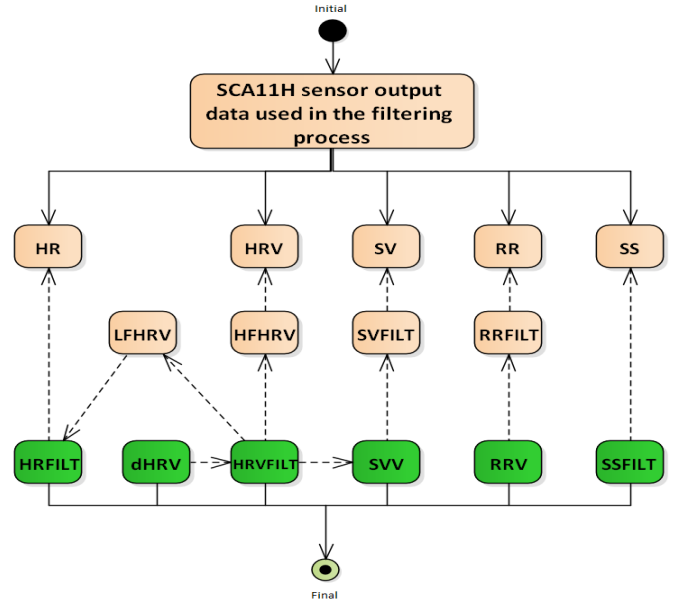


Fig. 4. Dependencies between the expressions present in the data filtering.

#### IV. TESTS AND RESULTS

The choice of the filtering constant ( $k$ ) is a very important process throughout this study. It's main objective is to obtain the  $k$  values for the filtering process. This constant may have a value between 0 and 1. All expressions involved in the filtering process have gone through this decision phase. That is, they were all subjected to similar tests with the same  $k$  values. In this case, the implied expression refers to HFHRV. Base 2 powers are used in the filter constants. These have the advantage of faster processing and less energy consumption, due to the fact that division operations are not used, but rather binary shifts made in just one clock cycle on the processor. In this case, the longer the HFHRV signal period, the lower the sampling frequency.

The need for filtering arises from the fact that we need to eliminate noise from signals. This is required in order to remove unwanted information from signals and to extract relevant information. In turn, the main objective is a signal with the minimum of noise and the maximum of relevant information. After this initial testing phase, for each signals subject to the filtering process, the best value for the filtering constant was obtained. In the expressions for LFHRV, HFHRV, SVV, dHRV, RRFILT, RRV and SSFILT the adopted value was  $\frac{1}{1024}$  and for the filtering expressions for HRFILT and SVFILT the value was  $\frac{1}{256}$ .

During evaluation, in this work we have also performed several tests that compare several classification algorithm, in order select the ones with more potential for sleep classification. Performance tests carried out for each classification algorithm under study, have used a set of test data. For this analyses, Murata Electronics (Finland) has provided sleep data, obtained with twelve people, women and men, aged between 24 and 46 years old, over twelve nights. These data consist of the SCA11H sensor output signals (BCG) and the sleep state reference value (PSG), taken every 30 seconds, for all those 12 nights. This has then been used to performance data analysis and to test the performance of numerous predictive models, that use several learning technics. For this, the Weka Experiment Environment tool [4] [3] has been used, with the aim to obtain the four best classification algorithms, that will be subsequently implemented in our application.

The classification algorithms under study were: rules.ZeroR, bayes.NaiveBayes, functions.Logistic, functions.SMO, lazy.IBk, rules.PART, trees.REPTree, trees.J48, trees.RandomForest. When analyzing Table III, it possible to conclude that the best classification algorithms, at the performance level, are as follows: trees.RandomForest [16] [14], lazy.IBk [15] [13], trees.J48 [9] [12] and rules.PART [11] [18].

TABLE III  
PERFORMANCE FOR DIFFERENT CLASSIFICATION ALGORITHMS.

Algorithms	Performance (%)
rules.ZeroR	44.35
bayes.NaiveBayes	51.02
functions.Logistic	55.57
functions.SMO	52.17
<b>lazy.IBk</b>	<b>90.29</b>
<b>rules.PART</b>	<b>85.47</b>
trees.REPTree	81.54
<b>trees.J48</b>	<b>86.42</b>
<b>trees.RandomForest</b>	<b>91.81</b>

We have obtained figuras, like the one shown in Figure 5, to the analyse the best nights resulting from the study for the selected four algorithms/models, among them, trees.RandomForest, lazy.IBk, trees.J48 and rules.PART. These confusion matrices are composed of the fundamental truth, that is, the sleep states referring to the PSG (columns) and the Machine learning, that is, the prediction values of the sleep phases referring to the BCG (lines). Thus, BCG and PSG values are compared using a confusion matrix. The

perfect confusion matrix will have values greater than zero only in cells that form a diagonal (figure 5). Analyzing Figure 5, the highest values between columns and rows are in the desired positions for a great performance. It is concluded that the result of the confusion matrix regarding the tree algorithm RandomForest (Figure 5) presents the best performance. That is, the highest values for each column are in the correct cells (0: 0 = 27; 1: 1 = 85; 2: 2 = 206; 3: 3 = 192) forming a expected diagonal as shown in orange.

	0	1	2	3
0	27	22	37	19
1	0	85	69	0
2	2	63	206	102
3	0	0	40	192

Fig. 5. Confusion matrix between the night sleep data PSG14 and the BCG prediction for the Weka trees.RandomForest algorithm.

We have also obtained figures, like the one shown in Figure 6, to compare predictions with reference data, in here comparing the sleep phase indicated by PSG and with that obtained with the prediction models using the BCG filtered signals. Figure 6 presents the sleep data from the PSG night 'PSG14' and the BCG prediction from the final model created with the support of the Weka trees.RandomForest classifier algorithm. Again, when analyzing this figure, it appears, as expected, that there is a great similarity in the transitions between the PSG signal and the BCG prediction signal.

Thus, the algorithm chosen to be implemented in the application is Random Forest. Finally, it is concluded that this prediction is effective for non-medical applications, in order to implemente actions on the environment, to improve sleep quality.

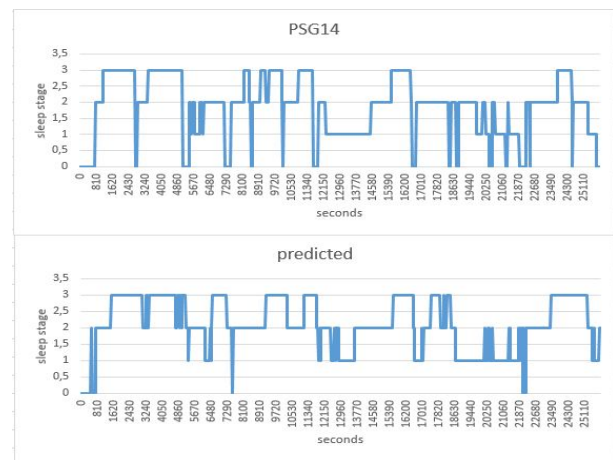


Fig. 6. Graphs between PSG14 night sleep state data and BCG prediction for the Random Forest algorithm.

## V. CONCLUSIONS

This work has as its main goal the monitoring of sleep in order to improve its quality. This monitoring includes a



strategy that tries to wake up users in the light sleep phase. The developed application for Android mobile phone has been a mean to achieve this goal.

It has been concluded that PSG monitoring is an intrusive, expensive and complex method in the analysis of sleep. While, BCG is a method that does not have any type of invasive mechanism and is the best solution for home use.

Regarding signal filtering process, a sliding exponential average has been used with an accessed specific constant value for each metric.

Subsequently, several classification algorithms were tested and analysed with a given reference data set. This evaluation resulted in the best classification algorithms, in terms of performance, Random Forest, J48, IBk (KNN), and PART. From these mentioned learning classification algorithms, Random Forest was the one to present the best values. The final model has created based on this algorithm. In the presence of this model, it is possible to make the predictions of the various sleep phases using BCG data and compare them with those indicated by PSG, using confusion matrices. With this, it can also be concluded that the Random Forest is, in fact, the classification algorithm that presents a better level of results for sleep monitoring.

With the prediction of the various phases of sleep, using the best classification algorithm in the implementation in the "GoToBed" application, it was possible to manipulate sleep states, so that the user wakes up in the lightest sleep phase, by using a relaxing sound. The functional analysis and the actions present in the application came to prove, that the application works and fulfills the proposed objective.

We can say that this application is useful to wake a person in the light phase of sleep and it makes possible, as a result, a calmer and more relaxing waking up experience by the user.

## REFERENCES

- [1] Apple. Beddit sleep monitor. Available: "https://www.beddit.com/". Accessed: 22.09.2020 [Online].
- [2] A. Azhari, S. Yoshimoto, T. Nezu, H. Iida, H. Ota, Y. Noda, T. Araki, T. Uemura, T. Sekitani, and K. Morii. A patch-type wireless forehead pulse oximeter for spo2 measurement. In *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4, Oct 2017.
- [3] J. Brownlee. How to work through a multi-class classification project in weka. Available: "https://machinelearningmastery.com/multi-class-classification-tutorial-weka/". Accessed: 30.07.2020 [Online].
- [4] J. Brownlee. *Machine Learning Mastery With Weka: Analyze Data, Develop Models, and Work Through Projects*. Machine Learning Mastery, 2016.
- [5] Murata Manufacturing Company. Bed sensor: Basic knowledge and application image of ballistocardiography (bcg). Available: "https://video.murata.com/en-global/detail/videos/accelerometers/video/5045769962001". Accessed: 27.08.2020 [Online].
- [6] Murata Manufacturing Company. Contactless bed sensor. Available: "https://www.murata.com/en-eu/products/sensor/accel/sca10h\_11h". Accessed: 27.08.2020 [Online].
- [7] Murata Manufacturing Company. Sca11h bed sensor installation guide. Available: "https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h\_11h/app-2077-rev3-sca11h-installation-guide-eng-id-18685.ashx?la=en-us&cvid=20170413232732051400". Accessed: 24.03.2020 [Online].
- [8] Murata Manufacturing Company. Sca11h operation modes. Available: "https://www.murata.com/-/media/webrenewal/products/sensor/accel/sca10h\_11h/product-specification-1324-rev1-sca11h-operation-modes-eng-id-15741.ashx?la=en-us&cvid=20160209090049206400". Accessed: 27.08.2020 [Online].
- [9] S. Das, S. Dahiya, and A. Bharadwaj. An online software for decision tree classification and visualization using c4.5 algorithm (odt). In *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 962–965, March 2014.
- [10] Dreem. Dreem 2 - sleep, finally. Available: "https://dreem.com/". Accessed: 23.09.2020 [Online].
- [11] K. Elekar, M. M. Waghmare, and A. Priyadarshi. Use of rule base data mining algorithm for intrusion detection. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–5, 2015.
- [12] Huihui Qian and Zhiwei Qiu. Feature selection using c4.5 algorithm for electricity price prediction. In *2014 International Conference on Machine Learning and Cybernetics*, volume 1, pages 175–180, July 2014.
- [13] Jiang Fuli and Chu Chu. Application of knn improved algorithm in automatic classification of network public proposal cases. In *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 82–86, 2017.
- [14] H. Lan and Y. Pan. A crowdsourcing quality prediction model based on random forests. In *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, pages 315–319, 2019.
- [15] G. Li and J. Zhang. Music personalized recommendation system based on improved knn algorithm. In *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 777–781, 2018.
- [16] G. Louppe. "Understanding Random Forest From Theory to Practice". PhD dissertation, University of Liège - Faculty of Applied Sciences Department of Electrical Engineering & Computer Science, Belgium, 2015.
- [17] Emfit Ltd. Emfit - sleep tracking with hrv. Available: "https://www.emfit.com/why-choose-emfit-for-sleep-analysis". Accessed: 22.09.2020 [Online].
- [18] M. M. Mazid, A. B. M. S. Ali, and K. S. Tickle. A comparison between rule based and association rule mining algorithms. In *2009 Third International Conference on Network and System Security*, pages 452–455, 2009.
- [19] Ulf Meriheinä. Bcg measurements in beds. Available: "https://www.murata.com/-/media/webrenewal/products/sensor/pdf/flyer/bcg-measurements-in-beds\_12\_11\_2019.ashx?la=en-us&cvid=20191129025013000000". Accessed: 27.08.2020 [Online].
- [20] M. Nagura, Y. Mitsukura, T. Kishimoto, and M. Mimura. A practical bcg measuring system with bed sensors and algorithm for heartbeat detection. In *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, pages 317–321, 2018.
- [21] S. Nurmi. "Nocturnal sleep quality and quantity analysis with ballistocardiography". Master's thesis, School of Electrical Engineering, Espoo, Finland, 2016.
- [22] Sami Nurmi, Tarja Saaresranta, Tero Koivisto, Ulf Meriheinä, Lauri Palva. Validation of an accelerometer based bcg method for sleep analysis. Available: "https://aaltodoc.aalto.fi/bitstream/handle/123456789/21176/isbn9789526068428.pdf?sequence=1&isAllowed=y". Accessed: 28.08.2020 [Online].
- [23] Pillow. Pillow automatic sleep tracker for the apple watch, iphone and ipad. Available: "https://neybox.com/pillow-sleep-tracker-en". Accessed: 23.09.2020 [Online].
- [24] REM-Fit. Zeeq - rem-fit. Available: "https://remfit.com/pages/zeeq". Accessed: 23.09.2020 [Online].
- [25] Jiri Richter. Sleep - sleep as android. Available: "https://docs.sleep.urbandroid.org/". Accessed: 24.09.2020 [Online].
- [26] A. Yilmaz, M. A. Mutlu, M. Günay, and A. U. Demir. Analysis of portable apnea devices: Comparison with polysomnographic (psg) records. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2019.