



Universidade do Minho
Escola de Engenharia

Inteligência Ambiente: Tecnologias e Aplicações

Sistema de Sensorização

Mestrado Integrado em Engenharia Informática

1º Semestre

2017-2018

Grupo 8

A75655 - Daniel Camelo Rodrigues

A74219 - Hugo Alves Carvalho

A74702 - José Manuel Gonçalves Leitão da Cunha

A74260 - Luís Miguel da Cunha Lima

3 de Dezembro de 2017

Braga

Resumo

Este documento relata o trabalho prático desenvolvido no âmbito da unidade curricular de **Inteligência Ambiente: Tecnologias e Aplicações**, tendo como tema principal a monitorização do batimento cardíaco, horas de sono e contabilização de passos.

Conteúdo

1	Introdução	5
2	Contextualização do Problema	6
3	Análise dos Dados	7
4	Métricas	8
4.1	Batimento Cardíaco	8
4.2	Controlo de Sono	9
4.3	Contabilização de Passos	10
4.4	Correlação de Métricas	11
4.4.1	Batimentos Cardíacos - Sono	11
4.4.2	Passos - Sono	12
5	Interface	13
6	Análise de Resultados	17
7	Conclusões e Trabalho Futuro	19

Lista de Figuras

1	Ambientes Inteligentes - Exemplos de Aplicações	5
2	Impacto de <i>Wearable Devices</i> [3]	6
3	Janela inicial para carregamento de ficheiros <i>csv</i>	13
4	Janela <i>Heartbeat</i> inicial	13
5	Janela com <i>Heartbeats</i> em intervalos de 1 dia	14
6	Janela de Análise do Sono	14
7	Janela de Análise dos Passos	15
8	Janela de Análise da correlação Batimentos Cardíacos - Sono	15
9	Janela de Análise da correlação Passos - Sono	16
10	Valores para Batimento Cardíaco - Homens	17
11	Valores para Batimento Cardíaco - Mulheres	17

1 Introdução

O conceito de Inteligência Ambiente, de acordo com a ISTAG (*IST Advisory Group of EC*), consiste numa nova forma de abordar o meio que nos rodeia, onde dispositivos com poder computacional estão espalhados por todos os lugares, incluídos eventualmente em objetos, permitindo que o ser humano possa interagir no mundo real de uma forma inteligente e discreta [1]. Assim, face à capacidade que os ambientes inteligentes apresentam em responder a diferentes tipos de comportamentos de forma personalizada, estes permitem auxiliar as pessoas nas mais variadas tarefas do seu dia-a-dia.

Cada vez mais, estes tipos de sistemas estão a ser inseridos num maior número de diferentes contextos. Atualmente, é possível encontrar ambientes inteligentes associados a áreas como medicina, física ou agricultura.

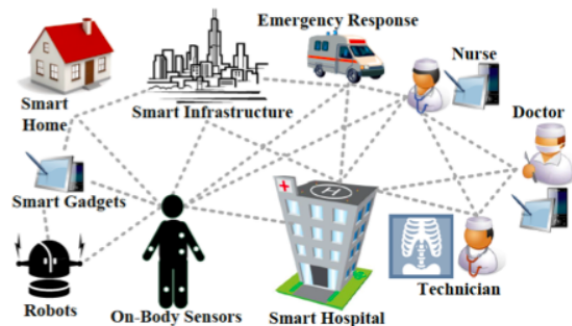


Figura 1: Ambientes Inteligentes - Exemplos de Aplicações

Como podemos verificar na figura acima, o conceito de ambiente inteligentes está diretamente associado à existência de sensores. Estes revelam a capacidade de captar informação de um determinado meio, de modo a realizar ações sobre o ambiente.

Deste modo, através de sistemas de sensorização, todo o quotidiano de um indivíduo pode ser facilitado, uma vez que estes apresentam a capacidade de comunicação e interação. Por outro lado, torna-se possível monitorizar um conjunto de ações que sem a ajuda deste sistemas, não seriam possíveis.

2 Contextualização do Problema

No âmbito da Unidade Curricular de Inteligência Ambiente: Tecnologias e Aplicações, foi proposto aos alunos a concepção e desenvolvimento de um sistema inteligente para a monitorização de sensores de captura de dados.

Perante os vários temas sugeridos, o grupo optou por realizar o projeto associado à monitorização de dados relativos a batimentos cardíacos, horas de sono e contabilização de passos. Esta escolha é facilmente justificada se considerarmos o notável impacto que os *wearable devices* estão a causar na sociedade, atraindo uma variedade alargada de utilizadores por todo o mundo.

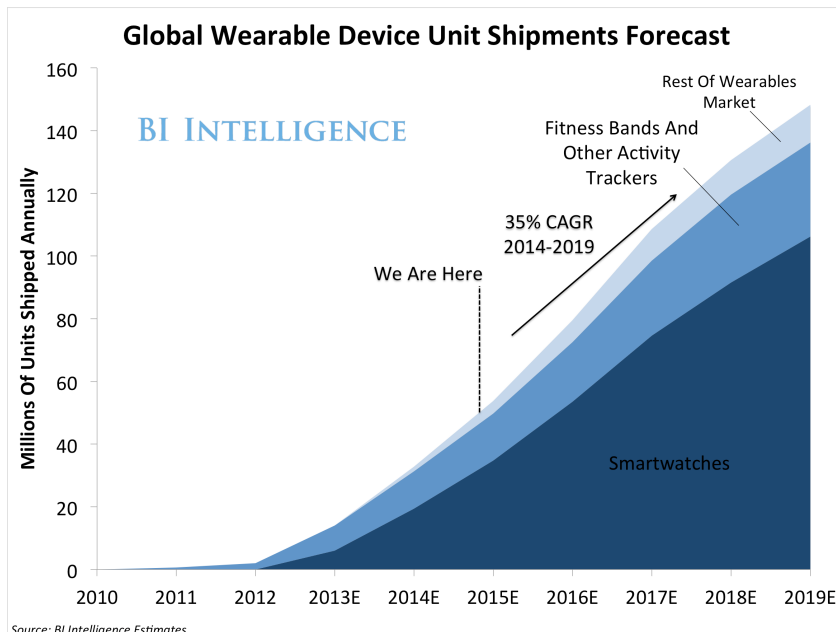


Figura 2: Impacto de *Wearable Devices* [3]

O objetivo inicial seria compreender os protocolos de comunicação e gestão de informação, referentes à fitness band disponibilizada pela equipa docente – "Withings Pulse O2". No entanto, devido à fraca documentação associada a este sensor e às restrições impostas pela marca, tornou-se impossível para o grupo a obtenção dos dados pretendidos.

Posto isto, e após debate com a equipa docente, foi decidido que nos seriam disponibilizados dados referentes a uma pulseira "Xiaomi Mi Band 2", nos quais teriam de ser geradas conclusões adequadas, e de extrema relevância para o utilizador.

Com este fim, foi desenvolvida uma aplicação para *desktop* onde é possível submeter dados referentes a este sensor, no formato csv, produzindo conclusões referentes aos seguintes tópicos:

- Análise de Batimentos Cardíacos;
- Análise do Sono;
- Análise de Contabilização de Passos;
- Correlação entre os três resultados anteriores, permitindo gerar conclusões globais;

3 Análise dos Dados

Como referido anteriormente, foi fornecido ao grupo, por parte da equipa docente, um conjunto de dados relativos a um utilizador com uma pulseira "Xiaomi Mi Band 2". Estes dados foram entregues em formato *sqlite*, contendo as tabelas *heartrate*, *sleep* e *steps*.

De modo a facilitar a consulta e análise de dados por parte da nossa aplicação, o primeiro passo passou por extrair todos estes dados para ficheiros no formato *csv*, que posteriormente seriam interpretados pela aplicação. Assim, cada tabela deu origem ao seu respetivo ficheiro *csv*.

Em seguida, foi necessário compreender os dados presentes em cada ficheiro, identificando quais seriam mais relevantes para a análise, e posteriormente para as conclusões finais. É importante realçar as diferentes frequências com que os dados se encontravam registados nos respetivos ficheiros:

- "heartrate.csv": dados registados com frequência de 2 em 2 minutos;
- "sleep.csv": dados referentes a cada período de sono do utilizador;
- "steps.csv": dados registados com frequência de 1 em 1 minuto;

Verificada a constituição de cada ficheiro, tornou-se fundamental identificar os possíveis erros que estes poderiam ter, adotando um conjunto de soluções para a sua resolução. Por exemplo, no ficheiro referente aos batimentos cardíacos do utilizador, foi possível verificar um número considerável de dados com valores muito baixos ou até nulos. Estes dados não devem ser considerados para análise, uma vez que é humanamente impossível que um indivíduo apresente valores nestes intervalos. A sua origem pode estar relacionada com períodos de tempo em que o utilizador está sem a pulseira colocada, ou até mesmo por falhas do sensor.

Deste modo, e após conversa com o professor, foi necessário elaborar algumas estratégias que permitissem uma correta e eficaz exploração dos dados, originando assim conclusões mais realistas.

Apresentam-se então, as decisões tomadas pelo grupo para garantir o correto funcionamento do sistema:

- No ficheiro "heartrate.csv", eliminar todas as entradas em que o valor de *heartrate* fosse inferior a 38 (778 linhas).
- No ficheiro "sleep.csv", remover as colunas *stages*, *deleted* e *manually*.
- No ficheiro "steps.csv", remover a coluna *category*.

4 Métricas

De modo a conseguirmos explorar os dados de uma forma correta e eficaz, implementou-se um conjunto de métricas de análise, que permitem tirar conclusões importantes perante a informação adquirida pelo sensor. Em seguida serão descritas todas as métricas implementadas, bem como os seus algoritmos correspondentes.

4.1 Batimento Cardíaco

Em relação aos dados presentes no ficheiro "heartrate.csv", decidiu-se que seria importante realizar várias análises, variando os intervalos de tempo. Deste modo, o grupo focou-se em implementar as seguintes funcionalidades:

- Média de batimentos cardíacos;
- Desvio Padrão de batimentos cardíacos;
- Top 10 de valores mais elevados, identificando a data e o número total de batimentos cardíacos;
- Top 10 de valores mais baixos, identificando a data e o número total de batimentos cardíacos;

Decididas as métricas a implementar, desenvolveu-se o algoritmo que permite guardar os dados tendo em conta um determinado intervalo de tempo definido pelo utilizador da aplicação. Para se compreender o código seguinte, é importante referir que este intervalo é guardado na variável "interval_in_min".

```
public void ImportData(HeartBeatMetrics hbm)
{
    int start_time = 0;
    int end_time = 0;
    int averageAux = 0;
    int count = 0;
    foreach (KeyValuePair<Int32, Int32> hb in hbm.GetDictionary())
    {
        if (start_time == 0 || (end_time != 0 && hb.Key > end_time))
        {
            //Quando ja passaram intervalo_in_min minutos
            if (end_time != 0 && hb.Key > end_time)
            {
                averageAux = averageAux / count;
                PairEndTimeHB pair = new PairEndTimeHB(end_time, averageAux);
                time_HrBeat.Add(start_time, pair);
            }

            start_time = hb.Key;
            end_time = start_time + interval_in_min * 60;
            averageAux = 0;
            count = 0;
        }

        //Enquanto a leitura pertence ao intervalo
        if (hb.Key <= end_time)
        {
            count++;
            averageAux += hb.Value;
        }
    }
}
```


4.2 Controlo de Sono

Os dados do arquivo "sleep.csv" apresentam a informação do tempo inicial e final do sono, o tempo acordado, o tempo de sono profundo e tempo de sono leve (start_time,end_time,awake,sleep,light). Assim, foi efetuada a importação da informação através da função **public void ImportFromCSV(string fileName)**, que recebe o ficheiro e realiza as operações necessárias para o *load* dos dados.

```
public void ImportFromCSV(string fileName)
{
    using (var reader = new StreamReader(fileName))
    {
        reader.ReadLine();
        while (!reader.EndOfStream)
        {
            var line = reader.ReadLine();
            var values = line.Split(' ');

            int time = Int32.Parse(values[1]);

            // public PullSleep(int sleep, int end_time, int awake, int deep,
            // int light)

            PullSleep pullstep = new PullSleep(Int32.Parse(values[2]), Int32.Parse(
            values[3]), Int32.Parse(values[4]), Int32.Parse(values[5]));
            startTime_sleep.Add(time, pullstep);
        }
    }
}
```

Com os dados já carregados em memória, seguiu-se com a implementação de funcionalidades que permitissem a sua análise. Tendo em conta todos os dados presentes, foram desenvolvidas as seguintes métricas:

- Média e Desvio Padrão do tempo total de sono;
- Média e Desvio Padrão do tempo que o utilizador está acordado durante o sono (*awake time*);
- Média e Desvio Padrão do tempo de sono profundo;
- Média e Desvio Padrão do tempo de sono leve;
- Top 5 de maiores/menores intervalos de sono, identificando-os pela data de início/fim, e indicando o valor de *awake*, *deep* e *light*.

Os cálculos associados às médias e desvios padrões, correspondem a simples operações percorrendo todos os dados anteriormente guardados em memória. Quanto à elaboração dos dois top's referidos, foi necessário ordenar um *dictionary* consoante o valor de tempo de sono (*end time-start time*).

```
public IEnumerable<KeyValuePair<int, PullSleep>> Top5(int x)
{
    /* Dictionary + variable sleep */
    Dictionary<Int32, PullSleep> sleep = new Dictionary<int, PullSleep>();

    foreach (KeyValuePair<Int32, PullSleep> pair in startTime_sleep)
    {
        // (end_time-start_time)
        int sleep_variable = pair.Value.GetEndTime() - pair.Key;

        PullSleep n = new PullSleep(sleep_variable, pair.Value.GetEndTime(),
        pair.Value.GetAwake(), pair.Value.GetDeep(), pair.Value.GetLight());

        sleep.Add(pair.Key, n);
    }
}
```

```

/* Sort by sleep value */
var sortedDict = from entry in sleep orderby entry.Value.GetSleep()
                  descending select entry;
var sortedDict2 = from entry in sleep orderby entry.Value.GetSleep()
                  ascending select entry;

/* Take 5 first elements */
IEnumerable<KeyValuePair<int, PullSleep>> newdic = sortedDict.Take(5);
IEnumerable<KeyValuePair<int, PullSleep>> newdic2 = sortedDict2.Take(5);

// ...
}

```

Todas estas métricas permitem uma análise ponderada sobre o sono do utilizador. Através das médias e desvios padrões, podemos verificar se a pessoa em questão está a dormir as horas necessárias, bem como a sua variação. Com a elaboração dos top's, é possível averiguar a influência dos fatores associados ao sono, relacionando-os com a sua respectiva duração.

4.3 Contabilização de Passos

No arquivo "steps.csv", consta a informação do tempo em que foram realizados os passos (com intervalos de um minutos), o número de steps e a intensidade dos mesmos (*time, steps, intensity*). Assim, o grupo optou por implementar um algoritmo de recolha de dados muito semelhante ao que foi apresentado para as métricas do sono. A informação é guardada em dois dicionários: *time_steps* e *date_steps*, sendo que a diferença apenas consta na chave utilizada, em formato *timestamp* e *DateAndTime*, respetivamente.

```

public void ImportFromCSV(string fileName)
{
    using (var reader = new StreamReader(fileName))
    {
        reader.ReadLine();//ignore the first line

        while (!reader.EndOfStream)
        {
            var line = reader.ReadLine();

            /* line : "time,intensity,steps"*/
            var values = line.Split(';');

            int time = Int32.Parse(values[0]);
            var date = DateAndTime(time).Date;
            PullSteps pullstep = new PullSteps(Int32.Parse(values[1]),
                                                Int32.Parse(values[2]));

            time_steps.Add(time, pullstep);

            if (!date_steps.ContainsKey(date))
            {
                date_steps.Add(date, pullstep);
                count_date_steps.Add(date, 1);
            }
            else
            {
                double intensity = date_steps[date].GetIntensity()+pullstep.GetIntensity();
                double steps = date_steps[date].GetSteps() + pullstep.GetSteps();
                PullSteps novo = new PullSteps(intensity, steps);
                date_steps[date] = novo;
                count_date_steps[date]++;
            }
        }
    }
}

```

Já como os dados em memória, efetuamos então um cálculo da média dos tempos, passos e intensidade,

possibilitando assim uma análise ponderada dos passos realizados pela pessoa. Além disso, tal como efetuado nas métricas do sono, implementamos o Top5 de melhores e piores passos, em intervalos de um minuto.

4.4 Correlação de Métricas

Todas as métricas descritas até agora apresentam valores de análise referentes unicamente a um único ficheiro. Deste modo, e como o objetivo principal do projeto é englobar todos estes ficheiros para obter conclusões importantes, o grupo decidiu acrescentar algumas funcionalidades ao sistema. Assim, foram desenvolvidas um conjunto de funcionalidades que permitem relacionar as três métricas descritas anteriormente.

4.4.1 Batimentos Cardíacos - Sono

Com o objetivo de aprofundar o estudo do sono, tornou-se necessário inferir a sua relação com as métricas associadas aos batimentos cardíacos. Assim, decidiu-se que era fundamental analisar duas situações específicas: batimentos cardíacos antes de um período de sono e batimentos cardíacos durante um período de sono.

Deste modo, o grupo definiu um conjunto de métricas importantes para a posterior análise de resultados sobre o utilizador:

- Média/Desvio Padrão de batimentos cardíacos antes de um período de sono;
- Média/Desvio Padrão de batimentos cardíacos durante de um período de sono;
- Top 5 de valores onde o batimento cardíaco é mais elevado/baixo, antes de um sono, identificando também a respetiva data.
- Top 5 de valores onde o batimento cardíaco é mais elevado/baixo, durante um sono, identificando também a respetiva data.

Foi criada a classe *HeartBeatForSleep*, possibilitando a correlação dos dados já obtidos nas classes *HeartBeatMetrics* e *SleepMetrics*. Para guardar os dados, utilizou-se 2 dicionários, um para o período antes do sono e outro para o período durante o sono.

```
/* Class Constructor For HeartBeat Sleep */

public HeartBeatForSleep(SleepMetrics spm, HeartBeatMetrics hbm, SistemaSensorizacao ss)
{
    programa = ss;

    // While Sleep
    heartBeatWhileSleep = new Dictionary<int, PairHeartBeatSleep>();
    foreach (KeyValuePair<Int32, PullSleep> sp in spm.GetDictionary())
    {
        // sp.Key -> start time sleep; sp.Value.GetEndTime() -> end time sleep
        PairHeartBeatSleep pairHBS = new PairHeartBeatSleep(hbm, sp.Key, sp.Value);
        if (pairHBS.GetHeartBeatMetrics().GetDictionary().Count > 0)
            heartBeatWhileSleep.Add(sp.Key, pairHBS);
    }

    // Before Sleep
    heartBeatBeforeSleep = new Dictionary<int, PairHeartBeatSleep>();
    int start_time = 0;
    foreach (KeyValuePair<Int32, PullSleep> sp in spm.GetDictionary())
    {
        //start time = 0 or = last end time; sp.Key = start time sleep
        PairHeartBeatSleep pairHBS = new PairHeartBeatSleep(hbm, start_time, sp.Key,
                                                             sp.Value);
        start_time = sp.Value.GetEndTime();
    }
}
```

```

        if (pairHBS.GetHeartBeatMetrics().GetDictionary().Count > 0)
            heartBeatBeforSleep.Add(sp.Key, pairHBS);
    }
}

```

4.4.2 Passos - Sono

O objetivo desta métrica passa por determinar a influência que a quantidade de passos diários pode ter sobre o sono, relacionando os dados presentes nas métricas de Passos e Sono.

Com este fim, decidiu-se implementar as seguintes métricas:

- Média/Desvio Padrão de contabilização de passos diários;
- Identificar o top 5 com maior/menor número de passos antes de um período de sono, assinalando o total da contagem, bem como os valores associados a *awake time*, *deep time* e *light time*, referentes ao sono.

Deste modo, foi criada a classe *StepsForSleep*, que relaciona os dados das classes já apresentadas, *StepsMetrics* e *SleepMetrics*. Estes dados são guardados num dicionário em que a chave é o tempo inicial do período de sono.

```

/* Class Constructor For Steps before Sleep*/

public StepsForSleep(SleepMetrics spm, StepsMetrics steps_m, SistemaSensorizacao ss)
{
    programa = ss;
    stepsForSleep = new Dictionary<int, PairStepsSleep>();
    int start_time = 0;
    foreach (KeyValuePair<Int32, PullSleep> sp in spm.GetDictionary())
    {
        PairStepsSleep pairStepsS = new PairStepsSleep(steps_m, start_time, sp.Key,
                                                         sp.Value);

        start_time = sp.Value.GetEndTime();
        if (pairStepsS.GetStepsMetrics().GetDictionary().Count > 0)
        {
            stepsForSleep.Add(sp.Key, pairStepsS);
            total_steps += pairStepsS.GetTotalSteps();
        }
    }
}

/* Top biggest steps */

public List<KeyValuePair<int, PairStepsSleep>> TopHigher(int top)
{
    var aux = new SortedDictionary<Int32, PairStepsSleep>(stepsForSleep);
    List<KeyValuePair<int, PairStepsSleep>> sorted =
        (from kv in aux orderby kv.Value.GetTotalSteps() descending select kv).ToList();
    return sorted.GetRange(0, top);
}

```

5 Interface

Com o objetivo de apresentar uma aplicação simples e intuitiva, foi desenvolvida uma interface que permite a execução de todas as métricas de análise mencionadas anteriormente. Em primeiro lugar, devem ser carregados os ficheiros *csv*:

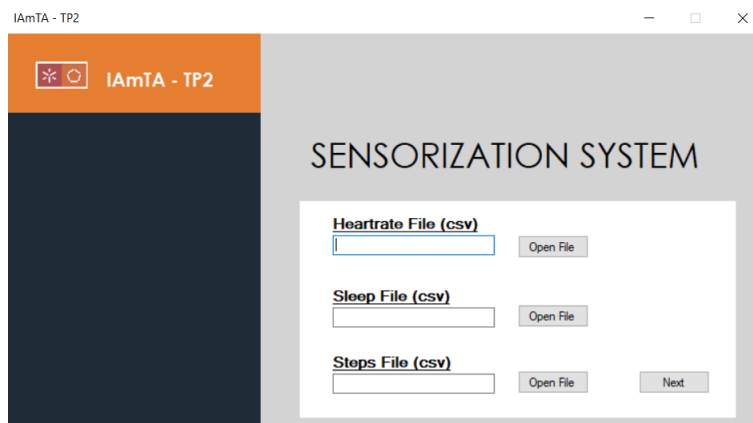


Figura 3: Janela inicial para carregamento de ficheiros *csv*

Após carregados os ficheiros, a aplicação executa as funções de análise e os resultados são apresentados. A interface encontra-se agora dividida em cinco divisões: *Heartbeat*, *Hours of Sleep*, *Accounting for Steps*, *Heart-Sleep*, *Steps-Sleep*.



Figura 4: Janela *Heartbeat* inicial

Inicialmente são apresentados os valores correspondentes às métricas dos batimentos cardíacos, associados a todas as entradas do ficheiro. O utilizador pode então optar por escolher um diferente intervalo de tempo, pressionando o botão "For Intervals". O tempo deve ser inserido em minutos.

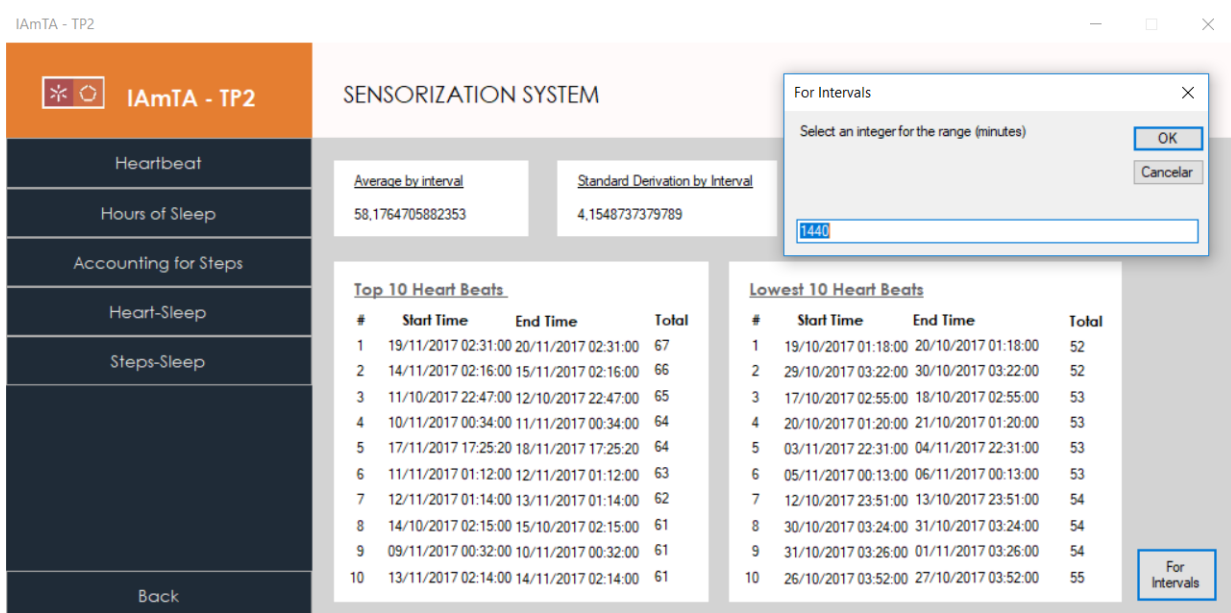


Figura 5: Janela com *Heartbeats* em intervalos de 1 dia

O utilizador pode também consultar os dados calculados para as restantes métricas anteriormente descritas.



Figura 6: Janela de Análise do Sono

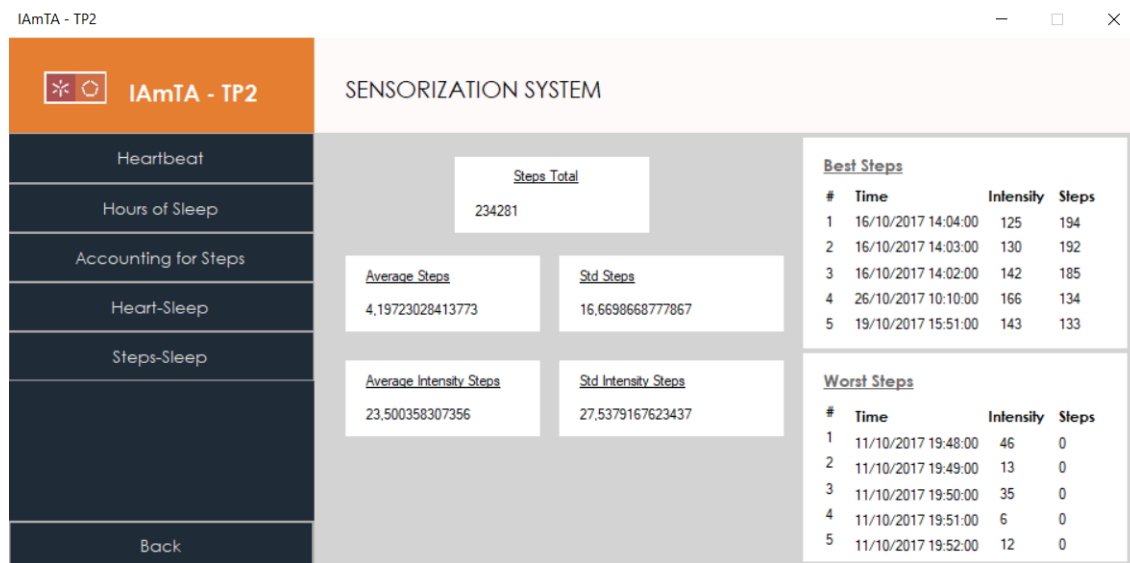


Figura 7: Janela de Análise dos Passos

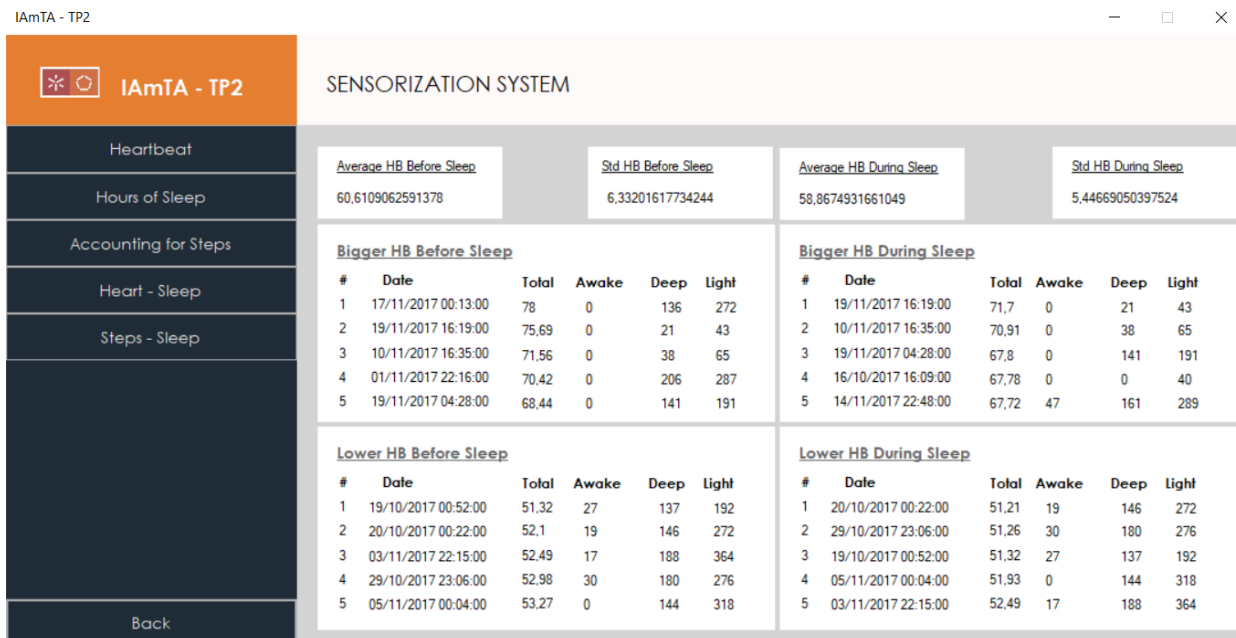


Figura 8: Janela de Análise da correlação Batimentos Cardíacos - Sono

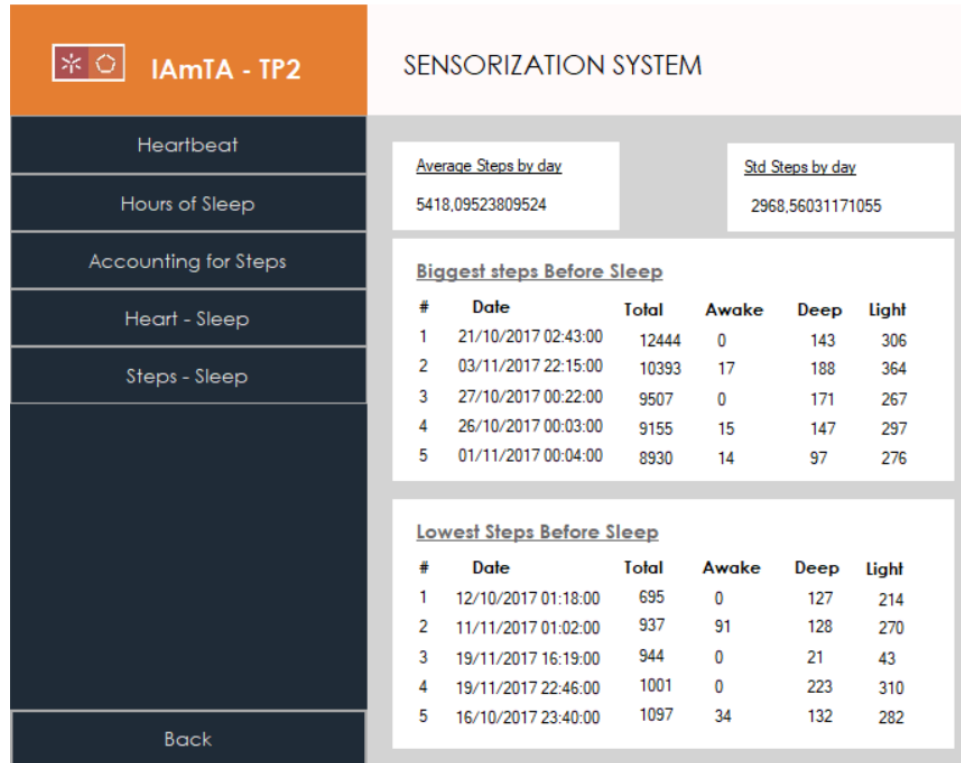


Figura 9: Janela de Análise da correlação Passos - Sono

6 Análise de Resultados

Após a apresentação dos resultados das métricas desenvolvidas, prosseguiremos à sua análise, de modo a consolidar o desempenho do sistema desenvolvido.

Considerando os valores apresentados para os batimentos cardíacos em intervalos de um minuto (4), facilmente se repara numa enorme diferença entre os valores mais altos e os valores mais baixos. Esta conclusão já seria esperada, uma vez que nos estamos a referir a intervalos de tempo muito baixos. No entanto, é importante analisar o valor da média e do desvio padrão.

De modo a compreendermos estes valores, é fundamental identificar quais as gamas de valores onde devem estar inseridos: [5]

	entre 18 e 25 anos	entre 26 e 35 anos	entre 36 e 45 anos
Excelente	49 a 55 bpm	49 a 54 bpm	54 a 59 bpm
Boa	57 a 61 bpm	57 a 61 bpm	60 a 62 bpm
Abaixo da média	71 a 73 bpm	72 a 74 bpm	73 a 76 bpm

Figura 10: Valores para Batimento Cardíaco - Homens

	entre 18 e 25 anos	entre 26 e 35 anos	entre 36 e 45 anos
Excelente	54 a 60 bpm	54 a 59 bpm	54 a 59 bpm
Boa	61 a 56 bpm	60 a 64 bpm	62 a 64 bpm
Abaixo da média	74 a 78 bpm	75 a 76 bpm	74 a 78 bpm

Figura 11: Valores para Batimento Cardíaco - Mulheres

Uma vez que o valor da média apresentado é de cerca de 59 batimentos cardíacos por minuto, e o desvio padrão é relativamente baixo, é possível concluir que o utilizador apresenta um bom nível de frequência cardíaca. Apesar de existirem valores altos, como os referidos no "Top 10 Heart Beats", estes podem representar momentos em que o utilizador esteja a exercitar ou momentos de emoções fortes. Por outro lado, os valores baixos apresentados podem estar associados aos períodos em que o utilizador está a dormir.

Ainda consultando os níveis de batimento cardíaco, podemos observar os valores apresentados para os batimentos cardíacos diários - 5. Neste caso, é possível verificar que as diferenças de valores já não são tão acentuadas, como seria de esperar. Por outro lado, a média praticamente mantém-se e o desvio padrão desce, permitindo assim reforçar a ideia que o utilizador apresenta valores adequados para esta métrica.

Nos resultados apresentados para a análise do sono (6), é essencial identificar as diferenças apresentadas entre melhores e piores períodos de sono. Como podemos observar, em ambos os casos o valor de tempo de sono leve é superior ao valor de sono profundo, sendo esta diferença mais acentuada quanto maior for o período de sono.

Por outro lado, é de notar que o utilizador tem uma média de período de sono de cerca de 7 horas, o que é um valor adequando tendo em conta as horas que um adulto deve dormir. No entanto, o valor

apresentado para o desvio padrão é de 2 horas, o que ainda representa uma significativa variância no valor médio. A mesma situação acontece com resultados exibidos para o tempo que o utilizador está acordado durante o sono, o tempo de sono leve e tempo de sono profundo.

Em relação aos resultados obtidos para a contabilização de passos, é de notar que os 3 primeiros valores do top "best steps" correspondem a momentos consecutivos, na mesma data. Podemos ainda observar os valores baixos de média de passos e intensidade, bem como o seu elevado desvio padrão. Esta situação facilmente é justificada uma vez que estão a ser considerados momentos em que o utilizador está a dormir, ou seja, não são contabilizados passos, mas estes entram nos cálculos. Este facto pode ser verificado também na tabela de "Worst Steps".

Analisando os valores correspondentes à correlação de dados entre batimentos cardíacos e sono, é possível observar a diminuição do valor médio de batimentos cardíacos durante o período em que o utilizador está a dormir. Por outro lado, verifica-se quanto menor é o valor de batimentos cardíacos (antes e durante o sono), maior é a soma entre *awake time*, *deep time* e *light time*.

Por fim, na janela de apresentação de resultados para a correlação das métricas associadas à contabilização de passos e horas de sono, é possível consultar a média de passos que o utilizador dá em cada dia, bem como o seu elevado desvio padrão. Esta variação pode também ser observada ao consultar os top's apresentados na aplicação, com o número de passos que um utilizador dá, antes de um período de sono. É também importante realçar que, de um modo geral, é possível verificar que quando o utilizador efetua um maior número de passos antes de dormir, o valor da soma entre *awake time*, *deep time* e *light time* também aumenta.

7 Conclusões e Trabalho Futuro

Terminada a realização do projeto, é então possível concluir alguns aspectos que foram determinantes para a correta implementação do sistema, e que levaram à demonstração de resultados relevantes.

Em primeiro lugar, foi essencial realizar uma análise de dados prévia, que permitiu eliminar dados insignificantes para a aplicação e que poderiam causar enormes variações nos resultados finais.

Editados os ficheiros de dados, seguiu-se com uma análise à sua estruturação, de modo a elaborar um conjunto de funcionalidades a que o sistema deveria responder. Deste modo, para cada um dos ficheiros, foi implementada uma lista de métricas que permitem verificar situações importantes e conclusivas.

Uma vez que o objetivo do trabalho passava por identificar o maior número de padrões possíveis entre os diferentes dados, tornou-se fundamental implementar também um conjunto de métricas que permitissem a correlação de informação, neste caso, entre batimentos cardíacos e passos com as respectivas horas de sono. Estes resultados permitem identificar situações do dia-a-dia de um utilizador, assim como diferenciar os valores apresentados para os diferentes contextos.

De um modo geral, o grupo ficou satisfeito com o trabalho desenvolvido, uma vez que conseguiu implementar todas as funcionalidades pretendidas, obtendo um conjunto de resultados importantes para o utilizador. Como trabalho futuro, seria interessante alargar os horizontes desta aplicação para outro tipo de sensores, atuando diretamente com os mesmos, o que não aconteceu neste projeto devido à fraca documentação e restrições do sensor fornecido ao grupo.

Referências

- [1] ISTAG (2001), *Scenarios for Ambient Intelligence in 2010*, European Commission Report.
- [2] Ramos, C., Augusto, J. C., Shapiro, D. (2008). *Ambient intelligence—the next step for artificial intelligence*. IEEE Intelligent Systems, 23(2), 15–18
- [3] <https://www.businessinsider.com.au/the-smartwatch-industry-report-2014-11>
- [4] <https://www.medicalnewstoday.com/articles/235710.php>
- [5] <https://www.tuasaude.com/frequencia-cardiaca/>