

MESTRADO EM ENGENHARIA BIOMÉDICA
TÉCNICAS DE INSTRUMENTAÇÃO E CONTROLO

**MONITORAMENTO DE BATIMENTOS
CARDÍACOS EM TEMPO REAL UTILIZANDO
SENSOR DE ECG
—— DESENVOLVIMENTO E TESTE ——**

AUTORES:

BRUNO PINTO DOS SANTOS

2019229168

HUGO TOMÁS PEREIRA ALEXANDRE

2019235256

21 DE JUNHO DE 2023

2º SEMESTRE

ÍNDICE

| | |
|--|----|
| 1. Descrição Geral | 3 |
| 1.1. Função do Instrumento | 3 |
| 1.2. Condições de Utilização | 5 |
| 2. Desenvolvimento do Sistema | 5 |
| 2.1. Diagrama de Blocos do Sistema | 6 |
| 2.2. Circuito Eletrónico | 7 |
| 2.2.1. Circuito Integrado AD8232 | 8 |
| 2.2.2. Arduino MKR1000 | 8 |
| 3. Desenvolvimento do Software | 8 |
| 3.1. Diagrama de blocos do <i>software</i> | 9 |
| 3.2. Aquisição de dados: Arduino MKR1000 | 10 |
| 3.2.1. Comunicação de Matlab para Arduino MKR1000 | 10 |
| 3.2.2. Comunicação de Arduino MKR1000 para Matlab | 11 |
| 3.3. Processamento e armazenamento de dados: Matlab | 11 |
| 3.3.1. Algoritmos de processamento | 12 |
| 3.3.2. Estrutura de Dados | 15 |
| 3.4. Interface de apresentação de dados: AppDesigner Matlab | 15 |
| 3.4.1. Login/Sign In | 16 |
| 3.4.2. Interface Principal | 16 |
| 3.4.3. Nova Aquisição | 17 |
| 3.4.4. Load de Aquisições | 17 |
| 4. Testagem do sistema | 18 |
| 5. Conclusão | 24 |
| 6. Bibliografia | 26 |
| Anexos | 27 |
| I. <i>Scripts</i> desenvolvidos e funcionamento da <i>GUI</i> implementada | 27 |
| II. Planeamento de Tarefas (17 de março de 2023) | 28 |
| III. Identificação dos Riscos (17 de março de 2023) | 29 |

1. Descrição Geral

Ao longo do atual semestre, propôs-se desenvolver um eletrocardiógrafo portátil, capaz de realizar as medições da atividade elétrica cardíaca em tempo real, bem como analisar o sinal resultante, por forma a poderem ser identificadas potenciais anomalias da função cardíaca, consequentes de eventuais patologias do paciente.

1.1. Função do Instrumento

Há cerca de dois séculos atrás, descobriu-se que o coração, durante a sua atividade, produz corrente elétrica, capaz de ser detetada e amostrada na forma de um gráfico, a que hoje chamamos de eletrocardiograma, *ECG*. Desde então que esta tecnologia se tornou uma ferramenta importante, especialmente, para o diagnóstico de doenças cardíacas, como as arritmias, tendo-se vulgarizado por hospitais e clínicas de todo o mundo. O *ECG* caracteriza-se por ser um exame altamente sensível, específico, preciso, rápido e não-invasivo, implicando unicamente o uso de elétrodos colocados de uma forma específica (**Figura 1**) [1]. Para além disso, o resultado do exame tem uma interpretação facilitada, uma vez que há comparação do mesmo com um gráfico padronizado para a condição de normalidade [2].

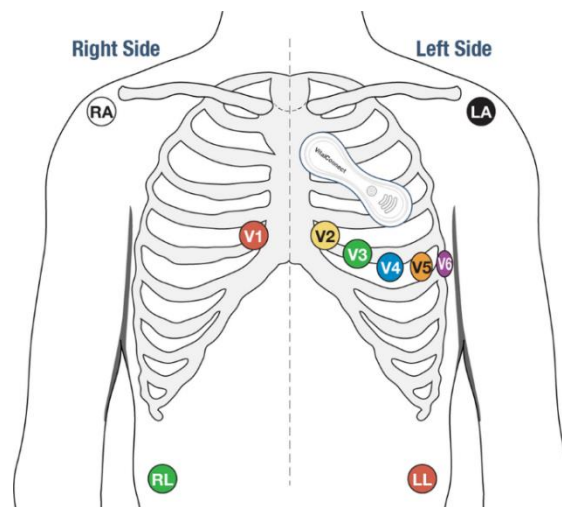


Figura 1. *Layout* de posicionamento dos elétrodos de um *ECG device* de doze elétrodos, segundo o posicionamento *Mason-Likar*. *RA*, *LA*, *RL*, and *LL* são, respetivamente, o elétrodo do braço direito, braço esquerdo, perna direita e perna esquerda.

De uma forma resumida, o sinal obtido pelos elétrodos, que captam variações na atividade elétrica do coração durante os períodos de contração e relaxamento do ciclo cardíaco, é uma sequência de ondas e complexos, nomeados de acordo com a relação aos eventos fisiológicos a ocorrer em concomitância. Através da **Figura 2** pode-se visualizar o formato da curva *ECG* [1,3]:

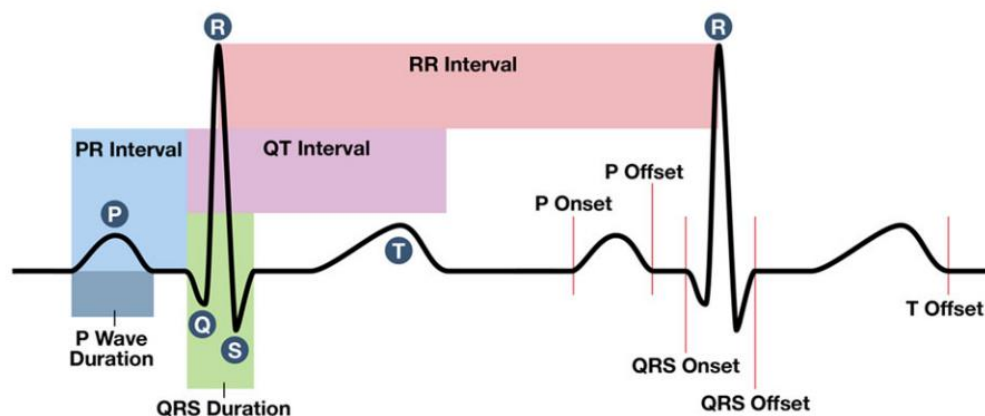


Figura 2. Esquema simplificado de uma curva ECG.

Nesta ótica, segundo o que já foi referido previamente, pretendeu-se reproduzir com o máximo de rigor um eletrocardiograma convencional. Como forma de diferenciação/ inovação pretendeu-se desenvolver um sistema *wireless* em tempo real, o que tem especial aplicabilidade no monitoramento remoto de paciente com doenças cardíacas, permitindo o acesso do exame em tempo real por parte dos profissionais de saúde. Para além disso, pretendeu-se estruturar um algoritmo capaz de interpretar o sinal resultante, e por sua vez devolver um diagnóstico rudimentar em caso de irregularidades cardíacas. Num fase inicial, quis-se que este dispositivo *pHealth* e respetivo *software* conseguisse fornecer informações sobre os batimentos cardíacos por minuto (*BPM*), o formato batimento cardíaco médio, a deteção das ondas e complexos característicos do sinal e, ainda, uma análise no domínio do tempo e das frequências da variabilidade da taxa cardíaca (*HVR – Heart Rate Variability*). Os pontos anteriores capacitaram, também, o instrumento com a funcionalidade de classificação do estado cardíaco entre estados de taquicardia, bradicardia ou situação normalizada [4].

De uma forma condensada, este tipo de *devices*, para além do uso mais facilitado, económico e cómodo para o paciente, tem tendência para uma deteção precoce dos problemas cardíacos, assim como um monitoramento contínuo da atividade elétrica do coração. Usualmente, este tipo de equipamentos vestíveis são utilizados por pacientes com condições pré-existent de cardiopatias, pacientes com sintomas ou de elevado risco de doenças cardíacas ou, até mesmo, atletas, cujo intuito passe pelo monitoramento em paralelo com a prática de atividade física [1,2].

O instrumento que se pretendeu projetar, apresenta uma aplicabilidade muito extensa, quer do ponto de vista médico, quer do ponto de vista do cuidado pessoal. Esta capacidade relaciona-se, essencialmente, com a polivalência do *device*, uma vez que em simultâneo com a recolha de dados do sinal elétrico do coração, útil para profissionais de saúde especializados na leitura de sinais *ECG*, é ainda possível que o utilizador tenha acesso a um controlo automatizado de alguns parâmetros chave do seu estado cardíaco.

1.2. Condições de Utilização

Em relação à utilização deste *gadget pHealth*, sendo este um equipamento não-invasivo, pretendeu-se que a sua utilização fosse minimalista e de acesso generalizado, sem necessidade de requerimento médico obrigatório. A única condição que esta tipologia de aparelhos médicos implica, à priori, é o bom contacto entre a pele e os elétrodo, bem como a colocação destes sensores em localizações anatómicas específicas. Devido a esse motivo, numa primeira fase, para garantir a correta captação do sinal de *ECG*, incorporou-se na interface desenvolvida um breve esquema de incorporação dos sensores. Num futuro próximo, perspetiva-se que a resolução da problemática passará pela integração do sistema atual com uma camisola, cujos recetores de sinal estarão previamente posicionados.

No que diz respeito à informação acessível pelo utilizador, atribuiu-se acesso dos vários dados recolhidos para fins de monitoramento corrente aos utilizadores, tendo-se, para tal, estruturado a apresentação dos mesmos de uma forma minimalista e acessível. Este facto leva a inferir que, à partida, não haverá qualquer pré-condição para o uso da interface associada a este aparelho.

Por fim, em termo dos dados obtidos, constatou-se que os mesmos devem ter em conta o historial clínico, pois a análise do sinal *ECG* varia, ligeiramente, quando se compara casos de pacientes saudáveis ou com pacientes com alguma cardiopatia. O contexto das medições também é relevante, pois uma variação na curva *ECG* devido a uma patologia, não tem o mesmo peso que a variação devido à execução de uma atividade física aleatória, mesmo que essa variação seja aparentemente igual.

2. Desenvolvimento do Sistema

No atual projeto, o protótipo desenvolvido, de uma forma sucinta, consistiu na interação entre a componente física, recetora dos sinais e a componente digital, responsável pelo processamento e amostragem dos mesmos.

No primeiro caso, optou-se por utilizar o *Arduino MKR1000*, associado ao Circuito Integrado *AD8232*. Este último, por sua vez, consiste num conjunto de componentes eletrónicos, que atuam como um todo para receber, manipular e transmitir sinais elétricos cardíacos, representados sob forma de curva *ECG*. Este circuito pode ser dividido em três elementos-chave: **Elétrodos (modelo HI24SG)**, **Cabos de ligação ECG**, o **Módulo AD8232 Heart Rate Monitor**. Para além disso, a componente física foi constituída por um regulador geral, o *PC*. De realçar que devido a algumas condicionantes, posteriormente explicitadas, não foi possível projetar um aparelho com todas as funcionalidades pretendidas, em especial a introdução da capacidade *wireless* do sistema. Desta forma, atribuiu-se ao *PC* as funcionalidades de fonte de energia e armazenamento, bem como de principal interveniente na comunicação física estabelecida (ligação por cabo *USB* entre

módulo *Arduino MKR1000* e *PC*). Ainda assim, futuramente, acredita-se ser possível a introdução de uma comunicação sem fios, fazendo uso de um módulo *Bluetooth* (a exclusão da alternativa por *Wi-Fi* deve-se ao facto dessa tecnologia de comunicação, para além de mais instável, implicar um maior consumo enérgico do aparelho, tornando-o por consequência ineficiente). De notar, que esta abordagem permitirá que seja o objeto mais complexo, neste caso concreto o *PC*, a ter a responsabilidade de, após receção dos dados, os transmitir para uma *Cloud* e, portanto, permitir o acesso de profissionais de saúde aos mesmos, em tempo quase-real. Já, como solução da fonte de energia do *gadget* portátil, uma opção teoricamente viável será a adição uma bateria de lítio de 3,7 V de 1024 *mAh*. Em relação à parte digital do projeto, esta será abordada detalhadamente em tópicos futuros.

2.1. Diagrama de Blocos do Sistema

O dispositivo *pHealth* contruído pode ser observado, de forma simplista, através do seguinte diagrama de blocos (Figura 3). Partindo deste esquema, descreveu-se, de forma mais detalhada, a função de cada elemento do circuito através da Figura 4. De destacar que o *PC* é a ferramenta que permitiu a transição do sistema físico e para o sistema digital.

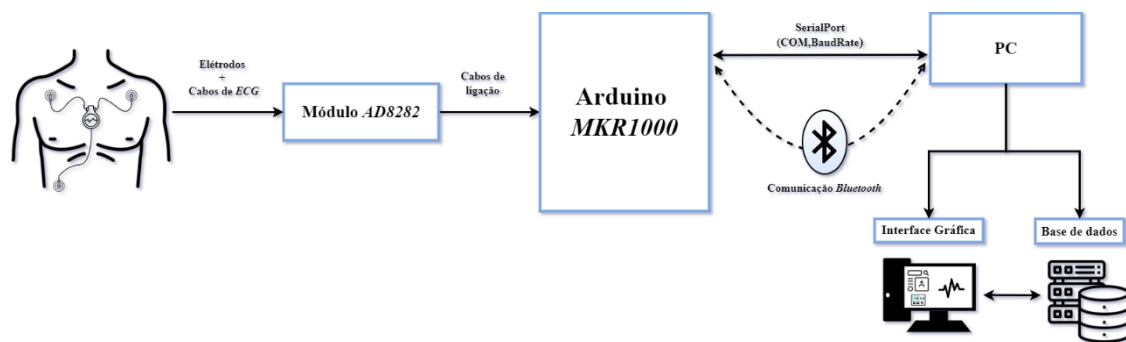


Figura 3. Diagrama de blocos geral do eletrocardiógrafo que se reproduziu. Ligação via *Bluetooth* entre *Arduino MKR100* e *PC* referida como sendo uma perspetiva futura.

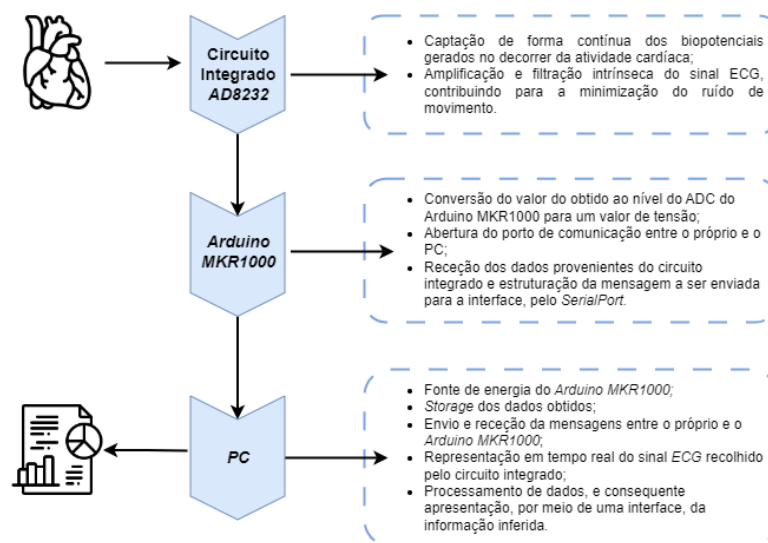


Figura 4. Esquemática detalhada das funcionalidade de cada elemento da componente física do protótipo.

2.2. Circuito Eletrônico

Tal como previamente mencionado, de seguida é possível visualizar os vários intervenientes físicos do protótipo desenvolvido. Na Figura 5 é apresentado o circuito eletrónico completo responsável pelo funcionamento do *device*. Neste é possível visualizar o *Arduino MKR1000*, o circuito integrador *AD8232* e os vários cabos de ligação entre pinos dos elementos anteriores. De realçar a não existência de resistência limitadoras de corrente, uma vez que o módulo de aquisição do sinal *ECG* está internamente preparado para lidar com as tensões fornecidas pelo microcontrolador. Para além disso, embora não representado, o *Arduino MKR1000* encontra-se ligado via *USB-B* ao *PC* (fonte de energia).

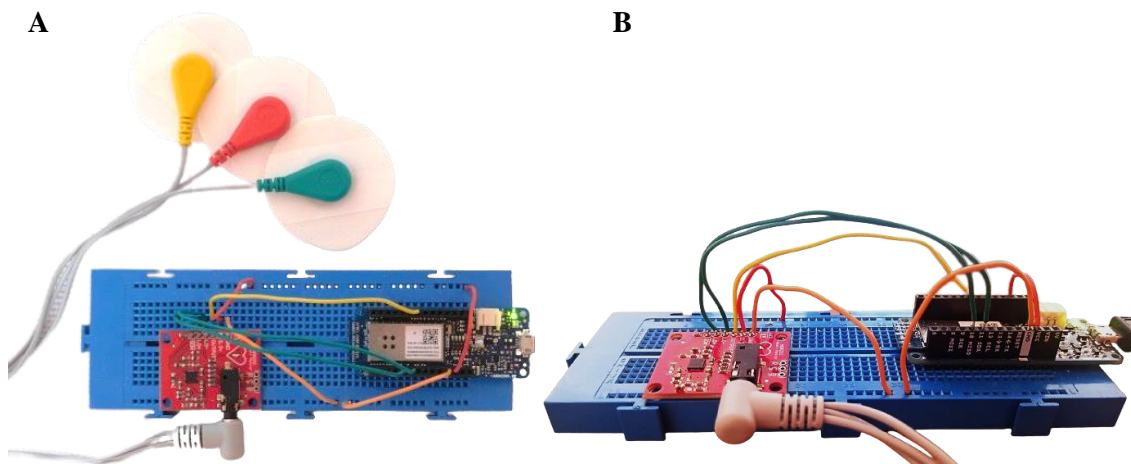


Figura 5. **A.** Visão de topo do circuito montado para o correto funcionamento do protótipo desenvolvido; **B.** Circuito equivalente ao ponto anterior, com a exceção da vista ser lateral. De notar que em ambos os casos não é mostrada a ligação entre o microcontrolador e a fonte de energia. Contudo, esta foi realizada via *USB-B* entre o *Arduino MKR1000* e o *PC*.

Considerando que a esquemática anterior possa não ter a resolução necessária para perceção das ligações, em seguida, surge, através da Tabela 1, os pinos do *Arduino MKR1000* e do Módulo *AD8232* ligados entre eles por fios de ligação:

Tabela 1. Lista do conjunto de ligações efetuadas no circuito evidenciado na Figura 5, entre o módulo *AD8232* o microcontrolador em uso. Para além desta, ainda que não sendo da mesma natureza, foram interligados o módulo *AD8232* com os elétrodos, através de um conector de áudio de 3,5 mm, e o *Arduino MKR1000* com o *PC*, via *USB-B*.

| <i>Arduino MKR1000</i> | Módulo <i>AD8232</i> |
|------------------------|----------------------|
| GND | GND |
| 3.3 V | 3.3 V |
| A6 | OUTPUT |
| 10 | LO+ |
| 11 | LO- |

De notar que, a saída *OUTPUT* do módulo *AD8232*, pode estar ligada a qualquer das saídas analógicas do Arduino, sendo apenas necessário manipular o código implementado no *Arduino IDE*, anexado ao atual documento.

2.2.1. Circuito Integrado *AD8232*

Tal como representado na Figura 4, o Circuito Integrado *AD8232* surge como o subsistema responsável pela captação, leitura e manipulação dos sinais *ECG*, tendo sido projetado para extrair, amplificar e filtrar pequenos sinais provocados por biopotenciais em situações de ruído, tais como o movimento ou a colocação de elétrodos remotos. Sobre este circuito é importante destacar o comportamento sinusoidal do sinal adquirido em condição de repouso, bem como a resposta adaptativa para minimização de ruído externo na condição ativa (permite que o sinal *ECG* mantenha as suas características principais mesmo com erros associados à leitura do mesmo) [5].

2.2.2. *Arduino MKR1000*

Após a recolha e condicionamento dos dados, é necessário que os mesmos sejam controlados e monitorizados por uma entidade programável, como o *Arduino MKR1000*. Este microcontrolador tem uma frequência de *clock* de 48 MHz, um ADC (Conversor Analógico-Digital) de 12-bits, uma memória *Flash* e *SRAM* de 256 KB e 32 KB e uma variedade de interfaces de comunicação [6]. Tais características permitiram que o sinal fosse recolhido a frequências próximas da gama de 300 a 500 Hz, em tempo quási-real. **De evidenciar que o detalhe sobre a comunicação é feita nos tópicos relacionados com o *software* desenvolvidos.**

3. Desenvolvimento do Software

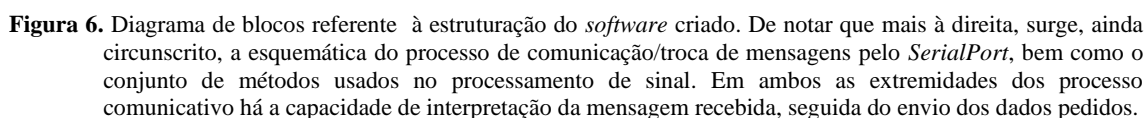
No que se refere à componente digital, primeiramente, o sistema desenvolvido baseou-se essencialmente na minimização dos efeitos negativos advindos dos constituintes físicos. Para tal, foi projetada uma rede de algoritmos de processamento de dados de *ECG*, condensada sob forma de uma interface *Matlab*, detalhada em tópicos subsequentes. De notar que, o *software* desenvolvido teve por base alguns conhecimentos prévios sobre a tipologia de sinais em estudo, bem como informações sobre os dados recebidos pós-transmissão do microcontrolador para o dispositivo de armazenamento e processamento, como as seguintes:

- Um sinal *ECG*, encontra-se, vulgarmente, numa gama de frequências entre 0,05 Hz e 150 Hz, podendo ser captadas diferenças de potenciais num intervalo entre os 0,5 mV e os 5 mV [7];
- Uma vez que o teorema da amostragem de *Nyquist-Shannon* implica a frequência de amostragem se, no mínimo, o dobro da frequência máxima do sinal a ser amostrado, a taxa

- O condicionamento do sinal (amplificação, filtração e adaptação a ruído de movimento) é, na grande maioria, feito ao nível do módulo *AD8232*.

Posto isso, decidiu-se limitar o *Arduino MKR1000*, apenas, à aquisição de dados, durante um determinado intervalo temporal e após uma mensagem de nova aquisição (dados enviados pelo *SerialPort* estabelecido por parte da interface, após personalização dos valores pelo utilizador).

De uma forma simplista, realizada a transmissão de dados de forma contínua e em tempo real para a interface de processamento, os resultados brutos são guardados, permitindo o acesso futuro, e amostrados de forma iterativa. De forma esquemática, em seguida surge a representação de todo o processo interno da solução apresentada (Figura 6).



3.2. Aquisição de dados: *Arduino MKR1000*

A placa *Arduino* foi programada tendo por base o código em anexo *ecg_reader.ino*. Através deste, o microcontrolador foi capacitado com a tarefa de comunicação com a interface *Matlab* em ambos os sentidos, ou seja, possibilitou a receção e interpretação de mensagens vindas do *Matlab*, para além do envio de estruturas com informações sobre a aquisição, levada a cabo pelo módulo *AD8232*. Para tal em ambos os ambientes, foi aberto um porto de comunicação estabelecido via ligação *USB-B*, após seleção de uma *Serialport* comum e definição de uma *baudRate* de 115200 bits por segundo. De realçar que todas as restantes especificações foram deixadas no modo *default*, uma vez que não intervinham diretamente com a performance da comunicação. Por fim, antes da inicialização de qualquer tipo de troca de mensagens entre os dois sistemas supracitados, garantiu-se o estado vazio inicial do porto de comunicação, através do comando *flush()*.

3.2.1. Comunicação de *Matlab* para *Arduino MKR1000*

Tal como já detalhado anteriormente, inicialmente, começou-se pela abertura de um porto de comunicação entre o *Arduino MKR1000* e o *Matlab*.

Considerando o facto de que o principal objetivo do trabalho era iniciar voluntariamente uma aquisição, foi desenvolvida uma metodologia semelhante a um botão digital na interface gráfica. Ou seja, o utilizador teria a opção de iniciar uma nova aquisição, definindo certos parâmetros como a frequência de amostragem e o tempo de aquisição. De seguida, essa indicação viajava pelo *SerialPort*, até encontrar o *Arduino MKR1000*, permanentemente à escuta, segundo o formato de mensagem “**aquisition_time | aquisition_frequency**” (desenvolvido pelas funções *serialCommunication.m* e *ecg_aquisition.m*). Neste ponto é muito importante referir que a frequência de amostragem mencionada é exclusivamente uma frequência requerida, ou seja, é comum que o microcontrolador induza a que o período de amostragem seja superior ao esperado, sendo, posteriormente, considerada como frequência de amostragem, o inverso do período médio, calculado diretamente com base no vetor dos tempos gerados aquando da aquisição.

Posto isso, havia o desencadeamento de uma resposta, baseada num conjunto de mensagem sucessivas, segundo o padrão “**data_time | data_ecg**”, enviada sequencialmente para a interface *Matlab*, via *SerialPort*.

Para tal, foram criadas duas funções, *msg_wait()* e *processCommand()*. Tal como o nome indica, a primeira função possibilita a espera até receção de um mensagem vinda do *SerialPort*, enquanto a segunda processa a informação dessa mesma mensagem.

3.2.2. Comunicação de Arduino MKR1000 para Matlab

Esta tarefa é facilmente alcançada, pelo uso da função *analogRead()* chamada no *loop()* do código inserido no *Arduino MKR1000*. De modo a evitar a pausa entre aquisições de amostras, e assim garantir uma certa taxa de amostragem, usou-se função *blink()*. Paralelamente com a amostra do sinal *ECG* recolhido, associou-se um valor temporal, gerado pelo comando *millis()*. Por fim, o processo de obtenção de dados termina após o tempo de aquisição findar.

3.3. Processamento e armazenamento de dados: Matlab

Do ponto de vista do *Matlab* é necessário uma extensa contextualização de como toda a informação foi manipulada. Assim, inicialmente é esperado que o paciente se registre na aplicação, de modo a poder salvar todas as aquisições efetuadas ao longo do tempo. Posto isso, o utilizador segue para um painel com a possibilidade de fazer *load* de aquisições futuras, bem com das estatísticas previamente obtidas, ou então de iniciar uma nova aquisição. Neste último caso, após uma eventual visualização do sinal em tempo real (devido ao peso gráfico requerido, pode haver *delays* associados, mesmo após otimização avançada da representação), existem algoritmos de processamento de sinal prontamente responsivos. Toda a informação até então gerada é guardada numa estrutura única (detalhada nos tópicos seguintes), própria do paciente em causa. De notar, que o processo de comunicação entre o *Arduino MKR1000* e o *Matlab* é fulcral, pelo que em casos de problemas na ligação entre ambos, é enviada uma mensagem de erro. Por fim o utilizador tem uma panóplia de funcionalidades a seu dispor, entre as quais:

- Manipulação dos parâmetros *standard* do processamento sinal;
- Eliminação de segmentos de sinal específicos (útil quando há erros durante a aquisição);
- Visualização do sinal pré-processado (*Baseline Removal* e aplicação de *Lowpass Filter*)
- Visualização dos picos P e T e complexo QRS;
- Visualização sobreposta de todos os segmentos correspondentes a um batimento cardíaco;
- Visualização do gráfico do batimento cardíaco médio;
- Cálculo do batimento cardíaco por minuto (*BPM*);
- Visualização da variação dos intervalos entre picos R;
- Análise no domínio das frequências da variação dos intervalos entre picos R;
- Cálculo de métricas relacionadas com a variabilidade da taxa cardíaca (*HRV*);
- Diagnóstico Rudimentar (não medicamente válido);

De seguida surge uma breve descrição teórico-prática do processamento executado sobre os dados originais, bem como a forma de estruturação interna dos dados. De realçar que, os algoritmos desenvolvidos concentram-se nas funções *ecg_processing.m* e *diagnostic.m*.

3.3.1. Algoritmos de processamento

Após aquisição do sinal de *ECG* bruto, foi necessário a aplicação de algoritmos de processamento capazes de manipular os dados e inferir, a partir dos mesmos, conclusões medicamente válidas. Desta forma, na seguinte secção explicita-se os vários métodos executados sobre a aquisição efetuada. De notar que, por forma a otimizar o tempo de apresentação de resultados, resolveu-se aplicar todos estas metodologias logo após uma nova aquisição (possível de se optar por um reprocessamento caso os valores padrões não se ajustem), sendo em seguida salvas.

3.3.1.1. *Baseline Removal e aplicação de Lowpass Filter*

Como em seguida será amostrado no tópico de *Testagem do Sistema*, os dados brutos geralmente apresentavam uma espécie de componente sinusoidal, proveniente das características intrínsecas do sensor. Para remoção desse padrão, inicialmente, começou-se por se aplicar um filtro *moving-average* de valor padrão 0,1 segundos (ou seja, média móvel de $0,1 \times fs$ amostras, sendo fs a frequência de amostragem). Tendo em conta o comportamento médio ao longo do tempo do sinal, de seguida, procedeu-se com a eliminação ponto a ponto do sinal de média móvel sobre os dados originais. Este passo permitiu que o sinal resultante contivesse, igualmente, as variações características de um *ECG*, embora sem *baseline* (este procedimento complementou a remoção de ruído de baixas frequências, proveniente por exemplo do movimento respiratório). Posto isso foi aplicado um filtro passa-baixo com frequência de corte de 30 Hz, por forma a ser removido todas as elevadas frequências sem qualquer tipo de informação medicamente relevante (Figura 7)

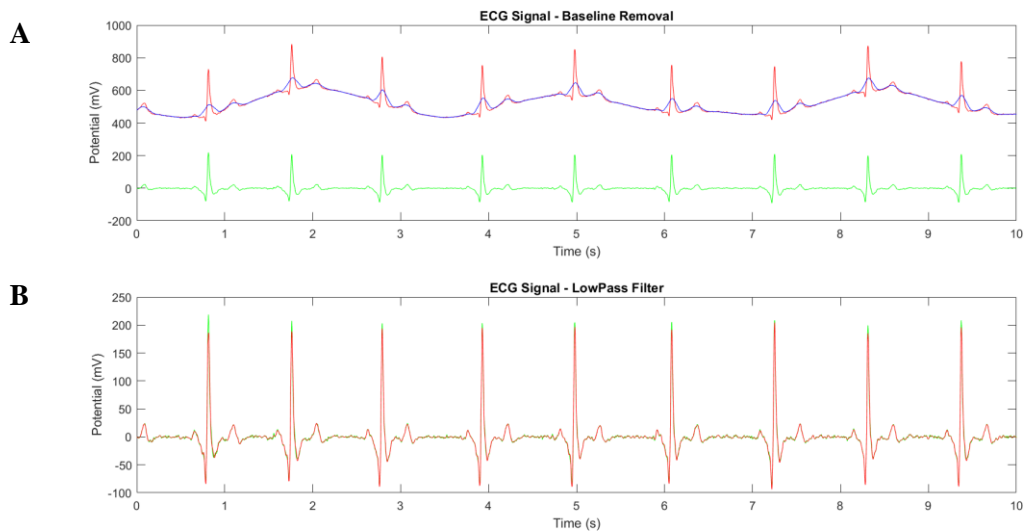


Figura 7. Gráficos exemplificativos do processo interno do algoritmo de remoção da *baseline* (A.) e filtragem das elevadas frequências (B.). Importante realçar que, neste caso concreto foram usadas as definições tomadas como padrão para o processamento do sinal *ECG*. Por sua vez é ainda possível visualizar a redução contínua da amplitude do sinal após cada manipulação do mesmo.

3.3.1.2. Detecção dos picos P e T e complexo QRS

No que se refere à segmentação do sinal e detecção dos picos do *ECG*, procurou-se numa fase inicial implementar o método de *Pan&Tompkins*, que de uma forma breve consiste no isolamento do complexo *QRS* através de um filtro passa-banda com 5 Hz de frequência limitadora inferior e 25 Hz de limitadora superior, seguido de uma diferenciação (intensificação das variações do sinal) e potencialização. Posteriormente foi aplicado um novo filtro *moving-average*, de modo que fosse identificado o ponto de maior energia, ou seja o pico R [10]. Em relação aos restantes picos a sua identificação foi executada através da função *findpeaks()*, proporcionando o aparecimento de erros, especialmente na identificação entre as ondas P e T (também ruído de movimento não eliminado surge à semelhança destas ondas com frequências abaixo dos 5 Hz). Num futuro próximo pretende-se otimizar este processo, garantindo uma maior capacidade de detecção dos pontos característicos do *ECG*.

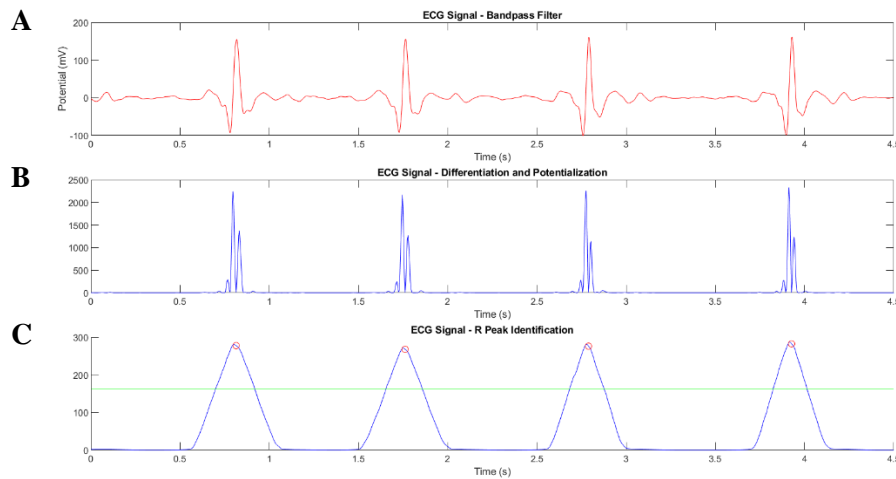


Figura 8. Passos do método *Pan&Tompkins* desenvolvido para detecção dos picos R do complexo QRS. Surgem então três representações gráficas relativas à aplicação do filtro passa-banda padrão sobre o sinal previamente processado (A.), à diferenciação e potencialização do sinal de A. e à aplicação de um filtro *moving-average* sobre o sinal de B. (C.). A procura dos picos do sinal são feitos por identificação do ponto máximo do *ECG* processado entre o ponto inicial e final acima da *threshold* (metade do valor máximo do sinal do gráfico C.), marcado a cor verde.

3.3.1.3. Segmentação de batimentos cardíacos e Batimentos cardíacos por minuto

Partindo do pressuposto que houve uma correta identificação dos picos R, para o cálculo dos batimentos cardíacos por minutos apenas se teve de aplicar a seguinte fórmula (Equação 1):

$$BPM = \frac{N^{\circ} \text{ de Picos R detetados} \times 60}{\text{Tempo de aquisição}} \text{ batimentos por minuto} \quad (1)$$

Para além dessa *feature*, foi possível a segmentação do sinal *ECG* processado, também associado aos tempos entre picos R. Neste caso, considerou-se um batimento individual a junção de duas contribuições: auricular (1/3 dos pontos antes do pico R) e ventricular (2/3 dos pontos depois do pico R). Posto isso, foi simples a sobreposição de todos batimentos numa única representação gráfica, bem como o cálculo da curva *ECG* média para determinada aquisição.

3.3.1.4. Variabilidade taxa cardíaca (HVR – Heart Rate Variability)

Uma das grandes potencialidades do sinal *ECG* é que de forma indireta conseguimos extrair outras métricas informativas do estado de saúde não só cardíaco como do sistema nervoso, por exemplo. Uma das metodologias é o uso de métricas sobre os intervalos entre picos R, relacionadas com a *Heart Rate Variability (HRV)*. As métricas pode ser distinguidas mediante o domínio de atuação: domínio do tempo e domínio da frequência. No primeiro caso são exemplos as seguintes métricas: *MEAN*; *SDNN* (Desvio Padrão dos intervalos RR normais); *SDSD* (Desvio padrão da média dos intervalos RR); *RMSSD* (Raiz quadrada da raiz quadrada média da soma de todas as diferenças entre intervalos RR sucessivos); *NN50* (Número de pares consecutivos de intervalos RR que diferem mais de 50 ms); *pNN50* (Porcentagem da métrica *NN50*). Já no que diz respeito ao domínio espectral, tem-se o *RaLH* (Razão entre as baixas e as altas frequências do *Tachogram*), como principal detetor de um bom balanço entre a atividade nervosa simpática (LF) e parassimpática (HF) (Figura 9) [11,12]. De realçar que, o cálculo das mesmas é executado em funções devidamente geradas para o efeito na diretoria *Matlab\functions\metrics*.

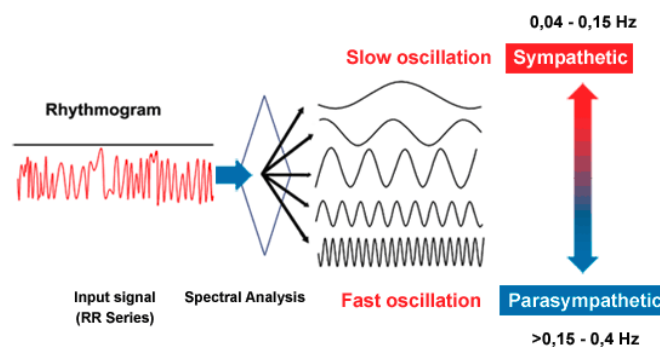


Figura 9. Esquema exemplificativo relativo ao fundamento teórico sobre o qual o estudo no domínio espectral da *HVR* foi executado.

3.3.1.5. Diagnóstico Rudimentar

Para finalizar o processamento de dados, foi, ainda, executado um pequeno diagnóstico não validado do ponto de vista médico. De ter em conta que devido ao facto de unicamente se basear em *thresholds* tabelados, não considerando o historial clínico e pessoal do paciente, esta funcionalidade ainda apresenta uma grande margem de progressão em termos da viabilidade (futuro poderá passar pela incorporação de soluções de aprendizagem computacional).

Assim, pretendeu-se de forma rudimentar detetar eventuais arritmias ou anormalidade no sinal *ECG*. Inicialmente distinguindo entre os estados de bradicardia ($BPM < 60$), taquicardia ($BPM > 100$) e normalidade ($60 \leq BPM \leq 100$), e posteriormente na especificação ou não em Fibrilação Atrial (desvio padrão dos intervalos RR > 0.120 milissegundos) [13]. Adicionalmente, os sinais foram ainda distinguidos tendo em conta o batimento ectópico, ou seja, presença ou ausência de contrações auriculares e ventriculares prematuras [13].

3.3.2. Estrutura de Dados

De modo a garantir que toda a informação seja guardada, logo após o primeiro *login* efetuado, é gerada uma *struct* designada *AllPatients.mat*, devidamente segmentada em vários campos e sub-campos. Numa primeira camada é possível aceder à lista de estruturas associadas a cada paciente que já criou conta. Dentro de cada uma desta, por sua vez, é possível aceder-se à informação do paciente (*PatientInfo*), à lista de aquisições brutas (*ECG_Aquisitions*), à lista todos os parâmetros obtidos pós-processamento para cada aquisição individual (*ECG_Info*), bem como o seu diagnóstico (*ECG_Diagnostics*).

| Field | Value |
|----------|------------|
| Patient1 | 1x1 struct |

| Field | Value |
|-----------------|------------|
| PatientInfo | 1x1 struct |
| ECG_Aquisitions | 1x1 struct |
| ECG_Info | 1x1 struct |
| ECG_Diagnostics | 1x1 struct |

| Field | Value |
|-------------|--------------|
| Name | 'Hugo' |
| Pass | '123' |
| Login | 1 |
| Age | '21' |
| Sex | 'M' |
| DateofBirth | '24/10/2001' |

| Field | Value |
|-----------------|----------------|
| ECG | 1x47865 double |
| Time | 1x47865 double |
| AcquisitionDate | 1x1 datetime |
| AcquisitionFreq | 266 |
| AcquisitionTime | 3 |

| Field | Value |
|------------------|----------------|
| ECG_Processed | 1x47865 double |
| ECG_Energy | 1x47864 double |
| ECG_Peaks | 1x1 struct |
| ECG_Beats_minute | 55.6667 |
| ECG_All_Beats | 167x201 double |
| ECG_Avg_Beat | 1x201 double |
| ECG_Beat_Points | 200 |
| ECG_RR_Intervals | 1x166 double |
| ECG_Metrics | 1x1 struct |

| Field | Value |
|---------------|--|
| Arrhythmia | 'Bradycardia' |
| Ectopic_Beats | 'With 20 Premature Atrial Contraction' |

Figura 10. Exemplificação da estrutura desenvolvida para armazenamento de todos os dados adquiridos e processados para cada aquisição. Ao longo do documento atual, é habitual mencionar esta estrutura através do termo “base de dados” do sistema atual.

Através deste formato e disposição dos dados recolhidos, a manipulação tornou-se mais eficiente e intuitiva. Contudo, o principal ponto negativo, que futuramente poderá ser alvo de otimização, prende-se com o tempo de carregamento da estrutura. Como na fase de testes apenas se trabalhou com um quantidade baixa de dados, não se consegue prever de que forma o sistema irá responder às constantes aberturas e armazenamentos quando este *file* tiver várias aquisições de vários paciente em simultâneo.

3.4. Interface de apresentação de dados: AppDesigner Matlab

De modo a se obter uma visualização mais organizada dos dados brutos e processados em tempo real, foi desenhada uma *GUI* pelo *AppDesigner* do *software Matlab*. Nos tópicos seguintes, será exposta um pouco da interface programada, tendo por base um caso de aquisição real, efetuada pelo sistema que se descreveu ao longo do atual documento. Como anexo a esta explicação, foi também desenvolvido um demo da forma de vídeo (*ECG_App.mp4*), que sintetiza não só as principais funções existentes na *app*, como evidencia uma demonstração de uma aquisição em tempo real.

3.4.1. Login/Sign In

Primeiramente, é importante mencionar que, devido ao baixo conhecimento de programação nesta ferramenta, a interface desenvolvida nada mais é que um conjunto de interfaces isoladas, ligadas entre si, de tal forma que é, unicamente, necessário a abertura do ficheiro “Matlab\app\login_signin.mlapp” (Figura 11). Caso o utilizador tenha conta, irá entrar diretamente nessa, tendo acesso aos seus dados pessoais e medições passadas. Caso a “base de dados” não reconheça o utilizador, é criada uma conta, sendo necessário o utilizador colocar as suas informações pessoais.

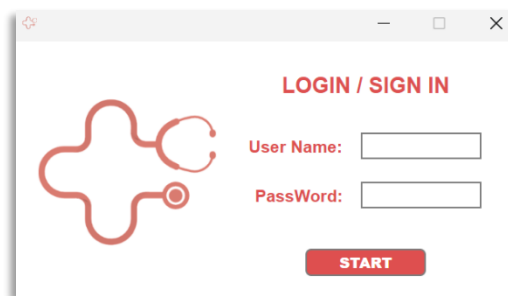



Figura 11. Paine de *Login* ou *Sign In* da *GUI* que serviu de base para o projeto atual.

3.4.2. Interface Principal

Após entrar na sua conta, o utilizador deparar-se-á com o ecrã principal da aplicação (Figura 12). Do lado esquerdo, pode-se visualizar as informações pessoais do *user*, atualizáveis a qualquer altura. Por baixo, temos uma lista com todas as aquisições já realizadas até ao momento do respetivo paciente. Sempre que há uma nova aquisição, esta lista pode ser atualizada clicando no ícone . Já o lado direito, é a secção destinada para o output de informações de pós-processamento e um pequeno diagnóstico rudimentar. No centro, visualiza-se o gráfico onde não só é representado o sinal em tempo real, como as restantes informações que o utilizador deseje observar, após seleção de determinado processo no menu do topo. Por fim, temos os botões que permitem ou começar uma nova aquisição ou carregar uma aquisição da lista de aquisições.

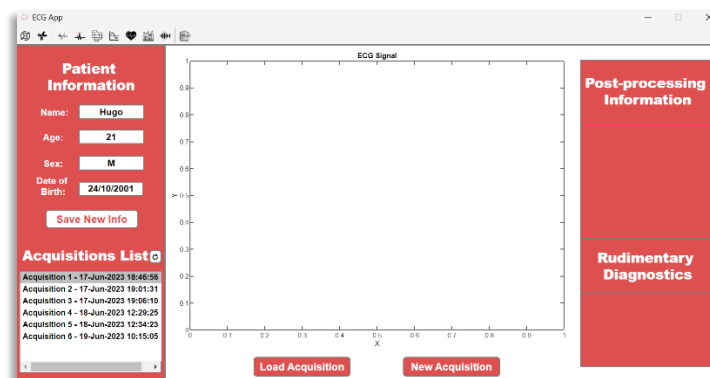


Figura 12. Interface principal da *GUI* que serviu de base para o projeto atual.

3.4.3. Nova Aquisição

Caso se deseje começar uma aquisição nova, deve ser pressionado o botão “New Acquisition”. Após isso, surge um pequeno aviso/demostração de como os elétrodos devem ser colocados por parte do utilizador do equipamento (Figura 13A.). Seguidamente na Figura 13B., surge uma nova janela, na qual se pode definir o tempo de aquisição, a taxa de amostragem, e o modo de visualização (em tempo real ou não) .

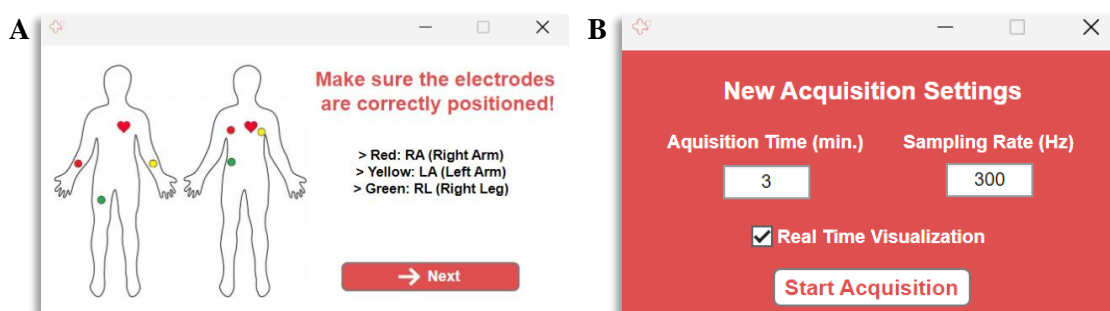


Figura 13. A. Alerta para uma boa colocação do elétrodos por parte do utilizador, antes do início efetivo da aquisição; B. Apêndice da *app* que permite a manipulação do tempo, a frequência de amostragem de cada aquisição, bem como optar pela visualização em tempo real ou não. Após o clique do botão “Start Acquisition”, a informação é enviada pelo *SerialPort* para o microcontrolador, que por consequência inicia a aquisição.

3.4.4. Load de Aquisições

No caso de se ter interesse em carregar uma aquisição mais antiga, o utilizador deve pressionar “Load Acquisition”, surgindo no seu ecrã uma pequena janela, tal como mostra a Figura 14. Com esta opção o *user* tem a capacidade de visualizar o sinal bruto como um todo ou apenas porção do mesmo (geralmente facilita a visualização do comportamento do sinal *ECG* padrão).

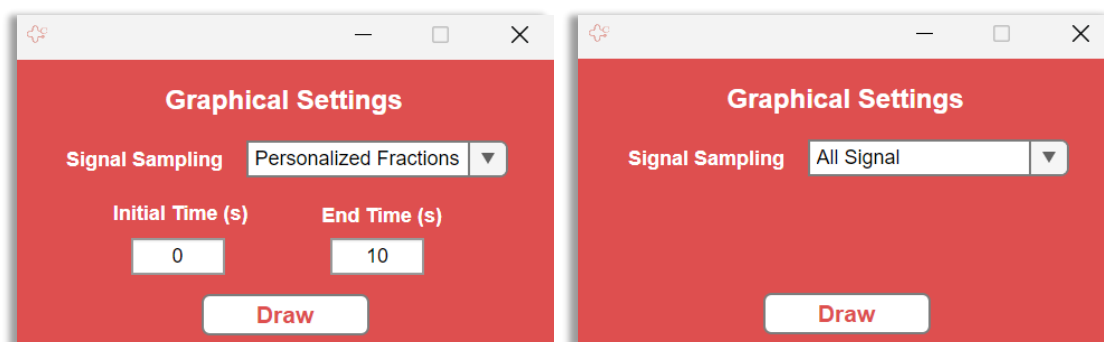


Figura 14. Apêndice da *app* responsável pelo Load de aquisições pré-executadas. De realçar que através da box “Signal Sampling” é possível visualizar o sinal na sua totalidade, ou em pequeno segmentos (geralmente permite uma melhor definição do comportamento do sinal *ECG*).

4. Testagem do sistema

Apesar de teoricamente todos os componentes do sistema estarem interligados entre si e todos eles funcionais, foi necessário iniciar-se uma fase de testes ao dispositivo, garantindo assim a eficiência da abordagem desenvolvida.

Com esse intuito, de seguida, será mostrado um estudo executado sobre um voluntário, à partida sem sinais de cardiopatias associadas. De realçar que o voluntário é do sexo masculino, e tem por hábito a prática de atividade física. Assim, por meio de várias figuras, em seguida será amostrada o conjunto de informação retirada de uma aquisição efetuada pelo dispositivo projetado e desenvolvido. De modo a uma melhor visualização, serão mostrados representações limitadas no domínio do tempo no intervalo [0;10] segundos.

Inicialmente, considerando o registo e *login* efetuado, o voluntário efetuou uma aquisição com frequência de amostragem solicitada de 500 Hz, durante um período de 3 minutos. Seguidamente, após clique no botão de *load*, conseguiu-se observar o sinal bruto associado a curva “sinusoidal” precedentemente explicada (Figura 15). Na ausência de processamento, não se conseguiu retirar nenhuma conclusão, à exceção de que a aquisição apresentou um comportamento semelhante ao esperado, sendo possível visualizar, desde logo, os vários segmentos de um batimento cardíaco (onda P, complexo QRS e onda T).

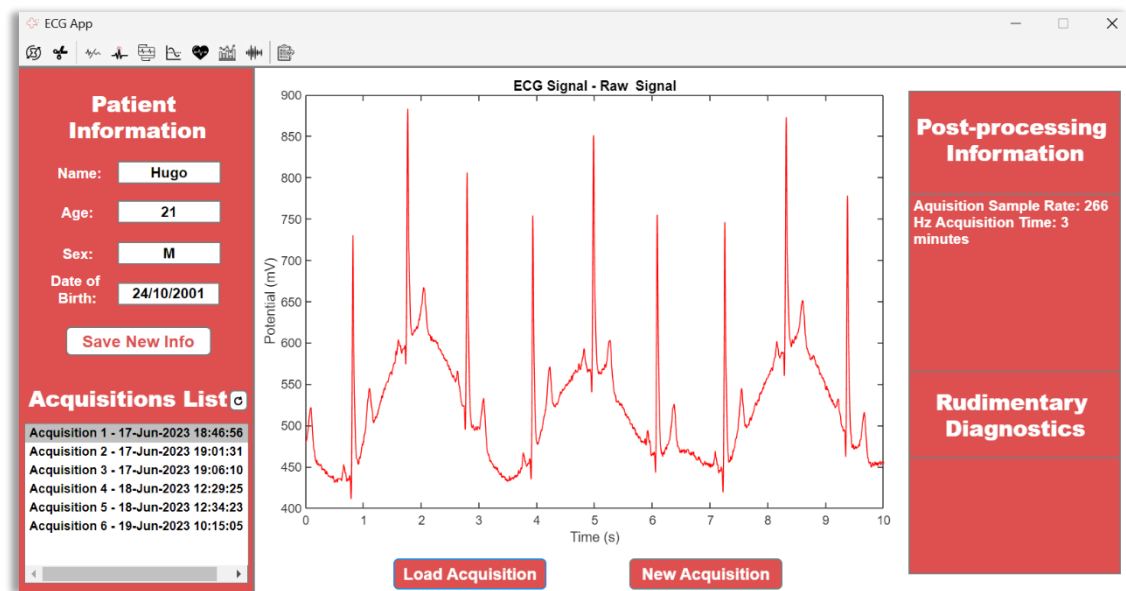


Figura 15. Representação gráfica do sinal atividade cardíaca bruto recolhido pelo sistema descrito ao longo do atual documento, para os primeiro 10 segundos de aquisição. Para além do comportamento sinusoidal do mesmo, destaca-se o valor corrigido da *sampling rate* (em “*Post-processing information*”), segundo os mecanismos referidos anteriormente.

No passo seguinte, desenvolveu-se a remoção da *baseline* e redução do ruído de altas frequências, segundo os parâmetros padrão de processamento. Como evidencia a Figura 16, a remoção da componente *DC* do sinal bruto foi executada com sucesso, obtendo-se assim um curva *ECG*, quase perfeita tendo em conta as condicionantes intrínsecas do projeto. De notar que, após este algoritmo, houve um realce das estruturas supracitadas.

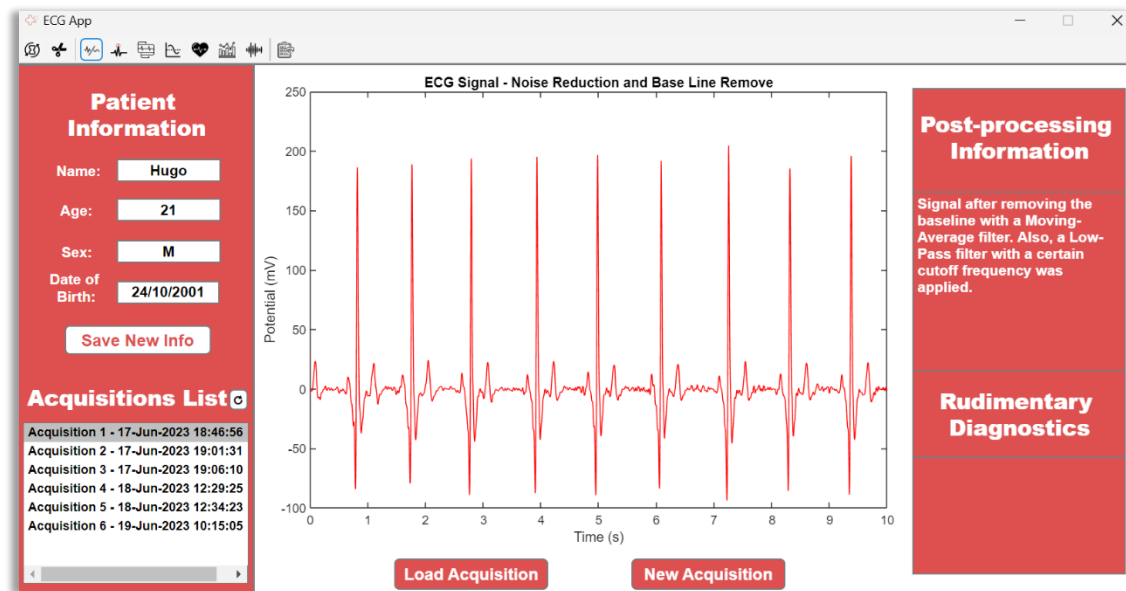


Figura 16. Representação gráfica do *ECG* após remoção da *baseline* e ruído de altas frequências, segundo os algoritmos descritos ao longo do atual documento, para os primeiro 10 segundos de aquisição.

De notar que, um dos pontos chave para a deteção correta dos vários pontos característicos deste biossinal, é a perfeição dos dados recolhidos. Ou seja, basta, por algum motivo (incorreta colocação dos elétrodos, por exemplo), o sinal apresentar a onda P com uma maior preponderância que a onda T, para que o algoritmo falhe completamente na deteção dos mesmos. No entanto, devido à aplicação isolada de uma metodologia mais robusta como o *Pan&Tompkins* sobre os picos R, acredita-se todas as restantes funcionalidades não sejam afetadas, por eventuais erros no procedimento efetuado pela função *findpeaks()*. Neste caso em concreto, na janela de visualização da Figura 17, é possível constatar, que, de facto, a deteção de picos não se revelou totalmente eficaz, na medida em que uma onda P não foi identificada. Apesar disso, é possível entender que os vários segmentos do sinal medidos respeitam, na sua maioria, as proporções padrão, estabelecidas do ponto de vista computacional.

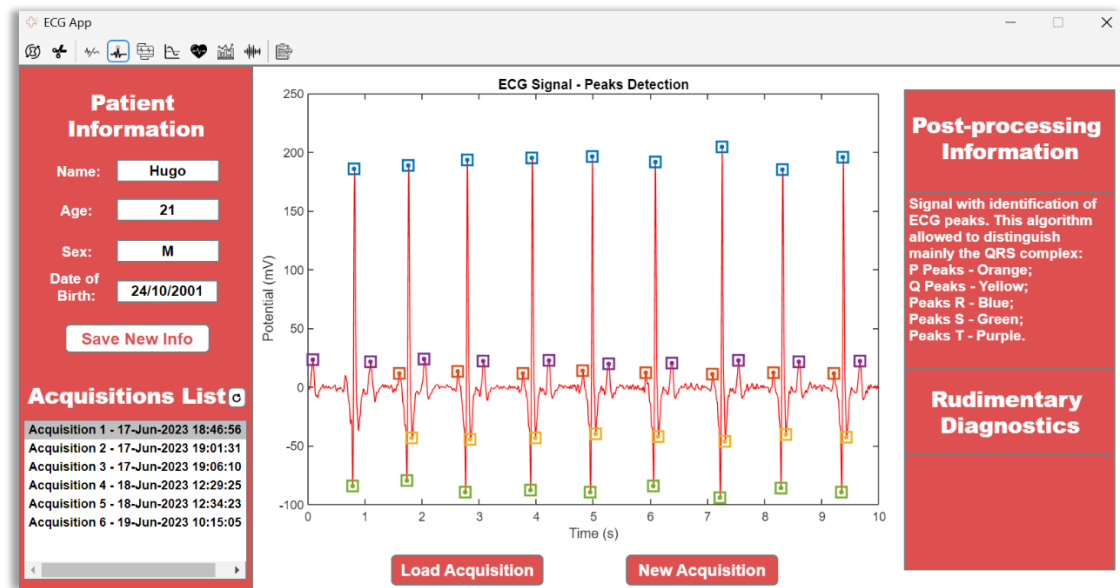


Figura 17. Representação gráfica do *ECG* pré-processado e das respectivas marcações dos pontos característicos do sinal, segundo os algoritmos descritos ao longo do atual documento, para os primeiros 10 segundos de aquisição. De notar que existem falhas associadas à detecção de picos diferentes do pico R, uma vez que a metodologia usada para este último foi diferente.

Na Figura 18, pode-se observar todos os batimentos cardíacos do voluntário de forma sobreposta. Para além de ser possível verificar que o algoritmo de segmentação de eventos cardíacos individuais está efetivamente funcional, em termos de análise médica da informação verifica-se que o voluntário apresenta uma homogeneidade elevada entre os vários segmentos, o que por sua vez, garante uma baixa variabilidade entre batimentos cardíacos ao longo da aquisição. Em alternativa, também, se observou a curva relativa ao batimento cardíaco médio do paciente, tendo-se revelado bastante próximo do que se esperaria para um indivíduo adulto saudável (Figura 19).

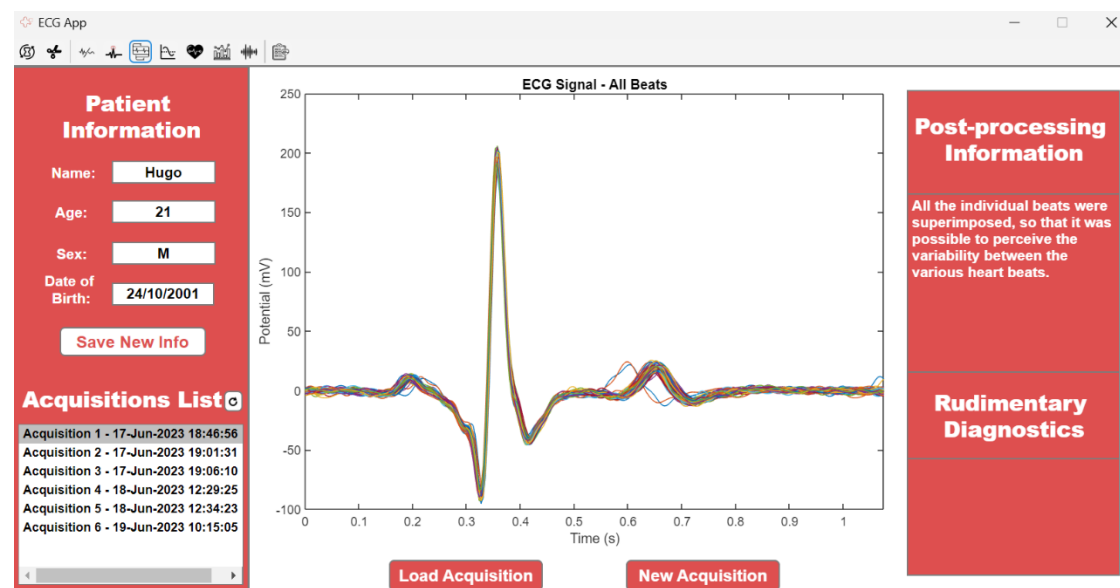


Figura 18. Representação gráfica, em *overlap*, dos vários batimentos cardíacos individuais, segmentados para a aquisição selecionada. Apesar de nem todos os batimentos apresentarem a mesma duração, neste caso a representação é feita tendo por base o valor temporal médio dos intervalos entre picos R.

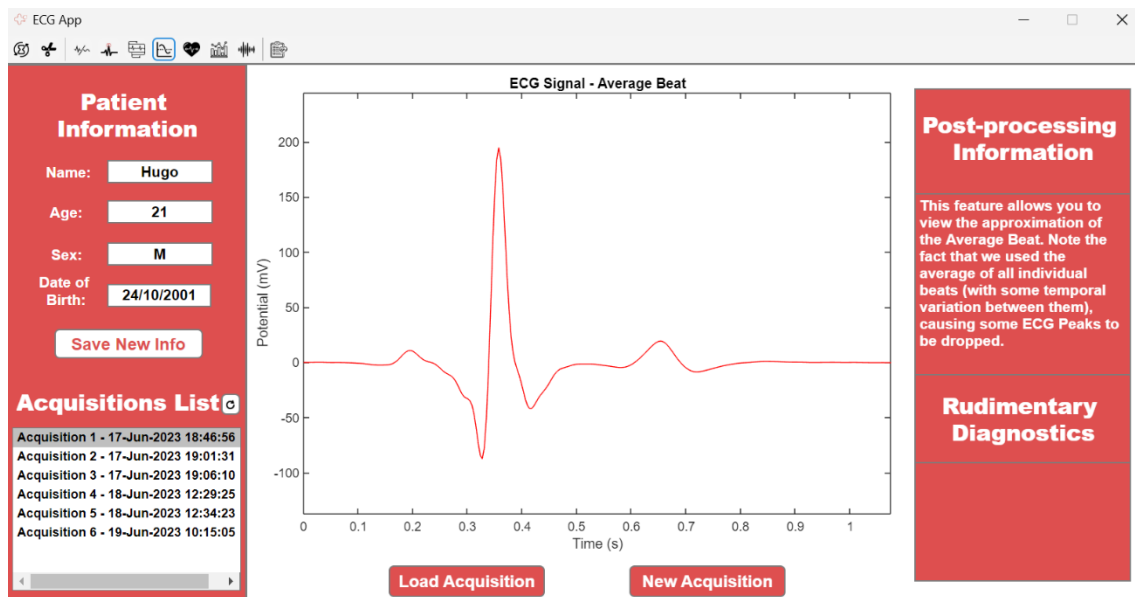


Figura 19. Representação gráfica da média ponto a ponto dos vários batimentos evidenciados na Figura 18.

Já em termos de batimentos cardíacos por minuto, o voluntário apresentou 55 batimentos por minutos. Tendo em conta que os valores normais estão entre os 60 e os 100 batimentos, percebe-se que o sujeito está numa condição de bradicardia (Figura 20). Tendo em conta o contexto clínico apresentado, pensa-se que efetivamente os valores determinados fazem sentido, não sendo espetável erros nesta porção de código desenvolvido.

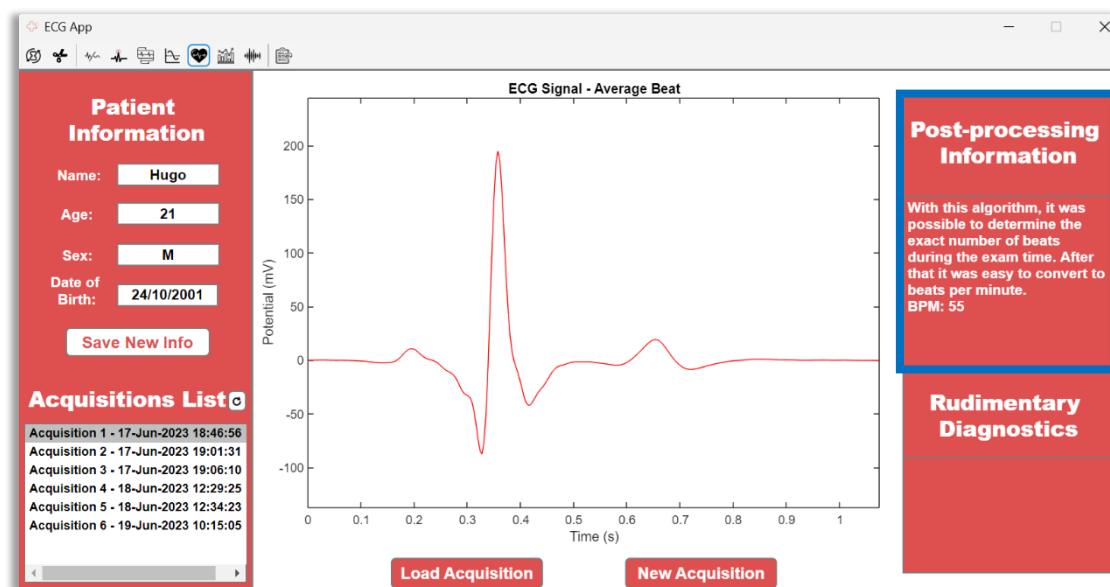


Figura 20. Informação relativa aos batimentos cardíacos por minuto (BPM) calculados para o estudo previamente abordado.

Finalmente, na antepenúltima e na penúltima opção do menu, referentes ao estudo da *Heart Rate Variability (HRV)*, os dados são mais difíceis de validar, visto que será necessário um conjunto elevado de aquisições de vários grupos de pessoas, de modo a entender se estes valores variam

mediante o indivíduo e da forma expectável. Ainda assim, no primeiro caso, foi possível a visualização da variação em torno da média dos intervalos RR do voluntário, juntamente com as métricas mencionadas em tópicos passados (Figura 21). Já no segundo efetuou-se a análise em frequência da *HVR* (Figura 22). Apesar da complexidade do código associado, acredita-se que o mesmo esteja a efetuar corretamente a funcionalidade ambicionada.

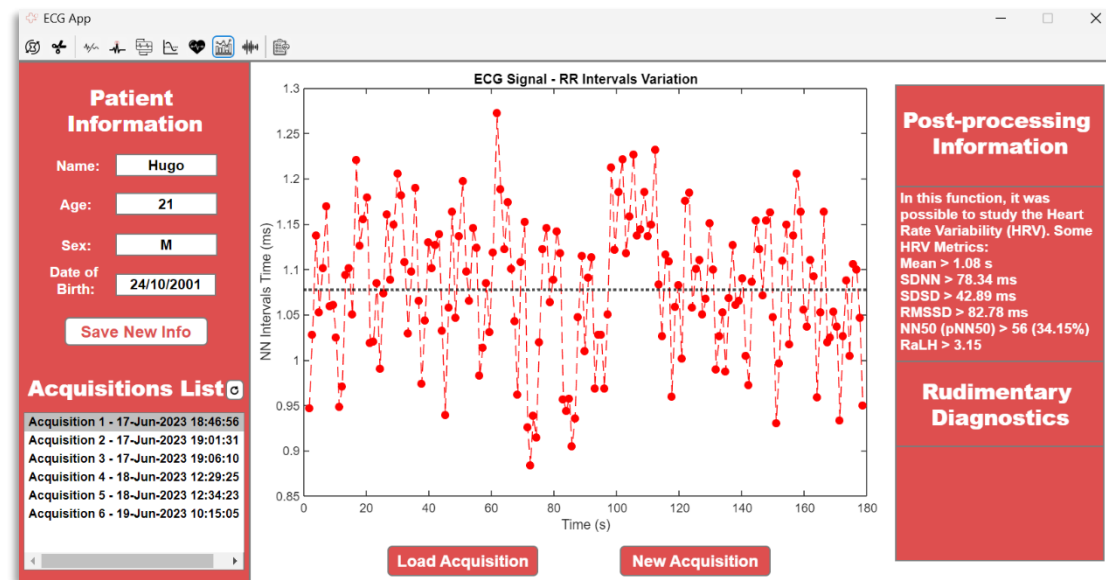


Figura 21. Análise gráfica no domínio temporal da *Heart Rate Variability* (*Tachogram*), bem como por via de métricas associadas aos intervalos RR (métrica *RaLH*, no entanto, é uma métrica recolhida após estudo a nível espectral).

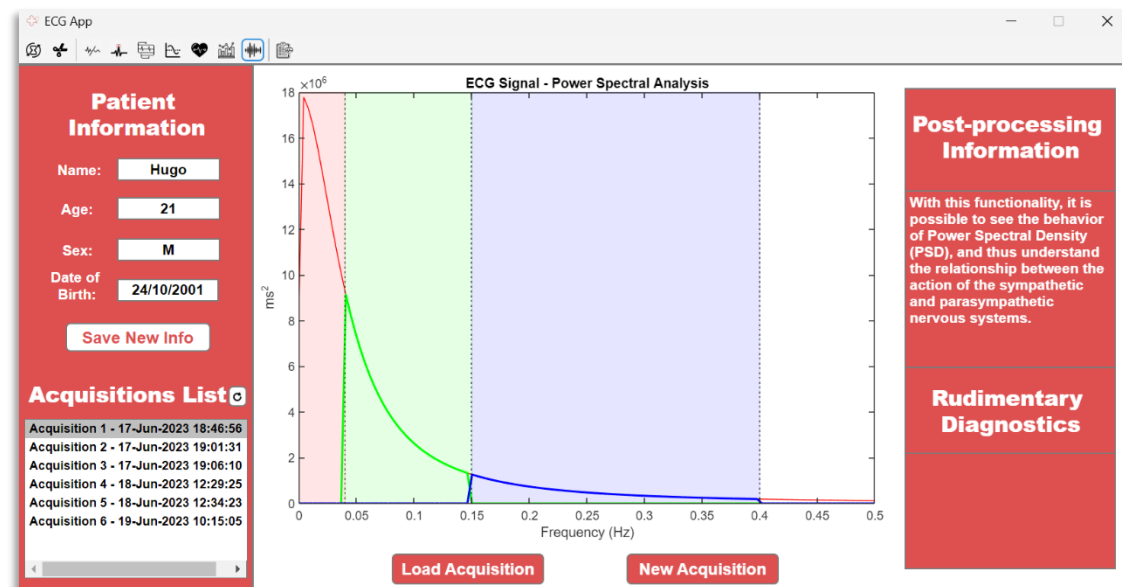


Figura 22. Análise no domínio das frequências do gráfico apresentado na Figura 21. Neste caso é possível delinear 3 porções do sinal, ou seja, a vermelho as *very-low frequencies*, a verde as *low frequencies* e a azul a *high frequencies*. Através da razão entre áreas das duas últimas porções mencionadas, pode-se estabelecer o *RaLH*. Através desta métrica é estabelecido um balanço entre a atuação do sistema nervoso simpático e do sistema nervoso parassimpático.

Por fim, através da última opção, tal como já havia sido mencionado, verifica-se que o doente possui um número reduzido de batimentos cardíacos num minuto, quando comparado com os valores *standard*. No que se refere à PAC ou PVC (*Permatute Atrial* ou *Ventricular Contractions*, respetivamente), a confirmação dos valores é um pouco mais difícil. No entanto, sendo o diagnostico baseado em metodologias de *thresholds*, pensa-se que, à partida, tudo estará correto (Figura 23).

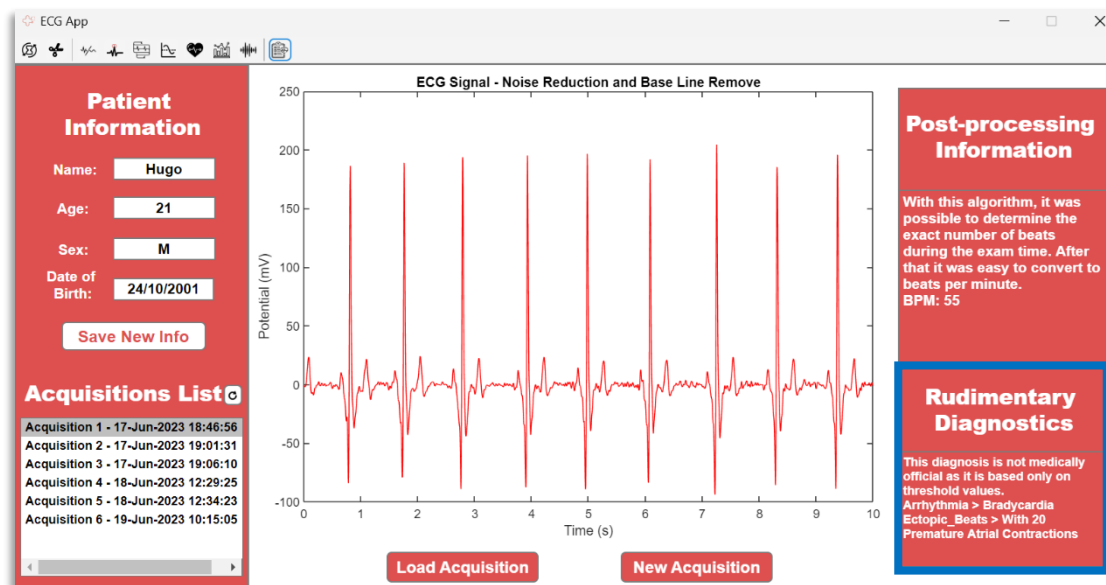


Figura 23. Destaque na funcionalidade de diagnóstico implementada no interface desenvolvida, como extensão do dispositivo físico constantemente descrito ao longo do trabalho atual.

Adicionalmente, como forma de nota, os dois botões iniciais não mencionados do menu do topo da interface, efetuam um novo processamento do sinal segundo um conjunto de parâmetros personalizáveis e corte do sinal numa zona de interesse, respetivamente. Apesar de neste caso concreto não necessitarem de ser usados, após aplicação dos mesmos, estes demonstraram uma atuação eficaz e de acordo com o esperado.

5. Conclusões

No fim do atual projeto, pode-se concluir que o mesmo foi bem conseguido, tendo em consideração o tempo de implementação e os recursos à nossa disposição. Apesar desse facto, existe consciência sobre falhas na implementação da inovação no nosso produto, como foi o caso da ausência de tecnologias *wireless*.

Tendo por base a matriz de risco e o plano de tarefas que se propôs inicialmente (anexados ao documento atual), demonstra-se a realização da montagem do circuito proposto, a criação de códigos em *Arduino IDE* e *Matlab* que permitissem a recolha e tratamento dos dados e a implementação de uma interface gráfica capaz de mostrar toda a informação pretendida ao utilizador de uma forma iterativa. No que se refere aos riscos associados previamente conjecturados, conclui-se que na generalidade dos casos foram superados sem grandes dificuldades associadas. Por sua vez o projeto peca pelo não estabelecimento da comunicação via *wireless* entre o *Arduino MKR1000* e o PC.

Contudo ao longo do trabalho desenvolvido, foram encontradas algumas dificuldades que não tinham sido antecipadas, e, portanto, não surgem mencionada na matriz de risco em anexo. Uma dessas contrariedades passou pela comunicação entre o *Arduino* e o *Matlab*, que inicialmente durante aquisições consecutivas, o sistema colapsava em termos comunicativos. Após várias horas de trabalho, entendeu-se que o problema se devia ao facto de, após a primeira aquisição, o *Arduino* estar a enviar sistematicamente informação pelo *SerialPort*, impedindo assim uma eventual leitura por parte do mesmo de uma mensagem enviada pelo *Matlab*, no momento de uma segunda aquisição. Como solução, tratou-se de se desenvolver um mecanismo capaz de entre aquisições o microcontrolador estar num estado de repouso e à escuta de mensagem.

Uma outra grande dificuldade encontrada passou pela representação em tempo real do sinal *ECG* adquirido, objetivo principal deste projeto. Inicialmente, o *live plot* era extremamente lento, não por motivos relacionados com a aquisição de valores, mas sim por motivos exclusivamente de processamento gráfico. Tendo em conta que as frequências de amostragem praticadas implicaram a representação de um conjunto alargado de dados, mesmo após otimização quase total do processo, continuou-se a evidenciar *delay* na resposta. Por fim, numa última abordagem efetuou-se uma mudança de paradigma, ou seja, passou-se de uma representação ponto a ponto, para uma representação de um conjunto de ponto em bloco. No caso atual, a representação após os $\frac{fs}{4}$ pontos iniciais está a ser feita efetuada ponto a ponto, pelo que em instantes futuros, a representação converte-se num representação de $\frac{fs}{4}$ pontos em $\frac{fs}{4}$ pontos, até ao final do sinal. Através da metodologia conseguiu-se obter uma representação muito perto de ser em tempo real (pode-se comprovar através da perturbação brusca do sinal adquirido pelos elétrodos). Futuramente

acredita-se que a representação possa se tornar mista, ou seja, em regiões com variação da componente y do sinal aproximadamente nula, a representação pode ser feita em bloco, e para zonas de variações elevadas a representação pode ser ponto a ponto. Contudo, não se descarta a possibilidade de se obter melhores resultados com *hardware* de maior capacidade para a tarefa requerida.

Sob forma de complementação, existiram ainda alguns problemas relacionados com o sistema físico, mais propriamente com os elétrodos. Sendo estas estruturas responsáveis pela captação dos potenciais bioelétricos do coração, visualizou-se vários erros de aquisição causados pela má aderência dos elétrodos sobre as regiões predefinidas. Por outro, lado não se percebeu nenhum tipo de problema no sensor usado no protótipo do *gadget* médico.

Concluindo, apesar das dificuldades inesperadas e dos objetivos não cumpridos, pensa-se ter executado com distinção o projeto proposto. Para além disso, o protótipo do instrumento e respetiva interface funcionam como o pretendido e podem ser considerados uma base sólida para desenvolvimentos futuros da temática explorada.

6. Bibliografia

- [1] P. L. Rajbhandary, G. Nallathambi, N. Selvaraj, T. Tran, and O. Colliou, "ECG Signal Quality Assessments of a Small Bipolar Single-Lead Wearable Patch Sensor," *Cardiovasc. Eng. Technol.*, vol. 13, no. 5, pp. 783–796, 2022, doi: 10.1007/s13239-022-00617-3.
- [2] Y. Sattar and L. Chhabra, "Electrocardiogram," *National Center for Biotechnology Information*, 2022. <https://www.ncbi.nlm.nih.gov/books/NBK549803/> (accessed Mar. 08, 2023).
- [3] V. N. Batchvarov, "Analysis and interpretation of the electrocardiogram by the computer," *Int. J. Cardiol.*, vol. 268, no. February, pp. 38–39, 2018, doi: 10.1016/j.ijcard.2018.05.066.
- [4] N. Rafie, A. H. Kashou, and P. A. Noseworthy, "ECG Interpretation: Clinical Relevance, Challenges, and Advances," *Hearts*, vol. 2, no. 4, pp. 505–513, 2021, doi: 10.3390/hearts2040039.
- [5] Analog Devices, "AD8232 Single-Lead ECG," 2013. [Online]. Available: www.analog.com/AD8232
- [6] Arduino Docs, "MKR 1000 WiFi." <https://docs.arduino.cc/hardware/mkr-1000-wifi?selectedStore=eu> (accessed Mar. 12, 2023).
- [7] Zia-ul-Haque, R. Qureshiy, M. Nawazy, F. Y. Khuhawar, N. Tunioz, and M. Uzairx, "Analysis of ECG signal processing and filtering algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 3, pp. 545–550, 2019, doi: 10.14569/IJACSA.2019.0100370.
- [8] C. L. Saver and J. M. Hurray, "Electrocardiogram Monitoring," *AORN J.*, vol. 52, no. 2, pp. 273–290, 1990, doi: 10.1016/s0001-2092(07)68156-3.
- [9] S. V. Por E, Kooten MV *et al.*, "Nyquist–Shannon sampling theorem," *IEEE Trans. Med. Imaging*, vol. 40, no. 9, pp. 2306–2317, 2020, [Online]. Available: http://link.springer.com/10.1007/978-3-319-24574-4_28%0Ahttps://iopscience.iop.org/article/10.1088/0957-0233/22/2/025803%0Ahttp://arxiv.org/abs/1804.03999%0Ahttps://www.jstage.jst.go.jp/article/mrms/20/2/20_mp.2019-0139/_article%0Ahttps://onlinelibrary.wi
- [10] M. A. Z. Fariha, R. Ikeura, S. Hayakawa, and S. Tsutsumi, "Analysis of Pan-Tompkins Algorithm Performance with Noisy ECG Signals," *J. Phys. Conf. Ser.*, vol. 1532, no. 1, 2020, doi: 10.1088/1742-6596/1532/1/012022.
- [11] S. Thanapaisal, S. Mosher, B. Trejo, and K. Robbins, "EEG-Beats: Automated analysis of heart rate variability (HVR) from EEG-EKG," *bioRxiv*, pp. 1–25, 2020, [Online]. Available: <https://doi.org/10.1101/2020.07.21.211862>
- [12] Ansanalysis, "HRV MEASURING PARAMETER," 2023. <https://www.ans-analysis.com/hrv/hrv-measuring-parameter.html> (accessed Jun. 20, 2023).
- [13] M. RakibulIslam, R. Hossain, M. Ziaul Haque Bhuiyan, T. Ahmed Margoob, M. Taslim Reza, and K. Khairul Islam, "Arrhythmia Detection Technique using basic ECG Parameters," *Int. J. Comput. Appl.*, vol. 119, no. 10, pp. 11–15, 2015, doi: 10.5120/21102-3819.

Anexos

I. *Scripts desenvolvidos e funcionamento da GUI implementada*

Para desenvolvimento do trabalho atual, foram elaborados um conjunto de *scripts* com funcionalidades muito particulares, mencionados ao longo do atual relatório. Estes podem ser distinguidos em três tipologias: *Scripts* em *Arduino IDE*; *scripts* de processamento e estabelecimento de comunicação em *Matlab* e *scripts* no formato *.mlapp* relativos à interface gráfica desenvolvida.

Desta forma, os mesmos, seguem anexados ao atual documento, para que possam ser acedidos e consequentemente para que se confirme o seu funcionamento. De destacar que o seu envio será executado através de uma pasta em formato *.zip*, na qual os vários documentos estarão devidamente acondicionados mediante a sua área de atuação.

Para além disso, é ainda anexado um pequeno demo, relativo ao funcionamento da *app*, nomeado como *ECG_App.mp4*.

II. Planeamento de Tarefas (17 de março de 2023)

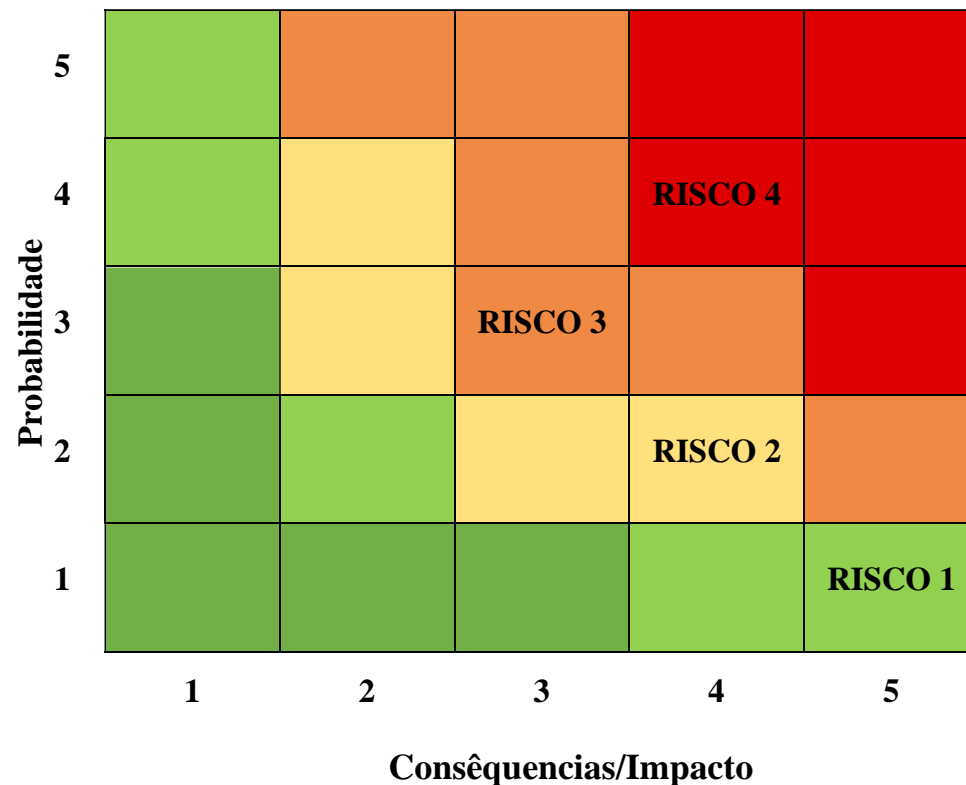
Tabela 2. Diagrama de Gantt relativo ao projeto a ser desenvolvido (a data final do projeto ainda não está fixada, portanto pode haver existir pequenas manipulações futuras do mesmo).

TAREFAS

| Tarefa | Início | Fim |
|--|-----------------|-----------------|
| Realização da montagem digital e física do circuito previamente proposto. | 19 mar. 25 mar. | 26 mar. 1 abr. |
| Criação de um código em Arduino para registo e visualização rudimentar dos dados provenientes dos dois tipos de sensores utilizados (comunicação feita através de ligação direta do microcontrolador ao PC). | 19 mar. 25 mar. | 2 abr. 8 abr. |
| Criação de funções no software, escolhido posteriormente para programação da interface, de processamento do sinal, capaz de identificar condições anómalas no sinal recolhido. | 2 abr. 8 abr. | 9 abr. 15 abr. |
| Testes intermédios ao sistema até então desenvolvido. | 9 abr. 15 abr. | 16 abr. 22 abr. |
| Criação da interface final para visualização de toda a informação que se pretende amostrar. | 16 abr. 22 abr. | 23 abr. 29 abr. |
| Implementação da comunicação via módulo <i>Wi-Fi</i> entre o dispositivo e o PC. | 23 abr. 29 abr. | 30 abr. 6 mai. |
| Testagem da viabilidade da comunicação <i>wireless</i> estabelecida. | 30 abr. 6 mai. | 7 mai. 13 mai. |
| Modulação e Impressão 3D de um suporte para todo o sistema eletrónico até então desenvolvido. | 7 mai. 13 mai. | 14 mai. 19 mai. |
| Integração do sistema final numa camisola, dando origem ao protótipo final. | 14 mai. 19 mai. | - |
| Testes finais, tendo em vista a aprovação final do protótipo. | 14 mai. 19 mai. | - |

III. Identificação dos Riscos (17 de março de 2023)

Tabela 3. Tabela de Riscos associados ao projeto apresentado ao longo do documento.



RISCO 1 – Relativo a potenciais dificuldades na montagem dos circuitos, devido à limitação de conhecimento em eletrônica. **Mitigação:** Pesquisar o máximo de exemplos de circuitos baseados no mesmo conceito que o atual projeto;

RISCO 2 – Associado à dificuldade no desenvolvimento de algoritmos capazes de comparar a curva ECG com a condição normalizada. **Mitigação:** Descobrir valores de *threshold* que permitam a comparação entre um estado patológico e normalizado;

RISCO 3 – Dificuldades associadas ao desenvolvimento de uma interface funcional. **Mitigação:** Usar a linguagem de programação mais direcionada para a finalidade em questão;

RISCO 4 – Estabelecimento incorreto da comunicação via *Wi-Fi* entre o Arduino *MKR1000* e o PC. **Mitigação:** Estudar ao detalhe os protocolos associados e procurar exemplos *openview* de aplicações em Arduino *MKR1000*.