

Projeto de Laboratórios de Telecomunicações e Informática II

SISTEMA DE MONITORIZAÇÃO DE ATIVIDADE FÍSICA

FASE FINAL

Mestrado Integrado em Engenharia de Telecomunicações e Informática
Grupo 7:

- Leandro Henrique Dantas Alves A82157
- André Martins Almeida A82211
- José Eduardo da Silva Santos..... A82350



Índice

Índice.....	2
Índice de figuras	6
1 Introdução.....	8
2 Arquitetura.....	8
3 Sistemas Sensores	9
3.1 Sensores Reais	9
3.1.1 Sensores.....	10
3.1.2 Calibração[5]	11
3.1.3 Relógio	11
3.1.4 Conexão com o concentrador.....	11
3.1.5 Recolha de dados	11
3.2 Sensores Simulados.....	12
3.3 <i>Changelog</i>	13
3.3.1 Fase A	13
3.3.2 Fase B.....	13
4 Concentradores	13
4.1 Protocolo de comunicação com Sistemas Sensores[7].....	13
4.1.1 Mensagem de anúncio (<i>CONN</i>)	14
4.1.2 Mensagem de aceitação (<i>ACCEPT</i>)	14
4.1.3 Mensagem de início (<i>START</i>)	14
4.1.4 Mensagens de dados (<i>DATA</i>).....	15
4.1.5 Mensagens de erro (<i>ERROR</i>).....	16

4.1.6	Mensagem de paragem (<i>STOP</i>)	16
4.2	Configuração	16
4.3	Tratamento de dados	18
4.4	Controlo de erros	18
4.5	Armazenamento em <i>log</i>	19
4.6	Interface de utilização	20
4.6.1	Identificações de Sujeitos e de Sistemas Sensores	21
4.6.2	Controlo dos sistemas sensores	22
4.6.3	Mensagens em tempo real	23
4.6.4	Leitura de dados armazenados	24
4.6.5	Apresentação e ocultação das mensagens em tempo real	24
4.7	<i>Changelog</i>	25
4.7.1	Fase A	25
4.7.2	Fase B	25
4.7.3	Final	25
5	Gestor de Serviço	26
5.1	Protocolo de comunicação com Concentradores[8]	26
5.1.1	Mensagem de conexão (<i>CONN</i>)	26
5.1.2	Mensagem de início (<i>START</i>)	26
5.1.3	Mensagens de dados (<i>DATA</i>)	27
5.1.4	Mensagens de erro (<i>ERROR</i>)	28
5.1.5	Mensagem de paragem (<i>STOP</i>)	28
5.1.6	Mensagens de pedido de lista de ISu (<i>USER</i>) e de envio da mesma	29
5.2	Configuração	29

5.3	Condições/comportamentos físicos	30
5.3.1	Normal – Parado	30
5.3.2	Normal – Deitado	30
5.3.3	Normal – Andamento.....	30
5.3.4	Normal – Correr	30
5.3.5	Alarme – Queda	30
5.3.6	Alarme – Agitado.....	31
5.3.7	Alarme – Inativo	31
5.4	Base de dados	31
5.4.1	Armazenamento das mensagens.....	31
5.4.2	Armazenamento dos comportamentos	32
5.5	Interface de utilizador	33
5.5.1	Áreas.....	34
5.5.2	Mensagens em tempo real.....	34
5.5.3	Leitura da base de dados.....	35
5.5.4	Apresentação e ocultação das mensagens em tempo real.....	36
5.6	<i>Changelog</i>	38
5.6.1	Fase B.....	38
5.6.2	Fase C.....	38
5.6.3	Final	38
6	Sistema Central	38
6.1	Protocolo aplicacional de comunicação.....	38
6.1.1	Registo de serviço	38
6.1.2	Registo de utilizador.....	38

6.1.3	Mensagens de monitorização.....	39
6.2	Base de dados relacional	39
6.2.1	Serviços.....	39
6.2.2	Utilizadores	39
6.2.3	Acessos	40
6.2.4	Pacientes.....	40
6.2.5	Mensagens do Gestor de Serviço	40
6.3	Estatísticas	41
6.4	Interface do utilizador	41
6.4.1	Início de sessão (<i>login</i>)	41
6.4.2	Dados estatísticos do paciente	42
6.4.3	Lista de pacientes.....	43
6.4.4	Registo de utilizadores	44
6.4.5	Estatísticas gerais	45
6.4.6	Edição e remoção de utilizadores	46
6.5	<i>Changelog</i>	47
6.5.1	Fase C.....	47
6.5.2	Final	47
7	Conclusões	49
8	Referências.....	50

Índice de figuras

Figura 2.1 – Arquitetura do sistema.....	8
Figura 3.1 – Ligações físicas do sistema sensor real.	9
Figura 3.2 – Funcionamento de um acelerómetro piezoelétrico[4].....	10
Figura 3.3 – Exemplo de ficheiro lido por um sistema sensor simulado.	12
Figura 4.1 – Estrutura da mensagem de anúncio.	14
Figura 4.2 – Estrutura da mensagem de aceitação.....	14
Figura 4.3 – Estrutura da mensagem de início.	14
Figura 4.4 – Estrutura da mensagem de dados.	15
Figura 4.5 – Estrutura da mensagem de erro.	16
Figura 4.6 – Estrutura da mensagem de paragem.....	16
Figura 4.7 – Ficheiro setup .cfg do concentrador.	17
Figura 4.8 – Ficheiro users .cfg do concentrador.	17
Figura 4.9 – Ficheiro setup .log do concentrador.	19
Figura 4.10 – Ficheiro data .log do concentrador.	20
Figura 4.11 – Ficheiro error .log do concentrador.	20
Figura 4.12 – Ficheiro input .log do concentrador.	20
Figura 4.13 – Menu do concentrador.	21
Figura 4.14 – Lista de sujeitos do concentrador.....	21
Figura 4.15 – Mensagem <i>START</i> enviada pelo concentrador.....	22
Figura 4.16 – Mensagem <i>STOP</i> enviada pelo concentrador.	22
Figura 4.17 – Mensagem <i>ACCEPT</i> para dois sistemas sensores.	23
Figura 4.18 – Mensagens <i>DATA</i> enviadas por dois sistemas sensores.	23
Figura 4.19 – Mensagem <i>ERROR</i> enviada por um sistema sensor.	24
Figura 4.20 – Leitura das mensagens de dados armazenadas em <i>log</i>	24
Figura 5.1 – Estrutura da mensagem de conexão.	26
Figura 5.2 – Estrutura da mensagem de início.	27
Figura 5.3 – Estrutura da mensagem de dados.	27
Figura 5.4 – Estrutura da mensagem de erro.	28
Figura 5.5 – Estrutura da mensagem de paragem.....	28
Figura 5.6 – Ficheiro setup .cfg do gestor de serviço.....	29
Figura 5.7 – Ficheiro areas .cfg do gestor de serviço.....	30
Figura 5.8 – Ficheiro messagedb.txt do gestor de serviço.	32
Figura 5.9 – Ficheiro conditiondb.txt do gestor de serviço.	33
Figura 5.10 – Menu do gestor de serviço.	33
Figura 5.11 – Lista de sujeitos do gestor de serviço.	34
Figura 5.12 – Mensagens em tempo real enviadas pelo concentrador.....	35
Figura 5.13 – Leitura das mensagens de comportamentos armazenadas na base de dados e filtradas por área.....	36
Figura 5.14 – Leitura das mensagens de comportamentos armazenadas na base de dados e filtradas por ISu.	36

Figura 5.15 – Visualização de mensagens em tempo real com valores dos sensores.	37
Figura 6.1 – UI de início de sessão do Super Administrador.	41
Figura 6.2 – E-mail enviado para reposição da palavra-passe.	42
Figura 6.3 – Visualização da página de paciente pelo próprio.	42
Figura 6.4 – Filtragem dos registos comportamentais a partir do gráfico de barras.	43
Figura 6.5 – Visualização da lista de pacientes por um administrador regular.	43
Figura 6.6 – Visualização da lista de administradores e pacientes pelo Super Administrador.	44
Figura 6.7 – Registo de um paciente.	44
Figura 6.8 – E-mail de boas vindas enviado ao utilizador que acabou de ser registado.	45
Figura 6.9 – Registo de um administrador.	45
Figura 6.10 – Estatísticas gerais.	46
Figura 6.11 – Modificação dos dados de um paciente por um administrador.	46
Figura 6.12 – Modificação dos serviços de um administrador pelo Super Administrador.	47
Figura 6.13 – Alteração de palavra-passe.	47

1 Introdução

Esta fase final consiste na correção de erros, preenchimento de lacunas e da complementação de todo o *hardware* e *software* desenvolvidos nas fases anteriores, assim como na implementação de novas funcionalidades opcionais com o intuito de valorizar o trabalho.

Neste relatório está abrangida uma compilação dos relatórios das fases anteriores, juntamente com as alterações e novas implementações feitas em cada uma e os motivos das mesmas.

2 Arquitetura

Este projeto foi desenvolvido através de uma arquitetura hierárquica Cliente-Servidor constituída por Sistema Central > Gestores de Serviços > Concentradores > Sistemas Sensores (reais e simulados) e representada na Figura 2.1.

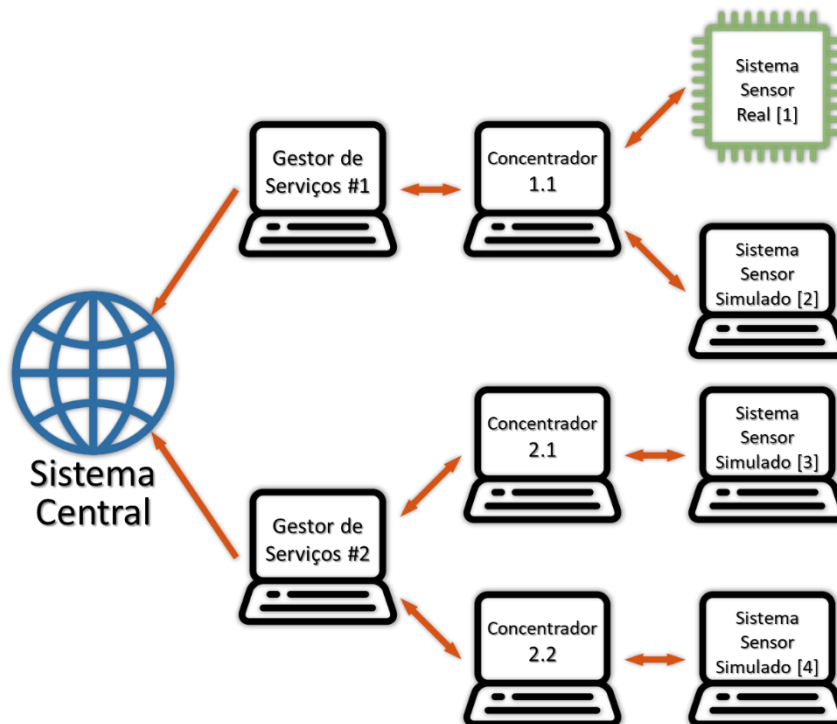


Figura 2.1 – Arquitetura do sistema.

A figura apresentada representa a arquitetura usada nos testes finais que serão demonstrados nos capítulos seguintes.

3 Sistemas Sensores

O sistema sensor é o sistema responsável por adquirir mensagens de dados obtidos pelos sensores integrados e enviá-las para um concentrador no ritmo definido pelo mesmo.

3.1 Sensores Reais

Foi usado apenas um sistema sensor real, tendo o seu *hardware* como placa de microcontrolador um módulo Arduino ESP32[1] ligado a um dispositivo sensor MPU-6050[2] através da interface i2C com uma biblioteca própria[3], que integra um acelerômetro de 3 eixos, um giroscópio de 3 eixos e um sensor de temperatura. A alimentação é feita através de uma *powerbank* de 5 V.

As ligações dos pinos de cada um dos dispositivos está apresentada na Figura 3.1.

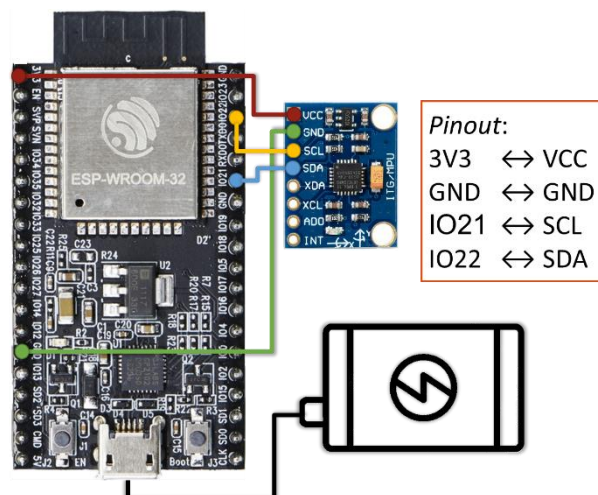


Figura 3.1 – Ligações físicas do sistema sensor real.

Optamos por usar Wi-Fi como tecnologia de ligação escolhida, pois como o sistema sensor deve ser um dispositivo *wearable*, o mesmo deve ser portátil, ou seja, necessita de ligação sem fios. Além disso esta tecnologia permite que seja usada a interface UDP/IP (*User Datagram Protocol / Internet Protocol*), que tem de ser usada pelo menos nos sistemas sensores simulados, ou seja permite que as interfaces e protocolos de comunicação dos dois tipos de sistemas sensores sejam idênticos.

Este sistema sensor está identificado por um ISS (Identificação de Sistema Sensor) cuja *tag* foi definida *hardcoded* no código do Arduino como '1'.

3.1.1 Sensores

Os valores dos eixos adquiridos pelos diversos sensores são obtidos em formato *raw* que são posteriormente convertidos pelo concentrador.

3.1.1.1 Acelerómetro

O sensor acelerómetro é responsável por medir a aceleração a três dimensões através dos eixos X, Y e Z numa escala de $\pm 2 \text{ g}$ ($1 \text{ g} = 9,80665 \text{ m/s}^2$), sendo os valores *raw* obtidos 16 384 LSB/g (*Least Significant Bit / standard gravity*).

Este tipo de sensor obtém os dados dos três eixos de acordo com a Figura 3.2.

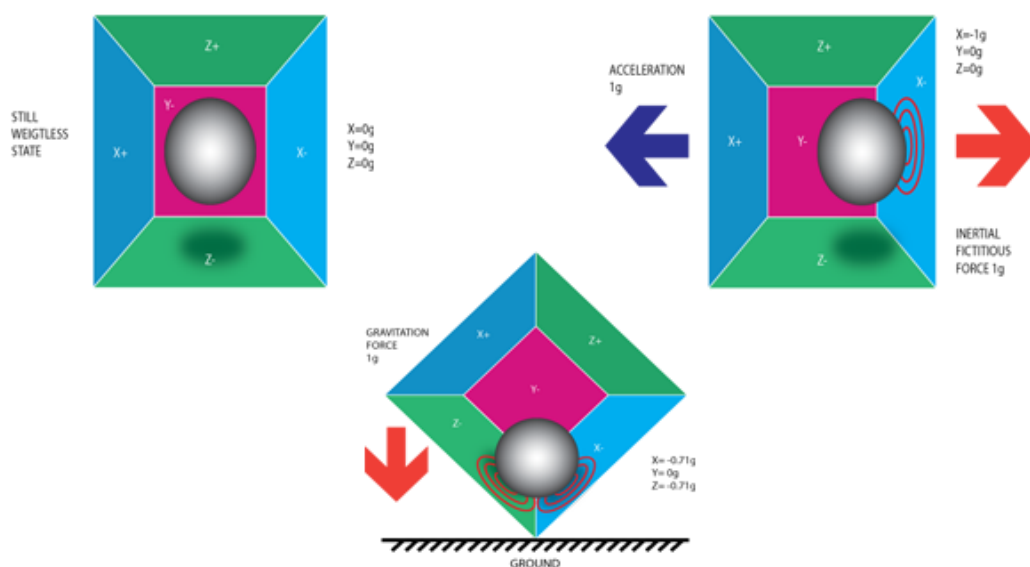


Figura 3.2 – Funcionamento de um acelerómetro piezoelétrico[4].

Quando o dispositivo está em repouso, este deve assumir os valores (0,00; 0,00; 1,00) g pois a única aceleração aplicada é a gravítica no eixo Z.

3.1.1.2 Giroscópio

O sensor giroscópio é responsável por medir a velocidade angular a três dimensões através dos eixos X, Y e Z numa escala de $\pm 250 \text{ }^\circ/\text{s}$, sendo os valores *raw* obtidos 131 LSB/($^\circ/\text{s}$).

Quando o dispositivo está em repouso, este deve assumir os valores (0,00; 0,00; 0,00) $^\circ/\text{s}$ pois o mesmo não está em rotação.

3.1.1.3 Temperatura

O sensor de temperatura é responsável por medir a temperatura interna do dispositivo numa escala de -40 a $+85 \text{ }^\circ\text{C}$, sendo os valores *raw* obtidos 340 LSB/ $^\circ\text{C}$ - 35.

3.1.2 Calibração[5]

Durante a inicialização do *software* do Arduino, é efetuada a calibração dos eixos de cada sensor de forma a que em repouso, os valores dos eixos do acelerómetro sejam (0,00; 0,00; 1,00) g e do giroscópio sejam (0,00; 0,00; 0,00) °/s.

Durante o seu processamento, é pedido que o sensor esteja em repouso. Depois, o Arduino recolhe 1000 amostras, calcula o valor médio dos eixos sensores acelerómetro e giroscópio e vai alterando o *offset* e recolhendo outras 1000 amostras até que os seus valores sejam (1,00; 0,00; 0,00) g e (0,00; 0,00; 0,00) °/s (em valores *raw* com uma tolerância de 10 LSB/g \approx 0,005 986 g e 2 LSB/(°/s) \approx 0,015 267 °/s) respetivamente.

3.1.3 Relógio

Como o *hardware* do Arduino não providencia um relógio com o tempo atual, este é obtido através de uma biblioteca específica[6] que vai buscá-lo à Internet através de Wi-Fi.

3.1.4 Conexão com o concentrador

Depois de calibrado, o Arduino conecta-se a uma rede Wi-Fi cujo endereço IP, porta e credenciais estão indicadas *hardcoded*. Depois disso, tenta estabelecer um *handshake*, ficando constantemente a enviar uma mensagem de anúncio contendo o seu ISS até que um concentrador responda com uma mensagem de aceitação.

3.1.5 Recolha de dados

Depois de conectado a um concentrador, o Arduino espera por uma mensagem de início que inclua os parâmetros PM (período de mensagem em milissegundos), PA (período de amostragem em microssegundos) e NS (*number of samples* – número de amostras). Os parâmetros que forem enviados com valores não positivos terão os mesmos substituídos pelos valores por defeito *hardcoded* no código Arduino, PM = 1000 ms, PA = 1000 μ s e NS = 5 amostras.

Estes parâmetros servem para estabelecer o ritmo de acumulação dos dados obtidos pelos sensores numa mensagem de dados e envio da mesma para o concentrador.

Esta recolha é suspensa quando o Arduino recebe uma mensagem de paragem, que pode ser retomada com uma nova mensagem de início.

3.2 Sensores Simulados

O sistema sensor simulado consiste numa aplicação *software* desenvolvida em linguagem C num computador à parte.

Esta aplicação tem um código semelhante ao do Arduino do sistema sensor real, contudo os valores dos eixos dos diferentes tipos de sensores são obtidos através de um ficheiro (`simuldb.txt`) preenchido pelos concentradores que contém valores *raw* dos eixos de várias amostras obtidas pelo sistema sensor real, exemplificado na Figura 3.3, portanto o processo de calibração é desnecessário. Além disso, em vez do tempo atual ser obtido através da Internet, este é obtido pelo sistema operativo do computador.

Acelerómetro			Giroscópio			Temperatura
x	y	z	x	y	z	
-4916	-14608	-7380	-8134	2471	-2895	-5152
-4504	-14300	-7388	-8149	2848	-2865	-5184
-4600	-14652	-7752	-8151	2749	-2841	-5136
-4492	-14684	-7808	-8299	3139	-2821	-5168
-4568	-14332	-7376	-7910	3009	-2484	-5152
-4564	-14668	-7324	-7981	3333	-2436	-5168
-4560	-14576	-7560	-7573	3059	-2485	-5168
-4492	-14732	-7752	-7361	3042	-2822	-5168
-4948	-14964	-7380	-7091	3183	-2845	-5200
-5180	-14720	-7060	-6978	2692	-2510	-5168
-5472	-14316	-7156	-6350	2492	-2252	-5200
-5044	-14340	-7748	-6044	2571	-2307	-5152
-4864	-14460	-8248	-6200	2348	-1984	-5184
-4992	-14460	-7876	-6039	1762	-2031	-5184
-4804	-14340	-8268	-6113	1451	-2046	-5168
-4520	-14340	-8488	-6116	1279	-2044	-5168
-4592	-13956	-8452	-6034	1189	-1929	-5168
-4732	-13780	-8360	-6202	1185	-2063	-5184
-4232	-13716	-8376	-5798	1196	-1912	-5152
-4456	-13884	-8760	-5942	1210	-1826	-5152
-3948	-14600	-8564	486	-860	105	-5168
-3884	-14368	-8468	499	-885	-83	-5168
-3764	-14344	-8324	852	-840	274	-5168
-4140	-13812	-8348	687	-828	370	-5136
-4128	-13928	-8388	1036	-836	567	-5168
-4212	-13856	-8360	896	-864	497	-5168
-4224	-13456	-8752	1011	-870	712	-5184
-4216	-13596	-8356	1382	-661	942	-5184
-3988	-13264	-8712	1191	-681	1174	-5152
-4016	-13332	-8404	1559	-755	1419	-5152
-4084	-13164	-8748	1660	-1147	1684	-5184
-4028	-13100	-8708	1484	-1308	2165	-5168
-4048	-13136	-8744	1833	-1544	2329	-5168
-4368	-12920	-8596	1899	-1508	2258	-5168

Figura 3.3 – Exemplo de ficheiro lido por um sistema sensor simulado.

Foram usados três sistemas sensores simulados, identificados pelos ISS '2', '3' e '4' com os parâmetros PM, PA e NS distintos.

3.3 Changelog

3.3.1 Fase A

- Implementação do sistema sensor real.

3.3.2 Fase B

- Implementação do sistema sensor simulado;
- Reposicionamento do sensor:
 - Em repouso, a gravidade é atuada no eixo Z em vez de X;
- Aceleração do processo de calibração do sistema sensor real:
 - As tolerâncias foram alteradas de 8 LSB/g e 1 LSB/(°/s) para 10 LSB/g e 2 LSB/(°/s);
- Otimização da recolha de dados dos sensores de modo que seja possível adquirir mensagens nos parâmetros $PM = PA \times 1000$ e $NS = 1$;
- Correção do envio constante de mensagens de erro idênticas quando basta uma.

4 Concentradores

O concentrador é o meio de interação entre o utilizador e os sistemas sensores e é responsável por receber, enviar e processar dados do mesmo. Para tal, foi necessário o desenvolvimento de diversos módulos, um para tratamento das comunicações com o sistema sensor, outro para a gestão de dados (tratamento de dados, controlo de erros e armazenamento em ficheiros de *log*), outro para interface de utilização (para inserir comandos e visualizar dados) e outro para o arranque do sistema consultando um ficheiro de configuração.

O desenvolvimento do seu *software* consistiu num programa de código escrito em linguagem C pois entre as duas opções disponíveis (C e C++), a primeira tornou-se na melhor devido a conhecimento e experiência com a mesma em UCs (Unidades Curriculares) de semestres anteriores, nomeadamente Métodos de Programação I e II (1.º ano) e Sistemas Operativos (3.º ano). A escolha desta linguagem para os sistemas sensores simulados deveu-se ao mesmo motivo.

4.1 Protocolo de comunicação com Sistemas Sensores[7]

A comunicação entre cada sistema sensor (real ou simulado) e o concentrador é feita ao bit através do protocolo UDP/IP, sendo o sistema sensor o cliente e o concentrador o servidor.

Quando o concentrador recebe uma mensagem, o endereço origem da mesma é guardado numa lista de endereços indexada pelo respetivo ISS. Isto serve para que no envio de mensagens para um sistema sensor seja possível configurar o endereço destino do mesmo.

4.1.1 Mensagem de anúncio (*CONN*)

Sistema Sensor → Concentrador

Sempre que um sistema sensor se tenta conectar a um concentrador, este envia constantemente uma mensagem do tipo *CONN* com a estrutura da Figura 4.1 até que um concentrador se conecte e lhe responda com uma aceitação.

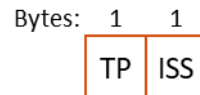


Figura 4.1 – Estrutura da mensagem de anúncio.

- TP ∈ [-128, 128] – tipo de mensagem (1 = *CONN*);
- ISS ∈ [0, 255] – Identificação de Sistema Sensor.

4.1.2 Mensagem de aceitação (*ACCEPT*)

Concentrador → Sistema Sensor

Quando o concentrador receber uma mensagem de anúncio, este guarda o endereço e a identificação do sistema sensor e responde ao mesmo com uma mensagem do tipo *ACCEPT* com a estrutura da Figura 4.2 para que este pare de enviar mensagens de anúncio.



Figura 4.2 – Estrutura da mensagem de aceitação.

- TP ∈ [-128, 128] – tipo de mensagem (2 = *ACCEPT*).

4.1.3 Mensagem de início (*START*)

Concentrador → Sistema Sensor

Sempre que o utilizador pretende iniciar a recolha de dados do sensor, este deve enviar a partir do controlador para o sistema sensor simulado uma mensagem do tipo *START* com a estrutura da Figura 4.3.

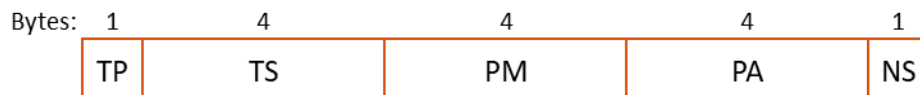


Figura 4.3 – Estrutura da mensagem de início.

- $TP \in [-128, 128]$ – tipo de mensagem (3 = *START*);
- $TS \in [0, 4\ 294\ 967\ 295]^1$ – *timestamp* em formato Unix do momento de envio da mensagem;
- $PM \in [0, 4\ 294\ 967\ 295]$ – período (ms) definido entre cada mensagem recebida;
- $PA \in [0, 4\ 294\ 967\ 295]$ – período (μ s) definido entre cada amostra recolhida;
- $NS \in [0, 255]$ – número definido de amostras recolhidas por mensagem.

4.1.4 Mensagens de dados (*DATA*)

Sistema Sensor \rightarrow Concentrador

Sempre que é atingido o PM (período entre mensagens de dados) definido, o concentrador deverá receber uma mensagem do sistema sensor simulado do tipo *DATA* com a estrutura da Figura 4.4.

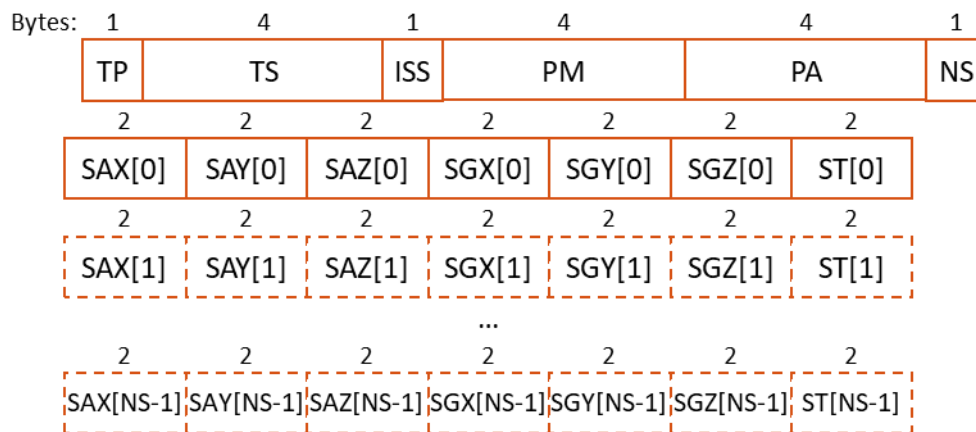


Figura 4.4 – Estrutura da mensagem de dados.

- $TP \in [-128, 127]$ – tipo de mensagem (4 = *DATA*);
- $TS \in [0, 4\ 294\ 967\ 295]$ – *timestamp* do momento da recolha da primeira amostra presente na mensagem;
- $ISS \in [0, 255]$ – Identificação de Sistema Sensor;
- $PM \in [0, 4\ 294\ 967\ 295]$ – período entre cada mensagem enviada;
- $PA \in [0, 4\ 294\ 967\ 295]$ – período entre cada amostra recolhida;
- $NS \in [0, 255]$ – número de amostras recolhidas por mensagem;
- $SAX[NS]$, $SAY[NS]$, $SAZ[NS] \in [-32\ 768, 32\ 767]$ – *arrays* de valores *raw* dos eixos X, Y e Z obtidos pelo acelerómetro das NS amostras recolhidas;
- $SGX[NS]$, $SGY[NS]$, $SGZ[NS] \in [-32\ 768, 32\ 767]$ – *arrays* de valores *raw* dos eixos X, Y e Z obtidos pelo giroscópio das NS amostras recolhidas.
- $ST[NS] \in [-32\ 768, 32\ 767]$ – *array* de valores *raw* das temperaturas recolhidas nas NS amostras.

¹ $TS \in [1970-01-01\ 00:00:00, 2106-02-07\ 06:28:15]$

4.1.5 Mensagens de erro (*ERROR*)

(Sistema Sensor) → Concentrador

Sempre que o sistema sensor simulado pretende indicar uma condição de erro ao concentrador, é enviada uma mensagem do tipo *ERROR* com a estrutura da Figura 4.5.

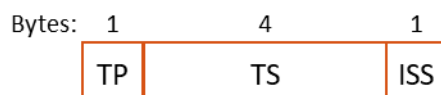


Figura 4.5 – Estrutura da mensagem de erro.

- TP ∈ [-128, 127] – tipo de mensagem (-2 = *ERROR*);
- TS ∈ [0, 4 294 967 295] – *timestamp* do momento da ocorrência da situação de erro;
- ISS ∈ [0, 255] – Identificação de Sistema Sensor;
- ER ∈ [-128, 127] – valor indicativo do tipo de erro.

4.1.6 Mensagem de paragem (*STOP*)

Concentrador → Sistema Sensor

Sempre que o utilizador pretende suspender a recolha e envio das amostras, este deve enviar a partir do concentrador para o sistema sensor simulado uma mensagem do tipo *STOP* com a estrutura da Figura 4.6.



Figura 4.6 – Estrutura da mensagem de paragem.

- TP ∈ [-128, 127] – tipo de mensagem (5 = *STOP*);
- TS ∈ TS ∈ [0, 4 294 967 295] *timestamp* do momento de envio da mensagem;
- SR[32] ∈ ASCII – razão para a paragem (“USERIN (<mensagem personalizada>)”).

4.2 Configuração

Quando o *software* do concentrador é inicializado, é invocado o módulo de arranque que é responsável por ler o ficheiro de configuração (*setup.cfg*) e extrair do mesmo os parâmetros PM (ms), PA (μs) e NS e os parâmetros necessários para estabelecer ligação com os sistemas sensores e com o gestor de serviços, com o seguinte formato:

```
“<PM> <PA> <NS>
<IP do Gestor>:<Porta do Gestor>
<Porta para os Sistemas Sensores>”
```

O ficheiro está exemplificado na Figura 4.7.

```
1000 2000 5
192.168.137.250:7778
12900
```

Figura 4.7 – Ficheiro `setup.cfg` do concentrador.

O concentrador só aceita valores positivos entre 50 e 36 000 000 ms (1 h) para PM, entre 1 e 60 000 000 μ s para PA e entre 1 e 30 amostras para NS. São também aceites valores não positivos que assumem que o concentrador pretende obter os valores por defeito do Arduino.

Foram estabelecidos limites aos parâmetros pois durante a execução dos testes, o Arduino não tinha capacidade para adquirir/enviar dados nos ritmos definidos por certos valores, ou para evitar *overflows* devido ao tamanho do tipo das variáveis usadas.

O ficheiro de configuração é atualizado caso sejam definidos novos parâmetros para PM, PA e NS ao ser enviada uma mensagem do tipo *START*.

Depois de lido o ficheiro de configuração, é aberto um *socket* de servidor para comunicar com os sistemas sensores, é feito o *binding* desse *socket* na porta definida pelo ficheiro de configuração, é aberto um *socket* de cliente para comunicar com o gestor de serviços e o concentrador fica à espera pela conexão ao endereço IP e porta do gestor de serviços também definidos pelo ficheiro de configuração.

Depois de ser estabelecida a conexão com o gestor de serviços, é lido o ficheiro que lista os ISu (Identificações de Sujeitos) e os ISS atribuídos (`users.cfg`), de seguida, as atribuições são guardadas, mas os ISu e o próprio ficheiro são atualizados pelo gestor de serviços depois de ser feito um pedido para tal.

A estrutura deste ficheiro é a seguinte:

```
<ISu[0]> <ISS[0][0]> <...> <ISS[0][M-1]>
<...>
<ISu[N-1]> <ISS[N-1][0]> <...> <ISS[N-1][L-1]>
```

- ISu[N] – conjunto das identificações dos sujeitos disponíveis;
- ISS[n][M] – conjunto das identificações dos sistemas sensores atribuídos ao ISu *n*.

Como se pode verificar, podem ser atribuídos vários ISS ao mesmo ISu.

Um exemplo de conteúdo do ficheiro é o da Figura 4.8.

```
1 1
2 2
3
4
```

Figura 4.8 – Ficheiro `users.cfg` do concentrador.

4.3 Tratamento de dados

Quando é recebida uma mensagem de dados de um sistema sensor, os valores dos eixos dos sensores acelerómetros, giroscópios e de temperatura são convertidos de *raw* para *standard gravity*, graus por segundo e graus *Celsius*, respetivamente. Além disso é verificada a existência de anomalias nas mesmas.

4.4 Controlo de erros

As mensagens de erro, quer sejam provenientes de um sistema sensor ou do próprio concentrador, incluem um identificador numérico do tipo de erro (ER). As descrições para cada tipo de erro são as seguintes:

1. Ficheiro de configuração (`setup.cfg` ou `users.cfg`) inexistente;
2. Tipo de comunicação não encontrado (detetado na leitura do endereço e portas de `setup.cfg`);
3. Tipo de comunicação inválido (não detetado);
4. Dados inválidos (detetado quando os valores dos eixos obtidos nas mensagens de dados não se encontram na escala definida pelo Arduino, portanto é assumido que os valores têm grande probabilidade de estarem errados);
5. Incapacidade de enviar mensagens de dados no ritmo definido (detetado pelo Arduino quando este tem problemas na aquisição de valores dos sensores no ritmo definido pelos parâmetros PM, PA e NS);
6. Falta de resposta do sistema sensor (detetado quando o concentrador não obtém uma mensagem de dados após PM (com uma tolerância do dobro de PM, arredondado em segundos));
7. Falha na abertura do *socket* (cliente ou servidor);
8. Falha no envio da mensagem para o sistema sensor;
9. Falha no *binding*;
10. Parâmetros inválidos (detetado na leitura dos parâmetros PM, PA e NS de `setup.cfg` quando os mesmos ultrapassam os limites definidos ou quando $PM \times 1000 < PA \times NS$);
11. Falha na conexão com o gestor de serviço;
12. Ligação com concentrador terminada abruptamente (exclusivo do gestor de serviços);
13. Falha na escuta do *socket* (exclusivo do gestor de serviços);
14. Falha no envio da mensagem para o gestor de serviços.

4.5 Armazenamento em *log*

São armazenadas em ficheiros de *log* específicos as mensagens circuladas entre o concentrador e os sistemas sensores, assim como mensagens geradas pelo concentrador que permanecem no mesmo, como mensagens de configuração e de erro.

Estão disponíveis quatro tipos de ficheiros de *log*:

- `data.log` – ficheiro de armazenamento de todas as mensagens do tipo *DATA*;
- `error.log` – ficheiro de armazenamento de todas as mensagens do tipo *ERROR*;
- `input.log` – ficheiro de armazenamento de todas as mensagens dos tipos *START* e *STOP*;
- `setup.log` – ficheiro de armazenamento de todas as mensagens dos tipos *SETUP* e *ACCEPT*.

As mensagens são adicionadas a uma nova linha do respetivo ficheiro.

4.5.1.1 *SETUP*

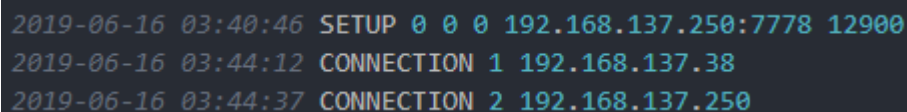
`<TS> SETUP <PM> <PA> <NS> <SERV_AD>:<SERV_PORT> <SS_PORT>`

A *timestamp* tem como formato Y-M-D H:M:S, *SERV_AD* refere-se ao endereço IP do gestor de serviços, *SERV_PORT* refere-se à sua porta e *SS_PORT* refere-se à porta para os sistemas sensores.

4.5.1.2 *ACCEPT*

`<TS> CONNECTION <ISS> <AD>`

Um exemplo de conteúdo de `setup.log` está apresentado na Figura 4.9.



```
2019-06-16 03:40:46 SETUP 0 0 0 192.168.137.250:7778 12900
2019-06-16 03:44:12 CONNECTION 1 192.168.137.38
2019-06-16 03:44:37 CONNECTION 2 192.168.137.250
```

Figura 4.9 – Ficheiro `setup.log` do concentrador.

4.5.1.3 *START*

`<TS> START <ISS> <AD> <PM> <PA> <NS>`

4.5.1.4 *DATA*

`<TS> DATA <ISS> <AD> <PM> <PA> <NS>|<SAX[0]> <SAY[0]> <SAZ[0]>
<SGX[0]> <SGY[0]> <SGZ[0]> <ST[0]>|<...>|<SAX[NS-1]> <SAY[NS-1]>
<SAZ[NS-1]> <SGX[NS-1]> <SGY[NS-1]> <SGZ[NS-1]> <ST[NS-1]>`

Um exemplo de conteúdo de `data.log` está apresentado na Figura 4.10.

```

2019-06-16 03:58:15 DATA 1 192.168.137.38 1000 2000 5|0.57 0.01 0.57 -110.94 -194.61 248.18 21.84|0.06
2019-06-16 03:58:16 DATA 1 192.168.137.38 1000 2000 5|0.90 -0.59 0.07 30.42 89.09 -13.08 21.87|0.06
2019-06-16 03:58:17 DATA 1 192.168.137.38 1000 2000 5|0.69 -0.16 0.89 -7.19 40.69 -33.15 21.87|0.06
2019-06-16 03:59:39 DATA 2 192.168.137.250 1000 3000 10|-0.03 -0.03 0.97 -4.05 -3.98 -3.95 25.02|0.06
2019-06-16 03:59:40 DATA 2 192.168.137.250 1000 3000 10|0.00 -0.03 0.96 0.00 0.09 0.00 24.98|0.06
2019-06-16 03:59:41 DATA 1 192.168.137.38 1000 3000 5|0.06 -0.02 0.98 -1.96 0.00 0.15 22.06|0.08
2019-06-16 03:59:42 DATA 2 192.168.137.250 1000 3000 10|0.14 -0.05 0.98 -3.96 -87.74 -10.47 24.98|0.06
2019-06-16 03:59:42 DATA 1 192.168.137.38 1000 3000 5|0.07 0.00 0.98 0.06 -2.10 -2.03 22.15|0.06
2019-06-16 03:59:43 DATA 2 192.168.137.250 1000 3000 10|1.00 -0.01 0.10 0.16 -30.74 -2.32 25.12|0.06
2019-06-16 03:59:43 DATA 1 192.168.137.38 1000 3000 5|0.06 0.04 1.00 -2.14 0.00 -3.47 22.11|0.07
2019-06-16 03:59:44 DATA 1 192.168.137.38 1000 3000 5|0.36 -0.08 1.21 -44.62 -84.25 -4.62 22.06|0.06
2019-06-16 04:00:31 DATA 2 192.168.137.250 1000 3000 10|0.99 0.17 0.00 -8.18 -0.15 -4.32 25.12|0.06
2019-06-16 04:00:32 DATA 2 192.168.137.250 1000 3000 10|0.33 0.01 0.85 -2.63 63.52 5.27 25.12|0.06

```

Figura 4.10 – Ficheiro data.log do concentrador.

4.5.1.5 ERROR

<TS> ERROR (ER) <ISS> <AD>

Um exemplo de conteúdo de error.log está apresentado na Figura 4.11.

```

2019-06-16 04:14:57 ERROR (5) 1 192.168.137.38

```

Figura 4.11 – Ficheiro error.log do concentrador.

4.5.1.6 STOP

<TS> STOP <ISS> <AD> <SR>

Um exemplo de conteúdo de input.log está apresentado na Figura 4.12.

```

2019-06-16 03:57:05 START 1 192.168.137.38 1000 2000 5
2019-06-16 03:57:11 STOP ALL USERIN
2019-06-16 03:58:07 START 1 192.168.137.38 1000 2000 5
2019-06-16 03:58:16 STOP ALL USERIN
2019-06-16 03:59:39 START ALL 1000 3000 DEFAULT
2019-06-16 03:59:43 STOP ALL USERIN
2019-06-16 04:00:31 START ALL 1000 3000 DEFAULT
2019-06-16 04:00:35 STOP ALL USERIN
2019-06-16 04:03:26 STOP 1 192.168.137.38 USERIN

```

Figura 4.12 – Ficheiro input.log do concentrador.

4.6 Interface de utilização

A interface de utilização é responsável por permitir ao utilizador controlar a comunicação entre o concentrador e cada sistema sensor através da invocação das mensagens do tipo *START* e *STOP*, de visualizar todos os tipos de mensagens geradas em tempo real e em formato interpretável, visualizar mensagens de dados ocorridas num intervalo de tempo definido, ocultar mensagens de dados em tempo real, listar atribuições de ISu a ISS, e fazer essas mesmas atribuições.

Quando o concentrador é inicializado, durante a sua configuração devem ser apresentadas as seguintes mensagens:

```
> A inicializar Concentrador...
> <TS> | Socket de servidor aberto;
> <TS> | Binding feito na porta <SS_PORT>;
> <TS> | Socket de cliente aberto;
> A estabelecer ligação com Gestor de Serviços | TCP/IP:
<SERV_AD>:<PORT_AD>...
> <TS> | Ligação estabelecida com Gestor de Serviços;
> <TS> | Concentrador inicializado | PM = <PM> ms | PA <PA> µs |
NS = <NS> amostra(s);
```

Depois da configuração, é apresentada a lista de sujeitos e o menu da Figura 4.13

```
> Menu de comandos disponíveis:
* l/list - mostrar lista de sujeitos monitorizados e áreas disponíveis,
* a/area <área> <ISu[N]> - atribuir área a N sujeitos (área inexistente: remover N sujeitos das respetivas áreas),
* d/database <ISu/área> <TSI (Y-M-D H:M:S)> <TSF (Y-M-D H:M:S)> - mostrar comportamentos de um sujeito ou numa área n
um intervalo de tempo,
* s/show <modo> - mostrar mensagens recebidas em tempo real (modo = 0: não mostrar valores dos sensores, modo > 0: mo
strar valores),
* h/hide - ocultar mensagens recebidas,
* m/menu - mostrar o menu;
```

Figura 4.13 – Menu do concentrador.

4.6.1 Identificações de Sujeitos e de Sistemas Sensores

Todos os ISu providenciados pelo gestor de serviços e os respetivos ISS atribuídos podem ser visualizados através do comando `list`, apresentando a seguinte lista:

```
> Lista de sujeitos (#ISu, [ISS]):
* #<ISu[0]> [<ISS[0][0]>] <...> [<ISS[0][M-1]>]
* <...>
* #<ISu[N-1]> [<ISS[N-1][0]>] <...> [<ISS[N-1][L-1]>]
```

Um exemplo de uma lista é o da Figura 4.14.

```
list
> Lista de sujeitos (#ISu, [ISS]):
* #1 [1]
* #2 [2]
* #3
* #4
```

Figura 4.14 – Lista de sujeitos do concentrador.

ISu podem ser atribuídos a ISS através do seguinte comando:

```
user <ISu> <ISS[0]> <ISS[1]> <...> <ISS[M-1]>
```

Se a um dos ISS's escolhidos já estiver atribuído um ISu, o mesmo será substituído pelo escolhido; se este não estiver disponível, esse ISS ficará sem ISu. A invocação deste comando irá alterar o respetivo ficheiro de configuração.

4.6.2 Controlo dos sistemas sensores

4.6.2.1 *START*

Sempre que o utilizador pretende iniciar a recolha de dados, este invocar o seguinte comando:

```
start (<ISS>) (<PM> <PA> <NS>)
```

Se ISS estiver absente ou tiver como valor 0, a mensagem será enviada para todos os sistemas sensores cujos endereços estão guardados na lista de endereços. Caso contrário, a mensagem será enviada para o endereço indexado pelo ISS definido.

A ausência dos parâmetros PM, PA e NS implicará que os mesmos são os definidos pelo ficheiro de configuração. Se esses parâmetros forem inseridos, os mesmos substituirão os que estão presentes no ficheiro de configuração. Se os parâmetros tiverem valores não positivos, o sistema sensor irá usar os seus parâmetros por defeito.

Este comando fará com que seja apresentado no terminal a seguinte mensagem:

```
> <TS> | START [<ISS>] | IP: <AD> - PM = <PM> ms | PA <PA>
μs | NS = <NS> amostra(s);
```

Um exemplo de mensagem está apresentado na Figura 4.15.

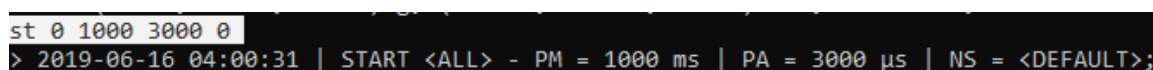
A terminal window with a black background and white text. The first line shows a prompt 'st 0 1000 3000 0'. The second line shows the output of the START command: '> 2019-06-16 04:00:31 | START <ALL> - PM = 1000 ms | PA = 3000 μs | NS = <DEFAULT>;'.

Figura 4.15 – Mensagem *START* enviada pelo concentrador.

4.6.2.2 *STOP*

Caso o utilizador pretenda suspender a receção de mensagens de dados, este deve invocar o seguinte comando:

```
stop (<ISS>) (<SR>)
```

Se não for indicada nenhuma razão de paragem (SR), será assumido que a mesma é simplesmente "USERIN".

Este comando fará com que seja apresentado no terminal a seguinte mensagem:

```
> <TS> | STOP [<ISS>] | IP: <AD> - <SR>;
```

Um exemplo de mensagem está apresentado na Figura 4.16.

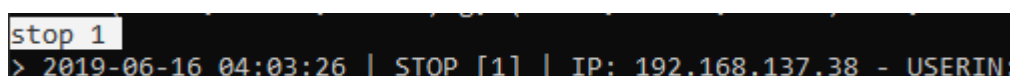
A terminal window with a black background and white text. The first line shows a prompt 'stop 1'. The second line shows the output of the STOP command: '> 2019-06-16 04:03:26 | STOP [1] | IP: 192.168.137.38 - USERIN;'.

Figura 4.16 – Mensagem *STOP* enviada pelo concentrador.

4.6.3 Mensagens em tempo real

Quando o concentrador recebe uma mensagem de um dos sistemas sensores, esta é apresentada em tempo real sendo a sua estrutura dependente do tipo (o formato dos campos é idêntico ao formato dos campos armazenados em *log*).

4.6.3.1 ACCEPT

```
> <TS> | Conexão estabelecida com [<ISS>] | UDP/IP: <AD>;
```

Dois exemplos de mensagens estão apresentados na Figura 4.17.

```
> 2019-06-16 03:44:12 | Conexão estabelecida com [1] | UDP/IP: 192.168.137.38;
> 2019-06-16 03:44:37 | Conexão estabelecida com [2] | UDP/IP: 192.168.137.250;
```

Figura 4.17 – Mensagem *ACCEPT* para dois sistemas sensores.

4.6.3.2 DATA

```
> <TS> | DATA [<ISS>] | IP: <AD> | PM = <PM> ns | PA = <PA> μs |
NS = <NS> amostra(s):
* S1: (<SAX[0]>, <SAY[0]>, <SAZ[0]>) g, (<SGX[0]>, <SGY[0]>,
<SGZ[0]>) °/s, <ST[0]> °C,
* <...>,
S<N>: (<SAX[N-1]>, <SAY[N-1]>, <SAZ[N-1]>) g, (<SGX[N-1]>, <SGY[N-1]>,
<SGZ[N-1]>) °/s, <ST[N-1]> °C;
```

Três exemplos de mensagens estão apresentados na Figura 4.18.

```
> 2019-06-16 03:59:41 | DATA [1] | IP: 192.168.137.38 - PM = 1000 ms | PA = 3000 μs | NS = 5 amostra(s):
* S1: ( 0.06, -0.02, 0.98) g, (-1.96, 0.00, 0.15) °/s, 22.06 °C,
* S2: ( 0.08, -0.02, 1.00) g, (-2.02, 0.04, -1.97) °/s, 22.11 °C,
* S3: ( 0.08, -0.02, 0.99) g, (-2.13, 0.01, 0.09) °/s, 22.11 °C,
* S4: ( 0.06, -0.02, 1.00) g, (-2.04, -2.08, 0.14) °/s, 22.06 °C,
* S5: ( 0.06, -0.01, 1.02) g, ( 0.04, 0.08, -1.96) °/s, 22.11 °C;
> 2019-06-16 03:59:42 | DATA [2] | IP: 192.168.137.250 - PM = 1000 ms | PA = 3000 μs | NS = 10 amostra(s):
* S1: ( 0.14, -0.05, 0.98) g, (-3.96, -87.74, -10.47) °/s, 24.98 °C,
* S2: ( 0.16, -0.01, 0.99) g, (-4.22, -92.03, -7.01) °/s, 25.07 °C,
* S3: ( 0.75, -0.04, 0.73) g, (-2.60, -26.24, -0.04) °/s, 25.02 °C,
* S4: ( 0.73, -0.03, 0.73) g, (-2.50, -24.58, -0.50) °/s, 25.16 °C,
* S5: ( 0.75, 0.02, 0.74) g, (-2.52, -32.22, -0.42) °/s, 25.07 °C,
* S6: ( 0.73, 0.06, 0.69) g, (-2.05, -35.63, 4.02) °/s, 24.98 °C,
* S7: ( 0.79, -0.01, 0.69) g, (-2.02, -32.63, 4.13) °/s, 25.12 °C,
* S8: ( 0.99, -0.01, 0.10) g, ( 0.68, -31.11, -2.05) °/s, 25.07 °C,
* S9: ( 0.99, -0.04, 0.10) g, ( 0.52, -31.09, -2.22) °/s, 25.07 °C,
* S10: ( 1.00, -0.04, 0.13) g, ( 0.23, -30.82, -2.03) °/s, 24.98 °C;
> 2019-06-16 03:59:42 | DATA [1] | IP: 192.168.137.38 - PM = 1000 ms | PA = 3000 μs | NS = 5 amostra(s):
* S1: ( 0.07, 0.00, 0.98) g, ( 0.06, -2.10, -2.03) °/s, 22.15 °C,
* S2: ( 0.06, -0.02, 1.01) g, (-2.09, -1.97, 0.13) °/s, 22.06 °C,
* S3: ( 0.06, -0.02, 1.00) g, (-2.02, -2.04, 0.12) °/s, 22.15 °C,
* S4: ( 0.06, -0.02, 1.01) g, (-2.05, 0.11, 0.08) °/s, 22.11 °C,
* S5: ( 0.06, -0.02, 0.98) g, (-2.04, -1.98, 0.07) °/s, 22.06 °C;
```

Figura 4.18 – Mensagens *DATA* enviadas por dois sistemas sensores.

4.6.3.3 ERROR

```
> <TS> | ERROR (<ER>) [<ISS>] | IP: <AD> - <ER_DESC>;
```

ER_DESC refere-se à descrição do tipo de erro (ER).

Um exemplo de mensagem está apresentado na Figura 4.19.


```
> 2019-06-16 04:14:57 | ERROR (5) [1] | IP: 192.168.137.38 - incapacidade de enviar mensagens de dados no ritmo definido;
```

Figura 4.19 – Mensagem *ERROR* enviada por um sistema sensor.

4.6.4 Leitura de dados armazenados

Para a visualização de mensagens de dados armazenadas dentro de um intervalo de tempo definido, o utilizador deve invocar o seguinte comando:

```
show (<ISS>) (<TSI> <TSF>)
```

Se ISS for inserido, serão apresentadas as mensagens armazenadas em `data.log` no intervalo entre TSI e TSF desse mesmo ISS. Caso contrário, serão apresentadas as mensagens de todos os ISS.

A apresentação destas mensagens tem a mesma estrutura das mensagens de dados em tempo real e está exemplificada na Figura 4.20.

```
show 1 2019-06-16 03:59:40 2019-06-16 03:59:44
> Comportamentos de [1] de 2019-06-16 03:59:40 até 2019-06-16 03:59:44:
> 2019-06-16 03:59:41 | DATA [1] | IP: 192.168.137.38 - PM = 1000 ms | PA = 3000 µs | NS = 5 amostra(s):
* S1: ( 0.06, -0.02, 0.98) g, (-1.96, 0.00, 0.15) g/s, 22.06 °C,
* S2: ( 0.08, -0.02, 1.00) g, (-2.02, 0.04, -1.97) g/s, 22.11 °C,
* S3: ( 0.08, -0.02, 0.99) g, (-2.13, 0.01, 0.09) g/s, 22.11 °C,
* S4: ( 0.06, -0.02, 1.00) g, (-2.04, -2.08, 0.14) g/s, 22.06 °C,
* S5: ( 0.06, -0.01, 1.02) g, ( 0.04, 0.08, -1.96) g/s, 22.11 °C;
> 2019-06-16 03:59:42 | DATA [1] | IP: 192.168.137.38 - PM = 1000 ms | PA = 3000 µs | NS = 5 amostra(s):
* S1: ( 0.07, 0.00, 0.98) g, ( 0.06, -2.10, -2.03) g/s, 22.15 °C,
* S2: ( 0.06, -0.02, 1.01) g, (-2.09, -1.97, 0.13) g/s, 22.06 °C,
* S3: ( 0.06, -0.02, 1.00) g, (-2.02, -2.04, 0.12) g/s, 22.15 °C,
* S4: ( 0.06, -0.02, 1.01) g, (-2.05, 0.11, 0.08) g/s, 22.11 °C,
* S5: ( 0.06, -0.02, 0.98) g, (-2.04, -1.98, 0.07) g/s, 22.06 °C;
> 2019-06-16 03:59:43 | DATA [1] | IP: 192.168.137.38 - PM = 1000 ms | PA = 3000 µs | NS = 5 amostra(s):
* S1: ( 0.06, 0.04, 1.00) g, (-2.14, 0.00, -3.47) g/s, 22.11 °C,
* S2: ( 0.07, -0.02, 1.01) g, (-2.12, -2.16, -2.00) g/s, 22.06 °C,
* S3: ( 0.06, -0.02, 1.00) g, ( 0.06, -2.19, -1.97) g/s, 22.11 °C,
* S4: ( 0.06, -0.02, 0.99) g, ( 0.12, -2.05, -2.00) g/s, 22.95 °C,
* S5: ( 0.06, -0.02, 1.00) g, (-2.00, 0.19, -2.05) g/s, 22.06 °C;
> 2019-06-16 03:59:44 | DATA [1] | IP: 192.168.137.38 - PM = 1000 ms | PA = 3000 µs | NS = 5 amostra(s):
* S1: ( 0.36, -0.08, 1.21) g, (-44.62, -84.25, -4.62) g/s, 22.06 °C,
* S2: ( 0.40, -0.15, 1.17) g, (-42.13, -90.20, -2.22) g/s, 22.11 °C,
* S3: ( 0.35, -0.12, 1.09) g, (-39.43, -85.86, -7.34) g/s, 22.15 °C,
* S4: ( 0.34, -0.11, 1.07) g, (-41.77, -86.92, -9.22) g/s, 22.15 °C,
* S5: ( 0.34, -0.11, 1.02) g, (-45.29, -82.05, -9.69) g/s, 22.15 °C;
* 4 resultados;
```

Figura 4.20 – Leitura das mensagens de dados armazenadas em *log*.

4.6.5 Apresentação e ocultação das mensagens em tempo real

Para que o utilizador possa interagir com o concentrador, principalmente para poder atribuir ISu a ISS, visualizar a lista de ISu disponíveis, mensagens armazenadas ou o menu, é-lhe dado a opção de ocultar as mensagens de dados em tempo real para que estas não sobreponham o ecrã (o concentrador continua a recebê-las e a armazená-las em *log*). Para tal, deve ser usado o comando `hide`.

Este comando é automaticamente invocado ao serem usados os comandos `list`, `user`, `show <ISS>` e `menu`.

Para que as mensagens voltem a ser apresentadas, deve ser usado o comando `show` sem argumentos.

4.7 Changelog

4.7.1 Fase A

- Implementação do concentrador.

4.7.2 Fase B

- Alteração no protocolo de comunicação de orientação ao caracter para orientação ao bit;
- Otimização do protocolo de comunicações:
 - Redução do tamanho do campo ISS das mensagens que o incluem de 2 bytes para apenas 1 pois achamos que 256 sistemas sensores era superior ao necessário;
- Comunicação com múltiplos sistemas sensores:
 - Foram armazenados os endereços de cada sistema sensor que envia uma mensagem;
 - Adição do parâmetro ISS no comando *START*;

4.7.3 Final

- Atribuição de ISu a ISS;
- Alteração do tipo de mensagens de erro de configuração de *SETERR* para *ERROR*;
- Alteração do formato dos campos guardados em *log*:
 - *Timestamps* são guardadas em formato Y-M-D H:M:S em vez de Unix e o tipo de mensagem substituiu o identificador numérico do mesmo, para que as mensagens sejam facilmente interpretáveis;
- Exclusão do armazenamento em *input.log* das mensagens que não são *START* nem *STOP*:
 - Invocações dos comandos de menu, leitura dos ficheiros *log* e apresentação/ocultação de mensagens em tempo real deixaram de ser registadas em *log* pois achamos isso desnecessário;
- Filtração da leitura das mensagens armazenadas por ISS;
- Otimização do protocolo de comunicação:
 - Remoção do campo do endereço nas mensagens que incluem o mesmo (*CONN* e *ERROR*) e adição do campo ISS às mensagens de erro. Achamos o campo de endereço desnecessário visto que pode ser obtido pelo concentrador através do *socket* que recebe a mensagem;
- Alterações mínimas na interface de utilização e do armazenamento e leitura de *log* e otimização geral do código do programa.

5 Gestor de Serviço

Assim como o concentrador, o gestor de serviço consiste numa aplicação *software* desenvolvida em linguagem C.

O papel do gestor de serviço é de receber mensagens de dados de concentradores que contêm os valores processados e recolhidos dos sistemas sensores.

Estes valores são interpretados pelo gestor de modo a obter o comportamento em tempo real de cada sujeito para ser armazenado numa base de dados e apresentado na UI.

5.1 Protocolo de comunicação com Concentradores[8]

O protocolo de comunicação entre um concentrador e o gestor de serviço foi construído à base dos mesmos tipos de mensagens usadas entre um concentrador e cada sistema sensor, enviadas ao bit através do protocolo de transporte TCP/IP (*Transmission Control Protocol*) [8], tendo o gestor o papel de aplicação servidor e os concentradores os clientes.

Para a comunicação com vários concentradores, é criado um processo e um *socket* por cada conexão aceite de um concentrador, que é fechado quando o mesmo se desconecta.

5.1.1 Mensagem de conexão (*CONN*)

Concentrador → Gestor de Serviço

Quando é estabelecida a conexão entre um concentrador e um sistema sensor, é enviada uma mensagem do tipo *CONN* do concentrador para o gestor de serviço, com a estrutura da Figura 5.1.



Figura 5.1 – Estrutura da mensagem de conexão.

- TP ∈ [-128, 127] – tipo de mensagem (1 = *CONN*);
- TS ∈ [0, 4 294 967 295] – *timestamp* do momento da conexão;
- AD ∈ [0.0.0.0, 255.255.255.255] – endereço IP do sistema sensor;
- ISU ∈ [0, 255] – identificação do sujeito.

5.1.2 Mensagem de início (*START*)

Concentrador → Gestor de Serviço

Sempre que a recolha de amostras dum sistema sensor é iniciada ou reiniciada após uma paragem controlada, é enviada uma mensagem do tipo *START* do concentrador para o gestor de serviço, com a estrutura da Figura 5.2.

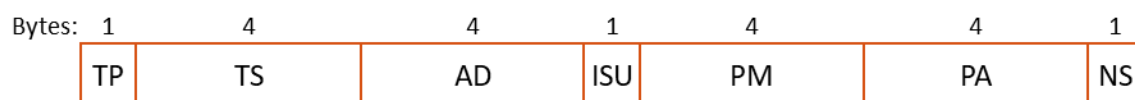


Figura 5.2 – Estrutura da mensagem de início.

- TP $\in [-128, 127]$ – tipo de mensagem (3 = *START*);
- TS $\in [0, 4\ 294\ 967\ 295]$ – *timestamp* do momento do início da recolha;
- AD $\in [0.0.0.0, 255.255.255.255]$ – endereço IP do sistema sensor;
- ISU $\in [0, 255]$ – identificação do sujeito (0 = todos os sujeitos);
- PM $\in [0, 4\ 294\ 967\ 295]$ – período (ms) definido entre cada mensagem de dados a ser recebida;
- PA $\in [0, 4\ 294\ 967\ 295]$ – período (μ s) definido entre cada amostra a ser recolhida por mensagem;
- NS $\in [0, 255]$ – número de amostras a serem recolhidas por mensagem.

5.1.3 Mensagens de dados (*DATA*)

Concentrador → Gestor de Serviço

Sempre que o concentrador recebe uma mensagem de dados, esta é reencaminhada para o gestor de serviço com o tipo *DATA* e a estrutura da Figura 5.3.

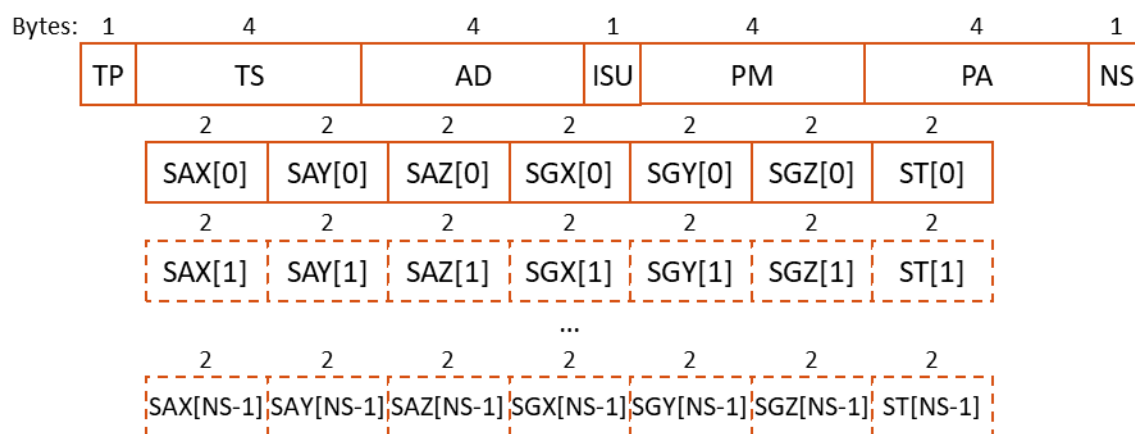


Figura 5.3 – Estrutura da mensagem de dados.

- TP $\in [-128, 127]$ – tipo de mensagem (4 = *DATA*);
- TS $\in [0, 4\ 294\ 967\ 295]$ – *timestamp* do momento da recolha da primeira amostra presente na mensagem;
- AD $\in [0.0.0.0, 255.255.255.255]$ – endereço IP do sistema sensor;
- ISU $\in [0, 255]$ – identificação do sujeito;
- PM $\in [0, 4\ 294\ 967\ 295]$ – período entre cada mensagem enviada;
- PA $\in [0, 4\ 294\ 967\ 295]$ – período entre cada amostra recolhida;
- NS $\in [0, 255]$ – número de amostras recolhidas por mensagem;
- SAX[NS], SAY[NS], SAZ[NS] $\in [-32\ 768, 32\ 767]$ – *arrays* de valores das acelerações ($100 \times g$) nos eixos X, Y e Z obtidos pelo acelerómetro das NS amostras recolhidas;

- $SGX[NS], SGY[NS], SGZ[NS] \in [-32\,768, 32\,767]$ – *arrays* de valores das velocidades angulares ($100 \times (^{\circ}/s)$) nos eixos X, Y e Z obtidos pelo giroscópio das NS amostras recolhidas.
- $ST[NS] \in [-32\,768, 32\,767]$ – *array* de valores das temperaturas ($100 \times ^{\circ}C$) recolhidas nas NS amostras.

5.1.4 Mensagens de erro (*ERROR*)

(Concentrador) → Gestor de Serviço

Sempre que o concentrador deteta um erro, este envia para o gestor de serviço uma mensagem do tipo *ERROR* com a estrutura da Figura 5.4.



Figura 5.4 – Estrutura da mensagem de erro.

- $TP \in [-128, 127]$ – tipo de mensagem (-2 = *ERROR*);
- $TS \in [0, 4\,294\,967\,295]$ – *timestamp* do momento da ocorrência da situação de erro;
- $AD \in [0.0.0.0, 255.255.255.255]$ – endereço IP do sistema sensor;
- $ISU \in [0, 255]$ – identificação do sujeito;
- $ER \in [-128, 127]$ – valor indicativo do tipo de erro.

5.1.5 Mensagem de paragem (*STOP*)

Concentrador → Gestor de Serviço

Sempre que a recolha de amostras é interrompida controladamente (sem qualquer condição de erro), o concentrador envia para o gestor de serviço uma mensagem do tipo *STOP* com a estrutura da Figura 5.5.

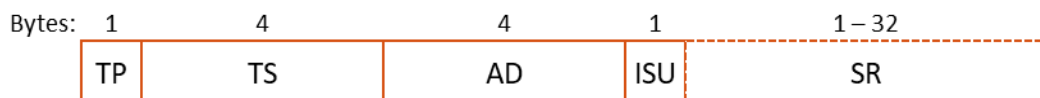


Figura 5.5 – Estrutura da mensagem de paragem.

- $TP \in [-128, 127]$ – tipo de mensagem (5 = *STOP*);
- $TS \in [0, 4\,294\,967\,295]$ – *timestamp* do momento da paragem da recolha;
- $AD \in [0.0.0.0, 255.255.255.255]$ – endereço IP do sistema sensor;
- $ISU \in [0, 255]$ – identificação do sujeito;

$SR[32] \in \text{ASCII}$ – razão para a paragem (“USERIN (<mensagem personalizada>)”).

5.1.6 Mensagens de pedido de lista de ISu (*USER*) e de envio da mesma

Concentrador → Gestor de Serviço

Sempre que o utilizador pretende listar ou atribuir ISS a ISu, é enviada em antecedência uma mensagem do tipo *USER* com apenas um byte que identifica o tipo (14).

Gestor de Serviço → Concentrador

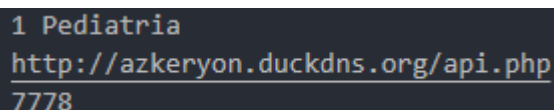
Quando o gestor de serviço recebe uma mensagem do tipo *USER*, responde ao concentrador com a lista de todos os ISu contidos no seu ficheiro de configuração `areas.cfg`.

5.2 Configuração

Quando o *software* do gestor de serviço é inicializado, é lido o ficheiro de configuração (`setup.cfg`) para serem extraídos do mesmo o seu identificador de serviço (ISe), a designação do serviço, o URL para o qual devem ser feitos os HTTP (*Hypertext Transfer Protocol*) *Requests* para o Sistema Central e a porta de conexão com os concentradores. Este ficheiro tem o seguinte formato:

```
"<ISe> <designação do serviço>
<URL da API do Sistema Central>
<Porta para os Concentradores>"
```

O conteúdo deste ficheiro está exemplificado na Figura 5.6.



```
1 Pediatria
http://azkeryon.duckdns.org/api.php
7778
```

Figura 5.6 – Ficheiro `setup.cfg` do gestor de serviço.

Depois de lido o ficheiro de configuração, é aberto um *socket* de servidor para comunicar com os concentradores, é feito o *binding* desse *socket* na porta definida pelo ficheiro de configuração, o ISe e a designação do serviço são enviados para o Sistema Central num HTTP *Request*.

Depois, é lido o ficheiro que lista as áreas disponíveis para o serviço com os ISu a que estão atribuídas, com a seguinte estrutura:

```
<AREA[0]> <ISu[0][0]> <...> <ISu[0][M-1]>
<...>
<AREA[N-1]> <ISu[N-1][0]> <...> <ISu[N-1][L-1]>
```

- AREA[N] – conjunto de áreas disponíveis;
- ISu[n][M] – conjunto das identificações dos sujeitos da área *n*.

O conteúdo deste ficheiro está exemplificado na Figura 5.7.

P.1	1	2
P.2	3	4

Figura 5.7 – Ficheiro areas . cfg do gestor de serviço.

5.3 Condições/comportamentos físicos

Para ser determinado o comportamento de um sujeito, devem ser usados seguidos algoritmos e fórmulas com os valores dos diversos sensores.

5.3.1 Normal – Parado

Um sujeito encontra-se parado se todas as amostras recolhidas tiverem a seguinte condição nos valores provenientes do acelerómetro:

$$|v_a| = \sqrt{a_x^2 + a_y^2 + a_z^2} \in [0,9; 1,1] \text{ g.}$$

5.3.2 Normal – Deitado

Um sujeito encontra-se deitado se todas as amostras recolhidas tiverem a condição de parado juntamente com a seguinte em pelo menos uma das amostras:

$$a_x < 0,9 \text{ g.}$$

5.3.3 Normal – Andamento

Um sujeito encontra-se em andamento se todas as amostras recolhidas tiverem a seguinte condição:

$$|v_a| \in [0,5; 1,4] \text{ g.}$$

Como esta condição está incluída na condição dum sujeito parado, basta uma amostra ultrapassar um dos limites dessa condição para que seja uma condição de andamento.

5.3.4 Normal – Correr

Considera-se que um sujeito está a correr se pelo menos uma das amostras recolhidas tiver a seguinte condição:

$$|v_a| > 1,4 \text{ g.}$$

5.3.5 Alarme – Queda

Um sujeito está a cair se pelo menos uma das amostras recolhidas tiver a seguinte condição:

$$|v_a| < 0,5 \text{ g.}$$

Esta condição é substituída pela de correr se a mesma se verificar (por exemplo se o módulo de uma das amostras for 0,4 g e o de outra for 1,8 g, a condição é de correr).

5.3.6 Alarme – Agitado

Um sujeito é considerado agitado se pelo menos uma das amostras recolhidas tiver a seguinte condição nos valores provenientes do giroscópio:

$$|v_{\omega}| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} > 300 \text{ }^{\circ}/\text{s}.$$

Esta condição sobrepõe-se às anteriores, exceto à condição de queda.

5.3.7 Alarme – Inativo

Um sistema sensor encontra-se inativo se os seus dados estiverem defeituosos, esta condição é detetada se todas as amostras tiverem como valores:

$$|v_a| = 0.$$

Esta condição sobrepõe-se ao comportamento agitado.

5.4 Base de dados

Estão disponíveis dois tipos de ficheiros para a base de dados:

- `messagedb.txt` – ficheiro de armazenamento de todas as mensagens provenientes do concentrador ou do sistema do gestor de serviço;
- `conditiondb.txt` – ficheiro de armazenamento e acesso a todos os comportamentos dos sujeitos registados com as respetivas áreas.

5.4.1 Armazenamento das mensagens

Cada linha do conteúdo do ficheiro de armazenamento das mensagens contem o conjunto de parâmetros relativos ao tipo de mensagem (idênticos aos recebidos no protocolo de comunicação).

5.4.1.1 SETUP

```
<TS> SETUP <ISe> <NAME> <URL> <PORT_HUB>
```

A *timestamp* tem como formato Y-M-D H:M:S, NAME refere-se à designação do serviço, URL (*Uniform Resource Locator*) refere-se ao URL do ao endereço IP do sistema central e PORT_HUB refere-se à porta para o concentrador.

5.4.1.2 CONN

Concentradores:

```
<TS> CONNECTION HUB <HUB_AD>:<PORT>
```

HUB_AD refere-se ao endereço IP do concentrador

Sistemas sensors:

```
<TS> CONNECTION <ISu> <SS_AD>
```

SS_AD refere-se ao endereço IP do sistema sensor.

5.4.1.3 START

<TS> START <ISu> <SS_AD> <PM> <PA> <NS>

5.4.1.4 DATA

<TS> DATA <ISu> <SS AD> <PM> <PA> <NS> | <SAX[0]> <SAY[0]> <SAZ[0]>
<SGX[0]> <SGY[0]> <SGZ[0]> <ST[0]> | <...> | <SAX[NS-1]> <SAY[NS-1]>
<SAZ[NS-1]> <SGX[NS-1]> <SGY[NS-1]> <SGZ[NS-1]> <ST[NS-1]>

5.4.1.5 ERROR

<TS> ERROR <ISu> <AD> <ER>

AD pode se referir tanto ao endereço IP do concentrador como ao do sistema sensor, dependendo do tipo de erro; no primeiro caso, "HUB" é escrito no parâmetro ISu.

5.4.1.6 STOP

<TS> STOP <ISu> <SS_AD> <SR>

Um exemplo de conteúdo de `messagedb.txt` está apresentado na Figura 5.8.

```
2019-06-16 03:40:19 SETUP 1 Pediatria http://azkeryon.duckdns.org/api.php 7778
2019-06-16 03:40:46 CONNECTION HUB 192.168.137.250:54359
2019-06-16 03:44:12 CONNECTION 1 192.168.137.38
2019-06-16 03:44:37 CONNECTION 2 192.168.137.250
2019-06-16 03:57:05 START 1 192.168.137.38 1000 2000 5
2019-06-16 03:57:06 DATA 1 192.168.137.38 1000 2000 5 | 0.00 -0.01 0.98 0.04 0.
2019-06-16 03:57:07 DATA 1 192.168.137.38 1000 2000 5 | 0.00 -0.01 0.99 -2.00 0
2019-06-16 03:57:08 DATA 1 192.168.137.38 1000 2000 5 | -0.01 -0.02 1.00 0.19 0
2019-06-16 03:57:09 DATA 1 192.168.137.38 1000 2000 5 | 0.00 -0.02 0.98 0.03 -1
2019-06-16 03:57:10 DATA 1 192.168.137.38 1000 2000 5 | 0.00 -0.01 1.00 0.02 -1
2019-06-16 03:57:11 STOP ALL USERIN
2019-06-16 04:14:57 ERROR (5) 1 192.168.137.38
2019-06-16 04:15:25 START 1 192.168.137.38 DEFAULT DEFAULT DEFAULT
2019-06-16 04:15:26 DATA 1 192.168.137.38 1000 1000 5 | -0.01 -0.01 1.00 -2.05
2019-06-16 04:15:27 DATA 1 192.168.137.38 1000 1000 5 | -0.02 -0.01 0.98 -2.05
2019-06-16 04:15:28 STOP ALL USERIN
2019-06-16 04:15:28 DATA 1 192.168.137.38 1000 1000 5 | 0.00 0.00 0.98 0.11 0.0
2019-06-16 04:50:40 DISCONNECTION HUB 192.168.137.250:54359
```

Figura 5.8 – Ficheiro `messagedb.txt` do gestor de serviço.

5.4.2 Armazenamento dos comportamentos

O formato de cada linha do conteúdo do ficheiro de armazenamento das condições dos sujeitos é a seguinte:

<TS> <STATUS> <ISu> <AREA>

STATUS refere-se ao comportamento do sujeito.

O seu armazenamento está demonstrado na Figura 5.9.

```

2019-06-16 03:59:43 DEITADO 2 P.1
2019-06-16 03:59:43 DEITADO 1 P.1
2019-06-16 03:59:44 ANDAR 1 P.1
2019-06-16 04:00:31 CORRER 2 P.1
2019-06-16 04:00:32 DEITADO 2 P.1
2019-06-16 04:00:32 ANDAR 1 P.1
2019-06-16 04:00:33 DEITADO 2 P.1
2019-06-16 04:00:33 CORRER 1 P.1
2019-06-16 04:00:34 DEITADO 2 P.1

```

Figura 5.9 – Ficheiro conditiondb.txt do gestor de serviço.

5.5 Interface de utilizador

A UI do gestor de serviços disponibiliza principalmente a visualização de todas as mensagens enviadas pelo utilizador em tempo real e num intervalo de tempo definido e a atribuição de áreas a sujeitos.

Durante a configuração do gestor, devem ser apresentadas as seguintes mensagens:

```

> A inicializar Gestor de Serviços...
> <TS> | Socket de servidor aberto;
> <TS> | Binding feito na porta <HUB_PORT>;
> A conectar com o Sistema Central | URL: <URL>...
> <TS> | Socket de cliente aberto;
> <TS> | Gestor de Serviços de <NAME> #<ISe> inicializado;

```

Depois de configurado, é apresentada a lista de áreas e o menu da Figura 5.10.

```

> Menu de comandos disponíveis:
* l/list - mostrar lista de sujeitos monitorizados e áreas disponíveis,
* a/area <área> <ISu[N]> - atribuir área a N sujeitos (área inexistente: remover N sujeitos das respetivas áreas),
* d/database <ISu/área> <TSI (Y-M-D H:M:S)> <TSF (Y-M-D H:M:S)> - mostrar comportamentos de um sujeito ou numa área n
um intervalo de tempo,
* s/show <modo> - mostrar mensagens recebidas em tempo real (modo = 0: não mostrar valores dos sensores, modo > 0: mo
strar valores),
* h/hide - ocultar mensagens recebidas,
* m/menu - mostrar o menu;
> À procura de conexões de concentradores...

```

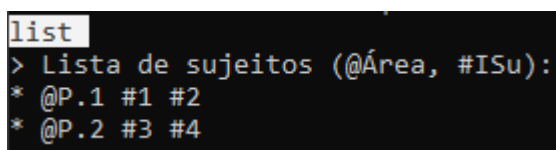
Figura 5.10 – Menu do gestor de serviço.

5.5.1 Áreas

Todas as áreas disponíveis e as respetivas identificações dos sujeitos a que estão atribuídas podem ser visualizadas através do comando `list`, apresentando a seguinte lista:

```
> Lista de sujeitos (@Área, #ISu):  
* @<AREA[0]> #<ISu[0][0]> <...> #<ISu[0][M-1]>  
* <...>  
* @<AREA[N-1]> #<ISu[N-1][0]> <...> #<ISu[N-1][L-1]>
```

Um exemplo de uma lista é o da Figura 5.11.



```
list  
> Lista de sujeitos (@Área, #ISu):  
* @P.1 #1 #2  
* @P.2 #3 #4
```

Figura 5.11 – Lista de sujeitos do gestor de serviço.

Estas áreas podem ser atribuídas a sujeitos através do seguinte comando:

```
area <AREA> <ISu[0]> <ISu[1]> <...> <ISu[M-1]>
```

Se um dos ISu escolhidos já estiver numa área, a mesma será substituída pela escolhida; se esta não estiver disponível, esse ISu ficará sem área. A invocação deste comando irá alterar o ficheiro de configuração.

5.5.2 Mensagens em tempo real

Quando o concentrador envia uma mensagem para o gestor de serviços, esta é apresentada em tempo real sendo a sua estrutura dependente do tipo (o formato dos campos é idêntico ao formato dos campos armazenados na base de dados).

5.5.2.1 CONN

```
> <TS> | Conexão aceite com concentrador | TCP/IP:  
      <HUB_AD>:<PORT>;
```

```
> <TS> | Conexão estabelecida com #<ISu> @<AREA> | IP: <SS_AD>;
```

5.5.2.2 Desconexão

```
> <TS> | Desconexão de concentrador | TCP/IP: <HUB_AD>:<PORT>;
```

5.5.2.3 START

```
> <TS> | START #<ISu> @<Area> | IP: <SS_AD> | PM = <PM> ns | PA =  
<PA> μs | NS = <NS> amostra(s);
```

5.5.2.4 DATA

```
> <TS> | <STATUS> #<ISu> @<AREA> | IP: <SS_AD> | PM = <PM> ns | PA  
= <PA> μs | NS = <NS> amostra(s);
```

5.5.2.5 ERROR

> <TS> | ERROR (<ER>) #<ISu> @<AREA> | IP: <AD> - <ER_DESC>;

ER_DESC refere-se à descrição do tipo de erro (ER).

5.5.2.6 STOP

> <TS> | STOP #<ISu> @<AREA> | IP: <SS_AD> | SR = <SR>;

Na Figura 5.12 é demonstrada a visualização das mensagens em tempo real.

```
> 2019-06-16 03:40:46 | Conexão aceite com concentrador | TCP/IP: 192.168.137.250:54359;
> 2019-06-16 03:44:12 | Conexão estabelecida com #1 @P.1 | IP: 192.168.137.38;
> 2019-06-16 03:44:37 | Conexão estabelecida com #2 @P.1 | IP: 192.168.137.250;
> 2019-06-16 03:57:05 | START #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 µs | NS = 5 amostras;
> 2019-06-16 03:57:06 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 µs | NS = 5 amostras;
> 2019-06-16 03:57:07 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 µs | NS = 5 amostras;
> 2019-06-16 03:57:08 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 µs | NS = 5 amostras;
> 2019-06-16 03:57:09 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 µs | NS = 5 amostras;
> 2019-06-16 03:57:10 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 µs | NS = 5 amostras;
> 2019-06-16 03:57:11 | STOP <ALL> | Razão de paragem: USERIN;
> 2019-06-16 03:59:39 | START <ALL> | PM = 1000 ms | PA = 3000 µs | NS = <DEFAULT>;
> 2019-06-16 03:59:39 | DEITADO #2 @P.1 | IP: 192.168.137.250 | PM = 1000 ms | PA = 3000 µs | NS = 10 amostras;
> 2019-06-16 03:59:40 | DEITADO #2 @P.1 | IP: 192.168.137.250 | PM = 1000 ms | PA = 3000 µs | NS = 10 amostras;
> 2019-06-16 03:59:41 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 3000 µs | NS = 5 amostras;
> 2019-06-16 03:59:42 | DEITADO #2 @P.1 | IP: 192.168.137.250 | PM = 1000 ms | PA = 3000 µs | NS = 10 amostras;
> 2019-06-16 03:59:42 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 3000 µs | NS = 5 amostras;
> 2019-06-16 03:59:43 | DEITADO #2 @P.1 | IP: 192.168.137.250 | PM = 1000 ms | PA = 3000 µs | NS = 10 amostras;
> 2019-06-16 03:59:43 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 3000 µs | NS = 5 amostras;
> 2019-06-16 03:59:43 | STOP <ALL> | Razão de paragem: USERIN;
> 2019-06-16 03:59:44 | ANDAR #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 3000 µs | NS = 5 amostras;
> 2019-06-16 04:00:31 | START <ALL> | PM = 1000 ms | PA = 3000 µs | NS = <DEFAULT>;
> 2019-06-16 04:00:31 | CORRER #2 @P.1 | IP: 192.168.137.250 | PM = 1000 ms | PA = 3000 µs | NS = 10 amostras;
> 2019-06-16 04:00:32 | DEITADO #2 @P.1 | IP: 192.168.137.250 | PM = 1000 ms | PA = 3000 µs | NS = 10 amostras;
> 2019-06-16 04:00:32 | ANDAR #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 3000 µs | NS = 5 amostras;
> 2019-06-16 04:00:33 | DEITADO #2 @P.1 | IP: 192.168.137.250 | PM = 1000 ms | PA = 3000 µs | NS = 10 amostras;
> 2019-06-16 04:00:33 | CORRER #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 3000 µs | NS = 5 amostras;
> 2019-06-16 04:14:57 | ERROR (5) #1 @P.1 | IP: 192.168.137.38 - incapacidade de enviar mensagens de dados no ritmo d
efinido;
> 2019-06-16 04:15:25 | START #1 @P.1 | IP: 192.168.137.38 | PM = <DEFAULT> | PA = <DEFAULT> | NS = <DEFAULT>;
> 2019-06-16 04:15:26 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 1000 µs | NS = 5 amostras;
> 2019-06-16 04:15:27 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 1000 µs | NS = 5 amostras;
> 2019-06-16 04:15:28 | STOP <ALL> | Razão de paragem: USERIN;
> 2019-06-16 04:15:28 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 1000 µs | NS = 5 amostras;
> 2019-06-16 04:50:40 | Desconexão de concentrador | TCP/IP: 192.168.137.250:54359;
```

Figura 5.12 – Mensagens em tempo real enviadas pelo concentrador.

5.5.3 Leitura da base de dados

A leitura da base de dados para revelar os comportamentos dos sujeitos num intervalo definido pode ser feita através do seguinte comando:

database (<ISu/AREA>) (<TSI> <TSF>)

O utilizador pode escolher filtrar os dados por sujeito ou por área dependendo do argumento que escolhe (se for numérico positivo, é assumido que se trata dum sujeito, caso contrário, tratar-se-á duma área; a ausência deste argumento não irá aplicar nenhum filtro).

Os parâmetros TSI e TSF são, respetivamente, a *timestamp* inicial e a *timestamp* final, ambos no formato Y-M-D D:M:S, sendo apresentadas todas as mensagens entre o intervalo definido.

A ausência de TSF assumirá que o mesmo se trata do tempo atual, sendo as mensagens apresentadas a partir de TSI.

As mensagens apresentadas têm a seguinte estrutura:

* <TS> | <STATUS> #<ISu> @<AREA> ,

Dois exemplos de leitura da base de dados (filtrados por área e ISu, respetivamente) são os da Figura 5.13 e Figura 5.14.

```
database P.1 2019-06-16 03:59:42 2019-06-16 04:00:34
> Comportamentos em @P.1 de 2019-06-16 03:59:42 até 2019-06-16 04:00:34:
* 2019-06-16 03:59:42 | DEITADO #2 @P.1,
* 2019-06-16 03:59:42 | DEITADO #1 @P.1,
* 2019-06-16 03:59:43 | DEITADO #2 @P.1,
* 2019-06-16 03:59:43 | DEITADO #1 @P.1,
* 2019-06-16 03:59:44 | ANDAR #1 @P.1,
* 2019-06-16 04:00:31 | CORRER #2 @P.1,
* 2019-06-16 04:00:32 | DEITADO #2 @P.1,
* 2019-06-16 04:00:32 | ANDAR #1 @P.1,
* 2019-06-16 04:00:33 | DEITADO #2 @P.1,
* 2019-06-16 04:00:33 | CORRER #1 @P.1,
* 2019-06-16 04:00:34 | DEITADO #2 @P.1,
* 2019-06-16 04:00:34 | DEITADO #1 @P.1,
* 12 resultados;
```

Figura 5.13 – Leitura das mensagens de comportamentos armazenadas na base de dados e filtradas por área.

```
database 2 2019-06-16 03:59:42 2019-06-16 04:00:34
> Comportamentos de #2 de 2019-06-16 03:59:42 até 2019-06-16 04:00:34:
* 2019-06-16 03:59:42 | DEITADO #2 @P.1,
* 2019-06-16 03:59:43 | DEITADO #2 @P.1,
* 2019-06-16 04:00:31 | CORRER #2 @P.1,
* 2019-06-16 04:00:32 | DEITADO #2 @P.1,
* 2019-06-16 04:00:33 | DEITADO #2 @P.1,
* 2019-06-16 04:00:34 | DEITADO #2 @P.1,
* 6 resultados;
```

Figura 5.14 – Leitura das mensagens de comportamentos armazenadas na base de dados e filtradas por ISu.

5.5.4 Apresentação e ocultação das mensagens em tempo real

Para que utilizador possa interagir com o gestor de serviço, principalmente para poder atribuir áreas a sujeitos ou visualizar a lista de áreas disponíveis, mensagens armazenadas ou o menu, é-lhe dado a opção de ocultar as mensagens em tempo real para que estas não sobreponham o ecrã (o gestor continua a recebê-las e a armazená-las na base de dados). Para tal, deve ser usado o comando `hide`.

Este comando é automaticamente invocado ao serem usados os comandos `list`, `area`, `database` e `menu`.

Para que as mensagens voltem a ser apresentadas, deve ser usado o seguinte comando:

`show <MODE>`

Se MODE estiver presente e tiver um valor numérico positivo, serão revelados os valores dos sensores recebidos nas mensagens do tipo *DATA*, sendo apresentados na seguinte estrutura:

```
> <TS> | <STATUS> #<ISu> @<AREA> | IP: <SS_AD> | PM = <PM> ns | PA
= <PA> μs | NS = <NS> amostra(s):
* S1: (<SAX[0]>, <SAY[0]>, <SAZ[0]>) g, (<SGX[0]>, <SGY[0]>,
<SGZ[0]>) °/s, <ST[0]> °C,
* <...>,
S<N>: (<SAX[N-1]>, <SAY[N-1]>, <SAZ[N-1]>) g, (<SGX[N-1]>, <SGY[N-1]>,
<SGZ[N-1]>) °/s, <ST[N-1]> °C;
```

Se não for dado nenhum argumento a MODE ou se for dado com valor 0, não serão revelados os valores dos sensores.

A demonstração deste comando está exemplificada na Figura 5.15.

```
> 2019-06-16 05:20:51 | AGITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 μs | NS = 3 amostras;
> 2019-06-16 05:20:52 | ANDAR #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 μs | NS = 3 amostras;
> 2019-06-16 05:20:53 | ANDAR #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 μs | NS = 3 amostras;
> 2019-06-16 05:20:54 | CORRER #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 μs | NS = 3 amostras;
s 1
> A mostrar valores dos sensores em tempo real...
> 2019-06-16 05:20:55 | ANDAR #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 μs | NS = 3 amostras:
* S1: (-0.07, -0.51, 0.40) g, (-250.13, -39.12, 6.08) °/s, 22.57 °C,
* S2: (0.02, -0.62, 0.68) g, (-250.13, -83.99, 2.93) °/s, 23.28 °C,
* S3: (0.10, -0.57, 0.44) g, (-250.13, -112.90, -4.80) °/s, 22.57 °C;
> 2019-06-16 05:20:56 | ANDAR #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 μs | NS = 3 amostras:
* S1: (0.00, -0.22, 1.33) g, (-92.25, -6.47, 2.75) °/s, 23.28 °C,
* S2: (0.00, -0.23, 1.32) g, (-105.58, -9.90, -1.53) °/s, 22.62 °C,
* S3: (0.06, -0.27, 1.26) g, (-120.11, -7.89, -8.74) °/s, 22.57 °C;
> 2019-06-16 05:20:57 | DEITADO #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 μs | NS = 3 amostras:
* S1: (0.02, -0.27, 1.02) g, (16.31, -2.45, -3.96) °/s, 23.23 °C,
* S2: (0.03, -0.28, 1.03) g, (16.87, 0.24, -1.35) °/s, 22.57 °C,
* S3: (0.02, -0.28, 1.01) g, (17.56, 0.45, 0.03) °/s, 22.62 °C;
> 2019-06-16 05:20:58 | CORRER #1 @P.1 | IP: 192.168.137.38 | PM = 1000 ms | PA = 2000 μs | NS = 3 amostras:
* S1: (0.09, 0.33, 1.42) g, (248.17, 76.83, -46.45) °/s, 22.57 °C,
* S2: (0.10, 0.41, 1.24) g, (235.01, 70.01, -36.24) °/s, 22.57 °C,
* S3: (0.05, 0.53, 1.08) g, (142.03, 61.42, -30.76) °/s, 23.28 °C;
h
> Mensagens em tempo real ocultas, escreva "s"/"show" para voltar a revelá-las.
```

Figura 5.15 – Visualização de mensagens em tempo real com valores dos sensores.

5.6 Changelog

5.6.1 Fase B

- Implementação do Gestor de Serviço.

5.6.2 Fase C

- Identificação dos gestores de serviços.;
- Implementação de HTTP *Requests* para o Sistema Central.

5.6.3 Final

- Comunicação com múltiplos concentradores em simultâneo;
- Envio da lista de ISu aos concentradores;
- Otimização das fórmulas para deteção do comportamento físico:
 - Exclusão do eixo X em vez da inclusão apenas do eixo Z na deteção do comportamento 'deitado' visto que há diferentes posições que um paciente pode estar deitado;
- Implementação do ficheiro de configuração `setup.cfg`:
 - Achamos mais conveniente ter o identificador do serviço, portas e endereços relevantes guardados num ficheiro em vez de *hardcoded*;
- Alterações mínimas na interface de utilizador e do armazenamento e leitura da base de dados e otimização geral do código do programa.

6 Sistema Central

6.1 Protocolo aplicacional de comunicação

O protocolo aplicacional de comunicação entre um gestor de serviço e o sistema central foi baseado em *web-service*, construído à base de HTTP *Requests* feitos pelo gestor quando é necessário enviar mensagens para o sistema central.

6.1.1 Registo de serviço

Sempre que o gestor de serviço se liga, é feito um *request* ao sistema central com os seguintes parâmetros:

- Identificação do serviço;
- Designação do serviço.

6.1.2 Registo de utilizador

Sempre que o gestor de serviço lê ou atualiza o ficheiro de áreas atribuídas a sujeitos, é feito um *request* ao sistema central por cada ISu existente com os seguintes parâmetros:

- Identificação do serviço;
- Identificação do sujeito;
- Área atribuída ao sujeito.

6.1.3 Mensagens de monitorização

Quando for gerada uma mensagem relativamente à monitorização dum sujeito, serão enviados *requests* com os seguintes parâmetros:

- Identificação do serviço;
- Identificação do sujeito;
- *Timestamp* em formato Y-M-D H:M:S;
- Condição física do sujeito.

6.2 Base de dados relacional

O sistema central disponibiliza uma base de dados para armazenar informação relevante para ser analisada. Esta base de dados inclui tabelas de armazenamento de serviços, utilizadores, pacientes e mensagens provenientes dos gestores de serviços. Para o sistema central aceder ao seu conteúdo ou para enviar conteúdo, são feitos HTTP *Requests*.

O serviço usado para o domínio da base de dados e de todo o sistema central foi o “Duck DNS”[9], controlada no serviço “phpMyAdmin”[10], alojado num Raspberry Pi 3 Model B+.

A manipulação dos ficheiros do sistema central foi feita através do programa de transferência de ficheiros “FileZilla”[11].

6.2.1 Serviços

Quando um gestor de serviço envia um *request* de registo, é inserida (caso esteja inexistente) uma nova entidade à tabela de serviços com os seguintes atributos:

- ISe – identificação numérica do serviço (chave primária);
- Designação do serviço.

6.2.2 Utilizadores

Quando um gestor de serviço envia um *request* de registo, será inserida (caso esteja inexistente) uma nova entidade à tabela de utilizadores com os seguintes atributos:

- ISu – identificação numérica do utilizador (chave primária);
- ISe – identificação numérica do serviço (0 = administração) (chave primária e estrangeira (atributo ISe da tabela Serviços));
- Área;
- E-mail;
- Nome de utilizador – parte do e-mail anterior ao ‘@domínio’;
- Palavra-passe.

O e-mail, nome de utilizador e palavra-passe são atribuídos pela área de registo do sistema central, sendo a respetiva entidade atualizada.

Inserções de utilizadores com serviço administrativo são feitos no registo do sistema central.

O administrador (ISe = 0) com ISu = 0 é o Super Administrador.

6.2.3 Acessos

No registo de um administrador, são-lhe atribuídos acessos a serviços específicos para que este os possa monitorizar e é inserida uma nova entidade à tabela de acessos com os seguintes atributos:

- ISu – identificação numérica do administrador (chave primária e estrangeira (atributo ISu da tabela Utilizadores));
- ISe – identificação numérica do serviço acedido (chave primária e estrangeira (atributo ISe da tabela Utilizadores));

6.2.4 Pacientes

A tabela de pacientes é preenchida ao ser registado um novo utilizador pelo sistema central com serviço não administrativo e tem os seguintes atributos:

- ISu – identificação numérica do sujeito (chave primária e estrangeira (atributo ISu da tabela Utilizadores));
- ISe – identificação numérica do serviço (chave primária e estrangeira (atributo ISe da tabela Utilizadores));
- Nome do paciente;
- URL da imagem de perfil;
- Sexo;
- Data de nascimento;
- Altura;
- Peso.

6.2.5 Mensagens do Gestor de Serviço

A tabela que contém as mensagens do gestor de serviço será preenchida sempre que for recebida uma mensagem do mesmo, tendo os seguintes atributos:

- ISe – identificação do (chave primária e estrangeira (atributo ISe da tabela Serviços));
- ISu – identificação do sujeito (chave estrangeira (atributo ID da tabela Utilizadores));
- *Timestamp* da mensagem (chave primária);
- Comportamento – “DEITADO”, “PARADO”, “ANDAR”, “CORRER”, “AGITADO” e “QUEDA”.

6.3 Estatísticas

As estatísticas que podem ser visualizadas pelos pacientes são:

- Lista atualizável em tempo real de todas as mensagens provenientes do respetivo gestor de serviço acerca do paciente. As mensagens são relativas aos comportamentos físicos do paciente e podem ser filtradas por um só comportamento ou a partir de uma *timestamp* à escolha;
- Gráfico de barras atualizável em tempo real relativo aos comportamentos físicos do paciente;
- Dados do paciente (serviço, área, nome de utilizador, imagem de perfil, nome, idade, altura e peso).

Os administradores têm acesso às estatísticas e dados de todos os pacientes e dos serviços que têm acesso:

- *Pie chart* dividido pelo número de pacientes de cada serviço;
- Gráfico de barras com o número de mensagens de cada comportamento físico em cada serviço.

6.4 Interface do utilizador

A interface do utilizador foi desenvolvida usando as linguagens HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheets*), PHP (*Hypertext Preprocessor*) e JavaScript, baseada na *template* “Klorofil” [12].

O endereço do *web site* é <http://azkeryon.duckdns.org/lti/> [13].

6.4.1 Início de sessão (*login*)

A página inicial do sistema central consiste numa página de *login*, apresentada na Figura 6.1, onde os utilizadores devem introduzir o seu endereço e-mail e palavra-passe para que possam aceder ao conteúdo que têm acesso.

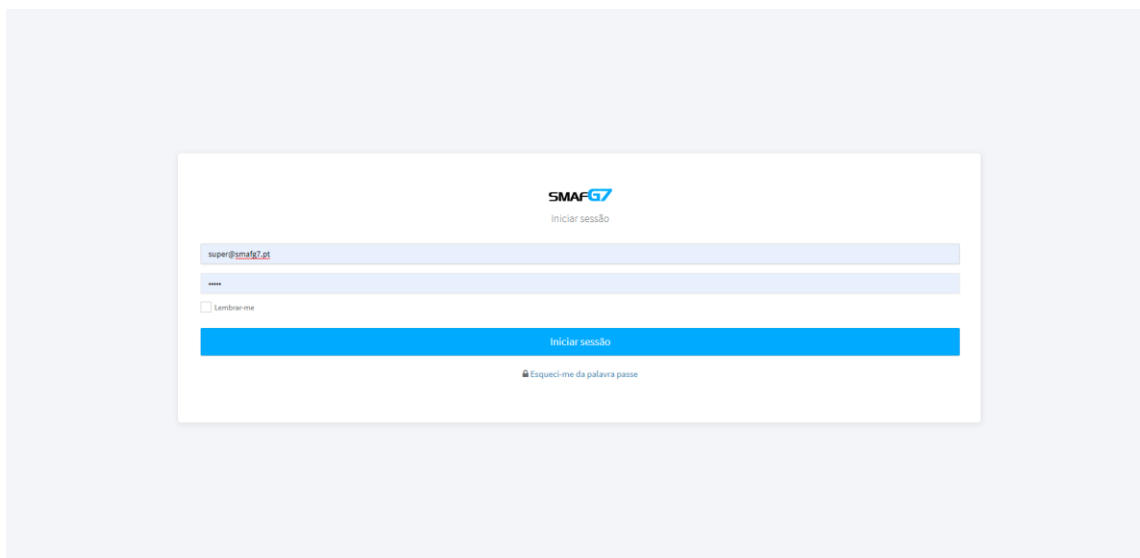


Figura 6.1 – UI de início de sessão do Super Administrador.

O botão “Esqueci-me da palavra-passe” faz com que a palavra-passe do utilizador lhe seja enviada por e-mail, como é demonstrado na Figura 6.2.

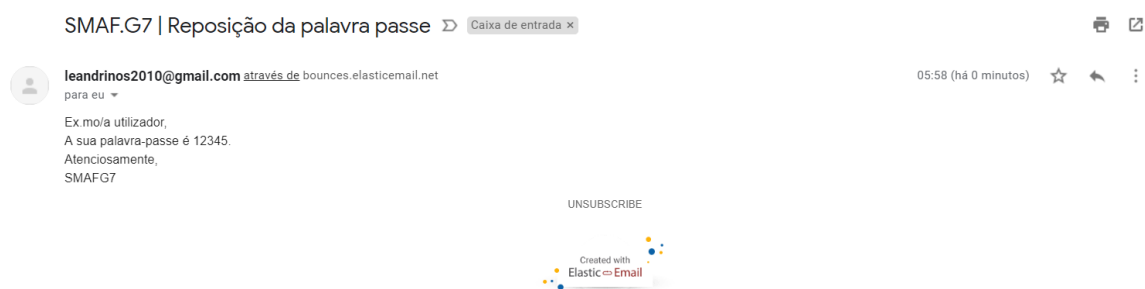


Figura 6.2 – E-mail enviado para reposição da palavra-passe.

O serviço de e-mail foi configurado através do serviço “SMTPJS”[14].

6.4.2 Dados estatísticos do paciente

Os pacientes, depois de iniciarem sessão, são apresentados pelas suas estatísticas (dados do paciente e registo dos comportamentos físicos), apresentadas na Figura 6.3.

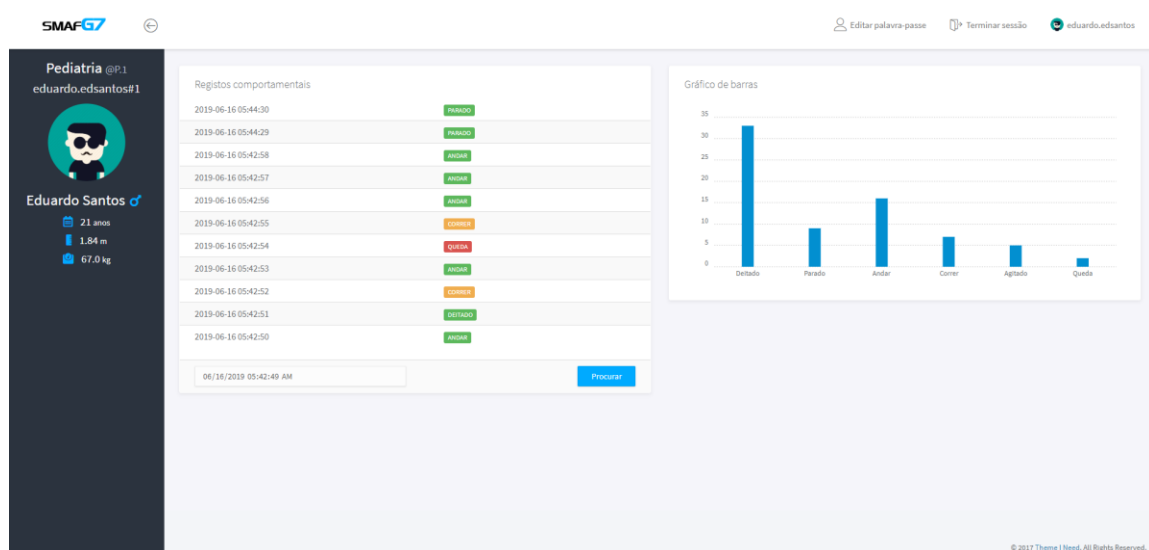


Figura 6.3 – Visualização da página de paciente pelo próprio.

Os administradores têm acesso às estatísticas de todos os pacientes dos serviços monitorizados, como o paciente da Figura 6.4.

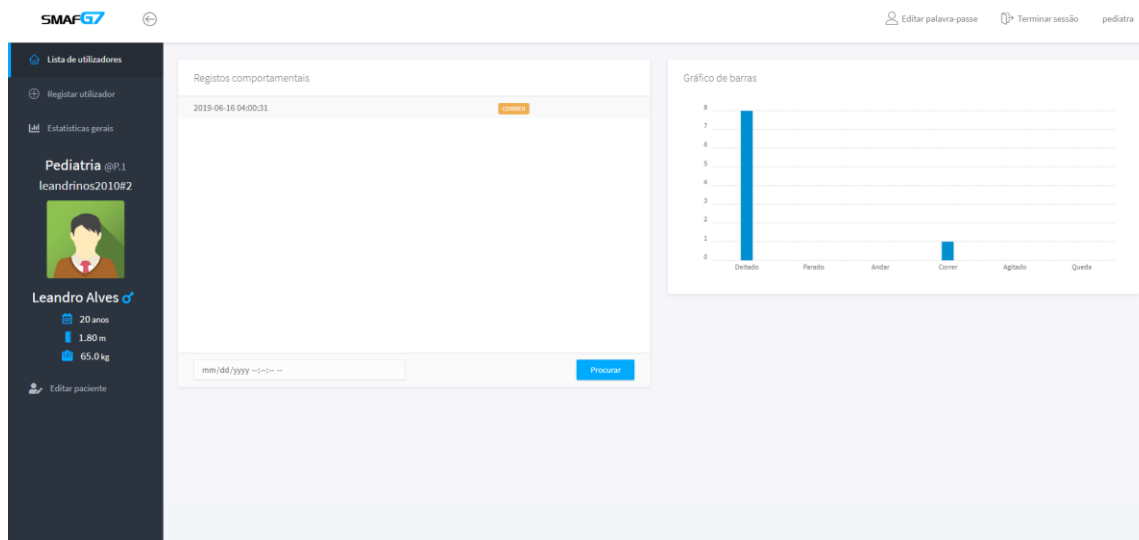


Figura 6.4 – Filtragem dos registos comportamentais a partir do gráfico de barras.

6.4.3 Lista de pacientes

Para que os administradores acedam às estatísticas de cada paciente, é lhes apresentado depois de iniciar sessão a lista de todos os pacientes registados nos serviços monitorizados, agrupados pelos mesmos. Nesta lista, demonstrada na Figura 6.5, é apresentado em cada paciente a imagem de perfil, nome de utilizador, ISu, nome completo, área e estado atual do mesmo.

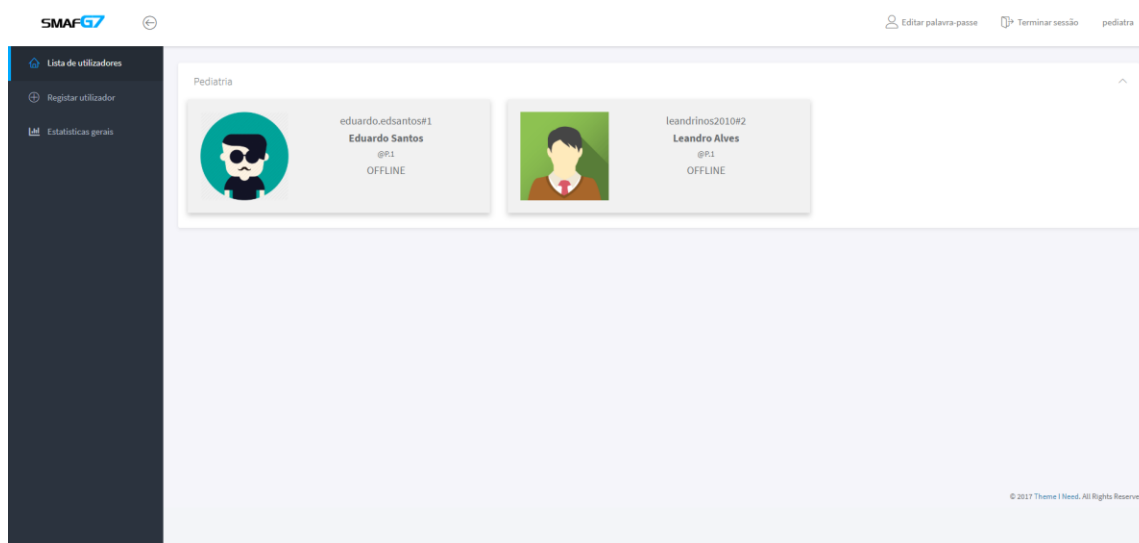


Figura 6.5 – Visualização da lista de pacientes por um administrador regular.

O Super Administrador também tem acesso à lista de administradores registados, como se pode ver na Figura 6.6.

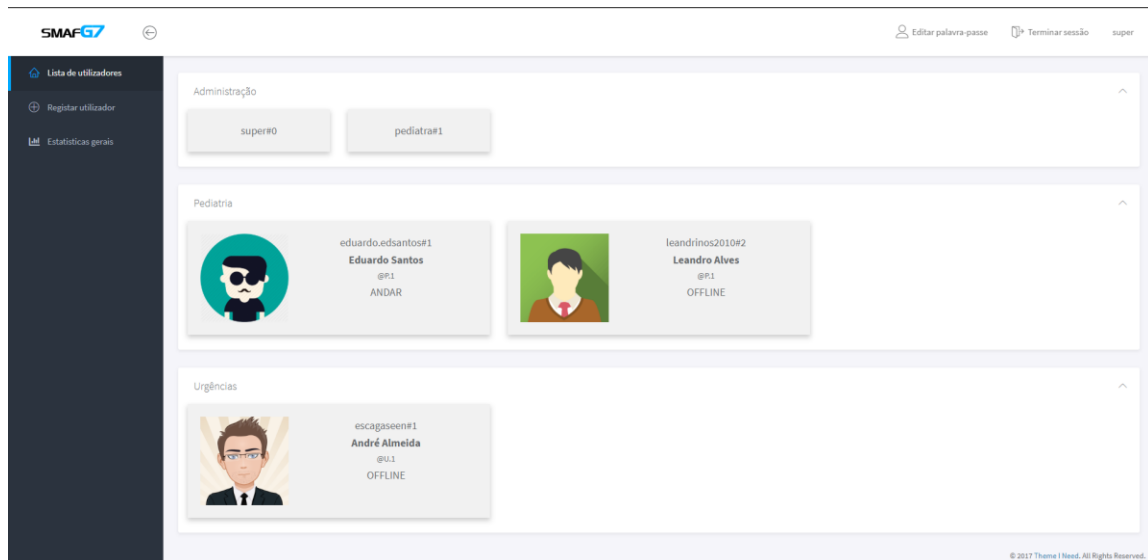


Figura 6.6 – Visualização da lista de administradores e pacientes pelo Super Administrador.

6.4.4 Registo de utilizadores

Os administradores têm acesso ao registo de pacientes, apresentado na Figura 6.7, tendo de preencher os campos de serviço, ISu, e-mail, palavra-passe, nome completo, imagem de perfil, data de nascimento, sexo, altura e peso. As imagens são selecionadas a partir do computador e são automaticamente enviadas para o *site* de hospedagem de imagens “Imgur”[15], gerando um URL que é enviado para a base de dados.

Figura 6.7 – Registo de um paciente.

Depois de um utilizador ser registado, é enviado para o e-mail do mesmo uma mensagem de boas vindas, demonstrada na Figura 6.8, juntamente com a sua palavra-passe e uma hiperligação para o *website*.

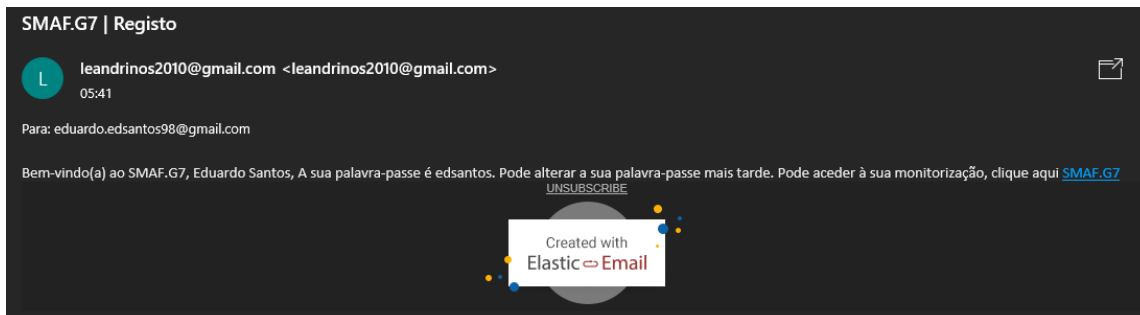


Figura 6.8 – E-mail de boas vindas enviado ao utilizador que acabou de ser registado.

O Super Administrador pode também registar administradores, como está apresentado na Figura 6.9, definindo o serviço do utilizador como “Administração”. Em vez de serem preenchidos os dados de paciente, são seleccionados os serviços que o administrador terá acesso. O ISu é atribuído automaticamente pela base de dados.

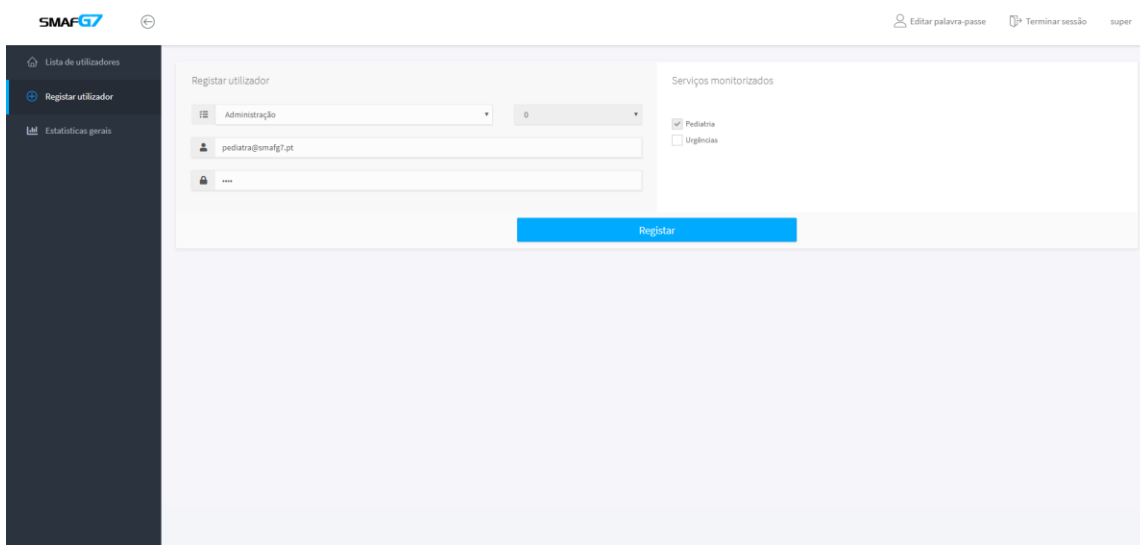


Figura 6.9 – Registo de um administrador.

6.4.5 Estatísticas gerais

Os administradores podem aceder às estatísticas gerais anteriormente referidas, apresentadas na Figura 6.10.

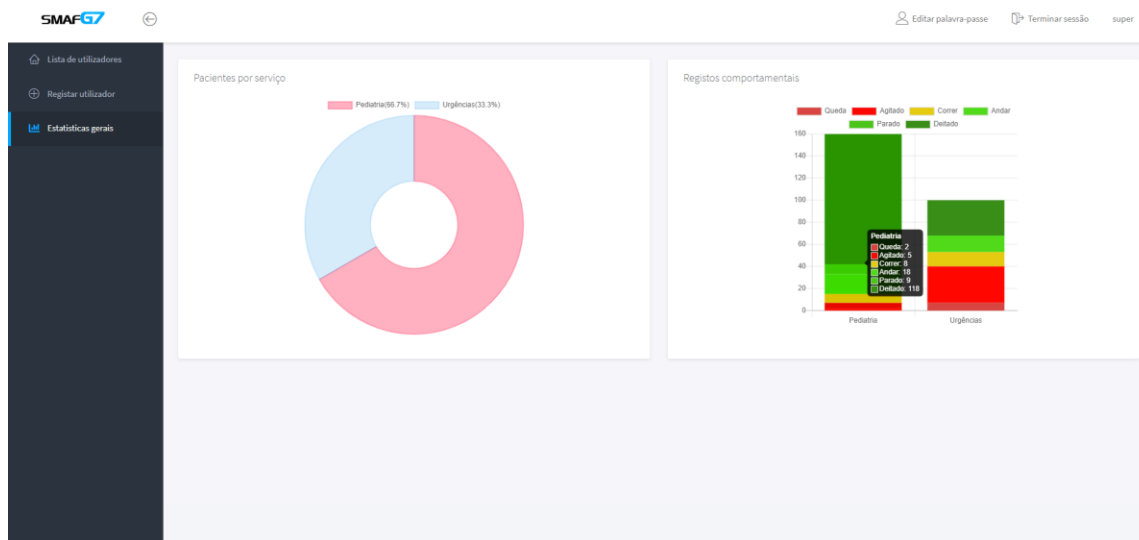


Figura 6.10 – Estatísticas gerais.

Apenas os serviços monitorizados pelo administrador são revelados.

6.4.6 Edição e remoção de utilizadores

Depois de ser selecionado um paciente para serem visualizados os seus dados estatísticos, os administradores podem editar os seus dados de paciente (apenas nome completo, imagem de perfil, data de nascimento, sexo, altura e peso), apresentados na Figura 6.11, ou remover o registo da conta do mesmo.

The screenshot shows the 'Editar dados do paciente' form in the SMAF67 dashboard. The form is for a patient named Leandro Alves. It includes a profile picture placeholder, a date of birth field (03/18/1999), a gender dropdown menu (Masculino), and fields for height (1.80 m) and weight (65.0 kg). There are three buttons at the bottom: 'Cancelar' (blue), 'Concluir' (blue), and 'Apagar' (red). The left sidebar shows the 'Estatísticas gerais' section with a patient profile for Leandro Alves, including his age (20 anos), height (1.80 m), and weight (65.0 kg).

Figura 6.11 – Modificação dos dados de um paciente por um administrador.

O Super Administrador pode também remover o registo de conta de outros administradores, como está apresentado na Figura 6.12, para tal, apenas deve seleccioná-lo na lista de administradores. Só podem ser alterados os serviços monitorizados.

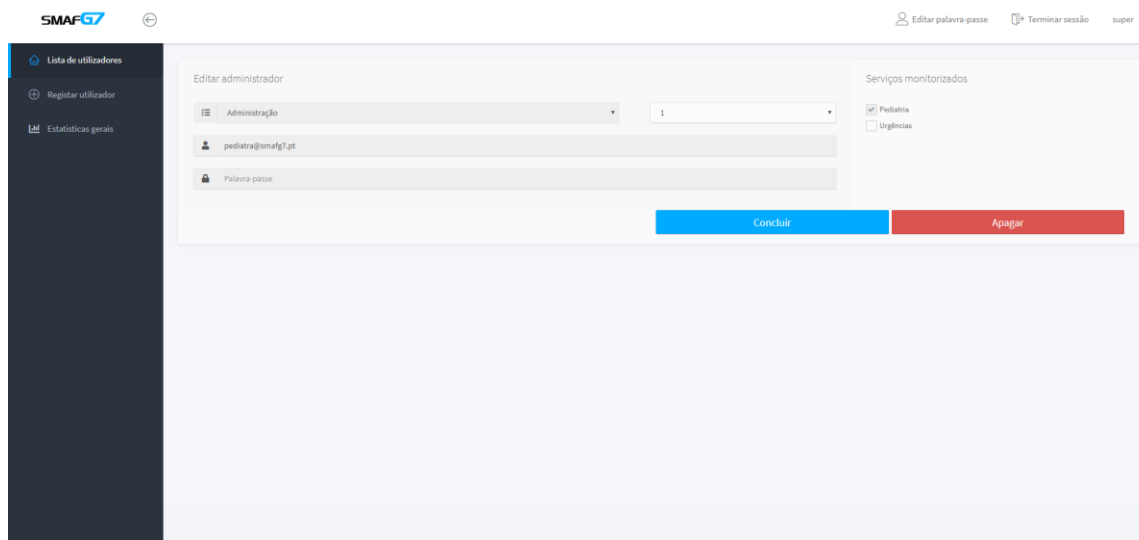


Figura 6.12 – Modificação dos serviços de um administrador pelo Super Administrador.

Todos os utilizadores, incluindo pacientes, podem editar as próprias palavras-passe, como se pode ver na Figura 6.13.

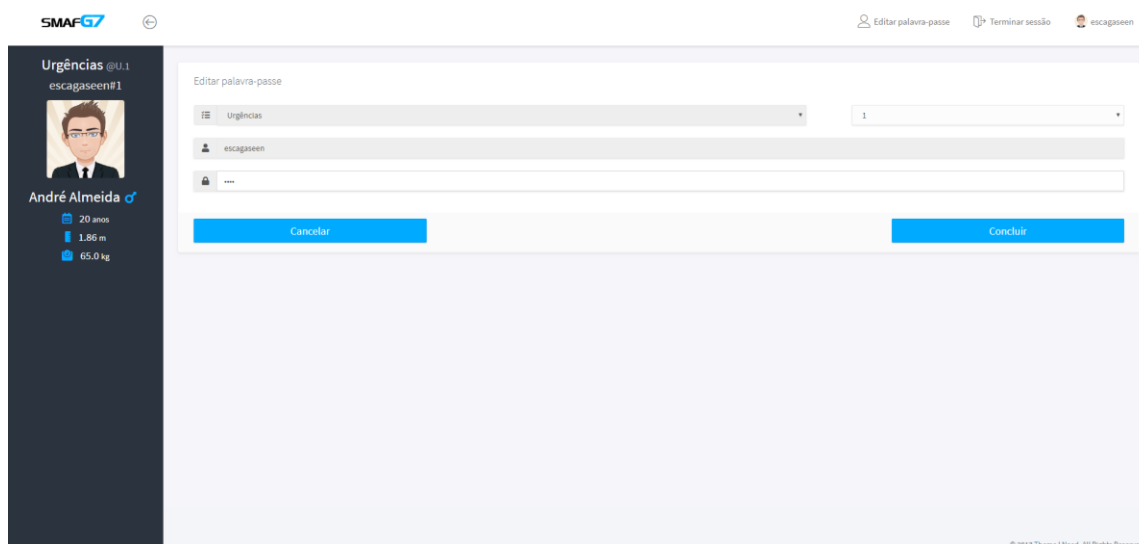


Figura 6.13 – Alteração de palavra-passe.

6.5 Changelog

6.5.1 Fase C

- Implementação do Sistema Central.

6.5.2 Final

- Implementação das estatísticas gerais;
- Substituição do nome de utilizador por um endereço de e-mail;
- Envio de mensagens por e-mail contendo a respetiva palavra-passe;
- Atribuição e edição de acessos a serviços específicos a administradores;

- Alteração nas tabelas e colunas da base de dados:
 - Adição da coluna de e-mail à tabela de utilizadores;
 - Criação da tabela de acessos de administradores a serviços;
- Alterações mínimas na interface de utilizador.

7 Conclusões

No desenvolvimento desta fase foram corrigidos os erros, preenchidas as lacunas e complementado todo o *hardware* e *software* desenvolvidos nas fases anteriores. Além disso foram implementadas novas funcionalidades opcionais.

Surgiram algumas dificuldades ao longo desta fase em termos de gestão do tempo para o desenvolvimento da mesma o que nos impediu de otimizar os sistemas da arquitetura a 100% mas achamos que desenvolvemos o suficiente para terminar esta fase e este projeto com sucesso.

8 Referências

- [1] Espressif.com. (2016). [online] Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf [Acedido a 18/02/2019].
- [2] Alldatasheet.com. (2012). *MPU-6050 Datasheet(PDF) - List of Unclassified Manufacturers*. [online] Disponível em: <http://www.alldatasheet.com/datasheet-pdf/pdf/517744/ETC1/MPU-6050.html> [Acedido a 18/02/2019].
- [3] Rowberg, J. *jrowberg/i2cdevlib*. [online] GitHub. Disponível em: https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050/examples/MPU6050_raw [Acedido a 11/02/2019].
- [4] Maker Pro. *How to Interface Arduino and the MPU 6050 Sensor | Arduino*. [online] Disponível em: <https://maker.pro/arduino/tutorial/how-to-interface-arduino-and-the-mpu-6050-sensor> [Acedido a 25/02/2019] [baseado].
- [5] Ródenas, L. (2014). *MPU6050 Calibration*. [online] Forum.arduino.cc. Disp. em: <https://forum.arduino.cc/index.php?topic=514498.msg3510560#msg3510560> [Acedido a 25/02/2019].
- [6] GitHub. (2016). *taranais/NTPClient*. [online] Disponível em: https://github.com/taranais/NTPClient?fbclid=IwAR31hn1SPGuobKID_Kr_du-FrizUUTZRIXjtb9a-lm1j74WxAsRz0HCq5iyo [Acedido a 03/03/2019].
- [7] Gieben, M. (2017). *udp server in C*. [online] Gist. Disponível em: <https://gist.github.com/miekg/a61d55a8ec6560ad6c4a2747b21e6128> [Acedido a 21/02/2019] [baseado].
- [8] GeeksforGeeks. (2017). *Socket Programming in C/C++ - GeeksforGeeks*. [online] Disponível em: <https://www.geeksforgeeks.org/socket-programming-cc/> [Acedido a 22/03/2019] [baseado].
- [9] Duckdns.org. *Duck DNS*. [online] Disponível em: <https://www.duckdns.org> [Acedido a 06/05/2019].
- [10] phpMyAdmin. *phpMyAdmin*. [online] Disponível em: <https://www.phpmyadmin.net/> [Acedido a 06/05/2019].
- [11] Filezilla-project.org. *FileZilla - The free FTP solution*. [online] Disponível em: <https://filezilla-project.org/> [Acedido a 06/05/2019].
- [12] Theme I Need. (2017). *Free Bootstrap Admin Template - Klorofil*. [online] Disponível em: <https://www.themeinneed.com/downloads/klorofil-free-bootstrap-admin-template/> [Acedido a 12/05/2019].
- [13] Azkeryon.duckdns.org. (2019). *SMAF.G7*. [online] Disponível em: <http://azkeryon.duckdns.org/lti/>.
- [14] Smtpps.com. *Smtpps.com - Send Email from JavaScript*. [online] Disponível em: <https://www.smtpps.com/> [Acedido a 14/05/2019].
- [15] Imgur. *Imgur*. [online] Disponível em: <https://imgur.com/> [Acedido a 31/05/2019].