

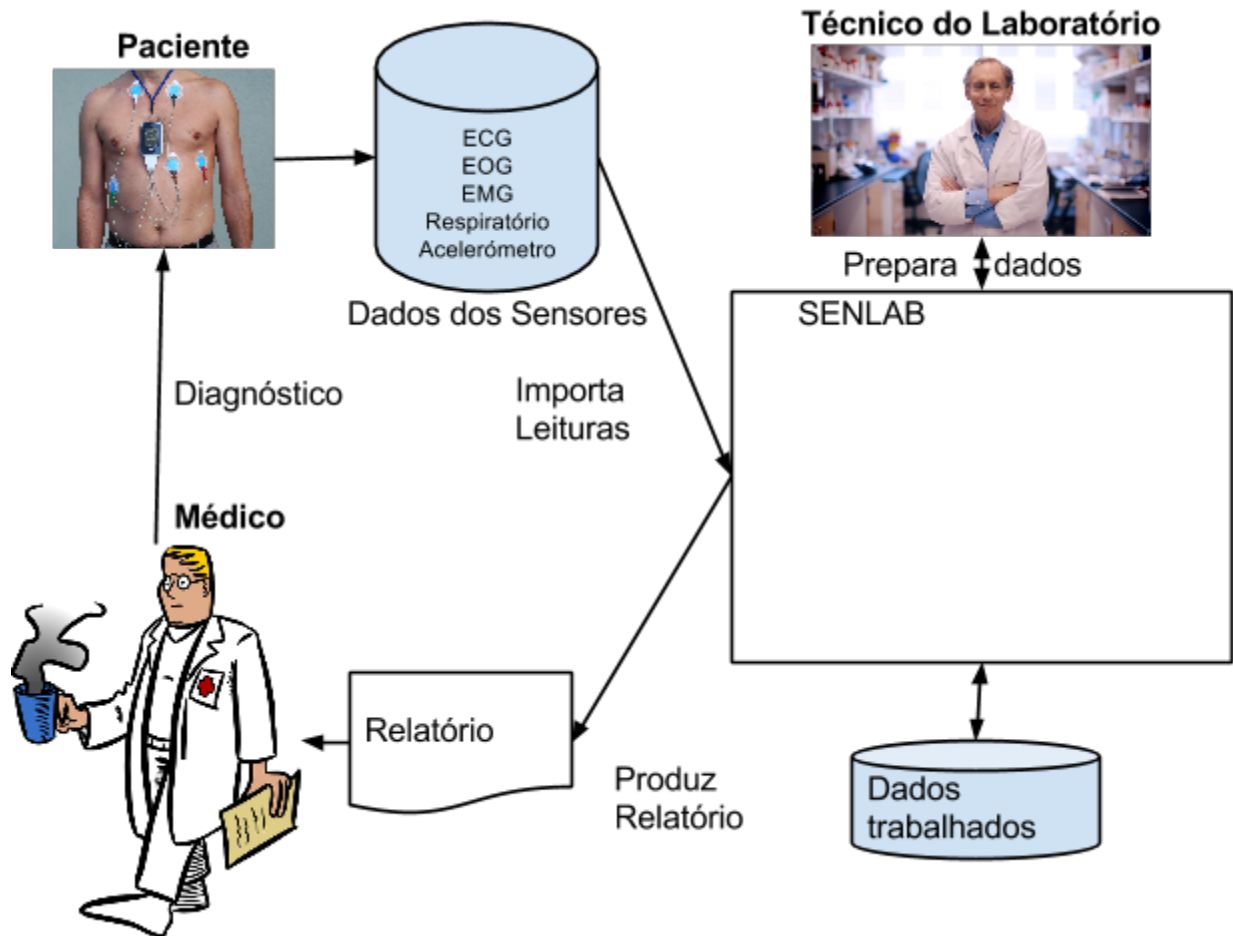
# Programação Orientada pelos Objetos B

## Trabalho prático 2014/2015

### SENLAB

#### Laboratório de Exames Fisiológicos

Versão 1.3, 23 de Maio de 2015



## 1. Introdução / Motivação

Os sinais fisiológicos têm um papel preponderante no diagnóstico de diversas patologias. Para os obter é necessário dar uso a sensores que diferem dependendo do tipo de sinal que se pretende captar. A título exemplificativo consideremos o eletrocardiograma (ECG) que mede sinais cardíacos, o eletrooculograma (EOG) que mede a atividade ocular, a electromiografia (EMG) que mede a atividade eléctrica produzida pelos músculos, um sensor de respiração e um acelerómetro que mede a aceleração nos três eixos. Todos estes são sinais que podem ser captados e analisados, à posteriori ou em tempo real, para se obter um diagnóstico sobre o estado de saúde dos pacientes.

Hoje em dia existem equipamentos para fazer estas medições, no entanto os mesmos têm diversas desvantagens: são relativamente grandes e pesados, de elevado custo e principalmente são diferentes por cada sinal que se pretende adquirir. Com este projeto pretende-se construir o software que recebe dados de um dispositivo médico captados de diversos sinais fisiológicos, tantos quanto se queira. Pretende-se com este sistema que o técnico/operador possa analisar o estado clínico de diversos pacientes, com base nas aquisições feitas dos diversos sinais, e poder extrair alguma informação clínica relevante, sob a forma de relatório, para apoiar o diagnóstico.

## 2. Prazo de Entrega

O trabalho deve ser entregue via moodle, até às **24h00 do dia 28 de Maio de 2015 (5ª-feira)**. Serão permitidas submissões no moodle fora do prazo, com uma penalização de **2 valores** por cada dia em atraso, até 30 de Maio de 2015 (sábado) inclusive.

### 3. Funcionalidades

O projeto visa o desenvolvimento do software para apoiar um sistema físico/hardware de aquisição, dando uso a equipamentos que contêm diversos sensores, de modo a permitir o processamento e análise de diversos sinais fisiológicos. A seguir são descritos os dados, funcionalidades e regras do sistema.

#### a. Leituras obtidas pelo dispositivo médico

O dispositivo médico apresentado neste projeto consiste num equipamento com diversos sensores. Há dois tipos de sensores (também chamados de canais em certos meios): os elétricos (ECG, EOG e EMG) e os não elétricos (sensor respiratório e acelerómetro). Qualquer um dos dois tipos de sensores tem uma variável que indica o seu nome e uma variável a indicar se já houve medição (apenas se pode realizar uma única medição por cada tipo de sensor), um vetor com os dados medidos e o tempo da medição realizada. Os dados dos sensores elétricos são dados com casas decimais, em oposto aos sensores não elétricos. Tem de ser possível saber se o sensor em particular é eléctrico ou não.

O dispositivo médico faz aquisições (em diversos canais) a cada segundo. O dispositivo médico permite realizar um conjunto de aquisições seguidas de diferentes sensores. Por exemplo, a cada segundo vamos obter leituras:

ECG ; EMG ; ...

guardando sempre os resultados de cada sensor para processamento. O software a desenvolver deve também disponibilizar uma análise do estado clínico do paciente (saudável ou não saudável) por sensor e pelo conjunto dos sensores medidos até então (apenas um não equivale a resultado de não saudável).

#### b. Sensores

Existem diversos algoritmos relativamente complexos para analisar dados de exames médicos, no entanto o mesmo não se encontra no âmbito desta disciplina pelo que no projeto são utilizados dados fictícios e propostas métricas para cada um dos sensores.

Para o grupo dos sensores elétricos apenas se pode realizar medição se a idade do paciente for inferior a 60 anos. De seguida são apresentados requisitos particulares de cada tipo de sensor para o paciente ser considerado saudável:

##### i. ECG

A média dos dados adquiridos é superior a 120 e inferior a 200.

ii. **EOG**

A média dos dados obtidos é superior a 22,5 e inferior a 27.

iii. **EMG**

O desvio padrão dos dados não excede 5.

iv. **sensor respiratório**

Dados apresentam um período de inspiração (conjunto consecutivo de 1's) com um período de expiração (conjunto consecutivo de 0's) - denominemos por 1 ciclo respiratório - em média por cada 15 segundos, a título exemplificativo: 111111110000000. Não pode apresentar em média por cada 20 segundos mais que 2 ciclos ou menos que 0,7 ciclos.

v. **acelerómetro**

Os dados depois de ordenados não podem diferir por cada cada dois valores seguidos de um valor superior a 5.

### **c. Pacientes**

Neste projeto iremos utilizar dados fictícios de pacientes. Cada paciente é caracterizado pelo seu número de identificação (ID), nome, idade, sexo, se toma medicamentos, se fuma, número de leituras realizadas, e pelo conjunto de leituras.

## 4. Funcionalidades

Pretende-se elaborar um interpretador de comandos que permita ao técnico de laboratório: importar resultados dos exames de um ou mais pacientes a partir de ficheiros, remover leituras de determinados sensores nos exames (por ter sido identificado, por exemplo, que esses canais estavam descalibrados) e persistir as modificações feitas pelo técnico.

Os comandos e funcionalidades a serem suportados pelo sistema estão classificadas doravante neste documento em 3 categorias:

- **S**, ou **básico**, onde a implementação é de cariz obrigatório para que dê direito a frequência, e a nota do projeto possa ser avaliada na gama de  $10 < \text{nota} \leq 12$ ;
- **L**, ou **desejável**,  $12 < \text{nota} < 17$  ;
- **XXL**, ou **super genial**,  $\text{nota} > 17$  .

Atenção que implementar estas funcionalidades não garantem automaticamente a nota mínima no intervalo apresentado. Outros critérios terão de ser tidos em conta, como critérios de boa prática de programação POO discutidas ao longo do semestre.

### (S/XXL) - Importando Exames da máquina (dados “raw”)

Cada ficheiro cujo nome tem o formato **exam-<Número>.lab** (ex. “exam-001.lab”) contém os valores da aquisição dos sensores durante o período de realização de um Exame. O número da leitura registada no sistema é o número que consta no nome do ficheiro. Os dados a carregar consistem simplesmente no seguinte formato por linha:

```
[ID:Int]TAB[Idade:Int]
[Nome:String]
[Sexo:Char F ou M]TAB[Fuma:Char S ou N]TAB[Medicamentos:Char S ou N]
[Data:String]
[ECG]TAB[EOG]TAB[EMG]TAB[Respiratorio]TAB[Acelerometro]
[Floats|Inteiros]
```

No caso de o grupo pretender cumprir com o grau de dificuldade S, considera-se que os leitores estão todos presentes, e **que existem medidas para todos**, pela ordem apresentada anteriormente, ou seja ECG EOG EMG Respiratorio Acelerometro (separados por tabs).

No caso de o grupo pretender chegar ao grau de dificuldade XXL, considera-se que os leitores podem não estar todos presentes e podem vir por ordem diferente da apresentada acima, por exemplo:

```
[ID:Int]TAB[Idade:Int]
[Nome:String]
[Sexo:Char F ou M]TAB[Fuma:Char S ou N]TAB[Medicamentos:Char S ou N]
[Data:String]
[Acelerometro]TAB[EMG]
[Inteiro|Float]
```

### (XXL) - Persistindo alterações

No início da execução, o programa deve carregar, se existir o ficheiro lab.db, todos os dados dos Exames anteriormente gravados pelo sistema em sessões anteriores.

### (L) - Relatório do técnico de laboratório

Relatório que adicionalmente aos dados tratados pelo técnico relativos a um determinado paciente num determinado exame ou histórico de exames. Adicionalmente será dado um parecer do técnico sobre os dados apresentados. Este relatório implica a geração de um ficheiro **relatorio.txt** com a respectiva informação.

### Manipulação de informação

As operações de manipulação de informação (inserir, apagar, listar), serão detalhadas na Secção 4. Comandos. Em seguida pode-se ver um conjunto de dados exemplificativos dos valores obtidos pelos sensores:

| ID            | 1234 | Idade | 55    | Nome         | Alberto Caminha | Sexo | M |
|---------------|------|-------|-------|--------------|-----------------|------|---|
| Time \ Sensor | ECG  | EOG   | EMG   | Respiratório | Acelerómetro    |      |   |
| 0             | 0.00 | 26.09 | 53.76 | 1            | 2               |      |   |
| 1             | 0.00 | 17.36 | 53.21 | 1            | 7               |      |   |
| 2             | 0.00 | 28.77 | 53.63 | 1            | 6               |      |   |
| 3             | 0.00 | 16.02 | 50.75 | 1            | 4               |      |   |
| 4             | 0.00 | 30.84 | 52.81 | 1            | 1               |      |   |
| 5             | 0.00 | 29.26 | 51.43 | 1            | 0               |      |   |
| 6             | 0.00 | 24.59 | 53.37 | 1            | 10              |      |   |
| 7             | 0.00 | 19.61 | 54.15 | 1            | 8               |      |   |
| 8             | 0.00 | 26.43 | 52.33 | 1            | 6               |      |   |
| 9             | 0.00 | 22.39 | 53.73 | 1            | 16              |      |   |
| 10            | 0.00 | 22.68 | 53.68 | 1            | 13              |      |   |
| 11            | 0.00 | 21.94 | 51.21 | 0            | 20              |      |   |
| 12            | 0.00 | 18.17 | 53.69 | 0            | 25              |      |   |
| 13            | 0.00 | 29.53 | 51.88 | 0            | 24              |      |   |
| 14            | 0.00 | 21.73 | 54.22 | 0            | 14              |      |   |
| 15            | 0.00 | 22.73 | 51.87 | 0            | 16              |      |   |
| 16            | 0.00 | 22.25 | 54.01 | 0            | 29              |      |   |
| 17            | 0.00 | 15.97 | 50.65 | 0            | 40              |      |   |
| 18            | 0.00 | 31.62 | 54.54 | 0            | 46              |      |   |

### Assuma a seguinte constantes no sistema:

Tempo das leituras nos sensores: 120 segundos

## 5. COMANDOS

Tal como treinado nas aulas práticas, o sistema especificado organiza a sua interação com o utilizador final por meio de um interpretador de comandos. Esta secção descreve os comandos que o sistema deve ser capaz de aceitar e executar. Nos exemplos apresentados, o texto a negrito representa a retroacção escrita pelo sistema, na consola, ao executar o comando com sucesso. Pode assumir que o utilizador não cometerá erros na introdução de argumentos nos comandos, para além dos descritos neste enunciado. Ou seja, apenas tem de tratar as situações de erro descritas neste enunciado.

Embora os exemplos dos comandos usem **maiúsculas**, o programa deve também aceitar os comandos em **minúsculas**.

### 5.1. (S) Comando S - Sair

Este comando termina a execução do programa. O comando não necessita de argumentos. O resultado a produzir deve ser o seguinte:

```
>S
Sessão terminada. Até à próxima.
```

Este comando tem sempre sucesso. Após a sua execução, o programa termina.

### 5.2. (S) Comando Ajuda

Este comando mostra a lista de comandos disponíveis para o utilizador. O comando não necessita de argumentos. O resultado a produzir deve ser o seguinte:

```
>Ajuda
*****
* Ajuda
*****
S   - Sair do programa
G   - Gravar

CNL - Carregar Novas Leituras (nome)

AS - Apagar sensor por código de Leitura
AL - Apagar Leitura
AP - Apagar Paciente

NP - Números de pacientes e leituras no sistema
```

```
LP - Listar pacientes
LE - Listar leituras feitas a um determinado paciente
LS - Listar pacientes saudáveis
LD - Listar pacientes com patologia

REA - produzir um relatório do histórico do paciente por datas
RES - produzir um relatório simples do paciente por datas
*****
```

Este comando tem sempre sucesso.

### 5.3. (S) Comando CNL (Carregar Nova Leitura)

Este comando lê uma nova Leitura e associa-o ao respectivo paciente no sistema, caso já tenha sido inserido previamente. Caso não haja no sistema, o paciente é inserido como novo e o programa informa disso.

```
>CNL
Qual o nome do Ficheiro?
>exam-1234.lab
Ficha de paciente Sara Gamas introduzida no sistema
Id:33234
Nome:Sara Gamas
Idade:35
Sexo:F
Fuma:S
Toma medicamentos:S

Foi introduzida no sistema a leitura de 22/4/2015 com código 1234

>CNL
Qual o nome do Ficheiro?
>exam-4567.lab
Foi introduzido no sistema a leitura de 2/3/2015 com código 4567 na
ficha do paciente:
Sara Gamas

>CNL
Qual o nome do Ficheiro?
>examo.lab
Ficheiro inexistente
```



#### 5.4. (XXL) Comando AS - Apagar sensor por número de leitura

```
>AS
Qual o código da leitura?
>1234
Leitura 1234 - Sensores disponíveis: ECG(1),EOG(2), EMG(3)
Qual o número do sensor que pretende apagar?
(caso nenhum, introduzir 0)
>4
não existe esse sensor
Leitura 1234 - Sensores disponíveis: ECG(1),EOG(2), EMG(3)
Qual o número do sensor que pretende apagar?
(caso nenhum, introduzir 0)
>2
apagado sensor EOG
Leitura 1234 - Sensores disponíveis: ECG(1), EMG(2)
Pretende apagar mais algum sensor?
(caso nenhum, introduzir 0)
>0
```

#### 5.5. (L) Comando AL- Apagar Leitura

```
>AL
Qual o código da leitura?
>1234
Apagar leitura 1234 de 22/4/2015 pertencente a Sara Gomes?[S/N]
>N
Exame não apagado
>AE
Qual o código da leitura?
>1234
Apagar Leitura 1234 de 22/4/2015 pertencente a Sara Gomes?[S/N]
>S
Leitura 1234 de 22/4/2015 apagado do sistema
>AL
Qual o código da leitura?
>1234
Leitura inexistente no Sistema
```

## 5.6. (S) Comando AP - Apagar Paciente

*Apagar do sistema determinado paciente. Isto implica apagar toda a informação relacionada (Leituras dos Sensores, etc ...).*

```
>AP
Qual o nome do paciente a apagar?
>Sara Gomes
Apagar paciente Sara Gomes?[S/N]
>N
Paciente não apagado
>AP
Qual o nome do Paciente a apagar?
>Sara Gomes
Apagar a Sara Gomes?[S/N]
>S
Paciente apagado do sistema
```

## 5.7. (S) Comando NP - Números de pacientes

```
>NP
O sistema tem 2 pacientes introduzidos e um total de 5 leituras
```

## 5.8. (S) LP - Listar pacientes

```
>LP
Pacientes no sistema
-----
Id:33234
Nome:Sara Gamas
Idade:35
Sexo:F
Leitura(s): 2

Id:11111
Nome:Mara Canas
Idade:35
Sexo:F
Leitura(s): 5
-----
```

### 5.9. (S) LE - Listar Leituras feitas a um determinado paciente

>LE

Qual o Paciente?

> Sara Gomas

Paciente inexistente

>LE

Qual o Paciente?

>Sara Gomas

-----  
Paciente

(ID, Nome, Sexo, Idade, Fuma, Medicamentos, Leituras)

1234 | Sara Gomas | F | 33 | S | N | 2

-----  
Exames:

(Data, Código, Indica Problemas)

22/4/2015 3332 Saudável

2/3/2015 1111 Doente

Total de leitura(s): 2  
-----

### 5.10. (L) LS - Listar pacientes saudáveis

Se o paciente não tem leituras não será considerado para esta listagem. O paciente apenas é considerado saudável se todos os canais indicarem saudável. Basta um indicar patológico para ser considerado patológico.

>LS

-----  
(ID, Nome, Sexo, Idade, Fuma, Medicamentos, Leituras)

1234 | Sara Gomas | F | 33 | S | N | 4

5555 | Pedro Gamito | M | 40 | N | N | 2

Total saudáveis(s): 2  
-----

### 5.11. (L) LD - Listar pacientes com patologias

Se o paciente não tem Leituras não será considerado para esta listagem. Este comando lista os pacientes que apresentam pelo menos um canal indicador de patologia (será decisão do técnico validar as patologias a apresentar no relatório ao médico antes do diagnóstico).

>LD

-----  
(ID, Nome, Sexo, Idade, Fuma, Medicamentos  
Códigos dos Leituras com indicadores de problemas)

-----  
3434 | Paulo Tareco | M | 50 | S | N  
432 222 335

5555 | Filipa Pomodoro | F | 30 | N | N  
234

Total doentes(s): 2  
-----

## 5.12. (L) REA - produzir um relatório do histórico do paciente por datas

Este comando permite ao técnico introduzir um sumário de opinião técnica e gerar um relatório para ser enviado ao médico do paciente. Este relatório será armazenado num ficheiro chamado *relatorio.txt*.

```
>REA
>Qual o nome do paciente?
>Paulo Tareco
Exames:
(Data, Código, Indica Problemas )
(1) 22/4/2015 3332 Saudável
    canais medidos : ECG, EOG, Respiratorio
(2) 2/3/2015 1111 Detectada patologia
    canais medidos: ECG, EOG

Introduza sumário técnico (uma frase):
> O paciente revela progresso tendo em conta as leituras nos exames
anteriores, contudo alguns indicadores não são conclusivos,
sugere-se uma contra prova de ECG e EOG.
Qual o código da sua cédula profissional?
> 3444553
Qual o seu nome?
> Alberto Bugarin
Relatório produzido
```

Ficheiro produzido na sequência das instruções anteriores **relatorio.txt**:

```
Nome Paciente: Paulo Tareco
Idade: 43
-----
Nome Técnico:Alberto Bugarin
Cédula Profissional: 3444553

Relatório base:
O paciente revela progresso tendo em conta as leituras nos exames
anteriores, contudo alguns indicadores não são conclusivos,
sugere-se uma contra prova de ECG e EOG.
```

-----  
Exames:

(Data, Código, Indica Problemas )

22/4/2015 3332 Saudável

canais medidos : ECG, EOG, Respiratorio

2/3/2015 1111 Detectada patologia

canais medidos: ECG, EOG

### 5.13. (L) RES - produzir um relatório simples a um exame do paciente

>RES

>Qual o código do exame?

>3332

(Data, Código, Indica Problemas )

22/4/2015 3332 Saudável

Sensores: ECG EOG Respiratório Acelerómetro

Disponibilizar resultados das leituras dos sensores eléctricos?[S/N]

>S

Disponibilizar resultados das leituras dos sensores não eléctricos?[S/N]

>S

| ID            | 1234 | Idade | 55    | Nome         | Alberto Caminha | Sexo | M |
|---------------|------|-------|-------|--------------|-----------------|------|---|
| Time \ Sensor | ECG  | EOG   | EMG   | Respiratório | Acelerómetro    |      |   |
| 0             | 0.00 | 19.83 | 53.96 | 1            | 2               |      |   |
| 1             | 0.00 | 30.36 | 54.19 | 1            | 7               |      |   |
| 2             | 0.00 | 28.69 | 53.06 | 1            | 6               |      |   |
| 3             | 0.00 | 15.47 | 52.59 | 1            | 4               |      |   |
| 4             | 0.00 | 33.22 | 51.87 | 1            | 1               |      |   |
| 5             | 0.00 | 31.90 | 52.96 | 1            | 0               |      |   |
| 6             | 0.00 | 16.41 | 51.34 | 1            | 10              |      |   |
| 7             | 0.00 | 20.88 | 53.51 | 1            | 8               |      |   |
| 8             | 0.00 | 23.96 | 54.34 | 1            | 6               |      |   |
| 9             | 0.00 | 27.20 | 52.32 | 1            | 16              |      |   |
| 10            | 0.00 | 15.58 | 51.31 | 1            | 13              |      |   |
| 11            | 0.00 | 15.42 | 54.85 | 0            | 20              |      |   |
| 12            | 0.00 | 30.83 | 55.00 | 0            | 25              |      |   |
| 13            | 0.00 | 28.78 | 53.60 | 0            | 24              |      |   |
| 14            | 0.00 | 17.65 | 51.95 | 0            | 14              |      |   |
| 15            | 0.00 | 33.20 | 54.44 | 0            | 16              |      |   |

|     |        |       |       |   |     |
|-----|--------|-------|-------|---|-----|
| 16  | 0.00   | 22.34 | 51.26 | 0 | 29  |
| 17  | 0.00   | 32.03 | 52.71 | 0 | 40  |
| 18  | 0.00   | 18.00 | 51.95 | 0 | 46  |
| 19  | 0.00   | 17.80 | 50.49 | 0 | 42  |
| 20  | 0.00   | 31.85 | 50.72 | 0 | 37  |
| 21  | 0.00   | 26.05 | 53.42 | 0 | 32  |
| 22  | 0.00   | 24.40 | 54.82 | 1 | 0   |
| 23  | 0.00   | 33.36 | 51.62 | 1 | 0   |
| 24  | 0.00   | 34.16 | 53.66 | 1 | 0   |
| 25  | 0.00   | 23.42 | 52.96 | 1 | 0   |
| 26  | 0.00   | 24.40 | 53.37 | 1 | 1   |
| 27  | 0.00   | 27.19 | 51.30 | 1 | 7   |
| ... |        |       |       |   |     |
| 118 | 314.27 | 34.84 | 54.16 | - | -85 |
| 119 | 345.83 | 16.29 | 53.26 | - | -90 |
| 120 | 372.68 | 32.59 | 53.50 | - | -95 |

*Introduza sumário técnico (uma frase):*

> Os valores obtidos não revelam patologias.

*Qual o código da sua cédula profissional?*

> 3444553

*Qual o seu nome?*

> Alberto Bugarin

*Relatório produzido*

Ficheiro produzido na sequência das instruções anteriores **relatorio.txt**:

**Exame:** 3332

*Data:* 22/4/2015

*Nome Paciente:* Paulo Tareco

*Idade:* 43

-----  
*Nome Técnico:* Alberto Bugarin

*Cédula Profissional:* 3444553

*Relatório base:*

*Saudável (ausência de sintomas de patologia)*

*Resumo Técnico:*

Os valores obtidos não revelam patologias.

-----  
*Detalhes*

| <i>ID</i>            | <i>1234</i> | <i>Idade</i> | <i>55</i>  | <i>Nome</i>         | <i>Alberto Caminha</i> | <i>Sexo</i> | <i>M</i> |
|----------------------|-------------|--------------|------------|---------------------|------------------------|-------------|----------|
| <i>Time \ Sensor</i> | <i>ECG</i>  | <i>EOG</i>   | <i>EMG</i> | <i>Respiratório</i> | <i>Acelerómetro</i>    |             |          |
| 0                    | 0.00        | 19.83        | 53.96      | 1                   | 2                      |             |          |
| 1                    | 0.00        | 30.36        | 54.19      | 1                   | 7                      |             |          |
| 2                    | 0.00        | 28.69        | 53.06      | 1                   | 6                      |             |          |
| 3                    | 0.00        | 15.47        | 52.59      | 1                   | 4                      |             |          |
| 4                    | 0.00        | 33.22        | 51.87      | 1                   | 1                      |             |          |
| 5                    | 0.00        | 31.90        | 52.96      | 1                   | 0                      |             |          |
| 6                    | 0.00        | 16.41        | 51.34      | 1                   | 10                     |             |          |
| 7                    | 0.00        | 20.88        | 53.51      | 1                   | 8                      |             |          |
| 8                    | 0.00        | 23.96        | 54.34      | 1                   | 6                      |             |          |
| 9                    | 0.00        | 27.20        | 52.32      | 1                   | 16                     |             |          |
| 10                   | 0.00        | 15.58        | 51.31      | 1                   | 13                     |             |          |
| 11                   | 0.00        | 15.42        | 54.85      | 0                   | 20                     |             |          |
| 12                   | 0.00        | 30.83        | 55.00      | 0                   | 25                     |             |          |
| 13                   | 0.00        | 28.78        | 53.60      | 0                   | 24                     |             |          |
| 14                   | 0.00        | 17.65        | 51.95      | 0                   | 14                     |             |          |
| 15                   | 0.00        | 33.20        | 54.44      | 0                   | 16                     |             |          |
| 16                   | 0.00        | 22.34        | 51.26      | 0                   | 29                     |             |          |
| 17                   | 0.00        | 32.03        | 52.71      | 0                   | 40                     |             |          |
| 18                   | 0.00        | 18.00        | 51.95      | 0                   | 46                     |             |          |
| 19                   | 0.00        | 17.80        | 50.49      | 0                   | 42                     |             |          |
| 20                   | 0.00        | 31.85        | 50.72      | 0                   | 37                     |             |          |
| 21                   | 0.00        | 26.05        | 53.42      | 0                   | 32                     |             |          |



|     |        |       |       |   |     |
|-----|--------|-------|-------|---|-----|
| 22  | 0.00   | 24.40 | 54.82 | 1 | 0   |
| 23  | 0.00   | 33.36 | 51.62 | 1 | 0   |
| 24  | 0.00   | 34.16 | 53.66 | 1 | 0   |
| 25  | 0.00   | 23.42 | 52.96 | 1 | 0   |
| 26  | 0.00   | 24.40 | 53.37 | 1 | 1   |
| 27  | 0.00   | 27.19 | 51.30 | 1 | 7   |
| ... |        |       |       |   |     |
| 118 | 314.27 | 34.84 | 54.16 | 0 | -85 |
| 119 | 345.83 | 16.29 | 53.26 | 0 | -90 |
| 120 | 372.68 | 32.59 | 53.50 | 0 | -95 |

**5.13. (XXL) G - Gravar**

Este comando consiste em registar todos os objectos de sistema (relativos à informação das leituras e dos pacientes) num ficheiro denominado lab.db sob a forma de ByteStream. Caso este ficheiro esteja disponível na pasta do projecto quando a aplicação arranca, então deverá ser carregado todo o seu conteúdo para a memória.

## 6. Materiais a entregar

Deve ser submetido na área moodle associada à disciplina um único arquivo ZIP (são também aceites RAR e 7Z), contendo o projeto Eclipse completo. Este deve incluir no arquivo a pasta raiz do projeto. Sugere-se que, antes de gerar o arquivo, execute a opção 'Project' > 'Clean...'

Será criado na área moodle da disciplina uma ligação para efectuar a submissão dos trabalhos. Para cada grupo, é solicitada apenas uma submissão. **Notem que podem submeter mais do que uma vez: nesse caso, cada nova versão sobrepõe-se à anteriores.**

O nome do arquivo a submeter deve iniciar-se com P<nº do turno>, seguido do números dos elementos do grupo, e.g., P5\_NAluno1\_NAluno2. Por exemplo, dois alunos do turno teórico-prático P2, com os números 12345 e 13456 deverão entregar no moodle um zip com o nome P2\_12345\_13456.

O trabalho está dimensionado para ser realizado por grupos de 2 alunos, não sendo admitidos grupos com mais que 2 alunos. Os alunos realizarão defesas dos seus trabalho, tendo ambos os elementos do grupo de estar presentes. A defesa é um elemento fundamental na avaliação do trabalho prático – entre outras coisas, permite aos docentes fornecer retroacção personalizada de cada trabalho.

## **7. Método de desenvolvimento do projeto**

É muito importante ser metódico em qualquer atividade com alguma complexidade e duração. Por isso, esta secção fornece algumas recomendações sobre uma metodologia de trabalho adequada para um projeto de desenvolvimento de um sistema de software. Desenvolver qualquer programa não trivial acarreta sempre uma sequência de atividades a levar a cabo de forma disciplinada, para se obter um “produto” final de qualidade, facilitando a legibilidade do programa, minorando a ocorrência de defeitos (bugs) e facilitando a sua detecção quando ocorrem, e ... ter uma boa nota.

### **a. Fase 1 - Compreensão e Esclarecimento do enunciado e dos objectivos do projeto**

Leia bem o enunciado, anotando todas as questões que lhe ocorram – de preferência por escrito e em papel (usar um lápis funciona bem com muita gente). Mantenha notas com as dúvidas sobre o enunciado e com as respostas que for obtendo. É útil saber, em cada momento, que dúvidas tem, e quais as respostas às dúvidas que teve, mas já esclareceu.

Esclarecer as suas dúvidas junto dos docentes da disciplina. Aproveite os horários de dúvidas e aulas práticas. Nesta fase eliminará as maiores dúvidas sobre o enunciado e sobre o que se pretende, mas claro que vão persistir dúvidas e surgir novas dúvidas à medida que for desenvolvendo o sistema. Por isso, prepare-se para manter o seu documento de dúvidas até ao fim do prazo de entrega.

### **b. Fase 2 - Análise da estrutura geral do sistema**

Após a primeira fase, o seu objectivo será definir a estrutura global do seu programa, ou seja as classes e relações entre elas. Um programa em Java consiste num conjunto de classes Java, cujos objectos representam as várias entidades do domínio do problema, e talvez uma ou outra classe do domínio da solução. Inclui também, necessariamente, uma classe especial – Main – que contém o “programa principal” (método main), e diversos métodos auxiliares (por exemplo, um por cada opção do interpretador de comandos, bem como métodos gestão do input / output).

Os objectos dessas classes representarão os conceitos em atividade durante a execução do programa. Manterão a informação relativa a cada instância desse conceito, e fornecerão grupos de operações apropriadas.

Numa primeira fase de análise não terá que programar, mas apenas que pensar e definir, para cada classe, a interface (conjunto de operações), assim como os constructores, as variáveis de instância e constantes que achar necessárias na classe. Certifique-se de que as operações que identificou nas interfaces devem mesmo pertencer à classe em que as colocou. Identifique também as variáveis e constantes que achar necessárias para representar o estado de cada objecto de cada classe, assim como o seu tipo. Não se

esqueça de que, frequentemente, a melhor abordagem é usar objectos de uma classe como valores de variáveis de outros objectos.

Quando tiver dúvidas sobre onde colocar uma determinada operação (em que classe), consulte os docentes da disciplina. Para que tudo fique bem documentado, para cada variável indique o seu tipo e explique a finalidade. Para cada método indique o tipo do seu resultado e o tipo dos seus parâmetros (se existirem). Explique ainda a sua finalidade (para que serve) de forma clara e intuitiva. Indique ainda que métodos são modificadores e que métodos são de consulta.

Para além das classes mais “óbvias”, provavelmente precisará de classes auxiliares, para programar, tais como colecções de objectos de certo tipo. Estas são exemplos de classes do domínio da solução. Outro bom exemplo são as classes dos iteradores. Cada classe que definir para representar colecções deve ter um conjunto de operações associadas que inclui em geral uma operação para inserir um elemento da colecção, pesquisar um elemento na colecção (dado o estado clínico, por exemplo), etc., etc. Pense bem que colecções precisa para programar o sistema. Cada tipo de colecção será representado por uma classe distinta.

Uma coisa que pode construir logo no início do desenvolvimento é o interpretador de comandos, a colocar na classe Main. Defina logo um método vazio (ou que envie uma mensagem esclarecedora para a consola) para cada comando especificado no enunciado (excepto Sair, claro). Lembre-se de que, como a Main não é usada por qualquer outra classe, os seus membros deve ser estáticos e privados. A Main é algo que fica feito de uma vez por todas e ajuda-o a ter uma primeira visão global do sistema. Depois, poderá preencher esses métodos à medida que vai desenvolvendo as diversas funcionalidades. O objectivo desta fase é um obter um programa que aceita os comandos e os dados associados, apesar de não fazer nada (excepto sair, com o comando Sair). Se adoptar esta abordagem, vai poder começar a testar as várias funcionalidades usando a linha de comando, à medida que as vai implementando.

No final desta fase, mostre o resultado da sua planificação aos docentes da disciplina. Estes poderão dar alguns conselhos, ou transmitir-lhe confiança sobre a adequação da sua proposta. Depois desta fase estar concluída estará bem preparado para programar.

### **c. Fase 3 - Construção do sistema**

Não tente fazer tudo ao mesmo tempo – sobretudo sem testar o que já fez! Divida o problema em “fatias” e pergunte-se sempre “qual é a próxima coisa que eu devo fazer”? Depois do interpretador de comandos, provavelmente será boa ideia continuar com as classes do domínio – sobretudo as que estarão associadas às interfaces especificadas no enunciado. Estas tenderão a ser classes dos elementos de colecções e é boa ideia começar primeiro pelos elementos antes das colecções que os usam. Ou seja, empregar uma abordagem bottom-up.

## d. Fase 4 - Recomendações gerais

É muito importante seguir a regra da versão estável: mantenha sempre uma versão do programa funcional, que compile sem erros e faz qualquer coisa, mesmo que ainda não tudo. Nunca comece a trabalhar em novas fases, sem que as anteriores estejam robustas, bem testadas. Assim terá mais segurança e uma base sólida para prosseguir pelas diversas fases.

Nunca esteja muito tempo sem que o seu programa compile sem erros, ou não faça o que era previsto fazer nessa tarefa! As versões estáveis são uma rede de segurança a que se pode agarrar. Nunca apague o código de cada versão estável, para poder voltar a essa se versão, se se perder no desenvolvimento ou se começar a ficar com um programa muito confuso. Ponha em prática uma política de salvaguardas.

Programa em grupo, com o seu parceiro, sentados à volta do mesmo computador e debatendo opções e aspectos da solução. Não tente dividir o trabalho em partes muito separadas: Não é boa ideia quando ainda não se tem muita experiência. As coisas depois não “colam”, e “martelar” peças para encaixarem à força também não costuma ser boa ideia.

Não escreva quilómetros de código sem testar! Teste as suas classes método a método. Se for preciso crie em cada classe pequenos métodos main para testar os métodos dessa classe.

Use bons nomes para os identificadores. Siga as regras de estilo recomendadas nas aulas. Cada método não deve ter mais que 15 linhas de código, e na média deve ter cerca de 10 ou menos. Defina métodos auxiliares (privados) sempre que tal for aconselhável para tornar o seu programa legível. A classe Main é a única que pode utilizar operações de input / output de ou para o ecrã ou de e para ficheiros (println, etc). Nenhum método de uma classe auxiliar pode conter operações de input / output. Quando chegar a altura, converse com os docentes sobre como ler e escrever a informação necessária para o ficheiro.

Todas as variáveis de estado dos objectos TÊM que ser privadas (private). Isto inclui as variáveis que sejam de tipo vector. Para os objectos que contêm colecções de outros objectos (guardados em vector), crie métodos de consulta e modificadores dos elementos dos vectores que sejam necessários. Nunca permita o acesso externo directo a um vector que seja criado dentro de um objecto. Se necessitar de listar o conteúdo de uma colecção, use SEMPRE um iterador.

Lembre-se de que os docentes vão classificar o seu trabalho com base na qualidade do programa: é por isso que estilo da programação é mesmo muito importante.

Não escreva código que não percebe como funciona. Se você não percebe como o SEU programa funciona, muito menos os docentes o vão perceber quando o forem avaliar! Em caso de dúvida, estude, consulte livros, os docentes, os colegas, whatever. Mas não programe por tentativa e erro: isso não vai levar a lado nenhum, senão ao chumbo. Vai ver que à medida que vai ganhando prática, vai cometendo menos erros e percebendo melhor

o que está a acontecer. Comente o seu código sempre que lhe parece que este não é claro.

Quando o compilador ou o seu programa der um erro, tente perceber bem porque o erro aconteceu, para não repetir o mesmo erro muitas vezes.

**Esta lista de indicações pode ainda crescer um pouco, veja as novas versões que vão saindo, pois podem conter informações úteis.**