

Programação Orientada por Objetos 2020/2021

---

Enunciado do Projeto 2

**“Base de dados hospitalar  
de doenças infectocontagiosas”**



## 1. Descrição

Neste projeto pretende-se que os alunos desenvolvam uma aplicação que permite gerir a base de dados de uma unidade hospitalar de doenças infectocontagiosas, permitindo a inserção e consulta de informação relevante.

A aplicação deve ser capaz de gerir os pacientes que dão entrada no hospital. Cada pessoa que se tenha tornado paciente ao dar entrada no hospital é caracterizada pelo ano de nascimento e pelo id (ID0001, ID0002, ID0003, ...). De forma a não criar entropia no normal funcionamento da unidade, é importante garantir que o id é único (até porque pode acontecer um médico ou enfermeiro dar entrada no hospital como paciente). Em cada hospital participam vários médicos e enfermeiros.

Num hospital, existem enfermeiros que podem ser enfermeiro-especialista ou enfermeiro-auxiliar. Os enfermeiros são caracterizados por um nome e pelos anos de carreira. Qualquer enfermeiro-especialista independentemente da especialidade pode ser promovido a enfermeiro-chefe pela administração, desde que cumpra os anos de carreira mínimos definidos pelo próprio hospital. Desta forma, periodicamente são atualizados os anos de carreira de todos os enfermeiros do hospital de forma incremental. Além disso, cada enfermeiro-especialista pode atender um máximo de 3 pacientes.

No hospital, os médicos tratam diferentes doenças infectocontagiosas (COVID-19, HIV, ebola, ...). Os médicos podem estar associados a um máximo de 3 pacientes. O enfermeiro-chefe, é responsável por destacar os enfermeiros-especialistas para um médico. Cada médico tem de ser acompanhado por, pelo menos, um enfermeiro-especialista em cada acto médico. E para cada médico acompanham tantos auxiliares quantos os necessários. É importante o médico comunicar a um dos enfermeiros-chefe quantos auxiliares precisa de forma a que este possa atribuí-los da *pool* de auxiliares disponível no hospital de forma aleatória. Se não for possível atribuir os enfermeiros-auxiliares que precisa, o médico tem de denunciar imediatamente a situação ao próprio hospital e efetuar um pedido de mais auxiliares por meio de uma *Exception*.

A administração do hospital de tempos a tempos atende aos pedidos de mais auxiliares, dando prioridade aos últimos pedidos a chegar. Periodicamente, e com receio de uma auditoria externa, alguns dos pedidos vão para a trituradora de forma aleatória.

O hospital encontra-se em contexto pandémico pelo que é importante gerar relatórios de forma semi-automática com métricas relativamente às patologias típicas de emergirem em contexto hospitalar, nomeadamente o COVID-19. Alguns exemplos como o número de casos positivos por patologia, número de óbitos de pacientes ou o rácio de pacientes testados positivamente pelo total de pacientes testados, entre muitos outros, são considerados aceitáveis. O hospital deverá manter um histórico (i.e. estruturas de dados) de forma a suportar estas métricas quando lhe forem solicitadas.

Um médico ao fazer o diagnóstico ao seu paciente, vai verificar se, no caso da COVID-19, há um valor elevado da temperatura. No caso da ebola, se tem sintomas gastrointestinais ou para o HIV se o nível de glóbulos brancos é baixo. Consoante o resultado desse diagnóstico, encarregará os enfermeiros que lhe estejam alocados (e para cada doença cujo diagnóstico

seja positivo) a respectiva aplicação do curativo. Os enfermeiros manterão uma agenda interna de pacientes a quem deverá ser administrado o curativo. Assim que seja aplicado o curativo, o paciente deverá ser encaminhado de volta para o médico de forma a que este lhe dê alta.

No entanto, o curativo poderá não ter tido o efeito pretendido, i.e. o médico procede novamente ao diagnóstico no momento de dar alta de forma a validar se o paciente continua com sintomas positivos das várias doenças. Deverá ser aplicado o curativo ao paciente (via enfermeiro) até que este tenha alta ou venha a contar como óbito.

### 1.1 Menus

Embora haja diferentes *roles* na aplicação (i.e. médicos, enfermeiros e administradores), para simplificar podemos assumir que os dados serão introduzidos numa única linha de comandos com diferentes menus, um por cada *role*. Assim, ao seleccionar certas opções dos Menus médico ou enfermeiro, poderá ser necessário fornecer como input o médico que, por exemplo, está a dar alta a um dado paciente ou o enfermeiro que está a aplicar o curativo a uma pessoa.

Descreve-se em seguida cada menu (e submenus) e o que cada opção deverá fazer:

#### 1. Menu médico:

- **“Listar pacientes em espera no hospital”**;
- **“Listar pacientes a aguardar alta”**: listagem dos pacientes que aguardam alta por parte de um médico após o enfermeiro aplicar o curativo;
- **“Diagnóstico ao paciente”**: Deverá ser seleccionada uma pessoa que deu entrada no hospital. Será feito o diagnóstico, e em caso(s) positivo(s) de patologias, deverá ser colocado na agenda dos enfermeiros alocados ao médico a aplicação do curativo;
- **“Dar alta hospitalar”**: Após um enfermeiro aplicar o curativo ao um paciente, este é devolvido ao médico de forma a este averiguar (leia-se diagnosticar) se está em condições de ser dada alta. Se sim é “devolvido” ao hospital para efeitos de histórico. Caso contrário são aplicados curativos até ser declarado o óbito (e “devolvido” ao hospital para efeitos de histórico);
- **“Requerimento de auxiliares”**: um médico comunica a um enfermeiro-chefe quantos auxiliares precisa para que este os coloque imediatamente ao dispor. Conforme referido, é importante levantar uma denúncia com pedido de mais auxiliares se não for possível proceder à operação com sucesso;

#### 2. Menu Enfermeiro (notar que é importante tratar dos cenários **excepcionais** em que se selecciona erradamente um enfermeiro ou auxiliar para fazer algo que supostamente é o chefe a fazer...):

- **“Listar enfermeiros de médico”**: listagem dos enfermeiros auxiliares e especialistas atribuídos a um dado médico;
- **“Listar pacientes a aguardar curativo”**: listagem dos pacientes a aguardar curativo para um dado enfermeiro;
- **“Atribuir enfermeiro-especialista a médico”**;
- **“Aplicar curativo a paciente”**: assume que o enfermeiro mantém uma “agenda” de pacientes que aguardam o curativo. Esta acção faz com que o enfermeiro aplique o curativo e devolva o paciente ao médico;

### 3. Menu Administrador:

- **“Criar médico”**;
- **“Criar enfermeiro-especialista”**;
- **“Criar enfermeiro-auxiliar”**;
- **“Criar novo paciente”**: deverá ser facultada a informação para uma nova pessoa;
- **“Promover enfermeiro a chefe”**;
- **“Aumentar anos de carreira de todos os enfermeiros”**;
- **“Listar enfermeiros”**: Listar os enfermeiros, mostrando a sua posição hierárquica (enfermeiro-chefe, enfermeiro-especialista, enfermeiro-auxiliar), anos de experiência e o médico ao qual estão alocado;
- **“Listar médicos”**;
- **“Listar pedidos para enfermeiros-auxiliares”**: na sequência de um requerimento falhado de enfermeiros auxiliares por parte de um médico, é gerado um pedido para mais enfermeiros-auxiliares. Esta funcionalidade lista-os;
- **“Listar pacientes em espera no hospital”**: equivalente ao do Menu médico;
- **“Atirar pedidos para enfermeiros-auxiliares para a trituradora”**: como referido, deverá ser descartado um número aleatório de pedidos para enfermeiros-auxiliares;
- **“Atende ao pedido para enfermeiros-auxiliares”**: o hospital coloca ao dispor um número razoável (dois ou três) de novos enfermeiros-auxiliares ao dispor dos médicos do hospital;
- **“Virus outbreak”**: deverão ser colocadas um conjunto aleatório de pessoas (incluindo médicos, enfermeiros, ex-pacientes, etc) como pacientes no hospital com patologias aleatórias;
- **“N-ésimo relatório hospitalar”**: As restantes entradas neste menu estão reservadas aos relatórios hospitalares. Deste modo, cada entrada deverá emitir um relatório que permita obter uma perspectiva interessante para o hospital em causa. São dados alguns exemplos anteriormente, mas é expectável que os alunos desenvolvam novos relatórios na medida das suas possibilidades de esforço e tempo de forma a valorizarem o trabalho;
- **“Sair da aplicação”**;

Deverá ser dada a possibilidade de inserir a informação (pacientes, médicos, enfermeiros, ...) de duas formas: através do teclado e através de um ficheiro de texto. No segundo caso, deverão criar um ficheiro com os dados do hospital que deverá ser entregue com o trabalho.

Os alunos deverão ponderar qual a informação a fornecer em cada uma das operações e, em cada uma delas, quais as situações que podem levar a situações de erro. Por exemplo, no caso da atribuição de um médico a um dado paciente, é necessário fornecer o ID do médico. A operação falha se não existir um médico com esse ID. Tal como num contexto empresarial, os requisitos podem ser inconsistentes, omissos ou vagos (p.e. relativamente às métricas dos relatórios hospitalares a apresentar), apela-se ao engenho dos alunos quanto à solução adotada do ponto de vista funcional, informativo e principalmente dos conceitos de Programação Orientado a Objetos de forma a valorizarem o seu trabalho.

## 2. Elementos de Avaliação do Projeto

### 2.1. Relatório e Código

O projeto deverá ser realizado pelos grupos previamente definidos e deverão ser entregues um relatório e código do sistema implementado em JAVA com as funcionalidades descritas anteriormente.

**Relatório:** No relatório devem descrever detalhadamente a abordagem e implementação do trabalho realizado. Deverão também incluir um diagrama de classes UML que descreve o modelo do projeto realizado. O diagrama deve preferencialmente ser realizado recorrendo a uma ferramenta de modelação. Finalmente, o relatório deverá incluir uma conclusão com uma apreciação geral do trabalho desenvolvido, indicando se conseguiram atingir os objetivos pretendidos. Evitem um relatório extenso, com repetição de partes do enunciado e erros ortográficos. **O relatório deve incluir o código do projeto em anexo como texto, NÃO como figura. Sem este requisito a entrega não será considerada válida.**

**Código:** Deverão entregar a pasta do projeto realizado no IDE Netbeans Apache. O código deverá estar devidamente estruturado e comentado para transmitir de forma breve o objetivo de cada componente.

**O relatório, anexo e o projeto devem ser entregues via Moodle. O relatório e anexo deverão também ser enviados para o Gabinete de Apoio ao Estudante.**

### 2.2. Defesa do Projeto

Após a realização do projeto, os alunos deverão fazer a defesa do mesmo. A defesa será **individual** pelo que os elementos do grupo devem dominar a totalidade do trabalho realizado.

As datas de entrega referentes a cada etapa estão indicadas na tabela que se segue.

Milestones	
Data	Entrega
Até 22 de Janeiro de 2021	Entrega do código e relatório descrevendo o trabalho desenvolvido (via Moodle + GAE).
25 a 29 de janeiro de 2021	Defesa individual do projeto desenvolvido.

## 3. Considerações Gerais

Durante a sua implementação, o problema deve ser abordado de forma gradual:

- Desenvolvam o sistema incrementalmente e vão testando as diferentes classes à medida que vão implementando.
- Construam o esqueleto da classe Main que trata da entrada e saída dos dados e da interação com a aplicação.

- Vão fazendo, testando, e avançando por incrementos, à medida que as funcionalidades vão sendo desenvolvidas. Se necessário, criem pequenos programas de teste auxiliares.
- É preferível apresentarem um projeto que funciona mas que não cumpre todas as funcionalidades do que apresentar um projeto que supostamente implementa tudo mas não funciona.
- Os pormenores de interface com o utilizador não são considerados elementos de avaliação para o presente trabalho.

#### **4. Avaliação**

O projeto tem a nota mínima de **8 valores**.

#### **5. Código de Honestidade Académica**

Espera-se que os alunos conheçam e respeitem o Código de Conduta da Universidade da Madeira (Capítulo III, 15.2) que rege esta disciplina e que pode ser consultado na página da UMa.