

Relatório de trabalho prático

Hospital de Campanha do IPCA

João Gomes

Aluno nº 12190

Trabalho realizado sob a orientação de:

Luís Ferreira

**Linguagens de Programação II**

**Licenciatura em Engenharia de Sistemas Informáticos**

Barcelos, 02 de Junho de 2020

**Índice**

[Introdução: 5](#_Toc42088548)

[Identificação do tema: 5](#_Toc42088549)

[Estruturação do Trabalho: 5](#_Toc42088550)

[Estrutura: 7](#_Toc42088551)

[I. Camada BO (Business Object Layer) 7](#_Toc42088552)

[II. Camada DL (Data Layer) 8](#_Toc42088553)

[III. Camada BL (Business Layer) 9](#_Toc42088554)

[IV. Camada “Excepcoes” 10](#_Toc42088555)

[V. Presentation Layer (Main-TP-12190) 11](#_Toc42088556)

[Fase 2: 13](#_Toc42088557)

[Conclusão: 13](#_Toc42088558)

**Lista de Figuras**

[Figura 1- Diagrama de classes (BO) 7](#_Toc42088648)

[Figura 2- Diagrama de classes (DL) 8](#_Toc42088649)

[Figura 3- Diagrama de Classes (BL) 9](file:///C:\Users\JPC\Google%20Drive\Licenciatura\LP2\Ano%202\12190_LP2\Relatorio_a12190_LP2.docx#_Toc42088650)

[Figura 4- Diagrama de Classes (Exceções) 10](file:///C:\Users\JPC\Google%20Drive\Licenciatura\LP2\Ano%202\12190_LP2\Relatorio_a12190_LP2.docx#_Toc42088651)

[Figura 5- Diagrama de Classes (Main) 11](file:///C:\Users\JPC\Google%20Drive\Licenciatura\LP2\Ano%202\12190_LP2\Relatorio_a12190_LP2.docx#_Toc42088652)

# **Introdução:**

O presente relatório serve de apresentação das aptidões adquiridas durante o contexto “laboral” da cadeira Linguagens e Programação II, dada pelo Professor Luís Ferreira.

No enunciado do trabalho foi-nos proposto a implementação de solução de gestão, o meu tema irá incidir na situação anormal que a nossa humanidade está a passar.

# **Identificação do tema:**

O objetivo é gerir um hospital de campanha criado para acolher doentes menos graves de covid-19. O Hospital terá duas salas com 100 camas cada, ao qual este programa irá gerir a entrada e saídas de doentes, além disso terá a 25 camas descritas como prioritárias, que só estão disponíveis através de reserva.

Portanto, ao longo do projeto, irei implementar métodos de inserção e remoção de dados e consulta dos mesmos com as várias exceções e problemas associados.

# **Estruturação do Trabalho:**

De forma a conseguir resolver a solução escolhida, optei por criar uma agenda que permite gerir a entrada e saídas de doentes.

Desta forma consigo ter uma gestão mais fluida e um controlo superior sobre a gestão ocupacional das salas.

Ao longo do trabalho irei utilizar várias listas e um Array de [365] dias que irá funcionar como se a cada posição do Array corresponde-se a um dia do Ano.

# **Estrutura:**

O trabalho ficou definido da seguinte maneira:

## **Camada BO (Business Object Layer)**

Nesta camada terei todas as classes relativas aos objetos genéricos.

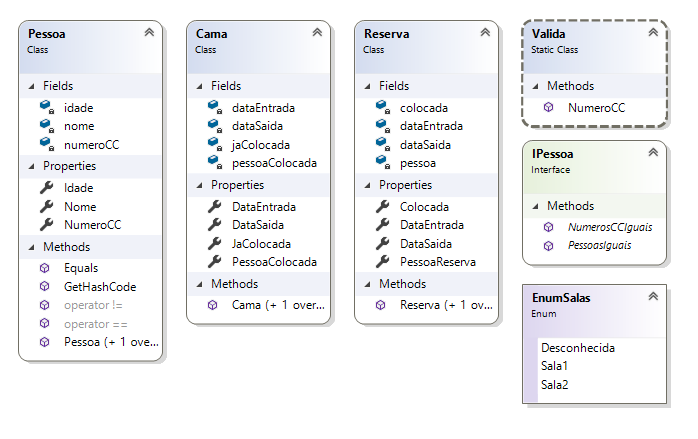


Figura 1- Diagrama de classes (BO)

**Reserva** - Com todos os atributos e propriedades duma Reserva.

**Valida** - Classe de controlo utilizada para validações do número do CC.

**Pessoa** - Classe que contem toda a estrutura para a criação de uma Pessoa.

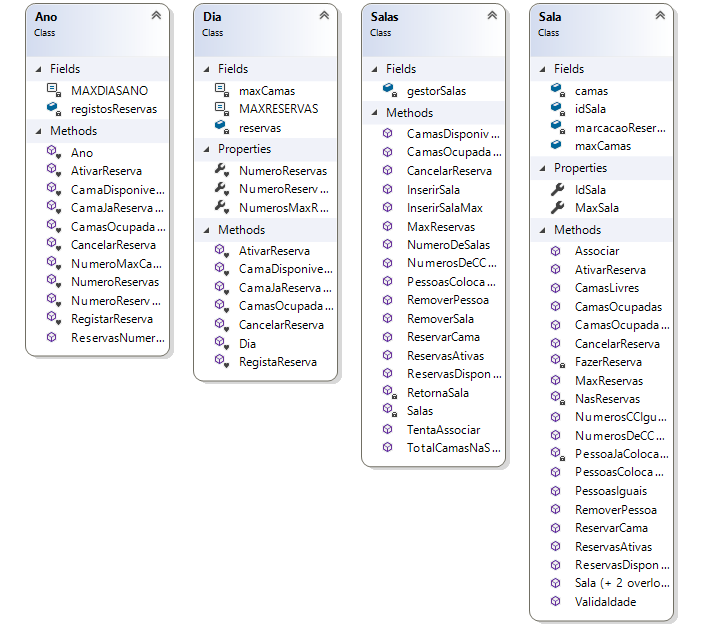
**Cama** - Propriedades de uma cama ocupada.

**IPessoa** - Implementação da Interface.

**EnumSalas** - Enumerador das salas que irei utilizar no PL.

## **Camada DL (Data Layer)**

Nesta camada terei todas as classes relativas à gestão do métodos, os quais, criação de pessoas, remoção, listagem, etc.



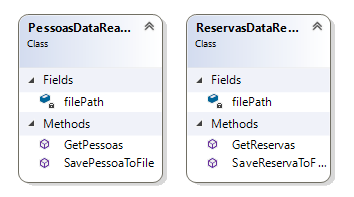


Figura 2- Diagrama de classes (DL)

**Ano** - Constituído por um Array de Dias

**Dia** - Constituído por uma lista de Reservas

**Sala** - constituído por uma lista de camas, mais uma estância de Agenda que contém reservas.

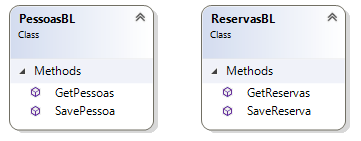
**Docas** - Classe de Modos Estático à qual a BL comunica. Contém uma lista de salas, e respetivos métodos de comunicação.

**PessoasDataReadWritter -** Classe que implementa a gravação e leitura dos dados das pessoas criadas para um ficheiro json.

**ReservasDataReadWritter -** Classe que implementa a gravação e leitura dos dados das reservas criadas para um ficheiro json.

## **Camada BL (Business Layer)**

Esta camada serve de comunicação entre o Main e o DL, será composta por métodos estáticos.



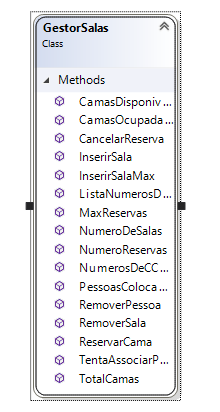


Figura 3- Diagrama de Classes (BL)

## **Camada “Excepcoes”**

Camada responsável por definir todas as exceções customizadas que irão ser utilizadas ao longo do projeto.

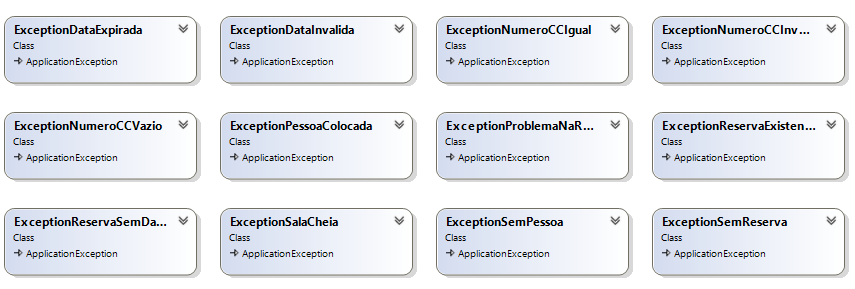


Figura 4- Diagrama de Classes (Exceções)

## **Presentation Layer (Main-TP-12190)**

Camada que será formada pelo I/O visual da aplicação para com o utilizador.

Esta camada utiliza as camadas BO, e BL. Sendo que faz uso das mesmas de forma a poder comunicar com a DL.

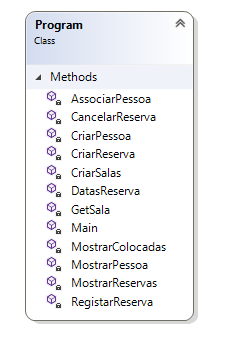


Figura 5- Diagrama de Classes (Main)

# **Fase 2:**

Durante a fase 2, acabei a implementação do DL, criei a camada BL de forma a fazer a comunicação com o Main.

Além disso, criei a camada exceções, de forma a criar as exceções que achei necessárias para melhorar a aplicação.

Implementei quase por completo o Main da aplicação, apenas faltam fazer alguns ajustes e colocar a aplicação a gravar e ler de um ficheiro.

Relativo ao código, quero alterar um pouco o fluxo de como as reservas estão a ser efetuadas.

Por último, corrigir possíveis erros que possam aparecer durante a utilização da aplicação.

# **Conclusão:**

Na fase 3, concluí a implementação das classes, implementei a gravação/leitura do programa para json, corrigi pequenos erros da interface e pequenos ajustes no fluxo do programa.

Sinto que o objetivo do trabalho foi bem conseguido, consegui implementar maioritariamente todo o que tinha previsto. Ao longo do trabalho, senti várias dificuldades, relativas à implementação dos métodos, mas com pesquisa e ajuda dos colegas da turma acabei por conseguir.

Penso que este trabalho foi uma mais-valia para a minha aprendizagem, ajudou muito a perceber a forma de programar em c#.