

Relatório de trabalho prático

Relatório do Trabalho Pratico LP2

Luís Martins

Aluno nº 16980

Daniel maltez

Aluno nº 17015

Trabalho realizado sob a orientação de:

Luís Ferreira

**Linguagens de Programação II**

**Licenciatura em Engenharia de Sistemas Informáticos**

Barcelos, Abril de 2020

Índice

1. **Introdução**

**1.1.** Descrição do Tema a Desenvolver

**1.2.** Camadas

1. **Desenvolvimento**

**2.1.** Diagrama de classes

1. **Conclusão**
2. **Bibliografia**

Lista de Figuras

Figura 1: Diagrama de Classes

Figura 2: Propriedades da classe Pessoa

Figura 3: Enumerador Sexo de uma pessoa

Figura 4: Propriedades da classe Paciente

Figura 5: Enumerador que determina condição de paciente

Figura 6: Propriedades da classe Paciente

Figura 7: Diagrama de Classes da camada Data Layer

Figura 8: Inicialização de listas para armazenar pacientes

Figura 9: Inicialização da lista para armazenar médicos

Figura 10: Diagrama de Classes da camada Business Rule Layer

# Introdução

## 1.1 Descrição do Tema a Desenvolver

Neste trabalho visamos desenvolver uma aplicação que consiga gerir a secção das urgências de um hospital detetando os níveis de urgência na condição de um paciente e atribuindo certos casos a certos médicos.

Também será possível consultar uma lista de espera que organiza os pacientes em termos da condição dos pacientes.

(…)

### 1.2 Camadas

Este projeto usa a arquitetura de camadas NTier, ou seja, o projeto é separado em 4 camadas ou “tiers” estes sendo:

* **Business Objects**: Esta camada contém as classes que definem os objetos de negócio importantes para o projeto. Ex: “Pessoa”, “Paciente”.
* **Data Layer**: Esta camada contém as classes que vão conter as estruturas de dados utilizadas no projeto (Listas), e também os métodos que nos permite manipular os objetos e os atributos dos mesmos nas estruturas de dados.
* **Business Rule Layer**: Esta camada é a que se vai encontrar no meio da camada de dados e a camada Front end, e vai permitir usar os métodos definidos na camada de dados no Main.
* **Main/Front End**: Esta camada é a camada principal que vai chamar os métodos necessários do business rule layer para manipular certos dados.

# Desenvolvimento

### 2.1. Diagrama de Classes

Uma imagem com captura de ecrã

Descrição gerada automaticamente

*Figura 1 – Diagrama de Classes da camada Business Objects*

Esta camada é composta por 3 classes, Pessoa, Paciente e Medico com as duas últimas sendo classes filho da classe Pessoa pois derivam certos dados da mesma.

**Pessoa:**

É constituída pelos atributos:

* Idade
* Nif
* Nome
* Sexo

E contem propriedades que permitem manipular os próprios atributos:

Uma imagem com captura de ecrã, monitor, preto, sentado

Descrição gerada automaticamente

*Figura 2 – Propriedades da classe Pessoa*

Para determinar o sexo de uma pessoa usamos um enumerador para diferenciar entre os estados:

Uma imagem com telefone, telemóvel, preto, rua

Descrição gerada automaticamente

*Figura 3 – Enumerador Sexo de uma pessoa*

**Paciente:**

É constituída pelos atributos:

* Altura
* Condição
* Data de Entrada
* Data de Saida
* Peso

E contem propriedades que permitem manipular os próprios atributos:

Uma imagem com captura de ecrã

Descrição gerada automaticamente

*Figura 4 – Propriedades da classe Paciente*

Semelhante a classe Pessoa, a classe Paciente também usa um enumerador para determinar a condição de um paciente:

Uma imagem com fotografia, preto, sentado, telefone

Descrição gerada automaticamente

*Figura 5 – Enumerador que determina condição de paciente*

**Medico:**

É constituída pelos atributos:

* Especialidade
* Lista de Pacientes

E contem propriedades que permitem manipular os próprios atributos:

Uma imagem com captura de ecrã, monitor, ecrã, preto

Descrição gerada automaticamente

*Figura 6 – Propriedades da classe Paciente*

*Uma imagem com captura de ecrã

Descrição gerada automaticamente*

*Figura 7 – Diagrama de Classes da camada Data Layer*

Esta camada é constituída por 2 classes, Pacientes e Médicos, estas sendo as camadas responsáveis pelas estruturas de dados usadas para armazenar os dados relacionados aos pacientes e médicos.

**Pacientes:**

Esta classe inicializa as listas que armazenam os dados de todos os pacientes que frequentaram o hospital, como também listas diferentes que separam os pacientes dependendo da condição dos mesmos.

Uma imagem com mesa, sentado, preto, portátil

Descrição gerada automaticamente

*Figura 8 –* *Inicialização de listas para armazenar pacientes*

**Métodos:**

* **AddPaciente(Paciente paciente) -** Adiciona paciente à lista de pacientes global, também compara a condição do paciente e adiciona o mesmo na lista dessa condição.
* **ListarTodosPacientes() –**Lista o histórico de pacientes que frequentaram o hospital (Outros métodos parecidos listam os pacientes com certas condições).
* **ObterPosPaciente(string nomPaciente) –** Obtém a posição de um paciente numa lista.
* **RemovePaciente(string nomePaciente) –** Remove um paciente de uma lista dependendo da condição do mesmo.
* **ExistePaciente(string nomePaciente) –** Verifica se existe um paciente numa lista.
* **SavePacientes() –** Guarda dados da lista atual para um ficheiro “.txt”.

**Médicos:**

Esta classe inicializa a lista que armazena os dados de todos os médicos no hospital.

Uma imagem com propriedade, telefone, sentado, mesa

Descrição gerada automaticamente

*Figura 9 – Inicialização da lista para armazenar médicos*

**Métodos:**

* **AddMedico(Medico medico) –** Adiciona um medico à lista de médicos.
* **ObterPosMedico(string nomMedico) –** Obtém a posição de um medico na lista.
* **AtribuirPacientes(string nomeMedico, string nomePaciente) –** Atribui um paciente a um medico na lista.
* **RemoveMedico(string nomeMedico) –** Remove um medico da lista.
* **ListarTodosMedicos() –** Lista todos os médicos guardados na lista.
* **ExisteMedico(string nomeMedico) –** Verifica se um medico existe na lista.
* **SaveMedicos() –** Guarda dados de médicos na lista para um ficheiro “.txt”.

Uma imagem com captura de ecrã

Descrição gerada automaticamente

*Figura 10 – Diagrama de Classes da camada Business Rule Layer*

Esta camada é constituída por apenas uma classe, esta sendo a classe que recebe os métodos todos da camada de dados para serem usado na camada Main.

# Conclusão

Em conclusão, com a realização deste trabalho aumentamos o nosso conhecimento da linguagem C# e também como ela se diferencia de C.

Também como pode ser aplicada em problemas reais.

# Bibliografia

* Código disponibilizado no Moodle
* https://stackoverflow.com/