

# Trabalho prático LPII

Sistema de Gestão de Hospitais

## Trabalho Prático

### Grupo

---

a18832	Ricardo Silva
a15255	Pedro Macedo



Licenciatura em Engenharia de Sistemas Informáticos  
Instituto Politécnico do Cávado e do Ave

2019-2020

# 1 Resumo

Este TP tem como objectivo a continuação do trabalho feito até ao momento com as suas alterações mencionadas no decorrer do relatório para uma possível melhoria.

# Conteúdo

<b>1</b>	<b>Resumo</b>	<b>2</b>
<b>2</b>	<b>Introdução</b>	<b>5</b>
2.1	Contextualização . . . . .	5
2.2	Motivação e objetivos . . . . .	5
2.3	Estrutura do documento . . . . .	6
<b>3</b>	<b>Implementação</b>	<b>7</b>
3.1	Descrição do problema . . . . .	7
3.2	Estruturas de dados . . . . .	8
3.3	Diagrama de classes . . . . .	10
3.4	Solução . . . . .	10
<b>4</b>	<b>Alterações no TP</b>	<b>11</b>
4.1	Doxygen . . . . .	11
4.1.1	Exemplo de execução . . . . .	12
4.1.2	Exemplo de código . . . . .	14
<b>5</b>	<b>Conclusão</b>	<b>15</b>
5.1	Lições aprendidas . . . . .	15
5.2	Apreciação final . . . . .	15

## Lista de Figuras

1	Pessoa . . . . .	8
2	Infeção . . . . .	9
3	Hospital . . . . .	9
4	Diagrama de classes . . . . .	10
5	Doxygen . . . . .	12
6	Opções Expert . . . . .	13
7	Comentários no Código . . . . .	14
8	Resultado Doxygen . . . . .	14

# Capitulo 1

## 2 Introdução

No capitulo introdutório será discutido o contexto do problema seguindo pela motivação e objetivos do projeto e por fim a estrutura do relatório.

### 2.1 Contextualização

Os impactos de grandes doenças nas últimas décadas tem impulsionado o desenvolvimento e a implementação de diversas abordagens para planeamento e resposta à emergência de novos contágios, incluindo sistemas de monitorização de novos casos. Nos últimos meses a humanidade tem vivido situações complicadas causada pela COVID-19, tendo desencadeado uma enorme luta na tentativa de controlar novos casos sendo que se trata de uma doença facilmente transmissível. Informaticamente, várias soluções tem surgido de modo a permitir a análise de dados sobre infecções e controlo de ocorrências num determinado país ou determinada cidade. Assim sendo, com recurso à tecnologia, utilizando uma linguagem multi-paradigma C, conseguimos efectuar a monitorização da situação epidemiológica de uma cidade, contabilizando casos de COVID-19 e também casos de doenças "normais". A utilização de um sistema de gestão deste tipo, podem ser usados para providenciar um enquadramento dos riscos da população.

### 2.2 Motivação e objetivos

O principal propósito deste projecto é o no desenvolvimento de um sistema informático, acessível ao utilizador bastando seguir os passos apresentados na aplicação e que integrasse todos os componentes de ídolo tecnológica para responder às seguintes necessidades:

- A inserção de pacientes com algum tipo de doença ou infeção;
- A inserção de informação relativa a infeções relativas;
- A inserção de informação de hospitais com a sua devida lotação;
- Apresentar o estado do paciente;
- Alterar o estado do paciente de doente para pessoa quando o seu "status" passar para curado;

## 2.3 Estrutura do documento

O documento encontra-se organizado em três capítulos:

- O capítulo introdutório, onde se faz uma abordagem ao contexto do problema, motivação e objetivos;
- O capítulo de implementação, onde é descrita toda a implementação do código e a sua devida explicação e o diagrama de classes;
- O capítulo de conclusão, onde são retiradas as conclusões desta terceira fase de desenvolvimento da aplicação;

## Capítulo 2

### **3 Implementação**

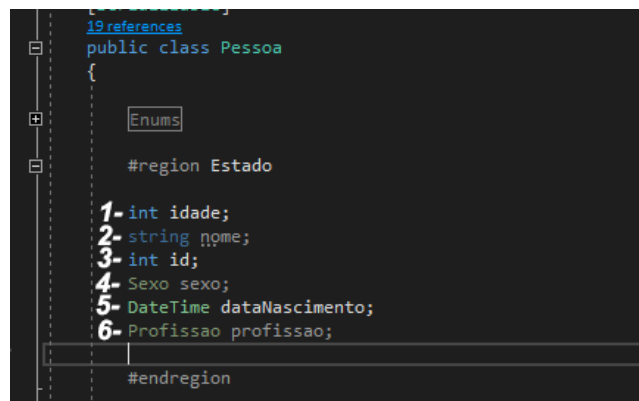
Neste capítulo é descrita a metodologia de gestão do projecto, bem como uma abordagem às estruturas de dados utilizadas.

#### **3.1 Descrição do problema**

O projeto consiste na criação de um sistema informático que permita gerir um hospital. Para isto é necessário que o sistema permita o registo de pessoas, infeções e até hospitais, também necessária a consulta desses registos como por exemplo consultar pessoas utilizando um ID.

## 3.2 Estruturas de dados

Nesta fase do trabalho cada classe está associado a um conjunto de parâmetros ou métodos.



```
19references
public class Pessoa
{
    #region Estado
    1-int idade;
    2-string nome;
    3-int id;
    4-Sexo sexo;
    5-DateTime dataNascimento;
    6-Profissao profissao;
    #endregion
}
```

Figura 1: Pessoa

Esta classe define uma **pessoa** :

- Uma valor inteiro representando a idade;
- Uma string representando o nome;
- Um valor inteiro único que representa a pessoa;
- Um enum type, com valores de 0 a 2 onde 0 = Masculino, 1= Feminino, 2=Indeciso;
- Uma estrutura para nos dar a data de nascimento da pessoa identificada por dia/mês/ano;
- Um enum type, com valores de 0 a 6 onde Advogado = 0, Medico = 1, Estudante = 2, DESEMPREGADO = 3, Dentista = 4, Policia = 5 e Cozinheiro = 6.



```

15 references
public class Infecao
{
    #region Estado
    //Tipo de infeçao ou doenca apresentada
    1- string tipo;
    // Nome
    2- string nome;
    3- // Exemplo: tipo - virus , nome - Varicela

    #endregion
}

```

Figura 2: Infecção

Esta classe define uma **infecção** :

- Uma string que caracteriza o tipo da infeçao do género ( virus);
- Uma string que caracteriza o nome da infeçao do género (Varicela).

```

15 references
public class Hospital
{
    #region Estado
    1- int lotacao;
    2- static int nDoentes;
    3- static Doentes[] doentes;
    4- string nome;

    #endregion
}

```

Figura 3: Hospital

Esta classe define um **Hospital**:

- Um valor inteiro que representa a lotação do Hospital;
- Um valor inteiro que representa o número de doentes;
- Uma estrutura para associar os doentes ao hospital;
- Um string que identifica o nome do Hospital;

### 3.3 Diagrama de classes

Na Figura 4 demonstramos o diagrama de classes que possibilita a análise de cada camada e do que está contido nela.

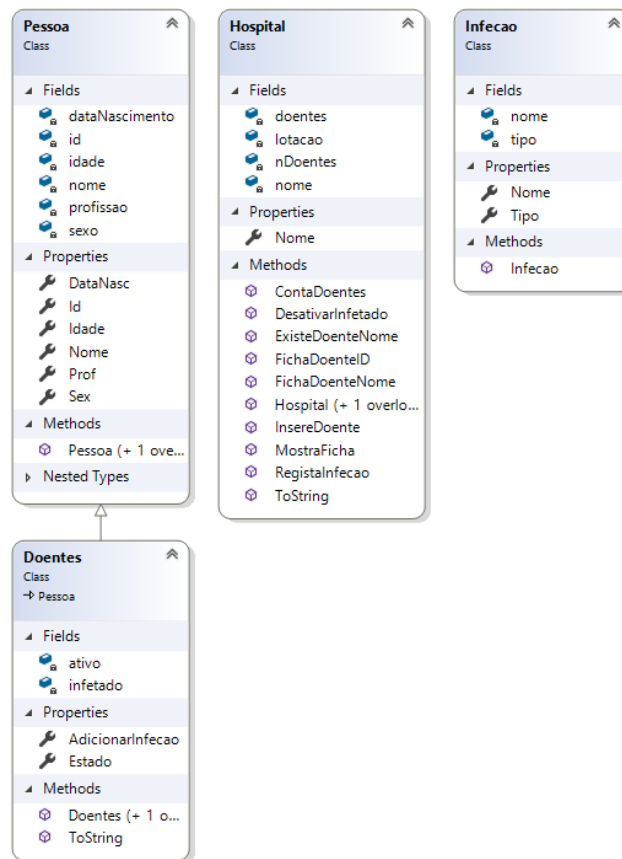


Figura 4: Diagrama de classes

### 3.4 Solução

O projecto consiste em melhorar os nossos conhecimentos na linguagem multi-paradigma C#. Todo o trabalho desenvolvido neste trabalho prático provem de um melhoramento do trabalho anterior que com o decorrer das aulas e o agregar de novos conhecimentos.

## Capitulo 3

### 4 Alterações no TP

Neste capitulo iremos abordar a alteração que realizamos para TP que foi programa que utilizamos para gerar a API, o Doxygen.

#### 4.1 Doxygen

Uma breve descrição do doxygen, é usado para gerar documentação compatível com C# , C, C++, java e mais linguagens. Doxygen é uma ferramenta para gerar documentação e referências de códigos sendo esta documentação escrita utilizando código fonte facilitando assim a sua escrita.

#### 4.1.1 Exemplo de execução

Aqui iremos demonstrar um exemplo de execução e como utilizamos o Doxygen.

Temos na figura 5 a enumeração com os passos que realizamos para executar o Doxygen, passando a explicar cada ponto:

- 1- Serve para mudar o nome do projeto que estará depois na main page da API;
- 2- Serve como uma introdução ao que está disposto na API;
- 3- Serve para identificar a versão ou ID do projeto;
- 4- Tal como o ponto 2 e 3 é um ponto facultativo onde podemos dar upload a um logo para o nosso projeto;
- 5- Seleccionamos a pasta onde temos o nosso código para que o Doxygen extraia os comentários e código desse projeto;
- 6- Definimos o destino do produto final do doxygen;
- 7- Aba run onde executamos o Doxygen;

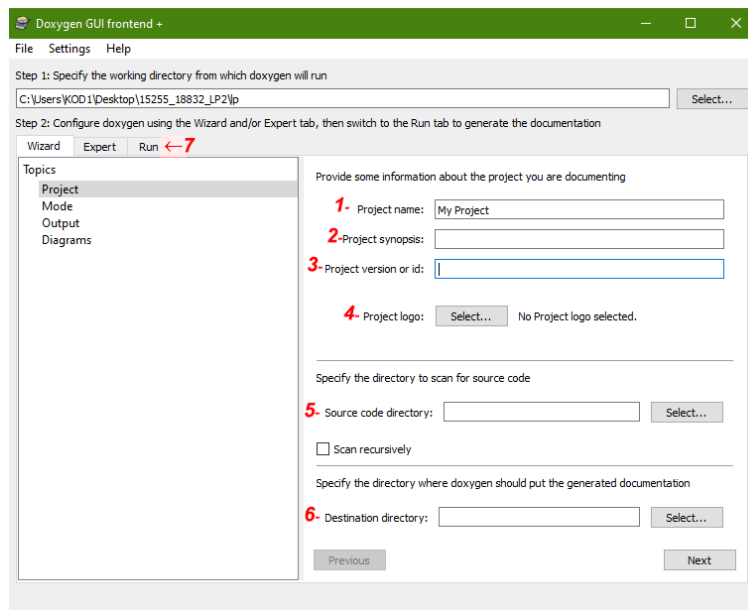


Figura 5: Doxygen

Neste projeto em c# tivemos que seleccionar algumas opções na aba expert para que extraísse todos os comentários e funções mesmo que fossem de classe privada ou static, pois sem essas opções o resultado final não seria o mesmo, essas opções podem ser observadas na Figura 6.

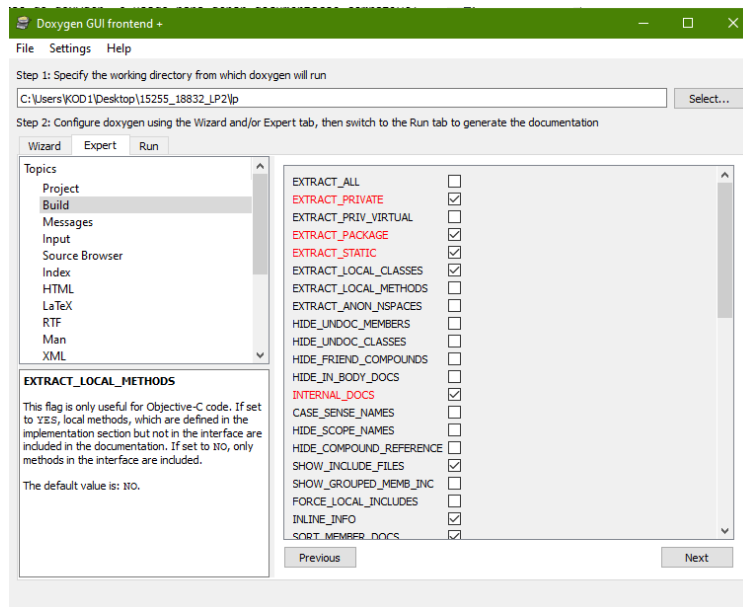


Figura 6: Opções Expert

### 4.1.2 Exemplo de código

Neste sub capítulo iremos demonstrar um exemplo de código e a sua aparência final no .html gerado pelo Doxygen. Na Figura 7 temos um exemplo de comentário e a sua aparência final na Figura 8.

```
/*! \mainpage Linguagem de programação II
 *
 * \section intro_sec Projeto I Fase 4 (Melhoria)
 *
 * Ferramenta de gestão de:
 *
 *     - Doentes e não doentes;
 *     - Infecções;
 *     - Hospitais;
 *
 * \subsection autores Autores:
 *
 *     Pedro Macedo - a15255
 *     Ricardo Silva - a18832
 *
 */
```

Figura 7: Comentários no Código

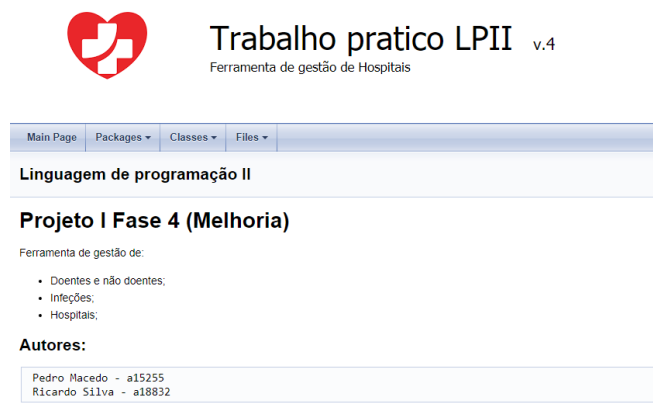


Figura 8: Resultado Doxygen

## Capítulo 4

## 5 Conclusão

Neste capítulo, será descrita a conclusão do trabalho desenvolvido e aprendizagens.

### 5.1 Lições aprendidas

O desenvolvimento deste projeto com várias fases permitiu-nos colocar em prática o conteúdo lecionado na U.C. de Linguagens de Programação II. Com este estilo de entrega em fases permitiu-nos também delinear objetivos semanais para cada fase.

### 5.2 Apreciação final

Em jeito de conclusão achamos que, este trabalho nos permitiu aprofundar os conhecimentos obtidos nas aulas teóricas e aplicá-los a nível prático. No geral, consideramos pouco satisfatórios os resultados apresentados pois não realizamos nenhuma melhoria em termos de código apenas acrescentamos o Doxygen uma ferramenta que já nos é familiar da UC do 1º semestre mas desta vez num projeto em `c#`, o não acrescentar de funcionalidades/ alteração de código deveu-se á falta de tempo de tentar implementar outras coisas porque pensávamos que a data de entrega iria ser outra e na data que nos foi proposta o nosso grupo infelizmente tinha outros recursos e decidiu focar-se nas cadeiras em que não passou e no fim tentar fazer a melhoria.