



INSTITUTO POLITÉCNICO  
DO CÁVADO E DO AVE  
ESCOLA SUPERIOR DE TECNOLOGIA

## RELATÓRIO DE TRABALHO PRÁTICO

# Sistema que permita gerir uma urgência de um hospital - Fase 2

**RICARDO SAMPAIO E CLÁUDIO SILVA**

**ALUNOS Nº18827 E Nº18843**

**REPOSITÓRIO GIT:**

**[HTTPS://GITHUB.COM/RICARDOSAMPAIO12121/18827\\_18843\\_LP2](https://github.com/RICARDOSAMPAIO12121/18827_18843_LP2)**

Trabalho realizado sob a orientação de:

Luís Ferreira

<b>1 Introdução</b>	<b>3</b>
1.1 Contextualização	3
1.2 Motivação e objetivos	4
1.3 Estrutura do documento	4
<b>2 Implementação</b>	<b>5</b>
2.1 Estrutura de dados	5
2.2 Design pattern	5
2.3 Diagrama de classes	6
HospitalER	6
HospitalBO	7
Datalayer	7
HospitalRN	8
<b>3 Conclusão</b>	<b>9</b>

## 1 Introdução

Neste capítulo falamos sobre a contextualização do projeto, a motivação e os objetivos e, para terminar, descrevemos a estrutura do documento.

### 1.1 Contextualização

Cada vez mais os hospitais aderem às tecnologias mais recentes de maneira a melhorar o atendimento e o tempo de espera das pessoas.

Com este projeto pretendemos criar uma aplicação em C# que permita gerir o serviço de urgências de um hospital pelos enfermeiros e médicos.

C# é uma linguagem de programação, desenvolvida pela *Microsoft*, orientada a objetos.

A programação orientada a objetos(POO) é um paradigma de programação baseado no conceito de “objetos” que podem conter atributos e métodos.

## 1.2 Motivação e objetivos

O objetivo deste projeto foi criar uma aplicação que fosse capaz de gerir o serviço de urgências de um hospital.

As funções da aplicação são:

- Fazer a ficha dos pacientes e editar o nome e a data se necessário;
- Adicionar pacientes à fila de espera para a triagem;
- “Criar” médicos, dando-lhes um código único;
- Editar nome de médicos e eliminá-los se necessário;
- Chamar pacientes para a triagem, sendo-lhes dado um código de 1(pouco grave) a 5(muito grave) de grau de prioridade dado pelo “médico” e adicioná-los à fila de espera para a consulta; → Chamar paciente para a consulta de acordo com a prioridade;
- É também possível listar todas as filas de espera se necessário.

## 1.3 Estrutura do documento

Este documento está estruturado em três partes.

Na primeira parte falamos sobre a contextualização do problema, a motivação e objetivos e a estrutura do documento.

Na segunda parte falamos acerca das estruturas de dados utilizadas e apresentamos um diagrama de classes.

## 2 Implementação

### 2.1 Estrutura de dados

Todos os dados dos pacientes e dos médicos são guardados num ficheiro binário individual.

Para guardar os pacientes que estão à espera para entrar na triagem, utilizamos uma *Queue genérica* visto ser ideal para este caso, que é inserir cada paciente no fim da *Queue* e retirar no início da *Queue*.

Os médicos em trabalho são armazenados numa *List* já que a remoção de cada um da lista não pode ser ordenada como na *Queue*.

Os pacientes, depois de passaram na triagem, são inseridos numa *Sorted List genérica*, escolhemos esta estrutura de dados porque, neste caso, precisamos de utilizar prioridades. Cada paciente, ao sair da triagem, é lhe atribuído uma prioridade dada pelo utilizador e, de seguida, são inseridos na *SortedList* pela ordem de prioridade (de 5 mínimo até 1 máximo). Para isto utilizamos o nosso próprio *Icomparar* dado que o comparador predefinido do *SortedList* não permite a inserção de prioridades repetidas.

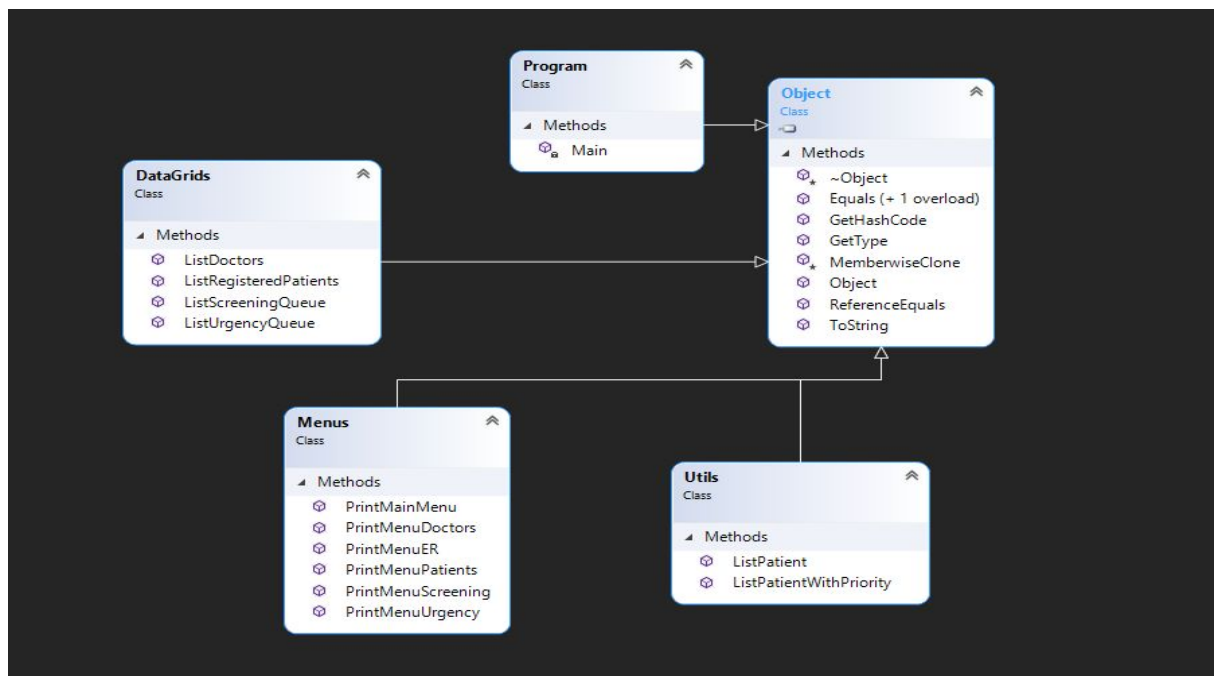
### 2.2 Design pattern

Como design pattern utilizamos o N-Tier por ser o que mais nos agradou. O programa está dividido em 6 projetos(.dll) que são os seguintes:

- 1) HospitalER: Trata da interação com o utilizadores;
- 2) HospitalBO: Trata dos objetos;
- 3) DataLayer: Trata dos dados;
- 4) HospitalRN: Trata de ser o intermediário entre o HospitalER e o DataLayer;
- 5) Exceptions: Trata das exceções;
- 6) Validations: Trata de validar os dados introduzidos pelo utilizador.

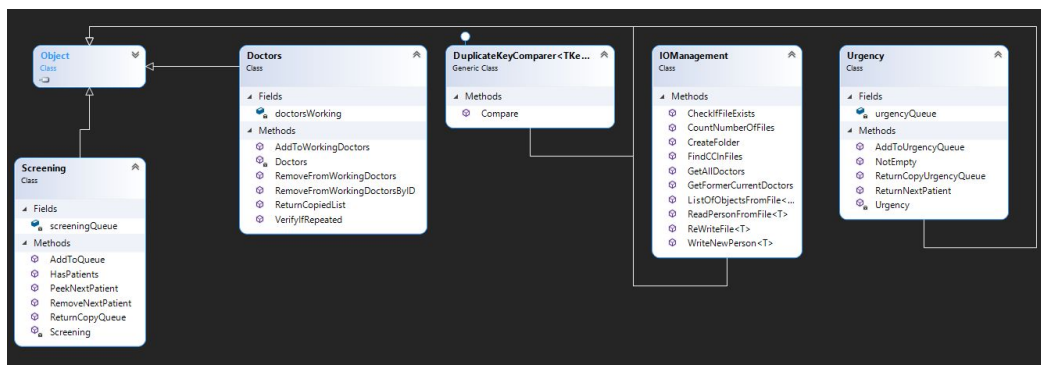
## 2.3 Diagrama de classes

### HospitalER

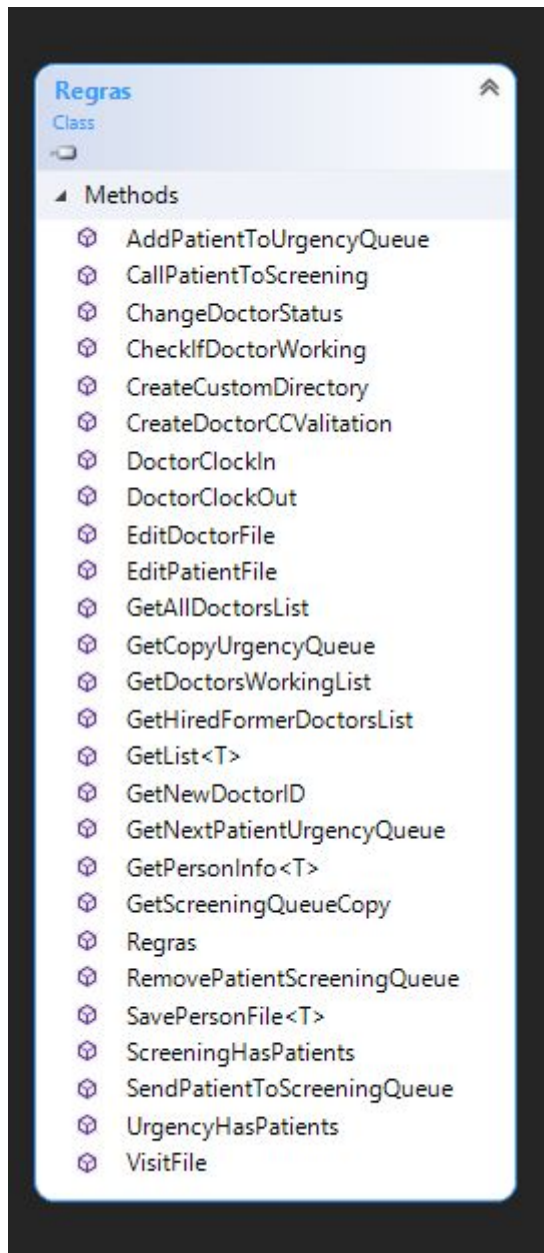


## HospitalBO

## Datalayer



## HospitalRN





### 3 Conclusão

Com este projeto conseguimos implementar e aprimorar o que aprendemos na cadeira de LP2 até à data.

Para a próxima fase, pretendemos, para além de melhorar o que já foi feito até agora, adicionar um registo para cada paciente, isto é, registar sempre que um paciente vai ao hospital e também um registo para armazenar quantos e que pacientes é que cada médico atendeu.