



Relatório

Linguagens de Programação I

Aluno: João Nuno Gonçalves Veloso

Professores: Luís Ferreira

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, junho, 2020

Resumo

Encontramo-nos perante um trabalho prático atinente à unidade curricular de Linguagens de Programação II. Este tem como principal finalidade a análise e implementação de uma solução na linguagem C# (csharp).

Foram apreendidos diversos conceitos no decorrer das aulas de Linguagens de Programação II, aplicando-os num sistema que permita gerir uma urgência de um hospital: listas de espera, prioridades, etc...;

Esta primeira fase foi realizada com recurso à linguagem C#, recorrendo da aplicação Visual Studio.

De salientar, ainda que, o armazenamento da informação introduzida no programa foi feito através do recurso a ficheiros binários.

Palavras-Chave: linguagem C#; sistema de urgência hospitalar; ficheiros.

Índice de Figuras

Figura 1 - fluxograma 1	9
-------------------------------	---

1. Introdução	5
1.1. Contextualização	5
1.2. Motivação e objetivos	5
1.3. Estrutura do Documento	5
2. Estudo da questão	6
3. Implementação	7
3.1. Descrição do problema	7
3.2. Solução	7
4. Programa	9
5. Conclusão	11
5.1. Lições aprendidas	11

1. Introdução

1.1.Contextualização

Neste trabalho prático relativo à unidade curricular de Linguagem de Programação II tem como principal objetivo a implementação em C# de uma resolução que permita gerir um sistema de urgência hospitalar. Nesta situação atual em que vivemos reconhecemos a necessidade de agilizar todo o processo de entrada no âmbito de uma urgência.

1.2.Motivação e objetivos

Em todos os trabalhos que são propostos, normalmente, deparo-me sempre com desafios. Estes são sempre importantes para a criação de uma maior motivação em busca de soluções mais eficazes.

Um dos grandes propósitos deste trabalho é consolidar as ideias, bem como as bases e noções assimiladas nas aulas, com o complemento de uma pesquisa que foi determinante para a elaboração do trabalho em questão.

Ao longo do presente trabalho tentarei descortinar e encontrar soluções pragmáticas e criativas para o problema em que nos encontramos.

1.3.Estrutura do Documento

O documento é apresentado de uma forma simplificado, de modo a que seja possível uma leitura e compreensão do conteúdo do mesmo.

2. Estudo da questão

Tivemos a apresentação de diversos assuntos e temas para a realização deste trabalho. Tendo em consideração a conjuntura em que vivemos nos dias de hoje, onde somos confrontados com os mais diversos problemas, principalmente, nos serviços dos nossos hospitais nomeadamente, nas urgências que ficaram afetadas com todo este novo mundo em que vivemos.

Assim sendo, torna-se necessário encontrar soluções que permitam a agilização e uma maior organização da entrada nas urgências hospitalares. Por conseguinte, o tema e o desafio escolhido foi encontrar uma solução, um sistema que permitisse gerir uma urgência de um hospital: listas de espera, prioridades.

3. Implementação

3.1.Descrição do problema

O cerne da questão trata-se em encontrar um sistema que permita gerir uma urgência de um hospital: listas de espera, prioridades, etc. No entanto, é categórico e necessário que este sistema seja o mais eficaz possível.

3.2.Solução

Dentro dos mais variados conceitos abordados destacamos, neste momento, o conceito de programação orientada a objetos (POO). Esta espécie de programação compreende, na sua grande maioria, o uso e emprego de classes, assumindo, estas um papel de extrema relevância para a resolução do problema aqui a tratar.

1. CLASSE

Uma classe é uma forma de definir um tipo de dado em uma linguagem orientada a objeto. Ela é formada por dados e comportamentos. Para definir os dados são utilizados os atributos, e para definir o comportamento são utilizados métodos. Depois da classe estar definida podem ser criados diferentes objetos que utilizam a classe.

1.1.VARIÁVEIS

Na programação, uma variável é um objeto (uma posição, frequentemente localizada na memória) capaz de reter e representar um valor ou expressão. Enquanto as variáveis só "existem" em tempo de execução, elas são associadas a "nomes", chamados identificadores, durante o tempo de desenvolvimento. Quando nos referimos à variável, neste caso, em c#, estamos a olhar para a lista de atributos de uma classe.

1.2. PROPRIEDADES

Uma propriedade fornece um mecanismo flexível para ler, escrever e trabalhar no valor de uma variável privada. Propriedades podem “proteger” a variável e é a partir disto que nós conseguimos “chegar” a ela.

1.3. CONSTRUTORES

Os construtores são tipos especiais de métodos usados para criar e inicializar objetos. É através deste tipo especial de método, chamado de construtor, que você cria instâncias de uma classe. Assim, os construtores permitem que o programador crie objetos da classe e defina valores padrão, limite a instanciação e grave códigos flexíveis e fáceis de ler.

2. BIBLIOTECAS

As bibliotecas não nada mais que classes “guardadas” num armazém. As bibliotecas são particularmente úteis para guardar classes que o programador não necessita de ter junto ao programa. Através disto, ele protege e guarda as classes sem ser necessário estarem “expostas” e em c# são guardadas como DLL's.

3. HERANÇAS

A herança é das coisas mais importantes de c#, pois permite-nos definir uma classe a derivar de outra classe, o que ajuda a ser mais fácil criar e fazer manutenção da aplicação. Isto também ajuda a reutilizar código e acelera o tempo de implementação.

4. FICHEIROS

O armazenamento de informação é importante para não a perder no encerramento da aplicação. Nesta aplicação apliquei o armazenamento da informação através de ficheiros binários, porque só é possível ler a informação dentro de o programa, assim a informação fica protegida.

4. Programa

CLASSES

Neste programa tenho 5 classes. As classes **Staff** e **Patient** herdam da classe **Person**.

A classe **Staff** representa todo o staff da urgência e a classe **Patient** representa todos os pacientes que cheguem à urgência.

A classe **MedicalAppointment** representa as consultas realizadas na urgência. Por fim temos a classe **Urgency** que serve para apenas para instanciar as listas e para interligar todas as classes.

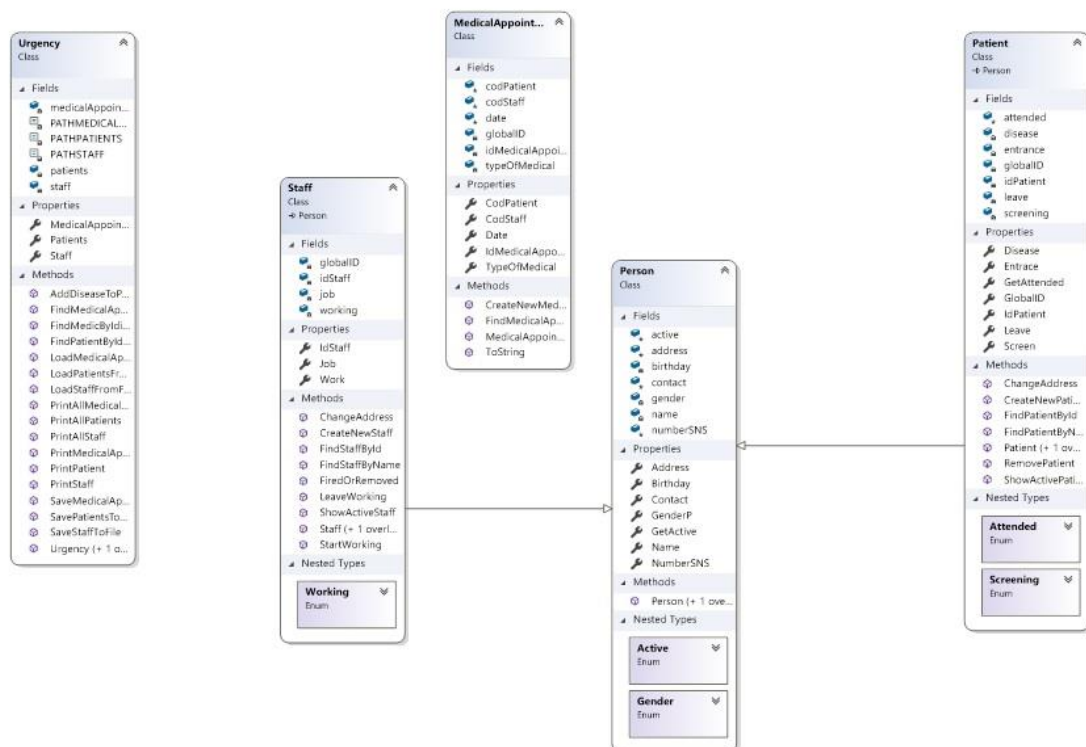


Figura 1- Diagrama de Classes

BIBLIOTECAS

Nesta aplicação usei diversas bibliotecas, para além das que já nos são fornecidas pelo C#, criei uma para guardar as classes que quero proteger chamada **ClassLibrary**

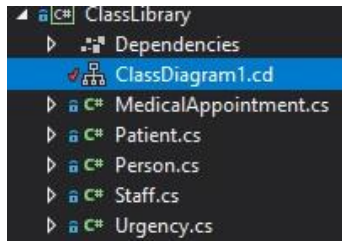


Figura 2- Biblioteca criada

▪ FICHEIROS

Como já referi, acabei por guardar cada lista, como possuo 3 listas acabei por gravar em 3 ficheiros.

```
private List<Patient> patients = new List<Patient>();  
private List<Staff> staff = new List<Staff>();  
private List<MedicalAppointment> medicalAppointments = new List<MedicalAppointment>();  
  
private const string PATHPATIENTS = @"..\..\Patients.bin";  
private const string PATHSTAFF = @"..\..\Staff.bin";  
private const string PATHMEDICALAPPOINTMENTS = @"..\..\MedicalAppointments.bin";
```

Figura 3- Listas e Ficheiros

5. Conclusão

5.1.Lições aprendidas

Chegados ao fim do trabalho, impõe-se tecer breves considerações finais.

Em primeiro lugar, gostaria de realçar que a abordagem feita nas aulas dos conceitos aqui retratados, juntamente com um denso trabalho de pesquisa, foram essenciais para a elaboração desta tarefa.

Adicionalmente, sinto que consegui fixar e consolidar os conteúdos lecionados em aula, mas também senti uma grande necessidade de complementar estes conteúdos com um grande trabalho de pesquisa, que me permitiram aprender mais.

Ora, esta linguagem, C#, destina-se a ser simples, com propósito geral e orientada a objetos. Contudo, é sempre uma nova linguagem e que, claramente diverge da linguagem C.

Como seria de esperar num desafio, encontrei, ao longo do caminho, alguns problemas e dificuldades, mas que acabaram sempre por ser solucionados.

Finalmente, é de salientar que o armazenamento da informação introduzida no programa foi feito através do recurso a ficheiros binários.

4.2.Apreciação final

No cômputo geral, considero que este trabalho teve um balanço bastante positivo, visto que permitiu que eu obtivesse uma maior à vontade com a linguagem em questão, C#.