**Relatório**

**Linguagens de Programação II**

**Aluno: Victor Abreu-18841**

**Professor: Luís Ferreira**

**Engenharia de Sistemas Informáticos**

Barcelos, abril, 2020

Resumo

O presente trabalho tem como tema o desenvolvimento de um programa que permita gerir pessoas infetadas, desde os seus dados pessoais, até seus dados de utentes em possíveis diferentes hospitais.

Todas as bases de dados são distribuídas para melhor organização e consulta de caso de infeção por região, idade, sexo, e por tipo de infeção, podendo ainda ser incrementado para demais pesquisas caso necessário. A aplicação também contabiliza os casos por cada secção de pesquisa alem das fichas de utentes da mesma.

**Palavras-Chave:** Infeção, Hospital, utente.

Índice

[1. Introdução 4](#_Toc38824004)

[1.1. Contextualização 4](#_Toc38824005)

[1.2. Motivação e objetivos 4](#_Toc38824006)

[1.3. Estrutura do Programa 4](#_Toc38824007)

[2. Implementação 5](#_Toc38824008)

[2.1. Descrição do problema 5](#_Toc38824009)

[2.2. Solução 5](#_Toc38824010)

[3. Conclusão 7](#_Toc38824011)

[3.1. Lições aprendidas 7](#_Toc38824012)

[3.2. Apreciação final 7](#_Toc38824013)

# Introdução

## Contextualização

O tema deste trabalho foi escolhido e pensado de acordo com a situação atual a que se ocorre no mundo devido a pandemia do vírus Covid-19 no ano de 2020.

## Motivação e objetivos

O objetivo central deste trabalho é desenvolver conhecimentos para a manipulação de dados na linguagem de C#, criação e modularização por classes tal como a sua implementação e desenvolvimento de métodos das mesmas.

## Estrutura do Programa

O programa desenvolvido está divido em 5 classes, tendo a sua classe principal Main onde a interação com utilizador é desenvolvida e com os métodos para chamar as funções das demais classes.

É constituído também com a classe “Pai” denominada de Pessoa a qual corresponde aos atributos principais de uma pessoa e que será usada para relacionar com dados de utentes.

A seguir o programa possui também a classe Utente “filho” que herda os dados de sua classe basse Pessoa, e acrescenta ao utente alem de seus atributos de pessoa um número de identificação de utente e o tipo de infeção que o mesmo possui.

Por fim o programa possui a classe Hospital onde são inseridos os utentes a partir de métodos de inserção e de contabilização dos mesmo ao atribuir o seu número de identificação. É nessa classe também que são feitas a manipulações dos dados de ficha de utentes pelas várias opções de pesquisa.

# Implementação

## Descrição do problema

O primeiro problema a ser pensado e concretizado é de que maneira poderia se distribuir as diferentes classes para a melhor economia e simplificação de código de maneira mais efetiva para o tratamento de dados

A seguir da criação de classes é necessário definir a escolhe de atributos de classe e de suas propriedades para a manipulação de seus atributos.

Por fim a criação de métodos eficazes para inserção de dados em arrays e a pesquisa nos mesmo de acordo com a região, idade, sexo e tipo de infeção.

## Solução

**Para resolução do primeiro problema foi a organização em classes.**

Primeiramente foi pensado de que forma se poderia distribuir as classes, então foi definida a classe Pessoa para a criação de atributos comuns a todas as pessoas, que são também utentes. Logo faz sentido que esta seja a classe pai, e que a classe Utentes seja a classe filho que herda os atributos de Pessoas e adiciona somente os atributos comuns de um utente.

A seguir foi definido que é necessário haver uma classe Hospital onde será feita a gestão e alocação de utentes, tal como os métodos para a manipulação dos utentes que poderá ser usada em cada hospital, como pesquisas e inserção e contabilização dos mesmos.

**Para resolução do segundo problema foi a definição de cada classe.**

A classe Pessoa e principal, foram atribuídas as variáveis para a inserção dos atributos de uma pessoa tais como o seu nome, uma identificação, região de morada, sexo e idade.

A seguir foi definido o construtor de classe com dados definidos exteriormente na classe Main do programa. E por fim suas propriedades para a manipulação dos atributos pessoa utilizando as funções GET e SET.

A classe Utente, foi definida como a classe “filho” de Pessoa, pois essa herda os atributos de uma pessoa e adiciona o tipo de infeção e o id à pessoa, deste modo não é necessária a criação de novos atributos e propriedades, visto os mesmo já foram definidos na classe pai.

O construtor da classe Utentes herda os atributos da classe pessoa e insere o tipo de infeção e seu id. E esses dados adicionais ~são também tratados com suas respetivas propriedades na presente classe.

A classe Hospital é a classe que cria um “hospital” ao qual são alocadas as informações de cada utente, e é nesta classe onde são definidos os métodos para a contabilização dos utentes desse hospital, os métodos para exibição de todas as fichas de utentes desse hospital e métodos para as pesquisas por região, idade, sexo e tipo de infeção.

**Para resolução do terceiro problema foi a definição de cada método.**

O método para exibição de todas as fichas foi definido como uma pesquisa do array de Utente a partir de um ciclo de repetição do tipo “for” de modo a exibir os dados de cada posição dos atributos inseridos neste arrays, ou seja, para a exibição das informações de cada utente.

Os demais métodos para a pesquisa de acordo com a região, idade etc. segue o mesmo padrão do método de exibição de todas as fichas, mas desta vez cada função recebe o dado inserido pelo utilizador e é feita a comparação do dado com a informação existente no arrays, assim deste modo, no ciclo de repetição só é exibido as fichas de acordo com a informação inserido pelo utilizador. Por exemplo: se o utilizador escolhe pesquisar as fichas de utentes de acordo com a região do “Porto”, o método faz a verificação de quais são as fichas que tem o atributo “região” igual ao inserido pelo utilizador, e exibe todas as fichas dessa “região”.

Em cada método de pesquisa e verificação é também criada uma nova variável auxiliar que ira se contabilizar o total de fichas que existe no atributo escolhido pelo utilizador, assim a variável de tipo inteiro funciona como um contador que dentro do ciclo for contabiliza cada ficha a cada vez que ele repete-se, logo deste modo economiza-se a criação de novos métodos somente para contabilização tal como a sobrecarga desnecessária de código.

# Conclusão

## Lições aprendidas

Em conclusão da primeira parte do trabalho proposto foi adquiro os conhecimentos acerca da distribuição de dados e métodos por classes, e do tipo de classes no que diz respeito a heranças, sendo assim os pilares para a programação orientada a objetos. Foi também reforçado a prática de manipulação de arrays em C#, da implementação de propriedades de classe e como instanciar no Main as mesmas.

## Apreciação final

Por fim a primeira parte do trabalho proposto consolida os conhecimentos inicias do programa em C# e orientada a objetos, que permite a criação de códigos efetivos e mais simples de acordo com o que foi aprendido no primeiro semestre na linguagem base C. Assim sendo uma linguagem de medio nível for apresentar funções pré-definidas tal com “Console.” Que facilita em inúmeras operações, possibilitou em minha opinião a resolução dos problemas a serem implementados de maneiras mais rápido e simples em relação a linguagem C.

Bibliografia

**Não existem origens no documento atual.**