**Resumo**

O presente relatório pretende dar a conhecer, o trabalho prático realizado na unidade curricular de Programação Orientada a Objetos, com o objetivo de projetar uma aplicação em C# para a gestão de um Centro de Saúde. Este trabalho foi dividido em duas partes para uma mais implementação facilitada.

Neste documento apresentamos uma explicação sobre como está estruturado o programa.

**Palavras-Chaves:** Programação Orientada a Objetos, Centro de Saúde, C#, Classes, Objetos, Atributos, Métodos, Herança.

**Abstract**

This report aims to present the practical work carried out on the Object Oriented Programming course, with the objective of designing an application in C# for the management of a Health Center. This work was divided into two parts for easier implementation.

In this document we present an explanation of how the program was structured.

**Keywords**: Object-Oriented Programming, Health, C#, Algorithms, Classes, Objects, Attributes, Methods, Inheritance.

* **Lista de Acrónimos e Notação**

UML Unified Modeling Language

* **Introdução**

Este projeto enquadra-se na Unidade Curricular de Programação Orientada a Objetos, lecionada pelo docente Tiago Castro, do curso de Engenharia Informática Médica do Instituto Politécnico do Cávado e do Ave. O objetivo deste trabalho é realizar uma solução em C# para um tema do quotidiano.

* **Objetivos**

Neste tópico, apresentamos os objetivos do trabalho proposto. Entre as metas estão a análise de problemas reais, o aprimoramento da experiência e das habilidades no desenvolvimento de software, e o fortalecimento do conhecimento da linguagem C#. Também procuramos estruturar uma solução utilizando um sistema de classes, bem documentado e realizar testes adequados no código desenvolvido.

* **Estrutura do documento**

Quanto à organização do documento, este será dividido em dois tópicos principais: Implementação Teórica e Implementação Prática. No tópico da implementação teórica iremos detalhar o planeamento realizado para abordar a resolução do projeto. Já no tópico da implementação prática será apresentada a execução prática do problema, mostrando como as soluções foram implementadas.

* **Enunciado**

Neste trabalho é-nos proposto nove temas para implementar na linguagem C#, sendo estes os seguintes:

* Gestão de Atividades de Socorro;
* Gestão de Obra de Construção Civil;
* Gestão de Condomínios;
* Gestão de rendas/imóveis;
* Gestão de alojamentos turísticos;
* Helpdesk;
* Comércio eletrónico;
* Gestão de jardim zoológico;
* Gestão de Centro de Saúde;

Para este projeto, pretendemos realizar um sistema que permita gerir um Centro de Saúde.

* **Implementação Teórica**

Neste capítulo estará descrito todo o planeamento feito para a resolução deste projeto. Este está dividido em vários tópicos que depois de definidos foram implementados na prática que iremos ver no capítulo a seguir.

* **3.1 Estruturação do Projeto em Camadas**

A solução para este trabalho será dividida em três projetos para uma maior organização do código.

* **CentrodeSaudeProject** – Este projeto possui todas as classes do projeto, assim como os enumeradores usados por nós na emplementação;
* **Gestão do CS** – apenas possui o ficheiro *program.cs* onde está localizada a função main;
* **Menus**  – Este projeto possui todos os menus do trabalho usados para facilitar a utilização do sistema;

Esta estrutura em camadas ajuda na legibilidade do código, permite também a modificação de uma camada sem afetar as outras. Por fim, facilita a compreensão do sistema como um todo.

* **3.2 Definição das Estruturas de Dados a utilizar**

Neste projeto, foi essencial definir uma estrutura de dados adequada para atender às necessidades do trabalho prático. Isso envolveu a criação de classes, listas e outros tipos de dados personalizados.

* **3.3 Classes**

Para a realização deste projeto, foi necessário definir algumas classes que servem como modelo para criar os objetos. Neste subcapítulo, apresentamos todas as classes do projeto. Após análise e discussão, identificamos como essenciais a criação de sete classes, que serão detalhadas a seguir:

* **ClassePaciente**: Representa um paciente do centro de saúde.
* **ClasseFuncionário**: Representa um funcionário do centro de saúde, podendo este ser um médico ou enfermeiro
* **ClasseConsulta:** Representa uma consulta associada à ficha do paciente
* **ClasseExame**: Representa um exame médico associado a uma consulta que por sua vez está associado ao paciente
* **ClasseCama**: Representa uma cama disponível ou ocupada no centro de saúde
* **ClasseQuarto**: Representa um quarto do hospital, com uma lotação máxima de duas camas
* **CentroSaude**: Representa o centro de saúde em si e centraliza os dados e operações do sistema de gestão

**3.4 Diagrama de Classes**

Um diagrama de classes é uma ferramenta de modelação utilizada para representar a estrutura de um sistema orientado a objetos. Este descreve as classes, anteriormente referidas no tópico anterior, juntamente com os seus atributos, métodos e as relações entre elas. Desta forma, o diagrama proporciona uma visão clara da organização do software e das interações entre os seus componentes.

Na imagem seguinte, é apresentado o diagrama de classes:



**Bibliografia**

[1] Moodle IPCA

<https://elearning2.ipca.pt/2223>

<consultado a 25-11-2022>

[2] Microsoft Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/>

<consultado a 12-12-2022>

[3] StackOverflow

<https://stackoverflow.com/>

<consultado a 28-12-2022>

[4] GeeksForGeeks

<https://www.geeksforgeeks.org/csharp-programming-language/>

<consultado a 28-12-2022>

[5] Draw.io

<https://app.diagrams.net/>

<consultado a 28-12-2022>

[5] Github

<https://github.com/>

<consultado a 30-12-2022>