



**INSTITUTO POLITÉCNICO  
DO CÁVADO E DO AVE  
ESCOLA SUPERIOR  
DE TECNOLOGIA**

**Instituto Politécnico do Cávado e do Ave**

**Escola Superior de Tecnologia**

**Licenciatura**

**em**

**Engenharia de Sistemas Informáticos**

**Gestão de Centro de Saúde – WinForm APP**

Gloria Martins - 22719

**Dezembro de 2024**

# Resumo

Este documento foi elaborado como um apêndice à solução desenvolvida para o trabalho avaliativo proposto na disciplina de Programação Orientada a Objetos. Nele, serão discutidas as lógicas implementadas em cada solução apresentada para os problemas propostos.

O trabalho proposto visa avaliar e reconhecer os conhecimentos absorvidos primordialmente sobre o funcionamento de classes, modificadores de acesso, interfaces, tipos de classes, interfaces, encapsulamento e criatividade na resolução do objetivo sugerido.

## Índice

<b>1.Introdução .....</b>	<b>4</b>
1.1 Enquadramento .....	4
1.2 Objetivos .....	4
1.3 Metodologia .....	4
<b>2.Estado de Arte .....</b>	<b>5</b>
<b>3.Especificação .....</b>	<b>6</b>
<b>4.Desenvolvimento .....</b>	<b>7</b>
<b>5. Conclusão e Trabalho Futuro.....</b>	<b>11</b>
<b>6. Webgrafia .....</b>	<b>12</b>

# **1.Introdução**

Nesta seção serão apresentados o propósito deste relatório, os objetivos a se concluir através desta documentação e por fim como o trabalho foi realizado e quais tecnologias foram utilizadas.

## **1.1 Enquadramento**

O presente trabalho caracteriza-se por uma componente avaliativa da cadeira de Programação Orientada a Objetos da Licenciatura em Engenharia de Sistemas Informáticos.

## **1.2 Objetivos**

Este relatório tem como principal objetivo documentar as soluções desenvolvidas para os problemas apresentados no trabalho prático de Programação Orientada a Objetos do ano letivo 2024/25, aplicando os conhecimentos adquiridos em aula até o presente momento.

## **1.3 Metodologia**

A realização deste trabalho decorreu individualmente no decorrer do 1º semestre do 2º ano de curso.

Para concretização da solução foi utilizado o ambiente de desenvolvimento integrado (IDE) Visual Studio 2022, em conjunto da linguagem de programação C#, a plataforma .NET e o Windows Forms como a framework de interface do utilizador (UI). Além das tecnologias citadas, recorreu-se ao MySQL como base de dados, juntamente da ferramenta MySQLWorkbench, para manipulação de dados.

## **2.Estado de Arte**

Em busca da consolidação e aprimoramento dos conhecimentos já obtidos, a disciplina Programação Orientada Objetos vem com o intuito de aprofundar o conhecimento no aluno na área de programação, abordando um dos paradigmas mais utilizados no que diz respeito ao desenvolvimento de software.

Neste projeto visa-se aumentar competências na programação estruturada por objetos/classes. A partir desta premissa é possível e expectável desdobrar diversas ramificações dentro do proposto, como: Tipos de classe, herança, polimorfismo, interfaces e ect.

### 3.Especificação

O trabalho proposto foi dividido em duas partes, cada uma com uma data de entrega distinta. Em um primeiro momento foi realizado a fase 01 e posteriormente a fase 2, sendo acompanhado por cada uma dessas fases uma defesa realizada oralmente perante o código desenvolvido.

#### **Fase 01 – Definição e Criação de Clases      Data de Entrega: 15 de Novembro de 2024**

Criação das classes Principais, tendo em conta os modificadores de acesso e o encapsulamento, e criação da base de dados

#### **Fase 2 – Finalização da Implementação**

Criação de novas classes, implementação de interfaces, conceitos de programação orientada a objetos e implementação do código restante.

## 4.Desenvolvimento

Nesta seção serão encontrados os métodos de soluções utilizados para a resolução das duas soluções requeridas no enunciado do trabalho.

### Resoluções

#### Solução Fase 1

Nesta primeira fase foi, de forma geral, foi definido o tema do projeto a de desenvolver e deu-se a criação da estrutura base da solução.

Para a escolha do tema, foi fornecido pelo docente uma lista sugestiva, que abordavam os mais variados “tópicos”, ainda oferecendo a possibilidade ao educando de criar seu próprio tema- este que posteriormente foi alvo de validação.

Para a realização desta avaliação, foi escolhido o tema “Gestão de Centros de Saúde”, onde foram definidas as funcionalidades esperadas, bem como o modelo de dados. Dentre as funcionalidades expectáveis foram determinadas:

- Gestão de Paciente por meio do Staff
- Gestão de Consultas e Exames efetuados pelos pacientes
- *Gestão de Staff, Exames, Categorias e Especialização de Staff por meio de um Admin do sistema (Facultativo)*

Em busca de uma melhor qualidade na resolução apresentada, foi dispensado o armazenamento de dados apenas em memória e implementado a utilização de uma base de dados relacional.

De forma a estruturar os relacionamentos da base de dados, foi recorrido ao Diagrama UML de Classes (*Figura 01*).

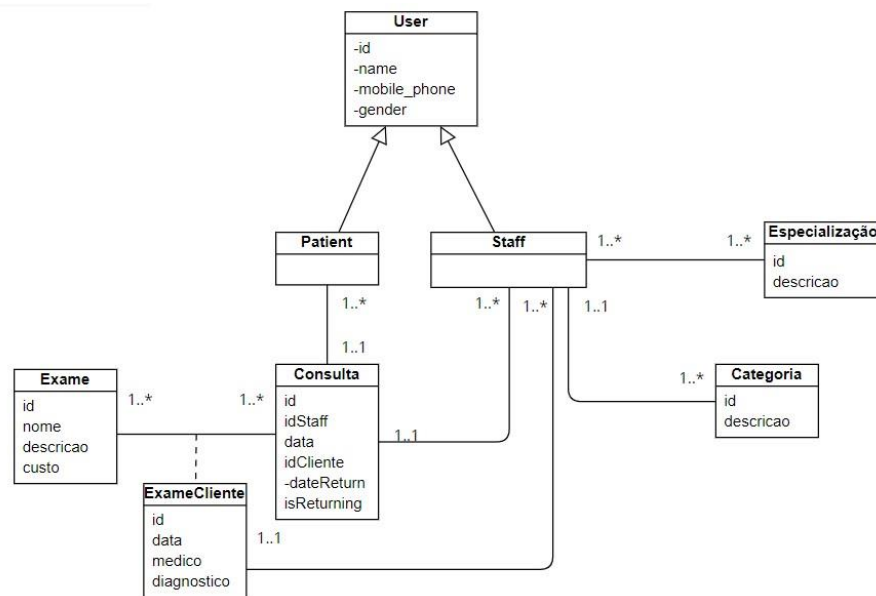


Figura 1 - Diagrama de Classes

Estabelecendo-se as relações entre as classes, passou-se a criação da base de dados em MySQL, na plataforma Workbench. Com a base de dados estruturada e as classes criadas, procedeu-se a conexão entre a aplicação e a própria.

### Conexão WinForms e Base de dados

De forma a pensar na estrutura do código, foi criado um ficheiro *App.config*, que armazena a *url* da base de dados local, de forma a evitar a instanciação da mesma. Então foi criada uma classe chamada *DatabaseConfig*, que é responsável pelo método de criação da conexão com o *MySQL*, permitindo a execução de *queries* através da app.

### Repositórios

Foram criados repositórios dedicados a cada “unidade” do sistema, sendo assim o projeto contém:

- Repositório de Pacientes
- Repositório de Staff
- Repositório de Exames
- Repositório de Consultas



## Solução Fase 2

Nesta fase ocorreu a maior parte do desenvolvimento do projeto, consolidando conhecimentos abrangidos e apreendidos em sala de aula. Dentre os principais fundamentos da programação orientada a objetos, nesse projeto é possível encontrar:

## Princípios Programação orientada a Objetos

### Herança

Tal como o próprio nome indica, herança é algo herdado, “passado” de pai para filho. Assim tudo que estiver presente na classe “pai” e possuírem os modificadores de acesso *public* – vistos globalmente - e *protected* – vistos somente por classes filhas, puderam aceder livremente os seus recursos. Ressaltando que propriedades e métodos com o modificador *private*, apenas são visíveis na própria classe em questão.

Nesta solução foram utilizadas a herança da classe de base dados para as classes de repositório, para que assim a conexão SQL fosse visível somente entre elas. Além da herança de uma interface, referida posteriormente, nas classes de repositório.

### Polimorfismo

Polimorfismo nada mais é que algo/objeto/método poder exercer mais uma forma em simultâneo. Na codificação deste projeto foi adotado um output “standard” em todas as classes de repositório, a frente explicada, para tal recorreu-se uma interface com um método chamado Log, essa interface foi posteriormente herdada em todas as classes de repositório, obrigando-as a implementação no método contido na interface.

### Encapsulamento

É um recurso presente em programação orientada a objetos, que torna visível para o restante código apenas aquilo necessário, ocultando e “otimizando” certos códigos. Um exemplo foi a criação da Propriedade Gender, pertencente ao modelo de dados *user*, em que automaticamente recebe o *input* introduzido pelo utilizador e converte o caracter representante do género para caixa alta

## Estruturação do Código

A solução foi estruturada tendo em conta o seguinte esqueleto:

- **Models:** Pasta onde serão encontrados todos os modelos de dados utilizados no projeto
- **Repository:** Pasta com classe que suportando a lógica e comunicação com o *MySQL*.
- **Services:** Pasta cujas classes fazem ponte entre o *repositor* e a interface gráfica.
- **Interface/Classe Abstrata:** Janelas de Visualizações utilizadas na aplicação e suas respectivas lógicas

## Considerações Adicionais

Para marcação de exames foi criado um algoritmo que agenda o exame selecionado para os próximos 7 dias de forma aleatória.

Ao clicar sobre qualquer consulta ou paciente, abrirá um novo *form* com mais detalhes sobre.

Cada consulta possui detalhes adicionais, onde o médico indica pontos relevantes sobre o paciente. Esse campo tem sua disponibilidade variável de acordo com o tempo.

- No dia da consulta o campo é aberto possibilitando o preenchimento e a edição do mesmo
- Caso a consulta seja acedida antes do dia da mesma, o campo de detalhes adicionais não aparecerá
- Caso a consulta seja acedida a uma posterior à consulta, será possível visualizar as anotações feitas pelo médico.

## **5. Conclusão e Trabalho Futuro**

A manutenibilidade de uma aplicação é algo a ser em atenção para que ela vá de encontro com os requisitos planejados e que melhore cada vez mais a experiência do utilizador.

A aplicação entregue permite adicionar pacientes, fazer a gestão dos mesmos, agendar exames e consultas.

De forma geral alguns pontos a serem tratados em futuras atualizações seria a gestão de staff e suas respectivas especialidades e categorias, melhorias gráficas e otimizações no código.

## 6. Webgrafia

<https://stackify.com/oop-concept-for-beginners-what-is-encapsulation/>

<https://www.coursera.org/articles/encapsulation>

<https://medium.com/@vinjenks/depth-first-hierarchy-in-c-f9e0599abeeb>

<https://medium.com/@vinjenks/depth-first-hierarchy-in-c-f9e0599abeeb>

<https://github.com/IPCA-Content/POO-LESI-PL-202425>