



Relatório

Linguagem de Programação II

Aluno/os: Pedro Vieira Simões e João da Costa Apresentação

21152
21140

Professor/es: Luís Ferreira

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, abril, 2021

Resumo

O relatório que se segue descreve o projeto realizado no âmbito da unidade curricular Linguagem de Programação 2, integrada no curso Engenharia de Sistemas Informáticos.

O projeto desenvolvido em C# baseia-se em um sistema que permite ao utilizador a gestão de pessoas infetadas numa situação de crise de saúde pública. Entre os quais permite a análise de dados em tempo real e própria inserção e alteração dos dados de uma pessoa entre muitas outras funcionalidades que proporcionam ao utilizador uma boa utilização dos recursos oferecidos.

Para o seu desenvolvimento foi utilizado o Visual Studio 2019.

1. Introdução

1.1. Motivação

Visto a crise de saúde pública que estamos a vivenciar atualmente, uma pandemia por parte do vírus Covid-19 decidimos abordar este tema no nosso trabalho e construir um programa que ajudasse na gestão e análise desses dados.

1.2. Estrutura do Relatório

O relatório consiste em uma pequena introdução que resume o que o programa faz de uma forma geral. Depois fala das funcionalidades de uma forma mais específica e de todas as melhorias realizadas em comparação com a 1ª fase do desenvolvimento deste projeto.

2. SOLID

Neste projeto também é adotado princípios SOLID que torna o programa bem implementado.

3. Ficheiros

Outro ponto importante que foi possível se adicionar nesta segunda fase foi a implementação das files, o que permitiu que não fossem perdidos os dados das nossas listas.

4. Coleções

O uso de lists no processo proporcionou a existência de memória dinâmica , o que com arrays teria sido complicado ainda permitiu o uso da imensa quantidade de funções que as lists nos podem proporcionar.

5. Exceções

Com exceções foi possível uma maior interação com os utilizadores caso ocorra a introdução de um dado de forma errada ou na leitura de um ficheiro para que o utilizador consiga entender o que ocorreu e dessa forma também não terminar o programa.

É utilizado as seguintes Exceções:

- `OverflowException;`
- `FormatException;`
- `FileLoadException;`
- `DirectoryNotFoundException;`

6. DLL

Foi utilizada dll e geradas para a implementação de NTier.

7. Arquitetura

Este projeto apresenta uma arquitetura desenvolvida por camadas NTier o que proporciona ao programador um ambiente mais organizada e melhor implementada.

O facto de estar dividida por camadas divide as responsabilidades e a velocidade de procura de algum método ou objeto.

O programa é constituído por 5 camadas entre as quais:

- BO (Business Object);
- BR (Business Rules);
- Menus;
- Gestao_Infetados;
- Dados;
- Inputs;

7.1. BO

Esta camada contém o objeto pessoa que contém:

Atributos:

- Id: posição dessa pessoa na lista;
- totTestes: total de testes a que uma pessoa se submeteu;
- nome;
- idade: idade da pessoa que é obtida da data de nascimento;
- cc: cartão de cidadão;
- dataNascimento;
- região: região onde reside;
- género;
- estado: estado obtido de uma enumeração que indica o estado do paciente (vacinado, infetado, recuperado, morto).

Construtor

O construtor da pessoa funciona como um construtor por defeito que não vai inicializar valores.

Propriedades

Vai tornar possível a utilização dos atributos fora da classe.

Métodos

- RegistrarPessoa regista uma pessoa completando todos os seus atributos e depois retorna ela;
- AlterarEstado declara uma pessoa morta;
- MostraPessoa;

7.2. BR

Esta camada contém uma class de verificações de dados introduzidos pelo utilizador em que retorna a veracidade deles:

- VerificaOP;
- VerificaDataNascimento;
- VerificaCCValido;

Contem outra class MyComparer que compara objetos de uma lista seguindo um sentido e usando um atributo escolhida pelo utilizador para noutra camada ser utilizada para ordenar uma lista de objetos ou mesmo comparação de objetos.

Esta camda ainda contem um conjunto de Regras para as pessoas que serve de ponte de ligação entre os menus e a class pessoas para assim garantir segurança dos dados e tornando o BR o único com acesso á camada de Dados.

Por fim uma class Data que chama todas as funções relacionadas com o carregar e guardar os dados nos files.

7.3. Dados

Contem 3 classes entre as quais:

Pessoas

Contem a lista das pessoas do tipo de objeto Pessoa e nesta class é feita a inserção da pessoa na lista depois da sua criação, listagem das pessoas que nela existam, pesquisas, contagens, ordenação, remoção de objetos da lista e ainda guardado os dados da lista em um file.

Registo

Objeto que contem algumas informações de uma ação feita na gestão do covid. O seu construtor recebe os dados e guarda eles. Vai desde remoções de pessoas a alterações no seu estado,...

Registos

Contem a lista de todos os registos.

7.4. Gestao_Infetados

Aqui encontra-se o Main que torna possível a execução do programa.

Nele tem o carregamento e guardar dos dados e ainda chama o Menu.

7.5. Menu

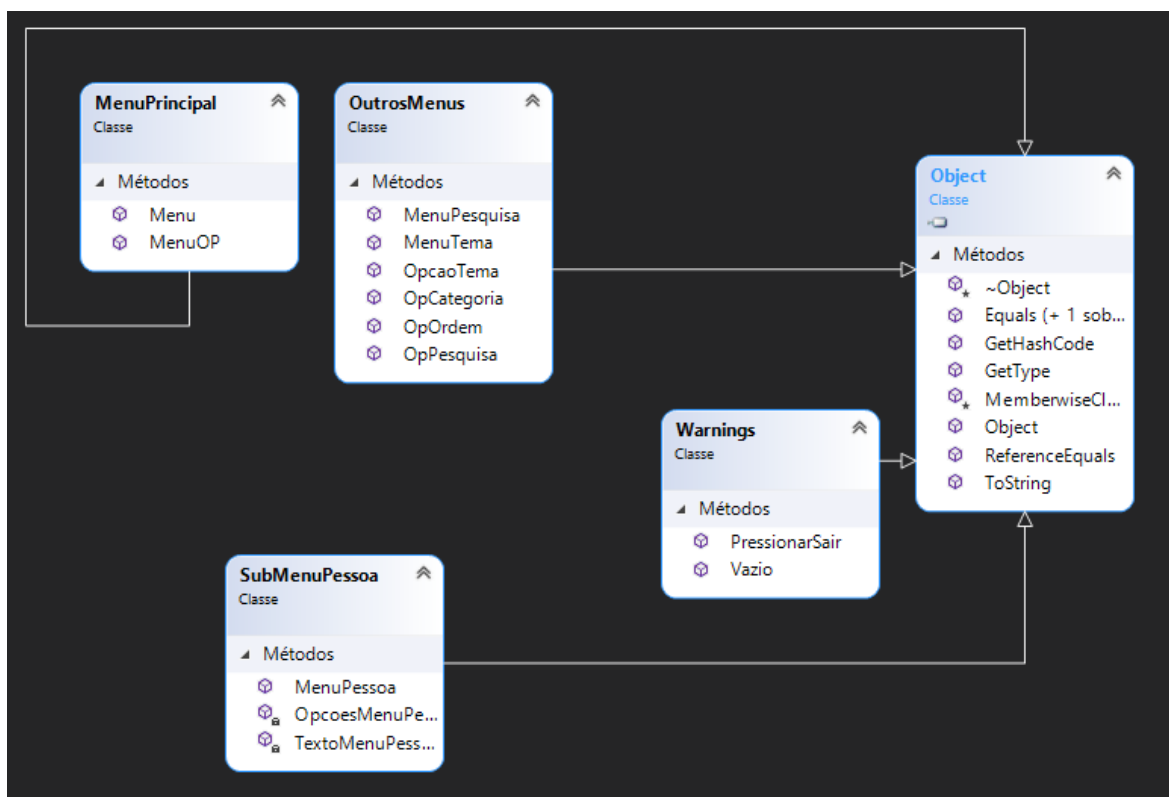
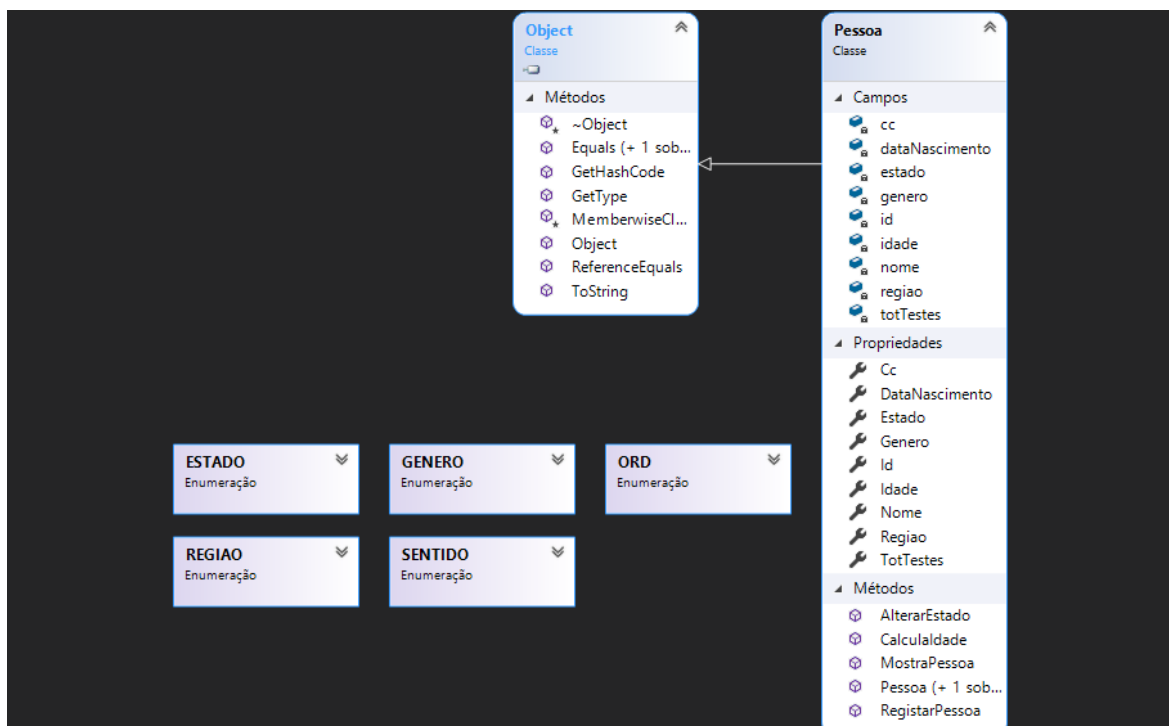
Todos os menus encontram-se aqui, desde o principal, ao menu pessoal e ainda tem uns mini menus para pequenas ações. Além disso tem uma class para avisos e outras pequenas informações bastante frequentes ao longo do programa.

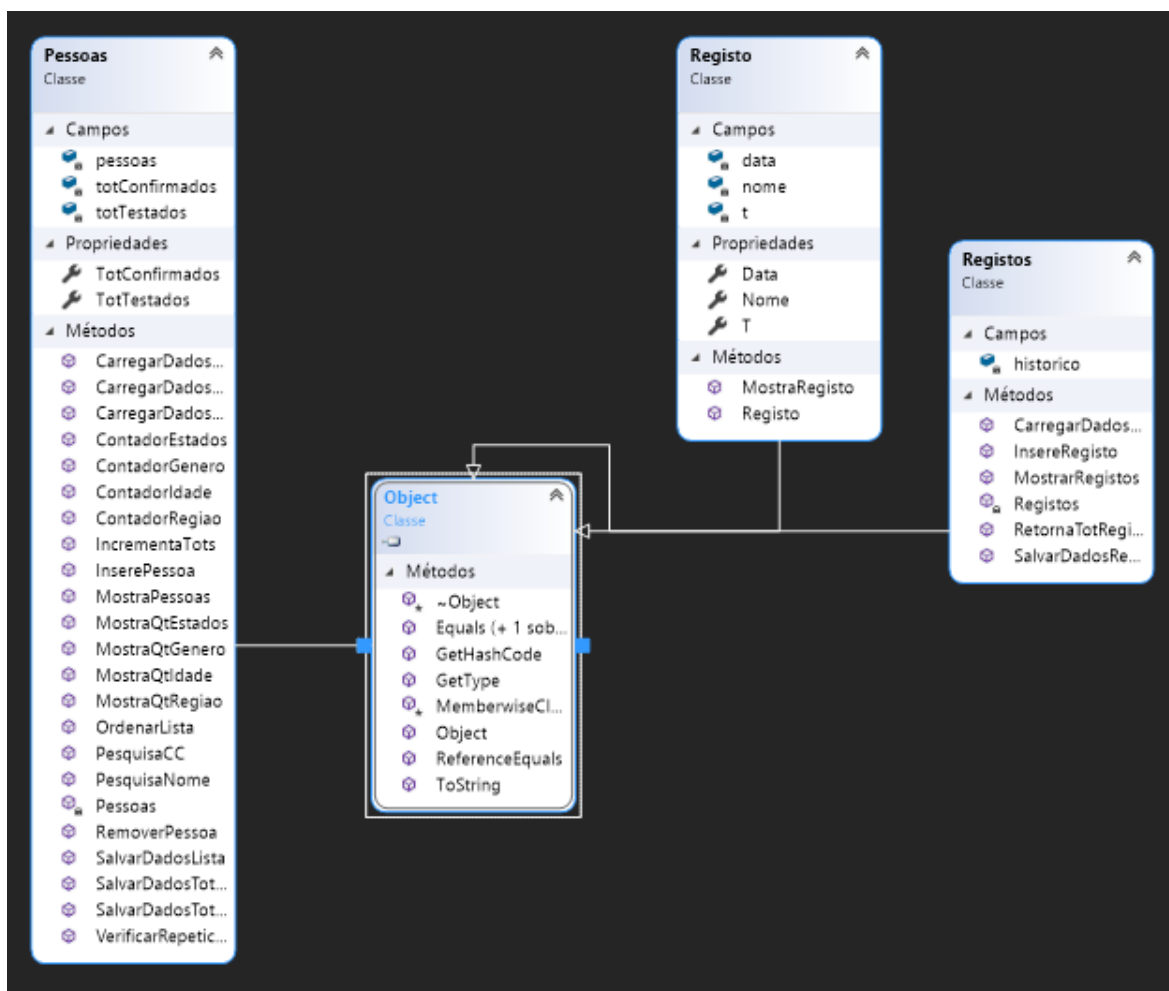
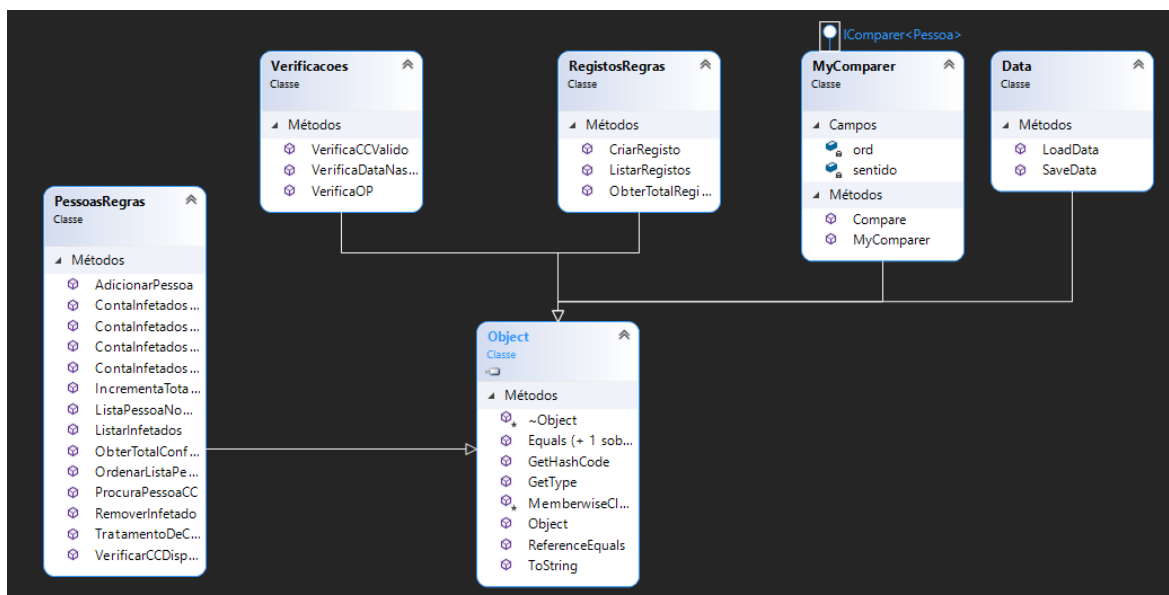
7.6. Inputs

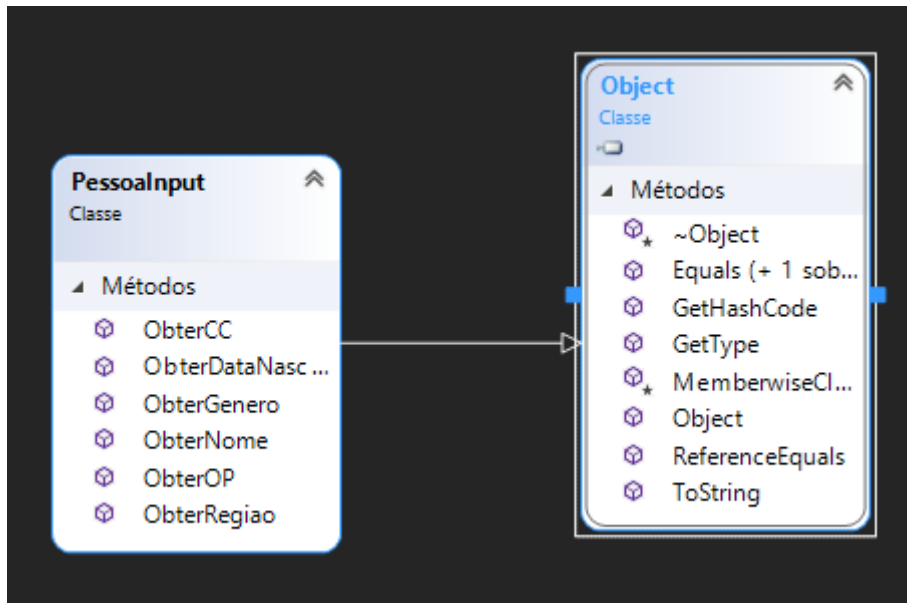
Camada que contem funções exclusivamente de introdução de dados que auxiliam a class Pessoa incluída na camada BO.

8. Diagramas

Como possível interpretação e análise de uma forma esquemática do nosso projeto e o respetivo funcionamento temos a demonstração através dos seguintes diagramas:







9. Conclusão

Com o desenvolvimento deste projeto foi possível a aplicação da maioria dos métodos recolhidos ao longo do semestre e das boas implementações e proporcionou uma evolução relativamente á primeira fase do mesmo.