INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE
ESCOLA SUPERIOR DE TECNOLOGIA

Trabalho de Programação Orientada a Objetos

Trabalho Realizado por: Ana Margarida Rodrigues (26418), Ariana Alves (26050), Gonçalo Soares (26056)

Unidade Curricular: Programação Orientada a Objetos

Curso: Engenharia Informática Médica

Docente: Luís Gonzaga Martins Ferreira

ÍNDICE

OBJETIVOS E ENQUADRAMENTO

O presente relatório referente ao trabalho prático, foi-nos proposto no âmbito da unidade curricular de Programação Orientada a Objetos e que tem como finalidade o desenvolvimento de uma solução em C#, com base na gestão de stress de profissionais de saúde. Para tal, iremos desenvolver uma aplicação que ajude no diagnóstico, controlo e redução dos níveis de stress destes profissionais.

A aplicação irá conter dados sobre os profissionais de saúde, como o nome, de modo a ser possível identificar os mesmos e a sua função dentro da unidade de saúde. Para além disso, também conterá dados relativos ao stress que serão coletados durante o exercício das funções de cada profissional dentro da unidade de saúde.

O objetivo será perceber quem são os profissionais de saúde que apresentam maior nível de stress e, tendo em conta a sua função na unidade de saúde, arranjar soluções que contribuam para a redução destes níveis, oferecendo assim uma melhor qualidade de vida e trabalho a cada profissional.

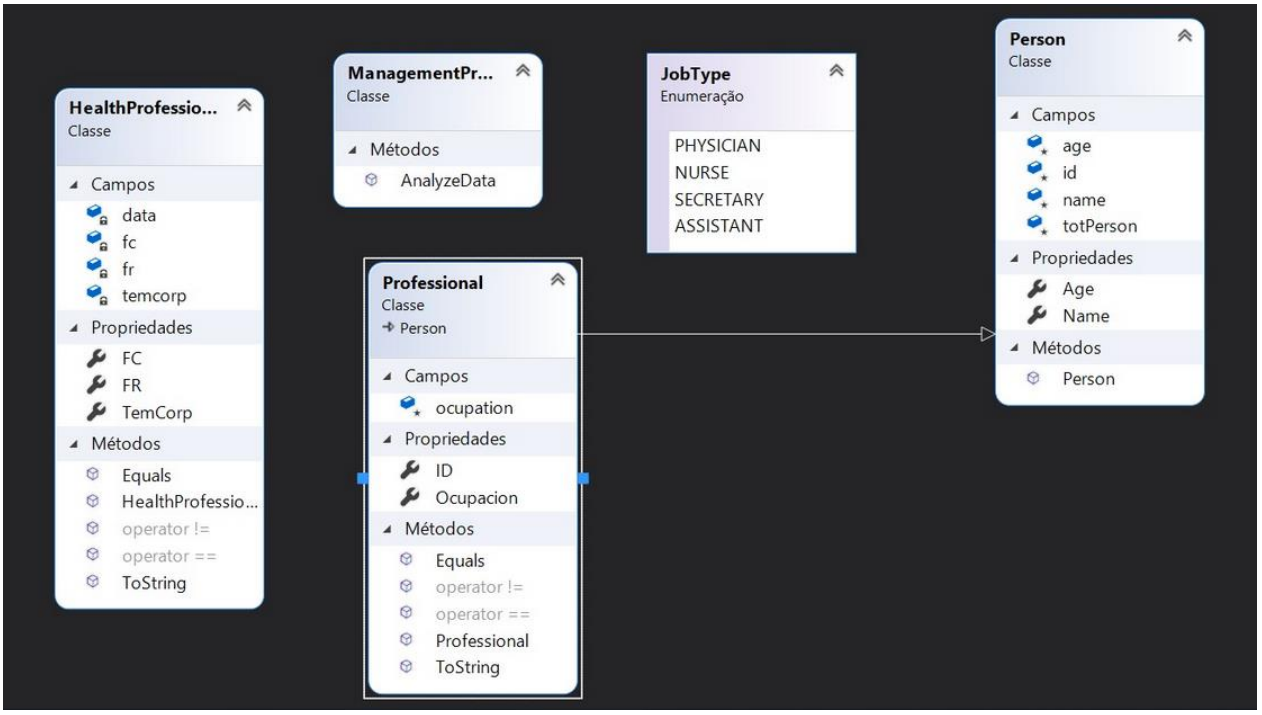
- **Fase 2**

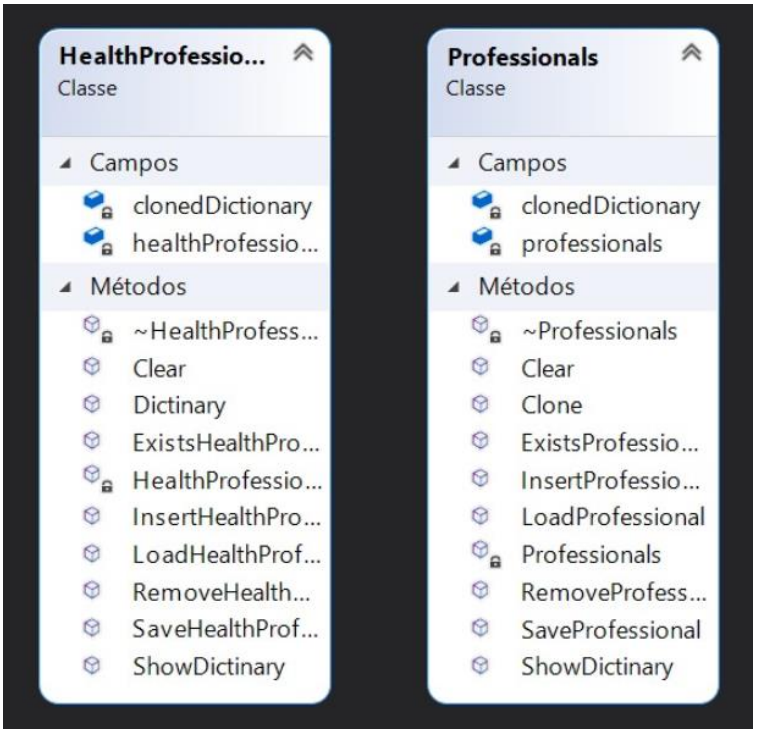
– Implementação final das classes e serviços

– Aplicação demonstradora dos serviços implementados

– Relatório final do trabalho realizado

DIAGRAMA DE CLASSES




HealthProfessio...
Classe
Campos
clonedDictionary
healthProfessio...
Métodos
~HealthProfess...
Clear
Dictinary
ExistsHealthPro...
HealthProfessio...
InsertHealthPro...
LoadHealthProf...
RemoveHealth...
SaveHealthProf...
ShowDictinary
Professionals
Classe
Campos
clonedDictionary
professionals
Métodos
~Professionals
Clear
Clone
ExistsProfessio...
InsertProfessio...
LoadProfessional
Professionals
RemoveProfess...
SaveProfessional
ShowDictinary


GereProfessionals
Classe
Métodos
Clear
ExistsProfessio...
InsertProfessio...
LoadProfessional
RemoverPofessi...
SaveProfessional
ShowProfessio...
GereStress
Classe
Métodos
Clear
ExistsHealthPro...
InsertHealthPro...
LoadHealthProf...
RemoveHealth...
SaveHealthProf...
Show


UnitTestHeathPr...
Classe
Métodos
TestMethd2
TestMethod1

ANÁLISE E ESPECIFICAÇÃO

Classes Objeto de Negócio:

- **HealthProfessional -** Recebe todos os dados/atributos relativos aos dados biométricos num determinado momento de um profissional de saúde, é responsável por criar uma ocorrência com os dados biométricos, a partir de um construtor. Contém redefinição dos operadores ==, != e Equals, de modo a ser possível comparar duas ocorrências de stress. Para além disso também nesta classe redefinimos ToString, de forma a poder ser usado nesta mesma classe.

- **Person -** Classe que vai permitir à classe Professional herdar os seguintes atributos: id, name, age e totPerson, sendo esta mesma classe que vai atribuir os ids de cada Profissional, a partir de totPerson que vai sendo incrementado a cada vez que é criado um novo profissional de saúde.

- **Professional -** Recebe todos os dados/atributos relativos a um profissional de saúde, é responsável por um registo do profissional no sistema, a partir de um construtor. Contém redefinição dos operadores ==, != e Equals, de forma a ser possível comparar dois profissionais de saúde. Para além disso também nesta classe redefinimos ToString, de modo a poder ser usado nesta mesma classe e mostrar os dados inseridos de um profissional.

- **ManagementProfessional -** Classe que contém implementada uma função que analisa a frequência cardíaca, frequência respiratória e temperatura corporal e a partir desses dados verifica se um profissional de saúde se encontra em stress num determinado momento.

Classes DL:

- **Professionals -** Estrutura de dados do tipo Dictionary <id,Professional>, responsável por guardar e gerir todos os registos de profissionais de saúde numa estrutura indexada e de pesquisa rápida. Nesta classe temos métodos para manipular o dictionary, tais como “Inserir”, “Ler” e “Guardar” em formato binário, “Mostrar”, “Remover”, “Existe” e “Clear” (elimina o conteúdo do dictionary).

- **HealthProfessionals -** Estrutura de dados do tipo Dictionary <id,HealthProfessional>, responsável por guardar e gerir todos os registos de dados biométricos de um profissional de saúde numa estrutura indexada e de pesquisa rápida. Nesta classe temos métodos para manipular o dictionary, tais como “Inserir”, “Ler” e “Guardar” em formato binário, “Mostrar”, “Remover”, “Existe” e “Clear” (elimina o conteúdo do dictionary).

Classes BL:

- **GereProfessionals -** Classe que valida os vários dados que foram inseridos relativos a um profissional, tendo em conta as regras de negócio definidas pelo cliente, antes de serem chamadas as funções que tratam da estrutura de dados Professional, como é o caso das funções “Inserir”, “Mostrar”, “Remover”, “Guardar” e “Ler”.

- **GereStress -** Classe que valida os vários dados biométricos inseridos, tendo em conta as regras de negócio definidas pelo cliente, antes de serem chamadas as funções que tratam da estrutura de dados HealthProfessional, como é o caso das funções “Inserir”, “Mostrar”, “Remover”, “Guardar” e “Ler”.

Classe TestHealthProfessional:

- **UnitTestHeathProfessional -** Classe que realiza testes unitários à estrutura HealthProfessionals, de modo a perceber como a mesma se comporta quando são inseridos dados inválidos.

IMPLEMENTAÇÃO/ESTRUTURA DE DADOS

Padronização NTier, na qual efetuados divisão por camadas tais como:

- BL – Camada responsável pela comunicação entre camadas de iteração e de dados, é responsável também pelas primeiras validações de dados ou de regras.
- DL – Para gestão de todas as estruturas de dados, inclusive gravação/carregamento recorrendo a ficheiros permanentes do tipo Bin.
- ObjetosdeNegocio – Camada responsável por gestão de objetos de negócio, nomeadamente objetos singulares utilizados por toda a estrutura de dados.
- PL – Camada de iteração responsável por devolver ou solicitar instruções ao utilizador. Esta camada valida em primeira instância a correta inserção dos dados necessários, assim como é responsável por mostrar todos os resultados das operações efetuadas.
- TestHealthProfessional – Testes unitários a funções da classe healthProfessional.

DOXYGEN

Em anexo é enviado também a documentação de todo o código do projeto efetuado durante a implementação.

CONCLUSÃO

Com este trabalho conseguimos adquirir os conhecimentos da Programação Orientada a Objetos, tais como os pilares deste tipo de linguagem de programação, qual a sua importância e como podem ser utilizados de forma a resolver problemas do quotidiano.