INSTITUTO POLITÉCNICO
DO CÁVADO E DO AVE
ESCOLA SUPERIOR DE TECNOLOGIA

RELATÓRIO DE TRABALHO PRÁTICO

Sistema de gestão para um *Hostel* em C# (Basics)

EMANUEL GOMES

A18869

E

NELSON CUNHA

A19241

Trabalho realizado sob a orientação de:

Luís Ferreira

Linguagens de Programação II

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, Abril de 2020

Índice

Lista de Tabelas

Lista de Figuras

1 Introdução

O presente trabalho desenvolvido no âmbito da disciplina Linguagens de Programação II, referente à avaliação da componente prática da disciplina, é baseado num sistema de gestão para um *Hostel.*

A ideia deste trabalho surgiu em seguimento de um encontro com um empresário no ramo hoteleiro no Minho.

Ou seja, para nós este trabalho não é apenas um trabalho académico queremos também tirar proveito de trabalho realizado e se possível aplicá-lo na prática.

A seguir, vamos apresentar brevemente os objetivos para este trabalho.

1.1 Objetivos

O trabalho tem como objetivo ser uma aplicação pratica e simples com o intuito de ajudar da melhor forma possível o gerente do hotel na manipulação dos dados dos hotéis.

2 Projeto: trabalho desenvolvido

Neste trabalho tentamos desenvolver ou melhor criar classes que achamos ser as mais pertinentes para estrutura e organizar da melhor forma possível esse projeto.

Criamos algumas classes que serão apresentadas no diagrama no capitulo seguinte, onde por exemplo pretendemos que certos campos de informação estejam disponíveis, por exemplo na tabela 1 que apresenta informação importante para o *Hostel*. É de salientar que ponderamos ainda acrescentar a posteriori mais um campo que seria o nº de contribuinte por exemplo.

Tabela 1:Hospedes

Nº de quarto	Nome	Morada	Mail	contacto	Check-in	Check-out	Total €

Na tabela abaixo, apresentamos outra exemplo de classe criada que é a classe funcionário, onde se destaca os campos que entendemos ser mais pertinentes para este projeto.

Tabela 2: Funcionários

Nome	Morada	Contacto	Mail	Cargo	Salario

Salientamos que na prática, os funcionários também podem reservar quartos no Hostel a condições preferências em relação aos Hospedes (*guest*).

Temos de referir ainda que já desenvolvemos uma parte do código que não consta no código enviado, isto porque tentamos dar mais enfase nesta primeira fase no desenvolvimento das classes que é uma fase crucial para bem estruturar o nosso trabalho.

Por exemplo para a classe ***quarto.cs*** onde já pensamos num método para verificar se um quarto pode alojar mais que um determinado número de pessoas e ainda um método que verifica se um determinado quarto já foi registado com o mesmo número via uma função booleana (ver em anexo).

Outro exemplo que está ainda em fase de estudo seria criar talvez uma classe ***Data.cs*** (propriamente dito) para gerir o "calendário" e aceder as datas com as respetivas reservas e estado dos quarto do hostel.

Outro ponto seria de inserir o nº de contribuinte (uma *string ncontribuinte;*) na classe ***hospede.cs*** (para emissão de fatura por exemplo). Ou seja depois iriamos acrescentar: *public string nContribuinte {set;get;}* e talvez implementar o método para verificar os dígitos inseridos (==9) do nº de contribuinte por exemplo.

Na classe **Program.cs** seria elaborado depois por exemplo um menu com uma opção de pesquisa (nº de reserva, nº hospede e funcionário entre outros(para a 2 fase do trabalho).

E finalmente na classe **Booking.cs** (reserva), desenvolver um método uma lista de reservas (ainda em fase de elaboração) e criar um método com uma *string* com informação da reserva.

3. Diagrama de classes

Neste capítulo, apresentamos as classes que achamos adequadas para este trabalho.

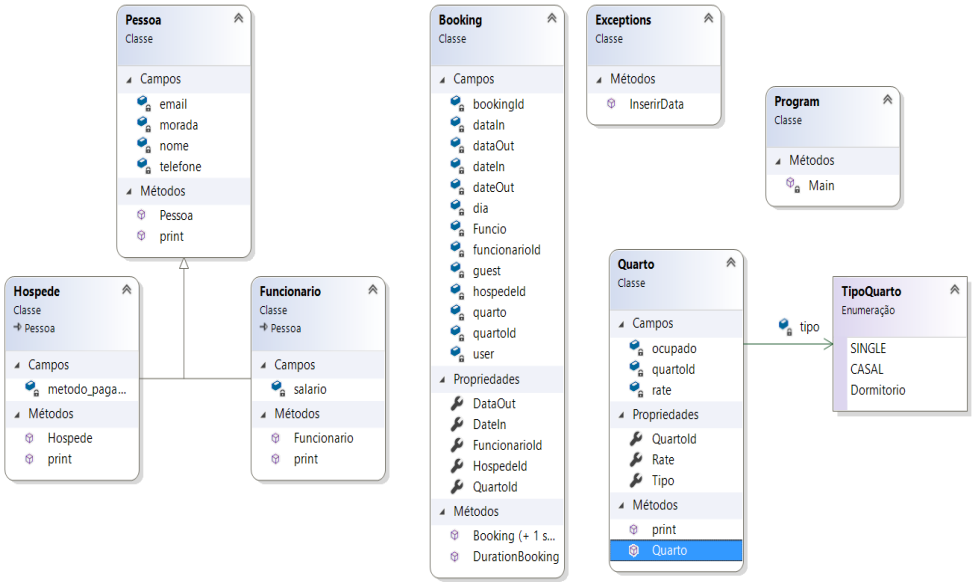

Figura 1 - Diagrama de classes

Salientamos que existe ainda algumas dúvidas quanto à necessidade de criar ainda mais uma ou até mais duas classes para este trabalho.

Por exemplo, a classe Pagamento, caso pretendemos gerir essa parte ou ainda da classe Data que seria uma espécie de calendário onde seria depois possível ver o estado do quarto (par exemplo, estado sujo, disponível ou ocupado) e as respetivas datas das reservas.

Um método que estamos a ponderar implementar na classe **Funcionario.cs** é a adição de prémios aos salários, mas para isso temos de ter uma atenção a novos métodos que possam ser necessários futuramente, como por exemplo , utilizar um novo método que retorne o numero de manutenções feitas pelo funcionário ao fim de um mês, e se esse numero for consideravelmente superior a media recebe a sua recompensa.

3 Conclusão

Apesar deste trabalho desenvolvido ainda se encontra numa fase muito inicial. No entanto, permitiu-nos perceber e entender a importância das classes, sendo um pilar para o bom desenvolvimento de qualquer trabalho, e ainda de familiarizar-nos com os critérios do *CLS compliance* (boas praticas).

Salientado ainda que no início (antes de assistir a respetiva cadeira de LP2), pensávamos que a classe e objeto era algo parecido, ora aprendemos que uma classe é uma estrutura/modelo (tipo esqueleto) enquanto que um objeto representa uma classe num dado momento de sua utilização, ou seja, o objeto é uma instância da classe (Stiefel & Oberg, 2001).

Neste trabalho surgiu ainda algumas dúvidas quanto à necessidade de criar certas classes nomeadamente as classes Data e Pagamento.

Mas acreditamos que para a segunda fase deste trabalho conseguiremos desenvolver um aplicativo simples e funcional que possa ser implementado.

Bibliografia

STIEFEL, M. & OBERG, R.J., (2001), Application Development Using C# and .NET, Prentice Hall;

Anexos

Código ainda em desenvolvimento:

Classe **Booking**

```
/// Método que realiza o registo de uma reserva.

public void RegistarReserva(int pessoasPorReserva, DateTime inicial, DateTime final)
{
    for (int i = 0; i < listaReserva.Length; i++)
    {
        if (listaReserva[i] == null)
        {
            listaReserva[i] = new Reserva(pessoasPorReserva, inicial, final);
            return;
        }
    }
}
```

Classe **Hospede**

inserir uma *string numcontribuinte*

Verificar se os dígitos inseridos do Contribuinte igual a 9:

```
public static bool CheckNumContri(string num)
    {
      if(num.Length == 9)
      {
        return true;
      }
      return false;
```

Classe **Quarto**:

```
public int  maxPessoas
    {
      get { return nmaxPessoas; }
      set { nmaxPessoas = value; }
    }

public Quarto(int nquarto, int nmaxPessoas)
    {
      this.nQuarto = nquarto;
      this.maxPessoas = nmaxpessoas;
    }
```

Método que verifica se um quarto pode albergar mais que um determinado número de pessoas.

```
public bool CheckNumeroPessoas(int nmaximopessoas)
   {
```

```
    if(nmaximopessoas > this.maxPessoas)
    {
       return false;
    }
    return true;
}
```