

Relatório

Programação Orientada a Objetos

Projeto Prático

Alunos:
Allan Sales Aleluia, a21990
Francisco Moreira Rebêlo, a16443
Rui Alexandre Borlido Magalhães, a22300

Docente:
Luís Ferreira

Escola Superior de Tecnologia
Licenciatura em Engenharia de Sistemas Informáticos - PL

Barcelos, 04 de dezembro de 2022

Índice

1. Introdução ao projeto – BusIPCA.....	5
2. Motivação	6
2.1. Pessoa	6
2.2. Local.....	7
2.3. Passe	7
2.4. Passes.....	8
2.5. Usuário	8
2.6. Program.....	9
3. Documentação de código	10
4. Conclusão.....	10

Índice de Imagens

Figura 1 - Exemplo do funcionamento das conexões.....	5
Figura 2 - Classe Pessoa.....	6
Figura 3 - Classe Local	7
Figura 4 - Classe Passe.....	7
Figura 5 - Classe Passes	8
Figura 6 - Classe Usuário	8
Figura 7 - Classe Program	9

1. Introdução ao projeto – BusIPCA

A BusIPCA é uma aplicação móvel que visa facilitar a utilização dos autocarros disponibilizados pelo IPCA. Nela será possível fazer aquisição de bilhetes, criação e carregamento de passes. Funções que hoje são feitas de maneira presencial e necessitam do deslocamento do utilizador até postos onde se possam fazer tais ações.

O tanto bilhete quanto passe iram gerar um QRCode, facilitando também a utilização e ajudando a preservar o meio ambiente com menos descarte de papel.

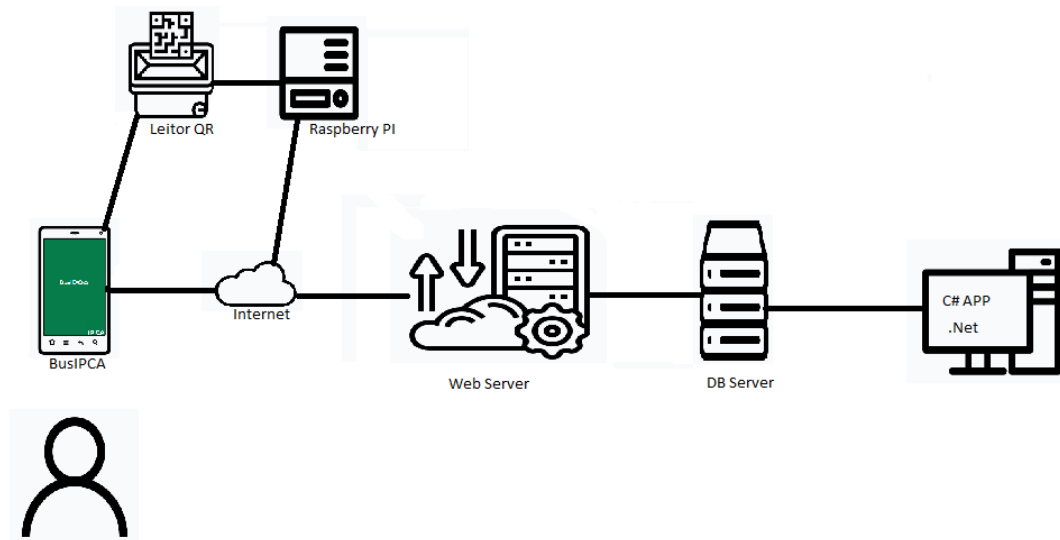
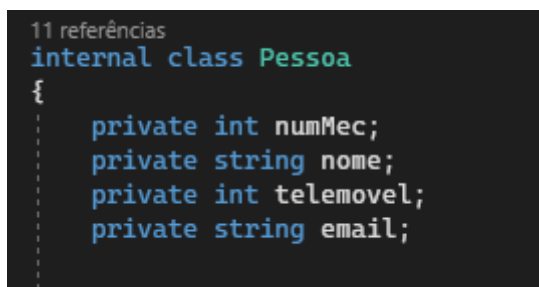


Figura 1 - Exemplo do funcionamento das conexões

2. Motivação

Nesta primeira fase do trabalho, pretende-se que sejam desenvolvidas soluções em C# para problemas reais. Neste caso foi escolhido de entre vários temas de um tema genérico «Smart Campus», onde seguem as principais classes envolvidas no mesmo:

2.1. Pessoa

A screenshot of a code editor showing the definition of the 'Pessoa' class in C#. The code is as follows:

```
11 referências
internal class Pessoa
{
    private int numMec;
    private string nome;
    private int telemovel;
    private string email;
}
```

Figura 2 - Classe Pessoa

A classe pessoa armazena a informação sobre os utilizadores da aplicação.

O numMec corresponde ao número mecanográfico do utilizador (aluno/professor)

São também armazenados o nome, telemóvel e email do utilizador na mesma classe.

2.2. Local

```
3 referências
internal class Local
{
    private string nome;
    private int codigo;
    List<Local> listaDeParagens = new List<Local>();
}
```

Figura 3 - Classe Local

A classe Local vai armazenar a informação sobre as paragens.

Tem Get e Set como propriedades para permitir que elementos públicos possam aceder aos valores armazenados na classe privada.

Tem como construtores o construtor por definição e o construtor que recebe todos os valores que correspondem às variáveis da classe Local.

2.3. Passe

```
16 referências
internal class Passe
{
    /// <summary>
    /// atributos da classe passe
    /// </summary>
    #region ATRIBUTOS
    private Usuario usuPasse; // atributo que utiliza a classe usuario para acessar os dados do passe
    private int numero;
    private int numeroAluno;
    private string nome;
    private bool status;
    private int ultimoNumero;
    private List<Passe> listaDePASSES;
}
```

Figura 4 - Classe Passe

A classe passe armazena a informação sobre os passes dos utilizadores da aplicação.

O usuPasse é o atributo que utiliza a classe usuario para aceder aos dados do passe.

Os restantes atributos (numero, numeroAluno, nome, status, ultimoNumero) são armazenados no array listaDePASSES.

2.4. Passes

```
8 referências
internal class Passes
{
    #region ATRIBUTOS
    private int idPasse;
    private List<Passes> listaPasses;
    private Passe dadosPasse;
    private int totalPasses;
    private int ultimoIdPasse;
    #endregion
}
```

Figura 5 - Classe Passes

A classe passes armazena e gera a lista de passes na aplicação.

São gerados posteriormente a lista de opções de passes na aplicação.

2.5. Usuário

```
7 referências
internal class Usuario : Pessoa
{
    #region ATRIBUTOS
    private Passe atribuirPasse;
}
```

Figura 6 - Classe Usuário

A classe Usuário é uma classe filha da classe Pessoa. Nela existe a mecânica da atribuição um passe a uma pessoa.

2.6. Program

```
0 referências
internal class Program
{
    0 referências
    static void Main(string[] args)
    {

        Usuario u1 = new Usuario(21990, "Allan", 999999999, "email@xpto");

        Passe p1 = new Passe();

        u1.AtribuirPasse = p1;
    }
}
```

Figura 7 - Classe Program

A classe program, é o nosso "main". Nela são criadas as pessoas e os passes com os seus respectivos valores. Nela também é chamado o método que aloca um passe a um utilizador.

3. Documentação de código

O nosso código foi documentado com recurso à ferramenta DoxyGen onde pode ser consultada toda a informação resumida sobre as classes e os seus respetivos métodos. Essa informação pode ser encontrada na pasta "Documentação".

BusIPCA.Local Class Reference
Classe LOCAL - Para criação das paragens More...
Public Member Functions
Local (string nome, int id) Construtor que recebe por parâmetro os valores de Local More...
Local () Construtor por defeito
Properties
string Nome [get, set]
int Codigo [get, set]
Private Attributes
string nome
int codigo
List< Local > ListaDeParagens = new List< Local >()
Detailed Description
Classe LOCAL - Para criação das paragens

4. Conclusão

Neste momento o projeto encontra-se ainda numa fase embrionária e falta ainda trabalhar melhor certas classes e a criação de novas classes que contribuem para o projeto, porém pretendemos que até à data da segunda entrega tenhamos já o código mais desenvolvido de forma a ir ao encontro do objetivo inicial do projeto.