

RELATÓRIO FINAL

Abraão Dantas - 14906

Adriano Veloso – 14815

Docente - Luís Gonzaga Martins Ferreira

Linguagem de programação II

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, 22 maio de 2020

Índice

I - Introdução.....	3
II - Linguagem C#	4
III - Objetivos do Projeto.....	5
IV - Desenvolvimento	6
4.1 - Desenvolvimento Global.....	6
4.2 - Funcionalidades desenvolvidas	6
4.2.1 – Login	6
4.2.2 – ID de Convocatória	6
4.2.3 – Verificação de disponibilidade	6
4.2.4 – Registo.....	7
4.3 - Classes	7
4.3.1 - Class Program	7
4.3.2 – Call	7
4.3.3 – User	8
V - Aprendizagem Adquirida.....	10
VI - Problemas encontrados	11
VII- Conclusao.....	12

I - Introdução

Este trabalho tem como tem como propósito o desenvolvimento de uma aplicação onde o treinador de uma equipa será capaz de gerir as convocatórias dos jogadores e que os jogadores sejam capazes de consultar as mesmas através de um sistema que possibilita a realização de um login de conta pessoal como treinador com privilégios para a alteração e consulta dos dados da convocatória e os jogadores com uma conta com a capacidade de consulta das convocatórias.

II - Linguagem C#

Desenvolvida quase do zero, esta linguagem possui um vasto leque de bibliotecas e a sua compatibilidade é enorme em relação aos sistemas operativos.

Funcionando numa lógica hierárquica, o C# constitui desde as operações básicas como a adição, multiplicação, subtração e soma às mais complicadas e únicas como encapsulamentos e sobrecargas múltiplas. No C# não existe herança múltipla, ou seja, cada classe só pode herdar uma e só uma classe, mantendo assim o código e a gestão de memória organizada. Possui também uma gestão de memória única denominada Coletor de Lixo que recicla a memória e obtém uma maior eficiência.

III - Objetivos do Projeto

Desenvolver uma aplicação capaz de fazer a gestão da convocatória dos jogadores onde o treinador tenha a capacidade de gerir os jogadores, verificar a sua disponibilidade e em que os jogadores consigam aceder a aplicação e verificar se foram ou não convocados.

Temos também como diretiva consolidar conceitos basilares do Paradigma Orientado a Objetos, melhorar a Análise de problemas reais e conseguir disponibilizar uma solução através de uma ferramenta prática e simples, com isto esperamos desenvolver capacidades de programação em C# e potenciar a experiência no desenvolvimento de software.

IV - Desenvolvimento

4.1 - Desenvolvimento Global

Ao longo do desenvolvimento deste Projeto, tivemos sempre cuidado com a apresentação do código, ou seja, deixar o código sempre organizado e devidamente comentado, mas dando sempre mais importância ao cumprimento dos requisitos essenciais.

4.2 - Funcionalidades desenvolvidas

4.2.1 – Login

A função Login é direcionada a entrada na aplicação feita pelo treinador e pelo jogador onde será feita a distinção dos mesmos para a distribuição dos privilégios no sistema como também a verificação de existência.

4.2.2 – ID de Convocatória

A funcionalidade de ID de Convocatória tem como diretiva a atribuição de um identificador único á convocatória para a distinção e organização das mesmas.

4.2.3 – Verificação de disponibilidade

Verificação de disponibilidade tem como função a elaboração de uma apuração acerca da disponibilidade dos jogadores para os jogos que se irão realizar.

4.2.4 – Registo

A funcionalidade de registo tem objetivo de armazenar os dados de cada conta existente e cada convocatória realizada para consulta futura.

4.3 - Classes

4.3.1 - Class Program

O programa é iniciado na classe Program, ou seja, é aqui que as primeiras instruções são dadas para o arranque do mesmo.

Main:

```
class Program
{
    static void Main(string[] args)
    {
        User d = new User();
        Start(d);
    }
}
```

4.3.2 – Call

Call é uma classe dedicada às convocatórias, responsável desde realizar uma convocatória, verificar se já existe uma convocatória para o dia de jogo pretendido, atribuir um valor aleatório ao ID da convocatória até apagar uma convocatória feita pelo treinador.

```
public class Call
{
    #region Var
    int GameDay, GameMonth, GameYear, GameHour, CallID;
    char GameLocation, GameOpponent;
    #endregion

    #region Properties
    public int Day { get => GameDay; set => GameDay = value; }
    public int Month { get => GameMonth; set => GameMonth = value; }
    public int Year { get => GameYear; set => GameYear = value; }
    public int Hour { get => GameHour; set => GameHour = value; }
    public char Location { get => GameLocation; set => GameLocation = value; }
    public char Opponent { get => GameOpponent; set => GameOpponent = value; }
    public int ID
    {
        get { return CallID; }
        set
        {
            if ((value.ToString()).Length == 4)
            {
                CallID = value;
            }
        }
    }
}
```

4.3.3 – User

A classe User serve para a criação de utilizadores da aplicação com conta única, esta também realiza a verificação da existência de utilizadores e futuramente os seus privilégios.


```
public class User
{
    #region Var
    string name, password;
    int UserID;

    #endregion

    #region Properties
    public string Name { get => name; set => name = value; }

    public string Password { get => password; set => password = value; }

    public int UserID { get => userid; set => userid = value; }

    #endregion

    #region Constructors
    /// <summary>
    /// Default Class constructor
    /// </summary>
    public User()
    {
        this.Name = default(string);
        this.Password = default(string);
    }

    // name = Nome do Utilizador
    // password = Password do Utilizador

    public User(string name, string password)
    {
        this.Name = name;
        this.Password = password;
    }

    #endregion
}
```

V - Aprendizagem Adquirida

O desenvolvimento deste trabalho foi rico em aprendizagem. A linguagem C# é muito vasta e complexa com imensas e diversas aplicações, por isso ao progredir, o trabalho requereu uma constante pesquisa de material complementar para ultrapassar as dificuldades que foram aparecendo. No fim deste projeto, o nosso nível de conhecimento de C# aumentou significativamente, mas também podemos dar como adquiridas certas bases da linguagem.

VI - Problemas encontrados

Não há muitas aplicações de gestão de convocatórias, visto que os treinadores fazem as mesmas através de um papel e caneta. Isto pode ser visto como um problema, pois não existe uma base para realizar um estudo de como terá que ser os requisitos necessários para uma boa funcionalidade e solução do problema, tendo também como dificuldade a implementação da funcionalidade de verificação de disponibilidade dada a sua complexidade.

VII- Conclusao

Este tabalho possibilitou a aprendizagem e aprefundamento do que foi lecionado nas aulas, promovendo a solução de problemas através da sua análise e pesquisa de soluções eficases, práticas não so por consulta de materiais existentes mas tambem reafirmando que a cooperação entre colegas com a finalidade de compartilhacao de ideias e conhecimentos pré-adquiridos é essecial para progressão nesta área.