

Licenciatura em Sistemas Multimédia

**Estruturas de Dados e Algoritmia I**

*Quem Quer Ser Fixe*

*Daniel Carneiro*

# Índice

[Índice 3](#_Toc31055498)

[1. Apresentação do sistema 5](#_Toc31055499)

[2. Estrutura do Projeto 5](#_Toc31055500)

[2.1 Jogar 5](#_Toc31055501)

[2.2 Gerir Perguntas 7](#_Toc31055502)

[2.2.1 Criar 7](#_Toc31055503)

[2.2.2 Listar/Editar 8](#_Toc31055504)

[2.2.3 Eliminar 9](#_Toc31055505)

[2.3 Pontuações 10](#_Toc31055506)

[3. Fluxograma 11](#_Toc31055507)

[4. Pseudocódigo 13](#_Toc31055508)

[5. Prova e Teste 14](#_Toc31055509)

[6. Programa em imagens 15](#_Toc31055510)

[6.1 Menu Principal 15](#_Toc31055511)

[6.2 Jogar > Perguntas 15](#_Toc31055512)

[6.3 Listar/Editar 16](#_Toc31055513)

[6.4 Listar/Editar > Editar 16](#_Toc31055514)

[6.5 Jogar > Resultados 17](#_Toc31055515)

[6.6 Gerir Perguntas 17](#_Toc31055516)

[6.7 Gerir Perguntas > Criar 18](#_Toc31055517)

[6.8 Gerir Perguntas > Criar > Escolher resposta correcta 18](#_Toc31055518)

[6.9 Gerir Perguntas > Eliminar 19](#_Toc31055519)

[6.10 Pontuações 19](#_Toc31055520)

[7. Codificação C# 20](#_Toc31055521)

[7.1 Crud.cs 20](#_Toc31055522)

[7.2 Jogar.cs 27](#_Toc31055523)

[7.3 Ler\_guardar\_service.cs 30](#_Toc31055524)

[7.4 Program.cs 33](#_Toc31055525)

[7.5 Service.cs 35](#_Toc31055526)

[7.6 Structs.cs 38](#_Toc31055527)

# 

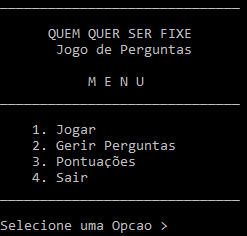
# 1. Apresentação do sistema

No âmbito da disciplina de Estruturas de Dados e Algoritmia I foram dadas algumas sugestões para a elaboração deste trabalho. O exemplo escolhido foi a elaboração de um jogo de perguntas semelhante ao Quem Quer Ser Milionário. Este jogo foi desenvolvido em C# no *Microsoft Visual Studio* e foi criado inteiramente para funcionar na consola do Windows.

# 2. Estrutura do Projeto

Entende-se por objetivo deste projeto a aplicação dos conhecimentos até a data adquiridos nas aulas, tendo isso em conta, a estrutura do projeto compreende-se por jogar, gerir as perguntas e ver as pontuações.

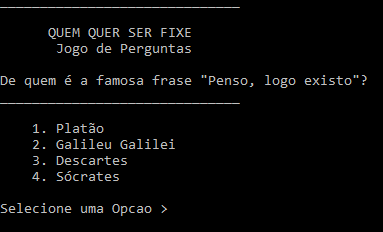
Todas as perguntas e pontuações serão guardadas em dois ficheiros Pontuacoes.txt e Perguntas.txt.



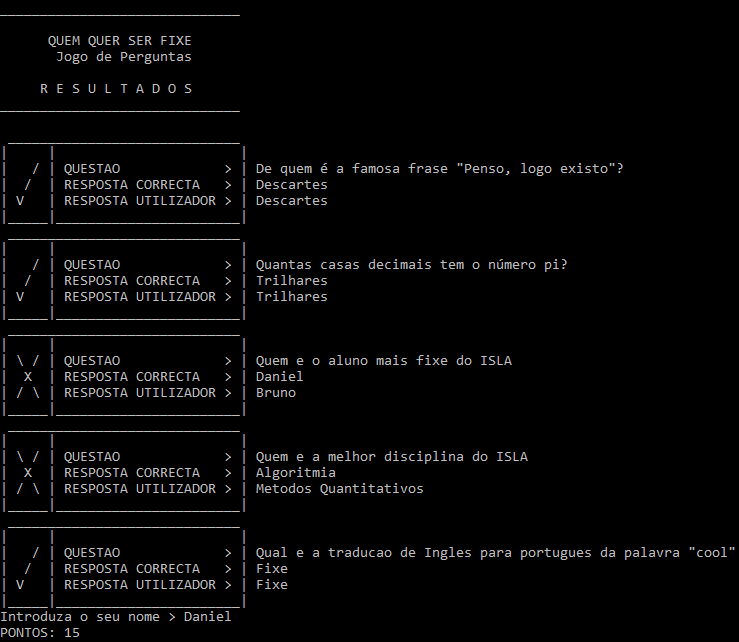
## 2.1 Jogar

Esta será a primeira opção disponibilizada ao utilizador e permite ao mesmo jogar com as perguntas registadas. No final será apresentada a pontuação dependente do número de perguntas certas.

Dependendo do número de perguntas que estão inseridas no jogo vai decidir quantas perguntas estão disponibilizadas para o utilizador. Cada pergunta é apresentada e obrigatório de escolher uma opção. Após selecionada a opção será redirecionado para a pergunta seguinte.



No final, quando todas as perguntas forem respondidas serão apresentados os resultados com as respetivas respostas corretas:

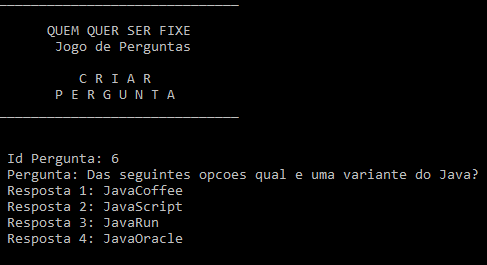


Será também requerido o nome do utilizador para guardar a pontuação na tabela e apresentados os pontos.

## 2.2 Gerir Perguntas

Para que seja possível jogar é necessário que existam perguntas, como tal foi criada uma opção para gerir todas as perguntas. Nesta opção é possível:

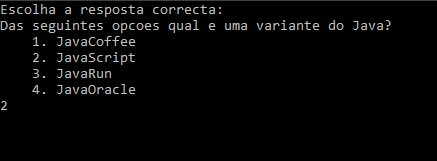
### 2.2.1 Criar



(Um id de pergunta será gerado automaticamente e será apresentado ao utilizador todos os campos necessários para criar uma pergunta)

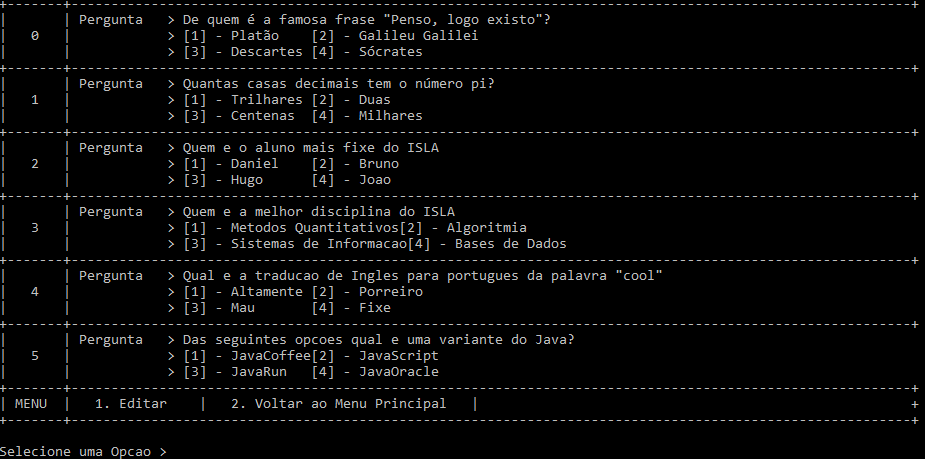
* Pergunta – A nova pergunta
* Resposta 1: A primeira resposta
* Resposta 2: A segunda resposta
* Resposta 3: A terceira resposta
* Resposta 4: A quarta resposta

Após o utilizador escrever todas as respostas será apresentada a pergunta e as respetivas respostas e este terá de escrever o número da resposta que considere correta:



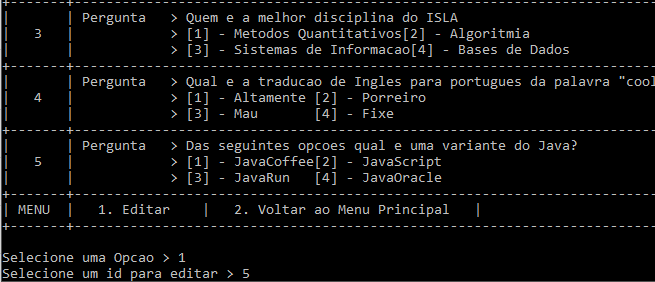
Depois de escolher a resposta esta estará criada e será adicionada ao jogo.

### 2.2.2 Listar/Editar

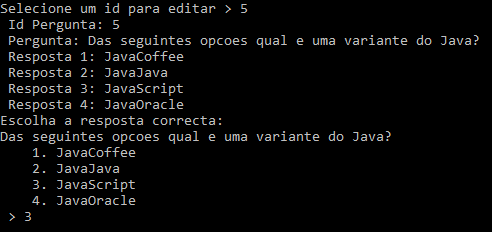
Se o utilizador desejar ver ou editar perguntas este será o menu correcto:  
  
No final da tabela será apresentado duas opções:

1. **Editar**

Após o utilizador selecionar a opção 1 vai ser requerido o id da pergunta que deseja editar:



Será apresentada a opção para editar as perguntas e respostas e selecionar a resposta correta:



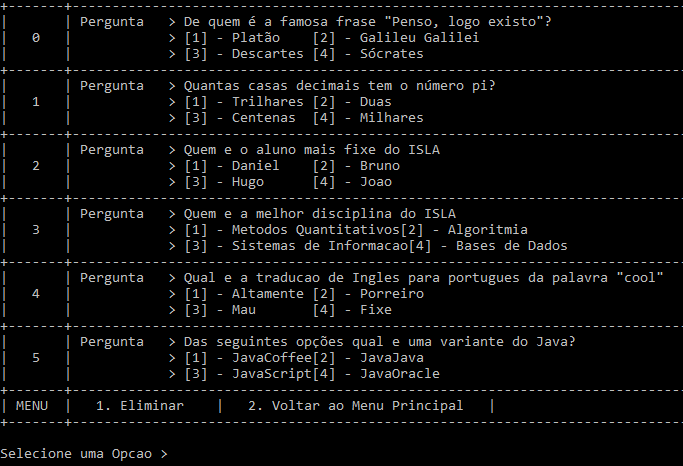
Após concluído será redirecionado para o menu principal.

1. **Voltar ao Menu Principal**

Esta opção foi adicionada para que se o utilizador apenas quiser ver a lista de perguntas não tenha de as editar.

### 2.2.3 Eliminar

No caso de o utilizador querer remover do jogo alguma pergunta isso será possível fazer nesta opção:

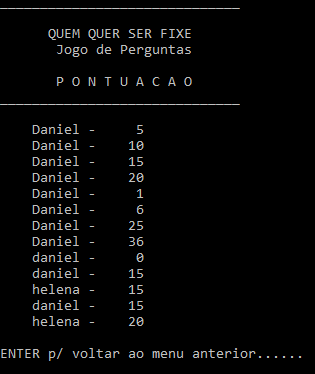


Caso o utilizador selecionar a opção 1 será requerido o id que deseja remover e esta pergunta será retirada do jogo.

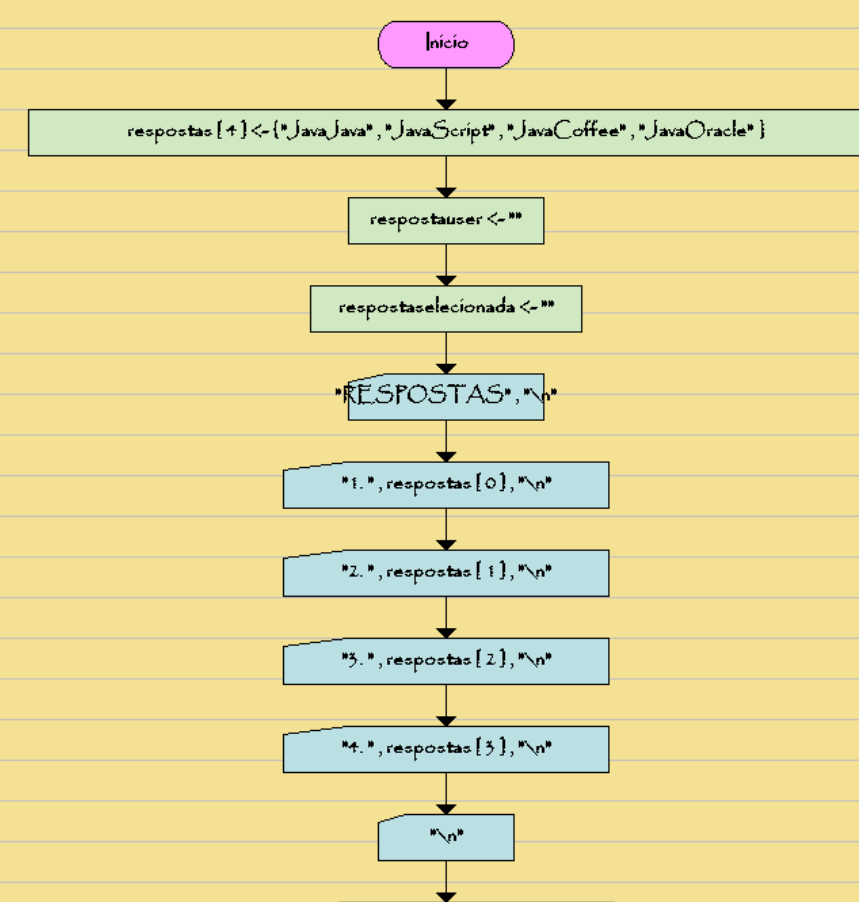
## 2.3 Pontuações

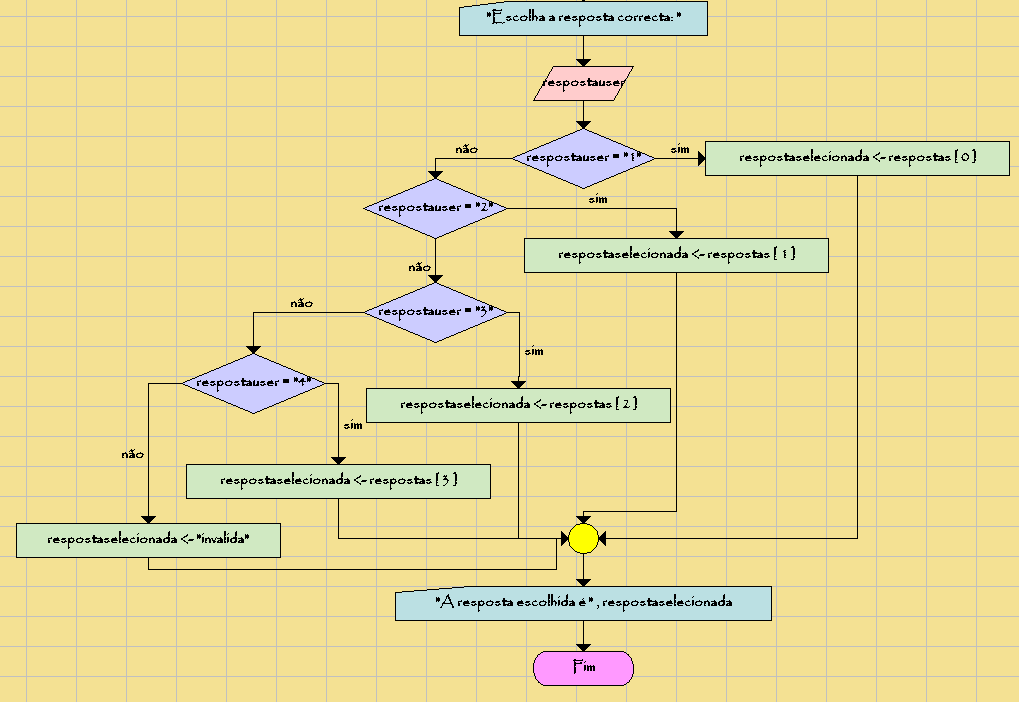
Para que o jogo se torne mais interessante foi adicionado um sistema de pontos. Após cada jogo serão atribuídos pontos ao utilizador dependendo das respostas corretas. Os pontos são calculados da seguinte forma:

* **Numero de respostas corretas \* numero total de respostas**



# 3. Fluxograma





# 4. Pseudocódigo

**inicio**

**constante** **texto** respostas[4] <- {"JavaJava" , "JavaScript" , "JavaCoffee" , "JavaOracle" }

**variavel** **texto** respostauser

**variavel** **texto** respostaSelecionada

**escrever** "RESPOSTAS", "\n"

**escrever** "1. ", respostas[0], "\n"

**escrever** "2. ", respostas[1], "\n"

**escrever** "3. ", respostas[2], "\n"

**escrever** "4. ", respostas[3], "\n"

**escrever** "\n"

**escrever** "Escolha a resposta correcta: "

**ler** respostauser

**escolhe** respostauser

**caso** "1":

respostaSelecionada <- respostas[0]

**caso** "2":

respostaSelecionada <- respostas[1]

**caso** "3":

respostaSelecionada <- respostas[2]

**caso** "4":

respostaSelecionada <- respostas[3]

**defeito**:

respostaSelecionada <- "invalida"

**fimescolhe**

**escrever** "A resposta escolhida é ", respostaSelecionada

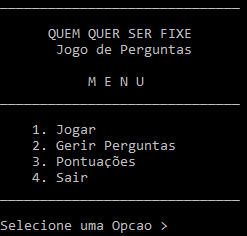
**fim**

# 5. Prova e Teste

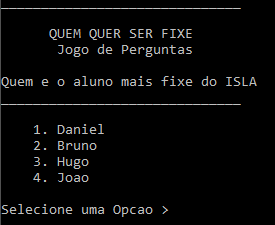
|  |  |  |  |
| --- | --- | --- | --- |
| Teste | RespostaUser | RespostaSelecionada | Resultado |
| 1 | 1 | JavaJava | A resposta escolhida é JavaJava |
| 2 | 2 | JavaScript | A resposta escolhida é JavaScript |
| 3 | 3 | JavaCoffee | A resposta escolhida é JavaCoffee |
| 4 | 4 | JavaOracle | A resposta escolhida é JavaOracle |
| 5 | 5 | invalida | A resposta escolhida é invalida |
| 6 | 0 | invalida | A resposta escolhida é invalida |
| 7 | aaa | invalida | A resposta escolhida é invalida |
| 8 | A11 | invalida | A resposta escolhida é invalida |
| 9 | 1a | invalida | A resposta escolhida é invalida |

# 6. Programa em imagens

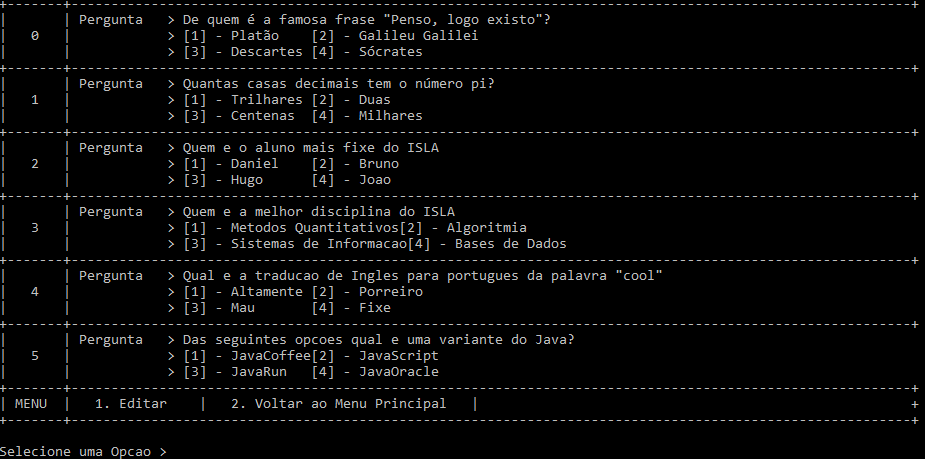
## 6.1 Menu Principal



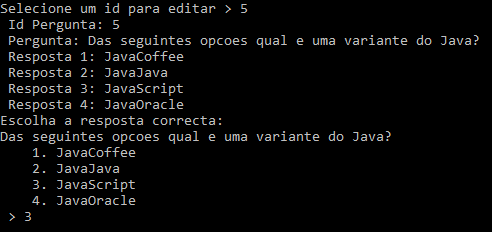
## 6.2 Jogar > Perguntas



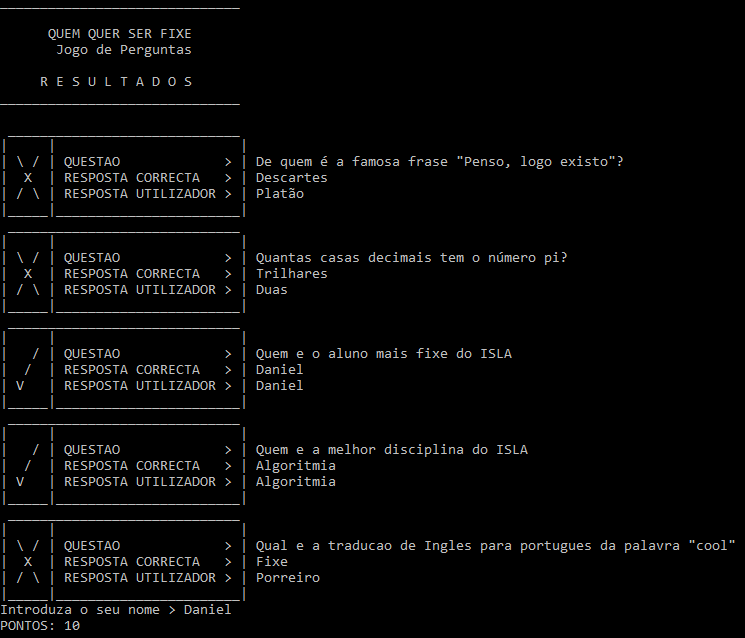
## 6.3 Listar/Editar



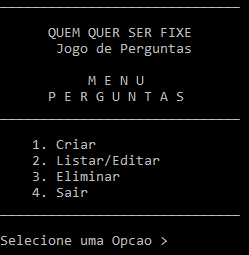
## 6.4 Listar/Editar > Editar



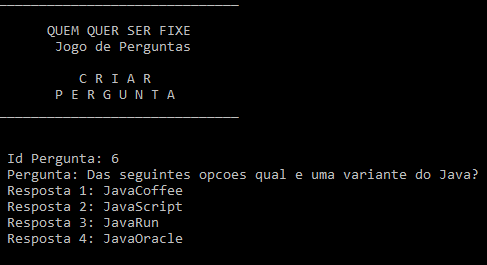
## 6.5 Jogar > Resultados



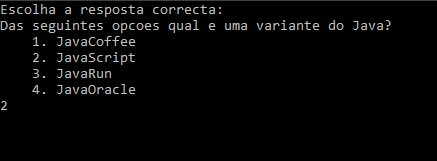
## 6.6 Gerir Perguntas



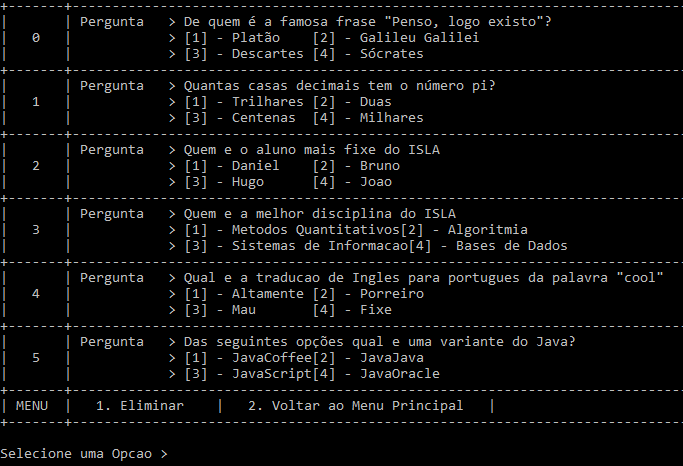
## 6.7 Gerir Perguntas > Criar



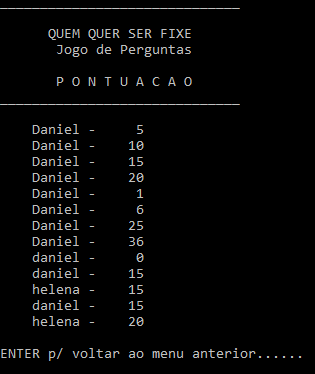
## 6.8 Gerir Perguntas > Criar > Escolher resposta correcta



## 6.9 Gerir Perguntas > Eliminar



## 6.10 Pontuações



# 7. Codificação C#

Para a codificação em c# ser mais legível, dividi em vários ficheiros.

## 7.1 Crud.cs

using Quem\_Quer\_Ser\_Fixe.service;

using System;

using System.IO;

using static Quem\_Quer\_Ser\_Fixe.structs.structs;

namespace Quem\_Quer\_Ser\_Fixe

{

    public static class Crud

    {

        /// <summary>

        /// Função que lista todas as perguntas do array quiz

        /// </summary>

        public static void VerPerguntas()

        {

            var space = "";

            for (int i = 0; i < Program.quiz.Length; i++)

            {

                Console.WriteLine("+-------+---------------------------------------------------------------------------------------------------------+");

                Console.WriteLine("|" + space.PadRight(7) + "| Pergunta   > " + Program.quiz[i].questao);

                Console.WriteLine("|" + space.PadRight(3) + Program.quiz[i].id + space.PadRight(3) + "|            > " + "[1] - " + Program.quiz[i].resposta.resposta1.PadRight(10) + "[2] - " + Program.quiz[i].resposta.resposta2);

                Console.WriteLine("|" + space.PadRight(7) + "|            > " + "[3] - " + Program.quiz[i].resposta.resposta3.PadRight(10) + "[4] - " + Program.quiz[i].resposta.resposta4);

                if (Program.quiz.Length == i + 1)

                {

                    Console.WriteLine("+-------+---------------------------------------------------------------------------------------------------------+");

                }

            }

        }

        /// <summary>

        /// Função que adiciona o menu e dá opção para editar uma pergunta

        /// </summary>

        public static void EditarPerguntas()

        {

            Console.WriteLine("| MENU  |   1. Editar    |   2. Voltar ao Menu Principal   |                                                      +");

            Console.WriteLine("+-------+---------------------------------------------------------------------------------------------------------+");

            Console.WriteLine("");

            Console.Write("Selecione uma Opcao > ");

            int opcao = int.Parse(Console.ReadLine());

            switch (opcao)

            {

                case 1:

                    Editar();

                    break;

                case 2:

                    Console.Clear();

                    Program.MainMenu();

                    break;

                default:

                    Console.Clear();

                    Program.MainMenu();

                    break;

            }

        }

        /// <summary>

        /// Função que edita uma pergunta por quiz.questaoId

        /// </summary>

        private static void Editar()

        {

            Console.Write("Selecione um id para editar > ");

            int questaoId = int.Parse(Console.ReadLine());

            Boolean encontrado;

            for (int i = 0; i < Program.quiz.Length; i++)

            {

                if (Program.quiz[i].id == questaoId)

                {

                    encontrado = true;

                    Console.WriteLine(" Id Pergunta: " + Program.quiz[i].id);

                    Program.quiz[i].id = Program.quiz[i].id;

                    Console.Write(" Pergunta: ");

                    Program.quiz[i].questao = Console.ReadLine();

                    Console.Write(" Resposta 1: ");

                    Program.quiz[i].resposta.resposta1 = Console.ReadLine();

                    Console.Write(" Resposta 2: ");

                    Program.quiz[i].resposta.resposta2 = Console.ReadLine();

                    Console.Write(" Resposta 3: ");

                    Program.quiz[i].resposta.resposta3 = Console.ReadLine();

                    Console.Write(" Resposta 4: ");

                    Program.quiz[i].resposta.resposta4 = Console.ReadLine();

                    Console.WriteLine("Escolha a resposta correcta:");

                    Console.WriteLine(Program.quiz[i].questao);

                    Console.WriteLine("    1. " + Program.quiz[i].resposta.resposta1);

                    Console.WriteLine("    2. " + Program.quiz[i].resposta.resposta2);

                    Console.WriteLine("    3. " + Program.quiz[i].resposta.resposta3);

                    Console.WriteLine("    4. " + Program.quiz[i].resposta.resposta4);

                    int respostaCorrecta = 0;

                    do

                    {

                        Console.Write(" > ");

                        respostaCorrecta = int.Parse(Console.ReadLine());

                        switch (respostaCorrecta)

                        {

                            case 1:

                                Program.quiz[i].resposta.respostaCorrectaNome = Program.quiz[i].resposta.resposta1;

                                Program.quiz[i].resposta.respostaCorrectaId = "1";

                                break;

                            case 2:

                                Program.quiz[i].resposta.respostaCorrectaNome = Program.quiz[i].resposta.resposta2;

                                Program.quiz[i].resposta.respostaCorrectaId = "2";

                                break;

                            case 3:

                                Program.quiz[i].resposta.respostaCorrectaNome = Program.quiz[i].resposta.resposta3;

                                Program.quiz[i].resposta.respostaCorrectaId = "3";

                                break;

                            case 4:

                                Program.quiz[i].resposta.respostaCorrectaNome = Program.quiz[i].resposta.resposta4;

                                Program.quiz[i].resposta.respostaCorrectaId = "4";

                                break;

                        }

                    } while (respostaCorrecta > 4 || respostaCorrecta <= 0);

                    if (!encontrado)

                    {

                        Console.WriteLine("Id Nao encontrado, tente novamente...");

                        EditarPerguntas();

                        return;

                    }

                    ler\_guardar\_service.Guardar();

                }

            }

            ;

        }

        /// <summary>

        /// Função que cria uma pergunta e guarda a mesma num ficheiro txt

        /// </summary>

        public static void CriarPergunta()

        {

            Service.Header("          C R I A R          \n       P E R G U N T A       ");

            Console.WriteLine();

            questoes[] novaQuestao = new questoes[1];

            //ler dados até que o utilizador introduza o numero 0

            int num;

            num = Program.quiz.Length + 1;

            Console.WriteLine(" Id Pergunta: " + num);

            novaQuestao[0].id = num - 1;

            Console.Write(" Pergunta: ");

            novaQuestao[0].questao = Console.ReadLine();

            Console.Write(" Resposta 1: ");

            novaQuestao[0].resposta.resposta1 = Console.ReadLine();

            Console.Write(" Resposta 2: ");

            novaQuestao[0].resposta.resposta2 = Console.ReadLine();

            Console.Write(" Resposta 3: ");

            novaQuestao[0].resposta.resposta3 = Console.ReadLine();

            Console.Write(" Resposta 4: ");

            novaQuestao[0].resposta.resposta4 = Console.ReadLine();

            Console.Clear();

            Console.WriteLine("Escolha a resposta correcta:");

            Console.WriteLine(novaQuestao[0].questao);

            Console.WriteLine("    1. " + novaQuestao[0].resposta.resposta1);

            Console.WriteLine("    2. " + novaQuestao[0].resposta.resposta2);

            Console.WriteLine("    3. " + novaQuestao[0].resposta.resposta3);

            Console.WriteLine("    4. " + novaQuestao[0].resposta.resposta4);

            int respostaCorrecta = 0;

            do

            {

                respostaCorrecta = int.Parse(Console.ReadLine());

                switch (respostaCorrecta)

                {

                    case 1:

                        novaQuestao[0].resposta.respostaCorrectaNome = novaQuestao[0].resposta.resposta1;

                        novaQuestao[0].resposta.respostaCorrectaId = "1";

                        break;

                    case 2:

                        novaQuestao[0].resposta.respostaCorrectaNome = novaQuestao[0].resposta.resposta2;

                        novaQuestao[0].resposta.respostaCorrectaId = "2";

                        break;

                    case 3:

                        novaQuestao[0].resposta.respostaCorrectaNome = novaQuestao[0].resposta.resposta3;

                        novaQuestao[0].resposta.respostaCorrectaId = "3";

                        break;

                    case 4:

                        novaQuestao[0].resposta.respostaCorrectaNome = novaQuestao[0].resposta.resposta4;

                        novaQuestao[0].resposta.respostaCorrectaId = "4";

                        break;

                }

            } while (respostaCorrecta > 4 || respostaCorrecta <= 0);

            guardarPergunta(novaQuestao);

            Console.WriteLine("Pergunta criada!");

            Console.WriteLine();

            Service.Pausa("voltar ao menu principal...");

            Console.Clear();

            Program.MainMenu();

        }

        /// <summary>

        /// Recebe a nova pergunta e adiciona ao array, depois guarda no ficheiro txt

        /// </summary>

        /// <param name="novaQuestao"></param>

        public static void guardarPergunta(questoes[] novaQuestao)

        {

            int index = novaQuestao[0].id;

            Array.Resize(ref Program.quiz, index + 1);

            StreamWriter FicheiroEscrita = new StreamWriter(Program.ficheiroPerguntas);

            Program.quiz[index].id = index;

            Program.quiz[index].questao = novaQuestao[0].questao;

            Program.quiz[index].resposta.resposta1 = novaQuestao[0].resposta.resposta1;

            Program.quiz[index].resposta.resposta2 = novaQuestao[0].resposta.resposta2;

            Program.quiz[index].resposta.resposta3 = novaQuestao[0].resposta.resposta3;

            Program.quiz[index].resposta.resposta4 = novaQuestao[0].resposta.resposta4;

            Program.quiz[index].resposta.respostaCorrectaId = novaQuestao[0].resposta.respostaCorrectaId;

            Program.quiz[index].resposta.respostaCorrectaNome = novaQuestao[0].resposta.respostaCorrectaNome;

            for (int i = 0; i < Program.quiz.Length; i++)

            {

                FicheiroEscrita.Write(

                    Program.quiz[i].id + ";" +

                    Program.quiz[i].questao + ";" +

                    Program.quiz[i].resposta.respostaCorrectaId + ";" +

                    Program.quiz[i].resposta.respostaCorrectaNome + ";" +

                    Program.quiz[i].resposta.resposta1 + ";" +

                    Program.quiz[i].resposta.resposta2 + ";" +

                    Program.quiz[i].resposta.resposta3 + ";" +

                    Program.quiz[i].resposta.resposta4 + "\n"

                    );

            }

            FicheiroEscrita.Close();

        }

        /// <summary>

        /// Adiciona o menu de opções para remover a pergunta e guarda o array no ficheiro txt

        /// </summary>

        public static void eliminar()

        {

            Crud.VerPerguntas();

            Console.WriteLine("| MENU  |   1. Eliminar    |   2. Voltar ao Menu Principal   |                                                    +");

            Console.WriteLine("+-------+---------------------------------------------------------------------------------------------------------+");

            Console.WriteLine("");

            Console.Write("Selecione uma Opcao > ");

            int opcao = int.Parse(Console.ReadLine());

            switch (opcao)

            {

                case 1:

                    Console.Write("Selecione um id para remover > ");

                    int idQuestao = int.Parse(Console.ReadLine());

                    for (int i = 0; i < Program.quiz.Length; i++)

                    {

                        if (Program.quiz[i].id == idQuestao)

                        {

                            Program.quiz = Service.RemoverDoArray(Program.quiz, i);

                        }

                    }

                    ler\_guardar\_service.Guardar();

                    Program.MainMenu();

                    break;

                case 2:

                    Program.MainMenu();

                    break;

                default:

                    Program.MainMenu();

                    break;

            }

        }

    }

}

## 7.2 Jogar.cs

using Quem\_Quer\_Ser\_Fixe.service;

using System;

using static Quem\_Quer\_Ser\_Fixe.structs.structs;

namespace Quem\_Quer\_Ser\_Fixe.jogar

{

    public static class Jogar

    {

        /// <summary>

        /// Abre o menu de gestao de perguntas

        /// </summary>

        /// <param name="questoes"></param>

        public static void MenuPerguntas(ref questoes[] questoes)

        {

            Console.Clear();

            Service.Header("           M E N U            \n      P E R G U N T A S       ");

            Console.WriteLine("    1. Criar                  ");

            Console.WriteLine("    2. Listar/Editar          ");

            Console.WriteLine("    3. Eliminar               ");

            Console.WriteLine("    4. Sair                   ");

            Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

            Console.WriteLine("");

            Console.Write("Selecione uma Opcao > ");

            int opcao = int.Parse(Console.ReadLine());

            switch (opcao)

            {

                case 1:

                    Console.Clear();

                    Crud.CriarPergunta();

                    Service.Pausa("continuar");

                    Console.Clear();

                    Program.MainMenu();

                    break;

                case 2:

                    Crud.VerPerguntas();

                    Crud.EditarPerguntas();

                    Service.Pausa("continuar");

                    Console.Clear();

                    Program.MainMenu();

                    break;

                case 3:

                    Console.Clear();

                    Crud.eliminar();

                    Console.Clear();

                    break;

                case 4:

                    Console.Clear();

                    Program.MainMenu();

                    break;

                default:

                    Console.WriteLine("Opcao seleccionada nao esta disponivel, prima enter para continuar...");

                    Console.ReadKey();

                    Console.Clear();

                    Program.MainMenu();

                    break;

            }

        }

        /// <summary>

        /// Mostra os resultados de todas as perguntas respondidas pelo user

        /// </summary>

        public static void mostrarResultados()

        {

            int pontos = 0;

            Service.Header("     R E S U L T A D O S      ");

            for (int i = 0; i < Program.quiz.Length; i++)

            {

                string respostaUser = obterResposta(Program.quiz[i]);

                if (Program.quiz[i].resposta.respostaUser == Program.quiz[i].resposta.respostaCorrectaId)

                {

                    Console.WriteLine(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

                    Console.WriteLine("|     |                       | ");

                    Console.WriteLine("|   / | QUESTAO             > | " + Program.quiz[i].questao);

                    Console.WriteLine("|  /  | RESPOSTA CORRECTA   > | " + Program.quiz[i].resposta.respostaCorrectaNome);

                    Console.WriteLine("| V   | RESPOSTA UTILIZADOR > | " + respostaUser);

                    Console.WriteLine("|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| ");

                    pontos = pontos + 1;

                }

                else

                {

                    Console.WriteLine(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

                    Console.WriteLine("|     |                       | ");

                    Console.WriteLine("| \\ / | QUESTAO             > | " + Program.quiz[i].questao);

                    Console.WriteLine("|  X  | RESPOSTA CORRECTA   > | " + Program.quiz[i].resposta.respostaCorrectaNome);

                    Console.WriteLine("| / \\ | RESPOSTA UTILIZADOR > | " + respostaUser);

                    Console.WriteLine("|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| ");

                }

            }

            Console.Write("Introduza o seu nome > ");

            string userName = Console.ReadLine();

            Console.WriteLine("PONTOS: " + (pontos \* Program.quiz.Length));

            ler\_guardar\_service.guardarPontuacao(pontos \* Program.quiz.Length, userName);

        }

        /// <summary>

        /// Recebe uma pergunta e retorna resposta do user

        /// </summary>

        /// <param name="questoes"></param>

        /// <returns></returns>

        public static string obterResposta(questoes questao)

        {

            string resposta;

            switch (questao.resposta.respostaUser)

            {

                case "1":

                    resposta = questao.resposta.resposta1;

                    break;

                case "2":

                    resposta = questao.resposta.resposta2;

                    break;

                case "3":

                    resposta = questao.resposta.resposta3;

                    break;

                case "4":

                    resposta = questao.resposta.resposta4;

                    break;

                default:

                    resposta = "Resposta Invalida";

                    break;

            }

            return resposta;

        }

        /// <summary>

        /// Função que mostra a pergunta e todas as respostas

        /// </summary>

        /// <param name="questao"></param>

        public static void FazerPergunta(questoes questao)

        {

            Service.Header(questao.questao);

            //respostas

            Console.WriteLine("    1. " + questao.resposta.resposta1);

            Console.WriteLine("    2. " + questao.resposta.resposta2);

            Console.WriteLine("    3. " + questao.resposta.resposta3);

            Console.WriteLine("    4. " + questao.resposta.resposta4);

            Console.WriteLine();

            Console.Write("Selecione uma Opcao > ");

        }

    }

}

## 7.3 Ler\_guardar\_service.cs

using System;

using System.IO;

using static Quem\_Quer\_Ser\_Fixe.structs.structs;

namespace Quem\_Quer\_Ser\_Fixe

{

    public static class ler\_guardar\_service

    {

        /// <summary>

        /// Função que recebe o nome do ficheiro e guarda ou em pontuações ou em perguntas

        /// </summary>

        /// <param name="nomeFicheiro"></param>

        public static void LerFicheiro(string nomeFicheiro)

        {

            string NomeFic = nomeFicheiro;

            StreamReader FicheiroLeitura = new StreamReader(NomeFic);

            string linha;

            int i = 0;

            if (nomeFicheiro == Program.ficheiroPerguntas)

            {

                while (!FicheiroLeitura.EndOfStream)

                {

                    linha = FicheiroLeitura.ReadLine();

                    string[] palavras = linha.Split(';');

                    Array.Resize(ref Program.quiz, i + 1);

                    Program.quiz[i].id = int.Parse(palavras[0]);

                    Program.quiz[i].questao = palavras[1];

                    Program.quiz[i].resposta.respostaCorrectaId = palavras[2];

                    Program.quiz[i].resposta.respostaCorrectaNome = palavras[3];

                    Program.quiz[i].resposta.resposta1 = palavras[4];

                    Program.quiz[i].resposta.resposta2 = palavras[5];

                    Program.quiz[i].resposta.resposta3 = palavras[6];

                    Program.quiz[i].resposta.resposta4 = palavras[7];

                    i++;

                }

            }

            if (nomeFicheiro == Program.ficheiroPontuacoes)

            {

                while (!FicheiroLeitura.EndOfStream)

                {

                    linha = FicheiroLeitura.ReadLine();

                    string[] palavras = linha.Split(';');

                    Array.Resize(ref Program.pontuacao, i + 1);

                    Program.pontuacao[i].userName = palavras[0];

                    Program.pontuacao[i].pontos = palavras[1];

                    i++;

                }

            }

            FicheiroLeitura.Close();

        }

        /// <summary>

        /// Guarda a pontuação no ficheiro e no array pontuacoes

        /// </summary>

        /// <param name="pontos"></param>

        /// <param name="userName"></param>

        public static void guardarPontuacao(int pontos, string userName)

        {

            Program.pontuacao = new pontuacoes[0];

            StreamReader FicheiroLeitura = new StreamReader(Program.ficheiroPontuacoes);

            string linha = FicheiroLeitura.ReadLine();

            int i = 0;

            while (!FicheiroLeitura.EndOfStream)

            {

                linha = FicheiroLeitura.ReadLine();

                string[] palavras = linha.Split(';');

                Array.Resize(ref Program.pontuacao, i + 1);

                Program.pontuacao[i].userName = palavras[0];

                Program.pontuacao[i].pontos = palavras[1];

                i++;

            }

            FicheiroLeitura.Close();

            int index = Program.pontuacao.Length + 1;

            StreamWriter FicheiroEscrita = new StreamWriter(Program.ficheiroPontuacoes);

            Array.Resize(ref Program.pontuacao, index);

            Program.pontuacao[index - 1].userName = userName;

            Program.pontuacao[index - 1].pontos = "" + pontos + "";

            FicheiroEscrita.WriteLine("nome;pontos");

            for (int a = 0; a < Program.pontuacao.Length; a++)

            {

                FicheiroEscrita.WriteLine(Program.pontuacao[a].userName + ";" + Program.pontuacao[a].pontos);

            }

            FicheiroEscrita.Close();

        }

        /// <summary>

        /// Função que guarda as perguntas existentes no array para o ficheiro txt

        /// </summary>

        public static void Guardar()

        {

            StreamWriter FicheiroEscrita = new StreamWriter(Program.ficheiroPerguntas);

            for (int i = 0; i < Program.quiz.Length; i++)

            {

                FicheiroEscrita.Write(

                    Program.quiz[i].id + ";" +

                    Program.quiz[i].questao + ";" +

                    Program.quiz[i].resposta.respostaCorrectaId + ";" +

                    Program.quiz[i].resposta.respostaCorrectaNome + ";" +

                    Program.quiz[i].resposta.resposta1 + ";" +

                    Program.quiz[i].resposta.resposta2 + ";" +

                    Program.quiz[i].resposta.resposta3 + ";" +

                    Program.quiz[i].resposta.resposta4 + ";" + "\n"

                    );

            }

            FicheiroEscrita.Close();

        }

    }

}

## 7.4 Program.cs

using System;

using System.IO;

using static Quem\_Quer\_Ser\_Fixe.structs.structs;

using Quem\_Quer\_Ser\_Fixe.service;

using Quem\_Quer\_Ser\_Fixe.jogar;

namespace Quem\_Quer\_Ser\_Fixe

{

    public class Program

    {

        public static int numRegisto = 0;

        public static string ficheiroPerguntas = "perguntas.txt";

        public static string ficheiroPontuacoes = "pontuacoes.txt";

        public static questoes[] quiz = new questoes[0];

        public static pontuacoes[] pontuacao = new pontuacoes[0];

        /// <summary>

        /// Função principal do projecto

        /// </summary>

        /// <param name="args"></param>

        static void Main(string[] args)

        {

            ler\_guardar\_service.LerFicheiro(ficheiroPerguntas);

            ler\_guardar\_service.LerFicheiro(ficheiroPontuacoes);

            MainMenu();

        }

        /// <summary>

        /// Função que mostra o menu principal do projecto

        /// </summary>

        public static void MainMenu()

        {

            Console.Clear();

            Service.Header("           M E N U            ");

            Console.WriteLine("    1. Jogar                  ");

            Console.WriteLine("    2. Gerir Perguntas        ");

            Console.WriteLine("    3. Pontuações             ");

            Console.WriteLine("    4. Sair                   ");

            Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

            Console.WriteLine("");

            Console.Write("Selecione uma Opcao > ");

            int opcao = int.Parse(Console.ReadLine());

            switch (opcao)

            {

                case 1:

                    jogar();

                    break;

                case 2:

                    gerirPerguntas();

                    break;

                case 3:

                    pontuacoes();

                    MainMenu();

                    break;

                case 4:

                    Console.Clear();

                    MainMenu();

                    break;

                default:

                    Console.WriteLine("Opcao seleccionada nao esta disponivel, prima enter para continuar...");

                    Console.ReadKey();

                    Console.Clear();

                    MainMenu();

                    break;

            }

        }

        /// <summary>

        /// Mostra todas as pontuações dos users

        /// </summary>

        private static void pontuacoes()

        {

            Console.Clear();

            Service.Header("       P O N T U A C A O      ");

            for (int i = 1; i < pontuacao.Length; i++)

            {

                Console.WriteLine("    " + pontuacao[i].userName + " - " + pontuacao[i].pontos.PadLeft(5));

            }

            Service.Pausa("voltar ao menu anterior...");

            Console.Clear();

        }

        /// <summary>

        /// Mostra o menu das perguntas

        /// </summary>

        private static void gerirPerguntas()

        {

            questoes[] questoes = new questoes[0];

            Jogar.MenuPerguntas(ref questoes);

            Console.Clear();

        }

        /// <summary>

        /// Função que mostra o quiz e depois os resultados

        /// </summary>

        private static void jogar()

        {

            Console.Clear();

            Service.Quiz();

            Jogar.mostrarResultados();

            Service.Pausa("continuar");

            MainMenu();

        }

    }

}

## 7.5 Service.cs

using Quem\_Quer\_Ser\_Fixe.jogar;

using System;

using static Quem\_Quer\_Ser\_Fixe.structs.structs;

namespace Quem\_Quer\_Ser\_Fixe.service

{

    public static class Service

    {

        /// <summary>

        /// Para o programa até uma proxima interacção do user

        /// </summary>

        /// <param name="texto"></param>

        public static void Pausa(string texto)

        {

            Console.WriteLine();

            Console.Write("ENTER p/ " + texto + "...");

            Console.ReadKey();

        }

        /// <summary>

        /// função cria o cabeçalho a partir de um titulo recebido

        /// </summary>

        /// <param name="titulo"></param>

        public static void Header(string titulo)

        {

            Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

            Console.WriteLine("                              ");

            Console.WriteLine("      QUEM QUER SER FIXE      ");

            Console.WriteLine("       Jogo de Perguntas      ");

            Console.WriteLine("                              ");

            Console.WriteLine(titulo);

            Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

            Console.WriteLine("                              ");

        }

        /// <summary>

        ///  Recebe o array e o indice que deseja remover. Remove o id selecionado e retorna o array sem esse id

        /// </summary>

        /// <param name="IndicesArray"></param>

        /// <param name="IndiceARemover"></param>

        /// <returns></returns>

        public static questoes[] RemoverDoArray(questoes[] IndicesArray, int IndiceARemover)

        {

            questoes[] novoArrayIndices = new questoes[IndicesArray.Length - 1];

            int i = 0;

            int a = 0;

            while (i < IndicesArray.Length)

            {

                if (i != IndiceARemover)

                {

                    novoArrayIndices[a] = IndicesArray[i];

                    a++;

                }

                i++;

            }

            return novoArrayIndices;

        }

        /// <summary>

        /// Função que mostra as perguntas quando o user inicia o jogo

        /// </summary>

        public static void Quiz()

        {

            for (int i = 0; i < Program.quiz.Length; i++)

            {

                Jogar.FazerPergunta(Program.quiz[i]);

                string respostaUser = Console.ReadLine();

                Program.quiz[i].resposta.respostaUser = respostaUser;

                if (Program.quiz[i].resposta.respostaUser == Program.quiz[i].resposta.respostaCorrectaId)

                {

                    Console.WriteLine("A resposta " + Program.quiz[i].resposta.respostaUser + " esta correcta!");

                }

                else

                {

                    Console.WriteLine("A resposta esta incorrecta!");

                }

                Service.Pausa("continuar");

                Console.Clear();

            }

        }

    }

}

## 7.6 Structs.cs

namespace Quem\_Quer\_Ser\_Fixe.structs

{

    public class structs

    {

       public struct questoes

        {

            public int id;

            public string questao;

            public respostas resposta;

        }

        public struct respostas

        {

            public string respostaUser;

            public string respostaCorrectaId;

            public string respostaCorrectaNome;

            public string resposta1;

            public string resposta2;

            public string resposta3;

            public string resposta4;

        }

        public struct pontuacoes

        {

            public string userName;

            public string pontos;

        }

    }

}