

Ficha de Trabalho 1: Resolução

1. Resolução:

```
public struct CodigoPostal
{
    public int Parte1;
    public int Parte2;
    public string DesignacaoPostal;

    public CodigoPostal(int parte1, int parte2, string designacaoPostal)
    {
        Parte1 = parte1;
        Parte2 = parte2;
        DesignacaoPostal = designacaoPostal;
    }

    public override string ToString()
    {
        return $"{Parte1}-{Parte2} {DesignacaoPostal.ToUpper()}";
    }
}
```

```
CodigoPostal c1 = new CodigoPostal(3000, 175, "coimbra");
Console.WriteLine(c1.ToString());
```

2. Resolução:

```
public struct CartaoCidadao
{
    public string Numero;
    public int CheckDigit1;
    public char[] Versao;
    public int CheckDigit2;

    public CartaoCidadao(string numero, int checkDigit1, char versao1, char versao2,
int checkDigit2)
    {
        Numero = numero;
        CheckDigit1 = checkDigit1;
        Versao = new char[2];
        Versao[0] = versao1;
        Versao[1] = versao2;
        CheckDigit2 = checkDigit2;
    }

    public override string ToString()
    {
        return $"{Numero} {CheckDigit1} {Versao[0]}{Versao[1]}{CheckDigit2}";
    }
}
```

```
CartaoCidadao c1 = new CartaoCidadao("00000000", 0, 'Z', 'Z', 4);
Console.WriteLine(c1.ToString());
```

3. Resolução:

```
// a)
public struct RegistoLimpeza
{
    public DateTime DataHora;
    public string Nome;
    public bool Reposicao;
    public string Observacoes;
}

// b)
public struct Dimensao
{
    public double Altura;
    public double Largura;
    public double Profundidade;

    public Dimensao(double altura, double largura, double profundidade)
    {
        Altura = altura;
        Largura = largura;
        Profundidade = profundidade;
    }

    public override string ToString()
    {
        return $"{Altura} cm x {Largura} cm x {Profundidade} cm";
    }
}

// c)
public struct Livro
{
    public string Titulo;
    public string Autor;
    public string Isbn;
    public string Editora;
    public int AnoEdicao;
    public int NumeroPaginas;
    public string Sinopse;
}

// d)
public struct Microondas
{
    public string Marca;
    public string Modelo;
    public int Potencia;
    public Dimensao Dimensoes;
    public int Capacidade;
    public string Cor;
    public decimal Preco;
    public double Peso;
    public bool PratoRotativo;
    public bool Ecra;
    public bool ModoEco;
    public bool BloqueioSeguranca;
```

```
public override string ToString()
{
    return $"Micro-ondas {Marca} {Modelo}\nPotência: {Potencia} W\nDimensões:
{Dimensoes.ToString()}\nCapacidade: {Capacidade} L\nCor: {Cor}\nPreço: {Preco}
€\nPeso: {Peso} Kg\nPrato rotativo: {(PratoRotativo == true) ? "Sim" :
"Não"}\nEcrã: { (Ecra == true) ? "Sim" : "Não"}\nModo eco: {(ModoEco == true) ?
"Sim" : "Não"}\nBloqueio de segurança: {(BloqueioSeguranca == true) ? "Sim" :
"Não"}";
}
```

```
// a) Registo de limpeza
```

```
RegistoLimpeza registo1;
registo1.DataHora = DateTime.Now;
registo1.Nome = "Fernando Gonçalves";
registo1.Reposicao = true;
registo1.Observacoes = string.Empty;
```

```
// b) Dimensões de produto
```

```
Dimensao d1 = new Dimensao(177, 54, 54);
Console.WriteLine($" \nDimensões do produto: {d1.ToString()}");
```

```
// c) Livro
```

```
Livro l1;
l1.Titulo = "Introdução ao Desenvolvimento de Jogos com Unity";
l1.Autor = "Alberto Simões";
l1.Isbn = "9789727228836";
l1.Editora = "FCA";
l1.AnoEdicao = 2017;
l1.NumeroPaginas = 224;
l1.Sinopse = "Atualmente, com o grande salto de qualidade dos smartphones, o
desenvolvimento de jogos cresceu enormemente e muitos programadores resolveram
aventurar-se na criação de jogos simples, que muitas vezes se revelam grandes
sucessos. Para facilitar o desenvolvimento de jogos, surgiu o Unity 3D, um motor de
jogo genérico mas cheio de potencialidades, que é para muitos a ferramenta mais
poderosa na criação de jogos de todos os tipos e para diferentes plataformas. Com o
objetivo de dar a conhecer as potencialidades desta poderosa ferramenta,
apresentamos este livro. Baseado num projeto prático, o leitor é conduzido numa
viagem ao desenvolvimento de um jogo completo, desde a modelação de um terreno,
passando pela inclusão de uma personagem animada dos seus inimigos, até à interface
gráfica. Cada capítulo implementa diferentes tipos de funcionalidades, abordando
diversas técnicas úteis no desenvolvimento de jogos com Unity. A obra encontra-se
organizada em torno de um único projeto, com recurso à linguagem de programação C#.
Cada capítulo apresenta um conjunto de funcionalidades independentes, que podem ser
experimentadas para aferir a compreensão dos tópicos, e de novas técnicas para a
implementação das funcionalidades necessárias ao funcionamento do jogo, que o leitor
pode utilizar. Trata-se de um livro introdutório, útil não só a programadores
experientes que pretendam ambientar-se à ferramenta e métodos de desenvolvimento de
jogos, mas também aos estudantes e curiosos que pretendam aprender a programar
usando a motivação pessoal de desenvolver um jogo";
```

```
// d) Micro-ondas
```

```
Microondas m1;
m1.Modelo = "1848FL";
m1.Marca = "Flama";
m1.Potencia = 800;
m1.Dimensoes.Altura = 33D;
m1.Dimensoes.Largura = 53.6D;
```

Ficha de Trabalho 1: Resolução

```
m1.Dimensoes.Profundidade = 43.6D;  
m1.Capacidade = 23;  
m1.Cor = "Preto";  
m1.Preco = 119.99M;  
m1.Peso = 14.9D;  
m1.PratoRotativo = true;  
m1.Ecra = false;  
m1.ModoEco = false;  
m1.BloqueioSeguranca = true;  
Console.WriteLine(m1.ToString());
```

4. Resolução:

```
public enum TipoQuarto  
{  
    Nulo = 0,  
    Standard,  
    Familiar,  
    Suite,  
    SuiteSuperior  
}  
  
public struct Estadia  
{  
    public DateTime CheckIn;  
    public DateTime CheckOut;  
    public string Nome;  
    public string CartaoCredito;  
    public TipoQuarto Tipo;  
  
    public int TotalNoites()  
    {  
        return (CheckOut - CheckIn).Days;  
    }  
  
    public decimal CalcularCusto()  
    {  
        decimal custoNoite;  
  
        switch (Tipo)  
        {  
            case TipoQuarto.Standard:  
                custoNoite = 40;  
                break;  
  
            case TipoQuarto.Familiar:  
                custoNoite = 50;  
                break;  
  
            case TipoQuarto.Suite:  
                custoNoite = 110;  
                break;  
  
            case TipoQuarto.SuiteSuperior:  
                custoNoite = 150;  
                break;  
  
            default:
```

Ficha de Trabalho 1: Resolução

```
        custoNoite = 0;
        break;
    }

    return custoNoite * TotalNoites() * 1.06M;
}
}
```

```
Estadia estadia1;
```

```
estadia1.CheckIn = DateTime.Parse("2019-12-01");
estadia1.CheckOut = DateTime.Parse("2019-12-07");
estadia1.Nome = "Maria Tavares";
estadia1.CartaoCredito = "1234567890";
estadia1.Tipo = TipoQuarto.SuiteSuperior;
```

```
Console.WriteLine($"{ "\nCusto estadia: {estadia1.CalcularCusto()} €");
```

5. Resolução:

```
public struct aluno
{
    public string Nome;
    public int NumeroProcesso;
    public int NumeroAluno;
    public int ClassifPortugues;
    public int ClassifMatematica;
    public int ClassifTic;

    public aluno(string nome, int numeroProcesso, int numeroAluno, int
classifPortugues, int classifMatematica, int classifTic)
    {
        Nome = nome;
        NumeroProcesso = numProcesso;
        NumeroAluno = numAluno;
        ClassifPortugues = classifPortugues;
        ClassifMatematica = classifMatematica;
        ClassifTic = classifTic;
    }
}
```

```
Random rnd = new Random();
aluno[] turma = new aluno[20];
string[] nomes = { "Leonilde", "Leonor", "Matias", "Liliana", "Maurício", "Miguel",
"Nelson", "Alexandra", "Adriano", "Afonso", "Rosário", "Sónia", "Patrícia",
"Sebastião", "Ruben", "Cláudia" };

// Popular o array
for (int i = 0; i < turma.Length; ++i)
{
    turma[i].Nome = nomes[rnd.Next(0, nomes.Length)];
    turma[i].NumeroProcesso = i + 1200;
    turma[i].NumeroAluno = i + 1;
    turma[i].ClassifPortugues = rnd.Next(1, 21);
    turma[i].ClassifMatematica = rnd.Next(1, 21);
    turma[i].ClassifTic = rnd.Next(1, 21);
}
```

```
// Mostrar os dados
Console.WriteLine("Dados da turma\n-----");
Console.Write("Nº");
Console.SetCursorPosition(4, 2);
Console.Write("Nº Processo");
Console.SetCursorPosition(17, 2);
Console.Write("Nome");
Console.SetCursorPosition(30, 2);
Console.Write("Português");
Console.SetCursorPosition(41, 2);
Console.Write("Matemática");
Console.SetCursorPosition(53, 2);
Console.Write("TIC");

for (int i = 0; i < turma.Length; ++i)
{
    Console.SetCursorPosition(0, i + 3);
    Console.Write(turma[i].NumeroAluno);

    Console.SetCursorPosition(4, i + 3);
    Console.Write(turma[i].NumeroProcesso);

    Console.SetCursorPosition(17, i + 3);
    Console.Write(turma[i].Nome);

    Console.SetCursorPosition(30, i + 3);
    Console.Write(turma[i].ClassifPortugues);

    Console.SetCursorPosition(41, i + 3);
    Console.Write(turma[i].ClassifMatematica);

    Console.SetCursorPosition(53, i + 3);
    Console.Write(turma[i].ClassifTic);
}

// Determinar os dados
double mediaPortugues = 0, mediaMatematica = 0, mediaTic = 0;
int classifMaisBaixaPortugues = turma[0].ClassifPortugues, classifMaisAltaPortugues = turma[0].ClassifPortugues;
int classifMaisBaixaMatematica = turma[0].ClassifMatematica, classifMaisAltaMatematica = turma[0].ClassifMatematica;
int classifMaisBaixaTic = turma[0].ClassifTic, classifMaisAltaTic = turma[0].ClassifTic;

for (int i = 0; i < turma.Length; ++i)
{
    // Classificações mais baixas/mais altas de Português
    if (turma[i].ClassifPortugues < classifMaisBaixaPortugues)
    {
        classifMaisBaixaPortugues = turma[i].ClassifPortugues;
    }

    if (turma[i].ClassifPortugues > classifMaisAltaPortugues)
    {
        classifMaisAltaPortugues = turma[i].ClassifPortugues;
    }

    // Classificações mais baixas/mais altas de Matemática
    if (turma[i].ClassifMatematica < classifMaisBaixaMatematica)
```

```
{
    classifMaisBaixaMatematica = turma[i].ClassifMatematica;
}

if (turma[i].ClassifMatematica > classifMaisAltaMatematica)
{
    classifMaisAltaMatematica = turma[i].ClassifMatematica;
}

// Classificações mais baixas/mais altas de TIC
if (turma[i].ClassifTic < classifMaisBaixaTic)
{
    classifMaisBaixaTic = turma[i].ClassifTic;
}

if (turma[i].ClassifTic > classifMaisAltaTic)
{
    classifMaisAltaTic = turma[i].ClassifTic;
}

// Médias
mediaPortugues += turma[i].ClassifPortugues;
mediaMatematica += turma[i].ClassifMatematica;
mediaTic += turma[i].ClassifTic;
}

Console.WriteLine("\n\nClassificação mais alta:");
Console.SetCursorPosition(30, 24);
Console.Write(classifMaisAltaPortugues);
Console.SetCursorPosition(41, 24);
Console.Write(classifMaisAltaMatematica);
Console.SetCursorPosition(53, 24);
Console.Write(classifMaisAltaTic);

Console.SetCursorPosition(0, 25);
Console.WriteLine("Classificação mais baixa:");
Console.SetCursorPosition(30, 25);
Console.Write(classifMaisBaixaPortugues);
Console.SetCursorPosition(41, 25);
Console.Write(classifMaisBaixaMatematica);
Console.SetCursorPosition(53, 25);
Console.Write(classifMaisBaixaTic);

mediaPortugues = mediaPortugues / turma.Length;
mediaMatematica = mediaMatematica / turma.Length;
mediaTic = mediaTic / turma.Length;

Console.SetCursorPosition(0, 27);
Console.WriteLine($"Média de Português: {mediaPortugues} valores");
Console.WriteLine($"Média de Matemática: {mediaMatematica} valores");
Console.WriteLine($"Média de TIC: {mediaTic} valores");
```