# Integração do C# com o Excel

## ? EXERCÍCIO 2: CRIAÇÃO DE FICHEIRO DO EXCEL

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO2
{public class CriaFicheiroExcel
 {static void Main(string[] args)
   {object Mv=Missing.Value;
    Excelapp Excel = new Excelapp();
    Excel.Visible = true;
    Excel.Workbooks.Add(Mv);
    Range R = Excel.ActiveCell;
    for (int x=1; x<=12; x++)
     { R.Cells[x, 1] = x;
       R.Cells[x, 2] = x*10+5;}
    try
    {Excel.ActiveWorkbook.SaveAs("c:\\IO2.xlsx", Mv, Mv, Mv, Mv,
    Mv,Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlNoChan
    ge,Mv, Mv, Mv, Mv, Mv);}
    catch {};
    Excel.Quit();}}}
```

## ? EXERCÍCIO 3: INSTANCIAÇÃO DE UM MODELO DO EXCEL

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO3
{public class InstanciaTemplate
  {static void Main(string[] args)
   {object mv = Missing.Value;
    Excelapp Excel = new Excelapp();
    Excel.Visible = true;
    Excel.Workbooks.Add("C:\\Teste.xltx");
```

```
     Worksheet Ws = (Worksheet)
     Excel.ActiveWorkbook.Worksheets.get_Item(1);
     Range R = Ws.get_Range("D2", "D8");
     DateTime D = System.DateTime.Now;
     for (int x = 1; x <= R.Rows.Count; x++)
      {R.Cells[x, 1] = D;
       D=D.AddDays(7);}
     Excel.DisplayAlerts = false;
     Excel.ActiveWorkbook.SaveAs("c:\\IO3.xlsx", mv,
mv, mv, mv, mv,
Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlNoChange,
mv, mv, mv, mv, mv);
Excel.Quit();}}}
```

## ? EXERCÍCIO 4: ABERTURA DE FICHEIRO DO EXCEL

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO4
{public class AberturaFicheiro
 {static void Main(string[] args)
  {Excelapp Excel = new Excelapp();
   object Mv = Missing.Value;
   Excel.Visible = true;
   string Pasta = "C:\\";
   string Fich = "IO4.xlsx";
   Excel.Workbooks.Open(Pasta+Fich, Mv, false, Mv, Mv, Mv, Mv,
   Mv, Mv,Mv, Mv, Mv, Mv, Mv, Mv);}}}
```

## ? EXERCÍCIO 5: GRAVAÇÃO DE FICHEIRO COM PALAVRA-CHAVE

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO5
{public class GravacaoComPalavraChave
 {static void Main(string[] args)
```

```
  {const string PalChave = "Porto";
   const string Ficheiro = "c:\\IO5.xlsx";
   Excelapp Excel = new Excelapp();
   object Mv = Missing.Value;
   Excel.Visible = true;
   Excel.Workbooks.Open(Ficheiro, Mv, false, Mv, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv, Mv, Mv);
   Worksheet Ws =(Worksheet)
   Excel.ActiveWorkbook.Worksheets.get_Item(1);
   MessageBox.Show("A proteger com palavra-chave");
   Ws.Protect(PalChave, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv);
   Excel.ActiveWorkbook.Save();
   Excel.Quit();}}}
```

## ? EXERCÍCIO 6: FORMATAÇÃO DE CÉLULAS

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO6
{public class PintaInferiores
 {static void Main(string[] args)
  {Excelapp Excel = new Excelapp();
   object Mv = Missing.Value;
   Excel.Visible = true;
   string Ficheiro = "c:\\IO6.xlsx";
   Excel.Workbooks.Open(Ficheiro, Mv, false, Mv, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv, Mv, Mv);
   Worksheet Ws =
   (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
   int Numero, T=0, L=14;
   string EndR = "A1:H20";
   Range R = Ws.get_Range(EndR, Mv);
   R.Interior.ColorIndex = -4142;
   Range Xend = Ws.get_Range("A1", Mv);
   foreach (Range C in R)
    {Numero = Convert.ToInt32(C.Value2);
     if (Numero < L)
      {C.Interior.ColorIndex = 35;
       T++;}}
   MessageBox.Show( T +" células foram pintadas");
   Excel.ActiveWorkbook.Save();
```

```
    Excel.Quit();}}}
```

## ? EXERCÍCIO 7: FÓRMULAS PARA *RANGES*

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO7
{public class FormulasParaRanges
 {static void Main(string[] args)
  {const string Ficheiro = "c:\\IO7.xlsx";
   const string EndR = "A1:B10";
   int[] Quant={10,10,15,5,12,10,20,40, 70,100};
   double[] Pu={2,2.5,1.5,5,10,2,2.5,4.2, 7,10};
   object Mv=Missing.Value;
   Excelapp Excel = new Excelapp();
   Excel.Visible = true;
   Excel.Workbooks.Add(Mv);
   Worksheet Ws =
   (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
   Range R = Ws.get_Range(EndR, Mv);
   for (int I = 0; I <= R.Rows.Count-1; I++ )
    {R.Cells[I+1,1] = Quant[I];
     R.Cells[I+1, 2] = Pu[I];}
   Range Valores = Ws.get_Range("C1", "C10");
   foreach (Range V in Valores)
    V.Formula = "=" +
                V.get_Offset(0,-2).get_Address(Mv, Mv,
                XlReferenceStyle.xlA1, Mv, Mv)+"*"+
                V.get_Offset(0,-1).get_Address(Mv, Mv,
                XlReferenceStyle.xlA1, Mv, Mv);
   MessageBox.Show("Continuamos?");
   try
   {Excel.ActiveWorkbook.SaveAs(Ficheiro, Mv, Mv, Mv, Mv,
   Mv,Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlNoChang
   e,Mv, Mv, Mv, Mv, Mv);}
   catch {};
   Excel.Quit();}}}
```

## ? EXERCÍCIO 8: ESCONDER COLUNAS

```
using System;
```

```
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO8
{public class EsconderEprotegerColunas
 {static void Main(string[] args)
  {const string AreaDeTrabalho = "A1:H20";
   const string EndR = "D1:D1";
   Excelapp Excel = new Excelapp();
   object Mv = Missing.Value;
   Excel.Visible = true;
   string Ficheiro = "c:\\IO8.xlsx";
   string PalChave="Porto";
   Excel.Workbooks.Open(Ficheiro, Mv, false, Mv, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv, Mv, Mv);
   Worksheet Ws =
   (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
   try
    {Range R = Ws.get_Range(AreaDeTrabalho, Mv);
     R.Locked = false;
     R = Ws.get_Range(EndR, Mv);
     if ((bool)(R.EntireColumn.Hidden) == false)
     R.EntireColumn.Hidden = true;
     string Desigcol = ((char)(64 + R.Column)).ToString();
     MessageBox.Show("Área de trabalho desprotegida e coluna " +
     Desigcol + " escondida");}
   catch
     {MessageBox.Show("Área de trabalho já estava desprotegida e
     a coluna já estava escondida "); }
   Ws.Protect(PalChave, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv);
   Excel.ActiveWorkbook.Save();
   Excel.Quit();}}}
```

## ? EXERCÍCIO 9: FOLHA DE RESPOSTAS

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO9
{public class FolhaDeRepostas
 {public static void EscreverCabecalho(Worksheet Ws, object Mv)
```

```
  {Range R = Ws.get_Range("A1:C2", Mv);
   R.Cells[1, 1] = "Escreva aqui o seu nome";
   R.Cells[2,1] = "Número da Pergunta";
   R.Cells[2,2] = "A sua Resposta";
   R.Cells[2,3] = "Resposta Correta";
   for (int I=1; I<=R.Columns.Count; I++)
    R.EntireColumn.AutoFit();}

  public static void EscreverPerguntas(Worksheet Ws, object Mv)
   {string EndR="A3:A13";
    Range R = Ws.get_Range(EndR, Mv);
    for (int I = 1; I <= 10; I++)
     R.Cells[I, 1] = "Q" + I.ToString();}

  public static void EscreverRespostas(Worksheet Ws, object Mv)
   {string EndR="C3:C13";
    string[] Resp = {"a","c","d","a", "b","a","c","b","a", "d"} ;
    Range R = Ws.get_Range(EndR, Mv);
    for (int I = 1; I <= 10; I++)
     R.Cells[I, 1] = Resp[I-1]; }

  public static void EsconderRespostas(Worksheet Ws, object Mv)
   {const string AreaLivre="A1:B13";
    const string PalChave="Porto";
    try
     {string EndR = "C1:C1";
      Range R = Ws.get_Range(AreaLivre, Mv);
      R.Locked = false;
      R = Ws.get_Range(EndR, Mv);
      if ((bool)(R.EntireColumn.Hidden) == false)
       R.EntireColumn.Hidden = true;}
    catch
     {}
    Ws.Protect(PalChave, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv,
    Mv, Mv, Mv, Mv, Mv);}

static void Main(string[] args)
 {string Ficheiro = "C:\\FolhaRespostas";
  object Mv = Missing.Value;
  Excelapp Excel = new Excelapp();
  Excel.Visible = true;
  Excel.Workbooks.Add(Mv);
  Worksheet Ws =
  (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
  EscreverCabecalho(Ws, Mv);
  EscreverPerguntas(Ws, Mv);
  EscreverRespostas(Ws, Mv);
  EsconderRespostas(Ws, Mv);
  MessageBox.Show("A folha de respostas está preparada");
```

```
  Excel.ActiveWorkbook.SaveAs(Ficheiro, Mv, Mv, Mv,
  Mv,Mv,Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlNoCha
  nge, Mv, Mv, Mv, Mv, Mv);
  Excel.Quit();}}}
```

## ? EXERCÍCIO 10: CORREÇÃO AUTOMÁTICA

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO10
{public class CorrecaoAutomatica
 {public static void Corrigir(Worksheet Ws, object Mv)
  {string EndR = "B3:B12";
   Range R = Ws.get_Range(EndR, Mv);
   int Classif=0;
   string Res="";
   string ResC="";
   foreach (Range c in R)
   {Res=c.Value2.ToString();
   ResC=c.get_Offset(0,1).Value2.ToString();
   if (Res.CompareTo(ResC)==0)
   {Classif++;}}
   Ws.get_Range("B1:B1",Mv).Value2 = Classif;
   MessageBox.Show(Classif.ToString());}

 public static void DesprotegerEMostrar(Worksheet Ws, object Mv)
 {const string PalChave="Porto";
 string EndR = "C1:C1";
 Ws.Unprotect(PalChave);
 Range R = Ws.get_Range(EndR, Mv);
 R.EntireColumn.Hidden = false;}

 static void Main(string[] args)
  {const string Pasta = "C:\\" ;
   string Fich="Resp1.xlsx";
   object Mv = Missing.Value;
   Excelapp Excel = new Excelapp();
   Excel.Visible = true;
   Excel.Workbooks.Add(Mv);
   Excel.Workbooks.Open(Pasta + Fich, Mv, false, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv);
   Worksheet Ws =
   (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
```

```
    Corrigir(Ws, Mv);
    DesprotegerEMostrar(Ws, Mv);
    Fich= "C:\\"+Ws.get_Range("A1:A1", Mv).Value2;
    MessageBox.Show("Continuamos?");
    Excel.DisplayAlerts = false;
    Excel.ActiveWorkbook.SaveAs(Fich, Mv, Mv, Mv, Mv, Mv,
    Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlNoChange,
    Mv, Mv, Mv, Mv, Mv);
    Excel.Quit();}}}
```

## ? EXERCÍCIO 11: MÚLTIPLOS DE TRÊS E DE CINCO

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO11
{public class Multiplos3E5
 {static void Main(string[] args)
 {Excelapp excel = new Excelapp();
  excel.Visible = true;
  excel.Workbooks.Add(Missing.Value);
  Range R = excel.ActiveCell;
  int N = 2475;
  int I = 1, J = 1;
  for (int X = 3; X <= N; X++)
   {if (I > 15 )
    {I = 1;
     J++;}
   if (X%3 == 0 && X%5==0)
    {R.Cells[I, J] = X;
     I++;}}
  try
   {excel.ActiveWorkbook.Save(); }
  catch { }
  excel.Quit();}}}
```

## ? EXERCÍCIO 12: NÚMEROS PRIMOS

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
```

```
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO3
{public class PrimosEmRange
 {static bool Primo(int X)
  {bool P = true;
   int div = 2;
   while ((P == true) && (div <= (int)Math.Sqrt(X)))
    {if (X % div == 0)
     P = false;
     else
      div++;}
  return P;}
 static void Main(string[] args)
  {Excelapp excel = new Excelapp();
   excel.Visible = true;
   excel.Workbooks.Add(Missing.Value);
   Range R = excel.ActiveCell;
   int N = 500;
   int I = 1, J = 1;
   for (int X = 1; X <= N; X++)
    {if (I % 10 == 0)
     {I = 1;
      J++;}
     if (Primo(X) == true)
      {R.Cells[I, J] = X;
      I++;
     }}
     try
     {excel.ActiveWorkbook.Save();}
   catch {};
   excel.Quit();}}}
```

## ? EXERCÍCIO 13: *RANGE* PARA VETOR

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO13
{public class RangeParaVector
 {static void Main(string[] args)
  {object Mv = Missing.Value;
```

```
    Excelapp Excel = new Excelapp();
    Excel.Visible = true;
    const string Ficheiro="c:\\IO13.xlsx";
    Excel.Workbooks.Open(Ficheiro, Mv, false, Mv, Mv, Mv, Mv, Mv,
    Mv, Mv, Mv, Mv, Mv, Mv, Mv);
    Worksheet Ws =
    (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
    string EndR = "A1:A6";
    Range R = Ws.get_Range(EndR, Mv);
    string[] Nomes =new string[R.Rows.Count];
    int I = 0;
    foreach (Range C in R)
     {string N = C.Value2.ToString();
      Nomes[I] = N;
      I++;}
    string TSNomes = "";
    foreach (string N in Nomes)
     TSNomes += N+"\n";
    MessageBox.Show(TSNomes,"Nomes");
    Excel.Quit();}}}
```

## ? EXERCÍCIO 14: *RANGE* PARA MATRIZ

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO14
{public class RangeParaMatriz
 {static void Main(string[] args)
  {const string Ficheiro = "c:\\IO14.xlsx";
   const string EndR = "A1:C6";
   object Mv = Missing.Value;
   Excelapp Excel = new Excelapp();
   Excel.Visible = true;
   Excel.Workbooks.Open(Ficheiro, Mv, false, Mv, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv, Mv, Mv);
   Worksheet Ws =
   (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
   Range R = Ws.get_Range(EndR, Mv);
   string[,] Nomes = new string[R.Rows.Count, R.Columns.Count];
   int I = 0, J=0;
   foreach (Range C in R)
    {if (J==Nomes.GetLength(1))
```

```
      {I=I+1;
       J=0;}
      string N = C.Value2.ToString();
      Nomes[I,J] = N;
      J++;}
    string TSNomes = "";
    for (I = 0; I <= Nomes.GetLength(0) - 1;I++ )
     {for (J = 0; J <= Nomes.GetLength(1) - 1; J++)
      {Nomes[I,J]=Nomes[I, J]+"\t\t";
       TSNomes += Nomes[I, J];}
       TSNomes+="\n";}
    MessageBox.Show(TSNomes,"Nomes");
    Excel.Quit();}}}
```

## ? EXERCÍCIO 15: *RANGE* PARA INSTANCIAÇÃO DE OBJETO

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO15
{public class Alunos
 {private string Nome;
  private double T1;
  private double T2;
  public Alunos()
    {}
  public Alunos(Range R)
   {Range N = (Range) R.get_Item(1,1);
    Nome= N.Value2.ToString() ;
    T1= (double)N.get_Offset(0,1).Value2;
    T2 =(double)N.get_Offset(0,2).Value2;}
  public string PNome
   {get
    {return Nome;}}
  public double Notafinal()
    {return Math.Round((T1+T2)/2,0,
                       System.MidpointRounding.AwayFromZero);}}

 public class RangeParaObjeto
  {static void Main(string[] args)
    {const string Ficheiro = "C:\\IO15.xlsx";
     const string EndR= "A1";
     object Mv = Missing.Value;
```

```
     Excelapp Excel = new Excelapp();
     Excel.Visible = true;
     Excel.Workbooks.Open(Ficheiro, Mv, false, Mv, Mv, Mv, Mv,
     Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv);
     Worksheet Ws =
     (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
     Range R = Ws.get_Range(EndR, Mv);
     Alunos A= new Alunos(R);
     MessageBox.Show(A.PNome + " --- " + A.Notafinal(),
                  "Nota Final");
    Excel.Quit();}}}
```

## ? EXERCÍCIO 16: *RANGE* PARA VETOR DE OBJETOS

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO16
{public class Alunos
 {private string Nome;
 private double T1;
 private double T2;
 public Alunos(Range R)
  {Range N = (Range) R.get_Item(1,1);
   Nome= N.Value2.ToString() ;
   T1= Convert.ToDouble(N.get_Offset(0,1).Value2);
   T2 = Convert.ToDouble(N.get_Offset(0,2).Value2);}
 public string PNome
  {get
   {return Nome;}}
 public double Notafinal()
  {return Math.Round((T1 + T2)/2,0,
  System.MidpointRounding.AwayFromZero);}}

 public class VetorDeObjetos
 {static void Main(string[] args)
  {const string Ficheiro = "c:\\IO16.xlsx";
   const string EndR = "A1:A5";
   object Mv = Missing.Value;
   Excelapp Excel = new Excelapp();
   Excel.Visible = true;
   Excel.Workbooks.Open(Ficheiro, Mv, false, Mv, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv);
```

```
    Worksheet Ws =
    (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
    Range R = Ws.get_Range(EndR, Mv);
    Alunos[] A= new Alunos[R.Rows.Count];
    int I = 0;
    string TSPauta = "Aluno" + "\t" + "Nota".PadLeft(18)+"\n";
    foreach (Range C in R)
     {A[I] = new Alunos(C);
      TSPauta+=A[I].PNome+"\t"+
      A[I].Notafinal().ToString().PadLeft(5)+"\n";
      I++;}
    MessageBox.Show(TSPauta, "Pauta");
    Excel.Quit();}}}
```

## ? EXERCÍCIO 17: DE INTERFACE GRÁFICA PARA UM *RANGE*

```
using System;
using Microsoft.Office.Core;
using Interf=System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO17
{public class FormParaLerUmNome: Interf.Form
 {private System.Windows.Forms.Label Etiqueta;
  private System.Windows.Forms.TextBox Texto;
  private System.Windows.Forms.Button Botao;
  public static string Lnome;
 public FormParaLerUmNome()
  {this.Etiqueta = new System.Windows.Forms.Label();
   this.Texto = new System.Windows.Forms.TextBox();
   this.Botao = new System.Windows.Forms.Button();
   Etiqueta.Text = "Digite o seu nome ";
   Etiqueta.Location = new System.Drawing.Point(20, 24);
   Etiqueta.Size = new System.Drawing.Size(300, 40);
   Texto.Location = new System.Drawing.Point(20, 70);
   Texto.Size = new System.Drawing.Size(200, 0);
   Botao.Location = new System.Drawing.Point(70, 120);
   Botao.Size = new System.Drawing.Size(90, 32);
   Botao.Text = "OK";
   Botao.Click += new System.EventHandler(this.Botao_Click);
   this.ClientSize = new System.Drawing.Size(240, 180);
   this.Controls.Add(this.Botao);
   this.Controls.Add(this.Etiqueta);
   this.Controls.Add(this.Texto);}
```

```
protected void Botao_Click(object sender, System.EventArgs e)
 {Lnome = Texto.Text;
  this.Close();}}

public class DeInterfaceParaRange
 {public static void Main()
  {object Mv = Missing.Value;
   Excelapp Excel = new Excelapp();
   Excel.Visible = true;
   Excel.Workbooks.Add(Mv);
   Worksheet Ws =
   (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
   Range R = Ws.get_Range("a1:a1", Mv);
   for (int x = 1; x <= 5; x++)
   { FormParaLerUmNome F = new FormParaLerUmNome();
   Interf.Application.Run(F);
   R.Cells[x, 1] = FormParaLerUmNome.Lnome;}
   Interf.Application.Exit();}}}
```

## ? EXERCÍCIO 18: TESTES ALEATÓRIOS

```
using System;
using Microsoft.Office.Core;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using Excelapp = Microsoft.Office.Interop.Excel.Application;
using Missing = System.Reflection.Missing;
namespace IO18
{public class GeraTestes

 {public static void LerTeste(Excelapp Excel, object Mv,
  string[,] Perg)
  {string Fich = "C:\\Teste.xlsx";
   Excel.Workbooks.Open(Fich, Mv, false, Mv, Mv, Mv, Mv, Mv, Mv,
   Mv, Mv, Mv, Mv, Mv, Mv);
   Worksheet Ws =
   (Worksheet)Excel.ActiveWorkbook.Worksheets.get_Item(1);
   Range R = Ws.get_Range("A1:F10", Mv);
   int I = 0;
   int J = 1;
   foreach (Range C in R)
    {if (J == Perg.GetLength(1))
     {I = I + 1;
      J = 1;}
     Perg[I, J] = C.Value2.ToString();
     J++;}
```

```
    Excel.ActiveWorkbook.Close(Mv, Fich, Mv);}

public static void EscreverCabecalho(Worksheet Ws, object Mv)
 {Range R = Ws.get_Range("A1:G2", Mv);
  R.Cells[1, 1] = "Escreva aqui o seu nome";
  R.Cells[2, 1] = "Pergunta";
  R.Cells[2, 6] = "A sua Resposta";
  for (int j = 1; j <= R.Columns.Count; j++)
  R.EntireColumn.AutoFit();}
public static void EscreverTeste(Worksheet Ws, object Mv,
string[,] Perg)
 {const int TPerg=5;
 for (int I=0; I<Perg.GetLength(0); I++)
   Perg[I, 0] = "L";
  Random Aleat = new Random();
  int Ind;
  Range R = Ws.get_Range("A3:F3", Mv);
  for (int K=1; K<=TPerg; K++)
   {Ind = Aleat.Next(10);
    while (Perg[Ind, 0] == "O")
     {Ind = Aleat.Next(10);}
    for (int E = 1; E < 6; E++)
     R.Cells[K, E] = Perg[Ind, E];
     R.Cells[K, 7] = Perg[Ind, 6];
     Perg[Ind, 0] = "O";
     }}

public static void EsconderCorretas(Worksheet Ws, object Mv,
string PalChave)
 {const string AreaLivre = "A1:F10";
  const string EndR = "G1:G10";
  try
   {Range R = Ws.get_Range(AreaLivre, Mv);
    R.Locked = false;
    R = Ws.get_Range(EndR, Mv);
    if ((bool)(R.EntireColumn.Hidden) == false)
    R.EntireColumn.Hidden = true;}
  catch
  {}
  Ws.Protect(PalChave, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv,
  Mv, Mv, Mv, Mv, Mv);}

static void Main(string[] args)
 {const string PalChave = "Porto";
  const string Pasta = "C:\\";
  object Mv = Missing.Value;
  Excelapp Excel = new Excelapp();
  Excel.Visible = true;
  Excel.DisplayAlerts=false;
```

```
string[,] Perg = new string[10,7];
LerTeste(Excel, Mv, Perg);
int N = 10;
for (int I = 1; I <= N; I++)
 {Excel.Workbooks.Add(Mv);
  Worksheet Ws = (Worksheet)
  Excel.ActiveWorkbook.Worksheets.get_Item(1);
  EscreverCabecalho(Ws, Mv);
  EscreverTeste(Ws, Mv, Perg);
  EsconderCorretas(Ws, Mv, PalChave);
  MessageBox.Show("O teste " +I.ToString()+ " está preparado");
  Ws.Protect(PalChave, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv, Mv,
  Mv, Mv, Mv, Mv, Mv);
  string Fich =Pasta+ I.ToString();
  Excel.ActiveWorkbook.SaveAs(Fich, Mv, Mv, Mv, Mv, Mv,
  Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlNoChange,
  Mv, Mv, Mv, Mv, Mv);
  Excel.ActiveWorkbook.Close(Mv, Fich, Mv);}
Excel.Quit();}}}
```