



GRAPHICAL USER INTERFACE

Interface Gráfica

Programação III

Prof. Edson Mota, PhD, MSc, PMP

O que você entende por
Interface Gráfica?



Interface Gráfica

- A interface gráfica é uma das partes mais importantes de qualquer sistema computacional
- Por que?
 - Ela [é] o sistema para a maioria dos usuários
 - Ela pode ser vista, ouvida e também pode ser tocada
 - As pilhas de código de software estão invisíveis, escondidas atrás de monitor, teclado e mouse
 - Toda complexidade do software pode ser ocultada por uma boa interface gráfica

Definições

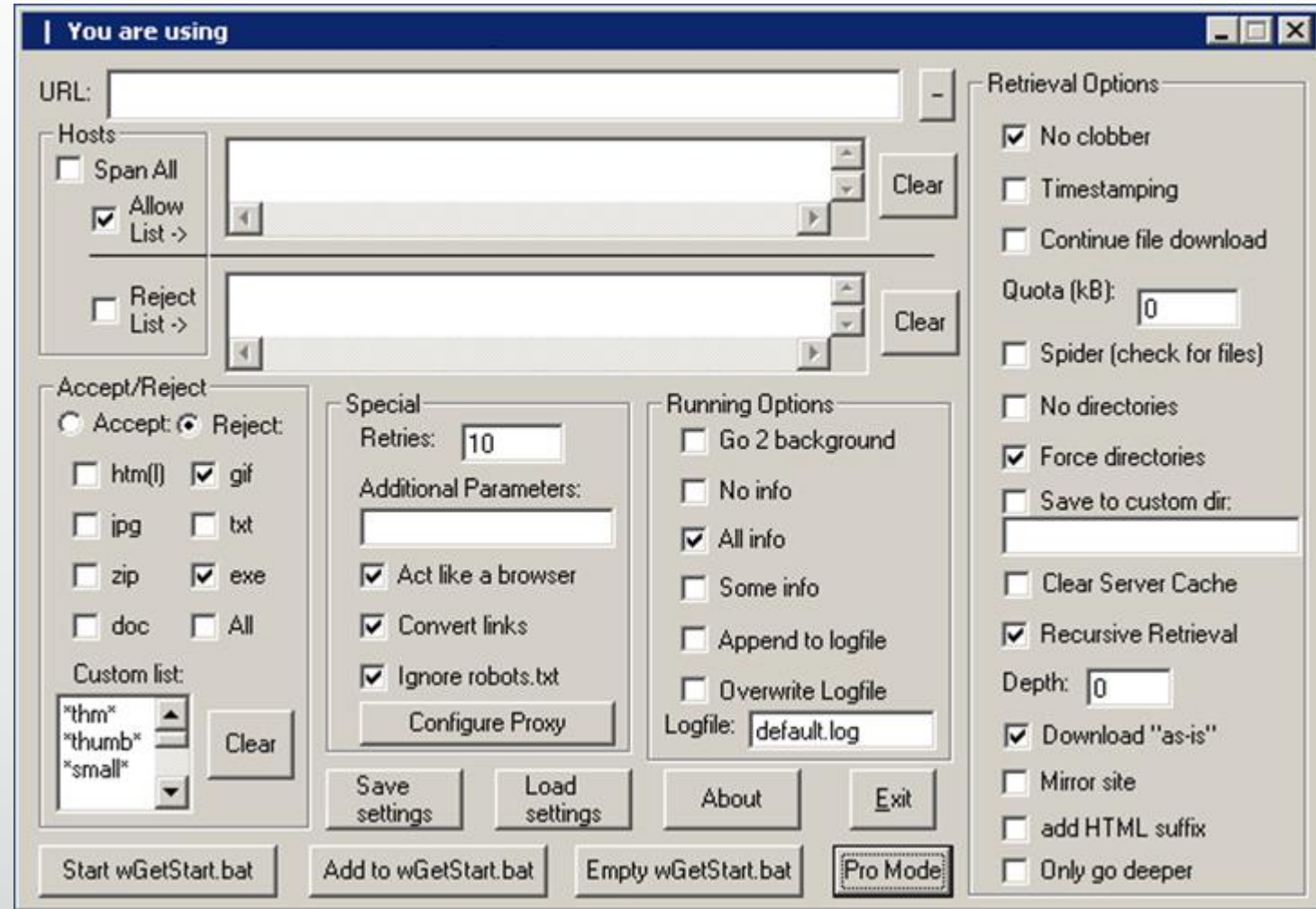
- Podemos definir a interface gráfica como a parte de um computador e de seu software que pessoas podem ver, ouvir, tocar, falar com....
- Possui, essencialmente, dois componentes: entrada e saída
 - Entrada (input) é como uma pessoa comunica as suas necessidades ou desejos ao computador
 - Saída (output) é como o computador comunica o resultado de suas computações e requisitos ao usuário

”

“A *melhor interface* é aquela que *não é notada*, aquela que *permite que o usuário mantenha o foco na informação e tarefa em mãos, não no mecanismo usado para apresentar a informação e realizar a tarefa*”

A importância de um bom design

- Com a tecnologia e ferramentas atuais, e com nossa motivação em criar interfaces e telas realmente efetivas e usáveis, **por que continuamos a produzir sistemas que são ineficientes e confusos?**



A importância de um bom design

- Uma interface com telas bem projetadas é fundamental para os usuários desta aplicação, seja ela web, desktop ou mobile
 - É a sua janela para visualização das capacidades do sistema
- Para muitos, é o sistema, sendo um dos poucos componentes visíveis do produto criado pelo desenvolvedor
- É também o veículo através do qual muitas tarefas críticas são apresentadas



A importância de um bom design

- Um layout e aparência de tela afetam uma pessoa de várias maneiras
 - Se ele é confuso e ineficiente, pessoas irão ter grande dificuldade para realizar seus trabalhos e irão cometer muitos erros
 - O design pobre pode até mesmo fazer com que pessoas abandonem permanentemente o uso de um sistema



A importância de um bom design

- Isso, tradicionalmente, se apresenta em uma relação de **amor e ódio**
 - **Equipe de TI vs Usuários** (Um tipo de nós contra eles)

Quando **não** funciona!



Quando funciona!

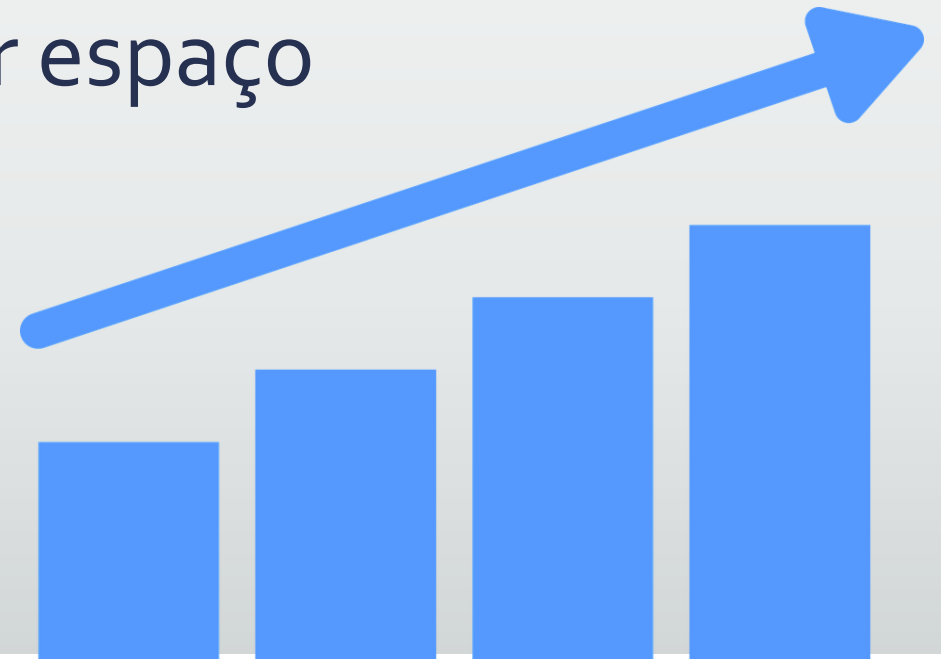


O aprimoramento contínuo das interfaces pode ajudar no engajamento e maximizar as chances de sucesso



Benefícios do Bom Design

- Um pesquisador tentou aumentar a clareza e legibilidade das telas tornando-as menos cheias
- Separou itens, os quais tinham sido combinados em uma mesma linha, para conservar espaço
- Resultado: **aumento de produtividade em 20%**



Benefícios do Bom Design



WWW.ULTRACOLORINGPAGES.COM

- Um outro pesquisador reformatou uma série de telas de usuários seguindo um conjunto de boas práticas
- Resultado: os usuários completaram as transações em um tempo **25% menor**, com **25% menos erros** que as telas originais

Década de 80

THE CAR RENTAL COMPANY		
RENTER >>	Name:	_____
	Telephone:	____ _
LOCATION >>	Office:	_____
	Pick-up Date:	____ _
	Return Date:	____ _
AUTOMOBILE >>	Class:	_____ (PR, ST, FU, MD, CO, SC)
	Rate:	_____
	Miles Per Day:	_____
The maximum allowed miles per day is 150.		
Enter F1=Help F3=Exit F12=Cancel		

Década de 90

THE CAR RENTAL COMPANY

RENTER

Name:

Telephone:


LOCATION



Office:



Pick-up Date:

Return Date:

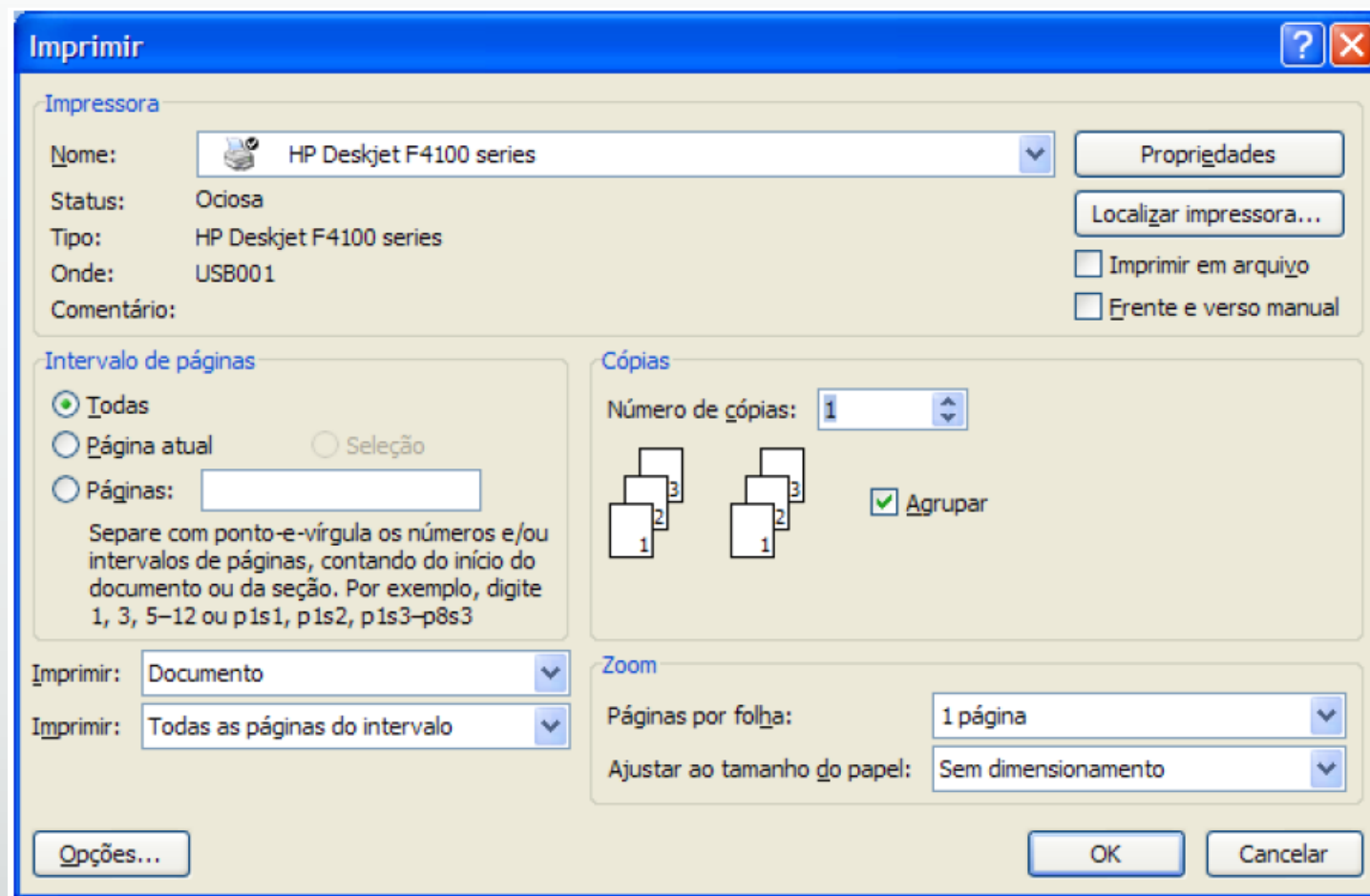
AUTOMOBILE

Class: 

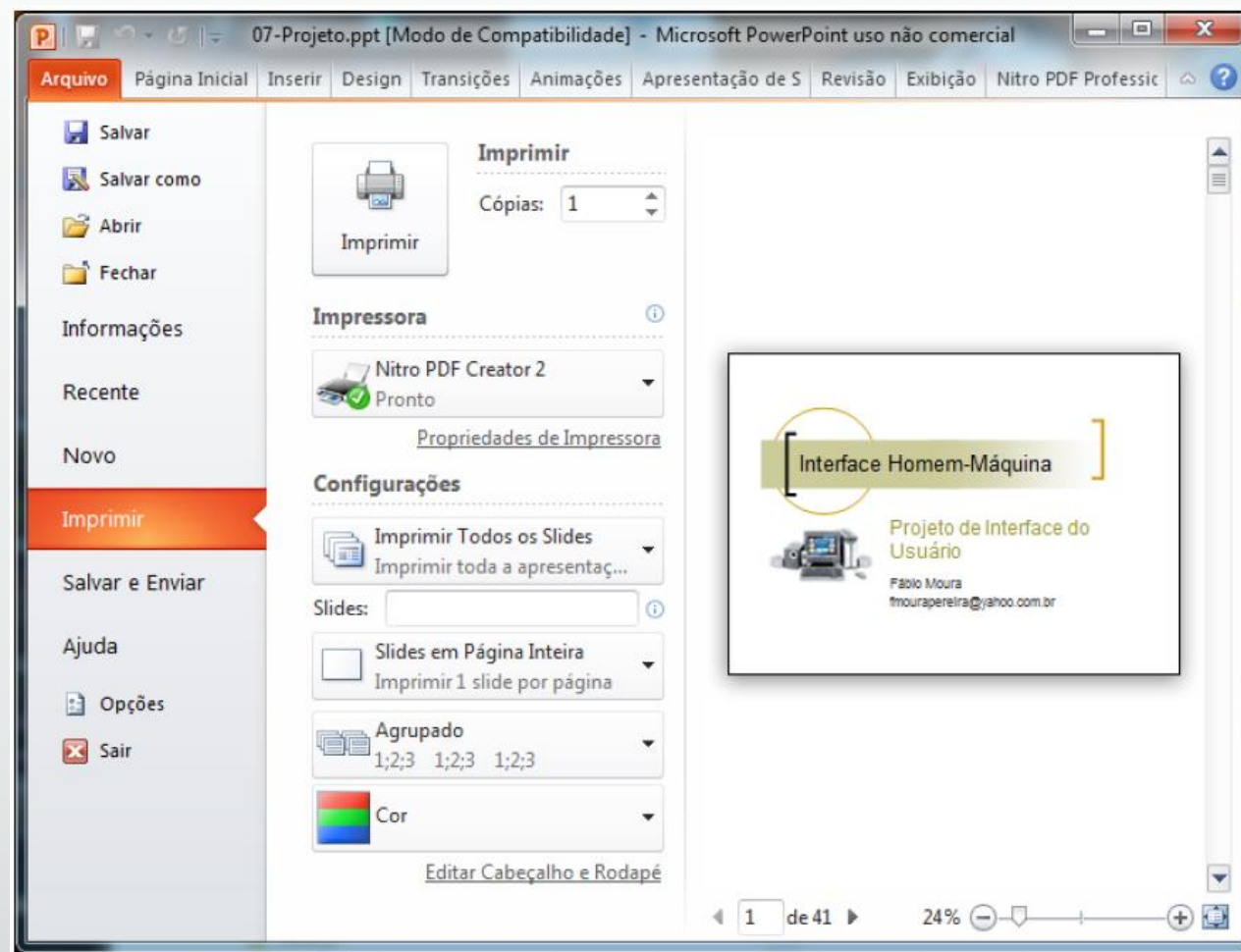
Rate:  

Miles Per Day:  

Década de 2000



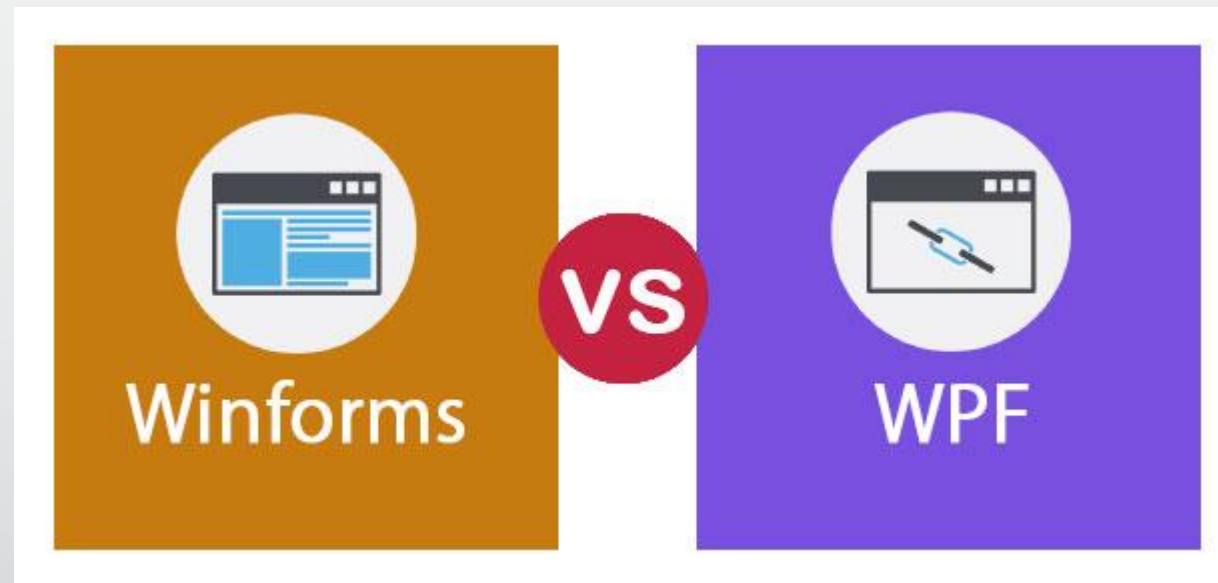
Década de 2010



Interface Gráfica Framework.NET

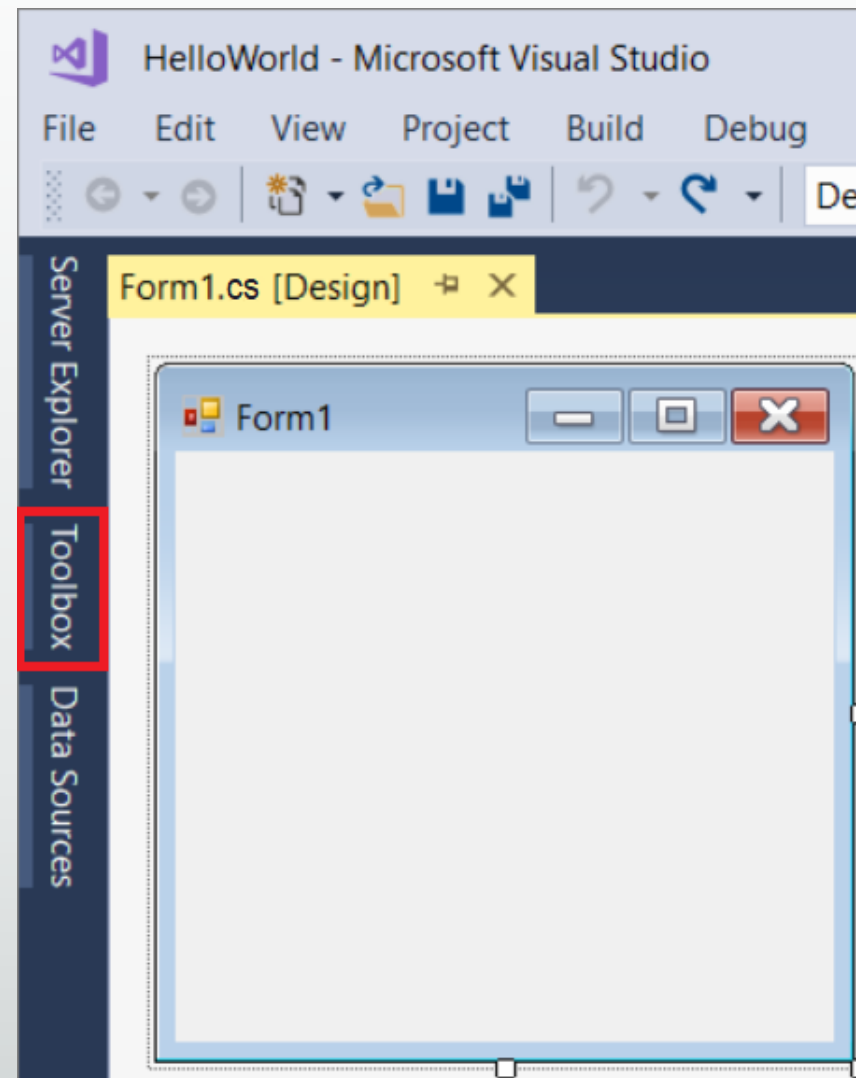
Dois Mundos!

- No mundo das aplicações desktop, disponíveis no Farmerwork.net, existem duas plataformas de desenvolvimento mais expressivas.



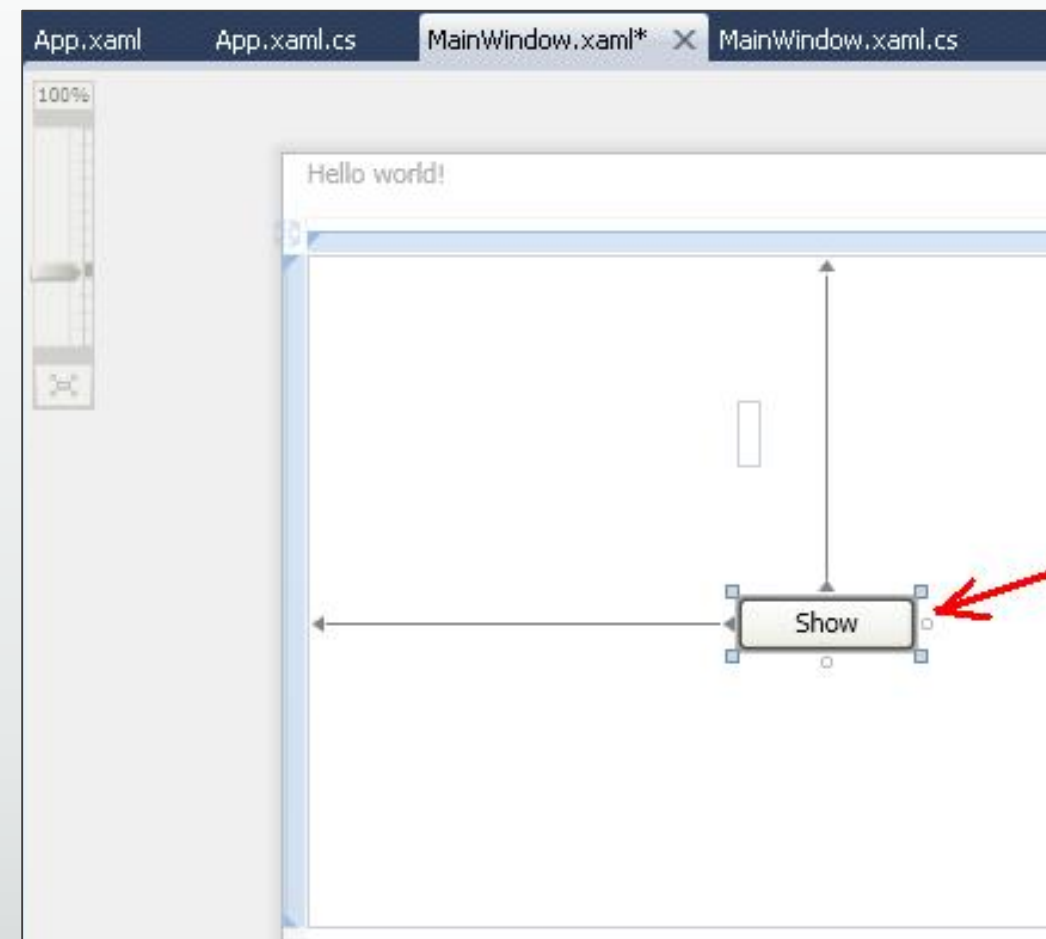
WinForms (Windows Form Application):

- Consiste em uma abordagem baseada em GUI introduzida no framework .NET.
- Antes do WPF / Silverlight, era a única API para .NET usada para construir a GUI.
- Permite o desenvolvimento de aplicativos de forma fácil, garantindo rapidez na implantação, atualização e gerenciamento.
- O desenvolvimento do WinForms é muito simples, pois é baseado apenas no posicionamento de arrastar e soltar dos controles da interface do usuário na tela.
- Além disso, essa abordagem possui um amplo conjunto de componentes e suporte à programação.

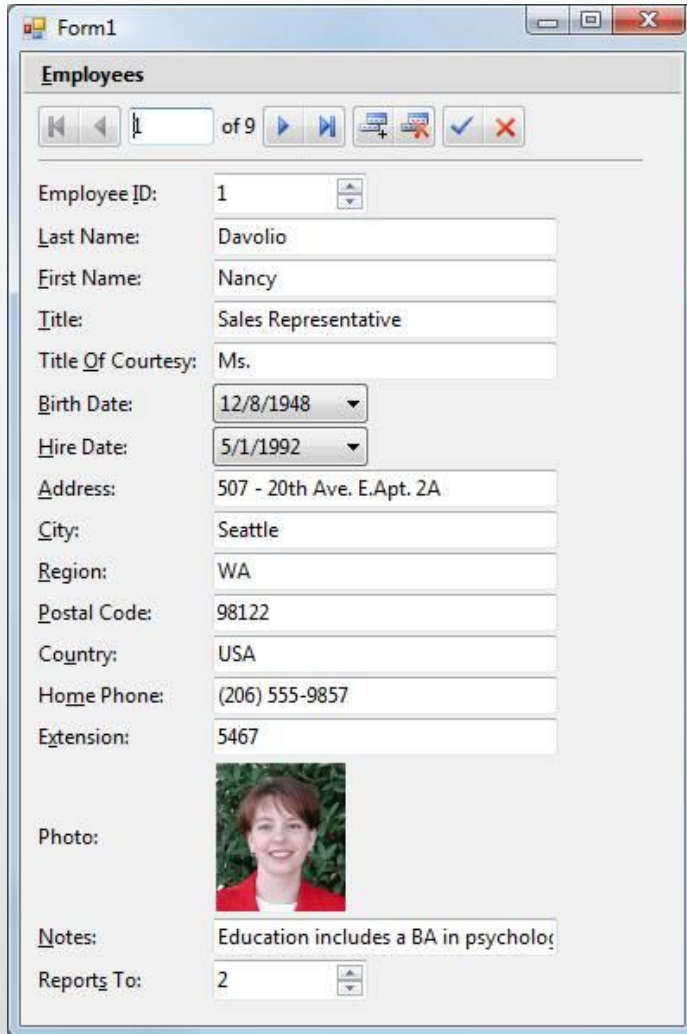


WPF (Windows Presentation Foundation):

- Consiste em uma estrutura de interface do usuário usada para desenvolver aplicativos Windows ou clientes de desktop.
- É a abordagem mais recente para a estrutura GUI usada com a estrutura .NET
- Ele geralmente fornece meios holísticos para combinar ou integrar UI, gráficos 2D e 3D, mídia digital e documentos.
- Ele é responsável por desenvolver, exibir e manipular a interface do usuário, documentos, imagens, etc.
- Ênfase em recursos e gerenciamento do conteúdo gráfico da aplicação



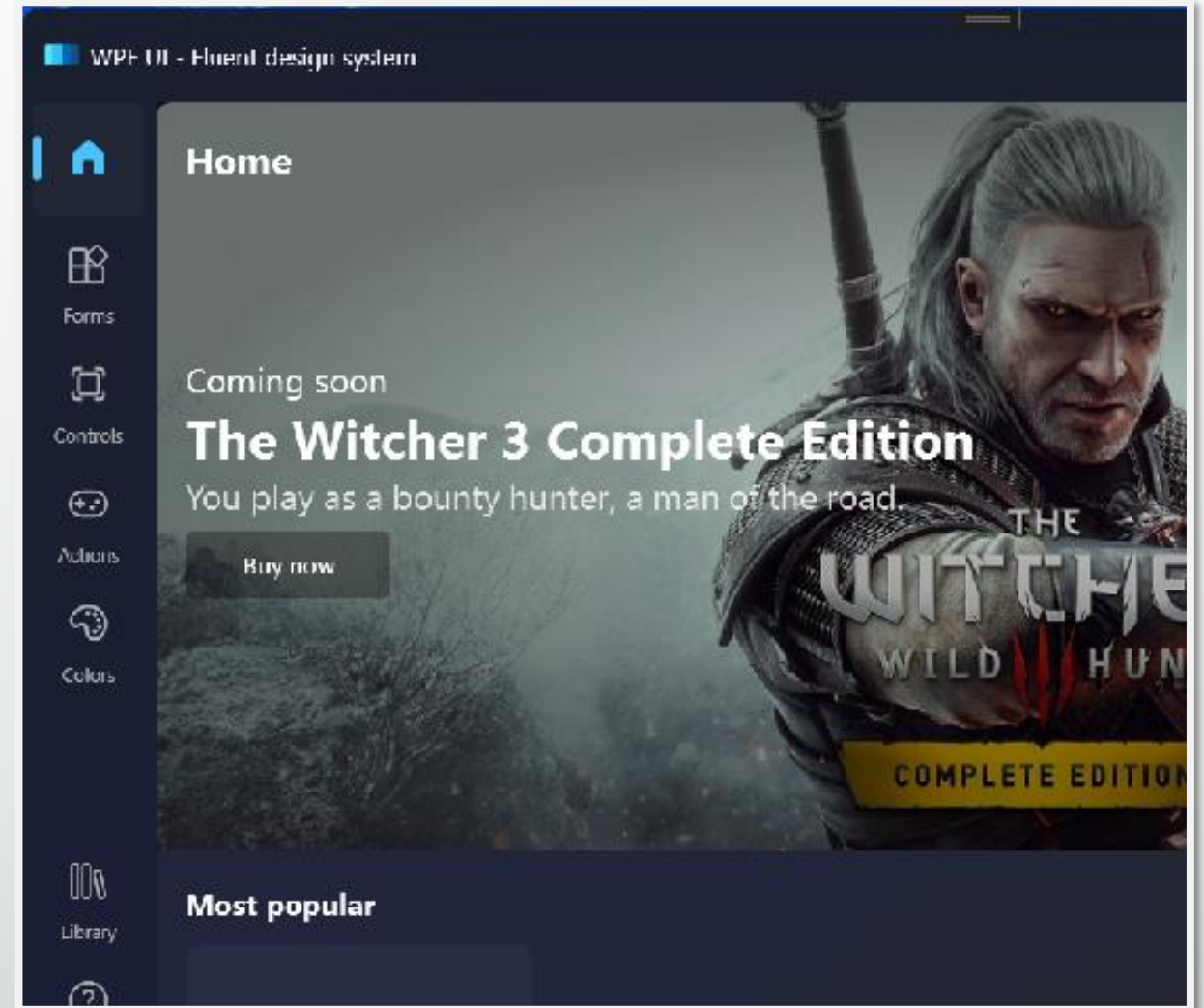
Abordagens Diferentes



A screenshot of a Windows Form titled "Form1". The form is titled "Employees" and displays a list of 9 items, with the first item selected. The form contains the following fields:

- Employee ID: 1
- Last Name: Davolio
- First Name: Nancy
- Title: Sales Representative
- Title Of Courtesy: Ms.
- Birth Date: 12/8/1948
- Hire Date: 5/1/1992
- Address: 507 - 20th Ave. E.Apt. 2A
- City: Seattle
- Region: WA
- Postal Code: 98122
- Country: USA
- Home Phone: (206) 555-9857
- Extension: 5467
- Photo: A small portrait photo of a woman with short brown hair.
- Notes: Education includes a BA in psychology
- Reports To: 2

Windows Form

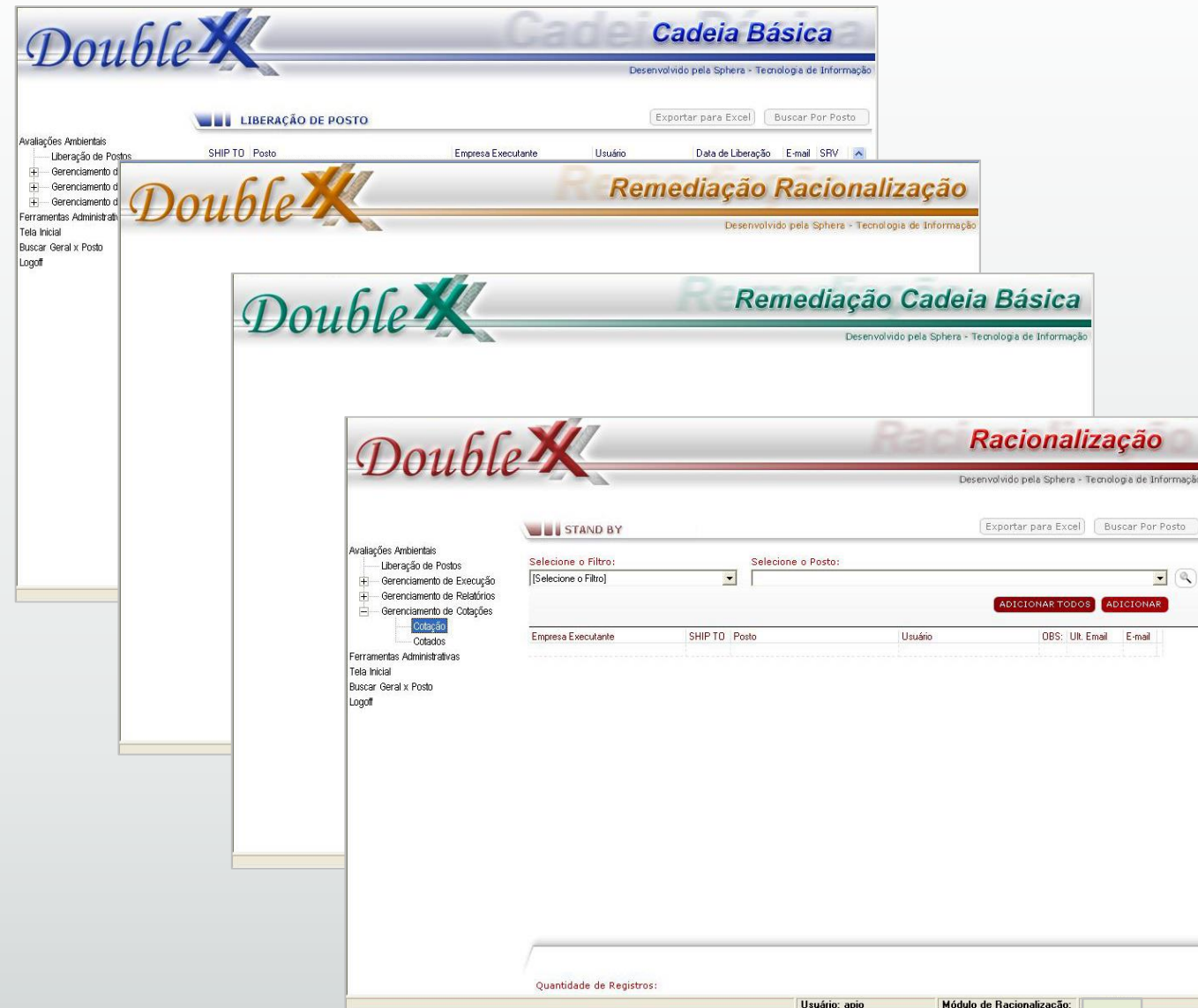


Windows Presentation Foundation

Alguns Cases

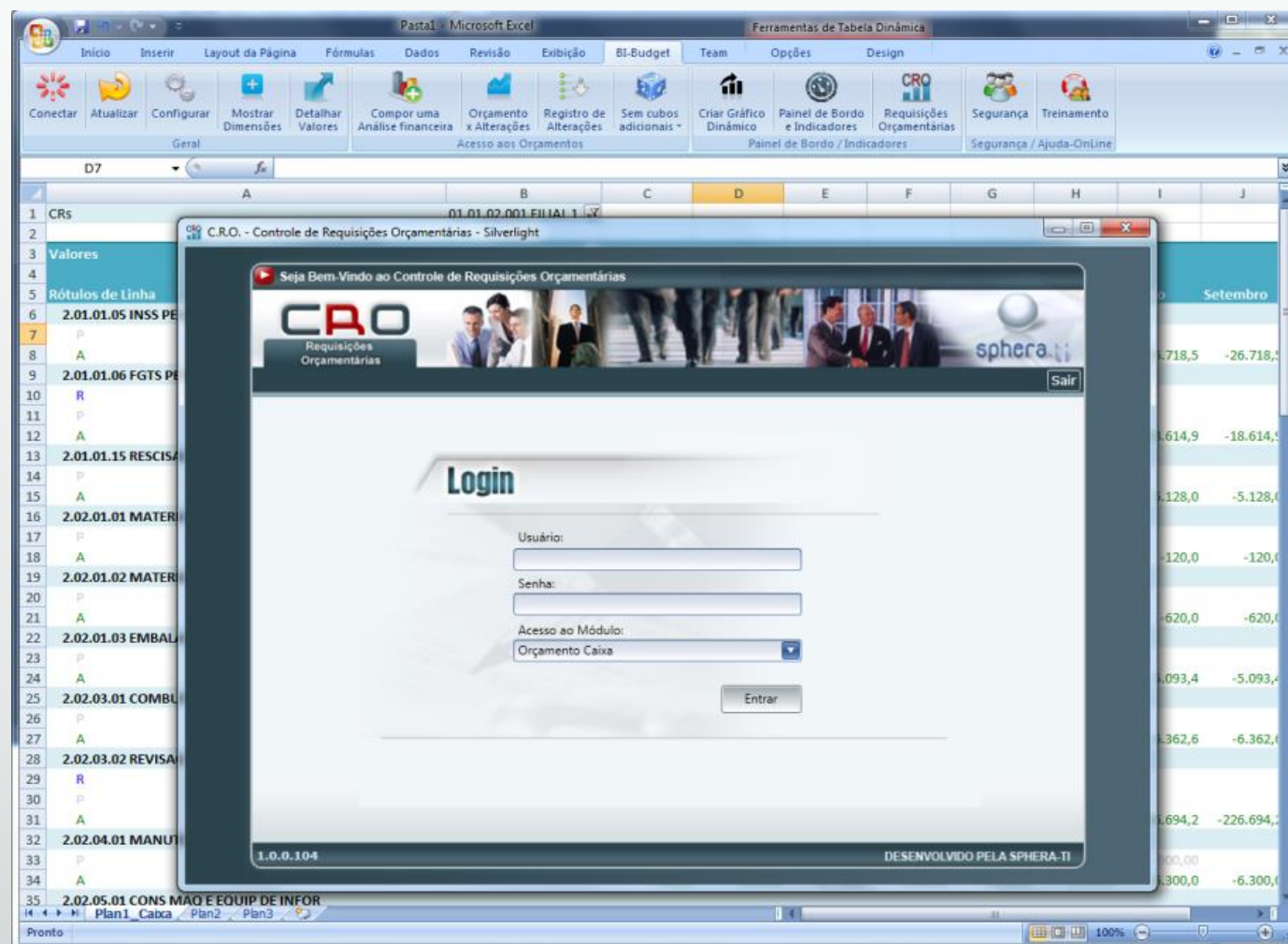
Aplicação DoubleX

- Desenvolvida utilizando Windows Form Nativo



Controle de Requisições Orçamentárias

- Aplicação WPF integrada ao Microsoft Excel e Plug-ins OBA



Implementando Windows Forms

Visual Studio 2022

Abrir recente

Pesquisar recentes (Alt+S)



Hoje



WPF-TEste.sln

29/03/2023 20:13

D:\...\ATIVIDADES\2023.1\cimaterc\disciplinas\PROGRAMAÇÃO III\codigos\WPF-TEste

Ontem



AV1.sln

28/03/2023 08:57

D:\...\AULAS\ATIVIDADES\2023.1\cimaterc\disciplinas\PROGRAMAÇÃO III\codigos\AV1

Esta semana



windows-form-hello.sln

22/03/2023 20:36

D:\...\2023.1\cimaterc\disciplinas\PROGRAMAÇÃO III\codigos\windows-form-hello



windows-form01.sln

22/03/2023 20:30

D:\...\2023.1\cimaterc\disciplinas\PROGRAMAÇÃO III\codigos\windows-form01



gerencia-usuarios.csproj

22/03/2023 20:02

D:\...\2023.1\cimaterc\disciplinas\PROGRAMAÇÃO III\codigos\gerencia-usuarios

Este mês



windiws-form.sln

15/03/2023 23:37

D:\...\2023.1\cimaterc\disciplinas\PROGRAMAÇÃO III\codigos\windiws-form

Introdução



Clonar um repositório

Obter o código de um repositório online, como o GitHub ou o Azure DevOps



Abrir um projeto ou uma solução

Abrir um projeto local do Visual Studio ou arquivo .sln



Abrir uma pasta local

Navegar e editar o código dentro de qualquer pasta










Criar um projeto

Escolha um modelo de projeto com scaffolding de código para começar

[Continuar sem código](#) →

Criar um novo projeto






Modelos de projeto recentes

-  Aplicativo do Windows Forms (.NET Framework) C#
-  Aplicativo WPF C#
-  Aplicativo do Console C#
-  Aplicativo do Windows Forms C#
-  Aplicativo do Windows Forms Visual Basic
-  Solução em branco
-  API Web do ASP.NET Core C#

Seleciona

Search bar: Limpar tudo

Filters: Todos os idiomas | Todas as plataformas | Todos os tipos de projeto

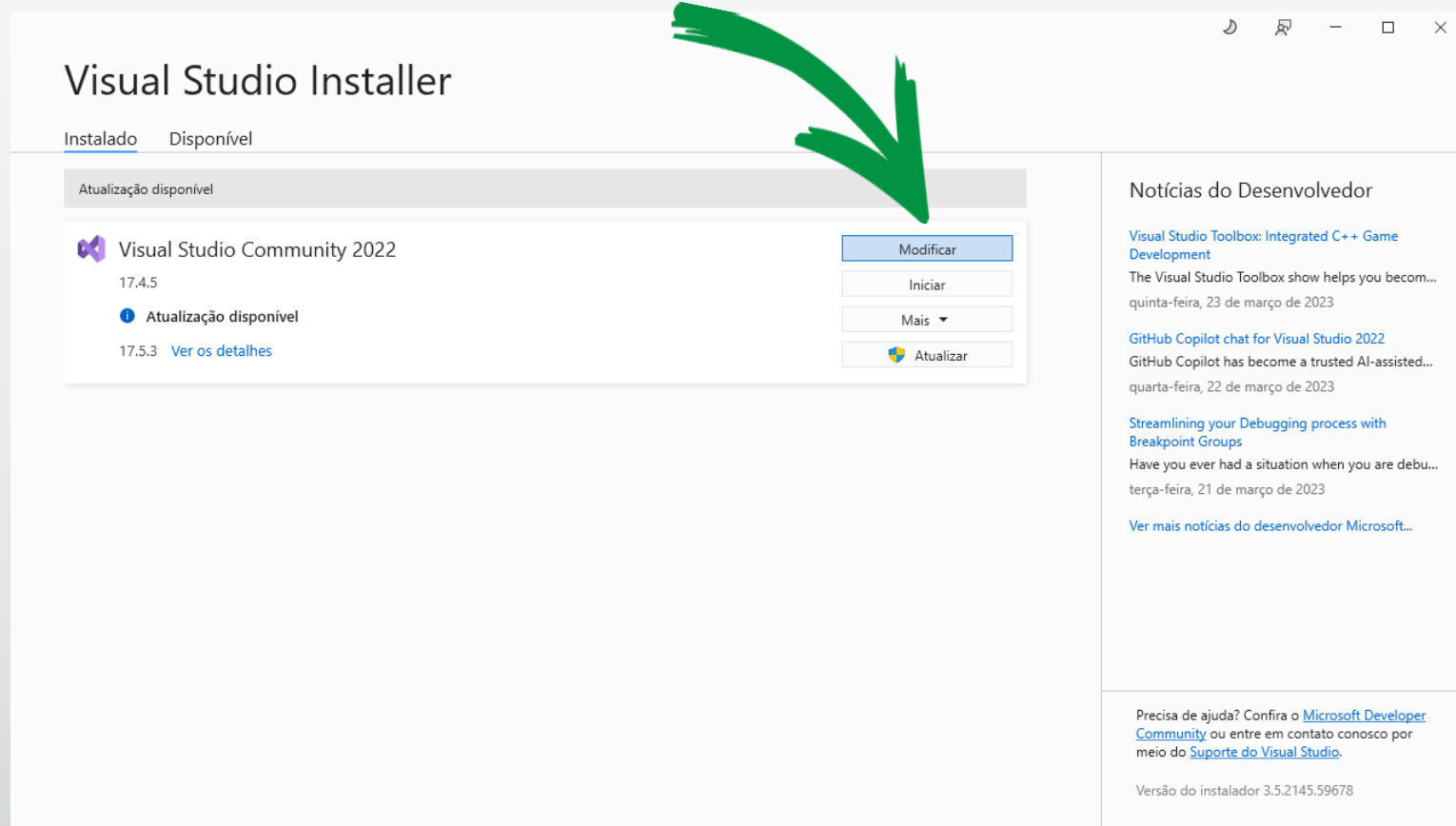
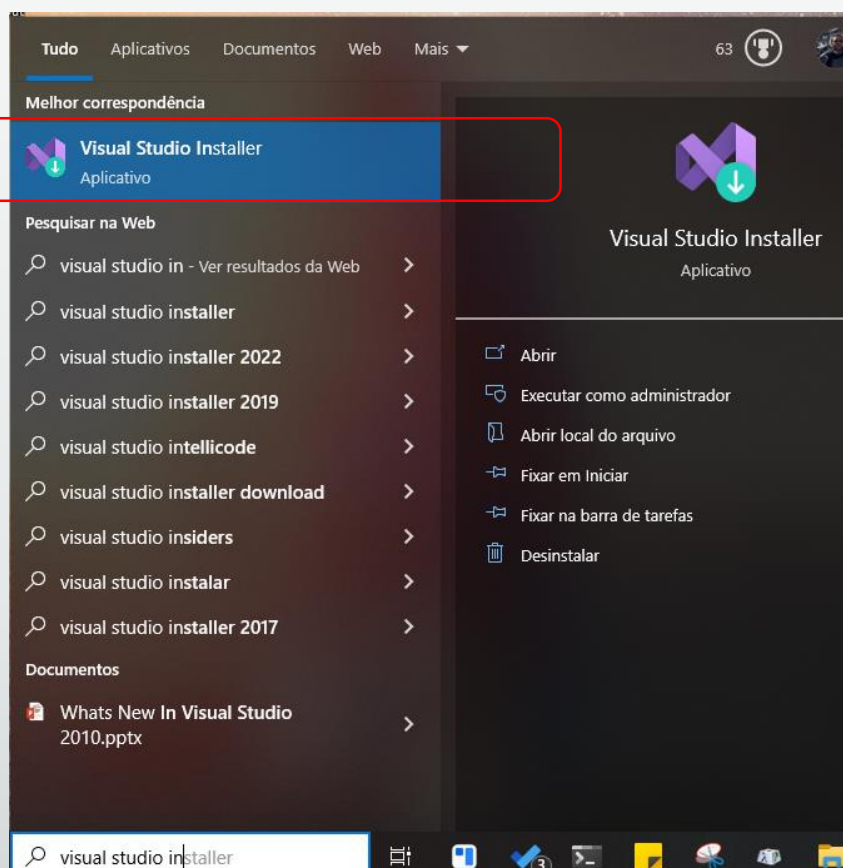
-  **Aplicativo do Windows Forms**
Um modelo de projeto para criar um Aplicativo do WinForms (Windows Forms) do .NET.
C# Windows Área de Trabalho
-  **Aplicativo do Windows Forms**
Um modelo de projeto para criar um Aplicativo do WinForms (Windows Forms) do .NET.
Visual Basic Windows Área de Trabalho
-  **Aplicativo do Windows Forms (.NET Framework)**
Um projeto para criar um aplicativo com uma interface do usuário do Windows Forms (WinForms)
C# Windows Área de Trabalho
-  **Biblioteca de Controles do Windows Forms (.NET Framework)**
Um projeto para criar controles a serem usados em aplicativos do Windows Forms (WinForms)
C# Windows Área de Trabalho Biblioteca
-  **Aplicativo do Windows Forms (.NET Framework)**
Um projeto para criar um aplicativo com uma interface do usuário do Windows Forms (WinForms)
Visual Basic Windows Área de Trabalho

Voltar Próximo

Pesquisa por templates

Não Localizei!

- Será preciso realizar a instalação do template



Desenvolvimento Desktop

Modificando — Visual Studio Community 2022 — 17.4.5

Cargas de trabalho Componentes individuais Pacotes de idiomas Locais de instalação

Web e Nuvem (4)

- ☒ **ASP.NET e desenvolvimento Web**
Crie aplicativos Web usando ASP.NET Core, ASP.NET, HTML/JavaScript e Contêineres, incluindo suporte a Doc...
- ☐ **Desenvolvimento Python**
Edição, depuração, desenvolvimento interativo e controle do código-fonte para Python.
- ☐ **Desenvolvimento para Azure**
SDKs, ferramentas e projetos do Azure para desenvolver aplicativos na nuvem e criar recursos usando o .NET e o
- ☐ **Desenvolvimento em node.js**
Crie aplicativos de rede escalonáveis usando Node.js, um tempo de execução do JavaScript conduzido pelo evento...

Área de trabalho e Dispositivos móveis (5)

- ☐ **Desenvolvimento de .NET Multi-Platform App UI**
Crie aplicativos Android, iOS, Windows e Mac a partir de uma única base de código usando C# com .NET MAUI.
- ☒ **Desenvolvimento para desktop com .NET**
Crie WPF, Windows Forms e aplicativos de console usando C#, Visual Basic e F# com .NET e .NET Framework.
- ☐ **Desenvolvimento para desktop com C++**
Crie aplicativos C++ modernos para Windows usando as ferramentas de sua escolha, incluindo MSVC, Clang, CMake...
- ☐ **Desenvolvimento com a Plataforma Universal do Wind...**
Crie aplicativos para a Plataforma Universal do Windows usando C#, VB, JavaScript ou como alternativa, C++...

Localização
C:\Program Files\Microsoft Visual Studio\2022\Community


Ao continuar, você concorda com a [licença](#) da edição do Visual Studio selecionada. Também oferecemos a possibilidade de baixar outro software com o Visual Studio. Esse software é licenciado separadamente, conforme [Avisos de Terceiros](#) ou na licença que o acompanha. Ao continuar, você também concorda com essas licenças.

Detalhes da instalação

- ▶ Editor de núcleo do Visual Studio
- ▶ ASP.NET e desenvolvimento Web
- ▼ Desenvolvimento para desktop co... •
 - ▼ Incluído
 - ✓ Ferramentas de desenvolvimento para des...
 - ✓ Ferramentas de desenvolvimento do .NET...
 - ✓ C# e Visual Basic
 - ▼ Opcional
 - ☒ Ferramentas de desenvolvimento para .NET
 - ☒ Ferramentas de desenvolvimento do .NET...
 - ☒ Ferramentas do Entity Framework 6
 - ☒ Ferramentas de criação de perfil do .NET
 - ☒ IntelliCode
 - ☒ Depurador Just-In-Time
 - ☒ Live Share
 - ☒ ML.NET Model Builder
 - ☒ Blend for Visual Studio
 - ☐ Suporte à linguagem F# da área de trabalho
 - ☐ De-Serialização Distribuída, De-Serialização...

Remove componentes sem suporte

Espaço total necessário 2,9 GB

Instalar durante o download ▾  Modificar

Criar um novo projeto

Modelos de projeto recentes

- Aplicativo do Windows Forms (.NET Framework) C#
- Aplicativo WPF C#
- Aplicativo do Console C#
- Aplicativo do Windows Forms C#
- Aplicativo do Windows Forms Visual Basic
- Solução em branco
- API Web do ASP.NET Core C#

Seleciona

Search bar: windows form

Filters: Todos os idiomas, Todas as plataformas, Todos os tipos de projeto

Results:

- Aplicativo do Windows Forms (C#)
Um modelo de projeto para criar um Aplicativo do WinForms (Windows Forms) do .NET.
C# Windows Área de Trabalho
- Aplicativo do Windows Forms (VB)
Um modelo de projeto para criar um Aplicativo do WinForms (Windows Forms) do .NET.
Visual Basic Windows Área de Trabalho
- Aplicativo do Windows Forms (.NET Framework) (C#)
Um projeto para criar um aplicativo com uma interface do usuário do Windows Forms (WinForms)
C# Windows Área de Trabalho
- Biblioteca de Controles do Windows Forms (.NET Framework) (C#)
Um projeto para criar controles a serem usados em aplicativos do Windows Forms (WinForms)
C# Windows Área de Trabalho Biblioteca
- Aplicativo do Windows Forms (.NET Framework) (VB)
Um projeto para criar um aplicativo com uma interface do usuário do Windows Forms (WinForms)
Visual Basic Windows Área de Trabalho

Buttons: Voltar, Próximo

Pesquisa por templates

Configurar seu novo projeto

Aplicativo do Windows Forms

C#

Windows

Área de Trabalho

Nome do projeto

winform-helloworld

Local

D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas\PROGRAMAÇÃO



Nome da solução ⓘ

winform-helloworld

☐

Colocar a solução e o projeto no mesmo diretório

Voltar

Próximo

Informações adicionais

Aplicativo do Windows Forms

C#

Windows

Área de Trabalho

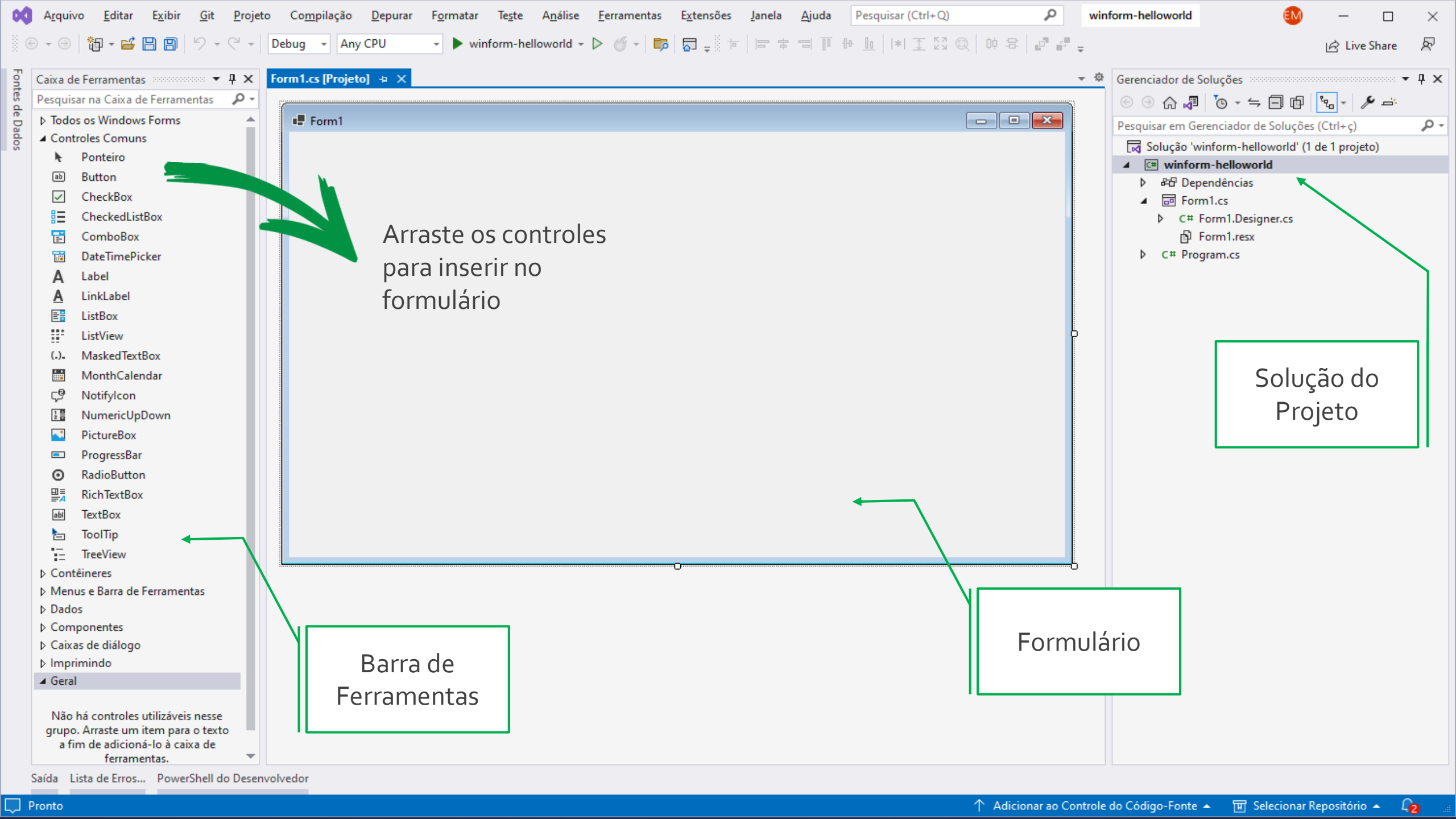
Estrutura ⓘ

.NET 6.0 (Suporte de Longo Prazo)

Voltar

Criar





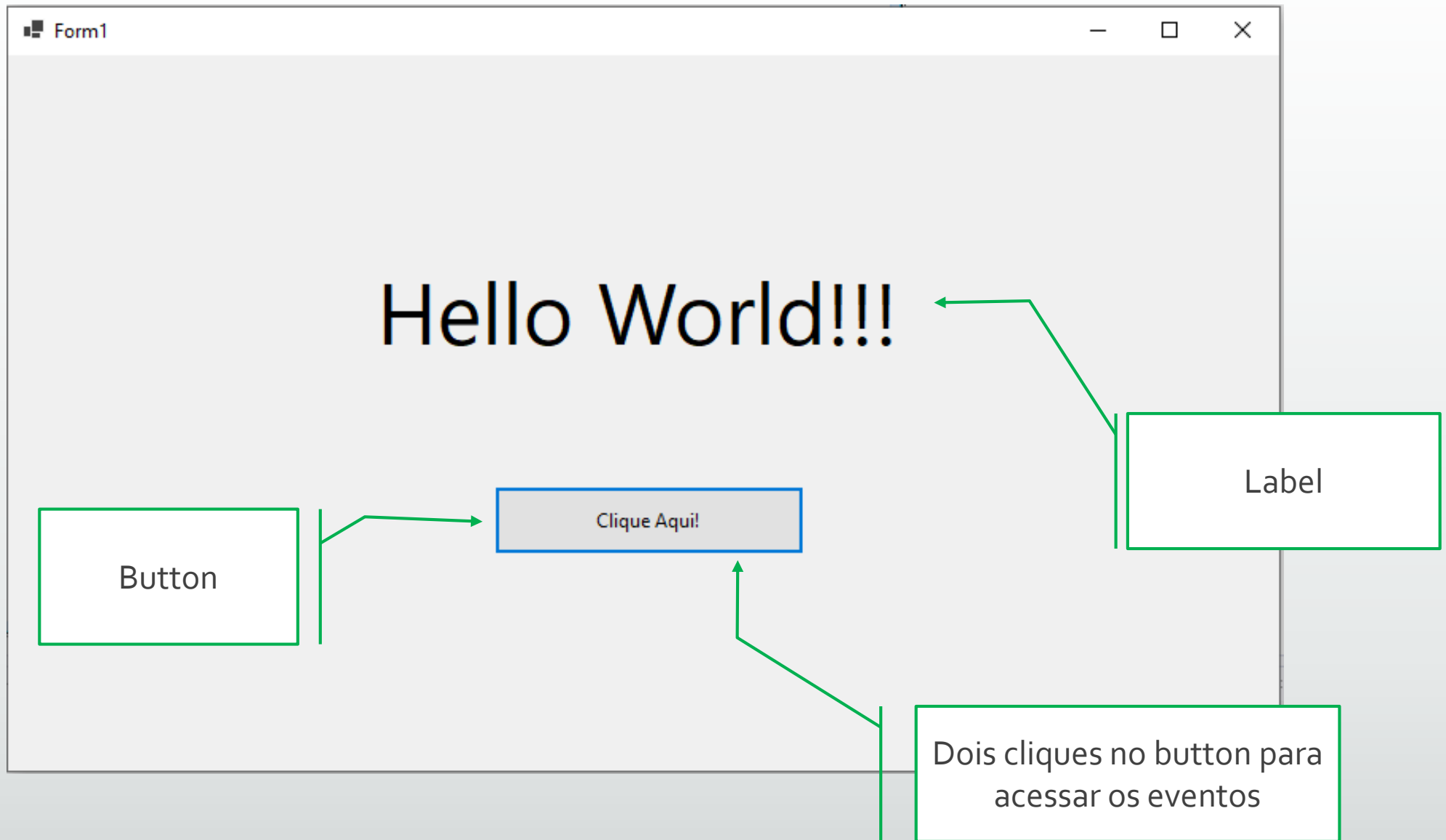
Arraste os controles
para inserir no
formulário

Barra de
Ferramentas

Formulário

Solução do
Projeto

Hello World



Revisão

- Implemente uma tela de autenticação de usuário
- A tela deve conter os campos:
 - Usuário
 - Senha (utilize uma máscara para que os caracteres da senha não sejam visíveis)
- O botão para conectar deve verificar se o nome e a senha coincidem com as credenciais definidas na aplicação
 - Emita uma mensagem de usuário conectado com sucesso, caso a senha esteja correta
 - Ou uma mensagem de permissão negada, caso a senha esteja errada
 - Sintaxe da mensagem: `MessageBox.Show("Mensagem desejada");`



Coleções

Arrays

- Derivado do namespace `System.Array`
- Unidimensionais
 - `int[] vetR;`
- Multidimensionais
 - `int[,] matR = new int[2,3];`
- Inicializações Declarativas
 - `int[,] matR = new int[2,3] { {1,2,3}, {4,5,6} };`

Arrays em C#

- Declaração

```
int[] primes;
```

- Alocação

```
int[] primes = new int[9];
```

- Inicialização

```
int[] prime = new int[] {1,2,3,5,7,11,13,17,19};  
int[] prime = {1,2,3,5,7,11,13,17,19};
```

- Acesso

```
prime2[i] = prime[i];
```

- Iteração

```
foreach (int i in prime) Console.WriteLine(i);
```

Iterando sobre Arrays

```
public class Program
{
    0 referências
    public static void Main(string[] args)
    {
        string[] Filmes = { "Mandalorian", "The last of Us", "Breaking Bad" };

        // Iterando com Enum
        foreach(string s in Filmes)
        {
            Console.WriteLine(s);
        }
    }
}
```


List em C#

List()

- Biblioteca de coleções
- Substituiu o vector
- Usar a importação da classe para utilizar esta
- Biblioteca: **using System.Collections;**

Sintaxe

```
List<String> a = new List<String>();
```

- A declaração acima cria um objeto da classe List vazio que pode armazenar *strings*
- String é o tipo de dado que será armazenado no Container
- Não é necessário mencionar o tamanho

Sintaxe

```
List<Pessoa> P = new List<Pessoa>();
```

- Neste caso, o conteúdo da lista será um objeto “Pessoa”
- Então, cada P[i] poderá conter:
 - Nome
 - Telefone
 - Endereço
 - Etc....

Método ADD

- Uma vez instanciado, o objeto funciona como um repositório (uma coleção) que armazena dados em memória.

```
List<string> a = new List<string>();
```

```
a.Add("Olá");
```

```
a.Add("Bom Dia");
```

```
Console.WriteLine(a.Count); // Qual o Resultado?
```

Método

- Método Remove : Remove um determinado elemento do arraylist

```
List<string> a = new List<string>();  
a.RemoveAt(1);
```

- Método Count : Saber o tamanho atual do List

```
List<String> a = new List<String>();  
int tamanho = a.count();
```

Revisão

- Então, alguma ideia de como podemos listar todos os itens de um List()?
- Como podemos fazer isso?

Iterando com uma variável de controle

```
List<string> list = new List<string>();
```

```
list.Add("Miguel");
```

```
list.Add("José");
```

```
list.Add("Marcos");
```

```
//iterando sob coleções
```

```
for (int i=0; i<list.Count(); i++) {
```

```
    String element = list[i];
```

```
    Console.WriteLine (element);
```

```
}
```


Iterando em uma coleção

```
List<string> list = new List<string>();
```

```
list.Add("Miguel");
```

```
list.Add("José");
```

```
list.Add("Marcos");
```

```
//iterando sob coleções
```

```
foreach (string element in list) {  
    Console.WriteLine(element);  
}
```



Existem muitas outras **funções disponíveis** para interação com **listas**, mas sempre podemos recorrer a **documentação** para pesquisar, **conforme a necessidade**.

<https://learn.microsoft.com/pt-br/dotnet/api/system.collections.generic.list-1?view=net-7.0>

Programando com List()

Revisão1

- Você foi contratado para implementar o cadastro de clientes de uma loja. Para isso, você resolve utilizar uma lista na qual irá armazenar os dados durante a execução da aplicação.
- O cadastro de cliente deve conter:
 - Nome
 - Telefone
 - Endereço
- Ao clicar no botão salvar, um novo item deve ser adicionado à lista
- Utilize a interface gráfica para desenvolver essa aplicação



Revisão2

- Você foi contratado para criar um sistema de gerenciamento de senhas de usuários. Para isso, o cliente especificou os seguintes requisitos:
- O sistema deve permitir ter um campo para indicar se o usuário é administrador ou usuário comum
- Na lista, o campo senha deve ser gravado utilizando uma estrutura base64, a fim de reduzir a visibilidade deste campo
 - `Convert.ToBase64String(System.Text.Encoding.UTF8.GetBytes(senha))`
- O sistema deve permitir além da inclusão, a deleção de uma conta de usuário
- Implemente a interface gráfica do projeto



Revisão 3

- Crie um programa para cadastrar filmes em um List()
 - O cadastro deve conter:
 - Título
 - Ano
 - Gênero
 - Classe: Filme
 - Método: Listar()
- Utilize o componente ListBox da interface manter uma lista atualizada dos filmes cadastrados



Bons estudos!