

Programacao Orientada a Objetos

GCT052 - Aula 2.1 - Comece a criar com C#

Bento Rafael Siqueira

Universidade Federal de Lavras (UFLA)

26 de agosto de 2025

Sumario

1. Introdução ao Capítulo 1
2. Aplicações Console (Base)
3. Transição: Console para WPF
4. Exemplo WPF: Calculadora com Interface Gráfica
5. Exemplo WPF: Jogo de Combinação
6. Configuração do Ambiente
7. Próximos Passos
8. Resumo

O que voce vai aprender hoje?

Programacao
Orientada a
Objetos

Bento Rafael
Siqueira

Introducao
ao Capitulo 1

Aplicacoes
Console
(Base)

Transicao:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Grafica

Exemplo
WPF: Jogo
de
Combinacao

Configuracao
do Ambiente

Objetivos da aula:

- **Objetivo:** Começar com Console e evoluir para WPF
- **Foco:** Aprendizado progressivo: Console → WPF
- **Resultado:** Base sólida + aplicação desktop funcional
- **Metodologia:** Aprender fazendo, do simples ao complexo

Conceitos fundamentais:

- Estrutura básica de programas C#
- Transição de aplicações Console para WPF
- Programação orientada a eventos
- Separação entre interface e lógica

Por que começar com Console?

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração
do Ambiente

Vantagens do Console para aprendizado:

- **Simplicidade:** Conceitos básicos sem distrações
- **Fundamentos:** Entrada/saída, variáveis, controle de fluxo
- **Progressão:** Console → WPF → Aplicações complexas
- **Base sólida:** Compreensão profunda dos conceitos

Benefícios educacionais:

- Foco nos conceitos fundamentais
- Aprendizado sequencial e lógico
- Construção de base sólida
- Transição natural para interfaces gráficas

Por que C# e WPF?

Vantagens das tecnologias:

- **C#**: Linguagem moderna, poderosa e versátil
- **WPF**: Framework avançado para interfaces gráficas
- **XAML**: Linguagem declarativa para UI
- **Integração**: C# + WPF + XAML = Aplicações desktop robustas
- **Carreira**: Habilidades muito valorizadas no mercado

Benefícios práticos:

- Desenvolvimento rápido e eficiente
- Interface rica e moderna
- Código organizado e manutenível
- Suporte oficial da Microsoft

O que é uma aplicação Console?

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração
do Ambiente

Características principais:

- Interface de linha de comando simples
- Entrada e saída via texto
- Execução sequencial de comandos
- Ideal para aprender conceitos básicos
- Base para scripts e ferramentas
- Portabilidade entre sistemas operacionais

Vantagens educacionais:

- Foco na lógica sem distrações visuais
- Entrada/saída simples e direta
- Execução linear e previsível
- Base para automação e scripts

Estrutura básica de um programa Console

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração
do Ambiente

Componentes essenciais:

- **using System** - Importa bibliotecas básicas do .NET Framework
- **namespace** - Organiza o código em um espaço de nomes lógico
- **class Program** - Define a classe principal (base da POO)
- **static void Main** - Método principal, ponto de entrada da aplicação

Conceitos fundamentais:

- Importação de bibliotecas - Acesso às funcionalidades do .NET
- Organização de código - Estruturação lógica e hierárquica
- Definição de classes - Base para criar objetos e estruturas
- Ponto de entrada - Onde a execução do programa começa

Conceitos basicos demonstrados

Programacao
Orientada a
Objetos

Bento Rafael
Siqueira

Introducao
ao Capitulo 1

Aplicacoes
Console
(Base)

Transicao:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Grafica

Exemplo
WPF: Jogo
de
Combinacao

Configuracao
do Ambiente

Elementos fundamentais:

- **Importação de bibliotecas** - Acesso às funcionalidades do .NET
- **Organização de código** - Estruturação lógica e hierárquica
- **Definição de classes** - Base para criar objetos e estruturas
- **Ponto de entrada** - Onde a execução do programa começa
- **Entrada e saída** - Comunicação com o usuário

Benefícios:

- Código organizado e estruturado
- Reutilização de funcionalidades
- Base sólida para POO
- Facilidade de manutenção

Exemplo pratico: Calculadora simples

Programacao
Orientada a
Objetos

Bento Rafael
Siqueira

Introducao
ao Capitulo 1

Aplicacoes
Console
(Base)

Transicao:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Grafica

Exemplo
WPF: Jogo
de
Combinacao

Configuracao
do Ambiente

Lógica implementada:

- Captura de dois números do usuário
- Seleção de operação matemática
- Validação de entrada (divisão por zero)
- Execução da operação selecionada
- Exibição do resultado formatado

Conceitos demonstrados:

- Estruturas de decisão (if/else)
- Switch case para múltiplas opções
- Tratamento de erros básico
- Lógica de cálculo matemático

O que muda ao ir para WPF?

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração
do Ambiente

Principais mudanças:

- Saída de texto → Interface gráfica
- Entrada via teclado → Controles visuais
- Programa sequencial → Programa orientado a eventos
- Um arquivo .cs → Arquivos .cs + .xaml
- Execução linear → Execução baseada em interações

Novos conceitos:

- Programação orientada a eventos
- Separação entre interface e lógica
- Controles visuais e layout
- Gerenciamento de estado da interface

Estrutura de um projeto WPF

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração
do Ambiente

Arquivos principais:

- **App.xaml** - Configuração global da aplicação
- **App.xaml.cs** - Código de inicialização e configuração
- **MainWindow.xaml** - Interface principal em linguagem declarativa
- **MainWindow.xaml.cs** - Lógica de negócio da janela principal
- **Properties/** - Configurações e recursos do projeto

Organização:

- Separação clara entre interface (XAML) e lógica (C#)
- Arquivos de configuração centralizados
- Estrutura hierárquica bem definida

Interface XAML da calculadora

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração
do Ambiente

Interface implementada:

- **Display numérico** para mostrar números e resultados
- **Botões de operação** para selecionar cálculos
- **Layout responsivo** com Grid e StackPanel
- **Design moderno** com espaçamento e margens

Conceitos demonstrados:

- Separação entre interface (XAML) e lógica (C#)
- Controles visuais do WPF
- Layout e posicionamento de elementos
- Estrutura de interface declarativa

Funcionalidades implementadas:

- **Armazenamento de estado** (números e operações)
- **Manipulação de eventos** de clique nos botões
- **Lógica de cálculo** matemático
- **Atualização da interface** em tempo real
- **Gerenciamento de estado** da calculadora

Conceitos demonstrados:

- Programação orientada a eventos
- Manipulação de controles WPF
- Lógica de aplicação desktop
- Gerenciamento de estado da interface

O que é o Jogo de Combinacao?

Programacao
Orientada a
Objetos

Bento Rafael
Siqueira

Introducao
ao Capitulo 1

Aplicacoes
Console
(Base)

Transicao:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Grafica

Exemplo
WPF: Jogo
de
Combinacao

Configuracao
do Ambiente

- **Objetivo:** Encontrar pares de cores iguais
- **Interface:** Grade de botoes coloridos
- **Mecanica:** Clicar em dois botoes para verificar
- **Vitoria:** Todos os pares encontrados
- **Tecnologias:** WPF + XAML + C#

Estrutura da interface:

- **Título do jogo** centralizado e destacado
- **Contador de pares** encontrados
- **Grade de jogo** com botões organizados
- **Layout responsivo** que se adapta ao conteúdo
- **Design limpo** e focado na jogabilidade

Conceitos demonstrados:

- Organização de elementos visuais
- Controles de layout avançados
- Estruturação hierárquica da interface
- Nomenclatura de elementos para acesso via código

Funcionalidades implementadas:

- Criação dinâmica do tabuleiro de jogo
- Gerenciamento de estado dos botões
- Lógica de verificação de pares
- Controle de turnos e seleções
- Atualização visual da interface

Conceitos demonstrados:

- Criação dinâmica de controles
- Arrays bidimensionais
- Gerenciamento de eventos complexos
- Lógica de jogo interativo

O que voce precisa instalar?

Programacao
Orientada a
Objetos

Bento Rafael
Siqueira

Introducao
ao Capitulo 1

Aplicacoes
Console
(Base)

Transicao:
Console para
WPF

Exemplo
WPF: Calculadora
com
Interface
Grafica

Exemplo
WPF: Jogo
de
Combinacao

Configuracao

Ferramentas essenciais:

- **Visual Studio 2022** (Community e gratuito)
- **.NET 8.0 SDK** (ou versão mais recente)
- **Workloads específicos:**
 - .NET desktop development
 - Desktop development with C#

Extensões recomendadas:

- **C# Dev Kit** - Desenvolvimento avançado
- **IntelliCode** - Inteligência artificial
- **Code Spell Checker** - Verificação ortográfica

Vantagens:

- Ambiente completo e integrado
- Ferramentas gratuitas e poderosas
- Suporte oficial da Microsoft

Como criar um projeto WPF

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração

Passos para criação:

1. **File** → **New** → **Project**
2. Selecionar **"WPF Application"**
3. Definir **nome** e **localização**
4. Escolher **.NET 8.0** (ou mais recente)
5. Clicar em **Create**
6. Projeto criado com estrutura básica

Configurações importantes:

- Framework .NET mais recente disponível
- Nome descritivo para o projeto
- Localização organizada no sistema
- Estrutura padrão do Visual Studio

Estrutura do projeto criado

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração

Arquivos e pastas:

- **App.xaml** - Configuração da aplicação
- **App.xaml.cs** - Código de inicialização
- **MainWindow.xaml** - Interface principal
- **MainWindow.xaml.cs** - Lógica da janela
- **Properties/** - Configurações do projeto
- **bin/** e **obj/** - Arquivos de compilação

Propósito:

- Arquivos de interface declarativa (XAML)
- Código de lógica de negócio (C#)
- Configurações e recursos do projeto
- Arquivos gerados automaticamente

O que vem depois desta aula?

Programação
Orientada a
Objetos

Bento Rafael
Siqueira

Introdução
ao Capítulo 1

Aplicações
Console
(Base)

Transição:
Console para
WPF

Exemplo
WPF:
Calculadora
com
Interface
Gráfica

Exemplo
WPF: Jogo
de
Combinação

Configuração
do Ambiente

Próximos passos:

- **Prática:** Implementar os exemplos completos
- **Conceitos:** Classes, objetos, herança
- **Controles:** Mais elementos da interface
- **Eventos:** Outros tipos de interação
- **Projeto:** Aplicação mais complexa
- **Deploy:** Distribuir a aplicação

Objetivos:

- Consolidar conceitos fundamentais
- Aplicar conhecimentos em projetos práticos
- Evoluir para aplicações mais complexas
- Preparar para desenvolvimento profissional

Materiais recomendados:

- **Livro:** "Use a Cabeça C#" (Quarta Edição)
- **Documentação:** Microsoft Learn
- **Vídeos:** Canais especializados em C#
- **Comunidade:** Stack Overflow, Reddit
- **Prática:** Exercícios e projetos pessoais
- **Projetos open source:** GitHub

Estratégias de aprendizado:

- Teoria + prática simultânea
- Projetos pessoais para aplicar conceitos
- Participação ativa na comunidade
- Estudo contínuo e atualizado

Resumo da Aula

Pontos principais:

- **Console → WPF:** Progressão natural de aprendizado
- **Conceitos fundamentais:** Estrutura, sintaxe e lógica
- **C#:** Linguagem poderosa e versátil
- **Prática:** Aprender fazendo é fundamental
- **Próximo:** Implementar os exemplos completos!

Conceitos aprendidos:

- Estrutura básica de programas C#
- Transição de Console para WPF
- Programação orientada a eventos
- Separação entre interface e lógica

Obrigado!

Dúvidas? Próxima aula: Implementação prática dos exemplos

Contato:

- **Professor:** Bento Rafael Siqueira
- **Disciplina:** GCT052 - Programação Orientada a Objetos
- **Universidade:** UFLA