

Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

Atividades Práticas - Fundamentos de POO em C#

Atividade 1: Criando sua Primeira Classe (Abstração e Classes/Objetos)

Objetivo

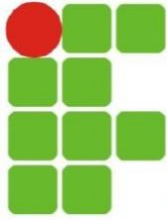
Compreender os conceitos de classe, objeto, atributos e métodos através da criação de uma classe Animal.

Instruções

1. **Crie uma classe Animal** com os seguintes atributos:
 - o Nome (string)
 - o Espécie (string)
 - o Idade (int)
 - o EstaDormindo (bool - inicia como false)
2. **Adicione os seguintes métodos:**
 - o Dormir() - define EstaDormindo como true e imprime "O animal está dormindo"
 - o Acordar() - define EstaDormindo como false e imprime "O animal acordou"
 - o FazerSom() - imprime "O animal fez um som"
 - o ApresentarSe() - imprime informações do animal
3. **Crie 3 objetos diferentes:**
 - o Um gato chamado "Mimi", espécie "Felino", 3 anos
 - o Um cachorro chamado "Rex", espécie "Canino", 5 anos
 - o Um pássaro chamado "Piu", espécie "Ave", 1 ano
4. **Teste os métodos** de cada objeto e observe como cada um mantém seu próprio estado.

Exemplo de Código Base

```
public class Animal  
{
```



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

// Seus atributos aqui

// Construtor

```
public Animal(string nome, string especie, int idade)
```

```
{
```

```
    // Sua implementação aqui
```

```
}
```

// Seus métodos aqui

```
}
```

Tempo Estimado: 20 minutos

Atividade 2: Estado e Comportamento (Jogador de Futebol)

Objetivo

Entender como o estado de um objeto influencia seu comportamento e como os comportamentos modificam o estado.

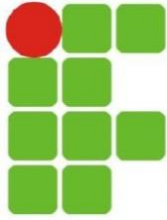
Instruções

1. **Crie uma classe JogadorFutebol** com os seguintes atributos:

- Nome (string)
- Energia (int - valor de 0 a 100, inicia com 100)
- Gols (int - inicia com 0)
- EstaJogando (bool - inicia como false)

2. **Implemente os seguintes métodos:**

- IniciarJogo() - define EstaJogando como true, imprime mensagem
- TerminarJogo() - define EstaJogando como false, imprime mensagem
- Correr() - diminui Energia em 10 (só se estiver jogando e energia > 0)



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

- FazerGol() - aumenta Gols em 1 e diminui Energia em 5 (só se estiver jogando)
- Descansar() - aumenta Energia em 20 (máximo 100, só se não estiver jogando)
- MostrarStatus() - exibe todas as informações do jogador

3. Regras de negócio:

- Jogador só pode correr ou fazer gol se estiver jogando
- Jogador só pode descansar se não estiver jogando
- Energia não pode ficar negativa ou passar de 100
- Não pode fazer ações se energia estiver em 0

4. Teste o comportamento:

- Crie um jogador, inicie um jogo
- Faça-o correr algumas vezes e fazer gols
- Termine o jogo e faça-o descansar
- Observe como o estado influencia as ações possíveis

Pergunta para Reflexão

Como o estado Energia afeta os comportamentos do jogador? O que acontece quando a energia chega a 0?

Tempo Estimado: 25 minutos

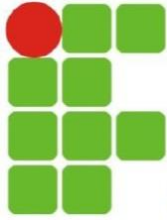
Atividade 3: Encapsulamento Básico (Conta Bancária Simples)

Objetivo

Aplicar conceitos de encapsulamento para proteger dados e controlar acesso através de métodos.

Instruções

1. **Crie uma classe ContaBancaria** com:
 - Atributo PRIVADO: saldo (decimal)



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

- Atributo PÚBLICO: NumeroConta (string)
- Atributo PÚBLICO: NomeTitular (string)

2. Implemente métodos públicos:

- Depositar(decimal valor) - adiciona valor ao saldo (só valores positivos)
- Sacar(decimal valor) - remove valor do saldo (só se houver saldo suficiente)
- ConsultarSaldo() - retorna o saldo atual
- ExibirExtrato() - mostra informações da conta e saldo

3. Regras de validação:

- Não permitir depósitos de valores negativos ou zero
- Não permitir saques maiores que o saldo
- Não permitir acesso direto ao saldo (deve ser privado)

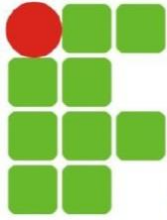
4. Teste sua implementação:

- Crie uma conta com saldo inicial
- Tente fazer depósitos válidos e inválidos
- Tente fazer saques válidos e inválidos
- Tente acessar o saldo diretamente (deve dar erro)

Exemplo de Estrutura

```
public class ContaBancaria
{
    private decimal saldo;

    public string NumeroConta { get; set; }
    public string NomeTitular { get; set; }
```



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

```
public ContaBancaria(string numeroConta, string nomeTitular, decimal
saldoInicial)
```

```
{
```

```
    // Sua implementação aqui
```

```
}
```

```
    // Seus métodos aqui
```

```
}
```

Pergunta para Reflexão

Por que é importante manter o saldo como privado? O que aconteceria se ele fosse público?

Tempo Estimado: 30 minutos

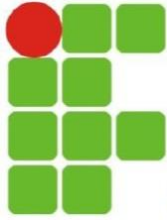
Atividade 4: Abstração com Interface Simples (Controle Remoto)

Objetivo

Entender abstração criando uma interface simples que esconde complexidade interna.

Instruções

1. **Crie uma classe ControleRemoto** que controlará uma TV, mas o usuário não precisa saber como funciona internamente.
2. **Atributos PRIVADOS:**
 - canalAtual (int - inicia em 1)
 - volumeAtual (int - inicia em 10)
 - estaLigada (bool - inicia como false)
 - canaisDisponiveis (int - fixo em 50 canais)
3. **Métodos PÚBLICOS (Interface Simples):**
 - LigarTV() - liga a TV



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

- DesligarTV() - desliga a TV
- AumentarVolume() - aumenta volume (máximo 30)
- DiminuirVolume() - diminui volume (mínimo 0)
- MudarCanal(int novoCanal) - muda para canal especificado (1-50)
- CanalAnterior() - volta ao canal anterior
- CanalPosterior() - vai para próximo canal
- MostrarStatus() - exibe estado atual da TV

4. Métodos PRIVADOS (Complexidade Oculta):

- ValidarCanal(int canal) - verifica se canal é válido (1-50)
- AjustarSinal() - simula ajuste de sinal (só imprime mensagem)
- VerificarEnergia() - simula verificação de energia

5. Regras:

- Todas as operações só funcionam se TV estiver ligada (exceto ligar)
- Volume tem limites (0-30)
- Canais têm limites (1-50)
- Ao mudar canal, sempre ajustar sinal automaticamente

Desafio Extra

Adicione um histórico dos últimos 5 canais assistidos (privado) e um método MostrarHistorico() (público).

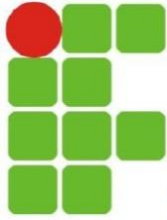
Pergunta para Reflexão

Como a abstração tornou o controle remoto mais fácil de usar? O que o usuário não precisa saber?

Tempo Estimado: 35 minutos

Atividade 5: Projeto Integrador (Sistema de Biblioteca)

Objetivo



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

Integrar todos os conceitos aprendidos (abstração, classes/objetos, atributos/métodos, estado/comportamento, encapsulamento) em um projeto mais completo.

Instruções

1. **Crie uma classe Livro** com:

- Atributos privados: titulo, autor, numeroPaginas, paginaAtual, estaEmprestado
- Métodos públicos para acessar e modificar dados com validação

2. **Crie uma classe Usuario** com:

- Atributos privados: nome, id, livrosEmprestados (lista)
- Métodos para gerenciar empréstimos

3. **Crie uma classe Biblioteca** que será a interface principal:

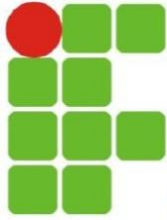
- Atributo privado: lista de livros disponíveis
- Métodos públicos: EmprestarLivro(), DevolverLivro(), ConsultarDisponibilidade()

Especificações Detalhadas

Classe Livro

```
public class Livro
{
    private string titulo;
    private string autor;
    private int numeroPaginas;
    private int paginaAtual;
    private bool estaEmprestado;

    // Propriedades públicas somente leitura
    public string Titulo { get { return titulo; } }
```



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

```
public string Autor { get { return autor; } }
```

```
public bool EstaEmprestado { get { return estaEmprestado; } }
```

```
// Construtor
```

```
// Métodos públicos: LerPaginas(), ReiniciarLeitura(), Emprestar(), Devolver()
```

```
// Método privado: ValidarPagina()
```

```
}
```

Funcionalidades Obrigatórias

1. Livro deve controlar:

- Páginas lidas (não pode passar do total)
- Status de empréstimo
- Progresso de leitura em porcentagem

2. Usuário deve poder:

- Pegar livros emprestados (máximo 3)
- Devolver livros
- Ver lista de livros que possui

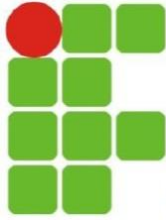
3. Biblioteca deve:

- Controlar estoque de livros
- Validar empréstimos (usuário não pode ter mais de 3 livros)
- Gerar relatórios simples

Teste Final

Crie um cenário onde:

1. A biblioteca tem 5 livros diferentes
2. 2 usuários fazem empréstimos
3. Um usuário lê algumas páginas de um livro



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

4. Um usuário tenta pegar mais de 3 livros
5. Gere um relatório do status da biblioteca

Perguntas para Reflexão

1. Como o encapsulamento protegeu os dados do sistema?
2. Onde você aplicou abstração para simplificar o uso?
3. Como o estado dos objetos influenciou os comportamentos disponíveis?
4. Quais validações foram necessárias e por quê?

Tempo Estimado: 45-60 minutos

Critérios de Avaliação para Todas as Atividades

Conceitos Aplicados Corretamente (70%)

- **Abstração:** Interface clara, complexidade oculta
- **Classes/Objetos:** Estrutura bem definida, objetos independentes
- **Atributos/Métodos:** Dados e comportamentos adequados
- **Estado/Comportamento:** Estado influencia comportamento corretamente
- **Encapsulamento:** Dados protegidos, acesso controlado

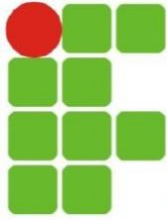
Implementação Técnica (20%)

- Código compila e executa
- Sintaxe C# correta
- Lógica de programação adequada

Teste e Validação (10%)

- Criação de cenários de teste
- Validação de regras de negócio
- Demonstração de funcionamento

Dicas Gerais



Atividade de Programação Orientada a Objetos

19/09/2025

Curso: **Tecnologia em Sistemas para Internet**

Modalidade: **Superior**

Ano/Período: **2025** Turno: **Vespertino**

Prof.: **Marcílio Meira**

Matrícula

Nome:

1. **Comece simples:** Implemente funcionalidades básicas primeiro
2. **Teste frequentemente:** Execute seu código a cada pequena adição
3. **Pense no usuário:** A interface pública deve ser intuitiva
4. **Proteja os dados:** Use private quando apropriado
5. **Valide entradas:** Sempre verifique se os dados estão corretos antes de usar

Recursos de Apoio

- Use Console.WriteLine() para debug e exibir informações
- Teste cada método individualmente antes de integrar
- Se houver dúvidas, revise os slides da apresentação
- Peça ajuda quando necessário - aprender é um processo colaborativo!