

Lista de Exercícios de C# - Do Fácil ao Difícil

FÁCIL (Básico e Lógica de Programação)

1. Imprimir 'Olá, Mundo!'.
2. Ler e exibir um número inteiro.
3. Somar dois números inteiros.
4. Calcular a média de três números.
5. Verificar se um número é par ou ímpar.
6. Calcular a área de um retângulo.
7. Calcular a área de um triângulo.
8. Calcular a área de um círculo.
9. Converter Celsius para Fahrenheit.
10. Calcular o perímetro de um quadrado.
11. Verificar se um número é positivo, negativo ou zero.
12. Verificar se o ano é bissexto.
13. Calcular a raiz quadrada de um número.
14. Verificar se um número é múltiplo de outro.
15. Trocar valores entre duas variáveis.
16. Calcular o fatorial de um número.
17. Imprimir a tabuada de um número.
18. Contar de 1 até 100.
19. Contar de 100 até 1.
20. Calcular a soma dos números de 1 a N.

MÉDIO (Estruturas, Vetores, Matrizes, Métodos)

1. Verificar se um número é primo.
2. Imprimir todos os números primos de 1 a 100.
3. Calcular a média de uma lista de números.
4. Calcular o maior e menor valor entre três números.
5. Criar uma função que retorne o maior valor entre dois números.
6. Criar um programa que inverta uma string.
7. Calcular a soma de todos os pares de 1 a 100.
8. Criar uma calculadora com as 4 operações.

9. Ler uma sequência de 10 números e contar os pares e ímpares.
10. Ordenar um vetor de inteiros.
11. Encontrar o maior valor em um vetor.
12. Buscar um número dentro de um vetor.
13. Contar quantas vezes um valor aparece em um vetor.
14. Calcular a média de cada linha de uma matriz 3x3.
15. Transpor uma matriz 3x3.
16. Implementar o jogo da velha em console.
17. Validar CPF.
18. Validar data (dd/mm/aaaa).
19. Implementar um relógio que atualiza a cada segundo.
20. Criar um menu com opções e repetir até o usuário escolher sair.

INTERMEDIÁRIO (POO, Arquivos, Exceções)

1. Criar uma classe Pessoa com nome e idade.
2. Criar uma classe Carro com propriedades e métodos.
3. Criar um método para verificar se uma pessoa é maior de idade.
4. Criar um sistema de cadastro de usuários com lista.
5. Implementar encapsulamento com propriedades.
6. Criar uma herança entre classes Animal, Cachorro e Gato.
7. Criar um método polimórfico para som emitido por animal.
8. Utilizar interface para definir comportamento comum.
9. Criar um sistema de banco com classes Conta, ContaCorrente, etc.
10. Implementar exceções personalizadas.
11. Ler e escrever em arquivos .txt.
12. Serializar e desserializar objetos em JSON.
13. Implementar um CRUD simples em console com lista em memória.
14. Salvar os dados de usuários em um arquivo JSON.
15. Criar um validador de senhas fortes.
16. Implementar log de erros em arquivo de texto.
17. Simular um sistema de login e senha com persistência.
18. Criar um conversor de moedas com base em taxas simuladas.
19. Calcular estatísticas de notas de alunos.
20. Simular uma fila de atendimento com Queue<T>.

AVANÇADO (LINQ, Async, Design Patterns, Projetos)

1. Filtrar uma lista de objetos com LINQ.
2. Ordenar uma lista com LINQ por múltiplos critérios.
3. Criar uma função que retorna os 5 maiores elementos de um vetor.
4. Agrupar dados com LINQ (por categoria, por ano, etc).
5. Implementar acesso assíncrono a arquivos.
6. Criar um método assíncrono para download de dados simulados.
7. Simular um chat entre usuários com tarefas assíncronas.
8. Criar uma calculadora polonesa reversa (RPN).
9. Implementar um sistema de votação com interface de console.
10. Criar um gerenciador de tarefas com prazos.
11. Usar delegates e eventos para simular um alarme.
12. Implementar um jogo simples (como forca ou adivinhação).
13. Criar um parser de expressões matemáticas.
14. Aplicar o padrão Strategy em um sistema de pagamento.
15. Aplicar o padrão Singleton em um logger.
16. Usar o padrão Factory para criação de objetos de forma flexível.
17. Simular uma loja virtual com produtos, carrinho e pedidos.
18. Criar uma API simples com ASP.NET Core (GET/POST).
19. Criar um sistema de estoque com atualização de quantidades.
20. Integrar um app console com uma API externa via HttpClient.
21. Gerar gráficos a partir de dados (via console ou interface).
22. Implementar logs com níveis (INFO, DEBUG, ERROR).
23. Criar um analisador léxico básico.
24. Criar um sistema de reservas com datas e horários.
25. Fazer scraping de um site simples usando HttpClient.

DESAFIOS EXTRAS E PROJETOS MAIORES

1. Implementar um sistema bancário completo.
2. Criar um gerenciador de biblioteca (livros, empréstimos, devoluções).
3. Desenvolver um mini ERP com produtos, vendas e estoque.
4. Criar um jogo de blackjack em console.
5. Implementar uma aplicação de gerenciamento de despesas pessoais.

6. Simular um caixa eletrônico com saques, depósitos e saldo.
7. Criar uma planilha virtual com funções básicas (soma, média, etc).
8. Desenvolver um controle de ponto de funcionários.
9. Criar um sistema de chat multiusuário com TCP/IP.
10. Implementar um reprodutor de música básico (interface e controle).
11. Desenvolver um projeto CRUD completo com interface gráfica (WinForms ou WPF).
12. Criar um gerador de senhas aleatórias com critérios de segurança.
13. Simular um jogo de dados com estatísticas de vitória.
14. Criar um sistema de agendamento com horários disponíveis.
15. Criar um sistema de quiz com perguntas e pontuação.
16. Criar um calendário interativo com feriados e eventos marcados.
17. Desenvolver um app console que interaja com um banco de dados SQLite.