

TRABALHO FINAL- ALGORÍTIMOS E TÉCNICAS DE PROGRAMAÇÃO

ALUNA: Maria Eduarda Wilken.

Enunciado:

Faça um programa gerencie o controle de vendas de quatro produtos. Para atender este propósito, Faça um programa gerencie o controle de vendas de quatro produtos. Para atender este propósito, siga as instruções abaixo:

- Ler um arquivo com o nome dos produtos e quantidade em estoque para armazenar em vetor.
- Registrar a quantidade diária de vendas durante um mês (registre em uma matriz).

o O usuário deverá digitar o número referente ao produto, o dia do mês e a quantidade vendida.

o Não permitir que a quantidade de vendas ultrapasse a quantidade em estoque

- Apresentar relatório com a quantidade em estoque atualizada.

Exemplo:

Produto A 152

Produto B 124

Produto C 85

Produto D 146

- Apresentar relatório com a quantidade vendida no mês.

Exemplo:

Produto A Produto B Produto C Produto D

Dia 1 5 7 2 0

Dia 2 2 5 1 8

Dia 3 3 6 9 1

...

Dia 30 0 3 6 7

- Gerar um arquivo com o total de vendas por produto.

Exemplo:

Produto A 20

Produto B 100

Produto C 40

Produto D 116

- Crie um menu principal com as opções para cada uma das ações citadas acima.

Exemplo:

1 – Importar arquivo de produtos

2 – Registrar venda

3 – Relatório de vendas

4 – Relatório de estoque

5 – Criar arquivo de vendas

6 - Sair

Comente o código indicando as principais decisões sobre os comandos escolhidos.

É importante criar métodos para as ações a serem realizadas.

Explore os comandos vistos em salas de aula.

EXECUÇÃO:

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.IO;
```

```
class Program
```

```
{
```

```
    static List<Produto> produtos = new List<Produto>();
```

```
    static int[,] vendas = new int[30, 4]; // Matriz de vendas para 30 dias e 4 produtos
```

```
static void Main(string[] args)
{
    int opcao;
    do
    {
        Console.WriteLine("Menu Principal:");
        Console.WriteLine("1 – Importar arquivo de produtos");
        Console.WriteLine("2 – Registrar venda");
        Console.WriteLine("3 – Relatório de vendas");
        Console.WriteLine("4 – Relatório de estoque");
        Console.WriteLine("5 – Criar arquivo de vendas");
        Console.WriteLine("6 – Sair");
        Console.Write("Escolha uma opção: ");
        opcao = int.Parse(Console.ReadLine());

        switch (opcao)
        {
            case 1:
                ImportarArquivoProdutos();
                break;
            case 2:
                RegistrarVenda();
                break;
            case 3:
                RelatorioVendas();
                break;
            case 4:
```

```

        RelatorioEstoque();

        break;

    case 5:

        CriarArquivoVendas();

        break;

    case 6:

        Console.WriteLine("Saindo...");

        break;

    default:

        Console.WriteLine("Opção inválida. Tente novamente.");

        break;

    }

} while (opcao != 6);

}

```

```

// Método para importar arquivo de produtos

static void ImportarArquivoProdutos()

{

    Console.Write("Digite o nome do arquivo de produtos: ");

    string nomeArquivo = Console.ReadLine();

    try

    {

        using (StreamReader sr = new StreamReader(nomeArquivo))

        {

            string linha;

            while ((linha = sr.ReadLine()) != null)

            {

```

```

        string[] partes = linha.Split(',');

        Produto produto = new Produto
        {
            Nome = partes[0].Trim(),
            Estoque = int.Parse(partes[1].Trim())
        };

        produtos.Add(produto);
    }
}

Console.WriteLine("Produtos importados com sucesso!");
}

catch (Exception e)
{
    Console.WriteLine("Erro ao importar o arquivo: " + e.Message);
}
}

```

// Método para registrar venda

```

static void RegistrarVenda()
{
    Console.Write("Digite o número do produto (1 a 4): ");
    int numeroProduto = int.Parse(Console.ReadLine()) - 1;

    Console.Write("Digite o dia do mês (1 a 30): ");
    int dia = int.Parse(Console.ReadLine()) - 1;

    Console.Write("Digite a quantidade vendida: ");
    int quantidade = int.Parse(Console.ReadLine());

    if (numeroProduto >= 0 && numeroProduto < 4 && dia >= 0 && dia < 30)

```

```

{
    if (quantidade <= produtos[numeroProduto].Estoque)
    {
        vendas[dia, numeroProduto] += quantidade;
        produtos[numeroProduto].Estoque -= quantidade;
        Console.WriteLine("Venda registrada com sucesso!");
    }
    else
    {
        Console.WriteLine("Quantidade vendida excede o estoque disponível.");
    }
}
else
{
    Console.WriteLine("Produto ou dia inválido.");
}
}

```

// Método para gerar relatório de vendas

```

static void RelatorioVendas()
{
    Console.WriteLine("Relatório de Vendas do Mês:");
    Console.WriteLine("Dia\tProduto A\tProduto B\tProduto C\tProduto D");
    for (int i = 0; i < 30; i++)
    {
        Console.Write((i + 1) + "\t");
        for (int j = 0; j < 4; j++)
        {

```

```
        Console.Write(vendas[i, j] + "\t\t");  
    }  
    Console.WriteLine();  
}  
}
```

// Método para gerar relatório de estoque

```
static void RelatorioEstoque()  
{  
    Console.WriteLine("Relatório de Estoque Atualizado:");  
    foreach (Produto produto in produtos)  
    {  
        Console.WriteLine(produto.Nome + ": " + produto.Estoque);  
    }  
}
```

// Método para criar arquivo de vendas

```
static void CriarArquivoVendas()  
{  
    string nomeArquivo = "RelatorioVendas_" +  
DateTime.Now.ToString("yyyyMMdd_HH:mm:ss") + ".txt";  
  
    try  
    {  
        using (StreamWriter sw = new StreamWriter(nomeArquivo))  
        {  
            for (int i = 0; i < produtos.Count; i++)  
            {
```

```

        int totalVendas = 0;

        for (int j = 0; j < 30; j++)
        {
            totalVendas += vendas[j, i];
        }

        sw.WriteLine(produtos[i].Nome + ": " + totalVendas);
    }
}

Console.WriteLine("Arquivo de vendas criado com sucesso: " +
nomeArquivo);

}

catch (Exception e)
{
    Console.WriteLine("Erro ao criar o arquivo: " + e.Message);
}

}
}

```

```

class Produto
{
    public string Nome { get; set; }
    public int Estoque { get; set; }
}

```