


```

        telefone = Console.ReadLine();

        pj = new PessoaJuridica();
        clientesPJ[codigo - 1] = pj.setPessoaJuridica(codigo, razaoSocial,
cnpj, endereco, telefone);

        codigo++;

        Console.WriteLine("Cliente inserido com sucesso!");

        break;
    default:
        Console.WriteLine("Este tipo de cliente não existe!");
        break;
    }
    Console.WriteLine();
    break;
case 2:
    Console.WriteLine("O cliente é PF ou PJ?");
    tipo_cliente = Console.ReadLine();

    bool remocao = false;

    switch (tipo_cliente) {
        case "pf":
            Console.WriteLine("Informe o código do cliente: ");
            cod = int.Parse(Console.ReadLine());

            for (int i = 0; i < codigo; i++)
                if (clientesPF[i] != null)
                    if (pf.getCodigoVetorPF(clientesPF[i]) == cod) {
                        clientesPF[i] = null;
                        remocao = true;
                        break;
                    }
            break;
        case "pj":
            Console.WriteLine("Informe o código do cliente: ");
            cod = int.Parse(Console.ReadLine());

            for (int i = 0; i < codigo; i++)
                if (clientesPJ[i] != null)
                    if (pj.getCodigoVetorPJ(clientesPJ[i]) == cod) {
                        clientesPJ[i] = null;
                        remocao = true;
                        break;
                    }
            break;
        default:
            Console.WriteLine("Este tipo de cliente não existe!");
            Console.WriteLine();
            break;
    }

    if (remocao)
        Console.WriteLine("Cliente removido com sucesso!");
    else
        Console.WriteLine("Este cliente não existe!");

    Console.WriteLine();
    break;
case 3:
    bool existe_pf = false;

    for (int i = 0; i < codigo - 1; i++) {
        if (clientesPF[i] != null) {
            existe_pf = true;
            break;
        }
    }
}

```

```

        if (existe_pf == true)
            Console.WriteLine("PESSOAS FÍSICAS:");
        else
            Console.WriteLine("PESSOAS FÍSICAS: Nenhum registro encontrado!");

        pf.getPessoaFisica(codigo, clientesPF);

        //-----
        bool existe_pj = false;

        for (int i = 0; i < codigo - 1; i++) {
            if (clientesPJ[i] != null) {
                existe_pj = true;
                break;
            }
        }

        if (existe_pj == true)
            Console.WriteLine("PESSOAS JURÍDICAS:");
        else
            Console.WriteLine("PESSOAS JURÍDICAS: Nenhum registro encontrado!");

        pj.getPessoaJuridica(codigo, clientesPJ);

        Console.WriteLine();
        break;
    default:
        if (opcao != 0)
            Console.WriteLine("Opção inválida!");
        else {
            Console.WriteLine("Programa encerrado!");
            Console.ReadKey();
        }
        Console.WriteLine();
        break;
    }
} while (opcao != 0);
}

public class Cliente {
    private int codigo;
    private string endereco;
    private string telefone;

    protected void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    protected void setEndereco(string endereco) {
        this.endereco = endereco;
    }

    protected void setTelefone(string telefone) {
        this.telefone = telefone;
    }

    protected int getCodigo() {
        return this.codigo;
    }

    protected string getEndereco() {
        return this.endereco;
    }

    protected string getTelefone() {
        return this.telefone;
    }
}

```

```

}

class PessoaFisica : Cliente {
    private string nome;
    private string cpf;

    public void setNome(string nome) {
        this.nome = nome;
    }

    public void setCpf(string cpf) {
        this.cpf = cpf;
    }

    public string getNome() {
        return this.nome;
    }

    public string getCpf() {
        return this.cpf;
    }

    public PessoaFisica setPessoaFisica(int codigo, string nome, string cpf, string endereco,
string telefone) {
        this.setCodigo(codigo);
        this.setNome(nome);
        this.setCpf(cpf);
        this.setEndereco(endereco);
        this.setTelefone(telefone);
        return this;
    }

    public void getPessoaFisica(int codigo, PessoaFisica[] clientesPF) {
        for (int i = 0; i < codigo - 1; i++) {
            if (clientesPF[i] != null)
                Console.WriteLine(clientesPF[i].getCodigo() + " - " +
                    clientesPF[i].getNome() + " - " +
                    clientesPF[i].getCpf() + " - " +
                    clientesPF[i].getEndereco() + " - " +
                    clientesPF[i].getTelefone());
        }
    }

    public int getCodigoVetorPF(PessoaFisica clientesPF) {
        return clientesPF.getCodigo();
    }
}

class PessoaJuridica : Cliente {
    private string razaoSocial;
    private string cnpj;

    public void setRazaoSocial(string razaoSocial) {
        this.razaoSocial = razaoSocial;
    }

    public void setCnpj(string cnpj) {
        this.cnpj = cnpj;
    }

    public string getRazaoSocial() {
        return this.razaoSocial;
    }

    public string getCnpj() {
        return this.cnpj;
    }
}

```

```

        public PessoaJuridica setPessoaJuridica(int codigo, string razaoSocial, string cnpj, string
endereco, string telefone) {
            this.setCodigo(codigo);
            this.setRazaoSocial(razaoSocial);
            this.setCnpj(cnpj);
            this.setEndereco(endereco);
            this.setTelefone(telefone);
            return this;
        }

        public void getPessoaJuridica(int codigo, PessoaJuridica[] clientesPJ) {
            for (int i = 0; i < codigo - 1; i++) {
                if (clientesPJ[i] != null)
                    Console.WriteLine(clientesPJ[i].getCodigo() + " - " +
                                     clientesPJ[i].getRazaoSocial() + " - " +
                                     clientesPJ[i].getCnpj() + " - " +
                                     clientesPJ[i].getEndereco() + " - " +
                                     clientesPJ[i].getTelefone());
            }
        }

        public int getCodigoVetorPJ(PessoaJuridica clientesPJ) {
            return clientesPJ.getCodigo();
        }
    }
}

```

Exercício 2

```

namespace Ex05_02
{
    class Inicio
    {
        static void Main(string[] args)
        {
            Venda v1 = new Venda();
            for (int i = 0; i < 6; i++) {
                Console.Write((i + 1) + "º Mês: ");
                v1.setVenda(double.Parse(Console.ReadLine()));
            }

            Venda v2 = new Venda();
            Console.WriteLine("Soma: " + v2.getVenda().ToString("0.00"));
            Console.ReadKey();
        }
    }

    class Venda {
        private static double vendaTotal = 0;

        public void setVenda(double vendaMes) {
            vendaTotal = vendaTotal + vendaMes;
        }

        public double getVenda() {
            return vendaTotal;
        }
    }
}

```

Exercício 3

```

namespace Ex05_03 {
    class Inicio {
        static void Main(string[] args) {
            int opcao;
            double numD, b, e;
            int numI;

```

```

do {
    Console.WriteLine("OPÇÕES:");
    Console.WriteLine("1) Calcular raiz quadrada");
    Console.WriteLine("2) Calcular potenciação");
    Console.WriteLine("3) Calcular fatorial");
    Console.Write("Digite sua opção: ");
    opcao = Int32.Parse(Console.ReadLine());

    switch (opcao) {
        case 1:
            Console.Write("Informe um número: ");
            numD = Double.Parse(Console.ReadLine());
            Console.WriteLine("Resultado: " + Calculo.ExecutarCalculo(numD));
            Console.WriteLine();
            break;
        case 2:
            Console.Write("Informe a base: ");
            b = Double.Parse(Console.ReadLine());
            Console.Write("Informe o expoente: ");
            e = Double.Parse(Console.ReadLine());
            Console.WriteLine("Resultado: " + Calculo.ExecutarCalculo(b, e));
            Console.WriteLine();
            break;
        case 3:
            Console.Write("Informe um número: ");
            numI = Int32.Parse(Console.ReadLine());
            Console.WriteLine("Resultado: " + Calculo.ExecutarCalculo(numI));
            Console.WriteLine();
            break;
        default:
            if (opcao != 0)
                Console.WriteLine("Opção inválida!");
            else {
                Console.WriteLine("Programa encerrado!");
                Console.ReadKey();
            }
            Console.WriteLine();
            break;
    }
} while (opcao != 0);
}

class Calculo {
    // Cálculo de raiz quadrada.
    public static double ExecutarCalculo(double num) {
        return Math.Sqrt(num);
    }

    // Cálculo de potenciação.
    public static double ExecutarCalculo(double b, double e) {
        return Math.Pow(b, e);
    }

    // Cálculo de fatorial.
    public static int ExecutarCalculo(int num) {
        int fatorial = 1;
        for (int i = 1; i <= num; i++)
            fatorial = fatorial * i;
        return fatorial;
    }
}
}

```

Exercício 4

```

namespace Ex05_04 {
    class Inicio {
        static void Main(string[] args) {

```

```

int opcao;
String texto = "";
String cadeiaCaracteres = "";
Pesquisa p = new Pesquisa();

do {
    Console.WriteLine("OPÇÕES:");
    Console.WriteLine("1) Informar texto");
    Console.WriteLine("2) Buscar string");
    Console.WriteLine("3) Buscar string no início");
    Console.WriteLine("4) Buscar string no fim");
    Console.Write("Digite sua opção: ");
    opcao = int.Parse(Console.ReadLine());

    switch (opcao) {
        case 1:
            Console.Write("Informe um texto: ");
            texto = Console.ReadLine();
            p = new Pesquisa();
            p.setTexto(texto);
            break;
        case 2:
            Console.Write("Informe uma cadeia de caracteres: ");
            cadeiaCaracteres = Console.ReadLine();
            p = new Pesquisa();
            if (!texto.Equals("")) {
                if (p.buscarString(cadeiaCaracteres))
                    Console.WriteLine("Cadeia de caracteres encontrada!");
                else
                    Console.WriteLine("Cadeia de caracteres não encontrada!");
            } else
                Console.WriteLine("Use a opção 1 para informar um texto!");
            break;
        case 3:
            Console.Write("Informe uma cadeia de caracteres: ");
            cadeiaCaracteres = Console.ReadLine();
            p = new PesquisaInicio();
            if (!texto.Equals("")) {
                if (p.buscarString(cadeiaCaracteres))
                    Console.WriteLine("Cadeia de caracteres encontrada no início!");
                else
                    Console.WriteLine("Cadeia de caracteres não encontrada no
início!");
            } else
                Console.WriteLine("Use a opção 1 para informar um texto!");
            break;
        case 4:
            Console.Write("Informe uma cadeia de caracteres: ");
            cadeiaCaracteres = Console.ReadLine();
            p = new PesquisaFim();
            if (!texto.Equals("")) {
                if (p.buscarString(cadeiaCaracteres))
                    Console.WriteLine("Cadeia de caracteres encontrada no fim!");
                else
                    Console.WriteLine("Cadeia de caracteres não encontrada no fim!");
            } else
                Console.WriteLine("Use a opção 1 para informar um texto!");
            break;
        default:
            if (opcao != 0)
                Console.WriteLine("Opção inválida!");
            else {
                Console.WriteLine("Programa encerrado!");
                Console.ReadKey();
            }
            Console.WriteLine();
            break;
    }
} while (opcao != 0);

```

```

    }
}

class Pesquisa {
    private static String texto;

    public String getTexto() {
        return texto;
    }

    public void setTexto(String text) {
        texto = text;
    }

    public virtual bool buscarString(String cadeiaCaracteres) {
        if (texto.Contains(cadeiaCaracteres))
            return true;
        else
            return false;
    }
}

class PesquisaFim : Pesquisa {
    public sealed override bool buscarString(String cadeiaCaracteres) {
        if (getTexto().EndsWith(cadeiaCaracteres))
            return true;
        else
            return false;
    }
}

class PesquisaInicio : Pesquisa {
    public sealed override bool buscarString(String cadeiaCaracteres) {
        if (getTexto().StartsWith(cadeiaCaracteres))
            return true;
        else
            return false;
    }
}
}

```