

# QUESTÃO 1

```
public class Fila {

    private String[] array;
    private int ultimo, primeiro;

    public Fila() {
        this(6);
    }

    public Fila(int tamanho) {
        array = new String[tamanho + 1];
        primeiro = 0;
        ultimo = 0;
    }

    public void inserir(String x) throws Exception {
        if (((ultimo + 1) % array.length) == primeiro) {
            throw new Exception("Erro!");
        }
        array[ultimo] = x;
        ultimo = (ultimo + 1) % array.length;
    }

    public String remover() throws Exception {
        if (primeiro == ultimo) {
            throw new Exception("Erro!");
        }
        String resp = array[primeiro];
        primeiro = (primeiro + 1) % array.length;
        return resp;
    }

    public void mostrar() {
        int i = primeiro;
        System.out.print("\n[");
        while (i != ultimo) {
            System.out.print(array[i] + " ");
            i = (i + 1) % array.length;
        }

        System.out.println("]");
    }

    public boolean pesquisar(String cliente) {
```

```

        boolean testa = false;
        for (int j = 0; j < array.length; j++) {
            if (cliente.equals(array[j])) {
                testa = true;
            }
        }
        return testa;
    }

    public boolean isVazia() {
        boolean testa = false;
        if (primeiro == ultimo) {
            testa = true;
        }
        return testa;
    }
}

```

```

public class Clientes {

    private String nome;
    private int cfp;
    private String endereco;
    private int telefone;

    public Clientes(String nome, int cpf, String endereco, int telefone)
    {
        this.nome = nome;
        this.cfp = cpf;
        this.endereco = endereco;
        this.telefone = telefone;
    }

    public void setNome(String nome) {

        this.nome = nome;
    }

    public String getNome() {
        return nome;
    }
}

```

```

    public void setCPF(int cpf) {
        this.cfp = cpf;
    }

    public int getCPF() {
        return cfp;
    }

    public void setEndereco(String endereco) {
        this.endereco = endereco;
    }

    public String getEndereco() {
        return endereco;
    }

    public void setTelefone(int telefone){
        this.telefone = telefone;
    }
    public int getTelefone(){
        return telefone;
    }
}

import java.util.*;

public class GerenciarClientes {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);

        Fila fi = new Fila(6);
        Clientes cli = new Clientes("Eduardo", 123456789, "Rua Leona,
Bairro Lurdes", 93356745);
        int op = 0;
        String nomes;

        do {

            System.out.println("\nMenu:");
            System.out.println("1) Inserir cliente na fila de espera");
            System.out.println("2) Remover um cliente da fila de
espera");
            System.out.println("3) Listar os nomes dos clientes que estão
na fila de espera");

```

```

espera");
        System.out.println("4) Pesquisar se o cliente está na fila de
espera");
        System.out.println("5) Verificar se a fila está vazia");
        System.out.println("6) Encerrar o programa");
        System.out.print("Opção:");
        op = sc.nextInt();

        switch (op) {
            case 1:
                System.out.println("\nDigite o nome");
                nomes = sc.next();
                fi.inserir(nomes);
                break;

            case 2:
                fi.remover();
                break;

            case 3:
                fi.mostrar();
                break;

            case 4:
                System.out.println("Digite o nome a pesquisar");
                nomes = sc.next();
                fi.pesquisar(nomes);
                if (fi.pesquisar(nomes)) {
                    System.out.println("Esse nome está na lista");
                } else {
                    System.out.println("Esse nome não existe na
lista");
                }
                break;

            case 5:
                if (fi.isVazia()) {
                    System.out.println("Fila Vazia");
                } else {
                    System.out.println("Fila contém elementos");
                }
                break;

            default:
                System.out.println("opção Inválida");
                break;
        }

    } while (op != 6);
}

```

```
}  
}
```

## QUESTÃO 3

```
import java.util.*;  
  
public class GerenciarClientes {  
    public static void main(String[] args) throws Exception {  
        Scanner sc = new Scanner(System.in);  
  
        int op;  
        String nome, endereco, cpf, telefone;  
  
        Fila fi = new Fila(6);  
  
        do {  
            System.out.println("\n Menu");  
            System.out.println("1- Para pesquisar se existe o nome");  
            System.out.println("2- Mostrar Filas de Objetos");  
            System.out.println("3- Para Inserir Objeto");  
            System.out.println("4- Para remover Objeto");  
            System.out.println("5- Remover Objeto");  
            System.out.print("Opção:");  
            op = sc.nextInt();  
  
            switch (op) {  
                case 1:  
                    System.out.println("\n Digite o nome");  
                    nome = sc.next();  
  
                    if (fi.pesquisar(nome)) {  
                        System.out.println("\n Objeto com nome " + nome +  
" encontrado!");  
                    } else {  
                        System.out.println("Esse Objeto não Existe!");  
                    }  
                    break;  
  
                case 2:  
                    fi.mostrar();  
                    break;  
  
                case 3:  
                    sc.nextLine();  
                    System.out.print("Nome:");
```

```

        nome = sc.nextLine();

        System.out.print("CPF:");
        cpf = sc.nextLine();

        System.out.print("Endereço:");
        endereco = sc.nextLine();

        System.out.print("Telefone:");
        telefone = sc.nextLine();

        Clientes clienteNovo = new Clientes(nome, cpf,
endereco, telefone);

        fi.inserir(clienteNovo);
        break;

    case 4:
        fi.remover();
        break;

    default:
        System.out.println("opção Inválida");
        break;
    }

    } while (op != 5);

    sc.close();
}
}

```

```

public class Fila {

    private Clientes[] array;
    private int ultimo, primeiro;

    public Fila() {
        this(6);
    }

    public Fila(int tamanho) {
        array = new Clientes[tamanho + 1];
        primeiro = 0;
        ultimo = 0;
    }

    public void inserir(Clientes x) throws Exception {

```

```

        if (((ultimo + 1) % array.length) == primeiro) {
            throw new Exception("Erro!");
        }
        array[ultimo] = x;
        ultimo = (ultimo + 1) % array.length;
    }

    public Clientes remover() throws Exception {
        if (primeiro == ultimo) {
            throw new Exception("Erro!");
        }
        Clientes resp = array[primeiro];
        primeiro = (primeiro + 1) % array.length;
        return resp;
    }

    public void mostrar() {
        int i = primeiro;
        System.out.print("\n");
        while (i != ultimo) {
            System.out.print(array[i].toString() + " \n");
            i = (i + 1) % array.length;
        }
    }

    public boolean pesquisar(String cliente) {

        boolean testar = false;
        for (int j = primeiro; j != ultimo; j = (j + 1) % array.length) {
            if (cliente.equals(array[j].getNome())) {
                testar = true;
            }
        }
        return testar;
    }

    public boolean isVazia() {
        boolean testa = false;
        if (primeiro == ultimo) {
            testa = true;
        }
        return testa;
    }
}

```

```
public class Clientes {

    private String nome;
    private String cfp;
    private String endereco;
    private String telefone;

    public Clientes(String nome, String cpf, String endereço, String
telefone) {
        this.nome = nome;
        this.cfp = cpf;
        this.endereco = endereço;
        this.telefone = telefone;
    }

    public void setNome(String nome) {

        this.nome = nome;
    }

    public String getNome() {
        return nome;
    }

    public void setCPF(String cpf) {
        this.cfp = cpf;
    }

    public String getCPF() {
        return cfp;
    }

    public void setEndereco(String endereço) {
        this.endereco = endereço;
    }

    public String getEndereco() {
        return endereco;
    }

    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }

    public String getTelefone() {
        return telefone;
    }
}
```



```

@Override
public String toString() {
    return "Clientes [nome=" + nome + ", cfp=" + cfp + ", endereco="
+ endereco + ", telefone=" + telefone + "];"
}

}

```

## Questão 4

```

import java.util.Scanner;

public class GerenciarPilha {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);

        int num1 = 1, num2 = 1;

        System.out.print("Digite um valor inteiro positivo n: ");
        int n = sc.nextInt();

        Pilha p = new Pilha(n);

        p.inserirFim(1);
        p.inserirFim(1);

        for (int i = 2; i < n; i++) {
            int fib = num1 + num2;
            p.inserirFim(fib);
            num1 = num2;
            num2 = fib;
        }
        p.mostrarPilhaInvertida();

        sc.close();
    }
}

public class Pilha {
    private int[] array;
    private int n;

```

```

public Pilha() {
    this(6);
}

public Pilha(int tamanho) {
    array = new int[tamanho];
    n = 0;
}

public void inserirFim(int x) throws Exception {
    if (n >= array.length)
        throw new Exception("Erro!");
    array[n] = x;
    n++;
}

public int removerFim() throws Exception {
    if (n == 0)
        throw new Exception("Erro!");

    return array[--n];
}

public void mostrarPilhaInvertida() {

    for (int i = n-1; i >= 0 ; i--) {
        System.out.print(array[i] + " ");
    }

}
}

```

## **Questão 5**

```

import java.util.Scanner;

public class notacaoPolonesa {

    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);

        String expressao;
    }
}

```

```

        System.out.print("Digite a expressão em notação polonesa reversa:
");
        expressao = sc.nextLine();

        Pilha pi = new Pilha(5);

        for (int i = 0; i < expressao.length(); i++) {
            char c = expressao.charAt(i);

            if (c == '+' || c == '-' || c == '*' || c == '/') {

                double a = pi.removerFim();
                double b = pi.removerFim();

                double resultado = 0;

                switch (c) {
                    case '+':
                        resultado = b + a;
                        break;
                    case '-':
                        resultado = b - a;
                        break;
                    case '*':
                        resultado = b * a;
                        break;
                    case '/':
                        resultado = b / a;
                        break;
                }

                pi.inserirFim(resultado);

            } else {
                int valorFinal = Character.getNumericValue(c);
                pi.inserirFim(valorFinal);
            }
        }

        pi.mostrarPilha();
    }
}

public class Pilha {

    private double[] array;
    private int n;

```

```
public Pilha() {
    this(6);
}

public Pilha(int tamanho) {
    array = new double[tamanho];
    n = 0;
}

public void inserirFim(double x) throws Exception {
    if (n >= array.length)
        throw new Exception("Erro!");
    array[n] = x;
    n++;
}

public double removerFim() throws Exception {
    if (n == 0)
        throw new Exception("Erro!");

    return array[--n];
}

public void mostrarPilha() {

    for (int i = 0; i < n; i++) {
        System.out.print(array[i] + " ");
    }

}

}
```