

Questão 1 - Classe

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Questão1ComCelula
{
    class Valor
    {
        public Valor(String valor)
        {
            obj = valor;
        }
        private string obj;

        public string Obj
        {
            get { return obj; }
            set { obj = value; }
        }
        public void Imprimir()
        {
            Console.WriteLine("Valor: {0}", obj);
        }
    }
    class Celula
    {
        private Valor valor;
        private Celula proximo;
        public Celula()
        {
            valor = null;
            proximo = null;
        }
        public Celula(Valor valor)
        {
            this.valor = valor;
            proximo = null;
        }
    }
}
```

```

public Celula Proximo
{
    get { return proximo; }
    set { proximo = value; }
}
public Valor Valor
{
    get { return valor; }
    set { valor = value; }
}
}
class Fila
{
    private Celula frente;
    private Celula tras;
    public Fila()
    {
        Celula sentinela = new Celula();
        frente = sentinela;
        tras = sentinela;
    }
    public bool Vazia()
    {
        if (frente == tras)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    public void Enqueue(Valor obj)
    {
        Celula aux = new Celula(obj);
        tras.Proximo = aux;
        tras = aux;
    }
    public Valor Dequeue()
    {
        Celula removida;
        Celula nextCelula;
        if (!Vazia())
        {

```

```

        removida = frente.Proximo;
        nextCelula = removida.Proximo;
        if (removida == tras)
        {
            tras = frente;
        }
        return removida.Valor;
    }
    else
    {
        throw new Exception("Fila Vazia!");
    }
}
public void Imprimir()
{
    Celula aux;
    if (!Vazia())
    {
        aux = frente.Proximo;
        while (aux != null)
        {
            aux.Valor.Imprimir();
            aux = aux.Proximo;
        }
    }
    else
    {
        throw new Exception("Fila Vazia!");
    }
}
public int Count()
{
    Celula aux;
    int cont = 0;
    if (!Vazia())
    {
        aux = frente.Proximo;
        while (aux != null)
        {
            cont++;
            aux = aux.Proximo;
        }
    }
    else

```

```

        {
            throw new Exception("Fila Vazia!");
        }
        return cont;
    }
    public Valor Peek()
    {
        return frente.Valor;
    }
}
}

```

Questão 1 - Main

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Questão1ComCelula
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Fila fila = new Fila();
            int escolha;
            do
            {
                Console.WriteLine("1. Listar a quantidade de aviões que estão
aguardando na fila de decolagem");
                Console.WriteLine("2. Autorizar a decolagem do primeiro avião da fila de
decolagem (isto é, retirar da fila de colagem)");
                Console.WriteLine("3. Adicionar um avião na fila de colagem");
                Console.WriteLine("4. Listar todos os aviões que estão na fila de
colagem");
                Console.WriteLine("5. Exibir o primeiro avião da fila de colagem");
                Console.WriteLine("6. Sair");
                escolha = int.Parse(Console.ReadLine());
                switch (escolha)
                {
                    case 1:

```

```
        Console.WriteLine(fila.Count()); //Não usei try-catch por já ter o throw
exception na parte da classe.
```

```
        break;
    case 2:
        Console.WriteLine("O avião {0} decolou", fila.Dequeue().Obj);
        break;
    case 3:
        Console.WriteLine("Insira o modelo do avião que deseja inserir: ");
        string aviao = Console.ReadLine();
        Valor obj = new Valor(aviao);
        fila.Enqueue(obj);
        break;
    case 4:
        fila.Imprimir();
        break;
    case 5:
        Console.WriteLine("O avião que está no topo é: {0}", fila.Peek().Obj);
        break;
    default:
        break;
    }
} while (escolha != 0);
}
}
```

Questão 2 - Classe Fila

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace Questão2ComCelula
{
    class Valor
    {
        public Valor(Pagina valor)
        {
            obj = valor;
        }
        private Pagina obj;
    }
}
```

```

public Pagina Obj
{
    get { return obj; }
    set { obj = value; }
}
public void Imprimir()
{
    Console.WriteLine("Valor: " + obj.Arquivo + ":" + obj.Numero);
}
}
class Celula
{
    private Valor valor;
    private Celula proximo;
    public Celula()
    {
        valor = null;
        proximo = null;
    }
    public Celula(Valor valor)
    {
        this.valor = valor;
        proximo = null;
    }
    public Celula Proximo
    {
        get { return proximo; }
        set { proximo = value; }
    }
    public Valor Valor
    {
        get { return valor; }
        set { valor = value; }
    }
}
class Fila
{
    private Celula frente;
    private Celula tras;
    public Fila()
    {
        Celula sentinela = new Celula();
        frente = sentinela;
        tras = sentinela;
    }
}

```

```

}
public bool Vazia()
{
    if (frente == tras)
    {
        return true;
    }
    else
    {
        return false;
    }
}
public void Enqueue(Valor obj)
{
    Celula aux = new Celula(obj);
    tras.Proximo = aux;
    tras = aux;
}
public Valor Dequeue()
{
    Celula removida;
    Celula nextCelula;
    if (!Vazia())
    {
        removida = frente.Proximo;
        nextCelula = removida.Proximo;
        if (removida == tras)
        {
            tras = frente;
        }
        return removida.Valor;
    }
    else
    {
        throw new Exception("Fila Vazia!");
    }
}
public void Imprimir()
{
    Celula aux;
    if (!Vazia())
    {
        aux = frente.Proximo;
        while (aux != null)

```

```

        {
            aux.Valor.Imprimir();
            aux = aux.Proximo;
        }
    }
    else
    {
        throw new Exception("Fila Vazia!");
    }
}
public int Count()
{
    Celula aux;
    int cont = 0;
    if (!Vazia())
    {
        aux = frente.Proximo;
        while (aux != null)
        {
            cont++;
            aux = aux.Proximo;
        }
    }
    else
    {
        throw new Exception("Fila Vazia!");
    }
    return cont;
}
public Valor Peek()
{
    return frente.Valor;
}
}
}

```

Questão 2 - Classe Página

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```



```

namespace Questão2ComCelula
{
    class Pagina
    {
        private int numero;
        public int Numero
        {
            get { return numero; }
            set { numero = value; }
        }
        private string arquivo;
        public string Arquivo
        {
            get { return arquivo; }
            set { arquivo = value; }
        }
        public Pagina(int numero, string arquivo)
        {
            this.arquivo = arquivo;
            this.numero = numero;
        }
    }
}

```

Questão 2 - Main

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Questão2ComCelula
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Fila fila = new Fila();
            int escolha;
            do
            {

```

```

Console.WriteLine("1. Inserir arquivo na fila de impressão");
Console.WriteLine("2. Executar impressão");
Console.WriteLine("3. Exibir fila de impressão");
Console.WriteLine("4. Sair");
escolha = int.Parse(Console.ReadLine());
switch (escolha)
{
    case 1:
        Console.WriteLine("Insira o arquivo e a quantidade de arquivos a ser
imprimido");
        string arquivo = Console.ReadLine();
        int qnt = int.Parse(Console.ReadLine());
        Pagina pagina = new Pagina(qnt, arquivo);
        Valor valor = new Valor(pagina);
        fila.Enqueue(valor);
        break;
    case 2:
        Valor imprimir = fila.Dequeue();
        Console.WriteLine("O arquivo: {0}, impressões: {1} foi retirado",
imprimir.Obj.Arquivo, imprimir.Obj.Numero);
        break;
    case 3:
        fila.Imprimir();
        break;
    default:
        break;
}
} while (escolha != 4);
Console.ReadLine();
}
}
}

```