

Instituto Federal Catarinense (*Campus Blumenau*)

Professor: Ricardo de La Rocha Ladeira

Matéria: Padrões de Projeto

Nomes: Gabrielli Danker

Turma: BCC 2025.1

Data de entrega: 10 de Abril de 2025

Exercícios

- 1) Simule a implementação de um sistema de banco de dados onde um *proxy* deve controlar o número máximo de conexões simultâneas. Se o limite for atingido, novas conexões devem ser recusadas. Assuma que o limite é de três conexões.

RESPOSTA:

```
public interface ConexaoBanco {
    void conectar();
    void desconectar();
}

public class ConexaoReal implements ConexaoBanco {
    private String nome;

    public ConexaoReal(String nome) {
        this.nome = nome;
    }

    @Override
    public void conectar() {
        System.out.println(nome + " conectado ao banco de
dados.");
    }

    @Override
    public void desconectar() {
        System.out.println(nome + " desconectado do banco de
dados.");
    }
}

import java.util.ArrayList;
import java.util.List;

public class ConexaoProxy implements ConexaoBanco {
    private static final int LIMITE_CONEXOES = 3;
```

```

        private static int conexoesAtivas = 0;
        private static List<ConexaoReal> conexoes = new
ArrayList<>();
        private ConexaoReal conexaoReal;

        private String nome;

        public ConexaoProxy(String nome) {
            this.nome = nome;
        }

        @Override
        public void conectar() {
            if (conexoesAtivas < LIMITE_CONEXOES) {
                conexaoReal = new ConexaoReal(nome);
                conexaoReal.conectar();
                conexoes.add(conexaoReal);
                conexoesAtivas++;
            } else {
                System.out.println(nome + " não pôde conectar: limite
de conexões atingido.");
            }
        }

        @Override
        public void desconectar() {
            if (conexaoReal != null) {
                conexaoReal.desconectar();
                conexoes.remove(conexaoReal);
                conexoesAtivas--;
            }
        }
    }

    public class DemoProxy {
        public static void main(String[] args) {
            ConexaoBanco conexao1 = new ConexaoProxy("Usuário1");
            ConexaoBanco conexao2 = new ConexaoProxy("Usuário2");
            ConexaoBanco conexao3 = new ConexaoProxy("Usuário3");
            ConexaoBanco conexao4 = new ConexaoProxy("Usuário4");

            conexao1.conectar();
            conexao2.conectar();

```

```

        conexao3.conectar();

        // Esta deve ser recusada
        conexao4.conectar();

        // Libera uma conexão
        conexao2.desconectar();

        // Tenta conectar novamente
        conexao4.conectar();
    }
}

```

O código está na pasta Proxy -> Questão 1.

Ele retorna isso:

```

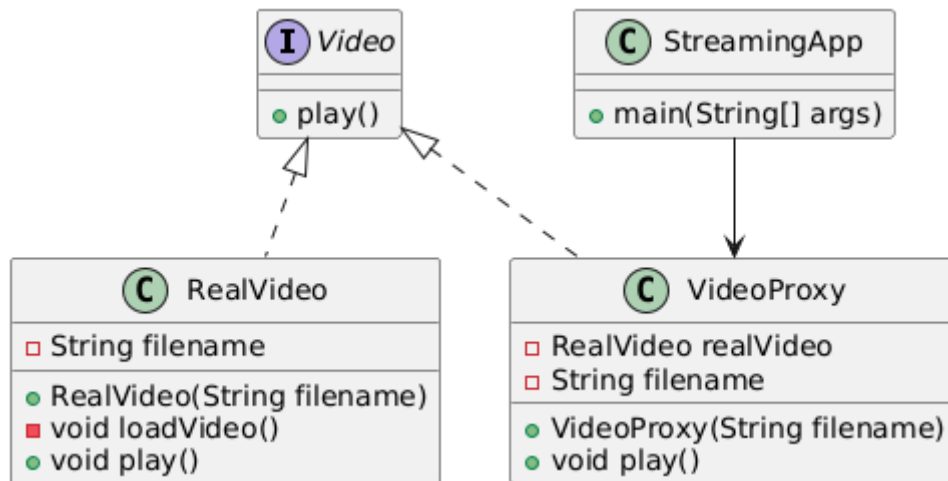
PS C:\Users\Blabl\Documents\site-simples\PAP> java DemoProxy
Usuário1 conectado ao banco de dados.
Usuário2 conectado ao banco de dados.
Usuário3 conectado ao banco de dados.
Usuário4 não pôde conectar: limite de conexões atingido.
Usuário2 desconectado do banco de dados.
Usuário4 conectado ao banco de dados.
PS C:\Users\Blabl\Documents\site-simples\PAP>

```

- 2) Em um sistema de *streaming* de vídeos, os vídeos podem ser carregados e reproduzidos sob demanda. No entanto, carregar um vídeo grande pode ser custoso. Para otimizar isso, o sistema deve utilizar um *Proxy* que carrega o vídeo apenas quando a reprodução é solicitada, evitando carregamentos desnecessários.

Considerando a situação descrita acima, utilize o padrão *Proxy* para implementar essa solução, garantindo que a classe `VideoProxy` gerencie o carregamento do vídeo real (`RealVideo`). O diagrama de classes a seguir (Figura 2) descreve a estrutura esperada para a solução. Com base nesse diagrama, implemente as classes necessárias.

Figura 2. Diagrama de Classes do sistema de *streaming* de vídeo .



Fonte: Elaborado pelo autor.

RESPOSTA:

```
public interface Video {
    void play();
}

public class RealVideo implements Video {
    private String filename;

    public RealVideo(String filename) {
        this.filename = filename;
        loadVideo(); // carrega o vídeo ao criar
    }

    public void loadVideo() {
        System.out.println("Carregando o vídeo: " + filename);
    }

    @Override
    public void play() {
        System.out.println("Reproduzindo o vídeo: " + filename);
    }
}

public class VideoProxy implements Video {
    private RealVideo realVideo;
    private String filename;

    public VideoProxy(String filename) {
        this.filename = filename;
    }
}
```

```

@Override
public void play() {
    if (realVideo == null) {
        realVideo = new RealVideo(filename); // só carrega se
necessário
    }
    realVideo.play();
}
}

public class StreamingApp {
    public static void main(String[] args) {
        Video video1 = new VideoProxy("filme_documentario.mp4");

        System.out.println("Usuário navegando pelo catálogo...");

        // Nenhum carregamento ainda
        System.out.println("Usuário decide assistir ao vídeo:");
        video1.play(); // agora carrega e reproduz
    }
}

```

O código é na pasta Proxy -> Questão 2.

Ele retorna isso:

```

PS C:\Users\Blabl\Documents\site-simples\PAP> java StreamingApp
Usuário navegando pelo catálogo...
Usuário decide assistir ao vídeo:
Carregando o vídeo: filme_documentario.mp4
Reproduzindo o vídeo: filme_documentario.mp4

```