

Lista de Exercícios  
Universidade Federal do Ceará  
Campus de Quixadá  
Projeto Detalhado de Software  
Paulyne Matthews Jucá

1. Modele a realização do caso de uso “criar conta de acesso ao sistema da empresa para novo funcionário” descrito abaixo utilizando um **diagrama de sequência**. O diagrama deve considerar todo o processo indo desde a interface até o serviço de armazenamento de dados.

Caso de uso:

- a. Administrador informa login e senha e seleciona de lista cadastrada previamente o departamento ao qual esse funcionário pertence e confirma a criação de uma nova conta de acesso ao sistema da empresa para o novo funcionário.
2. Os exemplos apresentados a seguir violam princípios, diretrizes ou padrões de projetos. Identifique qual o problema e o que fazer para arrumar.

a)

```
public interface Professor {  
    public void lecionarAula();  
    public void prepararAula();  
    public void orientarBolsista();  
    public void participarReuniaoColegiado();  
}
```

b)

```
public interface BibliotecaPessoal {  
    public void lerLivroDaLinhaDeComando();  
    public void imprimirLivroNaTela();  
    public void criarNovoLivro();  
    public void listarTodosOsLivrosNaTela();  
    public void salvarLivroNoBancoDeDados();  
    public void removerLivroDoBancoDeDados();  
}
```

c)

```
public class Gerente {  
    public void adicionarProduto(Loja loja, String nomeProduto, String tipo, float preço){  
        Produto produto = new Produto (nomeProduto, tipo, preço);  
        loja.getProdutos().add(produto);  
    }  
}
```

OBS: tipos de produtos são, livros, filmes e cds.

d)

```
public class Postagem{  
    String texto;
```

```

public Postagem (String texto){
    this.texto = texto;
}
public void conectarNoFacebook();
public void postarMensagemNoFacebook();
}

```

OBS: tornar flexível para evoluir para outras redes sociais

3. Considerando o código abaixo, como você faria modificações para desacoplar a interface gráfica da lógica e para atender ao padrão controlador?

```

package ufc.quixada.es.pds.grasp.controlador;

```

```

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JTextArea;
import javax.swing.JTextField;

```

```

@SuppressWarnings("serial")
public class PostarBlogJanela extends JFrame implements ActionListener {

```

```

    JButton publicar;
    JTextField titulo;
    JTextArea corpo;
    RepositorioPosts repositorioPosts;

```

```

    public PostarBlogJanela() {
        titulo = new JTextField("Digite o título...");
        this.add(titulo, BorderLayout.NORTH);

        corpo = new JTextArea("Corpo de texto do post...");
        this.add(corpo, BorderLayout.CENTER);

        publicar = new JButton("Publicar Post");
        publicar.addActionListener(this);
        this.add(publicar, BorderLayout.SOUTH);

        repositorioPosts = new RepositorioPosts();

```

```

    }
    public static void main(String[] args){

```

```

        PostarBlogJanela postarBlogJanela = new PostarBlogJanela();
        postarBlogJanela.mostrar();
    }
    private void mostrar() {
        this.setSize(400,400);
        this.setLocation(100, 100);
        this.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent publicarPost) {
        Post post = new Post();
        post.setTitulo(this.titulo.getText());
        post.setCorpoTexto(this.corpo.getText());
        repositorioPosts.adicionar(post);
    }
}

```

4. Considere o diagrama de comunicação abaixo. Aplique o princípio de Inversão de Dependência, e refatore o projeto dessas classes. Sua resposta deve ser um diagrama de classe.

