UNIVERSIDADE DO SUL DE SANTA CATARINA

RÔMULO VIEIRA DA SILVA

EXEMPLO DE APLICAÇÃO DOS PADRÕES DE PROJETO GOF PARA A ESPECIFICAÇÃO DE SIGAD

Florianópolis

2015

RÔMULO VIEIRA DA SILVA

EXEMPLO DE APLICAÇÃO DOS PADRÕES DE PROJETO GOF PARA A ESPECIFICAÇÃO DE SIGAD

Monografia apresentada ao Curso de Especialização em Engenharia de Projetos de Software da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Especialista em Engenharia de Projetos de Software.

Orientador: Prof. Dr. Jean Carlo Rossa Hauck

Florianópolis

2015

RÔMULO VIEIRA DA SILVA

EXEMPLO DE APLICAÇÃO DOS PADRÕES DE PROJETO GANG OF FOUR PARA A ESPECIFICAÇÃO DE SIGAD

Esta monografia foi julgada adequada à obtenção do título de Especialista em Engenharia de Projetos de Software e aprovada em sua forma final pelo Curso de Especialização em engenharia de Projetos de Software da Universidade do Sul de Santa Catarina.

Florianópolis, 25 de desembro de 2014.

_____

Professor e orientador Jean Carlo Rossa Hauck, Dr.
Universidade do Sul de Santa Catarina

_____

Prof. Aran Tcholakian Morales, Dr
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Agradeço a todos que de alguma forma contribuíram direta ou indiretamente ao sucesso deste trabalho.

RESUMO

Pessoas que trabalham com o desenvolvimento de software devem se preocupar em não apenas escrever código que funciona e resolva o problema, mas também deve se preocupar em escrever código seja entendido, mantido e reutilizável, principalmente por outras pessoas. Para contribuir com essa intenção, existem os padrões de projeto, que são, basicamente, o resultado da experiência de outros programadores compilada em catálogos. Este trabalho tem como principal objetivo exemplificar de maneira prática os 23 padrões de projeto de desenvolvimento de software presentes no catálogo publicado no livro *Design Patterns:Elements of Reusable Object-Oriented Software.* Para contextualizar e utilizar uma especificação real foi selecionada a especificação do Conselho Nacional de Arquivos (CONARQ) para sistemas informatizados de gestão arquivística de documentos (SIGAD). Devido a grande quantidade de requisitos presentes na especificação, foram utilizados na produção dos exemplos, apenas os requisitos dados como obrigatórios pela especificação.

Palavras-chave: Padrões de projeto. SIGAD. GoF.

ABSTRACT

People who work with the software development should be concerned with not only write code that works and resolve the issue, but should also worry about writing code to be understood, maintained and reusable, mainly by others. To contribute to this intention, there are design patterns, which are basically the result of the experience of other programmers compiled in catalogs. This work aims to exemplify in a practical way the 23 standards of software development project present in the catalog published in the book Design Patterns: Elements of Reusable Object-Oriented Software. To contextualize and use a real specification was selected the specification of the Conselho Nacional de Arquivos (CONARQ) to archival management system documents. Because the large amount of these requirements in the specification, were used in the production of examples only requirements required by the specification.

Keywords: Design Patterns.GoF.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A produção de software não é algo trivial, pois utiliza vários tipos diferentes de tecnologias, ferramentas, técnicas e padrões.

Os padrões de projeto são, em seu conceito, "descrições de objetos e classes comunicantes que precisam ser personalizadas para resolver um problema geral de projeto num contexto particular" (Gamma e outros, 2008, p.20).

A vantagem de conhecer os padrões de projeto, segundo Holzner (2006, p.29) é que ao resolver um problema de programação de software, o indivíduo não se preocupa apenas na solução do problema, mas também leva em consideração a capacidade de reuso, facilidade em estender e manter o a solução aplicada, que são os principais problemas atacados pelos padrões de projeto de software.

Este trabalho apresenta um dos principais catálogos de padrões de projeto utilizados. Segundo Holzner (2006, p.22), o primeiro catálogo de padrões de projeto foi publicado no livro chamado *Design Patterns:Elements of Reusable Object-Oriented Software*, no ano de 1995. Nesse catálogo, ao todo, são 23 padrões, subdividido em três grupos: de criação, de comportamento e de estrutura. Este catálogo é, segundo Metsker e Wake (2006, p.3) popularmente conhecido como GoF (Gang of Four) devido ao fato de terem 4 autores. Falando especificamente da linguagem Java[1], é possível citar como exemplo, as classes que encapsulam os tipos primitivos (ou seja, as que utilizam o padrão Adapter), como Integer, Long e outras. "[...] algumas dessas influencias podem ser vistas claramente nas primeiras APIs do Java, como as interfaces Iterator e Observer [...]. No AWT 1.0, o tratamento de eventos utilizava o padrão" (YODER, 2013, p. xv). Sobre outras tecnologias, como o GWT[2], que possui em sua arquitetura a utilização do padrão Proxy, para ter a sincronia entre instâncias diferentes da mesma classe, para que tenham os mesmos dados, tanto no cliente quando no servidor.

---

[1] Mais obre a linguagem Java disponível em: http://www.oracle.com/br/technologies/java/overview/index.html

[2] O Proxy está presente no RequestFactory do GWT. Mais detalhes em: http://www.gwtproject.org/doc/latest/DevGuideRequestFactory.html#proxies

Neste trabalho, para auxiliar na exemplificação de todos os padrões presentes no catálogo GoF, foi escolhida a especificação para SIGAD[3] adotada pelo Conselho Nacional de Arquivos (CONARQ). A partir dos requisitos presentes nesse projeto que serão elaborados os exemplos de uso de todos os 23 padrões de projeto do catálogo GoF.

1.1 OBJETIVOS

A seguir, serão apresentados os objetivos deste trabalho na forma de objetivo geral e objetivos específicos.

1.1.1 Objetivo geral

Aplicar todos os padrões de projeto do catálogo GoF na especificação de uma aplicação.

1.1.2 Objetivos específicos

a) Estudar os 23 padrões existentes no catálogo GoF;
b) Identificar possíveis exemplos de utilização em uma especificação existente;

[3] A sigla SIGAD quer dizer Sistemas Informatizados de Gestão Arquivística de Documentos.

c) Elaborar os diagramas de classe contendo o exemplo de utilização para cada padrão do catálogo;

1.2 JUSTIFICATIVA

O catálogo GoF foi publicado em 1995 e ainda assim é tão atual quanto era em seu lançamento, devido a necessidade ainda remanescente, de que o software produzido deverá ser entendido e mantido por outras pessoas. Apenas como exemplo, é possível citar a frase de Martin Fowler: "Qualquer um pode escrever um código que um computador possa entender, o truque é escrever código que humanos possam entender" [4] e há uma cultura que utiliza o termo artesão de software para quem produz software[5]. Tomando seu manifesto como exemplo, há um item interessante que diz: "Não apenas software em funcionamento, mas software de excelente qualidade". Este item é relevante, pois os padrões de projeto não se restringem em resolver o problema, mas vão além, também favorecem o aumento na qualidade do código produzido.

Para Metsker e Wake (2006, p.2), as pessoas aprenderam a programar orientado a objetos, mas se elas desejam serem programadores poderosos, eles deveriam aprender os padrões de projeto. Como primeiro passo, faz com que conhecer os padrões de projeto, seja quase um pré-requisito para quem deseja aprimorar suas habilidades em design de código e produzir códigos reutilizáveis e extensíveis.

Metsker e Wake (2006, p.3) dizem que tais padrões sugeridos servem como fundamento para iniciar o aprendizado de padrões oriundos de outras fontes.

---

[4] Disponível em: http://martinfowler.com/distributedComputing/refactoring.pdf

[5] Publicado originalmente no livro *Software Craftsmanship: The New Imperative* de Pete McBreen, disponível em http://manifesto.softwarecraftsmanship.org/#/pt-br.

1.3 ESTRUTURA DA MONOGRAFIA

O primeiro capítulo aborda a introdução ao tema, expondo os objetivos e a justificativa e, por fim, apresenta os procedimentos metodológicos.

O capítulo dois é composto pela revisão bibliográfica. Nele estão contidas todas as definições e conceitos a cerca do tema a ser estudado.

No capítulo três é apresentado o estudo de caso, bem como qual a especificação escolhida.

O quarto capítulo apresenta os diagramas contendo os exemplos de utilização dos padrões aplicados à especificação dos requisitos obrigatórios do SIGAD.

O quinto capítulo contém as conclusões sobre o estudo realizado juntamente com a proposta de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, são apresentados conceitos sobre UML, padrões de projeto, especificamente sobre o livro *Design Patterns: Elements of Reusable Object-Oriented Software* e o que é um sistema informatizado de gestão arquivística.

2.1 UNIFIED MODEL LANGUAGE – UML

Segundo Booch, Rumbaugh e Jacobson (2005, p.30) UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software que pode ser empregada para visualizar, especificar, construir e documentar os artefatos utilização em softwares. Fowler (2004, p.25) diz que a UML é uma família de notações gráficas que ajudam na concepção de projetos de software, principalmente aqueles que utilizam orientação a objetos.

A UML possui várias versões, a que será utilizada será a versão 2.0 que possui 14 diagramas divididos em dois grupos: diagramas estruturais e diagramas comportamentais.

Segundo Guedes (2011, p.101) o diagrama de classe é um dos mais importantes e partindo dele, é possível compor outros diagramas. Ele permite visualizar os atributos e métodos de cada classe, as classes que irão compor o sistema e como elas irão se relacionar.

Segundo Guedes (2011, p.46-47), uma classe é representada por um retângulo, que por sua vez, pode ser definido em até três partes: a primeira informa o nome da classe, a segunda informa os atributos que ela conte e a terceira mostra os métodos.

Segundo Hamilton e Miles (2006, p.100-105), a visibilidade dos tributos e métodos de uma classe na notação UML (exemplo na figura1), deve respeitar o seguinte padrão:

- “+” para públicos: permite o acesso direto de outra classe;
- “#” para protegidos: permite o acesso a métodos ou variáveis internas a classe ou subclasses;
- ”~” para pacotes: permite o acesso direto a outra classe, desde que estejam no mesmo pacote;
- “-” para privados: apenas métodos ou atributos da classe podem acessar.

Sobre os tipos de relacionamento entre as classes, Guedes (2011, p.106-117) afirma que eles podem ser: unária ou reflexiva, binária, associativa, agregação, composição, generalização, dependência realização.

- Unária ou reflexiva: se dá quando uma classe se relaciona com ela mesma;
- Binária: quando classes diferentes se relacionam, demonstrada com uma linha entre uma classe e outra. Sua representação visual é uma seta na extremidade para indicar o sentido;
- Associativa: conectam objetos de mais de duas classes. Sua representação visual é dada pela convergência das ligações para um losango;
- Agregação: quando uma classe precisa ser composta por outra classe. Sua representação visual é um losango ao lado da classe que conterá outra;
- Composição: representa um vínculo mais forte que a agregação, onde, a classe que contenha os objetos, é a única responsável por destruir os objetos compostos por ela. Sua representação visual é um losango preenchido ao lado da classe que conterá outra;
- Generalização/Especialização: representa a existência de herança no relacionamento entre as classes. Sua representação visual é um triangulo ao lado da classe que será herdada;
- Dependência: apresenta o grau de dependência entre as classes. Sua representação visual é uma linha tracejada entre duas classes contendo uma seta ao lado e apontando para classe de dependência;
- Realização: utiliza-se quando uma classe quer herdar o comportamento de uma classe, mas não sua estrutura. . Sua representação visual é uma

linha tracejada entre duas classes contendo uma seta vazia ao lado e apontando para a classe que tem funções realizadas por outra.

Sobre os motivos de utilizar a UML, Wazlawick (2011, p.4) afirma que apesar da quantidade de diagramas disponíveis na UML, nem todos os diagramas precisam ser utilizados no projeto de desenvolvimento de software e que deve ser usado apenas os que irão acrescentar alguma informação útil.

2.2 VISÃO GERAL DE PADRÕES DE PROJETO

A utilização dos padrões de projeto, do inglês *design patterns*, não é algo recente. De acordo com Sommerville (2007, p.279) e Metsker e Wake (2006, p.1) os padrões tiveram origem na obra de Christopher Alexander de 1977: *A Pattern Language: Towns, Buildings, Construction*, que mostrava padrões nas soluções de alguns problemas relacionados à construção civil, o que inspirou a criação de padrões voltados ao desenvolvimento de software.

Sommerville (2007, p.279) completa dizendo que o padrão não é uma especificação detalhada, mas sim a explicação de um problema com o acúmulo da experiência na solução deste problema, sendo de maneira abstrata, para que possa ser usado como solução para outros problemas com características semelhantes. Para Guerra (2013, p.15-16), um padrão de projeto descreve a solução mais utilizada para um problema em um contexto, por isso, geralmente estão acompanhados pela descrição e de usos conhecidos.

Segundo Holzner (2006, p.1) os padrões vão além, eles permitem o compartilhamento das soluções encontradas por programadores experientes com outras pessoas.

Segundo Sommerville (2007, p.279), os padrões de projetos normalmente fazem uso da herança e do polimorfismo para que possam ser usados de maneira genérica. Holzner (2006, p.8) completa dizendo que conhecer os padrões de projeto faz com que suas soluções sejam facilmente reusáveis, extensíveis e sustentáveis.

De acordo com Pressmann (2011, p.325), existem várias fontes de padrões disponíveis na internet. Alguns podem ser obtidos a partir de catálogos de padrões ou em repositórios. Por exemplo:

- Portland Pattern Repositoy - http://c2.com/ppr/index.html
- Hillside.net - http://hillside.net/patterns

2.3 PADRÕES DE PROJETO GOF

O catálogo conhecido como GoF(Gang of Four ou gangue dos quatro em tradução livre), cujos autores são Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, possui 23 padrões e são classificados em três categorias: padrões de criação, padrões estruturais e comportamentais. Para Larman (2005, p.296), o livro é considerado a "Bíblia" dos padrões de projeto de desenvolvimento de software e seus padrões são muito populares entre os desenvolvedores. Gamma e outros (2008, p. 25) afirmam que ao catálogo é organizado por finalidade, que pode ser de criação, estrutural ou comportamental.

2.3.1 Padrões criacionais

Os padrões de criação se preocupam com o processo de criação de objetos, os estruturais, com a composição e classes de objetos, e os padrões comportamentais, definem como as classes e objetos irão interagir e distribuir as responsabilidades.

De acordo com Gamma e outros, (2008, p.91), os padrões de criação abstraem a instanciação dos objetos, ajudando a tornar o sistema independente de como são criados, compostos e reapresentados.

2.3.1.1 Abstract Factory

Para Gamma e outros (2008, p.95), o objetivo deste padrão é fornecer uma interface de criação de objetos sem especificar suas classes concretas. Guerra (2013, p.136) afirma que este padrão é destinado à construção de uma família de objetos e quando usado, garante a consistência entre eles. Como complemento, Larman (2005, p.450), menciona que este padrão, em alguns momentos, uma classe que usa este padrão, pode ser utilizado em conjunto com o padrão Singleton.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.98):

- Isola as classes concretas;
- Torna fácil a troca de famílias de produtos;
- Promove a harmonia entre produtos;
- Torna difícil o suporte a novos tipos.

A seguir, a estrutura do padrão:

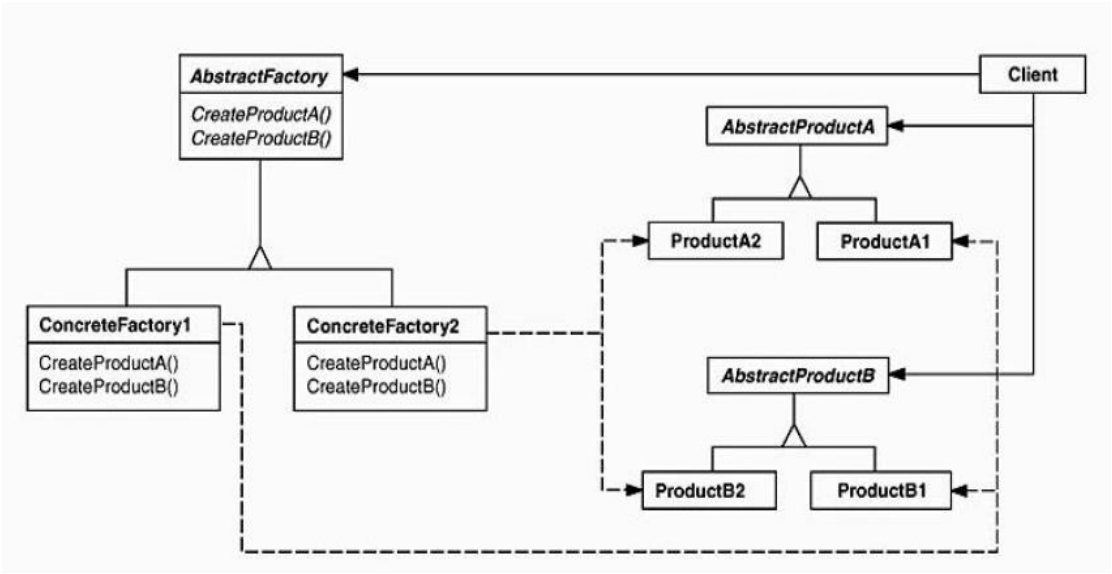


Figura 1: Estrutura do padrão Abstract Factory.
Fonte: Gamma e outros 2008, p.97.

2.3.1.2 Builder

Gamma e outros (2008, p.104) afirmam sobre este padrão: “separa a construção de um objeto complexo de sua representação, de modo que o mesmo processo de construção possa criar diferentes representações”. Para Holzner (2006, p.146), o padrão Builder proporciona mais flexibilidade ao processo de criação por dar à outra classe a responsabilidade do processo de construção do objeto.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.106):

- Permite variar a representação interna;
- Isola o código para a construção e representação;
- Oferece um controle mais fino sobre o processo de construção.

A seguir, a estrutura do padrão:

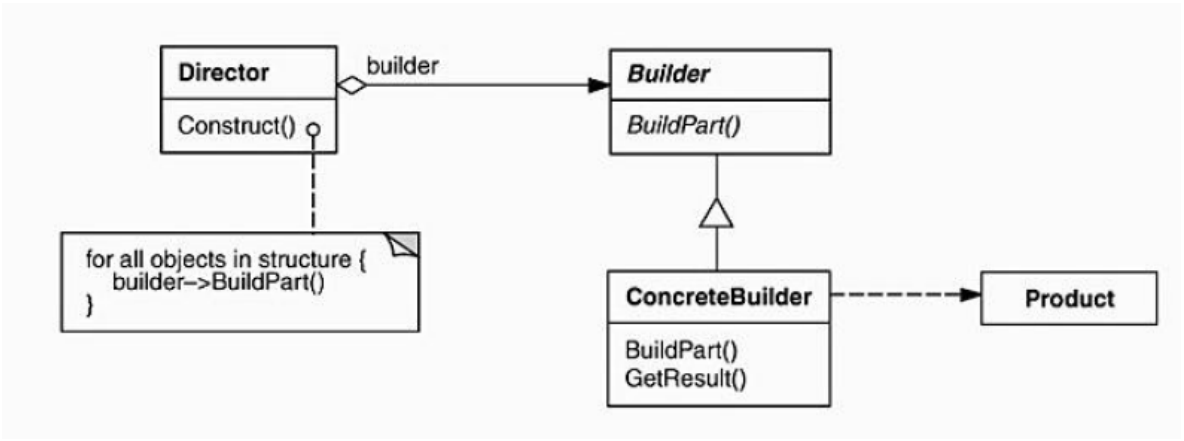


Figura 2: Estrutura do padrão Builder.
Fonte: Gamma e outros 2008, p.105.

2.3.1.3 Factory Method

Para Metsker (2002, p.132), este padrão requer que um método tenha a responsabilidade de instanciar objetos, fazendo com que o cliente não precisa conhecer detalhes da implementação. “Definir uma interface para criar um objeto, mas deixar as subclasses decidirem que classe instanciar. O Factory Method permite adiar a

instanciação para subclasses" (GAMMA,et al., 2008,p.112). Para Metsker e Wake (2002, p.180) este padrão podre evoluir e ser substituído pelo Abstract Factor, porém Holzner (2006, p.59) observa que quando os métodos das fábricas são constantemente sobrescritos, pode ser um sinal de uso do Factory Method.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.114-115):

- Fornece ganchos para subclasses;
- Conecta hierarquias de classe paralelas.

A seguir, a estrutura do padrão:

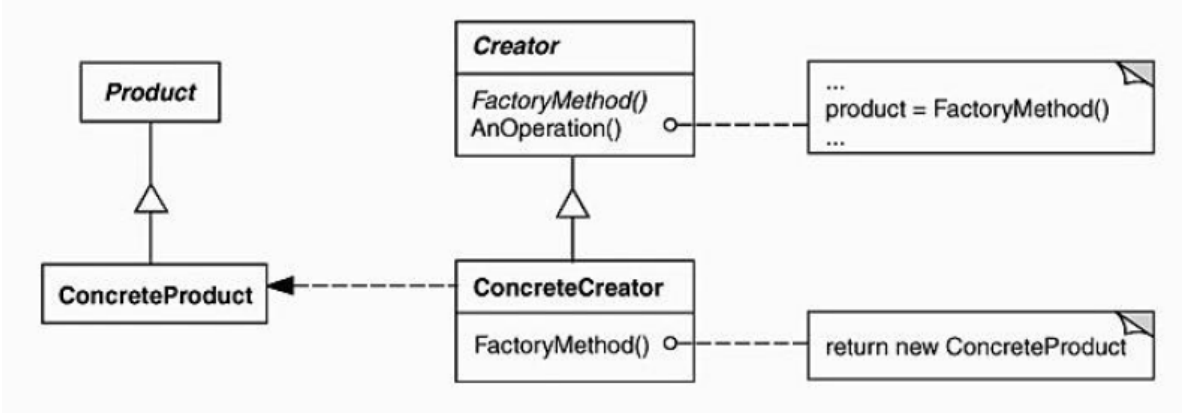


Figura 3: Estrutura do padrão Factory Method.
Fonte: Gamma e outros 2008, p.105.

2.3.1.4 Prototype

"Especificar os tipos de objetos a serem criados usando uma instância-protótipo e criar novos objetos pela copia desse protótipo." (GAMMA et al., 2008,p121). Segundo Holzner (2006, p.262), quando há muito código para criar um objeto, deve-se considerar fazer uma cópia desse objeto e customiza-lo. Para Metsker e Wake (2006, p.187), este padrão provê um novo objeto a partir da cópia de um objeto de exemplo ao invés de criar uma instancia como valores não inicializados.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.123-124):

- Acrescenta e remove instâncias como protótipo em tempo de execução;

- Especifica novos objetos pela variação de valores;
- Especifica novos objetos pela variação da estrutura;
- Reduz o número de subclasses;
- Configura dinamicamente uma aplicação com classes.

A seguir, a estrutura do padrão:

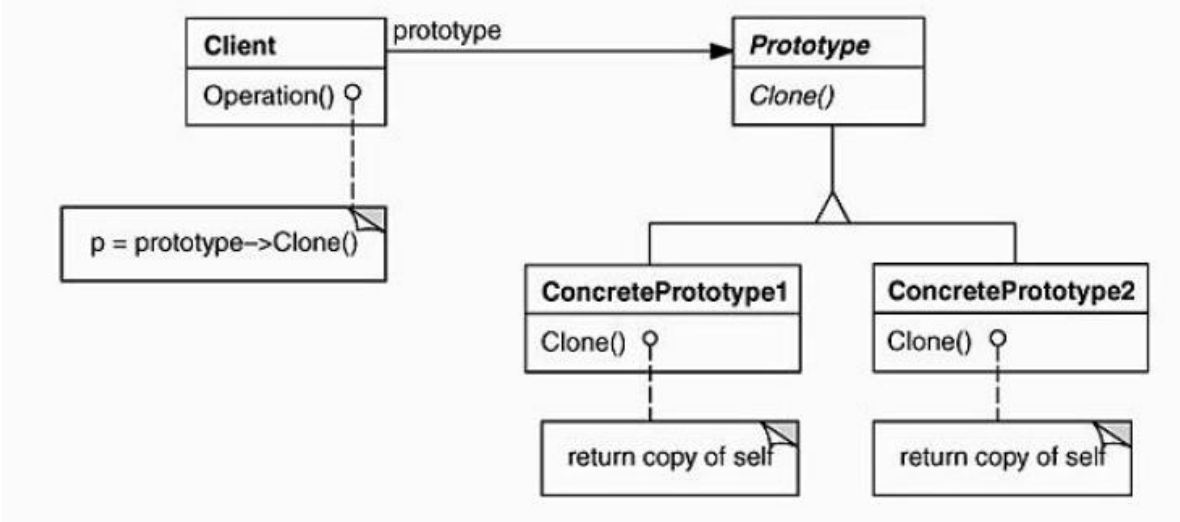


Figura 4: Estrutura do padrão Prototype.
Fonte: Gamma e outros 2008, p.123.

2.3.1.5 Singleton

"Garantir que uma classe tenha somente uma instancia e fornecer um ponto global de acesso para a mesma." (GAMMA et al., 2008,p.130). Metsker e Wake (2006, p.79) afirmam que este padrão concentra a responsabilidade em apenas um objeto provendo uma maneira de acesso global a ele. Holzner (2006, p.92-93) explica que este padrão impede que múltiplas instâncias de um objeto sejam criadas, e que deve ser usado quando se deseja limitar o uso de dados de um objeto e quando se quer que tais dados, estejam disponíveis a partir de um único ponto.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.131):

- Acesso controlado à instância única;

- Espaço de nomes reduzido;
- Permite um refinamento de operações e da representação;
- Permite um número variável de instâncias;
- Mais flexível do que operações de classe.

A seguir, a estrutura do padrão:

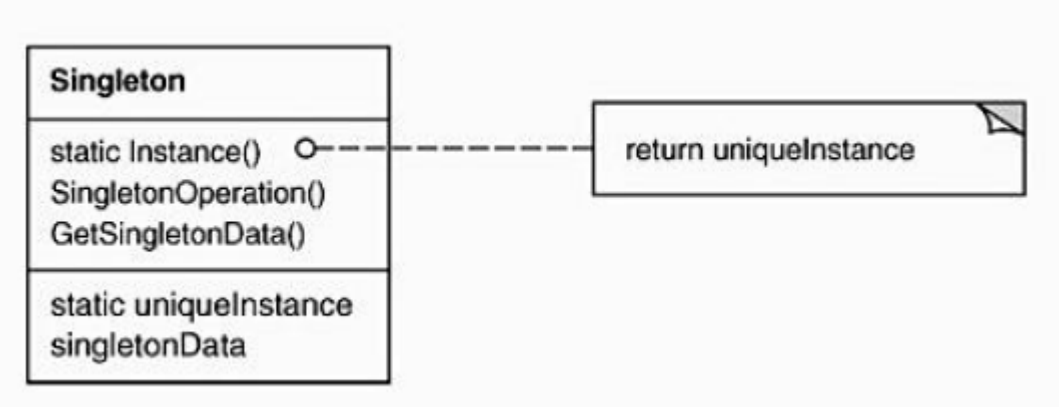


Figura 5: Estrutura do padrão Singleton.
Fonte: Gamma e outros 2008, p.130.

2.3.2 Padrões estruturais

De acordo com Gamma et al., (2008, p.139), “os padrões estruturais se preocupam com a forma como classes e objetos são compostos para formar estruturas maiores”

2.3.2.1 Adapter

Gamma e outros (2008, p.140) explicam que o padrão tem o objetivo de converter a interface de uma classe em outra que seja conhecida pelo cliente, permitindo que interfaces incompatíveis possam trabalhar juntas. Metsker e Wake (2006, p.17)

explicam que a intenção deste padrão é fornecer uma interface que o cliente espera usando serviços de outras interfaces.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.143-144):

- A quantidade de adaptação depende da similaridade entre as interfaces;
- Adaptadores conectáveis;
- A interface adaptada não pode ser usada no lugar da original.

A seguir, a estrutura do padrão:

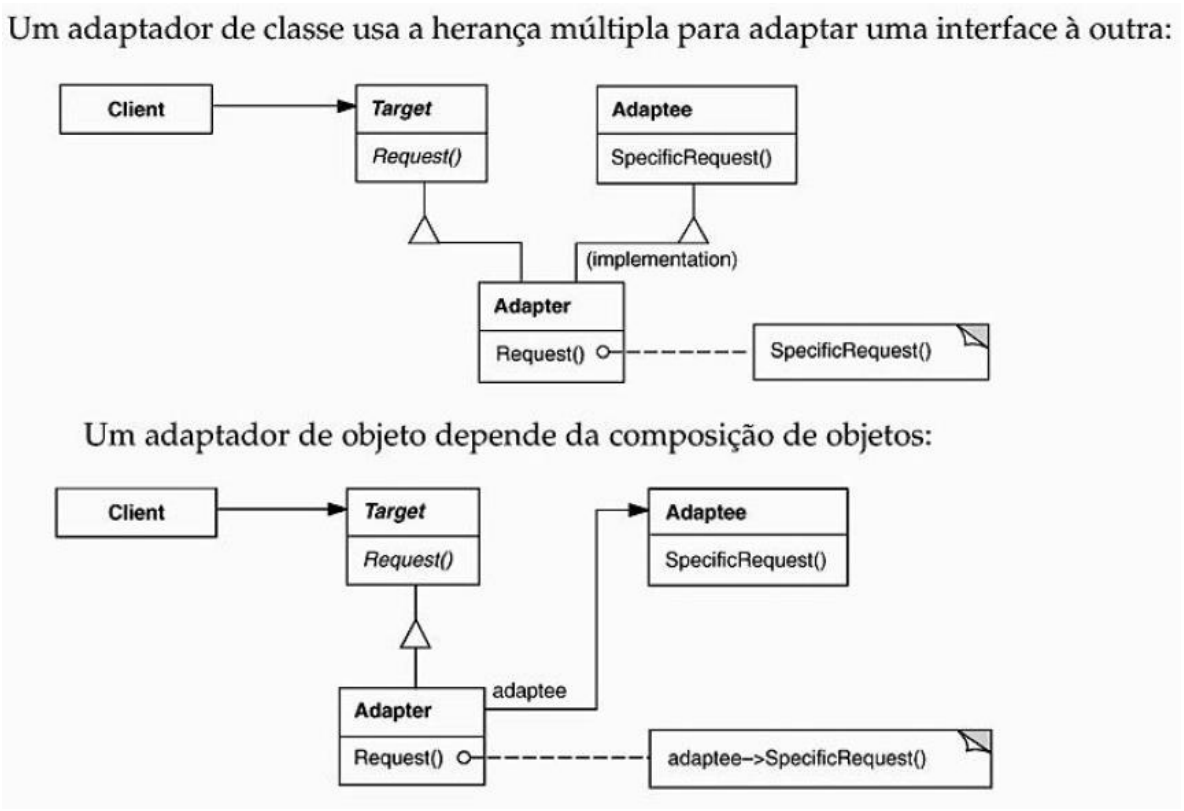


Figura 6: Estrutura do padrão Adapter.
Fonte: Gamma e outros 2008, p.142.

2.3.2.2 Bridge

Para Gamma e outros (2008, p.151) o padrão Bridge tem como objetivo: "desacoplar uma abstração da sua implementação, de modo que as duas possam variar independentemente.". Hozlner (2006, p.263) afirma que quando se tem duas abstrações

que podem mudar relacionada a uma implementação que também pode mudar, deve-se separa-las em duas. Guerra (2013, 50-51) explica que o padrão Bridge cria uma ponte entre duas hierarquias caracterizadas pela relação de composição permitindo que ambas variem de forma independente.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.154):

- Desacopla a interface da implementação;
- Extensibilidade melhorada;
- Oculta detalhes de implementação dos clientes.

A seguir, a estrutura do padrão:

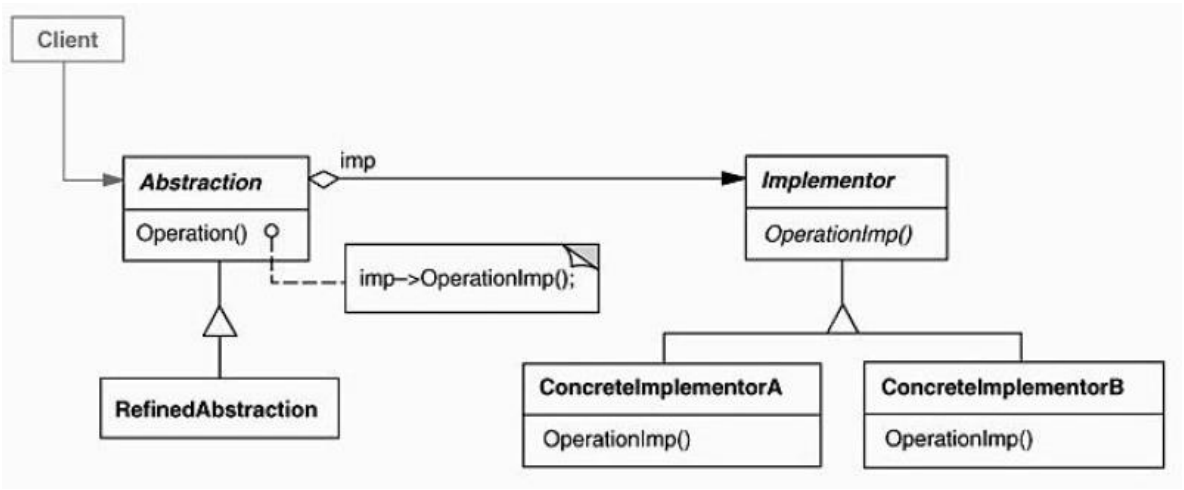


Figura 7: Estrutura do padrão Bridge.
Fonte: Gamma e outros 2008, p.153.

2.3.2.3 Composite

"Compor objetos em estruturas de arvore para representarem hierarquias partes-todo. Composite permite aos clientes tratarem de maneira uniforme objetos individuais e composição de objetos." (GAMA et al.,2008,p.160). Para Guerra (2013, p.81) o padrão permite que uma abstração possa ser utilizada para uma instância ou um conjunto delas.

> "[...] o padrão Composite cujo objetivo e prover uma solução para objetos que representam um conjunto de objetos, mas que compartilham a mesma abstração deles. Ele tem o potencial de encapsular uma logica complexa, dividindo-a em uma hierarquia de

> classes e simplificando a solução final. Todos que já utilizaram um Composite em um projeto real sabem que e um padrão realmente muito poderoso.” (GUERRA, 2013, p.81).

Consequências do uso do padrão, segundo Gamma e outros (2008, p.162-163):

- Suporta objetos simples ou compostos;
- Trata os objetos simples e compostos da mesma maneira;
- Permite um refinamento de operações e da representação;
- Suporta novas subclasses automaticamente, sem alteração;
- Pode tornar o projeto excessivamente genérico.

A seguir, a estrutura do padrão:

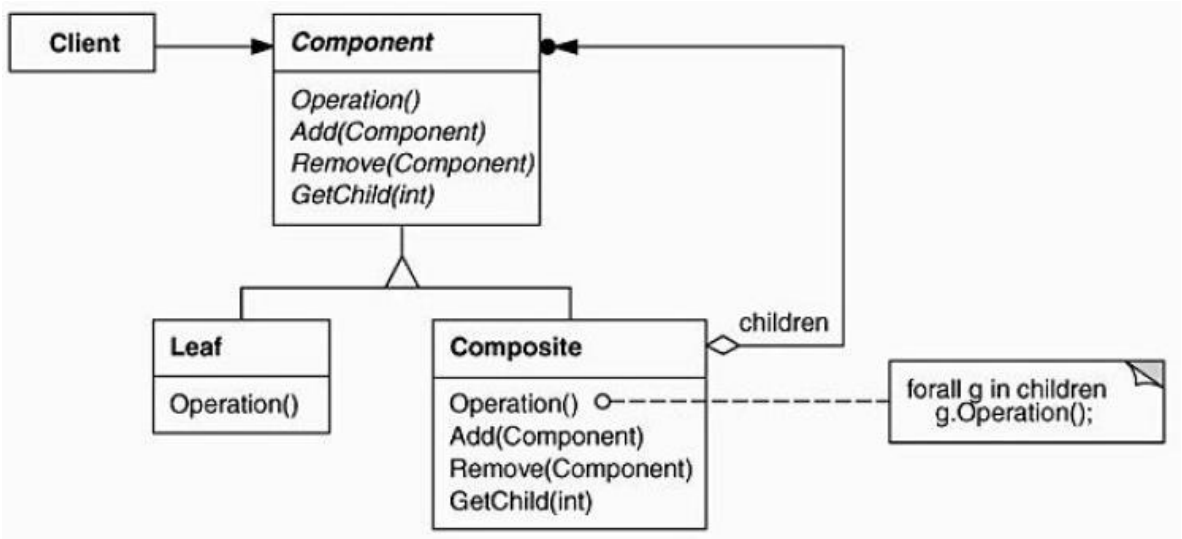


Figura 8: Estrutura do padrão Composite.
Fonte: Gamma e outros 2008, p.161.

2.3.2.4 Decorator

“Dinamicamente, agregar responsabilidades adicionais a um objeto. os decorators fornecem uma alternativa flexível ao uso de subclasses para extensão de funcionalidades.” (GAMA et al.,2008,p.170). Metsker e Wake (2006, p.287) complementam que o padrão permite adicionar comportamento ao objeto em tempo de execução.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.173):

- Oferece mais flexibilidade do que a herança estática;
- Permite adicionar comportamento apenas quando necessário;
- Um decorador e o seu componente não são idênticos;
- Grande quantidade de pequenos objetos.

A seguir, a estrutura do padrão:

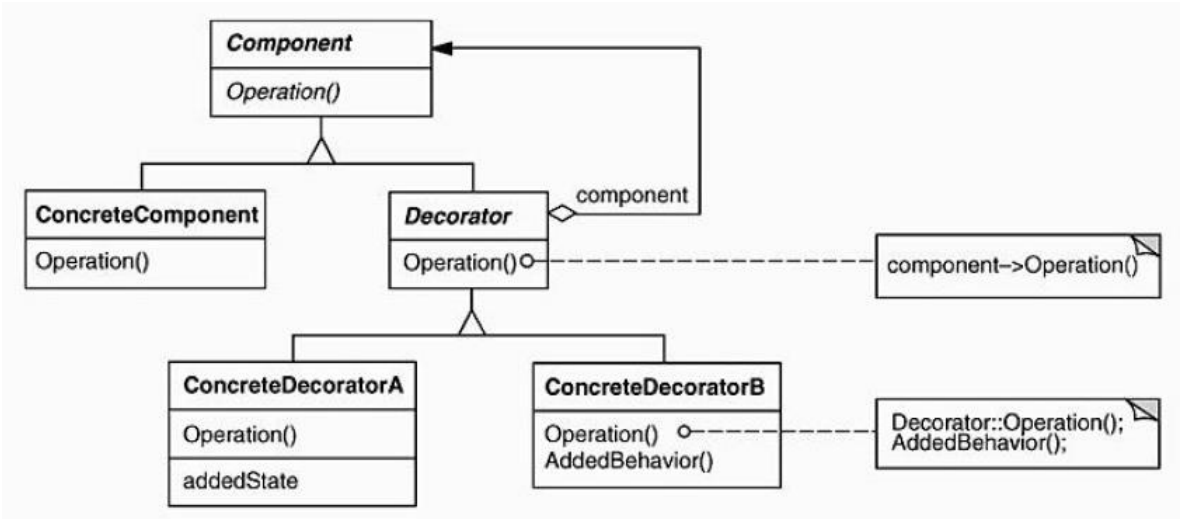


Figura 9: Estrutura do padrão Decorator.
Fonte: Gamma e outros 2008, p.172.

2.3.2.5 Façade (ou Facade)

"Fornecer uma interface unificada para um conjunto de interfaces em um subsistema. Façade define uma interface de nível mais alto que torna o subsistema mais fácil de ser usado." (GAMA et al.,2008,p. 179). Guerra (2013, p. 200) diz que sobre o padrão: "Essa classe encapsula a complexidade da interação entre os diversos componentes e desacopla o cliente das implementações".

Consequências do uso do padrão, segundo Gamma e outros (2008, p.181-182):

- Torna transparente ao cliente a utilização dos subsistemas;
- Promove acoplamento fraco entre o cliente e os subsistemas;
- Permite um refinamento de operações e da representação;
- Permite um número variável de instâncias;

- Mais flexível do que operações de classe.

A seguir, a estrutura do padrão:

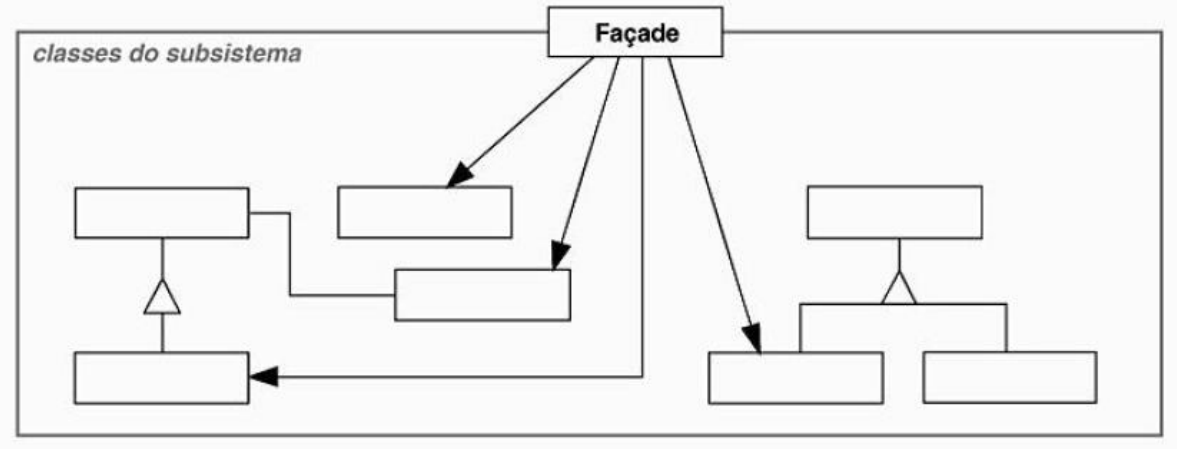


Figura 10: Estrutura do padrão Façade.
Fonte: Gamma e outros 2008, p.181.

2.3.2.6 Flyweight

"Usar compartilhamento para suportar eficientemente grandes quantidades de objetos de granularidade fina." (GAMA et al.,2008,p. 187). Para Metsker e Wake (2006, p. 152) os objetos que estão de acordo com este padrão devem ser imutáveis e garantir que eles sejam recuperados a partir de uma fábrica de objetos, a fim e de garantir o controle na criação de objetos.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.191-192):

- Economia no consumo de memória.

A seguir, a estrutura do padrão:

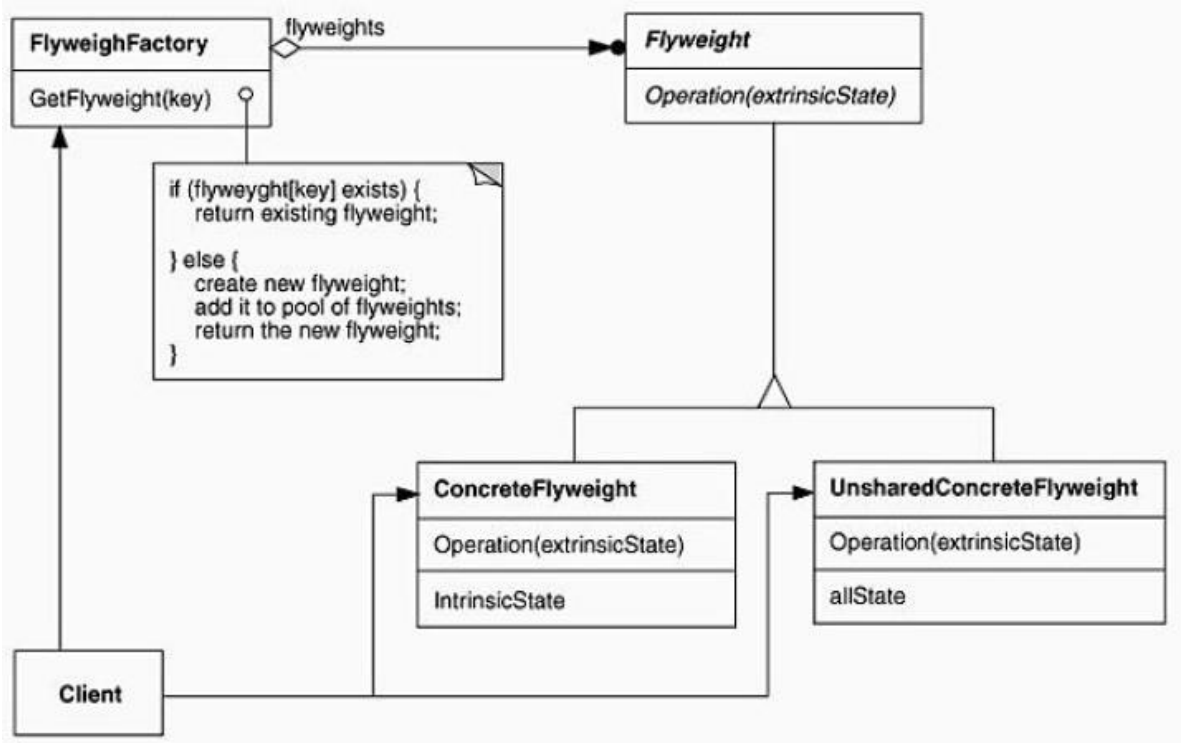


Figura 11: Estrutura do padrão Flyweight.
Fonte: Gamma e outros 2008, p.190.

2.3.2.7 Proxy

"Fornecer um substituto ou marcador da localização de outro objeto para controlar o acesso a esse objeto" (GAMA et al.,2008,p. 198). Holzner (2006, p.225) afirma que este padrão pode ser utilizado quando se deseja fazer com que um objeto remoto pareça local e quando se deseja proteger o acesso a um objeto.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.201):

- Pode tornar transparente: a invocação remota de objetos, realização de otimizações e aplicação de regras de segurança;

A seguir, a estrutura do padrão:

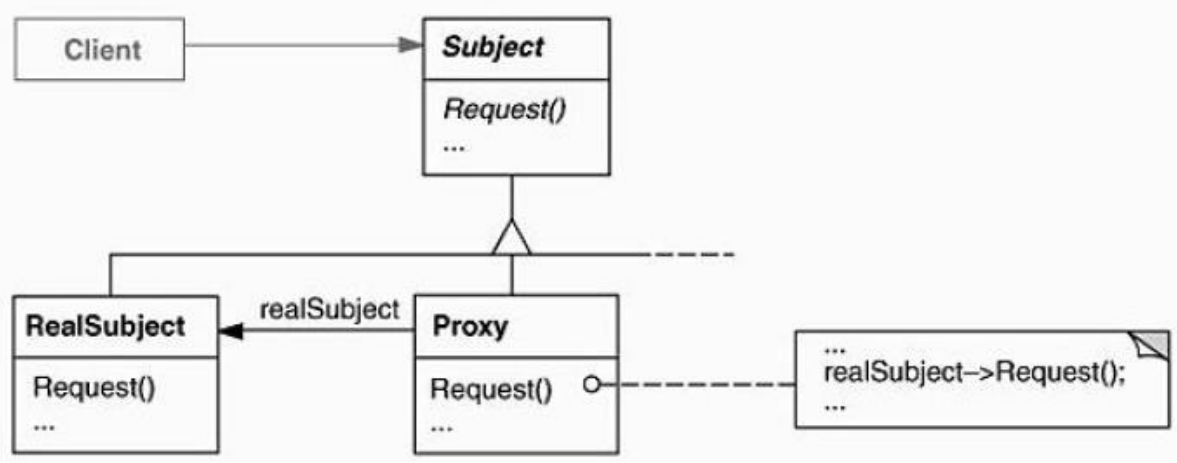


Figura 12: Estrutura do padrão Proxy.
Fonte: Gamma e outros 2008, p.200.

2.3.3 Padrões comportamentais

De acordo com Gamma e outros, (2008, p.211), os padrões eu definem comportamento se preocupam com algoritmos e atribuições de responsabilidades entre objetos e também os padrões de comunicação.

2.3.3.1 Chain of Responsibility

"Evitar acoplamento do remetente de uma solicitação ao seu receptor, ao dar a mais de um objeto a oportunidade de tratar a solicitação. Encadear os objetos receptores, passando a solicitação ao longo da cadeia até que um objeto a trate." (GAMA et al.,2008,p. 212). Para Freeman (2004, p. 616), este padrão deve ser usado quando se quer dar a chance para mais de um objeto tratar uma solicitação. De acordo com Metsker e Wake (2006, p. 403) este padrão distribui o tratamento entre a cadeia de execução e caso um objeto não possa trata-la, passará para o próximo objeto da cadeia.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.215):

- Reduz o acoplamento;

- Adiciona flexibilidade na definição de responsabilidades a objetos;
- Pode ser que ninguém trate a requisição por não ter ninguém com essa responsabilidade ou por equívocos na montagem da cadeia de tratamentos.

A seguir, a estrutura do padrão:

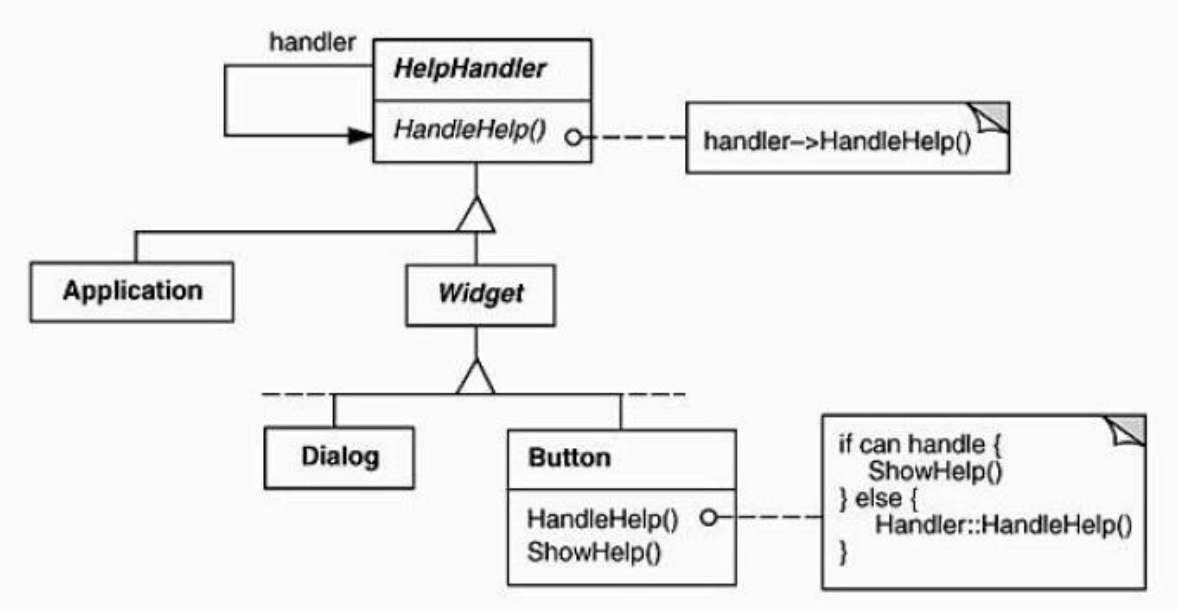


Figura 13: Estrutura do padrão Chain of Responsability.
Fonte: Gamma e outros 2008, p.214.

2.3.3.2 Command

"Encapsular uma solicitação como um objeto, desta forma permitindo parametrizar clientes com diferentes solicitações, enfileirar ou fazer o registro (log) de solicitações e suportar operações que podem ser desfeitas." (GAMA et al.,2008,p. 222).

Guerra (2013, p. 167) completa dizendo que o padrão parte do principio de que cada operação seja uma classe e que devem ter classes concretas que contenham as estruturas necessárias para a execução da operação.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.226):

- Reduz o acoplamento;
- Podem ser manipulado e estendido livremente;
- Um comando pode conter vários outros comandos;

- Facilidade ao adicionar novos comandos.

A seguir, a estrutura do padrão:

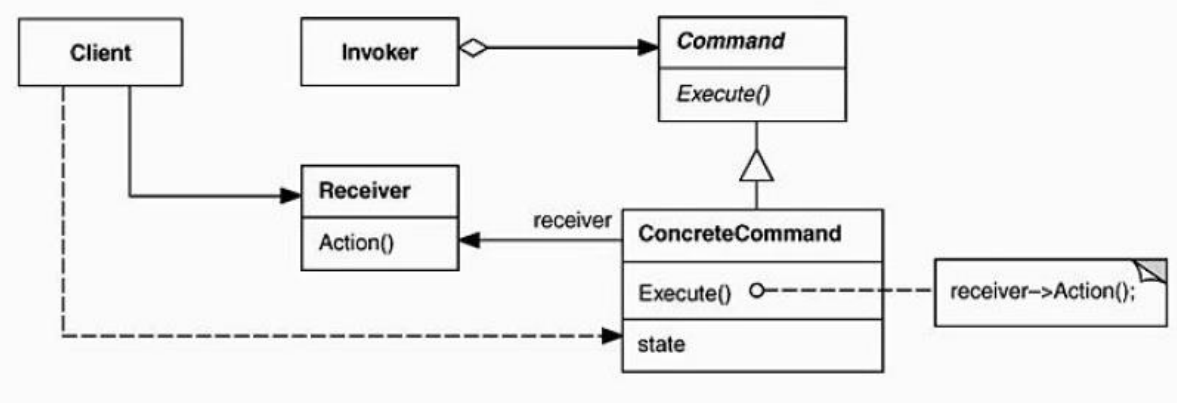


Figura 14: Estrutura do padrão Command.
Fonte: Gamma e outros 2008, p.225.

2.3.3.3 Interpreter

"Dada uma linguagem, definir uma representação para a sua gramatica juntamente com um interpretador que usa a representação para interpretar sentenças dessa linguagem." (GAMA et al.,2008,p. 231).

Metsker e Wake (2006, p. 261) explicam que permite compor objetos que podem ser utilizados de acordo com uma regra preestabelecida.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.235-236):

- Este padrão não tem responsabilidade de tratar a análise sintática;
- Gramáticas complexas se tornam difíceis de manter;
- Não dificulta a implementação da gramática;
- Facilita a adição de uma nova forma de interpretar expressões.

A seguir, a estrutura do padrão:

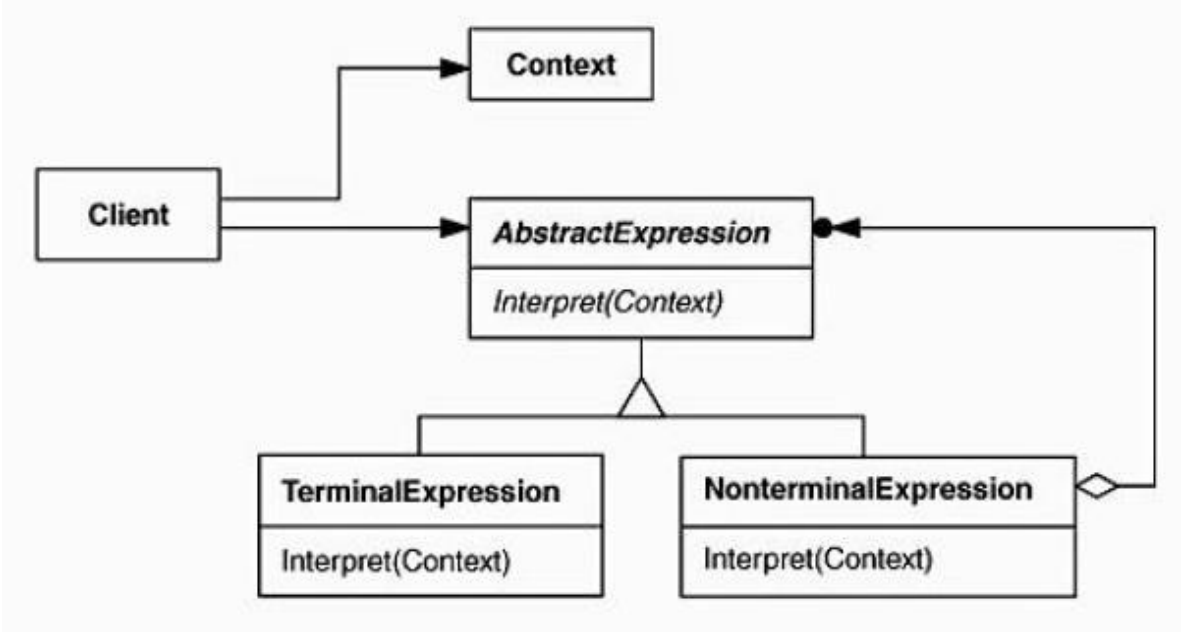


Figura 15: Estrutura do padrão Interpreter.
Fonte: Gamma e outros 2008, p.234.

2.3.3.4 Iterator

"Fornecer um meio de acessar, sequencialmente, os elementos de um objeto agregado sem expor a sua representação subjacente." (GAMA et al.,2008,p. 244).

Metsker e Wake (2006, p.305) afirmam que a intenção deste padrão é prover uma maneira de acessar sequencialmente os elementos de uma coleção.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.246-247):

- Suporta variações no percurso ao percorrer os elementos;
- Não expõe a implementação interna dos itens percorridos;
- Pode haver vários percursos ocorrendo ao mesmo tempo.

A seguir, a estrutura do padrão:

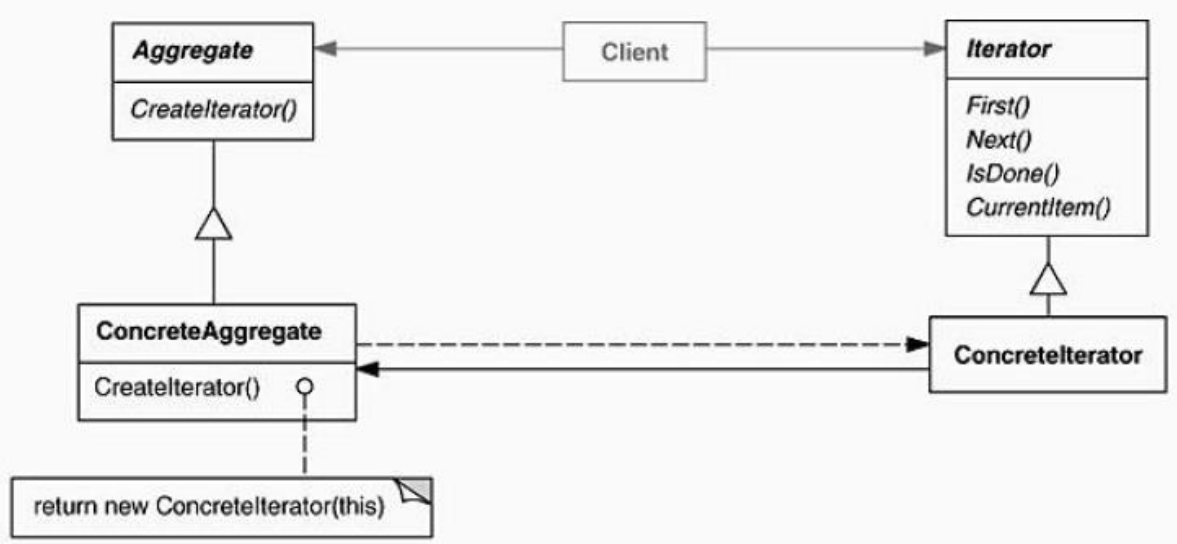


Figura 16: Estrutura do padrão Iterator.
Fonte: Gamma e outros 2008, p.246.

2.3.3.5 Mediator

"Definir um objeto que encapsula a forma como um conjunto de objetos interage. O Mediator promove o acoplamento fraco ao evitar que os objetos se refiram uns aos outros explicitamente e permite variar suas interações independentemente." (GAMA et al.,2008,p. 257).

Sobre este padrão, Guerra (2013, p. 211-212) diz que ele centraliza o gerenciamento em uma classe que é responsável por simplificar a interação entre elas, facilitando a adição de novos objetos relacionados ao contexto da iteração.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.261-262):

- Para mudar o comportamento, é preciso apenas uma subclasse de Mediator;
- Reduz o acoplamento;
- Simplifica e abstrai a comunicação entre os objetos envolvidos;
- Centraliza o controle.

A seguir, a estrutura do padrão:

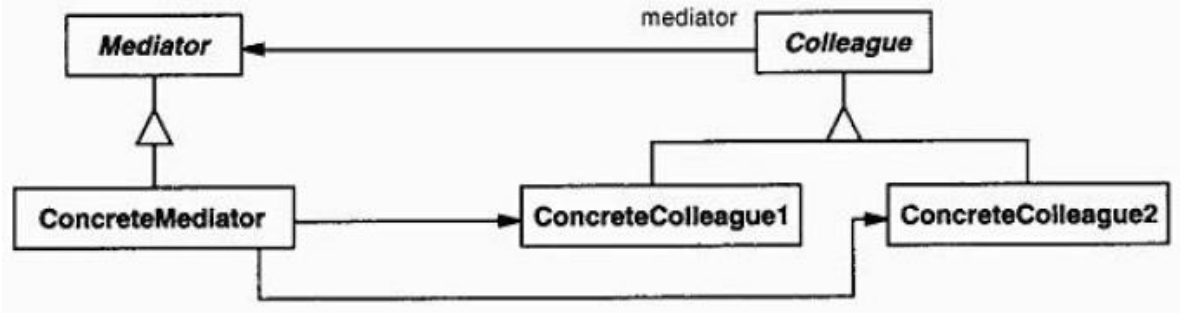


Figura 17: Estrutura do padrão Mediator.
Fonte: Gamma e outros 2008, p.260.

2.3.3.6 Memento

"Sem violar o encapsulamento, capturar e externalizar um estado interno de um objeto, de maneira que o objeto possa ser restaurado para esse estado mais tarde." (GAMA et al.,2008,p. 266).

Freeman (2004, p. 625) afirma que o padrão memento tem dois objetos: guardar o estado de um objeto e manter o encapsulamento dos atributos importantes.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.269-270):

- Favorece o encapsulamento;
- Mantem o estado dos objetos;
- Pode consumir muita memória para armazenar os estados dos objetos.

A seguir, a estrutura do padrão:

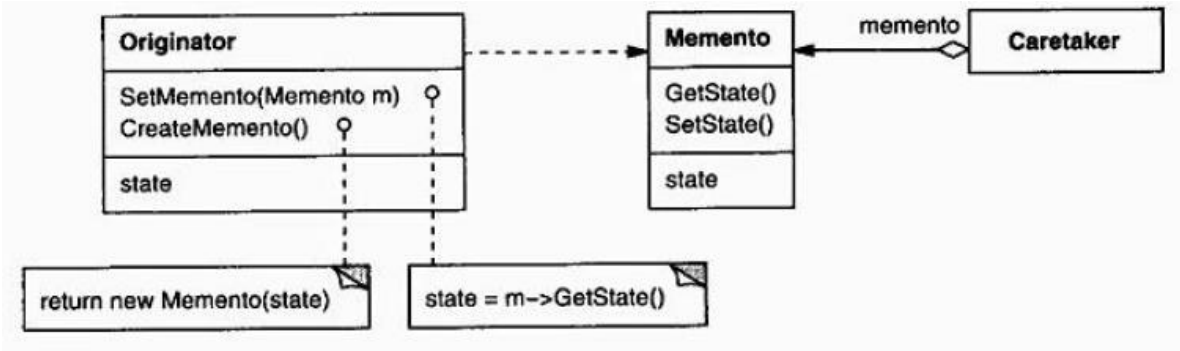


Figura 18: Estrutura do padrão Memento.
Fonte: Gamma e outros 2008, p.268.

2.3.3.7 Observer

"Definir uma dependência de um-para-muitos entre objetos, de maneira que quanto um objeto muda de estados todos os seus dependentes são notificados e atualizados automaticamente." (GAMA et al.,2008,p. 274).

De acordo com Guerra (2013, p. 66), afirma que este padrão deve ser usado quando os eventos de um objeto precisam ser observados por outros objetos.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.276-277):

- Pode haver alto custo computacional;
- É preciso manter a consistência, principalmente na remoção para objetos que estão sendo observados;
- É preciso cuidado ao pensar na estratégia de notificação: se enviar todo o objeto alterado ou apenas o que for necessário.

A seguir, a estrutura do padrão:

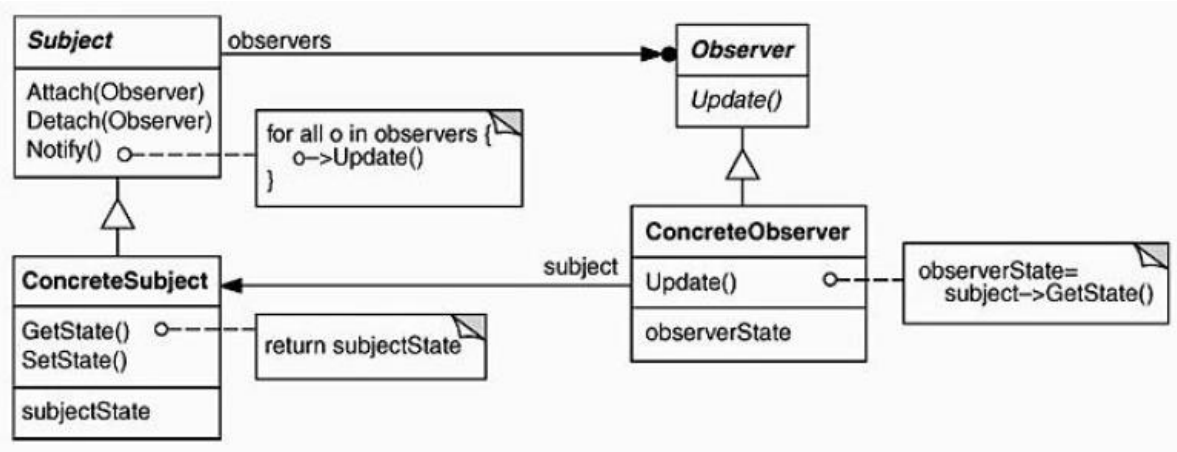


Figura 19: Estrutura do padrão Observer.
Fonte: Gamma e outros 2008, p.275.

2.3.3.8 State

"Permite um objeto alterar seu comportamento quando o seu estado interno muda. O objeto parecerá ter mudado sua classe." (GAMA et al.,2008,p. 284).

Para Holzer (2006, p. 209) este padrão é útil quando o código fica cada vez mais complexo à medida que evolui e que seja possível separar em compartimentos(estados) a respectiva lógica.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.286-287):

- Aumenta o numero de classes e reduz código condicional por distribuir o comportamento;
- A própria classe, define qual o estado, não necessitando conhecer valores de atributos, mas apenas, o seu comportamento;
- Objetos podem ser compartilhados.

A seguir, a estrutura do padrão:

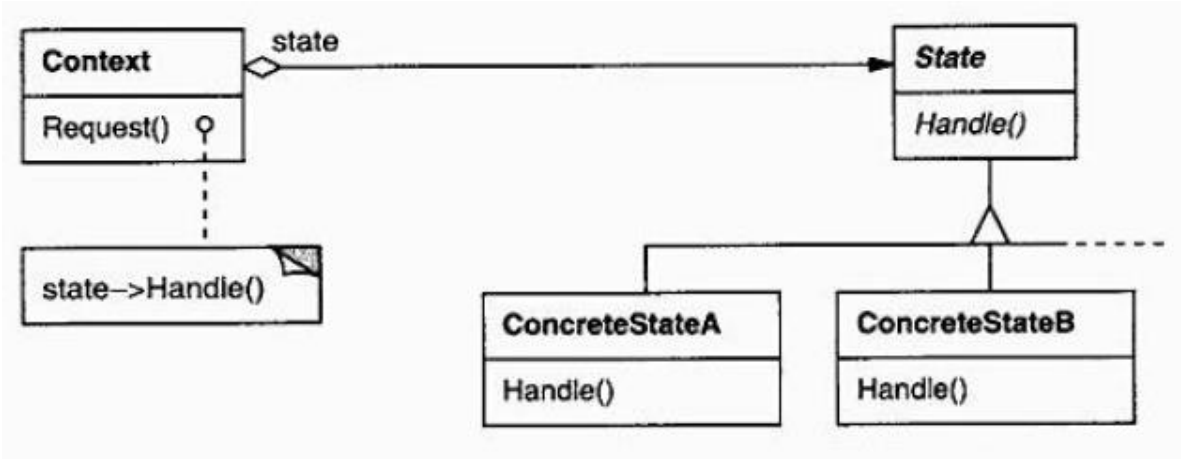


Figura 20: Estrutura do padrão State.
Fonte: Gamma e outros 2008, p.285.

2.3.3.9 Strategy

"Definir uma família de algoritmos, encapsular cada uma delas e torna-las intercambiáveis. Strategy permite que o algoritmo varie independentemente dos clientes que o utilizam." (GAMA et al.,2008,p. 292).

Para Metsker e Wake (2006, p. 241), As classes que programam as várias abordagens, programam a mesma operação e assim, são permutáveis entre si, apresentando as estratégias diferentes, mas a mesma interface para quem as utiliza.

Sobre este padrão, Guerra (2013, p. 14) afirma que ele deve ser usado quando uma classe possuir algoritmos que possam ser trocados.

Consequências do uso do padrão, segundo Gamma e outros (2008, p.295-296):

- Favorece a flexibilidade para a customização e extensão dos algoritmos;
- É uma alternativa ao uso de subclasses;
- Reduz a quantidade de código condicional.

A seguir, a estrutura do padrão:

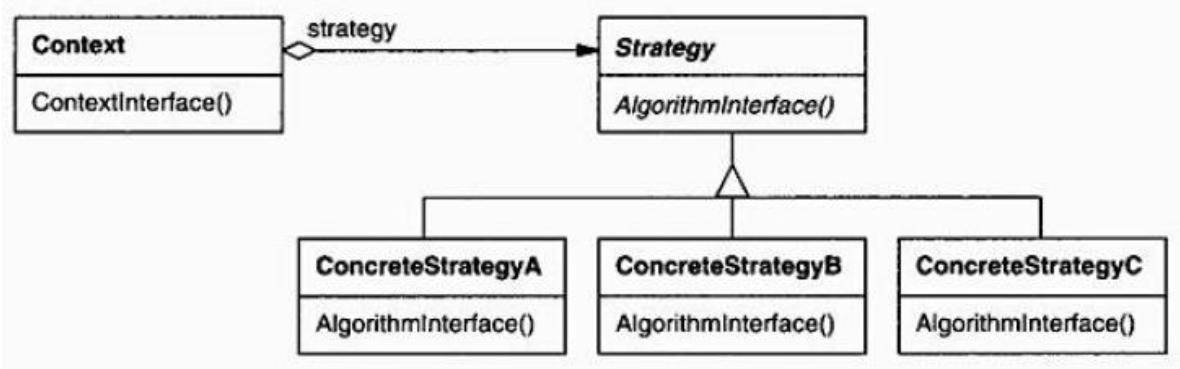


Figura 21: Estrutura do padrão Strategy.
Fonte: Gamma e outros 2008, p.294.

2.3.3.10 Template Method

"Definir o esqueleto de um algoritmo de uma operação, postergando alguns passos para as subclasses. Template Method permite que subclasses redefinam certos passos de um algoritmo sem mudar a estrutura do mesmo." (GAMA et al.,2008,p. 301).

Guerra (2013) explica que:

> "Este padrão é aplicável quando se deseja definir um algoritmo geral, que estabelece uma série de passos para cumprir um requisito da aplicação. Porém, seus passos podem variar e é desejável que a estrutura da implementação forneça uma forma para que eles sejam facilmente substituídos." (Guerra, 2013, p. 29).

Consequências do uso do padrão, segundo Gamma e outros (2008, p.303-304):

- Conduzem a uma estrutura de inversão de controle;

- O programador deve lembrar-se de chamar sobrescrever o método que deverá ser customizado.

A seguir, a estrutura do padrão:

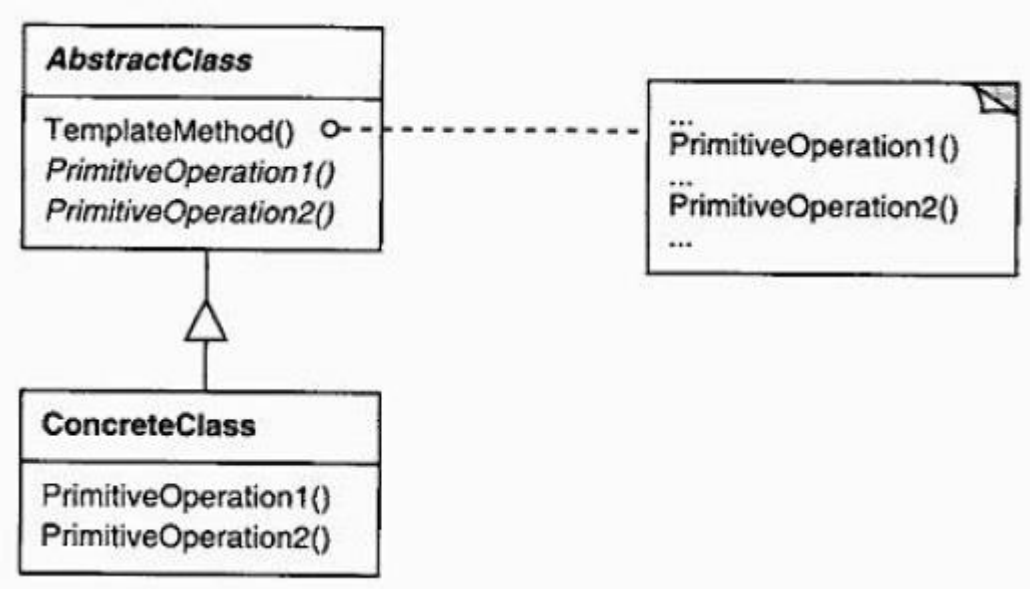


Figura 22: Estrutura do padrão Template Method.
Fonte: Gamma e outros 2008, p.302.

2.3.3.11 Visitor

"Representar uma operação a ser executada nos elementos de uma estrutura de objetos. Visitor permite definir uma nova operação sem mudar as classes dos elementos sobre os quais opera." (GAMA et al.,2008,p. 305).

Para Holzner (2008, p. 267) "você pode adicionar uma nova operação para uma estrutura de objetos, como uma estrutura composta, sem alterar os objetos na estrutura".

Consequências do uso do padrão, segundo Gamma e outros (2008, p. 309-310):

- Facilita a adição de novas operações;
- Permite que cada visitante tenha seu próprio comportamento;
- Dificuldade a criação de objetos que serão visitados;
- Centraliza o controle.

A seguir, a estrutura do padrão:

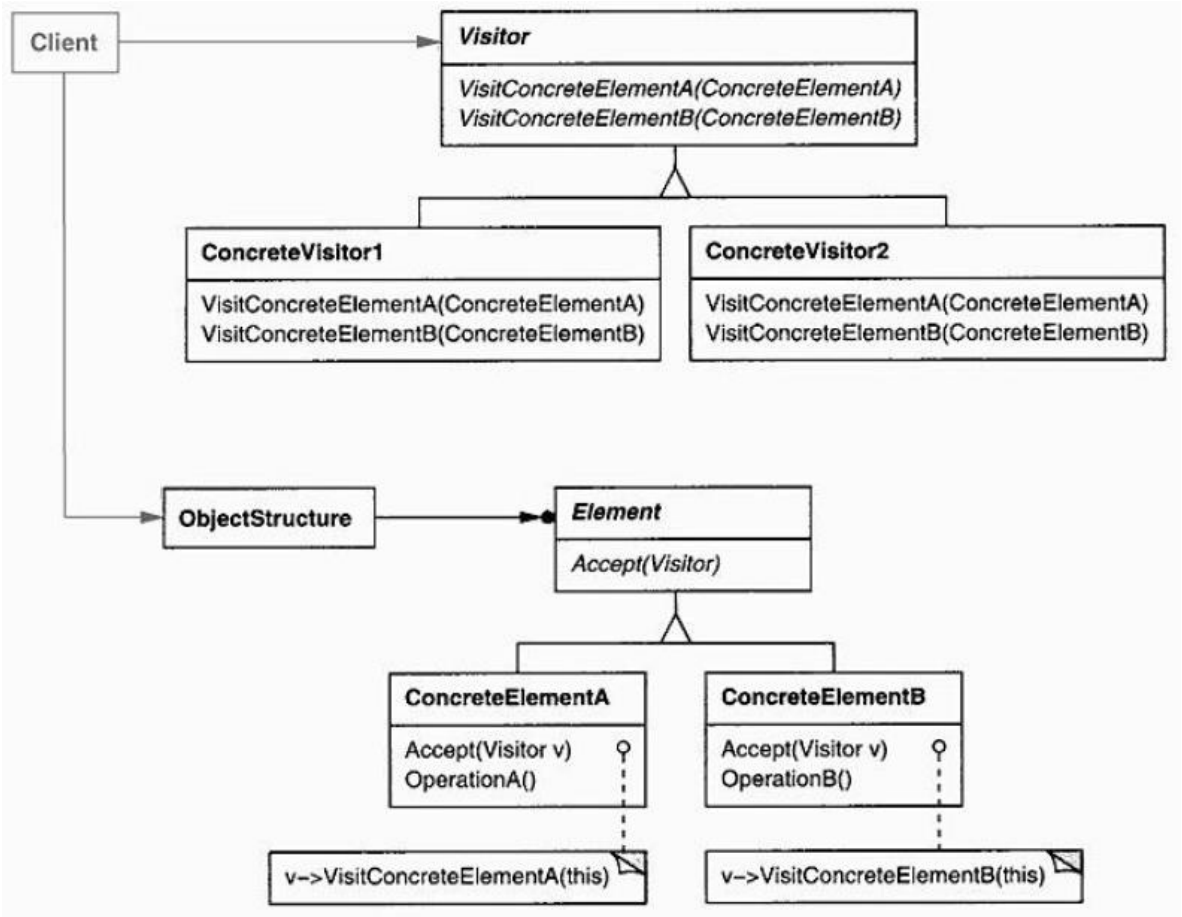


Figura 23: Estrutura do padrão Visitor.
Fonte: Gamma e outros 2008, p.308.

3 MÉTODO

Este capítulo tem como intenção apresentar a caracterização do tipo de pesquisa, como serão atingidos os objetivos e, por fim, a delimitação do trabalho.

3.1 CARACTERIZAÇÃO DO TIPO DE PESQUISA

Este trabalho tem sua natureza dada como aplicada, pois visa a solução de um problema específico, indo de encontro com a definição de Silva e Menezes (2005, p. 20) que diz que a natureza da pesquisa é aplicada quando: "objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos".

Do ponto de vista de seus objetivos, esta pesquisa classifica-se como exploratória, já que "visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou a construir hipóteses. Envolve levantamento bibliográfico; [...] Assume, em geral, as formas de Pesquisas Bibliográficas e Estudos de Caso." (SILVA; MENEZES, 2005, p.21).

Do ponto de vista dos procedimentos técnicos, pode-se classificar esta pesquisa como pesquisa bibliográfica e estudo de caso, que, segundo Gil (1991, apud SILVA; MENEZES, 2005, p. 21) é estudo de caso quando a pesquisa envolve o estudo de um ou poucos objetos permitindo o seu amplo e detalhado conhecimento e é uma pesquisa bibliográfica, porque é elaborada a partir de material já publicado, livros, artigos de periódicos e com material disponibilizado na Internet.

Tratando especificamente de software, Runeson e Höst (2008, p.137-138) apresentam os seguintes passos para estudo de caso aplicado ao software:

1. Definição do estudo de caso: definição e planejamento do estudo de caso;
2. Preparação dos dados: definição da estratégia de coleta de dados;
3. Coletando dados: execução das estratégias de coleta de dados;
4. Analise: análise dos dados coletado no estudo de caso
5. Relatório: conclusão da análise.

Runeson e Höst (2008,138-139) explicam que na etapa de definição do estudo de caso, é preciso planejar todo o ciclo do estudo de caso para que possa ser documentado na etapa seguinte.

Segundo Runeson e Höst (2008, p.140-141), a segunda etapa tem o objetivo de definir os passos a serem seguidos para a preparação dos dados, a fim de que nada seja perdido ou esquecido.

Sobre a coleta de dados, Runeson e Höst (2008, p. 144) afirmam que não se deve restringir a apenas uma fonte de dados, dessa forma, a conclusão da análise ganha mais força. Para a quarta e quinta etapas, Runeson e Höst (2008,150-155) afirmam que a análise dos dados pode ser de forma qualitativa e quantitativa, e que o relatório, contendo a conclusão, deva mostrar as descobertas do estudo e a fonte dos dados para avaliar a qualidade do estudo.

3.2 ETAPAS METODOLÓGICAS

Abaixo, são listadas as etapas que serão executadas para o cumprimento da solução proposta:

1. Listar os requisitos a serem utilizados da especificação do SIGAD;
2. Montar os casos de uso com os requisitos listados;
3. Cruzar os casos de usos com os 23 padrões de projeto, para encontrar os que se relacionavam;
4. Elaborar os diagramas de exemplo de uso dos padrões;
5. Elaborar a conclusão do estudo;

3.3 DELIMITAÇÃO

Os exemplos de aplicação dos 23 padrões de projeto serão elaborados tomando como base apenas os requisitos dados como obrigatórios pela especificação do

SIGAD adotada pelo CONARQ. Não serão discutidos a escrita dos requisitos ou qualquer outro texto oriundo da especificação.

4 DESENVOLVIMENTO

Neste capítulo serão apresentados os diagramas de classe que exemplificam a utilização de padrões de projeto na especificação do SIGAD.

4.1 MOTIVOS PARA ESCOLHA DA ESPECIFICAÇÃO DO SIGAD

Segundo Brasil (2014, p. 25), existe a Lei Federal nº 8.159, de 8 de janeiro de 1991, art. 1º, que estabelece que é dever do poder público a gestão documental e a proteção especial a documentos de arquivo e com a promulgação da lei federal nº 12.527, de 18 de novembro de 2011, que assegura amplo acesso às informações públicas de maneira mais detalhada, assegurando a todos o direito de receber dos órgãos públicos as informações de seu interesse particular ou de interesse coletivo que estão contidas em documentos de arquivos.

4.2 SIGAD

O governo, a administração publica e privada, a pesquisa cientifica e tecnológica e a expressão cultural dependem cada vez mais de documentos digitais, não disponíveis em outra forma, para o exercício de suas atividades.

Segundo Brasil (2011, p. 10), o Sistema Informatizado de Gestão Arquivística de Documentos (SIGAD) teve sua origem nos sistemas informatizados de gerenciamento de documento. Esses sistemas não suportavam os conceitos arquivísticos, fazendo-se necessária, então, uma adaptação para contemplar a gestão arquivística, surgindo o novo conceito, o SIGAD.

O SIGAD "é um conjunto de procedimentos e operações técnicas que visam o controle do ciclo de vida dos documentos, desde a produção até a destinação final,

seguindo os princípios da gestão arquivística de documentos e apoiado em um sistema informatizado.” (BRASIL, 2011).

Segundo o Brasil (2011), o e-ARQ Brasil estabelece requisitos mínimos para um SIGAD, independentemente da plataforma tecnológica em que for desenvolvido e/ou implantado. Nele, há a especificação das atividades e operações para a produção tramitação, utilização, arquivamento e destinação final de documentos aplicados à gestão arquivística. “O e-ARQ Brasil deve ser utilizado para desenvolver um sistema informatizado ou para avaliar um já existente, cuja atividade principal seja a gestão arquivística de documentos.” (BRASIL, 2011).

> “É uma especificação de requisitos a serem cumpridos pela organização produtora/recebedora de documentos, pelo sistema de gestão arquivística e pelos próprios documentos, a fim de garantir sua confiabilidade e autenticidade, assim como sua acessibilidade.” (BRASIL, 2011).

Esta especificação é dividida em duas partes. A primeira parte trata das questões arquivísticas e a segunda parte se refere aos requisitos do SIGAD.

Na segunda parte, é importante notar que os requisitos são classificados em obrigatório, altamente desejável e facultativo. “Os requisitos dirigem-se a todos que fazem uso de sistemas informatizados como parte do seu trabalho rotineiro de produzir, receber, armazenar e acessar documentos arquivísticos. [...]” (BRASIL, 2011).

4.3 APRESENTAÇÃO DO ESTUDO DE CASO

Como especificação de estudo, foi selecionada a especificação do CONARQ para o SIGAD. De maneira geral, o SIGAD é um sistema de gestão de documentos arquivísticos, onde é importante obter, guardar e recuperar documentos. Essa especificação foi escolhida, pois possui centenas de requisitos, mas que para o estudo foram selecionados apenas os caracterizados como obrigatórios pela especificação, e dada a grande quantidade de requisitos, seriam altas as chances que todos os 23 padrões pudessem ser exemplificados a partir da mesma especificação.

Foram listados todos os requisitos da especificação, como eram muitos, foram selecionados apenas os requisitos obrigatórios.

Com a lista montada, foi iniciada a elaboração dos casos de uso, que continham os requisitos que eles atendiam. Isto foi preciso, pois a especificação é grande, e era preciso ter uma melhor visão e entendimento do sistema.

Após a elaboração dos casos de uso, foram listados os 23 padrões e cruzadas as suas características com os requisitos, para que sejam encontrados os que tinham relação para a futura elaboração dos exemplos. Após a elaboração dos exemplos, foi feita a conclusão do estudo. A seguir, os diagramas exemplificando cada um dos 23 padrões do catálogo estudado.

4.3.1 Abstract Factory

Este padrão pode ser utilizado na geração de relatórios, solicitados no caso de uso CSU 06 Gerar relatórios.

A classe GeradorDeRelatorios solicita à classe FabricaDeRelatorios o relatório desejado, podendo ser uma subclasse de RelatorioDeClassificacao e RelatorioDeTemporalidade.

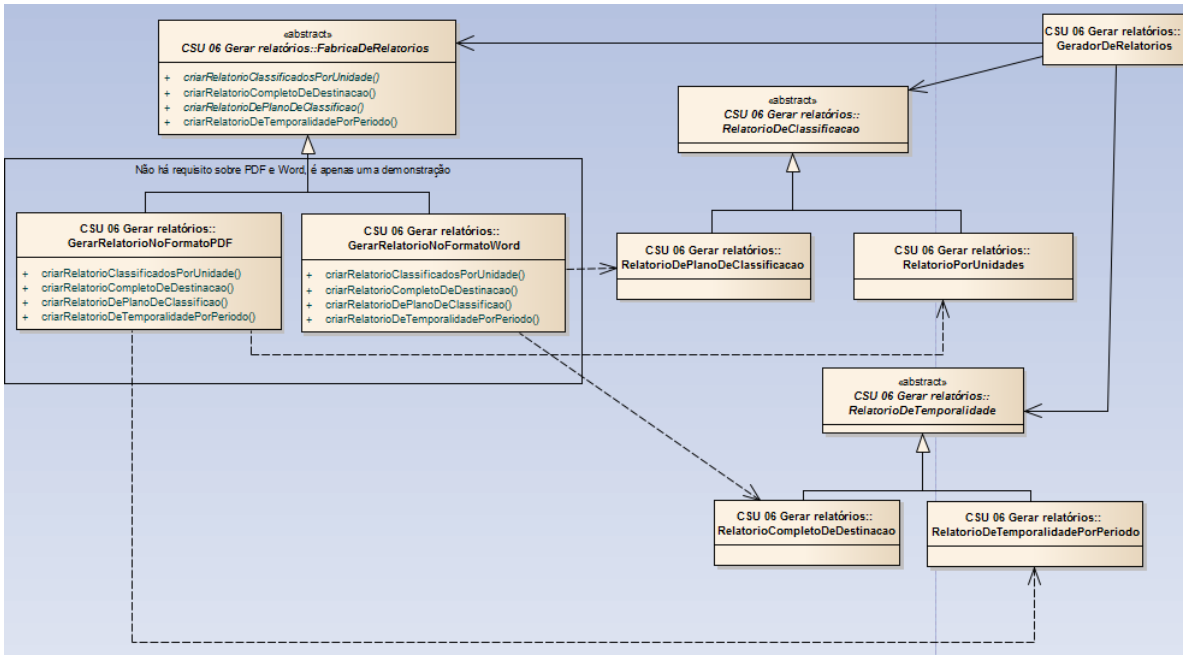


Figura 24: Proposta utilizando o padrão Abstract Factory para a geração de relatórios.
Fonte: O autor.

4.3.2 Builder

Este padrão pode ser utilizado no mecanismo de pesquisa por documentos. Este requisito é solicitado no caso de uso CSU 93 Pesquisar documento. Podem existir dois tipos de pesquisa: básica (onde são usados poucos critérios) e avançada (onde são usados vários critérios de busca).

A classe Pesquisador constrói um objeto do tipo ParametrosDePesquisa a partir da classe ContrutorDeParametrosDePesquisa. Primeiramente, é invocado o método construir, seguido do método getParametrosBasicos, para realizar uma pesquisa básica ou getParametrosAvancados, quando se deseja realizar uma pesquisa avançada.

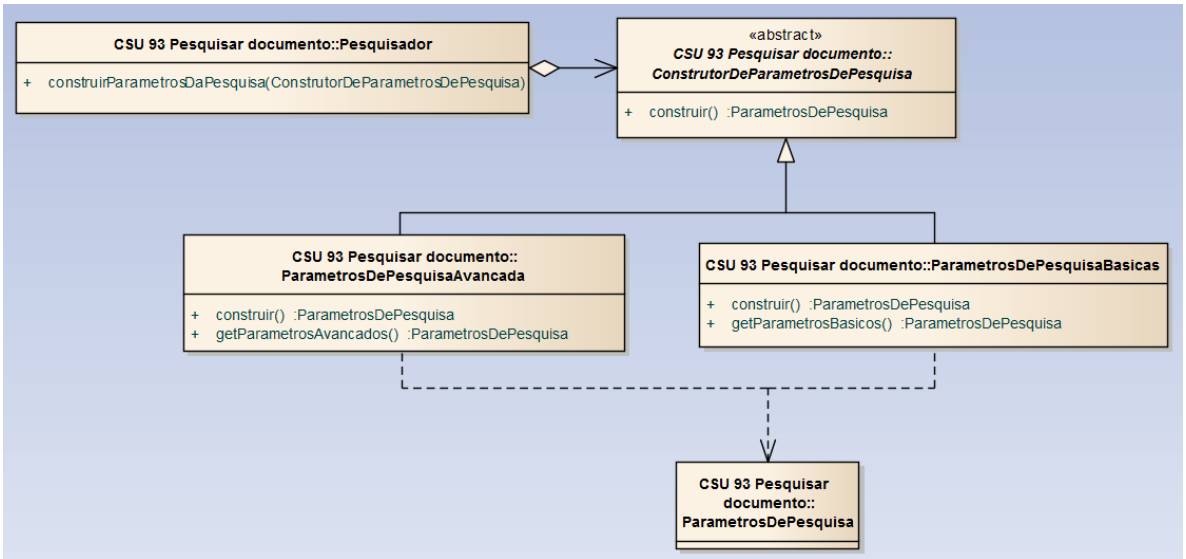


Figura 25: Proposta utilizando o padrão Builder para a pesquisa de documentos.
Fonte: O autor.

4.3.3 Factory Method

Este padrão pode ser utilizado na criação do objeto que irá armazenar os dados do usuário autenticado. Este requisito é solicitado no caso de uso CSU 100 Login.

É possível ter dois tipos de usuários autenticados: usuário simples e usuário administrador. Para construir o usuário simples é preciso utilizar a classe

CriadorDeUsuarioSimples, para o usuário administrador, deve-se utilizar CriadorDeUsuarioAdministrador.

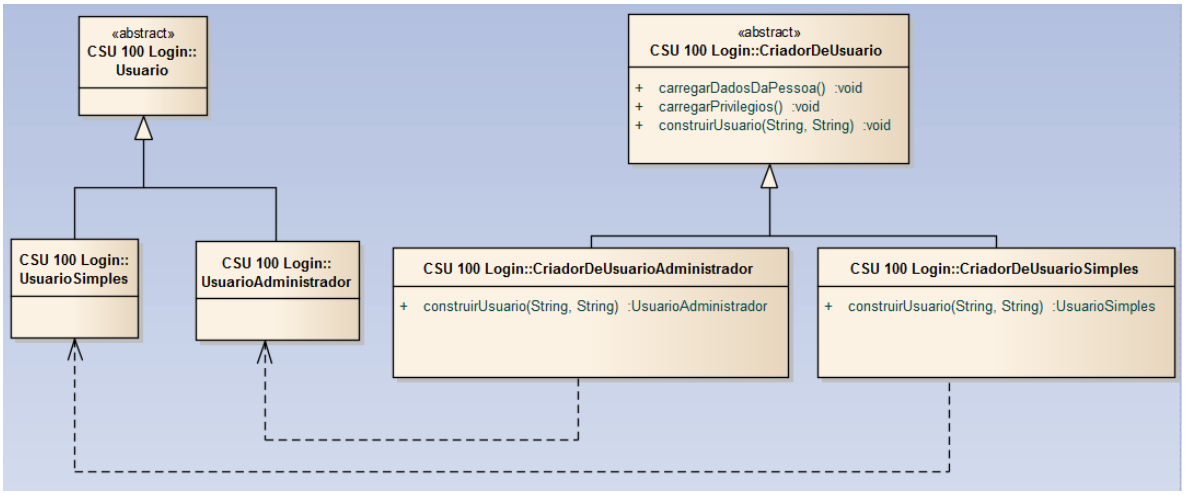


Figura 26: Proposta utilizando o padrão Factory Method para a criação de usuário.
Fonte: O autor.

4.3.4 Prototype

Este padrão pode ser utilizado na geração de relatório de auditoria, solicitado no caso de uso CSU 87 Gerar relatório de auditoria.

Para exemplificar a utilização deste padrão, é possível utilizar as classes RelatorioDeAuditoriaEmUnidadesDeArquivamento juntamente com a classe RelatorioDeAuditoriaEmDocumentosArquivisticos, cada uma com características específicas de geração do relatório. Como elas tendem a não mudar o estado interno, o padrão Prototype pode ser utilizado.

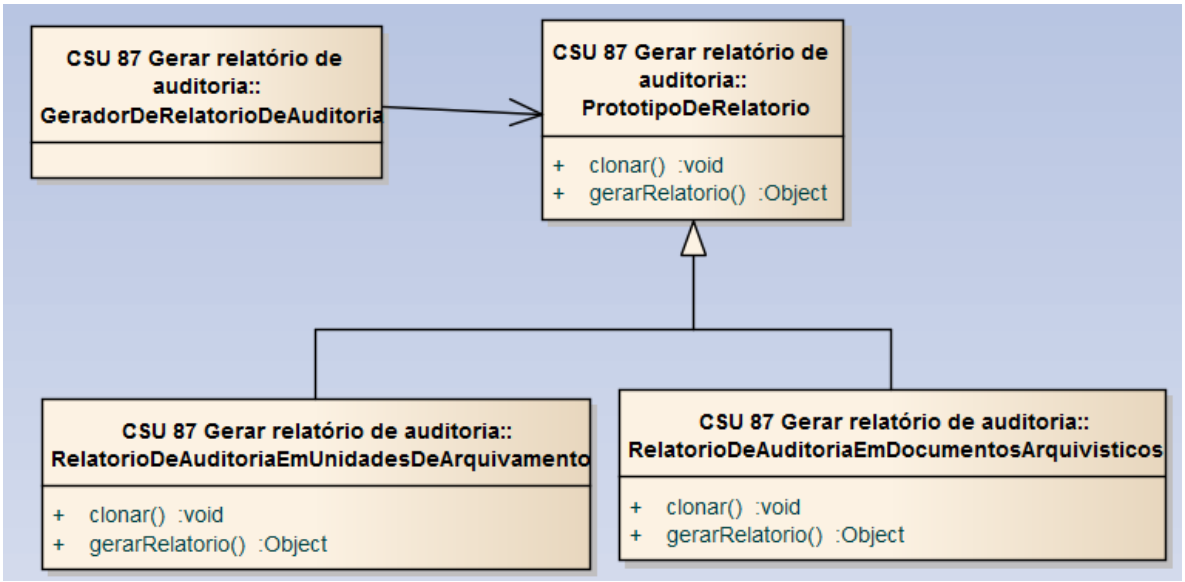


Figura 27: Proposta utilizando o padrão Prototype para geração de relatórios da auditoria.
Fonte: O autor.

4.3.5 Singleton

Este padrão pode ser utilizado na gerência de metadados em tempo de execução, solicitado no caso de uso CSU 67 Gerenciar metadados.

Os metadados podem ser carregados uma única vez, provavelmente no momento da inicialização, mantendo-se sempre o mesmo durante toda a execução.

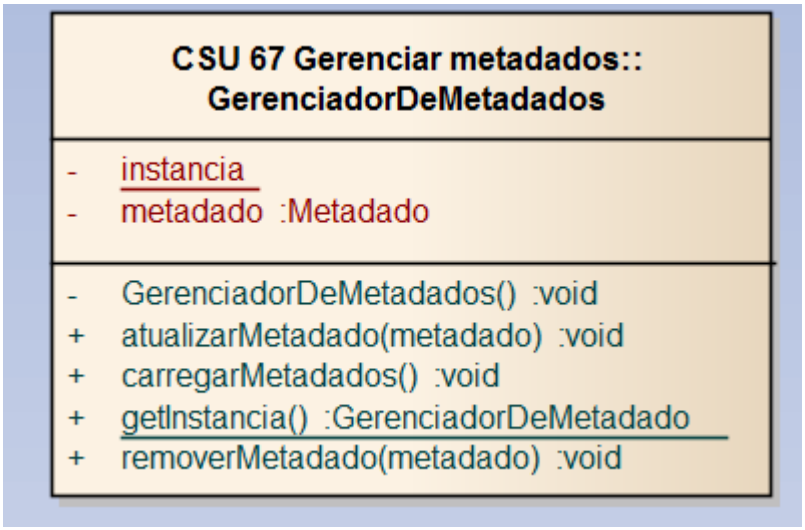


Figura 28: Proposta utilizando o padrão Singleton para gerenciar metadados.
Fonte: O autor.

4.3.6 Adapter

Este padrão pode ser utilizado na captura de documentos digitalizados, solicitado no caso de uso CSU 64 Capturar documento.

Neste exemplo, foi demonstrado como seria possível utilizar este padrão para encapsular a comunicação do sistema com o driver do aparelho scanner, responsável pela digitalização de um documento em papel.

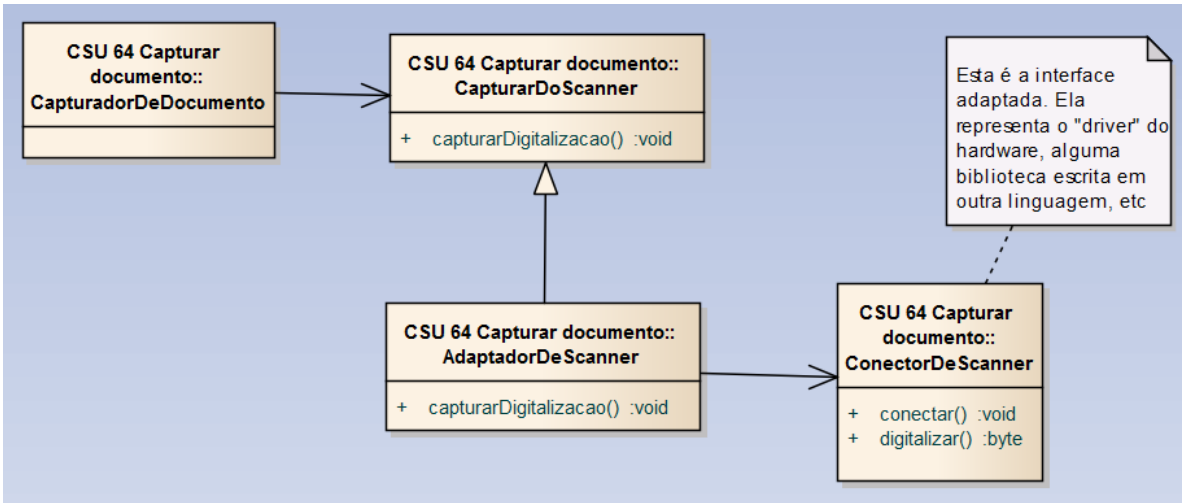


Figura 29: Proposta utilizando o padrão Adapter para a captura de documentos digitalizados.
Fonte: O autor.

4.3.7 Bridge

Este padrão pode ser utilizado na gestão de processos e dossiês, solicitado no caso de uso CSU 70 Gerenciar dossiê/processo.

As subclasses da classe Acao, são responsáveis por invocar o método registrarMetadados da classe RegistradorDeMetadados, que por sua vez, podem ser registrar vários tipos de metadados, através das classes: MetadadosGerais, DataDeEncerramento e DataDeAbertura.

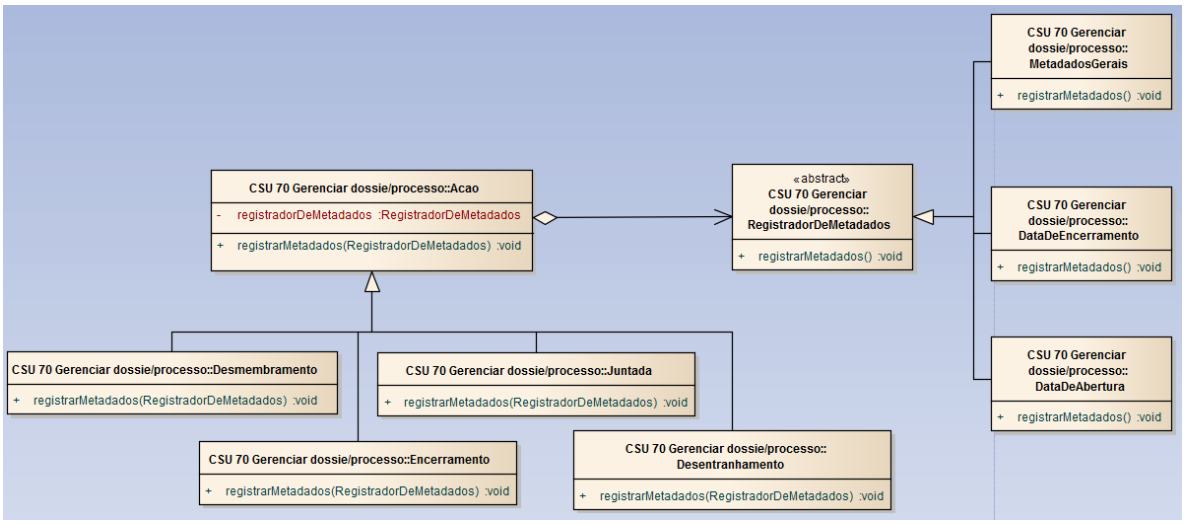


Figura 30: Proposta utilizando o padrão Bridge para registrar os metadados de processos e dossies.
Fonte: O autor.

4.3.8 Composite

Este padrão pode ser utilizado na gestão de unidades de arquivamento, solicitado no caso de uso CSU 85 Gerenciar unidades de arquivamento.

Uma classe pode conter várias unidades de arquivamento.

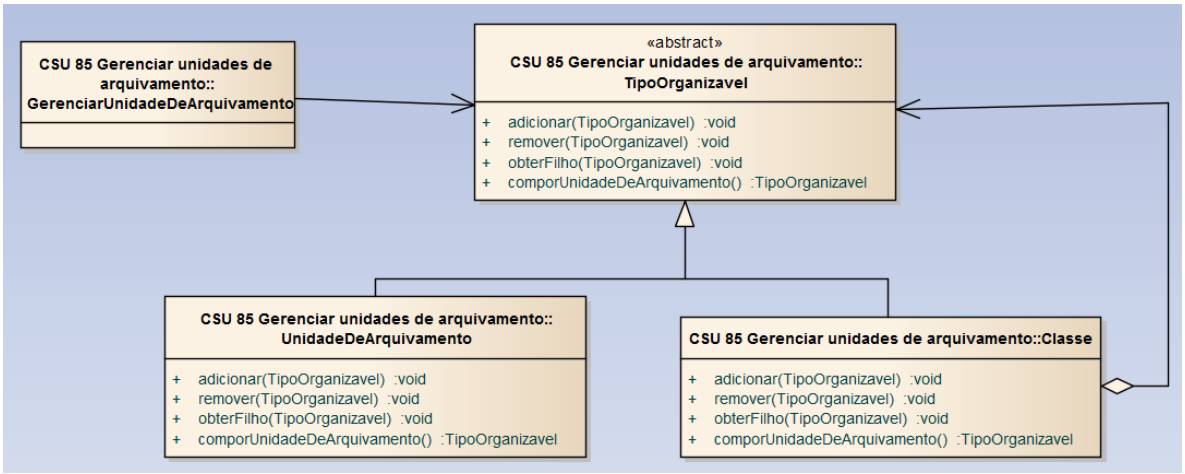


Figura 31: Proposta utilizando o padrão Composite gestão de unidades de arquivamento.
Fonte: O autor.

4.3.9 Decorator

Este padrão pode ser utilizado na exibição do detalhamento de um documento capturado, solicitado no caso de uso CSU 92 Exibir detalhes do documento capturado.

Um documento contendo textos que tenha sido capturado pode sofrer algumas alterações no momento da exibição. Através da classe MelhoriasNaVisualizacao, é possível, por exemplo destacar palavras-chave e valores de atributos contido nos metadados. Caso haja necessidade de destacar outra característica do documento exibido, basta estender a classe MelhoriasNaVisualizacao e implementar o comportamento necessário.

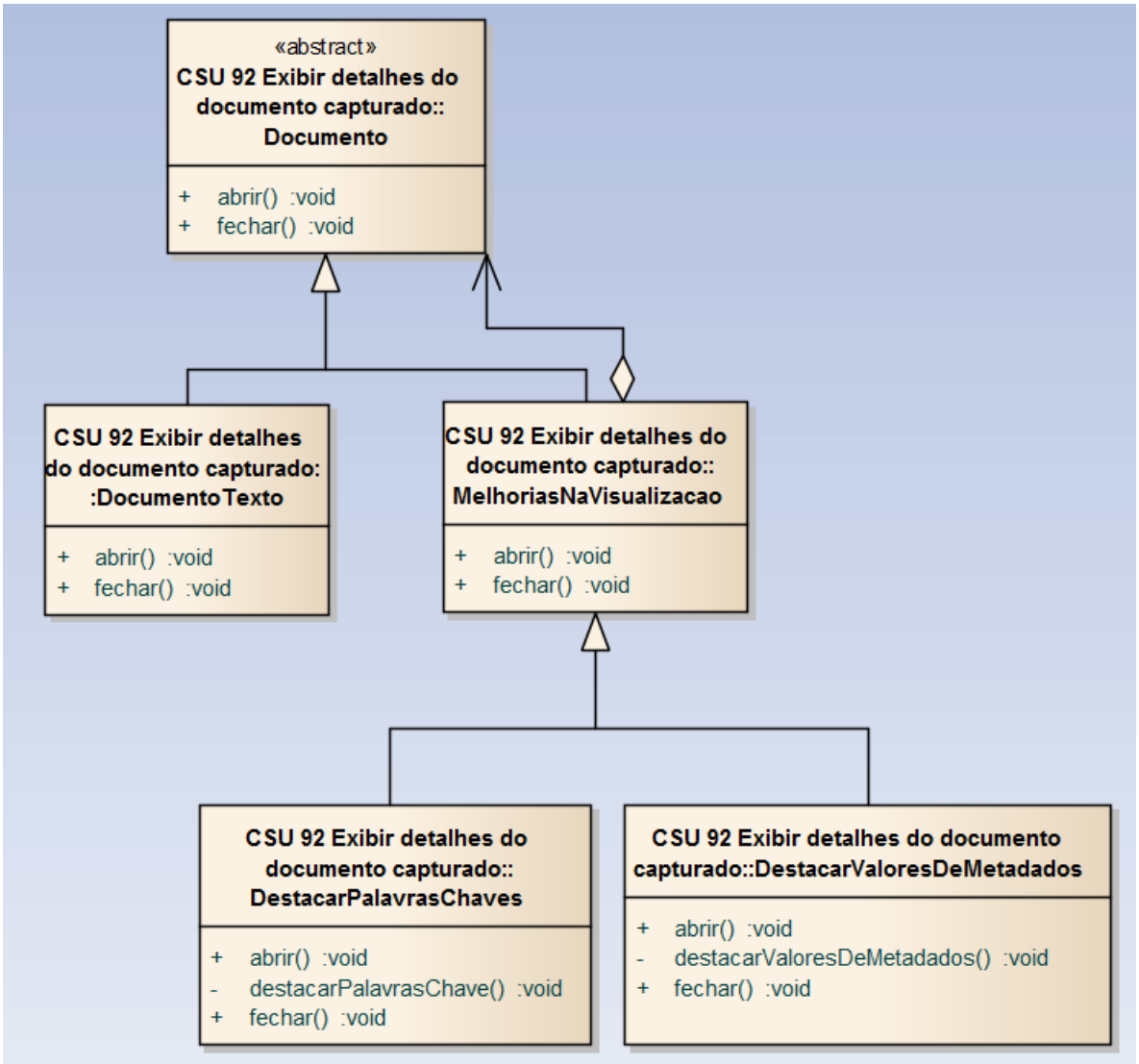


Figura 32: Proposta utilizando o padrão Decorator adicionar características na apresentação do documento que tenha sido capturado.
Fonte: O autor.

4.3.10 Façade (ou Facade)

Este padrão pode ser utilizado na rotina que realiza o backup dos dados do sistema, solicitado no caso de uso CSU 101 Salvar cópia de segurança.

A rotina de backup do sistema pode ser separada em várias etapas, que não necessitam estarem expostas ao cliente, afinal, para ele basta iniciar a geração do arquivo e ter a resposta. Desta forma, todos os sub-processos, que fazem parte da

geração do backup ficam encapsuladas pela classe FachadaDeExportacao que, por sua vez, se encarregar de realizar todos os passos necessários.

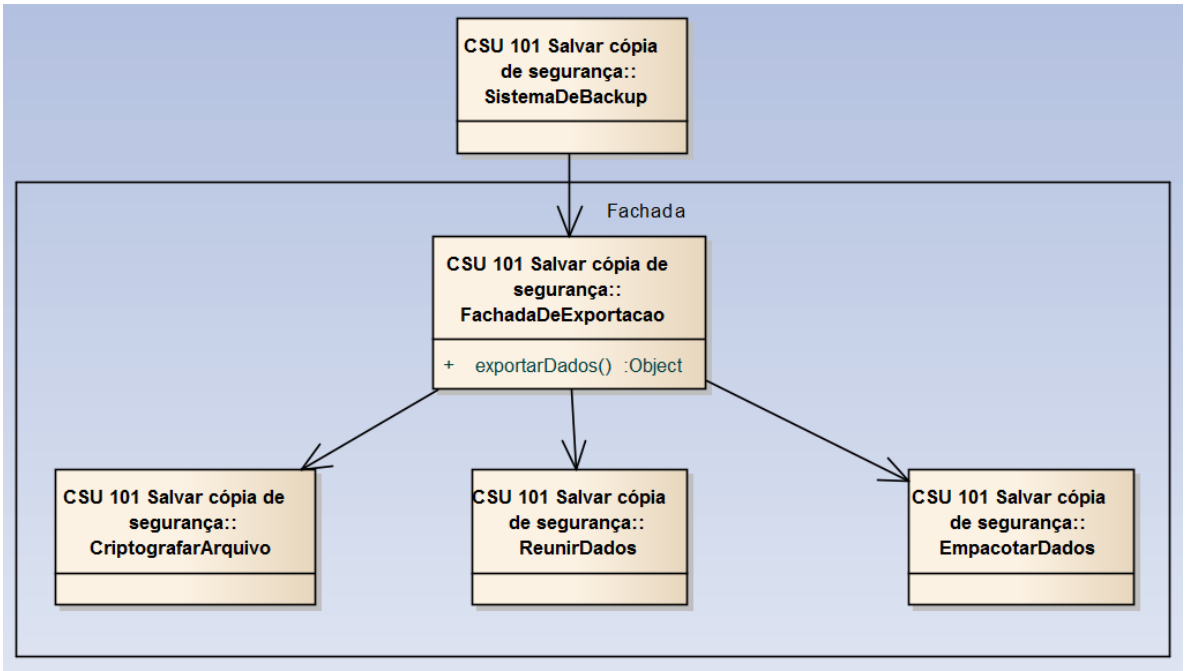


Figura 33: Proposta utilizando o padrão Façade para realizar a rotina de backup.
Fonte: O autor.

4.3.11 Flyweight

Este padrão pode ser utilizado na recuperação dos vários tipos de metadados utilizados sistema, solicitado no caso de uso CSU 64 Capturar documento.

O sistema pode conter uma grande quantidade de metadados, dado que deve manipular documentos nos mais variados formatos, tipos e finalidades. Devido há essa possibilidade de haver muitos metadados, a utilização do padrão pode ser útil, a fim de reduzir o consumo computacional, compartilhando as instâncias já existentes dos tipos de meta dados.

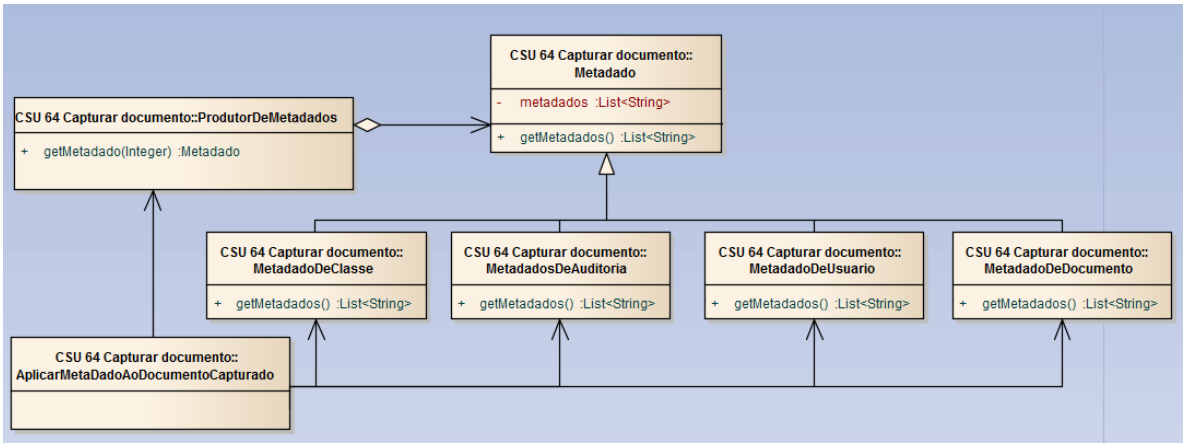


Figura 34: Proposta utilizando o padrão Flyweight na utilização dos metadados.
Fonte: O autor.

4.3.12 Proxy

Este padrão pode ser utilizado na manipulação de uma cópia do documento, para não alterar ou ter acesso direto ao documento original, solicitado no caso de uso CSU 94 Gerar extrato de documento.

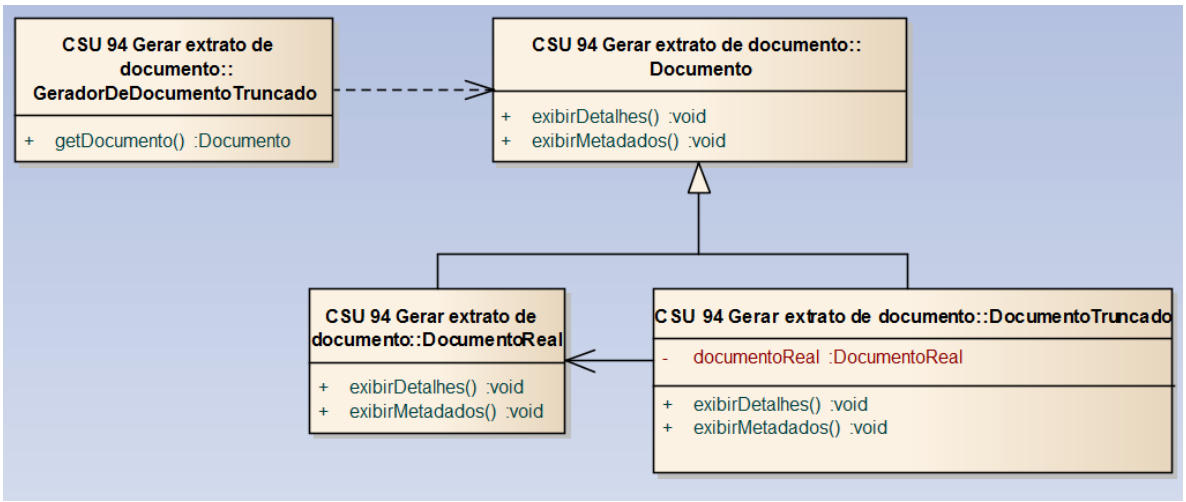


Figura 35: Proposta utilizando o padrão Proxy a manipulação de documentos que sejam cópias dos originais.
Fonte: O autor.

4.3.13 Chain of Responsibility

Este padrão pode ser utilizado para aplicar a regra de sigilo para cada tipo de artefato manipulado pelo sistema, solicitado no caso de uso CSU 90 Registrar transferência.

Cada artefato utilizado no sistema pode ter sua regra de sigilo específica, para que ela seja devidamente aplicada, pode utilizar o padrão para que o respectivo objeto que contenha essa regra tenha a chance de aplica-la corretamente.

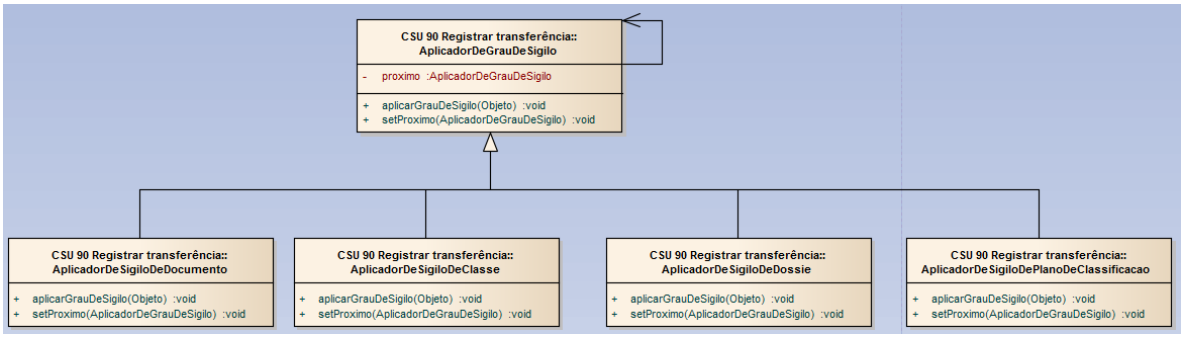


Figura 36: Proposta utilizando o padrão Chain of Responsability para a aplicação das regras de sigilo.
Fonte: O autor.

4.3.14 Command

Este padrão pode ser utilizado na gestão de classes, subclasses, grupos e subgrupos utilizados no sistema, solicitado no caso de uso CSU 07 Gerenciar classes, subclasses, grupos e subgrupos.

Neste caso, há dois exemplos de utilização. O primeiro trata a situação de registro de metadados. Este comando pode ser invocado pela classe RegistrarAberturaNoMetadado e realizado pela classe Metadados. O outro caso aborda a situação em que uma ação é realizada, que neste exemplo pode ser a mudança de nome ou apagar uma classe. A classe PersistenciaDeClasseEGrupo irá receber ambos os comandos.

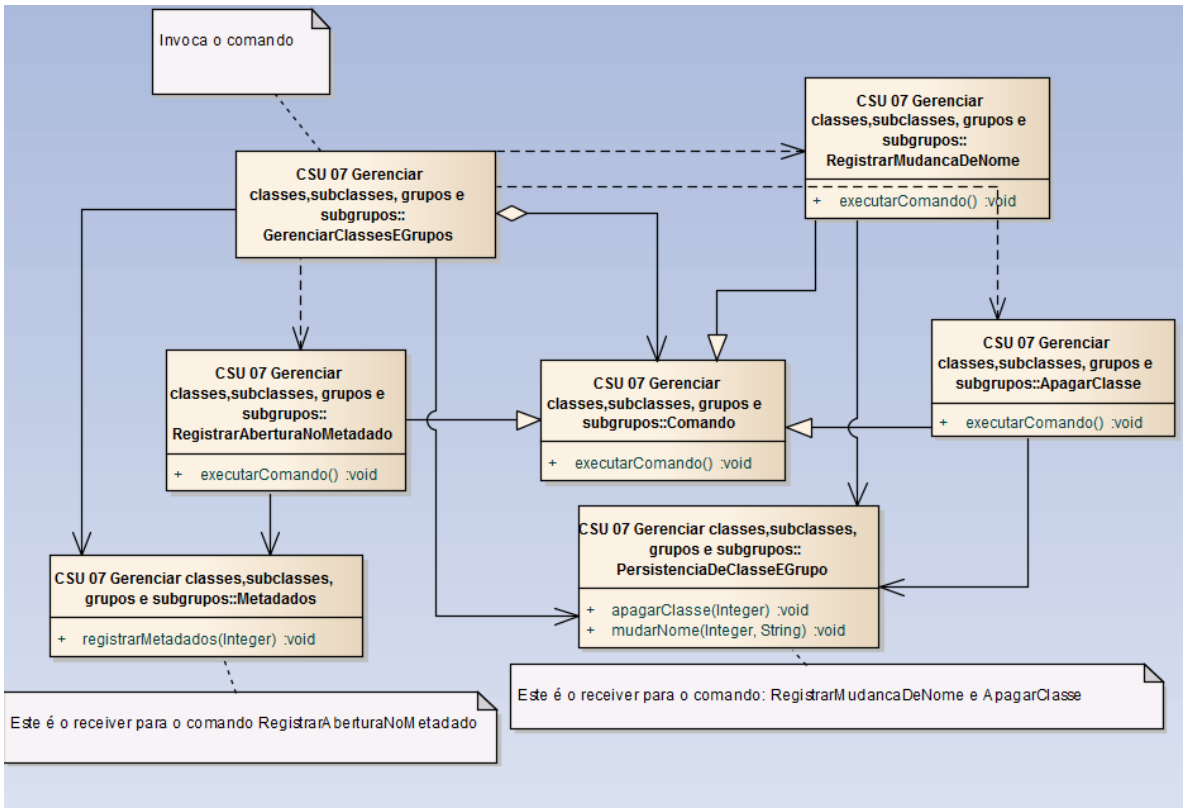


Figura 37: Proposta utilizando o padrão Command registar a abertura no metadado, apagar uma classe ou registrar a mudança de nome.
Fonte: O autor.

4.3.15 Interpreter

Este padrão pode ser utilizado no mecanismo de busca por um documento, solicitado no caso de uso CSU 93 Pesquisar documento.

Pode ser que um usuário com conhecimento avançado, queira realizar uma busca mais detalhada, inserindo o termo a ser pesquisado contendo os atributos do documento que ele já conheça. Para isso deve existir uma regra para a elaboração dos atributos a serem pesquisados, que pode ser: nome do atributo, seguido do símbolo de igualdade (=), seguido pelo valor e encerrado ponto-e-vírgula (;), exemplo: “titulo=nome do documento;data=01/01/2015;”.

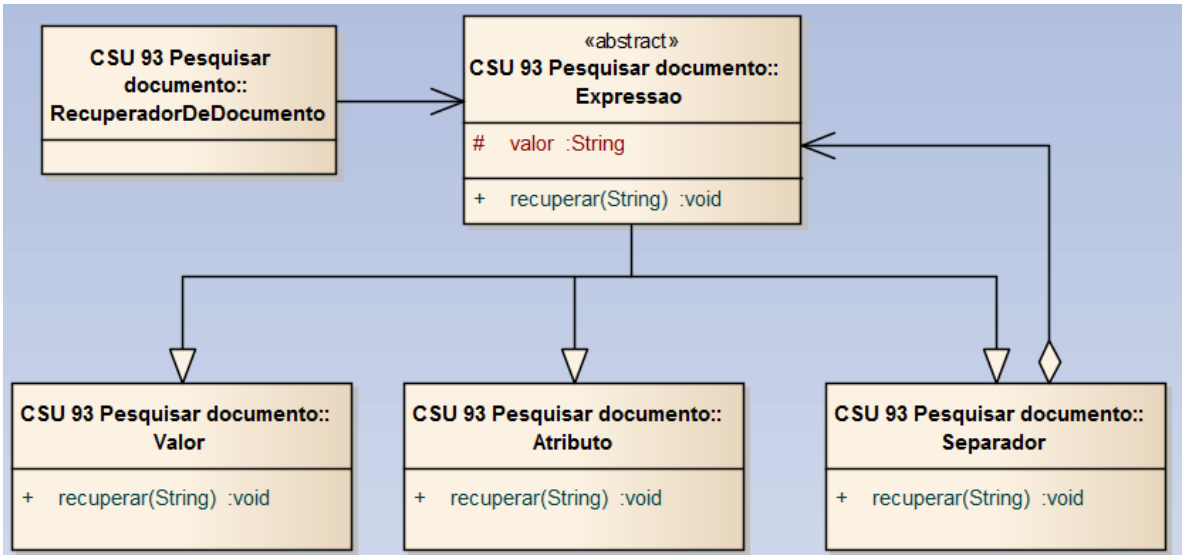


Figura 38: Proposta utilizando o padrão Interpreter para procurar por um documento no sistema.
Fonte: O autor.

4.3.16 Iterator

Este padrão pode ser utilizado para a exportação de um documento, solicitado no caso de uso CSU 59 Exportar documento.

Um documento pode conter outros documentos, ao exportar um documento, é preciso navegar por todos os documentos presentes em um documento e exporta-los.

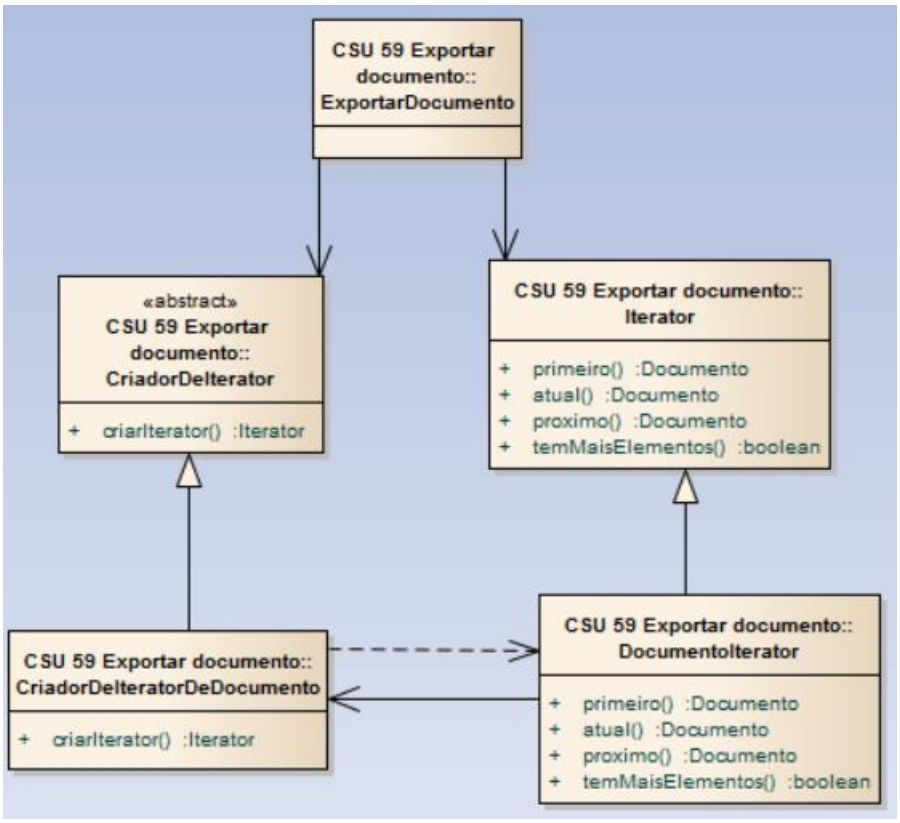


Figura 39: Proposta utilizando o padrão Iterator navegar em outros documentos contidos nele.
Fonte: O autor.

4.3.17 Mediator

Este padrão pode ser utilizado para geração do relatório de desempenho, solicitado no caso de uso CSU 65 Emitir relatório de desempenho.

Para gerar o relatório de desempenho, é necessário, por exemplo, recuperar dados de tramitação de documentos e desempenho dos participantes, para isso, a classe EmitirRelatorio solicita à classe EmitirRelatorioDeDesempenho o relatório, que por sua vez irá solicitar às respectivas classes TramitacaoDeDocumentos e DesempenhoDosParticipantes que forneçam seus relatórios. A classe EmitirRelatorioDeDesempenho irá juntar todos os relatórios e devolver como um único relatório.

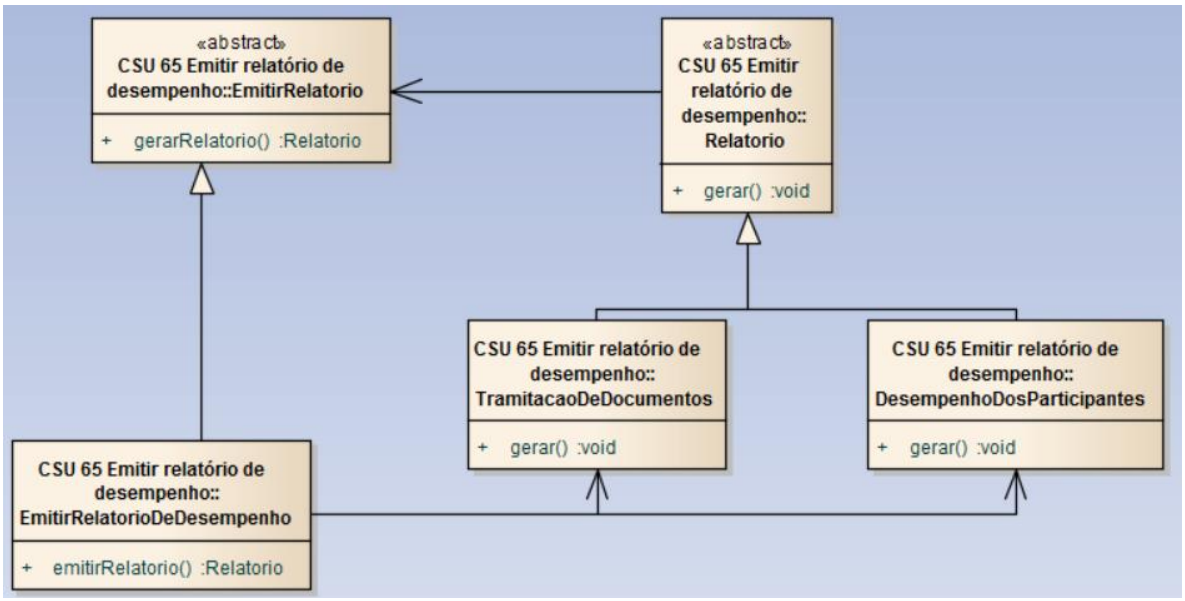


Figura 40: Proposta utilizando o padrão Mediator navegar em outros documentos contidos nele.
Fonte: O autor.

4.3.18 Memento

Este padrão pode ser utilizado para a restauração do sistema ao último estado consistente, solicitado no caso de uso CSU 82 Restaurar sistema.

Pode ser que ocorra alguma falha no sistema durante seu uso, desta forma, é preciso recuperar o último estado consistente do sistema. O exemplo a seguir, irá mostrar apenas para o documento, mas poderia ser aplicado a outros tipos. A Classe HistoricoDeDocumento se encarrega de guardar o documento que pode ser necessário futura recuperação. Como a necessidade é apenas para recuperar o último, o método recuperarUltimo pode ser utilizado, porem, caso seja necessário utilizar um documento em um momento específico, é possível ter acesso ao atributo lista que guarda todos os estados daquele documento, para isso, é preciso criar um método que permita o acesso a lista, que atualmente, não existe pois não há necessidade.

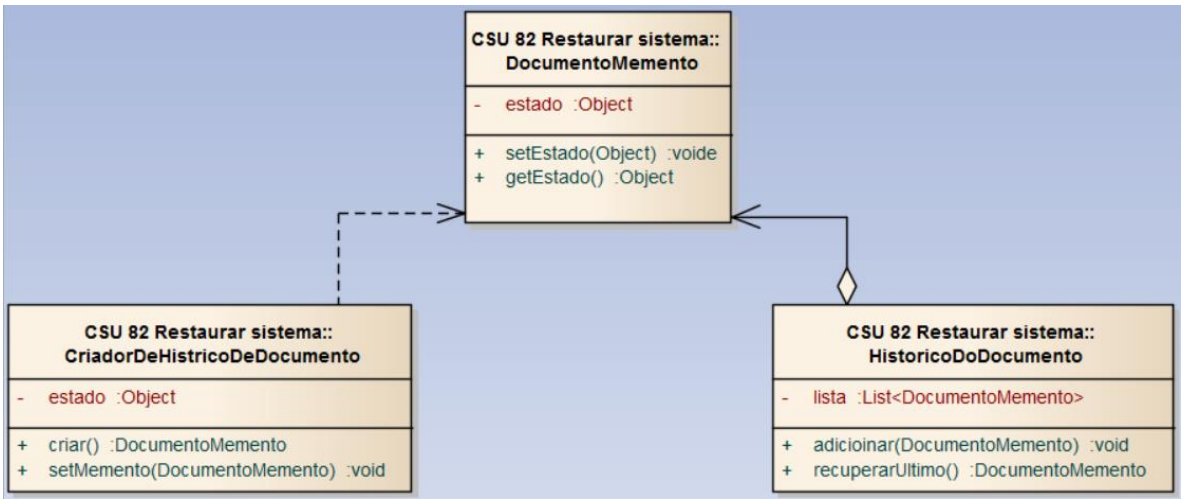


Figura 41: Proposta utilizando o padrão Memento para restaurar o sistema.
Fonte: O autor.

4.3.19 Observer

Este padrão pode ser utilizado para registrar na auditoria cada ação realizada no documento, solicitado no caso de uso CSU 69 Gerenciar documentos e processos/dossiês.

Ao manipular um documento, é preciso que a respectiva ação seja registrada na auditoria. Toda vez que uma ação é exercida em um documento pela classe OperacoesEmDocumento, ela notifica o observador (que no exemplo é a classe RegistradorDeAuditoria), que por sua vez, se encarrega de inserir o registro na auditoria.

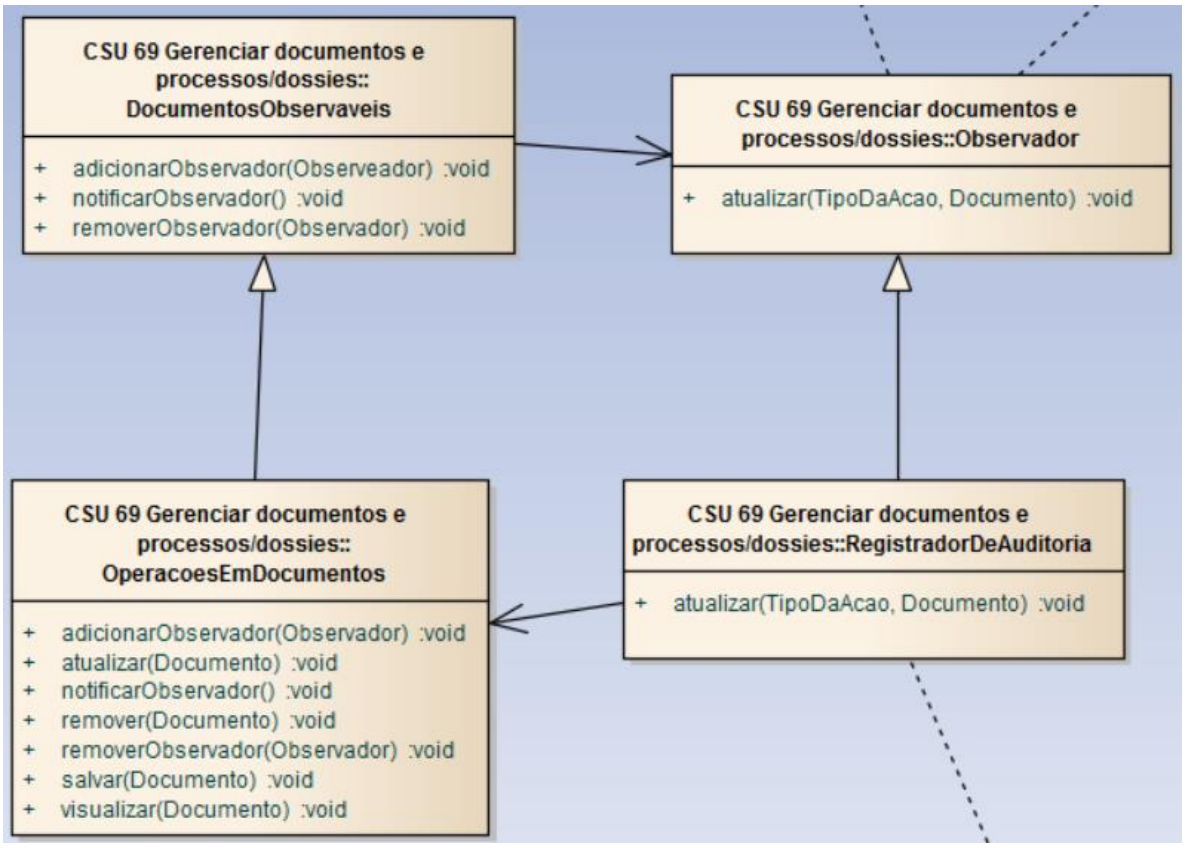


Figura 42: Proposta utilizando o padrão Observer para registrar na auditoria cada ação realizada no documento.
Fonte: O autor.

4.3.20 State

Este padrão pode ser utilizado para controlar a situação de um documento no fluxo, solicitado no caso de uso CSU 68 Gerenciar Tramitação e fluxo de trabalho.

Os fluxos do exemplo podem ser: guardado, disponível, em trânsito e publicado. A Tabela 1 mostra a relação entre os estados.

	Guardado	Disponível	Em trânsito	Publicado
Guardado				
Disponível				
Em trânsito				
Publicado				

Tabela 1: A cor amarela representa o próprio estado, a vermelha um próximo estado possível e a cor vermelha um estado que não poderá ser o próximo.

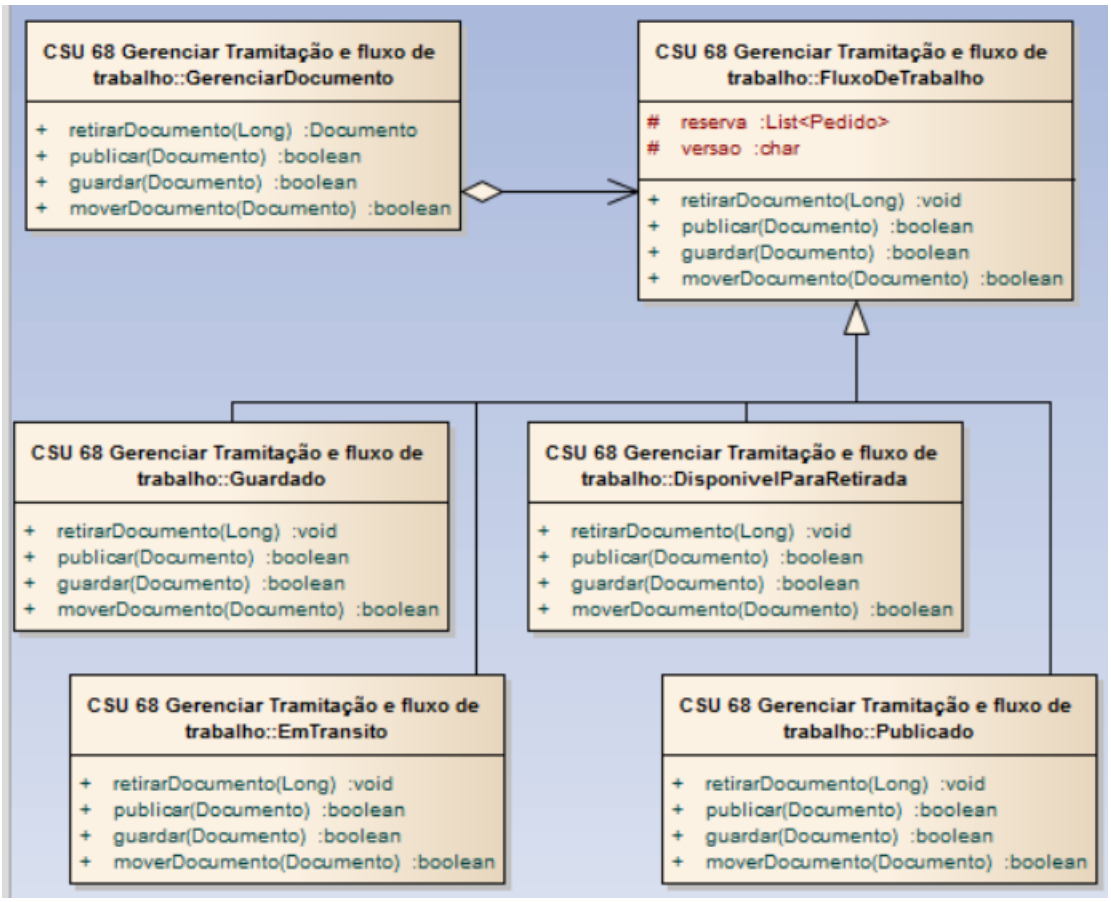


Figura 43: Proposta utilizando o padrão State para representar a relação entre os estados do documento no fluxo de trabalho.
Fonte: O autor.

4.3.21 Strategy

Este padrão pode ser utilizado na impressão documentos, solicitado no caso de uso CSU 80 Imprimir documento.

No exemplo, pode-se imprimir um documento em dois formatos: resumido e detalhado. Caso haja outro formato de impressão, para utiliza-lo é preciso estender a classe EstiloDeImpressao.

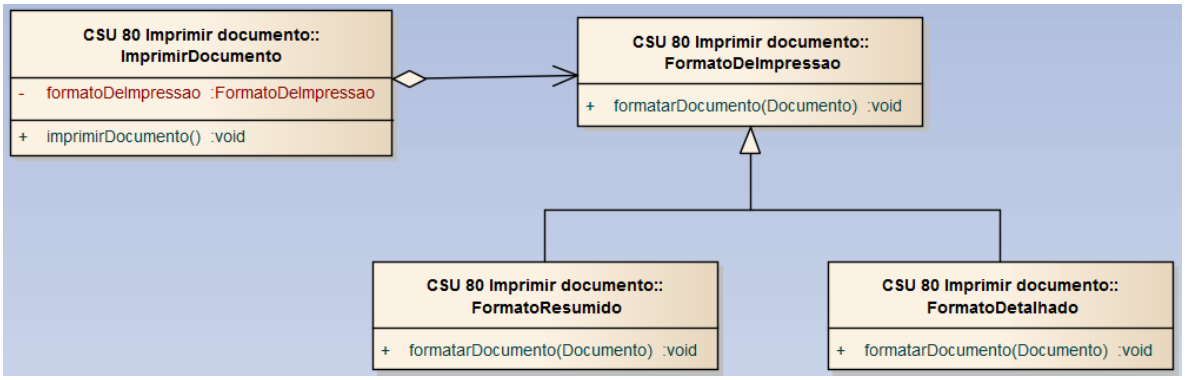


Figura 44: Proposta utilizando o padrão Strategy para impressão de documentos.
Fonte: O autor.

4.3.22 Template Method

Este padrão pode ser utilizado para aplicar uma sequencia de passo em cada interação com o documento, solicitado no caso de uso CSU 68 Gerenciar Tramitação e fluxo de trabalho.

As classes Tramitacao e FluxoDeTrabalho implementam seu comportamento específico para o passo registrarAuditoria, pois podem assumir comportamentos diferentes, porém, a classe GerenciarInteracaoComDocumento, executa os passos comuns para a interação.

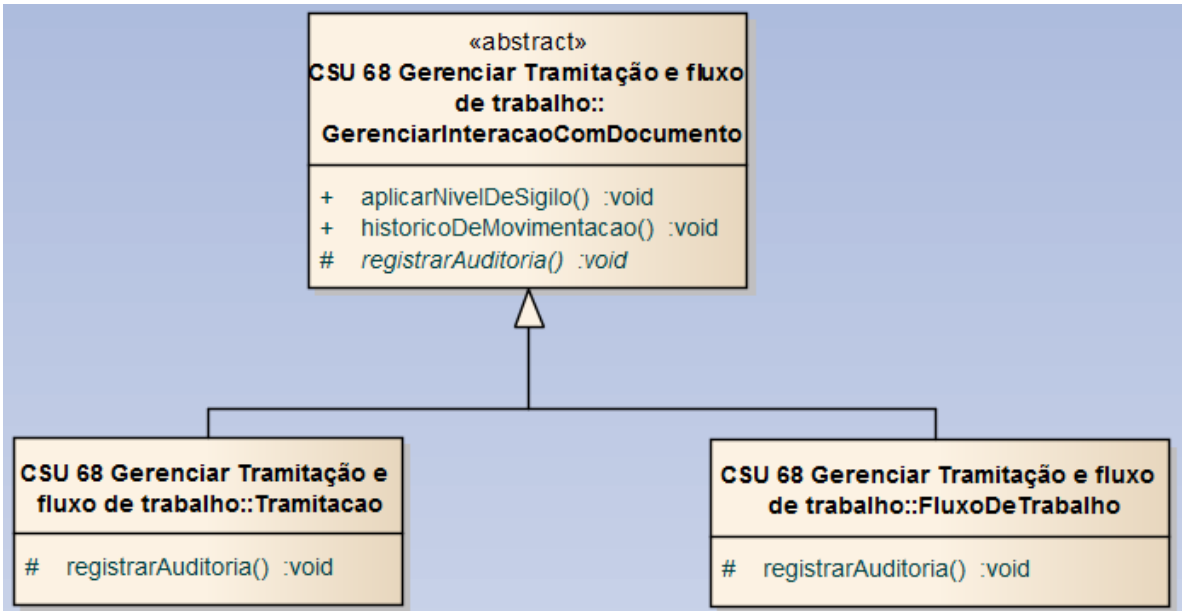


Figura 45: Proposta utilizando o padrão Template Method para impressão de documentos.
Fonte: O autor.

4.3.23 Visitor

Este padrão pode ser utilizado para definir o comportamento das classes ao aplicar o plano de classificação, solicitado no caso de uso CSU 50 Gerenciar plano de classificação.

Quando for necessário definir um comportamento para Classe, é preciso passar o comportamento desejado, que pode ser AtribuirIdentificado, Eliminacao e Recolhimento. O mesmo vale para Grupos.

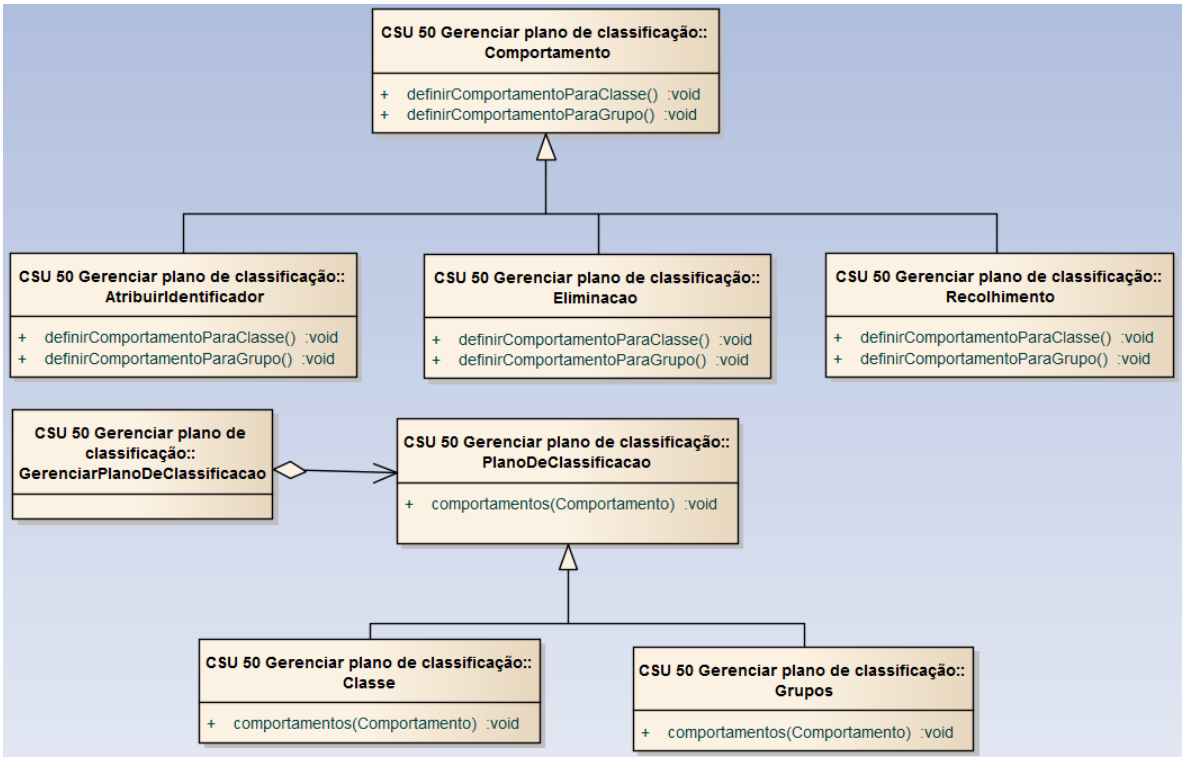


Figura 46: Proposta utilizando o padrão Visitor para definir o comportamento do plano de classificação.
Fonte: O autor.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo irá apresentar as conclusões deste trabalho seguido de proposta para melhorias ou evoluções a partir deste.

5.1 CONCLUSÃO

Este trabalho apresenta a aplicação prática dos 23 padrões de projeto do catálogo GoF utilizando a especificação do CONARQ para sistemas informatizados de gestão arquivística de documentos (SIGAD)

Ao iniciar o estudo, o maior trabalho foi à listagem e seleção dos requisitos, pois eram centenas, foi um trabalho demora e minucioso, para que nenhum requisito que não esteja no escopo, fosse esquecido.

Como a gestão de arquivos, não é uma área de conhecimento amplamente conhecida, houve um pouco de dificuldade em encontrar bons materiais de estudo, inclusive, foi necessário recorrer à ajuda de uma especialista da área nos momentos de dúvida.

A lista de requisitos ou casos de uso que seriam exemplificados foi alterada inúmeras vezes, dado que em algumas situações os requisitos não se encaixavam tão bem quanto inicialmente se pensou.

Dos exemplos elaborados, os mais complicados foram o interpreter e o prototype, por terem utilizações mais específicas e que não é o foco da especificação estudada e o visitor, dado que é um dos diagramas mais difíceis de compreender.

Os padrões de projeto realmente contribuem para a melhor construção do software. Eles não estão presos a um problema específico, podem ser utilizados para resolver um requisito ou um conjunto deles.

Caso a aplicação cresça e novos algoritmos, estratégias, fluxos ou outra necessidade apareça, por utilizar padrões de projeto em seu desenvolvimento, certamente o esforço será menor para a adequação.

Em alguns momentos, a diferença entre um padrão e outro é muito sutil, dificultando a diferenciação e principalmente saber qual o melhor padrão para o problema que se quer resolver.

É importante ter em mente que mesmo com todas as coisas boas que a utilização de padrões proporciona, é preciso usa-los com moderação, pois podem aumentar a quantidade de classes do software e pode comprometer a simplicidade do código. Também é preciso entender que os padrões não resolvem todos os problemas e algumas vezes, pode ser preciso mesclar padrões e até realizar adaptações em algum padrão para resolver um problema específico.

5.2 TRABALHOS FUTUROS

Este trabalho ficou restrito aos requisitos obrigatórios da especificação do SIGAD, os exemplos poderiam ser estendidos aos outros requisitos da especificação. Como há outros catálogos de padrões disponíveis na literatura, seria interessante selecionar outro catálogo e aplica-lo à especificação.

Como evolução, é interessante levar os diagramas além, construir os exemplos com código executável e até mesmo construir uma aplicação funcional utilizando os casos de uso e exemplo de uso dos padrões.

REFERÊNCIAS

BOOCH, Grady; RUMBAUGH, James e JACOBSON, Ivar. **UML Guia do Usuário.** Editora Campus, 2005.

BRASIL. Câmara Técnica de Documentos Eletrônicos. **E-ARQ Brasil:** Modelo de requisitos para sistemas informatizados de gestão arquivística de documentos. Rio de Janeiro: Arquivo Nacional, 2011. Disponível em: <http://www.conarq.arquivonacional.gov.br/media/carta.pdf>. Acesso em: 01 ago. 2014.

BRASIL. Conselho Nacional de Arquivos. **CARTA PARA A PRESERVAÇÃO DO PATRIMÔNIO ARQUIVÍSTICO DIGITAL.** Rio de Janeiro: Câmara Técnica de Documentos Eletrônicos, 2005. Disponível em: <http://www.conarq.arquivonacional.gov.br/media/carta.pdf>. Acesso em: 09 nov. 2014.

BRASIL. Conselho Nacional de Arquivos. **Criação e desenvolvimento de arquivos públicos municipais:** transparência e acesso à informação para o exercício da cidadania. Rio de Janeiro: Conselho Nacional de Arquivos, 2014. Disponível em: < http://www.conarqarquivosmunicipais.arquivonacional.gov.br/media/publicacoes/criacao_arquivos_municipais_site.pdf>. Acesso em: 09 nov. 2014.

GUEDES, Gilleanes T. A.. **UML 2:** uma abordagem prática. 2. ed. São Paulo: Novatec, 2011.

FREEMAN, Eric et al. **Design Patterns**. Sebastopol: O'Reilly, 2004.

FOWLER, Martin. **UML Essencial: Um Breve Guia para Linguagem Padrão.** 3. ed. São Paulo: Bookman, 2004.

GUERRA, Eduardo. **Design Patterns com Java:** Projeto Orientado a Objetos Guiado por Padrões. São Paulo: Casa do Código, 2013.

HAMILTON, Kim; MILES, Russell. **Learning UML 2.0.** Sebastopol: O'Reilly, 2006.

HOLZNER, Steve. **Design Patterns For Dummies.** Indianapolis: Wiley Publishing, 2006.

LARMAN, Craig. **Utilizando UML e Padrões.** 3. ed. Porto Alegre: Bookman, 2005.

METSKER, Steven John; WAKE, William C. **Design Patterns in Java.** 2. ed. Boston: Pearson Education, 2006.

METSKER, Steven John. **Design Patterns Java Workbook.** Indianápolis: Addison Wesley, 2002.

PRESSMAN, Roger S. **Engenharia de Software:** Uma abordagem profissional. 7. ed. São Paulo: Mcgraw Hill Brasil, 2011.

RUNESON, Per; HÖST, Martin. Guidelines for conducting and reporting case study research in software engineering. **Empirical Software Engineering.**Lund, p. 131-164. abr. 2009. Disponível em: <http://link.springer.com/article/10.1007/s10664-008-9102-8>. Acesso em: 03 fev. 2015.

SHALLOWAY, Alan; TROTT, James R. **Design Patterns Explained: A New Perspective on Object-Oriented Design.** Indianapolis: Addison Wesley, 2001.

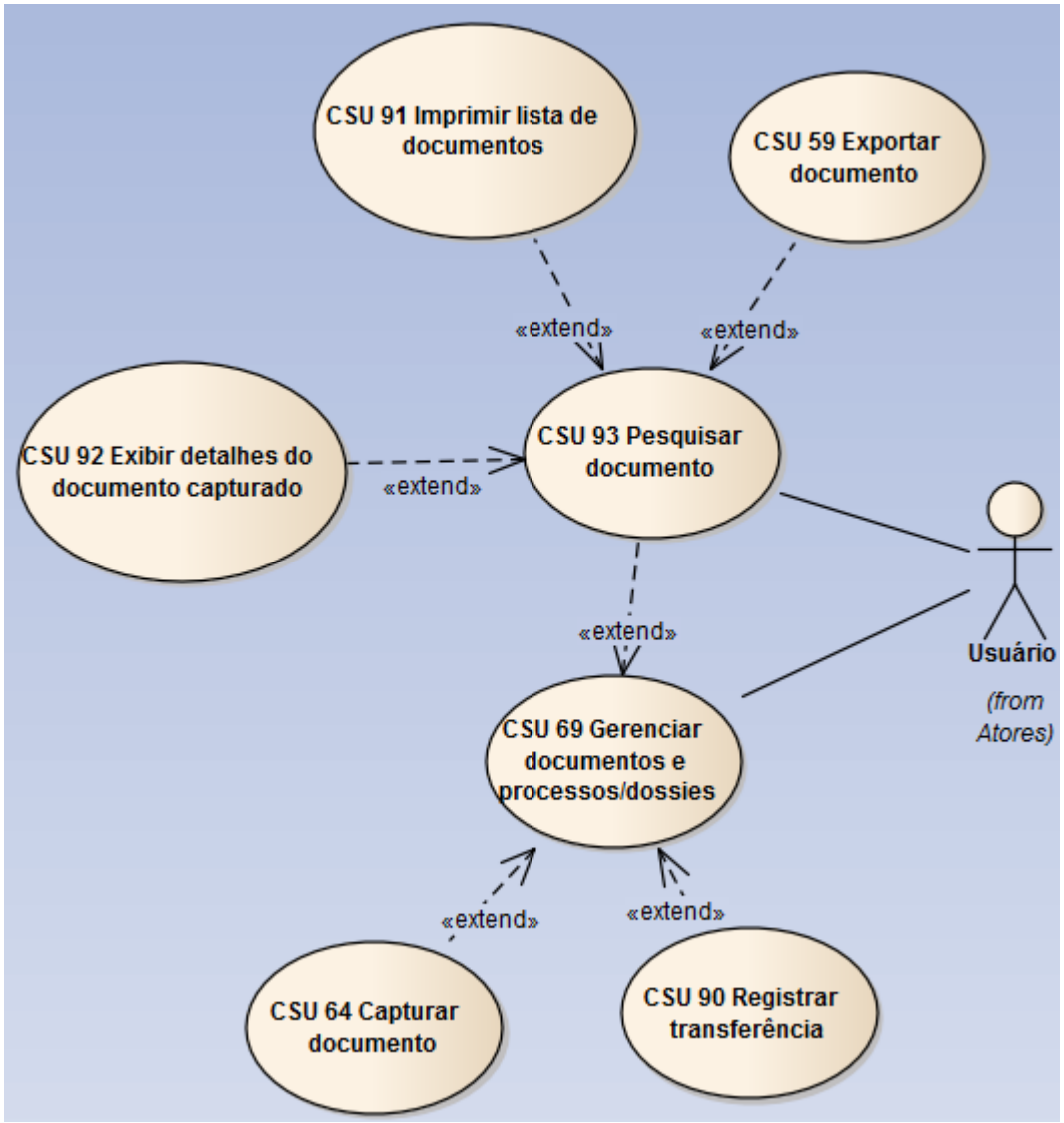
SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação.** 4. ed. Florianópolis: Universidade Federal de Santa Catarina, 2005. Disponível em: <https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes_4ed.pdf>. Acesso em: 03 jan. 2015.

SOMMERVILLE, Ian. **Engenharia de Software.** 8. ed. São Paulo: Addison-wesley Brasil, 2007.

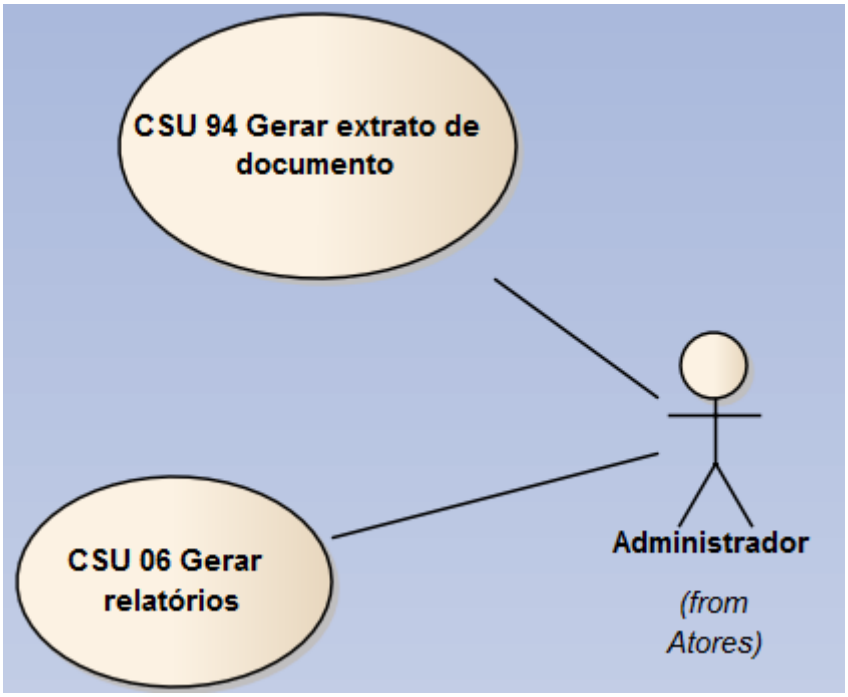
WAZLAWICK, Raul Sidnei. **Análise e Projeto de Sistemas da Informação.** 2. ed. Rio de Janeiro: Elsevier Brasil, 2011.

YODER, Joseph. Prefácio. In: GUERRA, Eduardo. **Design Patterns com Java:** Projeto Orientado a Objetos Guiado por Padrões. São Paulo: Casa do Código, 2013.

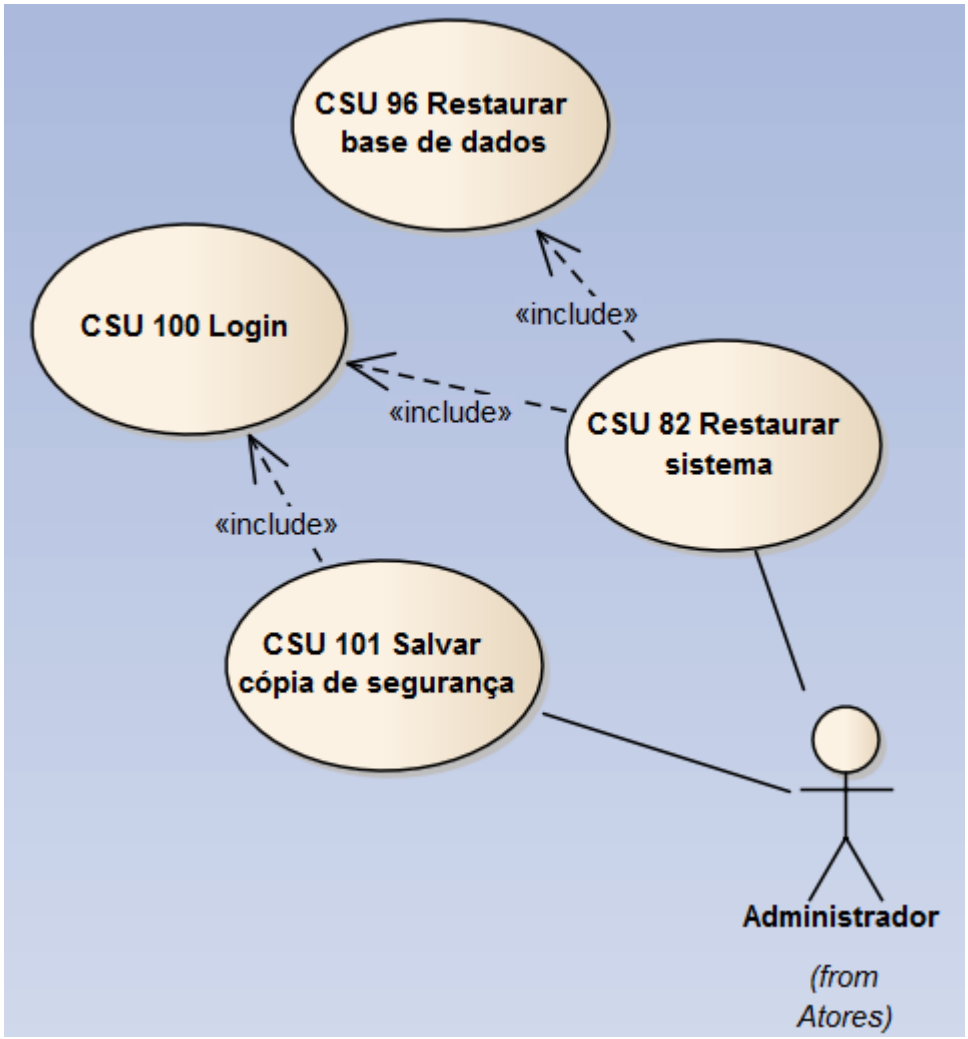
APÊNDICE A – Módulo de documentos



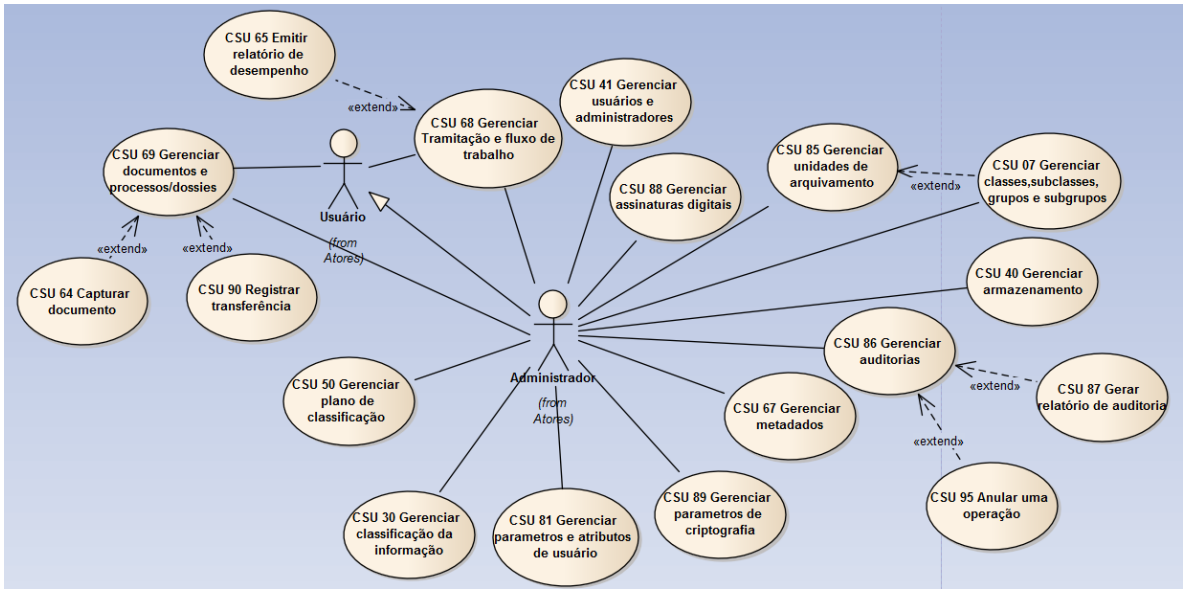
APÊNDICE B – Módulo de relatórios



APÊNDICE C – Módulo de restauração



APÊNDICE D – Módulo de gerenciamento



APÊNDICE E – Requisitos por caso de uso

Caso de uso: CSU 06 Gerar relatórios

RF 1.1.17 Um SIGAD tem que prover funcionalidades para elaboração de relatórios de apoio à gestão do plano de classificação incluindo a capacidade de:

- gerar relatório completo do plano de classificação;
- gerar relatório parcial do plano de classificação a partir de um ponto determinado na hierarquia;
- gerar relatório dos documentos ou dossiês/processos classificados em uma ou mais classes do plano de classificação;
- gerar relatório de documentos classificados por unidade administrativa.

RF 4.1.11 Um SIGAD tem que prover funcionalidades para elaboração de relatórios que apoiem a gestão da tabela de temporalidade e destinação, incluindo a capacidade de:

- gerar relatório completo da tabela de temporalidade e destinação de documentos;
- gerar relatório parcial da tabela de temporalidade e destinação de documentos a partir de um ponto determinado na hierarquia do plano de classificação;
- gerar relatório dos documentos ou dossiês/processos aos quais foi atribuído um determinado prazo de guarda;
- identificar as inconsistências existentes entre a tabela de temporalidade e destinação de documentos e o plano de classificação.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 07 Gerenciar classes, subclasses, grupos e subgrupos

RF 1.1.4 Um SIGAD tem que registrar a data de abertura de uma nova classe no respectivo metadado.

RF 1.1.5 Um SIGAD tem que registrar a mudança de nome de uma classe já existente no respectivo metadado.

RF 1.1.6 Um SIGAD tem que permitir o deslocamento de uma classe inteira, incluídas as subclasses, grupo, subgrupos e documentos nela classificados, para outro ponto do plano de classificação. Nesse caso, é necessário fazer o registro do deslocamento.

RF 1.1.8 Um SIGAD tem que permitir que um usuário autorizado apague uma classe inativa. Só pode ser apagada uma classe que não tenha documentos nela classificados.

RF 1.1.9 Um SIGAD tem que impedir a eliminação de uma classe que tenha documentos nela classificados. Essa eliminação pode ocorrer a partir do momento em que todos os documentos ali classificados tenham sido recolhidos ou eliminados, e seus metadados.

RF 1.6.3 Um SIGAD tem que permitir que um conjunto específico de metadados seja configurado para os documentos ou dossiês/processos convencionais e incluir informações sobre o local de arquivamento.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 100 Login

RF 6.2.2 Um SIGAD tem que exigir que o usuário esteja devidamente identificado e autenticado antes de iniciar qualquer operação no sistema.

RF 6.2.5 Um SIGAD tem que permitir acesso a funções do sistema somente a usuários autorizados e sob controle rigoroso da administração do sistema, a fim de proteger a autenticidade dos documentos arquivísticos digitais.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 101 Salvar cópia de segurança

RF 6.1.1 Um SIGAD tem que permitir que, sob controle do seu administrador, mecanismos de backup criem cópias de todas as informações nele contidas (documentos arquivísticos, metadados e parâmetros do sistema).

RF 6.6.1 Um SIGAD tem que usar criptografia no armazenamento, na transmissão e na apresentação de documentos arquivísticos digitais ao implementar a política de sigilo.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 8.2.1 Um SIGAD tem que manter cópias de segurança. As cópias de segurança devem ser guardadas em ambientes seguros, em locais diferentes de onde se encontra a informação original.

RF 8.2.8 Um SIGAD tem que suportar a transferência em bloco de documentos (incluindo as demais informações associadas a cada documento) para outros suportes e/ou sistemas, de acordo com as normas aplicáveis aos formatos utilizados.

Caso de uso: CSU 102 Verificar existência de erros

RF 8.2.2 Um SIGAD tem que possuir funcionalidades para verificação periódica dos dados armazenados, visando à detecção de possíveis erros. Nesse caso, recomenda-se o uso de um checksum robusto, ou seja, que permita a constatação da integridade dos dados e seja seguro quanto a fraudes.

Caso de uso: CSU 30 Gerenciar classificação da informação

RF 6.3.10 Um SIGAD tem que permitir somente aos administradores autorizados a possibilidade de alterar a configuração dos valores predefinidos (default) para os atributos de segurança e marcação de graus de sigilo, quando necessário e apropriado.

RF 6.3.11 Somente administradores autorizados têm que ser capazes de realizar as seguintes ações:

- remover ou revogar os atributos de segurança dos documentos;
- criar, alterar, remover ou revogar as credenciais de segurança dos usuários.

RF 6.3.2 Um SIGAD tem que implementar a classificação de grau de sigilo baseando-se nos seguintes atributos de segurança:

- grau de sigilo do documento;
- credencial de segurança do usuário.

O grau de sigilo tem que estar associado à credencial de segurança.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 40 Gerenciar armazenamento

RF 7.1.2 A escolha de dispositivos tem que ser revista sempre que a evolução tecnológica indicar mudanças importantes.

RF 7.1.3 Atividades de migração têm que ser efetivadas, preventivamente, sempre que se torne patente ou previsível a obsolescência do padrão corrente.

RF 7.1.4 Para as memórias secundárias, um SIGAD tem que manter registro de MTBF (mean time between failure), 45 bem como suas datas de aquisição.

RF 7.1.5 Para as memórias secundárias e terciárias, um SIGAD tem que fazer o gerenciamento das mídias por meio do registro de durabilidade prevista, data de aquisição e histórico de utilização. As informações técnicas sobre previsibilidade de duração de mídias referidas no item/elemento 7.1.3 devem ser obtidas, preferencialmente, a partir de órgãos independentes. Quando isso não for possível, podem ser utilizadas informações de fornecedores. Em ambos os casos deve ficar registrada a origem da informação.

RF 7.1.9 Quando se proceder à eliminação de documentos, as memórias de suporte têm que ser, devidamente, “sanitizadas”, isto é, ter suas informações, efetivamente, indisponibilizadas. Este requisito aplica-se, principalmente, às memórias secundária e terciária, por sua característica não volátil. As informações devem ser eliminadas de forma irreversível, incluindo, no caso de memória terciária, a possibilidade de destruição física das mídias.

RF 7.2.1 Um SIGAD tem que possuir capacidade de armazenamento suficiente para acomodação de todos os documentos e suas cópias de segurança. Para grandes volumes de dados, é conveniente o uso de dispositivos com maior capacidade unitária de armazenamento, a fim de reduzir a sobrecarga operacional.

RF 7.2.2 Em um SIGAD, tem que ser prevista a possibilidade de expansão da estrutura de armazenamento. A quantidade de memória primária deve ser superestimada no momento da aquisição, a fim de minimizar as indisponibilidades do SIGAD nas situações de expansão desse tipo de memória. Quando da aquisição de disk arrays, as possibilidades de expansão dos equipamentos de controle devem ser consideradas. Para backups em fita magnética, em sistemas com grande volume de informação, devem ser utilizados sistemas automáticos de seleção, troca e controle de fitas (robots).

RF 7.3.2 Um SIGAD tem que utilizar técnicas de restauração de dados em caso de falhas.

RF 7.3.3 Um SIGAD tem que utilizar mecanismos de proteção contra escrita, que previnam alterações indevidas e mantenham a integridade dos dados armazenados.

RF 7.3.5 A integridade dos dispositivos de armazenamento tem que ser, periodicamente, verificada.

RF 8.1.1 Os suportes de armazenamento de um SIGAD têm que ser acondicionados, manipulados e utilizados em condições ambientais compatíveis com sua vida útil prevista e/ou pretendida de acordo com as especificações técnicas do fabricante e de entidades isentas, e com base em estatísticas de uso. A vida útil pretendida de um suporte pode ser menor que sua vida útil prevista, o que permite condições ambientais mais flexíveis.

RF 8.1.3 Um SIGAD tem que permitir o controle da vida útil dos suportes para auxiliar no processo de atualização.

RF 8.2.1 Um SIGAD tem que manter cópias de segurança. As cópias de segurança devem ser guardadas em ambientes seguros, em locais diferentes de onde se encontra a informação original.

RF 8.2.2 Um SIGAD tem que possuir funcionalidades para verificação periódica dos dados armazenados, visando à detecção de possíveis erros. Nesse caso, recomenda-se o uso de um checksum robusto, ou seja, que permita a constatação da integridade dos dados e seja seguro quanto a fraudes.

RF 8.2.3 Um SIGAD tem que permitir a substituição dos dados armazenados que apresentarem erros.

RF 8.2.7 Ações de preservação têm que ser efetivadas sempre que se torne patente ou previsível a obsolescência da tecnologia utilizada pelo SIGAD.

RF 8.2.8 Um SIGAD tem que suportar a transferência em bloco de documentos (incluindo as demais informações associadas a cada documento) para outros suportes e/ou sistemas, de acordo com as normas aplicáveis aos formatos utilizados.

Caso de uso: CSU 41 Gerenciar usuários e administradores

RF 6.2.1 Para implementar o controle de acesso, um SIGAD tem que manter pelo menos os seguintes atributos dos usuários, de acordo com a política de segurança:

- identificador do usuário;
- autorizações de acesso;
- credenciais de autenticação.

Senha, crachá, chave criptográfica, token USB, smartcard, biometria (de impressão digital, de retina etc.) são exemplos de credenciais de autenticação.

RF 6.2.11 Um SIGAD tem que implementar a política de controle de acesso a documentos por grupos de usuários considerando:

- a identidade do usuário e sua participação em grupos;
- os atributos de segurança, associados ao documento arquivístico digital, às classes e/ou aos dossiês/processos.

RF 6.2.13 Um SIGAD tem que permitir que um usuário pertença a mais de um grupo.

RF 6.2.15 Um SIGAD tem que usar os seguintes atributos do usuário ao implementar a política de controle de acesso aos documentos digitais por papéis de usuários:

- identificação do usuário;
- papéis associados ao usuário.

RF 6.2.16 Um SIGAD tem que usar os seguintes atributos dos documentos digitais ao implementar a política de controle de acesso por papéis:

- identificação do documento digital;
- operações permitidas aos vários papéis de usuários, sobre as classes ou unidades de arquivamento a que o documento pertence.

RF 6.2.18 Um SIGAD tem que impedir que um usuário assuma papéis com direitos conflitantes.

RF 6.2.3 Um SIGAD tem que garantir que os valores dos atributos de segurança e controle de acesso, associados ao usuário, estejam dentro de conjuntos de valores válidos.

RF 6.2.7 Somente administradores autorizados têm que ser capazes de criar, alterar, remover ou revogar permissões associadas a papéis de usuários, grupos de usuários ou usuários individuais.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;

- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 50 Gerenciar plano de classificação

RF 1.1.1 Um SIGAD tem que incluir e ser compatível com o plano de classificação do órgão ou entidade.

RF 1.1.11 Um SIGAD tem que disponibilizar pelo menos dois mecanismos de atribuição de identificadores a classes do plano de classificação.

Prevendo a possibilidade de se utilizar ambos, separadamente ou em conjunto, na mesma aplicação:

- atribuição de um código numérico ou alfanumérico;
- atribuição de um termo que identifique cada classe.

RF 1.1.2 Um SIGAD tem que garantir a criação de classes, subclasses, grupos e subgrupos nos níveis do plano de classificação de acordo com o método de codificação adotado. Por exemplo, quando se adotar o método decimal para codificação, cada classe pode ter no máximo dez subordinações, e assim sucessivamente.

RF 4.1.1 Um SIGAD tem que prover funcionalidades para definição e manutenção de tabela de temporalidade e destinação de documentos, associada ao plano de classificação do órgão ou entidade.

RF 4.1.2 Um SIGAD tem que associar, automaticamente, ao dossiê/processo o prazo e a destinação previstos na classe em que o documento foi inserido.

RF 4.1.3 Um SIGAD tem que manter tabela de temporalidade e destinação de documentos com as seguintes informações:

- identificador do órgão ou entidade;
- identificador da classe;
- prazo de guarda na fase corrente;
- prazo de guarda na fase intermediária;
- destinação final;
- observações;
- evento que determina o início da contagem do prazo de retenção na fase corrente e na fase intermediária.

A tabela de temporalidade e destinação de documentos dos integrantes do SINAR deve estar de acordo com a legislação e ser aprovada pela instituição arquivística na específica esfera de competência.

RF 4.1.4 Um SIGAD tem que prever, pelo menos, as seguintes situações para destinação:

- apresentação dos documentos para reavaliação em data futura;
- eliminação;
- exportação para transferência;
- exportação para recolhimento (guarda permanente).

RF 4.1.5 Um SIGAD tem que prever a iniciação automática da contagem dos prazos de guarda referenciados na tabela de temporalidade e destinação de documentos, pelo menos, a partir dos seguintes eventos:

- abertura de dossiê;
- arquivamento de dossiê/processo;
- desarquivamento de dossiê/processo;
- inclusão de documento em um dossiê/processo.

Acontecimentos específicos, descritos na tabela de temporalidade e destinação, como, por exemplo, "cinco anos a contar da data de aprovação das contas", quando não puderem ser detectados automaticamente pelo sistema, deverão ser informados ao SIGAD por usuário autorizado.

RF 4.1.6 Um SIGAD tem que prever que a definição dos prazos de guarda seja expressa por:

- um número inteiro de dias ou
- um número inteiro de meses ou
- um número inteiro de anos ou
- uma combinação de um número inteiro de anos, meses e dias.

RF 4.1.7 Um SIGAD tem que limitar a definição e a manutenção (alteração, inclusão e exclusão) da tabela de temporalidade e destinação de documentos a usuários autorizados.

RF 4.1.8 Um SIGAD tem que permitir que um usuário autorizado altere o prazo ou destinação prevista em um item da tabela de temporalidade e destinação de documentos e garantir que a alteração tenha efeito em todos os documentos ou dossiês/processos associados àquele item. As alterações na tabela de temporalidade e destinação só poderão ser feitas como resultado de um processo de reavaliação realizado pela comissão de avaliação do órgão ou entidade em virtude de mudança do contexto

administrativo, jurídico ou cultural. Os integrantes do SINAR deverão ainda ter suas tabelas aprovadas pela instituição arquivística na específica esfera de competência.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 59 Exportar documento

RF 4.3.1 Um SIGAD tem que ser capaz de exportar documentos e dossiês/processos digitais e seus metadados para outro sistema dentro ou fora do órgão ou entidade.

RF 4.3.10 Um SIGAD tem que gerar listagem em meio digital e em papel para descrever documentos e dossiês/processos digitais que estão sendo exportados. Este requisito se aplica principalmente nos casos em que é feita exportação para transferência ou recolhimento a uma instituição arquivística pública. Nesse caso, a listagem deverá ser produzida no formato estabelecido pela instituição arquivística recebedora.

RF 4.3.2 Quando um SIGAD exportar os documentos e dossiês/processos de uma classe para executar uma ação de transferência ou recolhimento, tem que ser capaz de exportar todos os documentos e dossiês/processos da classe incluídos na ação de destinação

RF 4.3.3 Um SIGAD tem que ser capaz de exportar um documento e dossiê/processo ou grupo de documentos e dossiês/processos numa sequência de operações, de modo que:

- o conteúdo, o contexto e a estrutura dos documentos não se degradem;
- todos os componentes de um documento digital sejam exportados como uma unidade.

Por exemplo, uma mensagem de correio eletrônico e seus respectivos anexos;

- todos os metadados do documento sejam relacionados a ele de forma que as ligações possam ser mantidas no novo sistema;
- todas as ligações entre documentos, volumes e dossiês/processos sejam mantidas.

RF 4.3.6 Um SIGAD tem que ser capaz de exportar todos os tipos de documentos que está apto a capturar.

RF 4.3.7 Um SIGAD tem que produzir um relatório detalhado sobre qualquer falha que ocorra durante uma exportação. O relatório tem que identificar os documentos e dossiês/processos que originaram erros de processamento ou cuja exportação não tenha sido bem sucedida.

RF 4.3.8 Um SIGAD tem que conservar todos os documentos e dossiês/processos digitais que foram exportados, pelo menos até que tenham sido importados no sistema destinatário com êxito.

RF 5.3.17 Um SIGAD tem que incluir recursos destinados a transferir para suportes adequados documentos que não possam ser impressos, tais como documentos sonoros, vídeos e páginas web.

RF 6.3.1 Um SIGAD tem que implementar a classificação de grau de sigilo de documentos, dossiês/processos e classes do plano de classificação, e de todas as operações de usuários nos documentos.

RF 6.3.3 Um SIGAD tem que recusar o acesso de usuários a documentos que possuam grau de sigilo superior à sua credencial de segurança.

RF 6.6.1 Um SIGAD tem que usar criptografia no armazenamento, na transmissão e na apresentação de documentos arquivísticos digitais ao implementar a política de sigilo.

RF 6.6.2 Um SIGAD tem que limitar o acesso aos documentos cifrados somente àqueles usuários portadores da chave de decifração.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 64 Capturar documento

RF 3.1.1 A captura tem que garantir a execução das seguintes funções:

- registrar e gerenciar todos os documentos convencionais;
- registrar e gerenciar todos os documentos digitais, independentemente
- do contexto tecnológico;
- classificar todos os documentos de acordo com o plano ou código de
- classificação;
- controlar e validar a introdução de metadados.

RF 3.1.10 Um SIGAD tem que prever a adoção da numeração única de processos e/ou documentos oficiais de acordo com a legislação específica a fim de garantir a integridade do número atribuído ao processo e/ou documento na unidade protocolizadora de origem.

RF 3.1.16 Um SIGAD tem que garantir a visualização do registro de entrada do documento no sistema com todos os metadados inseridos automaticamente e os demais a serem atribuídos pelo usuário. Por exemplo, o sistema pode atribuir, automaticamente, o número identificador, a data de captura, o título, o originador, e requerer que o usuário preencha os demais metadados.

RF 3.1.18 Sempre que um documento tiver mais de uma versão, o SIGAD tem que permitir que os usuários selecionem pelo menos uma das seguintes ações:

- registrar todas as versões do documento como um só documento arquivístico;
- registrar uma única versão do documento como um documento arquivístico;
- registrar cada uma das versões do documento, separadamente, como um documento arquivístico.

RF 3.1.2 Um SIGAD tem que ser capaz de capturar documentos digitais das formas a seguir:

- captura de documentos produzidos dentro do SIGAD;
- captura de documento individual produzido em arquivo digital fora do SIGAD;
- captura de documento individual produzido em workflow ou em outros sistemas integrados ao SIGAD;
- captura de documentos em lote.

RF 3.1.21 No caso de documentos ou dossiês/processos constituídos por mais de um objeto digital, o SIGAD tem que:

- tratar o documento como uma unidade indivisível, assegurando a relação entre os objetos digitais;
- preservar a integridade do documento, mantendo a relação entre os objetos digitais;
- garantir a integridade do documento quando de sua recuperação, visualização e gestão posteriores;
- gerenciar a destinação de todos os objetos digitais que compõem o documento como uma unidade indivisível.

RF 3.1.22 Um SIGAD tem que emitir um aviso caso o usuário tente registrar um documento que já tenha sido registrado no mesmo dossiê/processo.

RF 3.1.4 Um SIGAD tem que aceitar o conteúdo do documento, bem como as informações que definem sua aparência, mantendo as associações entre os vários objetos digitais que compõem o documento, isto é, anexos e links de hipertexto.

RF 3.1.5 Um SIGAD tem que permitir a inserção de todos os metadados obrigatórios e opcionais definidos na sua configuração e garantir que se mantenham associados ao documento. Os metadados obrigatórios são:

- nome do arquivo digital;
- número identificador atribuído pelo sistema;
- data de produção;
- data e hora de transmissão e recebimento;
- data e hora da captura;
- título ou descrição abreviada;
- classificação de acordo com o plano ou código de classificação;
- prazos de guarda;
- autor (pessoa física ou jurídica);
- redator (se diferente do autor);
- originador;
- destinatário (e respectivo cargo);
- nome do setor responsável pela execução da ação contida no documento;
- indicação de anotação;
- indicação de anexos;
- indicação de versão;
- restrição de acesso;

- registro das migrações e data em que ocorreram.

Os metadados opcionais se referem a informações mais detalhadas sobre o documento, tais como:

- espécie/tipo/gênero documental;
- associações a documentos diferentes que podem estar relacionados pelo fato de registrarem a mesma atividade ou se referirem à mesma pessoa ou situação;
- formato e software (nome e versão) em que o documento foi produzido ou capturado;
- máscaras de formatação (templates) necessárias para interpretar a estrutura do documento;
- assunto/descritor (diferentes do já estabelecido no código de classificação);
- localização física; e
- outros que se julgarem necessários.

RF 3.1.6 Um SIGAD tem que prever a inserção dos metadados obrigatórios, previstos em legislação específica na devida esfera e âmbito de competência, no momento da captura de processos.

RF 3.1.7 Um SIGAD tem que ser capaz de atribuir um número identificador a cada dossiê/processo e documento capturado, que serve para identificá-lo desde o momento da captura até sua destinação final no SIGAD.

RF 3.1.8 O formato do número identificador atribuído pelo sistema deve ser definido no momento da configuração do SIGAD. O identificador pode ser numérico ou alfanumérico, ou pode incluir os identificadores encadeados das entidades superiores no ramo apropriado da hierarquia.

RF 3.1.9 Num SIGAD, o número identificador atribuído pelo sistema tem que:

- ser gerado automaticamente, sendo vedada sua introdução manual e alteração posterior; ou
- ser atribuído pelo usuário e validado pelo sistema antes de ser aceito.

Uma opção seria gerar o número identificador automaticamente, mas, nesse caso, ocultando-o do usuário e permitindo a este introduzir uma sequência não necessariamente única como um “identificador”. O usuário empregaria essa sequência como um identificador, mas o SIGAD a consideraria um metadado pesquisável, definido pelo usuário.

RF 3.3.1 Um SIGAD tem que permitir que, na fase de configuração, seja escolhida uma das seguintes operações:

- capturar mensagens de correio eletrônico após selecionar quais serão objeto de registro; ou
- capturar, automaticamente, todas as mensagens de correio eletrônico.

RF 3.4.1 O SIGAD tem que poder capturar também os documentos convencionais e/ou híbridos.

RF 3.5.1 Um SIGAD tem que possuir a capacidade de capturar documentos com diferentes formatos de arquivo e estruturas.

RF 3.5.3 Um SIGAD tem que capturar documentos que se apresentam com as seguintes estruturas:

- simples: texto, imagens, mensagens de correio eletrônico, slides digitais, som.
- composta: mensagens de correio eletrônico com anexos, páginas web, publicações eletrônicas, bases de dados.

RF 3.5.4 Um SIGAD tem que ser capaz de incluir novos formatos de arquivos à medida que forem sendo adotados pelo órgão ou entidade.

RF 4.5.1 Um SIGAD tem que aplicar a mesma tabela de temporalidade e destinação de documentos para os documentos convencionais, digitais ou híbridos.

RF 6.3.4 Um SIGAD tem que garantir que documentos sem atribuição de grau de sigilo, importados a partir de fontes externas ao SIGAD, estejam sujeitos às políticas de controle de acesso e de sigilo.

RF 6.3.5 Um SIGAD tem que ser capaz de manter a marcação de sigilo original durante a importação de documentos a partir de fontes externas ao SIGAD.

RF 6.3.7 Um SIGAD tem que permitir que um dos itens abaixo seja selecionado durante a configuração:

- graus de sigilo a serem atribuídos a classes e dossiês/processos;
- classes e dossiês/processos sem grau de sigilo.

RF 6.3.9 Um SIGAD tem que garantir que o grau de sigilo de um documento importado esteja associado a um usuário autorizado com a credencial de segurança pertinente para receber o documento.

RF 6.5.3 Um SIGAD tem que ser capaz de verificar a validade da assinatura digital no momento da captura do documento.

RF 6.7.1 Um SIGAD tem que ser capaz de recuperar informação contida em marcas d'água digitais.

RF 6.7.2 Um SIGAD tem que ser capaz de armazenar documentos arquivísticos digitais que contenham marcas d'água digitais, assim como informação de apoio relacionada à marca d'água.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 65 Emitir relatório de desempenho

RF 2.1.19 Um recurso de fluxo de trabalho de um SIGAD tem que fornecer meios de elaboração de relatórios completos para permitir que gestores monitorem a tramitação dos documentos e o desempenho dos participantes.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 67 Gerenciar metadados

RF 1.6.3 Um SIGAD tem que permitir que um conjunto específico de metadados seja configurado para os documentos ou dossiês/processos convencionais e incluir informações sobre o local de arquivamento.

RF 1.6.9 Um SIGAD tem que poder registrar na trilha de auditoria todas as alterações efetuadas nos metadados dos documentos ou dossiês/processos convencionais e híbridos.

RF 3.1.12 Um SIGAD tem que garantir que os metadados associados a um documento sejam inseridos somente por usuários autorizados.

RF 3.1.13 Um SIGAD tem que garantir que os metadados associados a um documento sejam alterados somente por administradores e usuários autorizados e devidamente registrados em trilhas de auditoria.

RF 3.1.17 Um SIGAD tem que garantir a inserção de outros metadados após a captura. Por exemplo, data e hora de alteração e mudança de suporte.

RF 3.4.2 O SIGAD tem que acrescentar aos metadados dos documentos convencionais informações sobre sua localização. Essa informação só será acessada por usuários autorizados.

RF 6.10.4 Em caso de erro na inserção de metadados, o administrador terá que corrigi-lo, e o SIGAD tem que registrar essa ação na trilha de auditoria.

RF 6.6.3 Um SIGAD tem que registrar os seguintes metadados sobre um documento cifrado:

- indicação sobre se está cifrado ou não;
- algoritmos usados na cifração;
- identificação do remetente;
- identificação do destinatário.

RF 6.6.6 Em caso de remoção da cifração do documento, os seguintes metadados adicionais têm que ser registrados na trilha de auditoria:

- data e hora da remoção da cifração;
- identificação do executor da operação;
- motivo da remoção da cifração.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;

- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 68 Gerenciar Tramitação e fluxo de trabalho

RF 2.1.1 Um recurso de fluxo de trabalho de um SIGAD tem que fornecer os passos necessários para o cumprimento de trâmites preestabelecidos ou aleatórios.

RF 2.1.12 Um recurso de fluxo de trabalho de um SIGAD tem que fornecer um histórico de movimentação dos documentos. O histórico de movimentação corresponde a um conjunto de metadados de datas de entrada e saída, nomes de responsáveis, título do documento, providências etc.

RF 2.1.14 Um recurso de fluxo de trabalho de um SIGAD tem que incluir processamento condicional, isto é, permitir que um fluxo de trabalho seja suspenso para aguardar a chegada de um documento e prossiga automaticamente quando este é recebido.

RF 2.1.16 Um recurso de fluxo de trabalho de um SIGAD tem que reconhecer indivíduos e grupos de trabalho como participantes.

RF 2.1.2 Um SIGAD tem que ter capacidade, sem limitações, de estabelecer o número necessário de trâmites nos fluxos de trabalho.

RF 2.1.20 Um recurso de fluxo de trabalho de um SIGAD tem que registrar a tramitação de um documento em seus metadados. Os metadados referentes à tramitação devem registrar data e hora de envio e recebimento, e a identificação do usuário.

RF 2.1.3 O fluxo de trabalho de um SIGAD tem que disponibilizar uma função para avisar um participante do fluxo de que um documento lhe foi enviado, especificando a ação necessária.

RF 2.1.5 O recurso de fluxo de trabalho de um SIGAD tem que permitir que fluxos de trabalho pré-programados sejam definidos, alterados e mantidos exclusivamente por usuário autorizado.

RF 2.1.7 Um recurso de fluxo de trabalho de um SIGAD tem que registrar na trilha de auditoria todas as alterações ocorridas neste fluxo.

RF 2.1.8 Um recurso de fluxo de trabalho de um SIGAD tem que registrar a tramitação de um documento a fim de que os usuários possam conhecer a situação de cada um no processo.

RF 2.2.1 Um recurso de fluxo de trabalho de um SIGAD tem que ser capaz de registrar o status de transmissão do documento, ou seja, se é minuta, original ou cópia.

RF 2.2.2 Um SIGAD tem que ser capaz de controlar as diversas versões de um documento que está tramitando.

RF 2.2.3 Um SIGAD tem que ser capaz de associar e relacionar as diversas versões de um documento.

RF 2.2.4 Um SIGAD tem que manter o identificador único do documento, e o controle de versões tem que ser registrado em metadados específicos.

RF 6.3.1 Um SIGAD tem que implementar a classificação de grau de sigilo de documentos, dossiês/processos e classes do plano de classificação, e de todas as operações de usuários nos documentos.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 69 Gerenciar documentos e processos/dossiês

RF 1.3.1 Um SIGAD tem que registrar nos metadados as datas de abertura e de encerramento do dossiê/processo. Essa data pode servir de parâmetro para aplicação dos prazos de guarda e destinação do dossiê/processo.

RF 1.3.2 Um SIGAD tem que permitir que um dossiê/processo seja encerrado por meio de procedimentos regulamentares e somente por usuários autorizados.

RF 1.3.3 Um SIGAD tem que permitir a consulta aos dossiês/processos já encerrados por usuários autorizados.

RF 1.3.4 Um SIGAD tem que impedir o acréscimo de novos documentos a dossiês/processos já encerrados. Dossiês/processos encerrados devem ser reabertos para receber novos documentos.

RF 1.3.6 Um SIGAD tem que impedir sempre a eliminação de uma unidade de arquivamento digital ou de qualquer parte de seu conteúdo, a não ser quando estiver de acordo com a tabela de temporalidade e destinação de documentos. A eliminação será devidamente registrada em trilha de auditoria.

RF 1.3.7 Um SIGAD tem que garantir sempre a integridade da relação hierárquica entre classe, dossiê/processo, volume e documento, e entre classe, pasta e documento, independentemente de atividades de manutenção, ações do usuário ou falha de componentes. Em hipótese alguma pode o SIGAD permitir que uma ação do usuário ou falha do sistema dê origem a inconsistência em sua base de dados.

RF 1.4.1 Um SIGAD tem que prever a formação/autuação de processos por usuário autorizado conforme estabelecido em legislação específica.

RF 1.4.10 Um SIGAD tem que prever o desarquivamento para reativação dos processos, por usuário autorizado e obedecendo a procedimentos legais e administrativos. Para manter a integridade do processo, somente o último volume receberá novos documentos ou peças.

RF 1.4.3 Um SIGAD tem que prever que os documentos integrantes do processo digital recebam numeração sequencial sem falhas, não se admitindo que documentos diferentes recebam a mesma numeração.

RF 1.4.4 Um SIGAD tem que controlar a renumeração dos documentos integrantes de um processo digital. Este requisito tem por objetivo impedir a exclusão não autorizada de documentos de um processo. Casos especiais que autorizem a renumeração devem obedecer à legislação específica na devida esfera e âmbito de competência.

RF 1.4.5 Um SIGAD tem que prever procedimentos para juntada de processos segundo a legislação específica na devida esfera e âmbito de competência. A juntada pode ser por anexação ou apensação. Este procedimento deve ser registrado nos metadados do processo.

RF 1.4.6 Um SIGAD tem que prever procedimentos para desapensação de processos segundo a legislação específica na devida esfera e âmbito de competência. Esse procedimento deve ser registrado nos metadados do processo.

RF 1.4.7 Um SIGAD tem que prever procedimentos para desentranhamento de documentos integrantes de um processo, segundo norma específica na devida esfera e âmbito de competência. Esse procedimento deve ser registrado nos metadados do processo.

RF 1.4.8 Um SIGAD tem que prever procedimentos para desmembramento de documentos integrantes de um processo, segundo norma específica na devida esfera e âmbito de competência. Esse procedimento deve ser registrado nos metadados do processo.

RF 1.4.9 Um SIGAD tem que prever o encerramento dos processos incluídos seus volumes e metadados.

RF 1.5.3 Um SIGAD tem que permitir que um volume herde, automaticamente, do dossiê/processo ao qual pertence, alguns metadados predefinidos, como, por exemplo, procedência, classes e temporalidade.

RF 1.5.6 Um SIGAD tem que assegurar que um volume conterá somente documentos. Não é permitido que um volume contenha outro volume ou outro dossiê/processo.

RF 1.5.7 Um SIGAD tem que permitir que um volume seja encerrado por meio de procedimentos regulamentares e apenas por usuários autorizados.

RF 1.5.8 Um SIGAD tem que assegurar que, ao ser aberto um novo volume, o precedente seja automaticamente encerrado. Apenas o volume produzido mais recentemente pode estar aberto; os demais volumes existentes no dossiê/processo têm que estar fechados.

RF 1.5.9 Um SIGAD tem que impedir a reabertura, para acréscimo de documentos, de um volume já encerrado.

RF 1.6.1 Um SIGAD tem que capturar documentos ou dossiês/processos convencionais e gerenciá-los da mesma forma que os digitais. Para o conceito de captura.

RF 1.6.2 Um SIGAD tem que ser capaz de gerenciar a parte convencional e a parte digital integrantes de dossiês/processos híbridos associando-as com o mesmo número identificador atribuído pelo sistema e o mesmo título, além de indicar que se trata de um documento arquivístico híbrido.

RF 1.6.4 Um SIGAD tem que dispor de mecanismos para acompanhar a movimentação do documento arquivístico convencional, de forma que fique evidente para o usuário a localização atual do documento.

RF 1.6.5 Um SIGAD tem que ser capaz de oferecer ao usuário funcionalidades para solicitar ou reservar a consulta a um documento arquivístico convencional, enviando uma mensagem para o detentor atual do documento ou para o administrador.

RF 1.6.7 Um SIGAD tem que assegurar que a recuperação de um documento ou dossiê/processo híbrido permita, igualmente, a recuperação dos metadados da parte digital e da convencional.

RF 1.6.8 Sempre que os documentos ou dossiês/processos híbridos estiverem classificados quanto ao grau de sigilo, um SIGAD tem que garantir que a parte convencional e a parte digital correspondente recebam a mesma classificação de sigilo.

RF 3.6.1 Um SIGAD tem que ser capaz de reconhecer três domínios para o controle dos procedimentos de gestão: espaço individual, espaço do grupo e espaço geral.

RF 3.6.2 Um SIGAD tem que ser capaz de operacionalizar as regras estabelecidas pelo sistema de gestão arquivística de documentos nos três espaços.

RF 3.6.3 Um SIGAD tem que impedir que o conteúdo de um documento seja alterado por usuários e administradores, exceto se a alteração fizer parte do processo documental.

RF 4.2.1 Um SIGAD tem que fornecer recursos integrados à tabela de temporalidade e destinação de documentos para implementar as ações de destinação.

RF 4.2.2 Para cada dossiê/processo, um SIGAD tem que acompanhar automaticamente os prazos de guarda determinados para a classe à qual pertence.

RF 4.2.3 Um SIGAD tem que prover funcionalidades para informar ao usuário autorizado sobre os documentos ou dossiês/processos que já cumpriram ou estão para cumprir o prazo de guarda previsto.

RF 4.2.4 Um SIGAD tem de prover funcionalidades para gerenciar o processo de destinação, que tem de ser iniciado por usuário autorizado e cumprir os seguintes passos:

- identificar automaticamente os documentos ou dossiês/processos que atingiram os prazos de guarda previstos;
- informar o usuário autorizado sobre todos os documentos ou dossiês/processos que foram identificados no passo anterior;
- possibilitar a alteração do prazo ou destinação previstos para aqueles documentos ou dossiês/processos, caso necessário;
- proceder à ação de destinação quando confirmada pelo usuário autorizado.

RF 4.2.5 Um SIGAD tem sempre que pedir confirmação antes de realizar as ações de destinação.

RF 4.2.7 Um SIGAD tem que restringir as funções de destinação a usuários autorizados.

RF 4.2.8 Quando um administrador transfere documentos ou dossiês/processos de uma classe para outra, em virtude de uma reclassificação, o SIGAD tem que adotar automaticamente a temporalidade e a destinação vigente na nova classe.

RF 4.2.9 Quando um documento digital (objeto digital) estiver associado a mais de um dossiê ou processo, e tiver prazos de guarda diferentes associados a ele, o SIGAD tem que automaticamente verificar todos os prazos de guarda e as destinações previstas para esse documento e garantir que ele seja mantido em cada dossiê/processo pelo tempo definido na tabela de temporalidade e destinação de documentos, de forma que:

- a remoção de um documento de um dossiê/processo não prejudique a manutenção desse mesmo documento em outro dossiê/processo, até que todas as referências desse documento tenham atingido o prazo de guarda previsto;
- a manutenção de um documento em um dossiê/processo por prazo mais longo não obrigue a permanência desse mesmo documento em outro dossiê/processo de prazo mais curto. Nesse caso o registro do documento com prazo mais curto tem que ser removido, mas o documento é mantido no SIGAD. Quando um documento digital estiver associado a mais de um dossiê ou processo, o SIGAD deverá criar um registro para cada referência desse documento. Cada registro estará vinculado ao mesmo objeto digital. No momento da eliminação, o objeto digital não poderá ser eliminado sem que antes se verifique a temporalidade de todas as referências associadas a ele. O objeto digital só poderá ser eliminado quando os prazos de guarda de todas as referências tiverem sido cumpridos. Antes disso, só se pode fazer a eliminação de cada registro individualmente.

RF 4.4.1 Um SIGAD tem que restringir a função de eliminação de documentos ou dossiês/processos somente a usuários autorizados.

RF 4.4.2 Um SIGAD tem que pedir confirmação da eliminação a um usuário autorizado antes que qualquer ação seja tomada com relação ao documento e dossiê/processo e cancelar o processo de eliminação se a confirmação não for dada.

RF 4.4.3 Um SIGAD tem que avisar o usuário autorizado quando um documento ou dossiê/processo que estiver sendo eliminado se encontrar relacionado a outro; os sistemas também têm de suspender o processo até que seja tomada uma das medidas abaixo:

- confirmação pelo usuário autorizado para prosseguir ou cancelar o processo;

- produção de um relatório especificando os documentos ou dossiês/processos envolvidos e todas as ligações com outros documentos ou dossiês/processos.

RF 4.4.5 Quando um documento tem várias referências armazenadas no sistema, um SIGAD tem que garantir que todas essas referências sejam verificadas antes de eliminar o objeto digital.

RF 4.4.6 Um SIGAD tem que produzir um relatório detalhando qualquer falha que ocorra durante uma eliminação. O relatório tem que identificar os documentos cuja eliminação não tenha sido bem sucedida.

RF 4.4.8 Um SIGAD tem que gerar relatório com os documentos e dossiês/processos que serão eliminados. Essa listagem deve seguir o formato da listagem de eliminação conforme o estabelecido na norma vigente.

RF 4.4.9 Um SIGAD tem que manter metadados relativos a documentos e dossiês/processos eliminados. O administrador deve indicar o subconjunto de metadados que deverá ser mantido.

RF 4.5.2 Um SIGAD tem que acompanhar os prazos de guarda dos documentos convencionais e deve dar início aos procedimentos de eliminação ou transferência desses documentos, tomando em consideração suas especificidades.

RF 4.5.3 Um SIGAD tem que alertar o administrador sobre a existência e a localização de uma parte convencional associada a um documento híbrido que esteja destinado a ser exportado, transferido ou eliminado.

RF 6.10.10 Um SIGAD tem que armazenar, na trilha de auditoria, qualquer alteração efetuada para satisfazer os requisitos desta seção.

RF 6.10.3 Em situações excepcionais, o administrador tem que ser autorizado a apagar ou corrigir dossiês/processos, volumes e documentos. Nesse caso, um SIGAD tem que:

- registrar integralmente a ação de apagar ou corrigir na trilha de auditoria;
- produzir um relatório de anomalias para o administrador;
- eliminar todo o conteúdo de um dossiê/processo ou volume, quando forem eliminados;
- garantir que nenhum documento seja eliminado se tal ação resultar na alteração de outro documento arquivístico;
- informar o administrador sobre a existência de ligação entre um dossiê/processo ou documento prestes a ser apagado e qualquer outro dossiê/processo ou documento, solicitando confirmação antes de concluir a operação;

- manter a integridade total do metadado, a qualquer momento.

RF 6.2.12 O acesso a documentos, a dossiês/processos ou classes tem que ser concedido se a permissão requerida para a operação estiver associada a pelo menos um dos grupos aos quais pertença o usuário.

RF 6.2.15 Um SIGAD tem que usar os seguintes atributos do usuário ao implementar a política de controle de acesso aos documentos digitais por papéis de usuários:

- identificação do usuário;
- papéis associados ao usuário.

RF 6.2.16 Um SIGAD tem que usar os seguintes atributos dos documentos digitais ao implementar a política de controle de acesso por papéis:

- identificação do documento digital;
- operações permitidas aos vários papéis de usuários, sobre as classes ou unidades de arquivamento a que o documento pertence.

RF 6.2.17 O acesso a documentos, dossiês/processos ou classes tem que ser concedido somente se a permissão requerida para a operação estiver presente em pelo menos um dos papéis associados ao usuário.

RF 6.3.1 Um SIGAD tem que implementar a classificação de grau de sigilo de documentos, dossiês/processos e classes do plano de classificação, e de todas as operações de usuários nos documentos.

RF 6.3.12 Um SIGAD tem que permitir somente ao usuário autorizado, mediante confirmação, a desclassificação ou redução do grau de sigilo de um documento.

RF 6.3.14 Um SIGAD tem que impedir que um documento sigiloso seja eliminado. Os documentos sigilosos têm que se tornar ostensivos para serem submetidos ao processo de avaliação e receberem a destinação prevista.

RF 6.3.15 Um SIGAD tem que implementar metadados nos níveis de dossiê, documento ou extrato de documento para controlar o acesso à informação sensível.

RF 6.3.3 Um SIGAD tem que recusar o acesso de usuários a documentos que possuam grau de sigilo superior à sua credencial de segurança.

RF 6.3.8 Em caso de erro ou reavaliação, o administrador tem que ser capaz de alterar o grau de sigilo de todos os documentos arquivísticos de um dossiê/processo ou de uma classe, numa única operação.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 70 Gerenciar dossiê/processo

RF 1.3.1 Um SIGAD tem que registrar nos metadados as datas de abertura e de encerramento do dossiê/processo. Essa data pode servir de parâmetro para aplicação dos prazos de guarda e destinação do dossiê/processo.

RF 1.3.2 Um SIGAD tem que permitir que um dossiê/processo seja encerrado por meio de procedimentos regulamentares e somente por usuários autorizados.

RF 1.3.3 Um SIGAD tem que permitir a consulta aos dossiês/processos já encerrados por usuários autorizados.

RF 1.3.4 Um SIGAD tem que impedir o acréscimo de novos documentos a dossiês/processos já encerrados. Dossiês/processos encerrados devem ser reabertos para receber novos documentos.

RF 1.3.6 Um SIGAD tem que impedir sempre a eliminação de uma unidade de arquivamento digital ou de qualquer parte de seu conteúdo, a não ser quando estiver de acordo com a tabela de temporalidade e destinação de documentos. A eliminação será devidamente registrada em trilha de auditoria.

RF 1.3.7 Um SIGAD tem que garantir sempre a integridade da relação hierárquica entre classe, dossiê/processo, volume e documento, e entre classe, pasta e documento, independentemente de atividades de manutenção, ações do usuário ou falha de componentes. Em hipótese alguma pode o SIGAD permitir que uma ação do usuário ou falha do sistema dê origem a inconsistência em sua base de dados.

RF 1.4.1 Um SIGAD tem que prever a formação/autuação de processos por usuário autorizado conforme estabelecido em legislação específica.

RF 1.4.10 Um SIGAD tem que prever o desarquivamento para reativação dos processos, por usuário autorizado e obedecendo a procedimentos legais e administrativos. Para manter a integridade do processo, somente o último volume receberá novos documentos ou peças.

RF 1.4.3 Um SIGAD tem que prever que os documentos integrantes do processo digital recebam numeração sequencial sem falhas, não se admitindo que documentos diferentes recebam a mesma numeração.

RF 1.4.4 Um SIGAD tem que controlar a renumeração dos documentos integrantes de um processo digital. Este requisito tem por objetivo impedir a exclusão não autorizada de documentos de um processo. Casos especiais que autorizem a renumeração devem obedecer à legislação específica na devida esfera e âmbito de competência.

RF 1.4.5 Um SIGAD tem que prever procedimentos para juntada de processos segundo a legislação específica na devida esfera e âmbito de competência. A juntada pode ser por anexação ou apensação. Este procedimento deve ser registrado nos metadados do processo.

RF 1.4.6 Um SIGAD tem que prever procedimentos para desapensação de processos segundo a legislação específica na devida esfera e âmbito de competência. Esse procedimento deve ser registrado nos metadados do processo.

RF 1.4.7 Um SIGAD tem que prever procedimentos para desentranhamento de documentos integrantes de um processo, segundo norma específica na devida esfera e âmbito de competência. Esse procedimento deve ser registrado nos metadados do processo.

RF 1.4.8 Um SIGAD tem que prever procedimentos para desmembramento de documentos integrantes de um processo, segundo norma específica na devida esfera e âmbito de competência. Esse procedimento deve ser registrado nos metadados do processo.

RF 1.4.9 Um SIGAD tem que prever o encerramento dos processos incluídos seus volumes e metadados.

RF 1.5.3 Um SIGAD tem que permitir que um volume herde, automaticamente, do dossiê/processo ao qual pertence, alguns metadados predefinidos, como, por exemplo, procedência, classes e temporalidade.

RF 1.5.6 Um SIGAD tem que assegurar que um volume conterá somente documentos. Não é permitido que um volume contenha outro volume ou outro dossiê/processo.

RF 1.5.7 Um SIGAD tem que permitir que um volume seja encerrado por meio de procedimentos regulamentares e apenas por usuários autorizados.

RF 1.5.8 Um SIGAD tem que assegurar que, ao ser aberto um novo volume, o precedente seja automaticamente encerrado. Apenas o volume produzido mais recentemente pode estar aberto; os demais volumes existentes no dossiê/processo têm que estar fechados.

RF 1.5.9 Um SIGAD tem que impedir a reabertura, para acréscimo de documentos, de um volume já encerrado.

Caso de uso: CSU 80 Imprimir documento

RF 5.3.11 Um SIGAD tem que proporcionar ao usuário formas flexíveis de impressão de documentos com seus metadados e possibilitar a definição dos metadados a serem impressos.

Caso de uso: CSU 81 Gerenciar parâmetros e atributos de usuário

RF 6.3.11 Somente administradores autorizados têm que ser capazes de realizar as seguintes ações:

- remover ou revogar os atributos de segurança dos documentos;
- criar, alterar, remover ou revogar as credenciais de segurança dos usuários.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

RF 9.1.1 Um SIGAD tem que permitir que os administradores, de maneira controlada e sem esforço excessivo, recuperem, visualizem e reconfigurem os parâmetros do sistema e os atributos dos usuários.

Caso de uso: CSU 82 Restaurar sistema

RF 6.9.3 Após falha ou descontinuidade do sistema, quando a recuperação automática não for possível, um SIGAD tem que ser capaz de entrar em modo de manutenção, no qual é oferecida a possibilidade de restaurar o sistema para um estado seguro. Na restauração ao estado seguro, um SIGAD deve ser capaz de garantir a recuperação de perdas ocorridas, inclusive dos documentos de transações mais recentes.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 85 Gerenciar unidades de arquivamento

RF 1.1.3 Um SIGAD tem que permitir a usuários autorizados acrescentar novas classes sempre que necessário.

RF 1.2.1 Um SIGAD tem que permitir a classificação das unidades de arquivamento somente nas classes autorizadas.

RF 1.2.2 Um SIGAD tem que permitir a classificação de um número ilimitado de unidades de arquivamento dentro de uma classe.

RF 1.2.3 Um SIGAD tem que utilizar o termo completo da classe para identificar uma unidade de arquivamento.

RF 1.2.4 Um SIGAD tem que permitir a associação de metadados às unidades de arquivamento e deve restringir a inclusão e alteração desses metadados a usuários autorizados.

RF 1.2.5 Um SIGAD tem que associar os metadados das unidades de arquivamento conforme estabelecido no padrão de metadados.

RF 1.2.6 Um SIGAD tem que permitir que uma nova unidade de arquivamento herde, da classe em que foi classificada, alguns metadados predefinidos. Exemplos desta herança são prazos de guarda previstos na tabela de temporalidade e destinação e restrição de acesso.

RF 1.2.9 Um SIGAD tem que permitir que uma unidade de arquivamento e seus respectivos volumes e/ou documentos sejam reclassificados por um usuário autorizado e que todos os documentos já inseridos permaneçam nas unidades de arquivamento e nos volumes que estão sendo transferidos, mantendo a relação entre documentos, volumes e unidades de arquivamento.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 86 Gerenciar auditorias

RF 6.4.14 Somente administradores autorizados têm que ser capazes de configurar o conjunto de eventos auditáveis e seus atributos.

RF 6.4.15 Somente administradores autorizados, acompanhados do auditor, têm que ser capazes de configurar o conjunto de eventos auditáveis e seus atributos.

RF 6.4.6 Um SIGAD tem que ser capaz de impedir qualquer modificação na trilha de auditoria.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 87 Gerar relatório de auditoria

RF 6.4.12 Um SIGAD tem que fornecer relatórios sobre as ações que afetam classes, unidades de arquivamento e documentos, em ordem cronológica e organizados por:

- documento arquivístico, unidade de arquivamento ou classe;
- usuário;
- tipo de ação ou operação.

RF 6.4.15 Somente administradores autorizados, acompanhados do auditor, têm que ser capazes de configurar o conjunto de eventos auditáveis e seus atributos.

RF 6.4.4 Um SIGAD tem que assegurar que as informações da trilha de auditoria estejam disponíveis para inspeção, a fim de que uma ocorrência específica possa ser identificada e todas as informações correspondentes sejam claras e compreensíveis.

RF 6.4.7 Somente administradores autorizados têm que ser capazes de exportar as trilhas de auditoria sem afetar a trilha armazenada, ou transferir as trilhas de auditoria de um suporte de armazenamento para outro. A trilha de auditoria não pode ser excluída antes da data indicada na tabela de temporalidade. Porém, a transferência implica a cópia da trilha para outro espaço de armazenamento, com a subsequente liberação do espaço original. A exportação é a cópia sem liberação do espaço.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 88 Gerenciar assinaturas digitais

RF 6.5.2 Somente administradores autorizados têm que ser capazes de incluir, remover ou atualizar no SIGAD os certificados digitais de computadores ou de usuários.

RF 6.5.4 Um SIGAD, no processo de verificação da assinatura digital, tem que ser capaz de registrar, nos metadados do documento, o seguinte:

- validade da assinatura verificada;
- registro da verificação da assinatura;
- data e hora em que ocorreu a verificação.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 89 Gerenciar parâmetros de criptografia

RF 6.6.5 Somente usuários autorizados têm que ser capazes de realizar as operações a seguir:

- incluir, remover ou alterar parâmetros dos algoritmos criptográficos instalados no SIGAD;
- incluir, remover ou substituir chaves criptográficas de programas ou usuários do SIGAD;
- cifrar e alterar a criptografia de documentos;
- remover a criptografia de um documento.

A remoção da cifração pode ocorrer quando sua manutenção resultar na indisponibilidade do documento. Por exemplo, se a chave de cifração/decifração estiver embarcada em hardware inviolável cuja vida útil esteja prestes a se esgotar ou se o documento for desclassificado.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 90 Registrar transferência

RF 6.3.1 Um SIGAD tem que implementar a classificação de grau de sigilo de documentos, dossiês/processos e classes do plano de classificação, e de todas as operações de usuários nos documentos.

RF 6.3.3 Um SIGAD tem que recusar o acesso de usuários a documentos que possuam grau de sigilo superior à sua credencial de segurança.

RF 6.8.3 A função de acompanhamento de transferência tem que registrar metadados que incluam:

- número identificador dos documentos atribuído pelo sistema;
- localização atual e localizações anteriores, definidas pelo usuário;
- data e hora de envio/transferência;
- data e hora da recepção no novo local;
- destinatário;
- usuário responsável pela transferência (sempre que for adequado);
- método de transferência

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;

- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 91 Imprimir lista de documentos

RF 5.3.13 Um SIGAD tem que permitir a impressão de uma lista dos documentos e dossiês/processos resultantes de uma pesquisa.

RF 5.3.14 Um SIGAD tem que permitir a impressão de uma lista dos documentos que compõem um dossiê/processo.

RF 5.3.16 Um SIGAD tem que permitir que todos os documentos de um dossiê/processo sejam impressos em uma única operação, na sequência determinada pelo usuário.

RF 6.3.1 Um SIGAD tem que implementar a classificação de grau de sigilo de documentos, dossiês/processos e classes do plano de classificação, e de todas as operações de usuários nos documentos.

RF 6.3.3 Um SIGAD tem que recusar o acesso de usuários a documentos que possuam grau de sigilo superior à sua credencial de segurança.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 92 Exibir detalhes do documento capturado

RF 5.3.10 Um SIGAD tem que ser capaz de exibir/reproduzir o conteúdo de documentos que incluam imagem fixa, imagem em movimento e som.

RF 5.3.11 Um SIGAD tem que proporcionar ao usuário formas flexíveis de impressão de documentos com seus metadados e possibilitar a definição dos metadados a serem impressos.

RF 5.3.19 Um SIGAD tem que ser capaz de realizar pesquisa e exibição de documentos e dossiês/processos, simultaneamente, para diversos usuários.

RF 5.3.8 Um SIGAD tem que ser capaz de exibir em tela todos os tipos de documentos capturados.

RF 5.3.9 Um SIGAD tem que ser capaz de imprimir os documentos capturados, preservando o formato produzido pelas aplicações geradoras.

RF 6.3.1 Um SIGAD tem que implementar a classificação de grau de sigilo de documentos, dossiês/processos e classes do plano de classificação, e de todas as operações de usuários nos documentos.

RF 6.3.3 Um SIGAD tem que recusar o acesso de usuários a documentos que possuam grau de sigilo superior à sua credencial de segurança.

RF 6.6.2 Um SIGAD tem que limitar o acesso aos documentos cifrados somente àqueles usuários portadores da chave de decifração.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 93 Pesquisar documento

RF 5.1.1 Um SIGAD tem que fornecer facilidades para pesquisa, localização e apresentação dos documentos.

RF 5.2.1 Um SIGAD tem que fornecer uma série flexível de funções que atuem sobre os metadados relacionados com os diversos níveis de agregação (documento, unidade de arquivamento e classe) e sobre os conteúdos dos documentos arquivísticos por meio de parâmetros definidos pelo usuário, com o objetivo de localizar e acessar os documentos e/ou metadados, seja individualmente ou reunidos em grupo.

RF 5.2.17 Um SIGAD tem que permitir a pesquisa e recuperação de uma unidade de arquivamento completa e exibir a lista de todos os documentos que a compõem, como uma unidade e num único processo de recuperação.

RF 5.2.18 Um SIGAD tem que limitar o acesso a qualquer informação (metadado ou conteúdo de um documento arquivístico) se restrições de acesso e questões de segurança assim determinarem.

RF 5.2.2 Um SIGAD tem que executar pesquisa de forma integrada, isto é, apresentar todos os documentos e dossiês/processos, sejam eles digitais, híbridos ou convencionais, que satisfaçam aos parâmetros da pesquisa.

RF 5.2.3 Um SIGAD tem que permitir que todos os metadados de gestão de um documento ou dossiê/processo possam ser pesquisados.

RF 5.2.5 Um SIGAD tem que permitir que um documento ou dossiê/processo possa ser recuperado por meio de um número identificador.

RF 5.2.6 Um SIGAD tem que permitir que um documento ou dossiê/processo possa ser recuperado por meio de todas as formas de identificação

- implementadas, incluindo, no mínimo:
- identificador;
- título;
- assunto;
- datas;
- procedência/interessado;
- autor/redator/originador;
- classificação de acordo com plano ou código de classificação.

RF 5.3.1 Um SIGAD tem que apresentar o resultado da pesquisa como uma lista de documentos e dossiês/processos digitais, convencionais ou híbridos que cumpram os parâmetros da consulta e deve notificar o usuário se o resultado for nulo.

RF 5.3.19 Um SIGAD tem que ser capaz de realizar pesquisa e exibição de documentos e dossiês/processos, simultaneamente, para diversos usuários.

RF 5.3.3 Após apresentar o resultado da pesquisa, um SIGAD tem que oferecer ao usuário as opções:

- visualizar os documentos e dossiês/processos resultantes da pesquisa;
- redefinir os parâmetros de pesquisa e fazer nova consulta.

RF 5.3.7 Um SIGAD tem que ser capaz de apresentar o conteúdo de todos os tipos de documentos arquivísticos digitais capturados, de forma que:

- preserve as características de exibição visual e de formato apresentadas pela aplicação geradora;

- exiba todos os componentes do documento digital em conjunto, como uma unidade.

RF 6.3.1 Um SIGAD tem que implementar a classificação de grau de sigilo de documentos, dossiês/processos e classes do plano de classificação, e de todas as operações de usuários nos documentos.

RF 6.3.3 Um SIGAD tem que recusar o acesso de usuários a documentos que possuam grau de sigilo superior à sua credencial de segurança.

RF 6.7.1 Um SIGAD tem que ser capaz de recuperar informação contida em marcas d'água digitais.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 94 Gerar extrato de documento

RF 6.10.5 Um SIGAD tem que permitir a um usuário autorizado fazer um extrato (cópia truncada) de um documento, com o objetivo de não alterar o original.

RF 6.10.7 Quando uma cópia truncada é produzida, um SIGAD tem que registrar essa ação nos metadados do documento, incluindo, pelo menos, data, hora, motivo e quem a produziu.

RF 6.3.1 Um SIGAD tem que implementar a classificação de grau de sigilo de documentos, dossiês/processos e classes do plano de classificação, e de todas as operações de usuários nos documentos.

RF 6.3.15 Um SIGAD tem que implementar metadados nos níveis de dossiê, documento ou extrato de documento para controlar o acesso à informação sensível.

RF 6.3.3 Um SIGAD tem que recusar o acesso de usuários a documentos que possuam grau de sigilo superior à sua credencial de segurança.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 95 Anular uma operação

RF 6.10.1 Um SIGAD tem que permitir, a um administrador autorizado, anular a operação em caso de erro do usuário ou do sistema. Anular uma operação não significa apagar um documento arquivístico capturado pelo SIGAD. A anulação da eliminação definitiva de documentos, por ser irreversível, não é possível.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

Caso de uso: CSU 96 Restaurar base de dados

RF 6.1.7 Um SIGAD tem que incluir funções para restituir os documentos de arquivo e metadados a um estado conhecido, utilizando uma combinação de cópias restauradas e rotinas de auditoria.

RF 6.9.5 Um SIGAD tem que garantir que as funções de controle de acesso sejam invocadas antes de qualquer operação de acesso e retornem sem erros antes do prosseguimento da operação.

RF 6.9.6 Um SIGAD tem que preservar um estado seguro de funcionamento, interrompendo completamente a interação com usuários comuns, quando ocorrer um dos erros a seguir:

- falha de comunicação entre cliente e servidor;
- perda de integridade das informações de controle de acesso;
- falta de espaço para registro nas trilhas de auditoria.

RF 7.3.2 Um SIGAD tem que utilizar técnicas de restauração de dados em caso de falhas.