

Abordagem Metodológica para Compreensão e Aplicação de Padrões de Projeto

Simone Nasser Matos^{1,2}, Clovis Torres Fernandes²

¹Coodenação de Informática - Universidade Tecnológica Federal do Paraná (UTFPR)
Campus Ponta Grossa – Ponta Grossa – PR – Brasil

²Departamento de Ciência da Computação – Instituto Tecnológico de Aeronáutica
(ITA) São José dos Campos – SP – Brasil

snasser@utfpr.edu.br, clovistf@uol.com.br

Abstract. *Design Patterns to solve problems that can be used in different application domain, during the project phase. The learning these patterns by students has no knowledge about they time consumer for its understanding. This work presents an approach can to be used during the learning about patterns, providing the students a better understanding in this issue. The application of approach has been illustrated for learning of fifteen design patterns.*

Resumo. *Os padrões de projeto resolvem problemas que podem ser usados em aplicações de domínio diferentes, durante a fase de projeto. A aprendizagem desses padrões por alunos que não os conhecem demanda certo tempo para compreensão. Este trabalho apresenta uma abordagem que pode ser usada no aprendizado dos padrões de projeto, proporcionando aos alunos uma melhor compreensão nesse assunto. A aplicação da abordagem proposta será ilustrada durante a aprendizagem de quinze padrões de projeto.*

1. Introdução

Os padrões de projeto (*Design Patterns*) representam soluções para problemas recorrentes no desenvolvimento de aplicações de software, com o objetivo de melhorar o desempenho, otimizando os esforços de produção (Ambler 1998).

Atualmente não existe um consenso sobre como catalogar, compreender e aprender um padrão: diferentes autores preferem diferentes estilos. Alguns optam pela expressividade em função da estruturação, enquanto outros preferem o sentido oposto.

Catalogar um padrão é uma maneira de orientar o reúso de soluções e compartilhar informações sobre a melhor forma de se resolver um problema relacionado ao desenvolvimento ou ao projeto de um *software* (Alexander 1977).

Para tais finalidades, a catalogação de um padrão de projeto deve ser feita como um processo bem definido de maneira a prover um vocabulário comum, reduzindo a complexidade do sistema e o uso de estruturas simplificadas independentes. Desta forma, projetos mais complexos podem ser construídos, como por exemplo, um *framework* (Gamma et al 1994; Lajoile, Richard and Keller 1994). Por meio dessa catalogação se pode alcançar uma melhor compreensão e aprendizagem dos padrões de projeto.

Este trabalho visa apresentar uma abordagem metodológica para o ensino de padrões de projeto, sendo composta por duas fases. A primeira é utilizada para propiciar os conhecimentos introdutórios sobre o tema e fornece como artefato de saída o modelo para catalogação do padrão. Neste trabalho, esse modelo baseou-se na proposta estabelecida por Gamma et al. (1994). A segunda fase é composta por um conjunto de atividades que têm como finalidade principal o aprendizado do padrão pelo aluno, gerando como artefatos de saída o padrão catalogado e implementado.

A abordagem proposta foi aplicada por três alunos, do Curso Superior de Tecnologia em Análise de Sistemas, que não tinham conhecimento no assunto, durante o processo de aprendizagem de quinze padrões de projeto.

Este artigo tem a seguinte organização. A Seção 2 relata a abordagem metodológica para o aprendizado, compreensão, catalogação e aplicação dos padrões de projeto. A Seção 3 descreve os resultados da aplicação da abordagem proposta. A última seção apresenta as conclusões e os trabalhos futuros relativos a este artigo.

2. Abordagem Proposta

A abordagem proposta é composta por duas fases como ilustrado na Figura 1. A primeira fase é utilizada para o conhecimento sobre o assunto e para as decisões tais como: ferramentas de desenvolvimento, linguagem de modelagem, modelo de catalogação, entre outras. A segunda fase é destinada ao aprendizado de cada padrão de projeto, sendo que o entendimento do padrão é realizado individualmente.

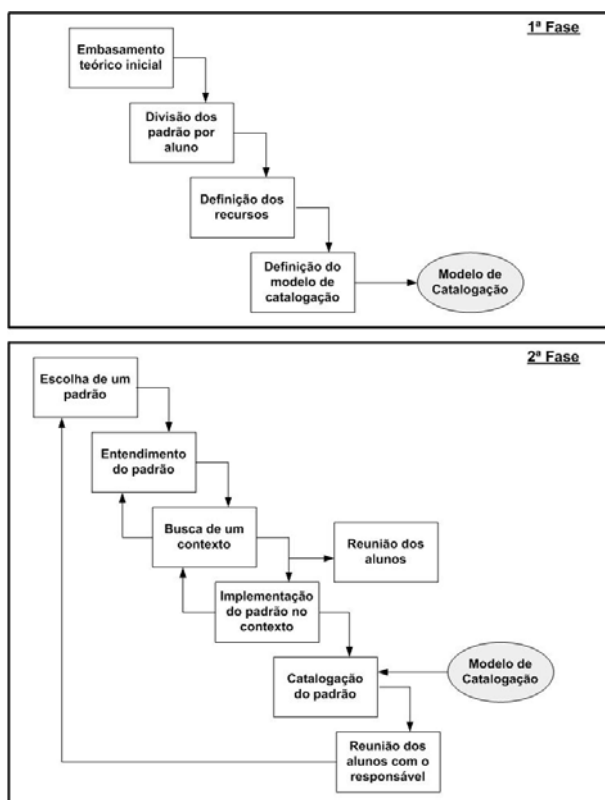


Figura 1. Processo utilizado para aprendizagem dos padrões de projeto.

As fases da abordagem proposta são compostas por um conjunto de atividades que estão descritas nas Tabelas 1 e 2, respectivamente.

Tabela 1 – Tarefas da 1ª Fase

Atividades	Descrição
Embasamento Teórico Inicial	Padrão de projeto é um tema bastante encontrado em publicações da literatura especializada. Entretanto, pouco se fala na parte de aprendizado e assimilação dos padrões antes de optar pela sua adoção. Assim sendo, nesta atividade se realiza a busca de material teórico sobre o assunto, tais como, livros, guias de desenvolvimento e tutoriais.
Divisão dos Padrões por Aluno	Como abordagem proposta preocupa-se com a divisão de tarefas de uma equipe, nesta atividade são estabelecidas às responsabilidades de cada integrante. Cada membro deve buscar os padrões de projeto que estão interligados, segundo o esquema de relacionamento proposto por Gamma et al. (1994), conciliando uma didática mais dinâmica com o emprego bem-sucedido daquilo que será aprendido.
Definição dos Recursos	Na criação de uma aplicação ou sistema, a fim de abordar um dado contexto, deve-se escolher os recursos, tais como, linguagem de programação, ferramenta e linguagem de modelagem, ambiente de desenvolvimento e persistência dos dados. Portanto, nesta atividade se identifica os recursos necessários para o desenvolvimento de uma dada aplicação.
Definição do Modelo de Catalogação	Não existe um consenso sobre como descrever um padrão: diferentes autores preferem diferentes estilos. Por esse motivo, nesta atividade busca-se selecionar da literatura especializada um modelo ideal para catalogação. A saída dessa atividade é o “Modelo de Catalogação” a ser utilizado pela equipe.

Tabela 2 – Tarefas da 2ª Fase

Atividades	Descrição
Escolha de um padrão	Cada participante deverá, nesta atividade, escolher um padrão para compreensão e implementação. Os padrões que serão implementados por cada integrante da equipe já foram definidos na primeira fase por meio da atividade “Divisão dos Padrões por Aluno”.
Entendimento dos padrões de projeto	Cada participante deve procurar material específico, principalmente exemplos práticos, sobre o padrão escolhido anteriormente, com o objetivo de compreender seu funcionamento para implementá-lo numa atividade posterior.
Busca de um contexto	Delimitar um contexto prático simples, no qual o padrão escolhido pode ser implementado.
Implementação do padrão no contexto	Implementação do padrão usando os recursos definidos na primeira fase por meio da atividade “Definição dos Recursos”.
Reunião dos alunos	Nesta atividade, os integrantes devem se reunir para repassar informações sobre o padrão implementado. Desta forma, todos os integrantes terão conhecimento sobre os padrões implementados pela equipe. Isso é importante, pois os padrões podem ser utilizados em conjunto na implementação de uma aplicação de um dado domínio.
Catalogação do padrão	A partir da atividade “Definição do Modelo de Catalogação”, primeira fase, o padrão será documentado. Por isso, possui como entrada o “Modelo de Catalogação”.
Reunião dos alunos com o responsável	O responsável por conduzir o aprendizado deverá validar com os alunos a implementação e catalogação dos padrões.

Na primeira fase as atividades ocorrem de forma sequencial, por sua vez na segunda as atividades são realizadas de forma iterativa. Ou seja, se cada integrante ficar responsável pela implementação de cinco padrões, então a segunda fase será executada cinco vezes para cada integrante.

3. Resultados da Aplicação da Abordagem Proposta

O objetivo do experimento foi verificar na prática como uma equipe que não conhecia os conceitos de padrões de projeto poderia aprendê-los. O experimento foi realizado com três alunos do Curso Superior de Tecnologia em Análise de Sistemas.

Seguindo o processo ilustrado na Figura 1, primeira fase, inicialmente os alunos realizaram pesquisas na Internet, livros e outros materiais, sobre o tema em questão. Posteriormente os alunos estudaram conceitos iniciais sobre padrões de projeto tais como: vantagens, desvantagens, classificação, bem como os modelos para sua descrição. Foi constatado neste estudo que não existia um consenso sobre como descrever um padrão de projeto.

O estudo inicial dos conceitos durou cerca de duas semanas, sendo que cada aluno gastou aproximadamente três horas diárias para as leituras e anotações. O próximo passo foi dividir os padrões que seriam analisados pelos alunos. Os padrões estudados e implementados por aluno estão listados na Tabela 3. Nesta tabela, também definiu-se o perfil do aluno.

Tabela 3. Padrões de projeto estudado e implementado por aluno.

Aluno	Padrão de projeto estudado e implementado	Perfil do Aluno			
		Afinidade com a Área	Interesse por estudos avançados	Capacidade para trabalho individual	Capacidade para trabalho em equipe
Aluno 1	Singleton; Adapter; Composite; Bridge; Builder	Médio	Médio	Médio	Médio
Aluno 2	Decorator; Facade; Chain of Responsibility; Iterator; Strategy	Excelente	Excelente	Excelente	Excelente
Aluno 3	Factory Method; Abstract Factory; Memento; Prototype; State	Bom	Excelente	Bom	Excelente

O critério utilizado para essa divisão dos padrões de projeto levou em consideração dois fatores principais:

- Evitar a ocorrência de padrões similares.
- Buscar padrões efetivamente diferentes, mas que estivessem relacionados de alguma maneira. Neste critério utilizou-se o relacionamento dos padrões fornecidos por Gamma et al. (1994). Por exemplo, o padrão *Factory Method* participa da estrutura do padrão *Abstract Factory*, e assim sucessivamente.

Ressalta-se que para a realização desta divisão foram gastos aproximadamente três horas de trabalho, em uma reunião que se fazia presente os alunos e o responsável pela orientação.

Logo após, houve a definição de alguns recursos que seriam utilizados, tais como: a linguagem de programação e o ambiente tanto para desenvolvimento quanto

para modelagem. Essa atividade levou em torno de cinco horas, visto que os alunos já tinham conhecimento dos recursos a serem usados para desenvolvimento.

A linguagem adotada para implementação dos padrões de projeto foi Java, devido sua portabilidade, ser gratuita e de possuir código aberto. Tomando como base essa linguagem, os ambientes de desenvolvimento e de modelagem foram selecionados com o objetivo de aumentar a produtividade e a qualidade do trabalho produzido. Todos os ambientes escolhidos provinham de interface gráfica baseada em widgets – elementos de tela como janelas e botões – e foram desenvolvidos utilizando a própria linguagem Java.

Os ambientes de desenvolvimento escolhidos foram o JDeveloper e o Eclipse. Ambos oferecem um ambiente gratuito e integrado de desenvolvimento que provêem recursos para modelar, programar, depurar código, otimizar e distribuir aplicações em linguagem Java. Por fim, o ambiente de modelagem selecionado foi a JUDE (*Java and UML Developer Environment*) por ser uma ferramenta gratuita para geração de diagramas de representação de objetos.

Na última atividade da primeira fase, os alunos adaptaram a estrutura de Gamma et al. (1994), ilustrado na Tabela 4, para estabelecer o modelo de catalogação que seria adotado.

Tabela 4. Modelo de descrição de padrão de projeto adotado (Laudelino et al. 2005, p.11)

Item	Descrição
Nome do padrão	Descreve a essência do padrão em um curto, porém expressivo, nome.
Conhecido como	Lista de sinônimos para o padrão.
Classificação	Mostra em qual das categorias o padrão se enquadra.
Propósito	Descreve o que o padrão faz.
Motivação	Fornece um exemplo de um problema e como o padrão o resolve.
Aplicabilidade	Lista as situações onde o padrão é aplicado.
Estrutura	Conjunto de diagramas de classes e objetos que descrevem o padrão.
Conseqüências	Descreve as vantagens e desvantagens em se utilizar determinado padrão.
Implementação	Ilustra um exemplo concreto do padrão utilizado na prática ou simulando suas funcionalidades.
Notas de aprendizado	Relata a contribuição do padrão de projeto no aprendizado e também descreve os dados da experiência dos desenvolvedores em implementar determinado padrão, com o objetivo de prover referência para estatísticas e trabalhos futuros.

O modelo ilustrado na Tabela 4 foi escolhido porque captura informações essenciais e necessárias para o entendimento do problema e a estruturação da solução. A definição desse modelo foi discutida em três reuniões, perfazendo um total de dez horas.

Depois de completado as atividades da primeira fase, cada aluno estudou e implementou seus padrões seguindo o processo ilustrado na Figura 1 na segunda fase.

As Tabelas 5, 6 e 7 ilustram o tempo que cada aluno usou para a aprendizagem do padrão, bem como suas conclusões sobre o padrão.

Tabela 5. Tempo destinado ao aprendizado dos padrões de projeto e conclusões do Aluno 1.

Nº	Padrão de Projeto	Tempo destinado para aprendizagem e implementação		Parecer fornecido pelo aluno sobre o padrão
		Pesquisa e Estudo	Implementação e Catalogação	
1	Singleton	2 dias	5 dias	Este é um padrão bastante simples para ser entendido. Os materiais de pesquisa são abundantes e variados. A maior dificuldade identificada foi encontrar um exemplo que realmente fosse mais complexo.
2	Adapter	8 dias	8 dias	A implementação normalmente é simples e possui grande material de consulta em livros e na Internet, entretanto, devem-se considerar alguns aspectos importantes na hora da implementação, tais como a escolha do exemplo.
3	Composite	8 dias	7 dias	A utilização desse padrão é grande e faz com que um sistema complexo se dividida em pequenas soluções. O material é abundante e suas aplicações são diversas. Esses fatores demandaram um tempo maior para pesquisa e estudo, até mesmo para poder ler e entender todo o material disponível.
4	Bridge	12 dias	10 dias	Este é um padrão fácil de ser entendido, porém os materiais de pesquisa são sucintos em quantidade e em qualidade, o que dificulta seu aprendizado. Esses fatores consumiram um tempo maior para pesquisa e implementação. A maior dificuldade encontrada foi efetuar às adaptações e correções na implementação de uma aplicação-exemplo.
5	Builder	2 dias	6 dias	Requer um nível maior de conhecimento, pois pode utilizar outros padrões como Composite e Factory Method. Além disso, separar métodos e atributos de uma classe em classes diferentes não é tão fácil. Deve-se ainda observar o fato de que a classe, que reservará os métodos, deve conhecer os atributos da classe aos quais deve preencher. Contudo, a tarefa mais difícil é o entendimento e a aplicação de estratégias para a criação de representações diferentes à partir de uma mesma classe, sem estender em demasia o código da aplicação – já que um dos objetivos de Builder é a simplificação.

Tabela 6. Tempo destinado ao aprendizado dos padrões de projeto e conclusões do Aluno 2.

Nº	Padrão de Projeto	Tempo destinado para aprendizagem e implementação		Parecer fornecido pelo aluno sobre o padrão
		Pesquis a e Estudo	Implementação e Catalogação	
6	Decorator	2 dias	3 dias	<p>Este padrão foi o que causou grande dificuldade, devido a confusões feitas por algumas fontes de informação sobre ele. Foram encontradas várias definições e exemplos de implementações diferentes, algumas, inclusive, que fugiam completamente da idealidade apresentada por Gamma et al. (1994).</p> <p>Depois de compreendido o padrão, é vista a sua importância, pois ele é utilizado principalmente quando o sistema exige uma interface complexa com o usuário, variando as características visíveis nesta interface de um usuário para outro.</p>
7	Facade	1 dia	1 dia	<p>Padrão fácil de ser aprendido, no entanto, inicialmente causa pensamentos do estilo: "Será que é isso mesmo? Pois parece ser tão óbvio!" ou "Tem alguma coisa errada, nada é tão simples".</p>
8	Chain of Responsibility	7 dias	7 dias	<p>Este padrão inicialmente parece simples, no entanto, ele causa indagações com relação a sua real utilidade. Não se chega a perceber em primeiro momento que ele é um padrão importante e útil. É difícil imaginar cenários aplicáveis a ele, o que dificulta sua implementação.</p>
9	Iterator	15 min	45 min	<p>O padrão pode entrar em discussão quanto ao tempo de aprendizagem. Na realidade, a compreensão de como ele funciona é extremamente rápida, entretanto, a clara finalidade do padrão pode variar muito com relação ao tempo de experiência com a linguagem Java.</p> <p>Programadores com uma experiência em Java o assimilaram com grande facilidade já que ele usa <code>java.util.Iterator</code> e o <code>java.util.Enumeration</code> da J2SE API. Além de que no JEE o padrão <code>ValueListHandler</code> também introduz conceitos semelhantes.</p>
10	Strategy	1 h	1h30min	<p>Padrão simples para a aprendizagem, já que é percebido de início que a mudança é no algoritmo contido nas classes que implementam a estratégia. Ou seja, cada uma define de forma própria os rumos que pretende seguir, mas receberá os mesmos dados e os retornará do mesmo tipo.</p> <p>Esse padrão possui grande aplicabilidade na implementação de jogos, pois no decorrer da história do mesmo, pode haver cenários semelhantes, mas a possibilidade de tomadas de rumos diferentes.</p>

Tabela 7. Tempo destinado ao aprendizado dos padrões de projeto e conclusões do Aluno 3.

Nº	Padrão de Projeto	Tempo destinado para aprendizagem e implementação		Parecer fornecido pelo aluno sobre o padrão
		Pesquisa e Estudo	Implementação e Catalogação	
11	Factory Method	4 dias	8 dias	<p>Por possuir um nível elevado de abstração, levou um tempo maior para ser entendido, mesmo porque usa quase que obrigatoriamente outros dois padrões como o Singleton e o Factory Method.</p> <p>As maiores dificuldades ficaram no entendimento da abstração, envolvendo vários grupos relacionados de objetos reunidos por uma interface, como uma classe interna e a interface gráfica. Essa interface foi projetada para receber os produtos das famílias de maneira generalizada, sem, entretanto, perder o sentido do domínio da aplicação.</p>
12	Abstract Factory	2 dias	6 dias	<p>Este é um padrão simples para ser entendido. Os materiais de pesquisa são abundantes em quantidade, qualidade e variedade. A maior dificuldade encontrada foi às adaptações que tiveram que ser feitas para adequar a interface gráfica do sistema, uma vez que a execução se dava por linha de comando e seu retorno era feito pela interface texto do sistema.</p>
13	Memento	1 dia	2 dias	<p>É um padrão simples de ser implementado, apenas pede atenção ao criar a classe de Mementos, dando atenção ao tamanho e, se houver, a persistência desses elementos de estado de objetos.</p> <p>Ainda, a recuperação ordenada de Mementos armazenados ou persistidos, utilizando vetores ou índices, pede dedicação, para que a aplicação não perca sua consistência.</p>
14	Prototype	2 dias	3,5 dias	<p>Há uma dificuldade inicial é entender os “níveis” de clonagem oferecidos pelo padrão Prototype. Superado esse obstáculo, a implementação se torna simples; porém requer maior atenção, pois lida com a possibilidade de reduzir o número de classes de um projeto já existente. Nessas condições, o problema reside em criar classes-exemplo que sejam genéricas o bastante e que possibilitem a implantação do Prototype, sem, entretanto, perder a especificidade do domínio da aplicação, o que obriga a impor o detalhamento – ou a especialização – dos objetos após o processo de “clonagem”.</p>
15	State	20 min	1h	<p>O padrão State evita a complexidade, reduz as estruturas condicionais, garante a facilidade de interpretação de algoritmos que o utilizarão e assegura que o objeto irá se comportar de acordo com o contexto. Por isso, sua aprendizagem é rápida, já que o mesmo não apresenta uma estrutura complexa.</p>

Cada aluno conseguiu aprender e implementar os padrões a ele designado utilizando o tempo que foi apresentado nas Tabelas 5, 6 e 7. Nessas tabelas, o total de

dias relatados não representa o número de horas efetivamente gastas para a aprendizagem, uma vez que a média de horas trabalhadas por dia foi de aproximadamente três horas. Por isso, a Figura 2 ilustra um grafo comparativo da média de horas trabalhada por aluno.

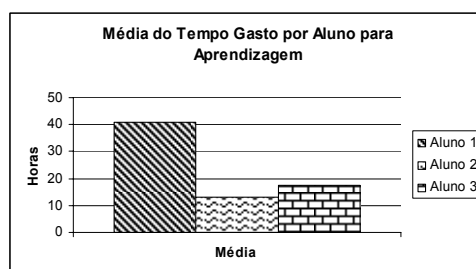


Figura 2. Média de horas para aprendizagem dos padrões de projeto.

Tomando como base a Figura 2 e os perfis dos alunos descritos na Tabela 3, pode-se concluir que o Aluno 2, o qual apresenta uma melhor pontuação nos itens analisados, conseguiu aprender e implementar os cinco padrões em aproximadamente treze horas, que foi o melhor tempo. Por sua vez, o Aluno 3 desenvolveu suas atividades em dezessete horas e possuía a segunda colocação sobre os itens analisados. Por fim o Aluno 1, terceiro colocado na pontuação, desenvolveu seus trabalhos em uma média de aproximadamente quarenta horas. Mas, apesar das diferenças de perfis todos cumpriram com seus objetivos.

A dificuldade encontrada pelo grupo, foi que nas reuniões realizadas quinzenalmente o Aluno 1 não repassava todo o conhecimento sobre o padrão, visto que o mesmo muitas vezes tinha compreendido o padrão, mas não havia conseguido implementá-lo.

As reuniões, quarta atividade da segunda fase, eram realizadas com o objetivo de permitir que cada aluno repassasse o conhecimento de seus padrões.

No final da segunda etapa, constatou-se que a maior dificuldade dos alunos, muitas vezes não estava em compreender o padrão, mas sim em encontrar um exemplo no qual ele pudesse ser aplicado. Sabe-se que na internet, livros e outras referências existem uma grande quantidade de exemplos, mas muitos deles estão implementados de forma errada, ou possuem erros de compilação e até mesmo de execução.

Ressalta-se que por isso, muitos dos exemplos implementados pelos alunos foram criados e até mesmo adaptados. Dos quinze padrões implementados, sete foram criados pelos próprios alunos.

4. Conclusões

O presente trabalho descreveu a abordagem utilizada para aprendizado, compreensão e catalogação de quinze padrões de projeto. Essa abordagem mostrou-se eficiente para a aprendizagem, pois os padrões eram catalogados por meio de uma aplicação simples.

A divisão das tarefas aumentou a produtividade da equipe, pois cada aluno era responsável por implementar cinco padrões, proporcionando a realização das atividades de maneira concomitante. Além disso, como foram estabelecidos os padrões de

catalogação na primeira fase, não houve problema durante a integração dos materiais produzidos pela equipe, pois todos os integrantes seguiram os padrões previamente estabelecidos.

Portanto, a metodologia proposta pode ser utilizada para diminuir o impacto do ensino de padrão de projeto durante o curso de graduação, tornando fácil o entendimento e o aprendizado neste assunto.

Este trabalho poderá ser objeto de pesquisa futura, em se pode aplicar a abordagem proposta para o aprendizado, catalogação e implementação de outros padrões de projeto, bem como realização de comparações de quando aplicar padrões de projeto no desenvolvimento de sistemas mais complexos.

5. Referências

- ALEXANDER, C. (1977). **A Pattern Language: Towns, Buildings, Constructions**. New York: Oxford University Press.
- AMBLER, S. W. (1998). **Análise e projeto orientados a objetos**. Rio de Janeiro: Infobook.
- GAMMA, E. HELM, R. JOHNSON, R. VLISSIDES, J. (1994). **Design Patterns: Elements of Reusable Object-Oriented Software**. Reading, MA: Addison-Wesley.
- LAJOIE, R. KELLER, R. K. (1994). Design and Reuse in Object-Oriented Framework: Patterns, Contracts and Motifs in Concert. **Proceedings of the 62nd Congress of the Association Canadienne Francaise pour l'Avancement des Sciences**, Canad.
- LAUDELINO, A. S.; KRIK, D. O.; MAIA, F. E. P. (2005). **Aplicabilidade de Padrões de Projeto Catalogação, Desenvolvimento e Implementação**. 151f. Trabalho de Conclusão de Curso - Universidade Tecnológica Federal do Paraná.
- METSKER, S. J. **Padrões de Projeto em Java**. Bookman, Tradução Werner Loeffler. Porto Alegre: Bookman, 2004.