

Metodologia de Ensino de Padrões de Projeto Baseado no Modelo 4C/ID

Adilson Vahldick¹, Pablo Schoeffel¹, Paolo Moser¹

¹Departamento de Engenharia de Software, Centro de Educação Superior do Alto Vale do Itajaí, Universidade do Estado de Santa Catarina (Udesc) – Ibirama, SC – Brasil

{adilson.vahldick, pablo.schoeffel, paolo.moser}@udesc.br

Abstract. *For three semesters, for the Design Patterns discipline, we used an instructional project called Four Components/Instructional Design (4C / ID), which favors complex learning, which, based on one of its components, was developed. instructional content and practical exercises. With the beginning of a new semester, students from the previous semester were asked about these practices in the course. The results indicated that the students agreed with the adoption in the way the standards were taught and the exercises were proposed and organized. Students were able to understand how to achieve reuse and produce better quality code for their maintenance.*

Resumo. *Durante três semestres, para a disciplina de Padrões de Projeto, utilizou-se um projeto instrucional chamado Four Components/Instructional Design (4C/ID), que favorece a aprendizagem complexa, em que, baseando-se em um de seus componentes, foram desenvolvidos o conteúdo instrucional e os exercícios práticos. Com o início de um novo semestre, os alunos do semestre anterior foram inquiridos sobre essas práticas na disciplina. Os resultados apontaram que os alunos concordaram com a adoção na forma como os padrões foram ensinados e como foram propostos e organizados os exercícios. Os alunos conseguiram compreender como alcançar a reutilização e produzir código com melhor qualidade para a sua manutenção.*

1. Contextualização

[Gamma et al. 2000] definiram um catálogo de 23 padrões de projeto. Um padrão de projeto descreve um problema e como solucioná-lo, considerando objetos e interfaces, e que pode ser usado um milhão de vezes, sem nunca o fazer da mesma maneira. Os padrões de projeto ajudam a escolher soluções que tornam um sistema reusável e, por consequência, facilitam a sua manutenção. Os padrões de projeto raramente são usados isoladamente e, na maioria das vezes, precisam ser modificados e adaptados para coexistirem com outros padrões. Conforme [Astrachan et al. 1998], os padrões de projeto facilitam a transição para um modo de pensar orientado a objetos, pois apresentam técnicas de como usar a orientação a objetos correta e eficientemente.

Os alunos têm dificuldade em perceberem a necessidade dos padrões de projeto. A utilização de estudos de caso apropriados pode potencializar o aprendizado dos padrões [Gestwicki 2007]. “O aprendizado correto e contínuo de padrões de projeto deve representar uma prática tradicional em qualquer curso que queira oferecer uma sólida formação no paradigma de orientação a objetos” [Silveira e Silva 2006]. Ainda, conforme [Pillay 2010], ensinar padrões de projeto também não é uma tarefa fácil: primeiro, lembrar os alunos que já usaram padrões em outras disciplinas (por exemplo, de repetição em

disciplinas introdutórias), apresenta-los cada padrão, fornecer exercícios para que eles pratiquem, e ainda deixar claro que os padrões não são usados isoladamente, mas que normalmente, mais de um padrão é usado simultaneamente, e ainda elaborar avaliações para que os alunos decidam qual padrão é melhor aplicável e como fazê-lo.

A eficiência dos padrões de projeto é acessível para aqueles com domínio em tecnologias orientada a objetos [Astrachan et al. 1998]. Por essa razão, a disciplina Padrões de Projeto é oferecida no quinto semestre no Curso de Bacharelado em Engenharia de Software da Udesc. A Figura 1 apresenta as disciplinas pré-requisitos para cursar Padrões de Projeto, cada uma com o número de seu semestre e a sua carga horária semanal. Pode-se observar que os alunos precedem com 252 horas/aula de programação e ainda 144 horas/aula específicas de engenharia de software (requisitos, ferramentas case, modelagem, UML, etc).

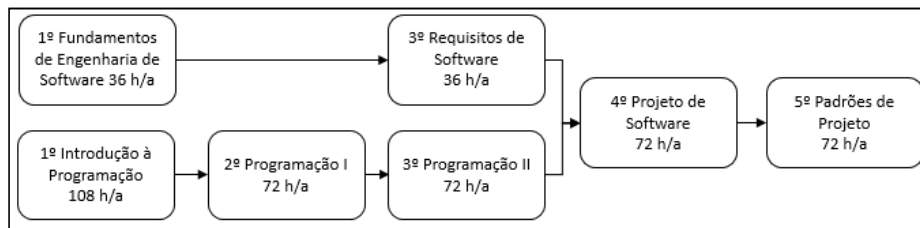


Figura 1 - Pré-requisitos da Disciplina de Padrões de Projeto

O objetivo desse artigo é apresentar a metodologia de ensino usada nessa disciplina, principalmente o projeto instrucional de como cada padrão foi ensinado. O projeto instrucional descreve métodos de como ensinar e as situações em que esses métodos devem ser aplicados [Wiley 2000]. Essa metodologia foi aplicada e refinada durante três semestres (2018-1, 2018-2, 2019-1). No semestre atual (2019-2), na disciplina de Estratégias de Experimentação e Observação (sexto semestre), foi aplicado o instrumento de avaliação eVALUate [Oliver et al. 2008] com os alunos que fizeram a disciplina em 2019-1, para avaliarem seu engajamento e os resultados da aprendizagem.

Para modelar o ensino de cada padrão foi adotado o *Four Component/Instructional Design* (4C/ID), que busca reduzir a carga cognitiva do aluno durante o seu aprendizado, facilitando a aquisição de conhecimento complexo [Merriënboer e Kirschner 2018]. Esse modelo é composto por quatro componentes que fornecem uma maneira de organizar e estruturar o material instrucional, com seus conteúdos e atividades (Figura 2). O componente foco desse trabalho é relativo às Tarefas de Aprendizado que são estudos de caso, projetos, problemas, entre outros, que visam a integração entre as habilidades, conhecimentos e atitudes. O objetivo primário dessas tarefas é induzir o aluno a construir esquemas cognitivos a partir de experiências concretas. Essas tarefas devem ser organizadas do nível mais fácil para o difícil, diferenciando-se pela quantidade de apoio ao aluno, iniciando por bastante apoio até pouco ou nenhum apoio.

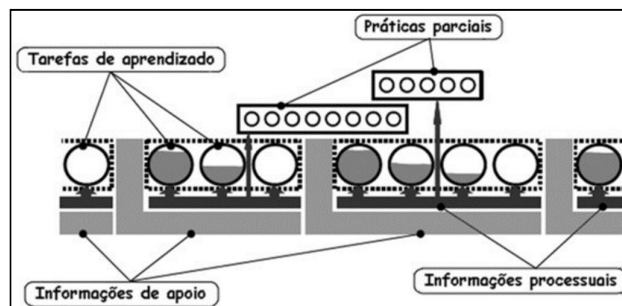


Figura 2 - Arquitetura do Modelo 4C/ID

As Informações de Apoio ajudam os alunos na execução de situações não rotineiras das tarefas de aprendizado, as quais frequentemente envolvem raciocínio e resolução de problemas. As Informações Processuais permitem que o aluno aprenda e realize as situações rotineiras das tarefas de aprendizado, e devem ser apresentadas ao aluno apenas quando necessário, para lembrar algum assunto. As Práticas Parciais também se referem aos aspectos rotineiros das tarefas, mas aqui, em vez de informações, são oferecidos exercícios sobre estes aspectos, de forma a proporcionar um nível mais elevado de automação na execução das tarefas. Estas práticas são necessárias apenas quando as Tarefas de Aprendizado não fornecem uma quantidade de repetição suficiente. O projeto instrucional desse modelo define 10 passos para produzir esses quatro componentes [Merriënboer e Kirschner 2018], onde os três primeiros passos se referem à produção de tarefas de aprendizado. No primeiro passo é especificado um conjunto de tarefas de aprendizado que representa uma habilidade complexa que o estudante precisa estar apto a desempenhar após cumprir toda a instrução. O passo 2 se refere à articulação dos objetivos instrucionais que o estudante precisa alcançar. Nesse passo são determinados se os objetivos foram atendidos e dá aos estudantes o necessário *feedback* na qualidade do seu desempenho. Quando se tem um conjunto de tarefas de aprendizado com os instrumentos para avaliar o desempenho, as tarefas precisam ser ordenadas de tal forma que o processo de aprendizado seja otimizado (passo 3). Isso pode ser feito pela especificação da progressão das tarefas conforme sua complexidade, e para cada novo nível de complexidade, deve-se diminuir o suporte e a orientação. Outra forma, é utilizar os resultados da avaliação para dinamicamente selecionar as tarefas conforme o progresso do estudante.

O trabalho está estruturado da seguinte maneira: nesta seção foi apresentado o problema e descrito o modelo 4C/ID. Na seção 3 será descrita a metodologia de ensino adotada na disciplina e na quarta será apresentada a avaliação e discussão dos resultados dessa investigação. As considerações finais são descritas na quinta seção.

3. Metodologia de Ensino de Padrões de Projeto Adotada

O objetivo da disciplina Padrões de Projeto no curso de Bacharelado em Engenharia de Software da Udesc era “produzir software com base em reusabilidade de código, considerando o uso de padrões de projeto e a implementação de frameworks”. O desafio nessa disciplina foi permitir que os alunos conhecessem o que é um padrão de projeto, compreendessem alguns padrões e conseguissem aplicar o padrão correto conforme a situação problema. A linguagem de programação adotada era Java.

A partir de 2018/1 foi estabelecida a necessidade de adotar alguma metodologia de ensino que consiga fornecer código dos padrões que já fossem úteis, e que os alunos pudessem resolver novos problemas aplicando padrões, assim como um problema maior aplicando todos os padrões ensinados num único projeto.

A disciplina foi dividida em três módulos que encerram com uma avaliação (prova individual com componentes teóricos e práticos). O curso era noturno e a disciplina tinha 36 aulas (72 horas/aula), sendo ofertada a noite inteira (2 aulas por dia, 18 semanas de aula). Isso permitiu adotar a estratégia de ensinar um padrão diferente por noite, contemplando a parte teórica e seguindo com a prática. Foram definidos 16 (dos 23) padrões de projeto que mais são usados na prática: criação (*Abstract Factory*, *Builder*, *Factory Method*, *Prototype* e *Singleton*), estruturais (*Adapter*, *Composite*, *Decorator*, *Façade* e *Proxy*) e comportamentais (*Command*, *Observer*, *Visitor*, *State*, *Strategy* e *Template Method*). Em algumas noites foram ensinados mais de um padrão devido a sua simplicidade. Além dos 16 padrões de projeto, a disciplina contemplava o padrão de

arquitetura MVC que era ensinado na primeira parte, consumindo oito aulas, pela quantidade e complexidade dos exercícios.

Para cada padrão foi aplicado o projeto instrucional 4C/ID, criando quatro ou cinco projetos com diferentes situações e estudos de caso na seguinte forma: o primeiro continha a solução completa do padrão sendo aplicado, com seu código fonte, o qual também era usado na explicação do início da aula; no segundo projeto faltava alguma parte pequena da aplicação do padrão a ser implementado; no terceiro (e por vezes um quarto) faltavam mais partes da aplicação do padrão; e no último projeto era apresentado o enunciado de um estudo de caso e, quando necessário, uma ou outra classe de modelo, para que o aluno implementasse o projeto do princípio. Quase sempre os exemplos e exercícios disponíveis na literatura e na internet acabavam por ser distantes ou muito simplistas para demonstrar o uso do padrão. Para cada padrão de projeto, como primeiro passo na produção dessas tarefas, foram formulados problemas reais e com maior complexidade, os chamados exemplos matadores [Alphonse, Caspersen e Decker 2007] que apresentavam contextos maiores para demonstrar a utilidade e os benefícios no uso de padrões. O segundo passo (em paralelo com o terceiro) foi a definição dos objetivos em cada tarefa, e assim especificar a quantidade de orientação e suporte (código pronto) que estaria disponível.

As aulas eram conduzidas: (1) com base na cópia física do livro de [Gamma et al. 2000], interpretava-se primeiro a intenção do padrão e discutia-se a estrutura de classes proposta pelos autores; (2) a explicação do padrão era complementada demonstrando a sua aplicação com o código disponível no primeiro projeto; (3) os três (ou quatro) projetos seguintes eram fornecidos como exercícios para que os alunos pratiquem e fundamentem o seu conhecimento. Normalmente, acontecia a correção do segundo projeto com toda a turma e, depois, conforme o ritmo de cada um, os alunos iam apresentando individualmente ao professor as suas soluções durante a própria aula. O último projeto costumava ser complexo e era resolvido extraclasse, com os alunos procurando o professor para resolver as suas dúvidas em horário além da aula. Com essa metodologia os alunos tinham a chance de iniciarem o aprendizado com um exemplo de código fonte completo e funcional, e a partir dele, evoluir em cada exercício, até alcançar a última tarefa em que ele precisava produzir um projeto completo do início.

Para o padrão *Command*, por exemplo, foram criadas as seguintes tarefas (todas com a necessidade de implementar métodos para desfazer e refazer, que fazem parte do padrão):

1. Calculadora com as operações de adição e subtração encapsuladas em comandos. Como exercício complementar dessa primeira aplicação era sugerido criar um novo *Command* concreto para a operação de multiplicação;
2. Controle de estoque em que faltava o comando para retirar do estoque;
3. Sistema para cadastro de pessoas com o comando de listar as pessoas implementado. Faltavam comandos para incluir, excluir e recuperar uma pessoa;
4. Fornecida uma aplicação similar ao Paint, a qual devia ser feita a refatoração para aplicar o padrão *Command*.

Para complementar a compreensão no funcionamento de cada padrão de projeto, no primeiro dia de aula era fornecido o código em Java de um jogo inspirado no Zanic¹, utilizando Graphics2D. Na Figura 3 são mostradas as classes da versão entregue pelo

¹ <https://pt.wikipedia.org/wiki/Zanac>

professor. Para facilitar a visualização foram suprimidos os atributos e métodos. A *Nave* controlada pelo jogador dispara objetos *Bullet* para destruir os inimigos (*Duster*, *Luster* e *Box*). O inimigo *Luster* dispara objetos *Lead*. A nave é destruída com o contato de um dos inimigos ou com os objetos disparados pelo *Luster*. Após a destruição de uma quantidade de inimigos aparecem três objetos *Box* que voam em sincronia. Um deles possui um *PowerChip*, que aparece após o objeto *Box* ser destruído. Esse *PowerChip* aumenta o poder de fogo da *Nave*, disparando mais objetos *Bullet* a cada vez. O projeto entregue aos alunos foi desenvolvido sem aplicação de padrões e sem preocupações de reaproveitamento. O professor trazia problemas reais quanto à necessidade de reusabilidade, em que novos requisitos aparecem e o jogo devia sofrer manutenção com o mínimo de impacto no código existente. O projeto fomentava a discussão de como planejar e projetar uma aplicação que suporte sua evolução. Os alunos faziam a refatoração do projeto com a aplicação de alguns padrões de projeto, aumentando a facilidade de manutenção quando novos requisitos apareciam. A cada aula, o projeto evoluía com a aplicação do padrão aprendido naquela aula.

Foram definidos cinco instrumentos de avaliação para a disciplina. Conforme já mencionado, ao término de cada um dos três módulos era aplicada uma prova individual. Além disso, em duplas, os alunos desenvolviam um projeto durante o semestre, com uma entrega parcial ao término do segundo módulo, e outra ao fim do semestre. Em cada uma das entregas eram avaliadas a aplicação de um conjunto de padrões (a versão final complementa a primeira entrega). Em 2018/1 e 2018/2, o projeto foi desenvolvido individualmente e os alunos podiam escolher qualquer linguagem de programação. Durante esse período houveram projetos, além de Java, em PHP, Javascript, Android e Embarcadero Delphi. Porém, o professor verificou que os projetos feitos em outras linguagens ficaram com a qualidade inferior na aplicação dos padrões, não pelas limitações das linguagens, mas provavelmente pelo conhecimento dos alunos no domínio dessas linguagens. Por essa razão, a partir de 2019/1, o professor definiu que o projeto poderia ser em duplas, porém sempre com a linguagem Java. Os enunciados dos projetos eram fornecidos pelo professor com objetivo de desenvolver versões digitais de jogos de tabuleiro: 2018/1-Dou Shou Qi, 2018/2-Hnefatafl e 2019/1-Haru Ichiban.

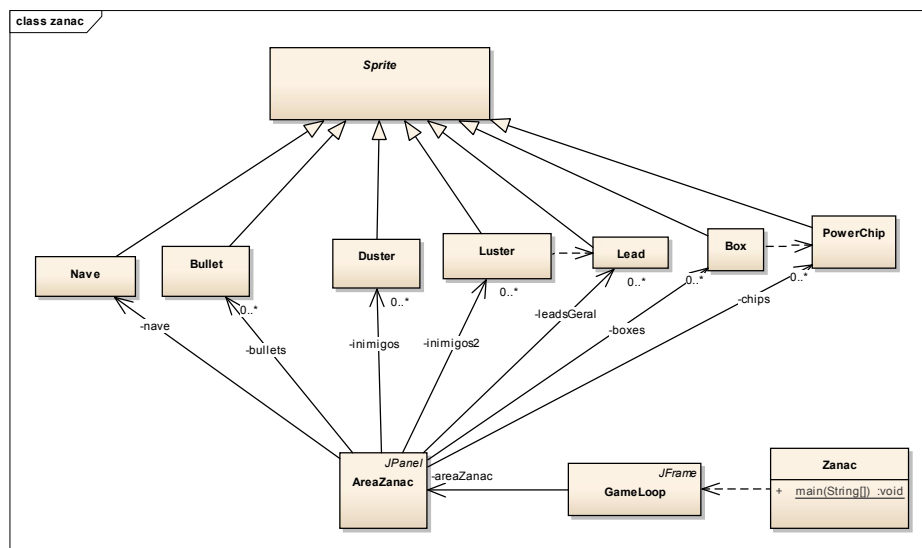


Figura 3 – Classes do Projeto Zanac (Versão Inicial)

4. Avaliação da Metodologia de Ensino

A fim de avaliar a percepção dos estudantes a respeito da metodologia de ensino adotada na disciplina, foi aplicado o instrumento *eVALUate* [Oliver et al. 2008] com dezessete alunos que participaram da disciplina de Padrões de Projeto em 2019-1. O *eVALUate* é um questionário que visa identificar três aspectos principais: (1) o que ajuda os estudantes a aprender, (2) o que os estudantes trazem para a parceria de ensino-aprendizagem, e (3) a satisfação geral dos estudantes com a unidade. O instrumento foi traduzido para o português e é composto por onze itens com cinco opções de resposta numa escala *likert* de cinco pontos (totalmente em desacordo a totalmente de acordo), conforme Quadro 1.

Além disso, o instrumento possui duas perguntas descritivas: (1) Quais foram os aspectos mais úteis desta disciplina? e (2) Como você acha que esta disciplina pode ser melhorada?

Quadro 1 – Instrumento de avaliação eVALUate

- | |
|---|
| <ol style="list-style-type: none"> 1. Os resultados da aprendizagem nesta disciplina foram claramente identificados. 2. As experiências de aprendizagem nesta disciplina me ajudaram a alcançar os resultados da aprendizagem. 3. Os recursos de aprendizado desta disciplina me ajudaram a alcançar os resultados de aprendizagem. 4. As tarefas de avaliação nesta disciplina avaliaram minha conquista dos resultados da aprendizagem. 5. O <i>feedback</i> sobre o meu trabalho nesta disciplina me ajudou a alcançar os resultados da aprendizagem. 6. A carga de trabalho nesta disciplina foi apropriada para a obtenção dos resultados da aprendizagem. 7. A qualidade do ensino nesta disciplina ajudou-me a alcançar os resultados da aprendizagem. 8. Estive motivado para alcançar os resultados de aprendizagem nesta disciplina. 9. Eu fiz o melhor uso das experiências de aprendizado nesta disciplina. 10. Eu ficava pensando em como eu podia aprender mais efetivamente nesta disciplina. 11. De forma geral, eu fiquei satisfeito com essa disciplina. |
|---|

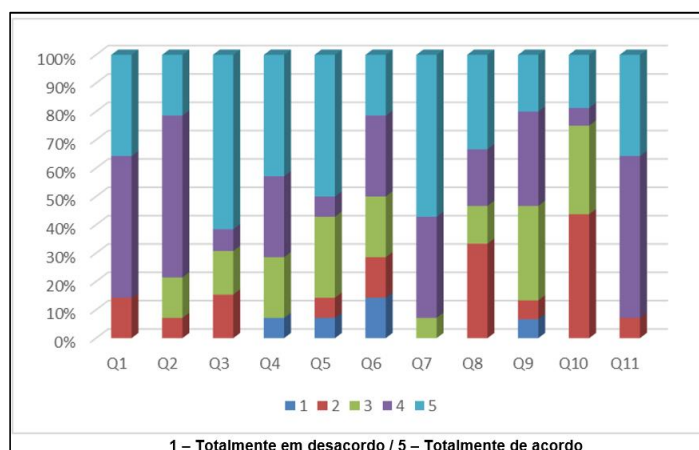
A fim de validar o instrumento utilizado, foi calculado o coeficiente alpha de Cronbach para avaliar a confiabilidade interna do questionário, obtendo o valor 0,8566. Existem diferentes relatos sobre os valores aceitáveis do alpha de Cronbach, no intervalo de 0,70 a 0,95 [Tavakol e Dennick 2011].

Para evidenciar a validade convergente e discriminante dos itens do questionário, foram calculadas as correlações entre os itens (intercorrelações) e a correlação item-total, usando uma matriz de correlação [Deville 2003], conforme a Tabela 1. O ideal é que as médias das intercorrelações estejam entre 0,15 e 0,50 indicando que medem bem o mesmo construtor [Clark e Watson 1995]. De qualquer forma, a média da correlação entre os itens foi de 0,435, entendendo-se que estão medindo satisfatoriamente um mesmo construtor.

A Figura 4 mostra o detalhamento das respostas por questão. É possível perceber que os itens relacionados aos recursos e metodologia de ensino-aprendizagem (itens 1 a 5 e 7) foram melhor avaliados. Os itens relacionados ao engajamento dos estudantes foram piores avaliados (itens 8, 9 e 10), especialmente o item 10, que foi o único que teve a maioria das respostas neutras ou negativas.

Tabela 1 – Intercorrelação dos itens do instrumento de avaliação

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Q1	-									
Q2	0,727	-								
Q3	0,377	0,529	-							
Q4	0,514	0,531	0,303	-						
Q5	0,334	0,445	0,299	0,883	-					
Q6	0,259	0,204	0,552	0,057	0,218	-				
Q7	0,532	0,805	0,249	0,283	0,237	-0,043	-			
Q8	0,445	0,395	0,363	0,383	0,434	0,424	0,356	-		
Q9	0,451	0,193	0,210	0,129	0,151	0,129	0,173	0,488	-	
Q10	0,338	0,129	0,102	0,308	0,258	0,316	0,000	0,742	0,348	-
Q11	0,722	0,781	0,280	0,744	0,682	0,034	0,746	0,594	0,342	0,336

**Figura 4 – Respostas por item**

Um índice fundamental, proveniente dos estudos da teoria da informação de [Shannon 1948] é a Entropia de Shannon ($H(x)$), que mede a “quantidade” de informação contida uma distribuição estatística: quanto maior a Entropia, mais informativa é a distribuição (considera-se aqui, cada questão como uma distribuição estatística independente das demais). $H(x)$ é calculada por:

$$H(x) = - \sum_{i=1}^k P(x_i) \log_2 P(x_i)$$

A escala *likert* é de natureza qualitativa ordinal, pois existe uma ordem natural entre os itens da escala (p.e., entre discordo totalmente e concordo totalmente). Para realizarem-se operações algébricas normalmente acaba-se transformando a escala em quantitativa discreta (neste caso, pontuando-a de 1 a 5). Quando isso é feito, admite-se incorretamente que existe uma relação linear entre os níveis de concordância. Essa relação linear não existe subjetivamente. Acrescenta-se que todos os descritores supracitados apresentam propriedades que, de certa forma, enfraquecem seu potencial sintético/explicativo, quando aplicados às escalas *likert*. $H(x)$ é invariante à ordem, i.e., a entropia é sempre a mesma independente da ordem das classes da escala, o que invariavelmente contradiz a natureza ordinal da escala *likert*. Além disso, $H(x)$ leva em consideração apenas a distribuição $P(x_i)$, ignorando os valores da variável aleatória x_i .

Neste contexto, [Tastle, Russell e Wiermann 2008] propõem um novo índice para complementar a interpretação dos resultados obtidos a partir dos questionários. O nome

deste índice é *Consenso* ($Cns(x)$), e é calculado pela equação abaixo, onde $E(x)$, é a esperança matemática, ou seja, “o que se espera com mais e mais replicações dos questionários”.

$$Cns(x) = 1 + \sum_{i=1}^k P(x_i) \log_2 \left(1 - \frac{|x_i - E(x)|}{X_{max} - X_{min}} \right)$$

O $Cns(x)$ deve ser interpretado como um percentual de concordância interna da distribuição no que diz respeito a $E(x)$. Como $0 \leq Cns(x) \leq 1$, entende-se que, quanto mais próximo de 1, mais os respondentes “concordam” com o valor esperado para a questão. Um conceito complementar ao *Consenso* é o de *Divergência* ($Dvg(x)$) e esta é dada simplesmente por $1 - Cns(x)$.

Todas as métricas supracitadas foram calculadas e estão apresentadas na Tabela 2. Percebe-se então que, para a maioria das questões, o consenso dos alunos é acima de 0,60. Porém, as maiores divergências ($Dvg(x) > 0,50$) foram para os itens Q5, Q6 e Q8. Esses itens referem-se ao *feedback* sobre o trabalho do estudante (Q5), adequação da carga horária (Q6) e a motivação em manter-se estudando (Q8). Inclusive, esses dois últimos tinham uma esperança matemática abaixo de 4,0, indicando como os itens mais críticos a serem revistos para as próximas edições.

Além das questões objetivas, os estudantes responderam duas questões descritivas, descrevendo pontos positivos e negativos da disciplina. Foram categorizadas as respostas e como principal ponto positivo, foram mencionados o uso de exercícios práticos e aplicáveis (Tabela 3) que é justamente a aplicação do modelo 4C/ID.

Tabela 2 – Resultados quantitativos percentuais para cada questão

Respostas	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
1	0,00	0,00	0,00	5,88	5,88	11,76	0,00	0,00	5,88	0,00	0,00
2	11,76	5,88	11,76	0,00	5,88	11,76	0,00	35,29	11,76	41,18	5,88
3	5,88	17,65	11,76	17,65	23,53	17,65	11,76	11,76	29,41	35,29	5,88
4	41,18	47,06	17,65	35,29	5,88	23,53	29,41	23,53	35,29	5,88	47,06
5	41,18	29,41	58,82	41,18	58,82	35,29	58,82	29,41	17,65	17,65	41,18
E(x)	4,12	4,00	4,24	4,06	4,06	3,59	4,47	3,47	3,47	3,00	4,24
Consenso	0,68	0,75	0,60	0,63	0,48	0,44	0,75	0,49	0,59	0,63	0,73
Divergência	0,32	0,25	0,40	0,37	0,52	0,56	0,25	0,51	0,41	0,37	0,27

Tabela 3 – Pontos positivos da avaliação descritiva

Categorias	Qtde
Conteúdo	
Reutilização/coesão/acoplamento/manutenibilidade/escalabilidade	3
Organização/Código Limpo	3
Eficiência/desempenho/qualidade	4
Utilidade	1
Metodologia	
Exercícios práticos/aplicação	9
Clareza/organização do conteúdo	3
Ler o livro de padrões	1

Tabela 4 - Pontos negativos da avaliação descritiva

Categorias	Qtde
Trabalho do semestre ser com problemas diferentes de jogos	6
Carga horária	3
Complexidade/dificuldade	3
Monitor	1
Variar estratégias de ensino	3

A Tabela 4 mostra um panorama geral das avaliações descritivas negativas. Os alunos gostariam que o trabalho do semestre fosse diferente do desenvolvimento de jogos. Por exemplo, eles gostariam que fosse desenvolvimento de uma parte de uma aplicação de problemas reais, como controle de estoques, vendas, financeiro e assim por diante.

5. Considerações finais

Após três semestres adotando e aprimorando uma metodologia de ensino de Padrões de Projeto, foi realizada uma avaliação com os alunos para verificar a percepção deles sobre a metodologia e a aprendizagem. Foi aplicado um questionário (*eVALUate*) com 11 itens em cinco níveis de concordância na escala *likert*, usando o Consenso para determinar uma medida real das respostas, e mais duas perguntas descritivas.

Nunca havia sido dito aos alunos que estávamos experimentando alguma metodologia de ensino diferenciada. Avaliando as perguntas descritivas, os alunos concordaram que as tarefas de aprendizado (um dos componentes do modelo 4C/ID), como estavam sendo oferecidas, permitiram que eles compreendessem a utilidade e aplicação de cada padrão isoladamente. Eles afirmaram que a metodologia permitiu que eles aprendessem como produzir código reutilizável, seguindo as boas práticas orientação a objetos quanto ao acoplamento e coesão. Com o objetivo de aplicarem em conjunto os padrões e praticarem a identificação de quando usar o padrão, durante o semestre os alunos tinham de implementar a versão digital de um jogo de tabuleiro. Contudo, os alunos sugeriram que o escopo do trabalho final fosse mais próximo de uma aplicação real que eles tenham contato na sua vida profissional, como o desenvolvimento de uma parte de um sistema de informação.

Para corroborar com a opinião subjetiva, os maiores consensos ($Cns(x) \cong 0,75$) foram quanto às experiências de aprendizagem e a qualidade do ensino para terem auxiliado eles a alcançarem os resultados de aprendizagem, além da satisfação geral com a disciplina.

Três itens merecem atenção para melhoria na disciplina: (1) metade dos alunos concorda em não estarem totalmente motivados para estudarem para a disciplina, porque (2) eles discordam que a carga horária da disciplina está de acordo com a quantidade de trabalho exigida na disciplina; (3) e que existe *feedback* suficiente sobre o andamento do trabalho deles. Os alunos comentaram nas questões discursivas que o trabalho do semestre é bastante complexo porque envolve além da aplicação de padrões, o domínio no Java Swing para dar suporte às interações de um jogo de tabuleiro. Além de trocar o escopo do trabalho, podem-se estabelecer alguns marcos intermediários para o acompanhamento do trabalho do semestre durante as aulas.

O assunto da disciplina é complexo e exige que o aluno venha anteriormente com conhecimentos sólidos em programação orientada a objetos. A proposta de oferecer aos alunos projetos sequencialmente cada vez mais incompletos permite a redução da carga cognitiva através da redução na quantidade de informação a ser lembrada para resolver cada um dos problemas. Qualquer disciplina em que exija a aprendizagem complexa, pode aproveitar essa estratégia, uma vez que o professor já está de posse das respostas dos seus exercícios, bastando organizá-los de forma que, a cada exercício, os alunos gradativamente recebam uma parte menor das respostas e, conseqüentemente, tenham de desenvolver soluções mais completas e complexas.

Referências

Alphonse, C.; Caspersen, M.; Decker, A. (2007) Killer “Killer Examples” For Design Patterns. In *ACM SIGCSE Bulletin*, V. 39, N. 1, pp. 228-232.

- Astrachan, O. et al. (1998) Design Patterns : An Essential Component of CS Curricula. In *ACM SIGCSE Bulletin*, V. 30, N. 1, pp. 153–160.
- Clark, L. A.; Watson, D. (1995) Constructing Validity: Basic Issues In *Objective Scale Development. Psychological Assessment*, V. 7, N. 3, pp. 309–319.
- Deville, R. F. (2003) Scale Development: Theory and Applications. 2.ed. Thousand Oaks, California: Sage Publications, Inc.
- Gamma, E. et al. (2000) Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos. Porto Alegre, Brasil: Bookman.
- Gestwicki, P. V. (2007) Computer Games as Motivation For Design Patterns. In: *ACM SIGCSE Bulletin*, V. 39, N. 1, pp. 233-237.
- Merriënboer, J. J. G. Van; Kirschner, P. A. (2018) Ten Steps To Complex Learning: A Systematic Approach To Four-Component Instructional Design. 3th. ed. Routledge.
- Oliver, B. et al. (2008) Evaluate : An Evaluation Instrument For Measuring Students' Perceptions of their Engagement and Learning Outcomes of their Engagement and Learning Outcomes. In *Assessment & Evaluation In Higher Education*, V. 33, N. 6, pp. 619–630.
- Pillay, N. (2010) Teaching Design Patterns. In *SACLA Conference*.
- Shannon, C. E. (1948) A Mathematical Theory of Communication. In *Bell System Technical Journal*, V. 27, N. 3, pp. 379–423.
- Silveira, I. F.; Silva, L. (2006) Aprendizagem de Padrões de Projeto em Ciência da Computação através de Jogos Digitais. In; *XIV Workshop Sobre Educação em Computação. XXVI Congresso da Sociedade Brasileira de Computação*.
- Tastle, W. J.; Russell, J.; Wiermann, M. J. (2008) A New Measure to Analyze Student Performance Using the Likert Scale. In *Information Systems Education Journal*, V. 6, N. 35, pp. 1–7.
- Tavakol, M.; Dennick, R. (2011) Making Sense Of Cronbach's Alpha. In *International Journal of Medical Education*, V. 2, pp. 23–55.
- Wiley, D. A. (2000) Learning Object Design and Sequencing Theory. Tese de doutorado. Brigham Young University.