

## Aplicando Padrões de Seleção no Ensino de Programação de Computadores para Estudantes do Primeiro Ano do Ensino Médio Integrado

Alexis V. de A. Leal<sup>1</sup>, Deller J. Ferreira<sup>2</sup>

<sup>1</sup>Instituto Federal de Mato Grosso – *campus* Barra do Garças (IFMT)  
Estrada de acesso a BR-158, S/N – 78.600-000  
Barra do Garças – MT – Brazil

<sup>2</sup>Instituto de Informática – Universidade Federal de Goiás (UFG)  
Caixa Postal 131 – 79-001-970 – Goiânia – GO – Brazil  
{alexisleal,deller}@inf.ufg.br

**Abstract.** *This paper describes an experience related to the use of programming patterns in the teaching of algorithms and computer programming to students the first year of high school in integrated mode. We present and discuss the preliminary results of ongoing research, which addresses the combined use of algorithmic programming patterns with concrete materials and games.*

**Resumo.** *O presente artigo descreve uma experiência relacionada ao uso de padrões de programação no ensino de algoritmos e programação de computadores, para estudantes do primeiro ano do Ensino Médio na modalidade Integrado. São apresentados e discutidos os resultados preliminares da pesquisa em andamento, que aborda a combinação do uso de padrões de programação de programação e jogos com materiais concretos.*

### 1 - Introdução

A preocupação com a falta de mão de obra em tecnologia, especialmente na área de Informática, vem chamando muito a atenção nos dias atuais, tanto por parte do governo, quanto pela iniciativa privada. Conforme o MINISTÉRIO DA EDUCAÇÃO [2010], o Governo Federal investirá R\$ 1,2 bilhão nos Institutos Federais de Educação, Ciência e Tecnologia até o ano de 2014. Parte destes recursos será destinada à formação de profissionais de informática de nível superior e médio técnico, conforme Lei 11.892/2008, a prioridade é a formação na modalidade Integrada ao Ensino Médio. Grande parte dos Institutos Federais no Brasil oferece o curso Técnico em Informática Integrado ao Ensino Médio.

Com relação aos cursos onde a base da formação técnica deve ser em programação, o objetivo inicial é o domínio dos conceitos básicos de algoritmos e programação de computadores, justamente para o prosseguimento dos estudos nos anos subsequentes. Grande parte dos estudantes encontram grandes dificuldades, sejam elas decorrentes de uma formação deficitária no Ensino Fundamental ou o total

desconhecimento do foco do curso no qual se está ingressando, já que muitos são exímios usuários de tecnologias computacionais e aliam a escolha do curso a este fator.

Para suprir a demanda dos profissionais de Tecnologia da Informação, é fundamental que se aprimore a qualidade do processo de ensino e aprendizagem nos cursos da área de computação e informática. Um dos gargalos destes cursos é o alto índice de reprovação e/ou evasão, desde o início do curso, em disciplinas de algoritmos e programação [PEREIRA JÚNIOR *et al*, 2005]. Em geral, os estudantes que ingressam nos cursos técnicos integrados são adolescentes, com idade entre 13 e 15 anos. No caso do IFMT, metade das vagas dos cursos é destinada a alunos que cursaram todos os nove anos do Ensino Fundamental em escolas públicas e a outra metade, para o chamado acesso universal [IFMT, 2011], que são as vagas destinadas a alunos que, ou não cursaram todo o Ensino Fundamental em instituições de ensino públicas ou são egressos de instituições privadas.

Neste sentido, dadas às particularidades dos estudantes em questão, faz-se necessária a busca de novas abordagens e metodologias de ensino, em contraponto ao ensino tradicional de algoritmos e programação de computadores. No chamado modo tradicional, o conteúdo de tal disciplina é apresentado de forma expositiva concentrando-se na solução de problemas. Conforme apresentado em PEREIRA JUNIOR & RAPKIEWICZ [2004], no modo tradicional, o professor apresenta o conteúdo através do chamado *portugol*, apresenta alguns exemplos e propõe exercícios para a turma. Segundo BORGES [2000], o modo tradicional não consegue facilmente motivar os alunos a se interessar pela disciplina, pois não é claro para os alunos a importância de certos conteúdos para sua formação.

Desta forma, este artigo descreve uma primeira experiência, realizada no primeiro bimestre de 2012, entre os meses de abril e junho, numa turma de primeiro ano do Ensino Médio Integrado a Informática. A experiência ocorreu no IFMT *campus* Barra do Garças, na disciplina Algoritmos e Lógica de Programação, utilizando quatro padrões *de programação* de seleção para resolução dos problemas apresentados em uma lista de exercícios.

O restante deste artigo está organizado da seguinte forma. A Seção 2 faz um estudo do uso de padrões para ensino de programação. O desenvolvimento do trabalho é apresentado na Seção 3. Na Seção 4, apresentamos a conclusão e as possibilidades de trabalhos futuros e por último, na seção 5, agradecemos aos órgãos que apoiaram esta pesquisa.

## **2. Ensino e Padrões de Programação**

Estudantes novatos em programação apresentam muitas dificuldades, tais como problemas relacionados com a capacidade de abstração, falta de habilidades para lidar com problemas e, no caso de estudantes recém-saídos do nono ano do Ensino Fundamental, problemas de aprendizagem, principalmente em conteúdos de Ciências Exatas. Dois grandes desafios para o estudante, conforme WINSLOW [1996] são aprender uma nova linguagem de programação e aprender como criar uma solução computacional que resolva um dado problema. Com relação ao primeiro problema, aprender as regras de sintaxe e semântica da linguagem torna-se relativamente

complexo, ainda mais se considerarmos que os comandos, como no caso da linguagem C++, são em inglês. Entretanto, o maior desafio para o programador iniciante, realmente é interpretar o problema, resolvê-lo algorítmicamente e codificar o algoritmo em uma linguagem de programação.

Programadores experientes conseguem, a partir do problema a ser resolvido, identificar qual estrutura usar, quais os tipos de dados envolvidos, bem como outras formas de resolver o mesmo problema usando estratégias diferentes. Resolução de novos problemas utilizando experiências prévias, adaptando as soluções para resolvê-las, é discutida por JOHNSON e SOLOWAY [1984] através do sistema PROUST, que trabalha com experiências de programação anteriores dos estudantes para o ensino da linguagem de programação Pascal. Estas experiências prévias contêm a base para os padrões de programação, que são soluções que, com frequência, aparecem na resolução de problemas computacionais. Padrões traduzem estratégias de programação criadas por especialistas. Desse modo, os alunos tiram proveito de boas práticas de programação.

Estes padrões elementares, também chamados de padrões pedagógicos de programação [PORTER e CALDER, 2003], contêm o esqueleto no qual o problema será resolvido pelo então iniciante em programação de computadores. Padrões de programação formam uma base inicial de micro soluções que podem ser adaptadas e combinadas pelos alunos durante a elaboração de seus programas.

Um padrão é um modelo que relata uma solução para um problema e fornece informações sobre o contexto em que ele deve ser aplicado [BARROS, *et al*, 2005]. Assim, o estudante iniciando seus estudos em programação, ao se deparar com determinado tipo de problema, pode direcionar sua resolução a partir dos padrões já conhecidos, sendo que o professor tem um papel fundamental neste processo, que será discutido mais adiante. CLANCY e LINN [1999] descreveram como padrões podem ser utilizados no ensino de programação, concluindo que os estudantes necessitam fazer o processo de aprender, criar e reformular padrões, tornando isso uma prática regular em seu processo de aprendizagem de programação. Desta forma, os exercícios devem ser criados pelo professor de forma a estimular o uso de padrões simples, combinações entre diferentes padrões, bem como estimular a criação de novos padrões pelo próprio estudante.

Padrões de programação são soluções para problemas computacionais recorrentes, formando blocos para o desenvolvimento de algoritmos. Padrões de programação se referem à classificação de problemas computacionais de acordo com o seu objetivo. [MULLER, 2005]. Assim, o aluno pode, através de uma classificação identificar, a partir do problema a ser solucionado, qual padrão pode ser utilizado para elaboração do algoritmo que soluciona o problema. Outro problema para o iniciante em programação é a questão da capacidade de abstração. Em HABERMAN e MULLER [2008], é proposto um método de ensino de abstração, combinado com resolução de problemas através de padrões. Neste trabalho, as autoras pesquisaram a influência da combinação de padrões com ensino baseado em tipos abstratos de dados. Sendo assim, o estudo mostrou que é possível combinar ensino baseado em padrões com outros métodos de ensino.

Outros problemas relacionados ao ensino de programação são discutidos por MULLER [2005]. Um deles é a capacidade do estudante de reconhecer similaridades entre problemas, particularmente entre o problema em questão e problemas encontrados e resolvidos previamente. Por conta disso, existe a necessidade de se categorizar, catalogar e descrever os principais padrões de forma que, quando o estudante se deparar com determinado problema, possa inferir qual padrão ou combinação de padrões de programação pode ser utilizado na resolução do mesmo.

As categorias de padrões mais citadas em artigos da área estão descritas em BERGIN [1999] com relação a padrões de seleção e em ASTRACHAN & WALLINGFORD [1998], que descreve os padrões de repetição. BERGIN [1999] se preocupa também em apresentar, além dos padrões de seleção propriamente ditos, padrões de estratégia de resolução de problemas, padrões auxiliares e padrões estilísticos. Todavia, neste estudo, foram abordados apenas quatro padrões de seleção que serão descritos posteriormente.

### **3. Desenvolvimento do trabalho**

A presente Seção demonstra os detalhes de como o trabalho foi desenvolvido. Na Seção 3.1, demonstramos o ambiente escolar, destacando como é ofertado o Ensino Médio Integrado no IFMT, o perfil dos estudantes que participaram do estudo, bem como alguns problemas encontrados no trabalho. A Seção 3.2 pontua os quatro padrões de programação abordados no trabalho. A Seção 3.3 discorre sobre o método de ensino utilizado discutindo alguns resultados obtidos.

#### **3.1 - Caracterizações do ambiente escolar**

Na modalidade Ensino Médio Integrado, os alunos do IFMT frequentam as aulas das disciplinas do núcleo comum, tais como Língua Portuguesa, Matemática, História, Geografia dentre outras no período da manhã, das 07:00 às 12:30, sendo que as disciplinas do núcleo profissionalizante são cursadas no período da tarde. Nesta modalidade, o *campus* Barra do Garças oferece os cursos de Técnico em Alimentos, Técnico em Comércio, Técnico em Controle Ambiental e Técnico em Informática.

O trabalho foi desenvolvido com alunos do primeiro ano do Ensino Médio, nos meses de abril, maior e parte de junho, o que corresponde ao primeiro bimestre de 2012. A classe, inicialmente, era composta por 39 estudantes, na faixa etária entre 13 a 15 anos, sendo que a distribuição por sexo se dá em 23 alunos do sexo masculino e 17 alunas do sexo feminino. A maioria, num total de 25 estudantes, é egressa de escolas públicas e o restante de escolas privadas. As aulas da disciplina Algoritmos e Lógica de Programação aconteceram às terças-feiras, das 14:30 às 17:30, com intervalo de 20 minutos iniciando às 16:00. A disciplina é ofertada anualmente para o primeiro ano do Ensino Médio, com 120 horas/aula de duração, sendo 30 horas/aula por bimestre letivo.

#### **3.2 - Padrões de Seleção Utilizados no Trabalho**

Para a apresentação e discussão dos padrões de programação com os alunos foi desenvolvido material instrucional baseado nos padrões de programação apresentados por BERGIN [1999]. Os padrões de programação apresentados e aplicados neste trabalho, bem como o foco do material instrucional, são apresentados a seguir.

### 3.2.1 - Seleção única

Este padrão de seleção utilizado quando existe a necessidade de testar se uma condição é apropriada dependendo da condição de teste. No exemplo abaixo, o comando  $c = a + b$  só será executado se a condição de teste for verdadeira, caso contrário o programa continuará sua execução.

```
if (a > b) {
    c = a + b;
}
```

### 3.2.2 - Escolha alternativa

Este padrão é utilizado quando se tem exatamente duas condições possíveis no teste lógico relacional. Ao contrário do padrão seleção única este padrão contém uma segunda parte, que é executada caso o primeiro teste for falso. No exemplo abaixo, caso a variável  $a$  seja menor ou igual a  $b$ , será executada a parte do *else*, que é  $c = a + b$ .

```
if (a > b) {
    c = b - a;
} else {
    c = a + b;
}
```

### 3.2.3 - Escolha sequencial

O padrão escolha sequencial é utilizado quando existem mais de duas condições possíveis no teste, desde que haja a necessidade de se fazer testes relacionais entre uma ou mais variáveis e um valor, ou entre uma ou mais variáveis com outras variáveis. O exemplo abaixo contém o trecho de um programa que verifica que tipo de triângulo é formado, dada as entradas realizadas pelo usuário. O uso do padrão escolha sequencial é indicado, pois existem mais de dois testes a serem realizados e existem testes entre variáveis.

```
if (a == 0 || b == 0 || c == 0) {
    cout << "Nao eh um triangulo ";
} else if (a > b+c || b > a+c || c > a+b) {
    cout << "Nao eh um triangulo ";
} else if (a == b && b == c) {
    cout << "Triangulo equilatero ";
} else if (a == b || b == c || a == c) {
    cout << "Triangulo isosceles ";
} else {
    cout << "Triangulo escaleno ";
}
}
```

### 3.2.3 - Série de Possibilidades

Este padrão é utilizado quando existem várias condições possíveis no teste, desde que não seja necessário algum teste relacional entre a variável de teste com algum valor ou da uma variável de teste com outra variável. O exemplo a seguir utiliza o padrão Série de Possibilidades, para fazer um programa que simula uma calculadora contendo as quatro operações básicas. Cada possibilidade possui uma instrução *break*, que interrompe a execução do comando relacionado ao padrão.

```

switch (opc) {
    case 1:
        result = a + b;
        break;
    case 2:
        result = a - b;
        break;
    case 3:
        result = a * b;
        break;
    case 4:
        result = a / b;
    default:
        cout << "Escolha invalida";
}

```

### 3.3 - Método de Ensino Utilizado

Nesta Seção, descrevemos o a aplicação do método de ensino utilizado no trabalho que resultou em um experimento bem sucedido. O experimento foi dividido em duas partes: apresentação e discussão dos padrões de programação e aplicação de uma lista de exercícios a serem codificados em uma linguagem de programação, explorando os padrões apresentados.

#### 3.3.1 - Apresentação e discussão dos padrões de programação

Primeiramente, os alunos foram apresentados aos padrões, tentando manter sempre um paralelo entre o que eles vivem diariamente. Por exemplo, o padrão da escola é não utilizar boné ou bermuda em sala de aula. Ou o padrão das carteiras do laboratório é a cor azul. Assim, é solicitado que eles apontem o que está fora dos padrões dentro de sala de aula. As apresentações dos padrões desta forma têm como objetivo deixar a aula mais leve, buscando que os exemplos sejam retirados do mundo no qual o aluno vive, não o mundo abstrato da programação de computadores.

Os alunos trabalharam em duplas nos computadores do laboratório. Conforme McDOWELL [2002], estudantes que estudam programação em pares produzem melhores programas que alunos que estudam individualmente. Parte do trabalho também foi feito em ambientes fora do laboratório de informática.

Após a discussão dos padrões, os mesmos são mostrados em forma de linguagem de programação. A linguagem de programação utilizada nas aulas foi a linguagem C++ e os exemplos foram mostrados utilizando a IDE Dev C++ versão 4.9.2.

#### 3.3.2 - Implementação da Lista de Exercícios

Após a apresentação dos padrões, foi lançada uma lista contendo 10 exercícios, com diferentes graus de dificuldade e foi disponibilizada no grupo virtual *IFMT-Informática* na rede social *Facebook*. Para cada exercício da lista, foi solicitado que, através de um comentário no cabeçalho do código de implementação, o estudante responda qual padrão está sendo utilizado na resolução do problema bem como o porquê da utilização do referido padrão. Assim, para cada exercício, o estudante poderá refletir, tanto individualmente, quanto em dupla, como se chegou à resolução do problema. Dessa forma, ele pode utilizar a resolução de um problema através de um padrão, para resolver outro problema semelhante, utilizando a mesma estrutura.

Com o objetivo de diagnosticar qual padrão os alunos tiveram mais dificuldades de compreender para a resolução da lista de exercícios, foi lançada uma enquete via internet, onde os estudantes tinham que responder, dentre os padrões utilizados em sala, baseando-se na resolução da lista de exercícios, qual foi o que o ele teve mais

dificuldades para utilizar. A enquete foi criada no *Wordpress*, com bloqueios através de *cookies* e endereço IP, para garantir que um aluno participasse mais de uma vez da enquete. Dos 39 alunos, 15 responderam a enquete on-line. O padrão considerado de maior dificuldade de compreensão foi o Série de Possibilidades, com 71,43% (11 votos). A tabela a seguir, mostra o resultado da enquete.

**Tabela 1: resultado da enquete on-line**

Padrão	Votos	Percentual
Seleção única	0	0.00%
Escolha alternativa	1	7.14%
Escolha sequencial	3	21.43%
Série de possibilidades	11	71.43%

Considerando minha experiência prévia como docente, ao basear as soluções dos problemas em padrões de programação, os estudantes conseguiram soluções mais rápidas para os problemas dados. Muitos alunos, apesar da solução do problema estar correta, confundiram alguns comandos, especialmente os de relação entre uma variável e outra, bem como uma variável e um valor (igual, atribuição, diferente, maior, maior ou igual, menor, menor ou igual).

#### 4. Conclusão e Trabalhos Futuros

A abordagem utilizando-se padrões de programação contextualizados no mundo real mostrou-se promissora, pois desta forma, o estudante consegue, através de modelos previamente estudados, resolver problemas de programação com maior eficiência com relação ao método tradicional de ensino. Interpretar o problema, buscando sua resolução através de soluções já conhecidas e testadas (padrões), parece ser uma abordagem eficaz no ensino de algoritmos para estudantes iniciantes em programação no Ensino Médio.

Trabalhos futuros serão elaborados utilizando padrões e jogos com materiais concretos em ambientes não formais para o ensino de programação de computadores. Outros trabalhos futuros poderão envolver estudos de combinação de padrões de seleção e padrões de programação de repetição.

#### 5. Agradecimentos

Agradecemos o auxílio financeiro da Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG), e a cessão do espaço para a realização da pesquisa pelo Instituto Federal de Mato Grosso (IFMT), *campus* Barra do Garças.

#### 6. Referências

ASTRACHAN, O. , WALLINGFORD E. (1998). “Loop Patterns”, <http://www.cs.duke.edu/~ola/patterns/plopd/loops.html>, Junho 2012.

- BARROS, L. N., MOTA, A. P. S., DELGADO, V. K., MATSUMOTO, P. M. (2005) “*A Tool for Programming Learning with Pedagogical Patterns*”, In: Proceedings of the 2005 OOPSLA - Workshop on Eclipse Technology eXchange, ACM Press, p. 125-129.
- BORGES, M. A. F. (2000) “*Avaliação de uma Metodologia Alternativa para a Aprendizagem de Programação*”. In: VIII Workshop de Educação em Computação – WEI 2000. Curitiba, PR, Brasil.
- BERGIN, J. (1999). “*Patterns for Selection Version 4*”, <http://csis.pace.edu/~bergin/patterns/PatternsV4.html>, Junho 2012.
- CLANCY, J. e LINN, M. C. (1999) “*Patterns and Pedagogy*”, In: Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education, ACM Press, p. 37-42.
- HABERMAN, B. e MULLER, O. (2008) “*Teaching Abstraction to Novices: Pattern-Based and ADT-Based Problem-Solving Processes*”, In: 38th ASEE/IEEE Frontiers in Education Conference, IEEE, p. F1C-7-F1C12.
- IFMT (2011). “*Exame de Seleção 2012/1 - Ensino Médio Integrado - Edital N° 61/2011*”, [http://selecao.ifmt.edu.br/download.aspx?cod\\_arquivo\\_download=3864](http://selecao.ifmt.edu.br/download.aspx?cod_arquivo_download=3864), Junho, 2012.
- JOHNSON, W. L. e SOLOWAY, E. (1985) “*PROUST: Knowledge-based program understanding*”. In: IEEE, Transactions on Software Engineering, IEEE, p. 369-380.
- MINISTÉRIO DA EDUCAÇÃO. “*Institutos Federais – Uma Conquista de Todos os Brasileiros*”, [http://portal.mec.gov.br/dmdocuments/folheto\\_setec.pdf](http://portal.mec.gov.br/dmdocuments/folheto_setec.pdf), Junho 2012.
- MULLER, O. (2005) “*Pattern Oriented Instruction and the Enhancement of Analogical Reasoning*”, In: Proceedings of the first international workshop on Computing education research (ICER), ACM Press, p. 57-67.
- MULLER, O., HABERMAN, B.,GINAT, D. (2007) “*Pattern-Oriented Instruction and its Influence on Problem Decomposition and Solution Construction*”, In: Proceedings of the 12th Annual SIGCSE - Conference on Innovation and Technology in Computer Science Education (ITiCSE'07), ACM Press, p. 151-155.
- PEREIRA JÚNIOR, J. C. R., RAPKIEWICZ, C.E. (2004) “*O Processo de Ensino-Aprendizagem de Programação: Uma Visão Crítica da Pesquisa no Brasil*”, In: WEI - Workshop de Informática na Educação 2004.
- PEREIRA JÚNIOR, J. C. R., RAPKIEWICZ, C. E., DELGADO, C., XEXEO, J. A. M. (2005). “*Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio*”. In: XIII Workshop de Educação em Computação - SBC2005, 2005, São Leopoldo.
- PORTER, R. e CALDER, P. “*Pattens in Learning to Program – An Experiment?*”(2004), In: Proceedings of the Sixth Australasian Conference on Computing Education, ACM Press, p. 241-246.
- SOLOWAY, E. (1986) “*Learning to Program = Learning to Construct Mechanisms and Explanations*”, In: Communications of the ACM, ACM Press, p. 850-858.
- WINSLOW, L. E. (1996) “*Programming Pedagogy – A Psychological Overview*”, In: ACM SIGCSE Bulletin, ACM Press, p. 17-22.