

# Análise da Motivação em um Estudo Integrado de Programação Baseado em PBL

Ayala L. Ribeiro, Roberto A. Bittencourt, Bianca L. Santana

<sup>1</sup> UEFS – Universidade Estadual de Feira de Santana  
Av. Transnordestina, s/n, Novo Horizonte  
Feira de Santana – BA, Brasil – 44036-900

{ayalaedavi,biancasantana.ls}@gmail.com, roberto@uefs.br

**Abstract.** *The motivation of students majoring in computing in programming courses is a relevant factor that may contribute to learning success. This work analyzes, through the ARCS framework, the motivation of students in an object-oriented programming integrated course that uses Problem-Based Learning (PBL). The levels of motivation found vary during the course, being higher in the beginning, decreasing during the process, but increasing at the end of the course. We conclude that the adequate design of the problems to be solved by students may positively or negatively affect the motivation, and that the levels of motivation found are affected by students' intrinsic motivation.*

**Resumo.** *A motivação de estudantes da área de computação em disciplinas de programação é um fator relevante que pode contribuir para o sucesso no aprendizado. Este trabalho analisa, através do framework ARCS, a motivação dos estudantes em um período letivo do estudo integrado de programação orientada a objetos que utiliza a abordagem de Aprendizagem Baseada em Problemas (PBL). Os níveis de motivação encontrados variam durante o decorrer da disciplina, sendo mais altos no início, reduzindo-se no decorrer do processo, mas aumentando ao final da disciplina. Conclui-se que o design adequado dos problemas trabalhados pelos estudantes pode afetar positiva ou negativamente a motivação e que os níveis de motivação encontrados são afetados pela motivação intrínseca dos estudantes.*

## 1. Introdução

Desmotivação, reprovação e evasão são problemas constantes enfrentados por cursos da área de computação, especialmente nas disciplinas de Algoritmos e Programação. Estas disciplinas costumam ter altos índices de evasão e reprovação, dificultando ou impedindo a continuidade dos alunos no curso [Bennedsen and Caspersen 2007]. Dentre possíveis fatores que contribuem para esta situação, pode-se elencar: i) falta de capacidade de abstração e raciocínio lógico para desenvolver soluções algorítmicas; ii) falta de motivação do estudante, que, muitas vezes, encara a disciplina como um grande obstáculo a ser superado; e iii) abordagem de ensino instrucionista, que pode não despertar o interesse do estudante [Jenkins 2002].

É relativamente comum introduzir programação através do paradigma imperativo e, posteriormente, apresentar o paradigma orientado a objetos nas disciplinas de programação de cursos de computação. Esta transição cria um problema

adicional pois provoca um conflito cognitivo nos aprendizes, geralmente demorado de resolver [Bittencourt et al. 2013, Jenkins 2002]. Associado aos fatores anteriores, esta organização curricular aumenta a complexidade na aquisição de habilidades de programação, gerando desmotivação e, conseqüentemente dificultando a aprendizagem e a retenção do conhecimento.

A programação orientada a objetos (POO) tornou-se, nos últimos anos, o paradigma de programação mais influente, sendo amplamente utilizada na educação e na indústria. No entanto, aprender POO não é fácil [Kölling 1999]. As dificuldades podem ser causadas pela complexidade dos conceitos a serem aprendidos em um curto período de tempo, pela complexidade intrínseca destas linguagens e dos ambientes de desenvolvimento profissionais, agravadas pelo uso de metodologias de aprendizagem centradas no professor.

A utilização de metodologias de aprendizagem ativa é uma alternativa proposta pela comunidade para atacar as dificuldades de aprendizagem de programação. É o caso, por exemplo da Aprendizagem Baseada em Problemas (PBL, do inglês, Problem-Based Learning), uma abordagem instrucional centrada no estudante onde parte importante do estudo ocorre em pequenos grupos que se reúnem para resolver problemas propostos que desencadeiam e motivam o processo de aprendizagem [Santos et al. 2007]. A aprendizagem é autodirigida, baseada na reflexão e no fomento das questões envolvidas. O uso de PBL contribui para aquisição de habilidades como autonomia, iniciativa, comunicação, pensamento crítico, resolução de problemas, trabalho em grupo, além da retenção do conhecimento e sua aplicação em diferentes contextos [Delisle 1997].

Alguns trabalhos relatam a utilização da metodologia PBL como alternativa para o ensino de programação orientada a objetos [Bittencourt et al. 2013, Angelo et al. 2014]. Mas, embora estes trabalhos relatem as experiências em detalhes, ainda não foi feita uma avaliação científica aprofundada do uso de uma abordagem como PBL no ensino de POO, especialmente levando em conta questões de motivação.

O objetivo deste trabalho é avaliar a motivação dos estudantes em uma abordagem de ensino-aprendizagem de programação orientada a objetos no curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS), no Estudo Integrado de Programação do segundo semestre do curso. A avaliação foi realizada através de uma metodologia de pesquisa quantitativa e uma abordagem de estudo de caso.

A partir do objetivo acima, procuramos responder às seguintes questões de pesquisa:

1. Como a abordagem utilizada influencia a motivação dos estudantes?
2. Quais os resultados de motivação nos diversos componentes do estudo integrado e como se comparam estes resultados?

Os resultados indicam que os níveis de satisfação são relativamente altos durante o semestre letivo. Os níveis de confiança tiveram as menores medianas. Os níveis de atenção e satisfação foram relativamente altos em três dos quatro problemas PBL aplicados no estudo integrado. O cuidado na elaboração do problema, a escolha do domínio do problema, a complexidade do problema, o excesso de conceitos e a inserção da novidades são questões que interferem diretamente na motivação dos estudantes.

## **2. Fundamentação Teórica**

Esta seção apresenta os conceitos essenciais para a compreensão deste trabalho.

### **2.1. Motivação**

Keller (1987) define motivação como aquela que explica a direção e a magnitude do comportamento, ou, em outras palavras, explica quais os objetivos que as pessoas escolhem perseguir e quão ativamente ou intensamente os perseguem. Ray (1964) descreve a motivação fazendo um exame cuidadoso da palavra (motivo) e de seu uso, e conclui que a motivação deverá fazer referência a três componentes: o comportamento de um sujeito; a condição biológica interna relacionada; e a circunstância externa relacionada.

Jenkins (2001) alega que para inspirar os estudantes através de uma instrução verdadeiramente motivacional na aprendizagem de programação, a solução é maximizar os efeitos positivos de cada um desses fatores. E para isso é necessário reconhecê-los. Ele menciona cinco tipos de motivação: extrínseca, intrínseca, social, de realização e nula. Na motivação extrínseca, o fator motivacional é a carreira e as recompensas associadas que resultarão da conclusão bem-sucedida do curso. Na motivação intrínseca, o principal fator motivacional é um interesse profundo na computação (ou especificamente em programação) para seu próprio bem-estar. Na motivação social, o fator motivacional é o desejo de agradar a um terceiro cuja opinião é importante. Na motivação de realização, o fator motivacional é “fazer bem” para a satisfação pessoal. Uma quinta categoria corresponde à “motivação nula”, que se relaciona com casos que não se encaixam nas categorias previamente descritas.

Baseando-se na importância da motivação na aprendizagem, Keller (1987) desenvolveu o modelo ARCS da motivação, que se propõe a fornecer estratégias para auxiliar a reconhecer e ajudar a resolver problemas motivacionais dos estudantes. O modelo analisa as necessidades motivacionais dos estudantes através de quatro categorias: Atenção, Relevância, Confiança e Satisfação.

### **2.2. Aprendizagem de programação orientada a objetos – POO**

O ensino de POO como segundo paradigma é muito discutido na literatura. As dificuldades podem estar associadas a aspectos como a complexidade natural destas linguagens ou a complexidade dos ambientes de desenvolvimento profissionais. Alguns autores afirmam que a introdução de paradigmas como orientação a objetos na disciplina introdutória de programação, que é ministrada no início dos cursos de computação, não fornece evidências significativas de facilitar a aprendizagem [Burton and Bruhn 2003]. Em contraposição, Kolling (1999) sugere que os conceitos de orientação a objetos devem ser ensinados desde o início, argumentando que as dificuldades estão nas ferramentas utilizadas para o ensino e não no paradigma em si.

### **2.3. Aprendizagem Baseada em Problemas – PBL**

A Aprendizagem Baseada em Problemas (PBL) é uma metodologia ativa desenvolvida por Howard Barrows na Universidade de McMaster, no Canadá [Barrows and Tamblyn 1980]. Inicialmente idealizada para a área de saúde, esta metodologia ganhou aceitação e está se tornando cada vez mais presente em uma variedade de disciplinas no ensino superior. Problemas inspirados no mundo real são o ponto de

partida para os estudantes, responsáveis pela construção de suas próprias aprendizagens. O professor atua como facilitador e é responsável pela concepção do problema a ser solucionado. O estudante tem autonomia para analisar e trilhar os possíveis caminhos em direção à solução do problema [Delisle 1997].

Em PBL, não só os estudantes são constantemente incentivados a aprender, mas também a desempenhar um papel ativo no processo de construção da aprendizagem [Cintra and Bittencourt 2015]. Um problema em PBL é um gatilho para motivar o estudo. Geralmente, segue-se um ciclo de aprendizagem, repetido enquanto durar o problema: 1) Os estudantes são apresentados a um problema antes de qualquer preparação ou estudo; 2) Os estudantes se reúnem em grupos para organizar ideias e recordar conhecimentos anteriores relacionados ao problema; 3) Através da discussão, os estudantes colocam questões, conhecidas como questões de aprendizagem, que tratam de aspectos do problema que não entendem; 4) As questões de aprendizagem são classificadas em ordem de importância, e os alunos definem metas de aprendizagem para estudo independente e em grupo.

#### **2.4. PBL na Educação em Computação**

Uma revisão sistemática com o objetivo investigar como PBL está sendo utilizada nos currículos da área de computação encontrou 63 artigos em disciplinas de vários cursos da área [O'Grady 2012]. Nestes cursos, há uma grande variedade de disciplinas que usam PBL como abordagem de ensino: engenharia de software, programação de computadores, qualidade de software e sistemas operacionais, dentre outras. Concluem constatando que a penetração do PBL nos currículos de computação ainda é superficial.

No Brasil, a utilização de PBL na área de computação ainda é muito restrita. PBL é empregada em seis componentes curriculares no curso de Engenharia de Software da Universidade Federal do Pampa (UNIPAMPA). Esses componentes integram, de modo interdisciplinar e transversal, diferentes conteúdos na abordagem de uma situação-problema que se aproxima da realidade profissional [Cheiran et al. 2017]. Outra instituição que introduziu a abordagem com PBL integrada ao currículo durante todo o curso é a Universidade Estadual de Feira de Santana (UEFS), que adotou PBL no curso de Engenharia da Computação desde a sua criação [Santos et al. 2007].

Bittencourt et al. (2013) relatam uma experiência de integração das disciplinas de programação orientada a objetos, estruturas de dados e projeto de sistemas em um estudo integrado semestral com dez horas semanais, utilizando uma metodologia de aprendizagem baseada em problemas e projetos. A experiência foi aplicada em dois semestres letivos e as principais lições aprendidas foram: a integração de conhecimentos possibilita experiências mais autênticas e práticas de produção de software mais disciplinadas; a metodologia PBL permite adquirir competências mais amplas de comunicação, trabalho em equipe e autodidatismo; problemas de manutenção de software podem reduzir a motivação por acúmulo de deficiências e devem ser evitados; a dosagem de novos conceitos nos problemas propostos deve respeitar um processo gradual e a capacidade de assimilação dos estudantes.

### **3. Metodologia**

Nesta seção são apresentados o cenário educacional, o estudo de caso e os procedimentos de coleta e análise de dados.

### 3.1. Cenário

O curso de Engenharia da Computação da Universidade Estadual de Feira de Santana (UEFS) tem adotado PBL desde sua criação em 2003. Este curso é caracterizado pela integração e interdependência entre componentes curriculares que agrupam disciplinas com conteúdos relacionados em um mesmo período letivo, compartilhando trabalhos, desafios e oportunidades de aprendizado, evidenciadas particularmente pelos componentes curriculares denominados Estudos Integrados [Bittencourt and Figueiredo 2003]. O Estudo Integrado (EI) objetiva ser um componente integrador sobre certo tema e é organizado em módulos. Durante o estudo integrado, o estudante é apresentado a certo tema ou problemas abrangentes e, para resolver os problemas, torna-se necessário adquirir novos conhecimentos, os quais são agrupados em módulos.

Os conceitos de programação estão presentes em dois estudos integrados desse curso. O EI de Algoritmos é um componente oferecido no primeiro semestre, que faz um estudo introdutório integrando as ideias de algoritmos, estruturas de dados básicas (arrays e registros) e programação estruturada em uma linguagem imperativa. No segundo semestre, o EI de Programação integra a programação orientada a objetos, algoritmos e estruturas de dados avançadas e projeto de sistemas [Bittencourt et al. 2013]. O objetivo geral do EI de Programação é que o estudante seja capaz de projetar e desenvolver software orientado a objetos, utilizando apropriadamente algoritmos e estruturas de dados, com domínio dos fundamentos subjacentes às metodologias e ferramentas utilizadas.

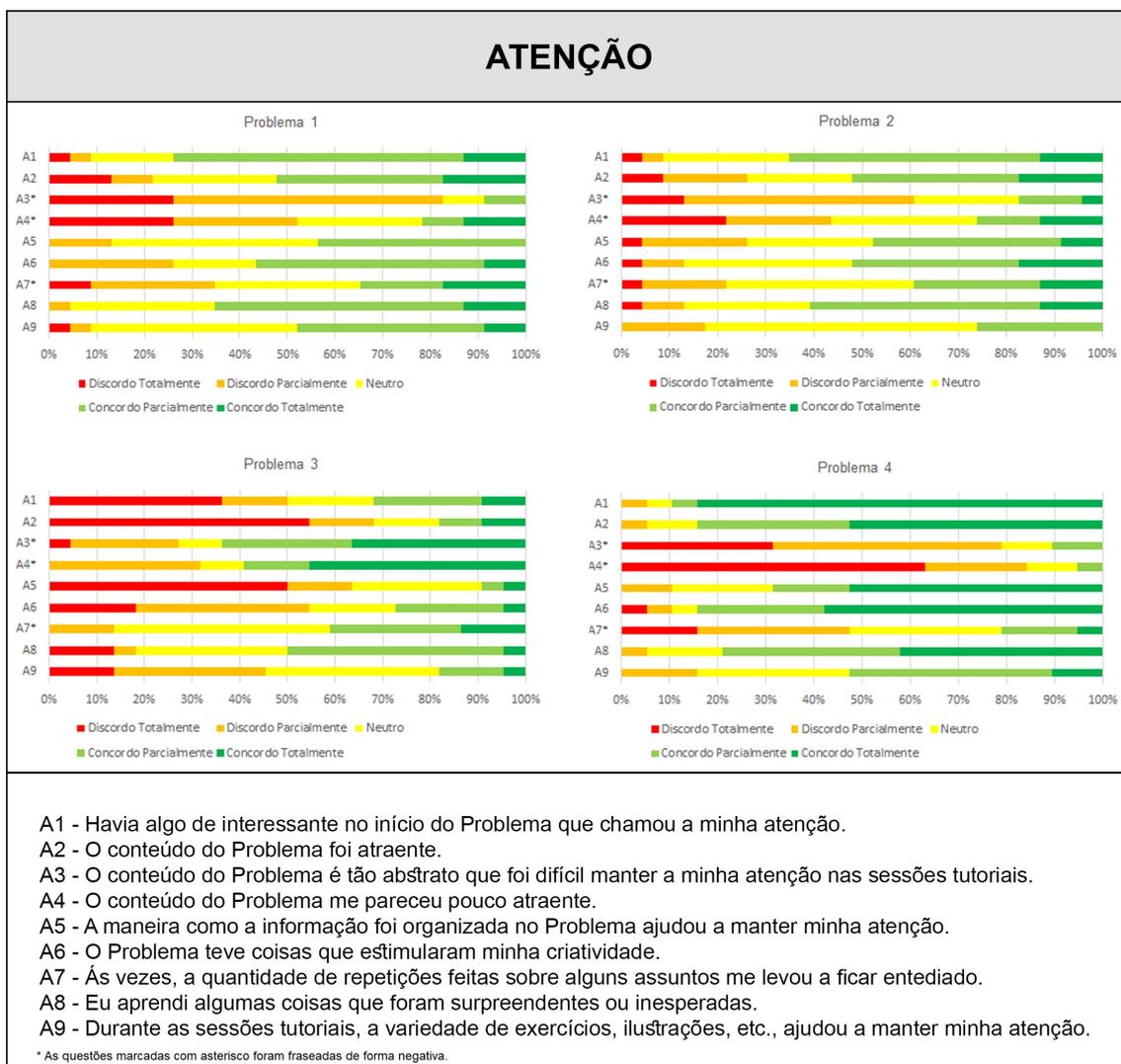
### 3.2. Estudo de caso

O estudo de caso foi conduzido ao longo do segundo semestre de 2017. Avaliamos o Estudo Integrado (EI) de Programação do curso de Engenharia da Computação da UEFS, componente curricular formado pelos módulos: Algoritmos e Programação II, Estrutura de Dados, Projeto de Sistemas e o Módulo Integrador (MI) de Programação. As turmas eram heterogêneas, compostas por 25 estudantes novatos e veteranos. Para atender às questões éticas e preservando o direito ao anonimato dos participantes da pesquisa, um Termo de Consentimento Livre e Esclarecido (TCLE) foi assinado pelos estudantes que desejaram participar. Por limitações de espaço do artigo, o planejamento do MI de Programação<sup>1</sup> e os problemas<sup>2</sup> são apresentados em outros locais.

O estudo de caso foi iniciado com a aplicação de um questionário pré-intervenção para identificar os dados demográficos dos estudantes, bem como suas impressões em relação à computação e, mais especificamente, à programação em si. No MI de Programação, onde ocorrem as sessões tutoriais de PBL, um questionário foi aplicado ao final de cada um dos quatro problemas propostos para avaliar a motivação dos estudantes e suas percepções sobre a aprendizagem; a seção de motivação do questionário replica o instrumento IMMS – Instructional Materials Motivation Survey [Keller 2010]. IMMS é um instrumento projetado para medir reações a materiais de instrução autodirigidos, embora possa ser adaptado para situações presenciais com foco nos materiais instrucionais. Das 36 questões originais do IMMS, três questões da categoria atenção explicitamente relacionadas a materiais instrucionais puramente textuais foram suprimidas. Os questionários foram analisados através de estatística descritiva e inferencial. A significância dos testes foi determinada considerando o nível de significância de 0,05.

<sup>1</sup><https://sites.google.com/site/wei2018programacao>

<sup>2</sup><http://sites.ecomp.uefs.br/mip-20172/home/tutorial>

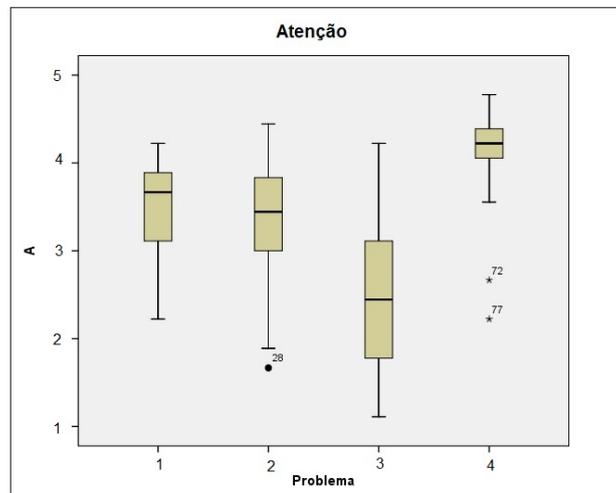


**Figura 1. Resultados para a categoria Atenção nos Problemas**

## 4. Resultados

Reunimos os resultados dos questionários IMMS para cada um dos quatro problemas propostos a partir das categorias de Atenção, Relevância, Confiança e Satisfação do modelo ARCS e os apresentamos a seguir. As respostas aos questionários são baseadas em uma escala de Likert de cinco níveis. Para facilitar a interpretação, consideramos que houve concordância quando as respostas foram Concordo Parcialmente ou Totalmente e discordância quando as respostas foram Discordo Parcialmente ou Totalmente.

Para cada categoria do ARCS, produzimos um escore a partir da soma dos valores das respostas de cada questão, convertendo a escala nominal para uma escala numérica variando de 1 (Discordo Totalmente) a 5 (Concordo Totalmente), e tratando as questões fraseadas na forma negativa com uma escala invertida de 5 a 1. Os escores foram normalizados entre 1 e 5, dividindo cada soma pelo número de questões de cada categoria. Confirmamos a normalidade de todos os escores através do teste de aderência de Kolmogorov-Smirnov.



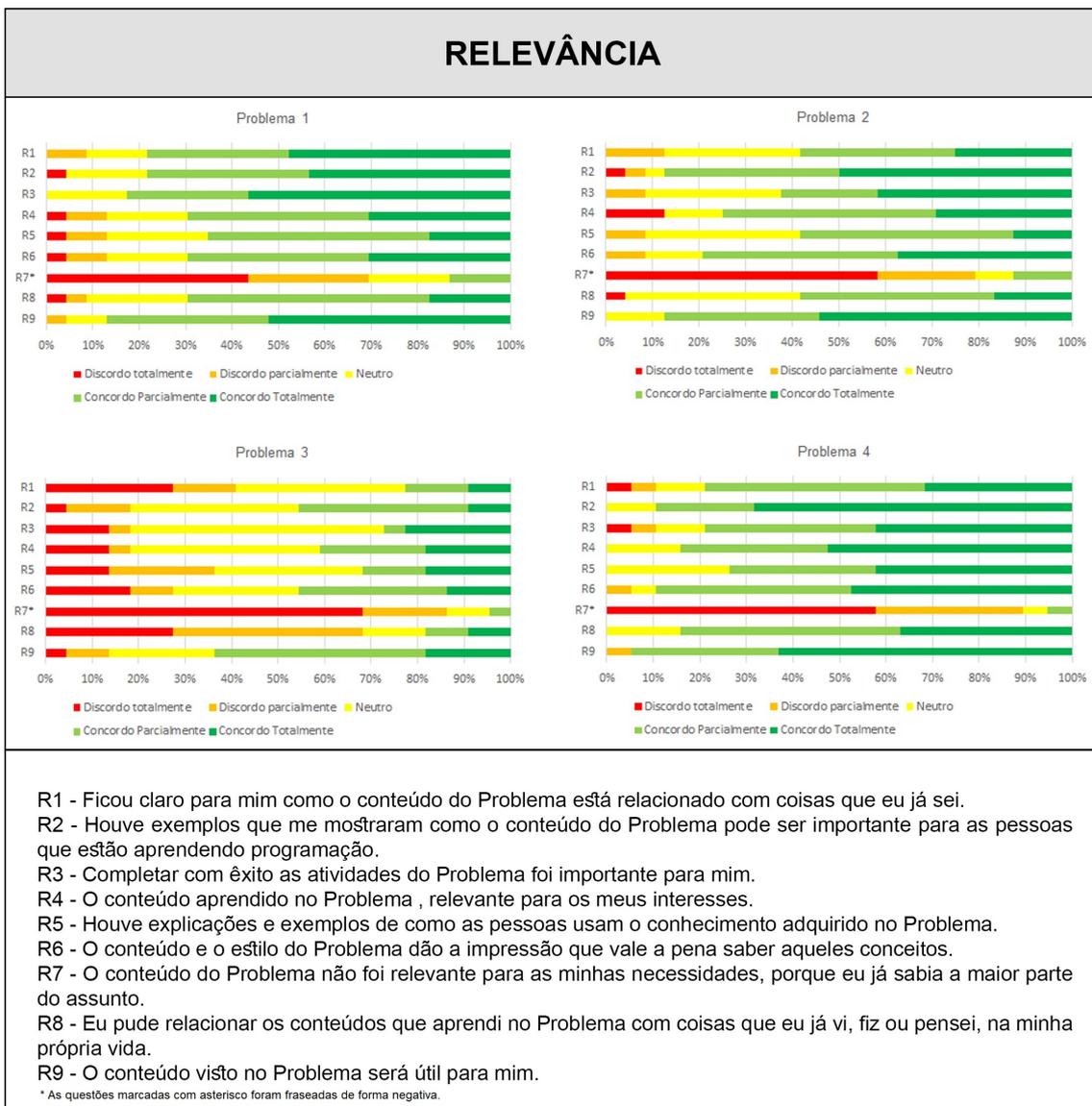
**Figura 2. Box-Plot dos escores de Atenção nos Problemas**

#### 4.1. Atenção

A Figura 1 apresenta os resultados sobre a categoria Atenção nos problemas do MI de Programação. Em todos os problemas, os estudantes afirmaram que aprenderam coisas surpreendentes ou inesperadas (A8 – concordância de 65% em P1, 65% em P2, 50% em P3, 79% em P4). A maior parte deles concordou que havia coisas interessantes que chamavam atenção (A1 – 74% em P1, 65% em P2, 89% em P4), e achou o conteúdo dos problemas atraente (A2 – 52% em P1, 52% em P2, 84% em P4). Entretanto, a maioria dos estudantes considerou o Problema 3 pouco atraente (A4 – 68%), tampouco gostou da maneira como a informação foi organizada (A5 – 65%), além de considerar que o conteúdo foi abstrato, dificultando a atenção nas sessões tutoriais (A3 – concordância de 64%). O Problema 4 obteve os melhores resultados. Uma parcela significativa de estudantes concordou que havia coisas que estimulavam a criatividade (A6 – 84%), não achou o conteúdo abstrato (A3 – discordância de 79%), e julgou que a maneira como a informação foi organizada contribuiu para manter a atenção (A5 – concordância de 69%).

A Figura 2 traz os box-plots do escore da categoria Atenção nos problemas. Observamos que a mediana da atenção obteve resultados positivos nos Problemas 1 e 2 (escores entre 3 e 4), bastante positivos no Problema 4 (escore acima de 4), e negativos no Problema 3 (entre 2 e 3). Sobre a dispersão dos dados, nos Problemas 1 e 2, notamos uma dispersão menor que a do Problema 3 e maior que a do Problema 4. Assim, os escores sofreram menos variações nos Problemas 1 e 2, o que é ainda mais evidente no Problema 4. A dispersão dos escores é maior no Problema 3.

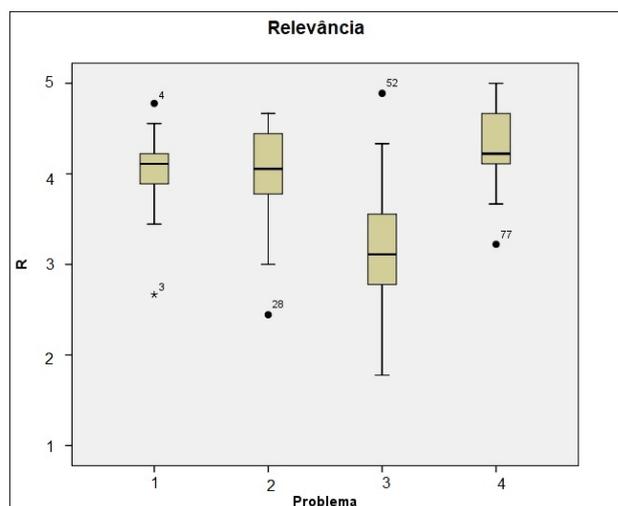
Utilizamos a análise de variância (ANOVA) para testar se o escore de Atenção varia significativamente entre os Problemas, o que foi confirmado ( $F = 18,01$ ,  $p < 0,001$ ). O resultado evidencia que a distribuição de pelo menos um dos grupos difere das demais, mas não indica entre quais grupos a diferença é significativa. Para se determinar quais pares de escores são diferentes entre si, adotamos o Teste *post-hoc* de Tukey. Houve diferenças estatisticamente significativas (valor- $p < 0,05$ ) entre os Problemas: a) 1 e 3; b) 1 e 4; c) 2 e 3; d) 2 e 4; e) 3 e 4. Os níveis de atenção foram mais altos no Problema 4, seguidos pelos Problemas 1 e 2 e mais baixos no Problema 3.



**Figura 3. Resultados para a categoria Relevância nos Problemas**

## 4.2. Relevância

Na Figura 3, a categoria Relevância nos problemas é ilustrada. Estudantes reconheceram a utilidade do conteúdo adotado em todos os problemas (R9 – concordância de 87% em P1, 88% em P2, 64% em P3, 95% em P4). Constataram a importância dos conteúdos, principalmente para aqueles que estão aprendendo a programar ( R4 – 78% em P1, 88% em P2, 89% em P4). A percepção da relevância fica menos evidente no Problema 3, pois a maioria não relacionou os conteúdos aprendidos com coisas que já foram feitas, vistas ou pensadas (R8 – discordância de 68% em P3) e considerou que o conteúdo não foi relevante para atender suas necessidades (R7 – 86% em P3). Em contrapartida, no Problema 4, a percepção da relevância é mais significativa pois uma parcela expressiva dos estudantes reconheceu os benefícios de aprender os conceitos com o conteúdo e o estilo usados (R6 – concordância de 89% em P4), e concordaram (R8 – 84% em P4) ao relacionar os conteúdos aprendidos com coisas que já foram vistas, feitas ou pensadas.



**Figura 4. Box-Plot dos escores de Relevância nos Problemas**

A Figura 4 exibe o box-plot do escore da categoria Relevância nos problemas. A mediana da relevância é maior nos problemas 1, 2 e 4, menor no problema 3, porém positiva em todos os problemas. Em relação à dispersão dos dados, observamos que nenhum dos problemas teve dispersão expressiva.

Utilizamos ANOVA para testar se a Relevância sofre mudanças significativas entre os Problemas, confirmado com  $F = 13,67, p < 0,001$ . Utilizamos o Teste *post hoc* de Tukey para identificação das diferenças específicas entre os pares de escores. O resultado demonstrou diferenças significativas entre os Problemas: a) 1 e 3; b) 2 e 3; c) 3 e 4. Os estudantes entenderam que todos os problemas foram relevantes, porém, consideraram mais relevantes os Problemas 1, 2 e 4.

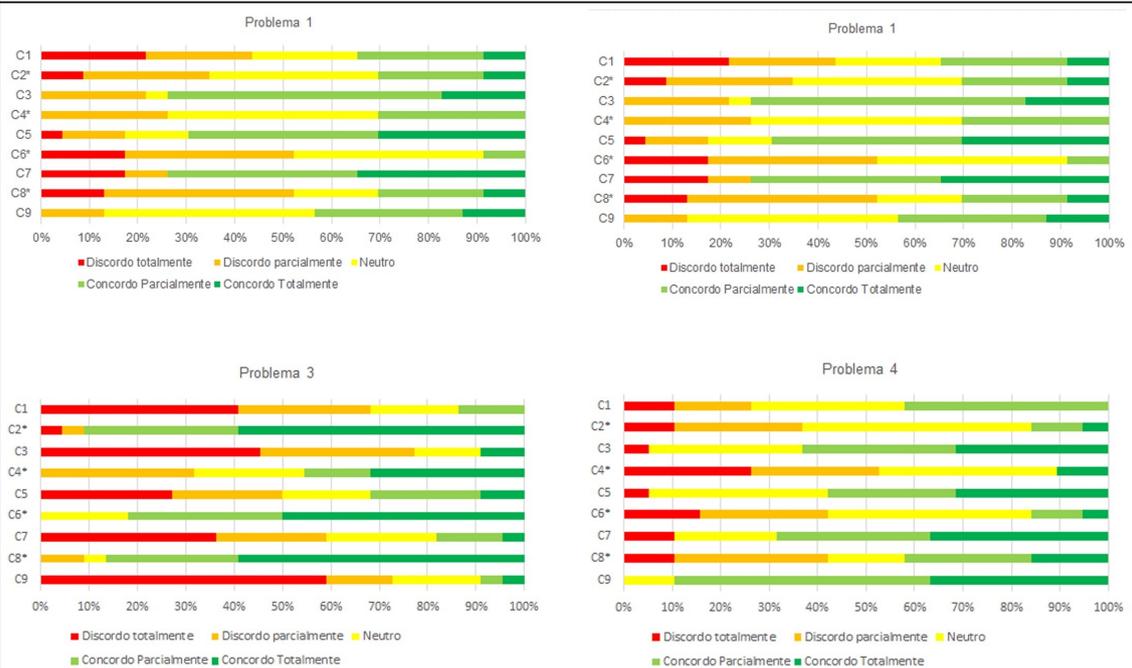
### 4.3. Confiança

A Figura 5 apresenta os resultados sobre a categoria Confiança nos problemas do MI de Programação. Os estudantes concordaram que após as primeiras sessões tutoriais sentiram-se mais confiantes do que deviam aprender (C3 – 74% em P1, 71% em P2, 63% em P4) e perceberam que eram capazes de passar na avaliação (C7 – 74% em P1, 74% em P2, 69% em P4). Observamos que o Problema 3 não despertou a confiança dos estudantes da mesma maneira que os outros problemas, visto que a maioria qualificou os assuntos como mais difíceis do que gostariam que fossem (C2 – 91%), não conseguiram compreender como algumas coisas eram feitas (C8 – 86%), bem como não se sentiram confiantes do que realmente deveriam aprender (C3 – discordância de 77%).

Na Figura 6, está representado o box-plot do escore da dimensão Confiança nos problemas. Observamos que a mediana da confiança alcançou resultados positivos nos problemas 1, 2 e 4. O problema 3 apresentou mediana negativa (abaixo de 3). A dispersão dos dados é homogênea nos Problemas 1 e 2. Os Problemas 3 e 4 apresentam dispersão maior, indicando uma maior variabilidade nos escores.

Usamos ANOVA para testar se a Confiança varia significativamente entre os Problemas, o que foi confirmado com  $F = 16,84, p < 0,001$ . O Teste *post-hoc* de Tukey revela que os resultados apresentaram diferenças significativas entre os grupos, cons-

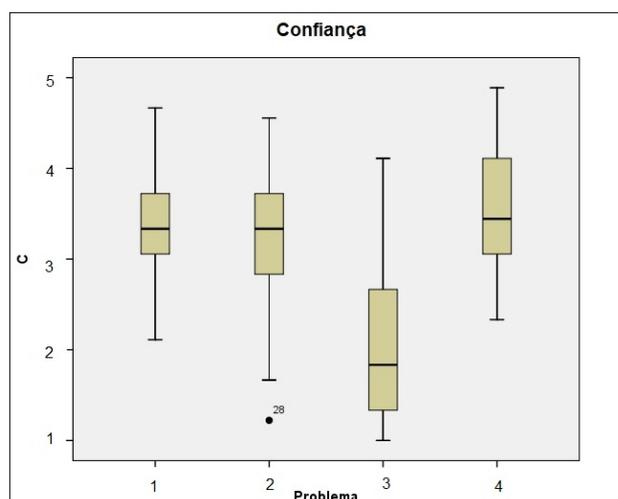
## CONFIANÇA



- C1 - Quando eu vi os assuntos do Problema pela primeira vez, eu tive a impressão de que seriam fáceis para mim.
- C2 - Os assuntos do Problema foram mais difíceis de entender do que eu gostaria que fossem.
- C3 - Depois das primeiras sessões tutoriais do Problema, senti-me confiante de que eu sabia o que eu devia aprender.
- C4 - Várias sessões do Problema tinham tanta informação que era difícil escolher e lembrar quais os pontos importantes.
- C5 - Enquanto eu participava das sessões tutoriais do Problema, eu estava confiante de que eu poderia aprender o conteúdo.
- C6 - Os exercícios no Problema foram muito difíceis.
- C7 - Após trabalhar no Problema por um tempo, senti-me confiante de que eu seria capaz de passar na avaliação deste problema.
- C8 - Eu não pude compreender como algumas coisas eram feitas no Problema.
- C9 - A boa organização do conteúdo no Problema ajudou-me a sentir confiança que eu aprenderia o assunto.

\* As questões marcadas com asterisco foram fraseadas de forma negativa.

**Figura 5. Resultados para a categoria Confiança nos Problemas**



**Figura 6. Box-Plot dos escores de Confiança nos Problemas**

tatado entre os Problemas: a) 1 e 3; b) 2 e 3; c) 3 e 4. Os estudantes permaneceram confiantes nos Problemas 1, 2 e 4, contudo não se mantiveram assim no Problema 3.

#### 4.4. Satisfação

Na Figura 7, exibimos os resultados sobre a categoria Satisfação nos Problemas do MI de Programação. A maioria dos estudantes afirmou que ao completar os exercícios sentiram-se uma sensação gratificante de realização (S1 – 74% em P1, 71% em P2, 84% em P4), e que se sentiram bem ao concluir o desafio com êxito (S5 – 61% em P1, 71% em P2, 63% em P4). Todavia, verificamos que a satisfação no Problema 3 demonstrou resultados insatisfatórios. Boa parte dos estudantes discordou que foi um prazer estudar a metodologia utilizada (S6 – 64%), e também discordou que gostava de estudar programação (S3 – 59%), assim como de saber mais sobre o conteúdo (S2 – 55%). Por outro lado, o Problema 4 se destacou no quesito satisfação, quando os estudantes relataram o desejo de saber mais sobre os conteúdos abordados (S2 – 79%), e afirmaram que realmente gostaram de estudar programação (S3 – 79%).

A Figura 8 apresenta o box-plot do escore da categoria Satisfação nos problemas. Verificamos que a mediana da satisfação obteve resultados bastante positivos nos Problemas 1, 2 e 4. O Problema 3 apresentou mediana negativa. Em relação à dispersão dos dados, nos Problemas 1, 2 e 3 verificamos uma dispersão maior que a do Problema 4. Assim, os escores sofreram mais variações nos Problemas 1, 2 e 3. A dispersão dos escores é menor no Problema 4, logo, os dados são mais homogêneos.

Utilizamos ANOVA para testar se a Satisfação difere significativamente entre os Problemas, o que foi confirmado com  $F = 9,533$ ,  $p < 0,001$ . Utilizamos o Teste *post hoc* de Tukey para identificação das diferenças específicas entre os pares de escores. O resultado demonstrou diferenças significativas entre os Problemas: a) 1 e 3; b) 2 e 3; c) 3 e 4. Os estudantes alcançaram níveis de satisfação mais altos nos Problemas 1, 2 e 4 e menores no Problema 3.

## 5. Discussão

Nesta seção, discutimos a motivação a partir da perspectiva dos problemas realizados, das categorias do modelo ARCS, além sintetizar as lições aprendidas.

### 5.1. Diferenças entre os problemas

As categorias Atenção, Confiança, Relevância e Satisfação, de modo geral, apresentaram resultados razoáveis nos Problemas 1 e 2, negativos no Problema 3, e mais positivos no Problema 4. As hipóteses que os justificam são baseadas em evidências coletadas qualitativamente a partir das nossas observações, percepções e do *feedback* dos estudantes.

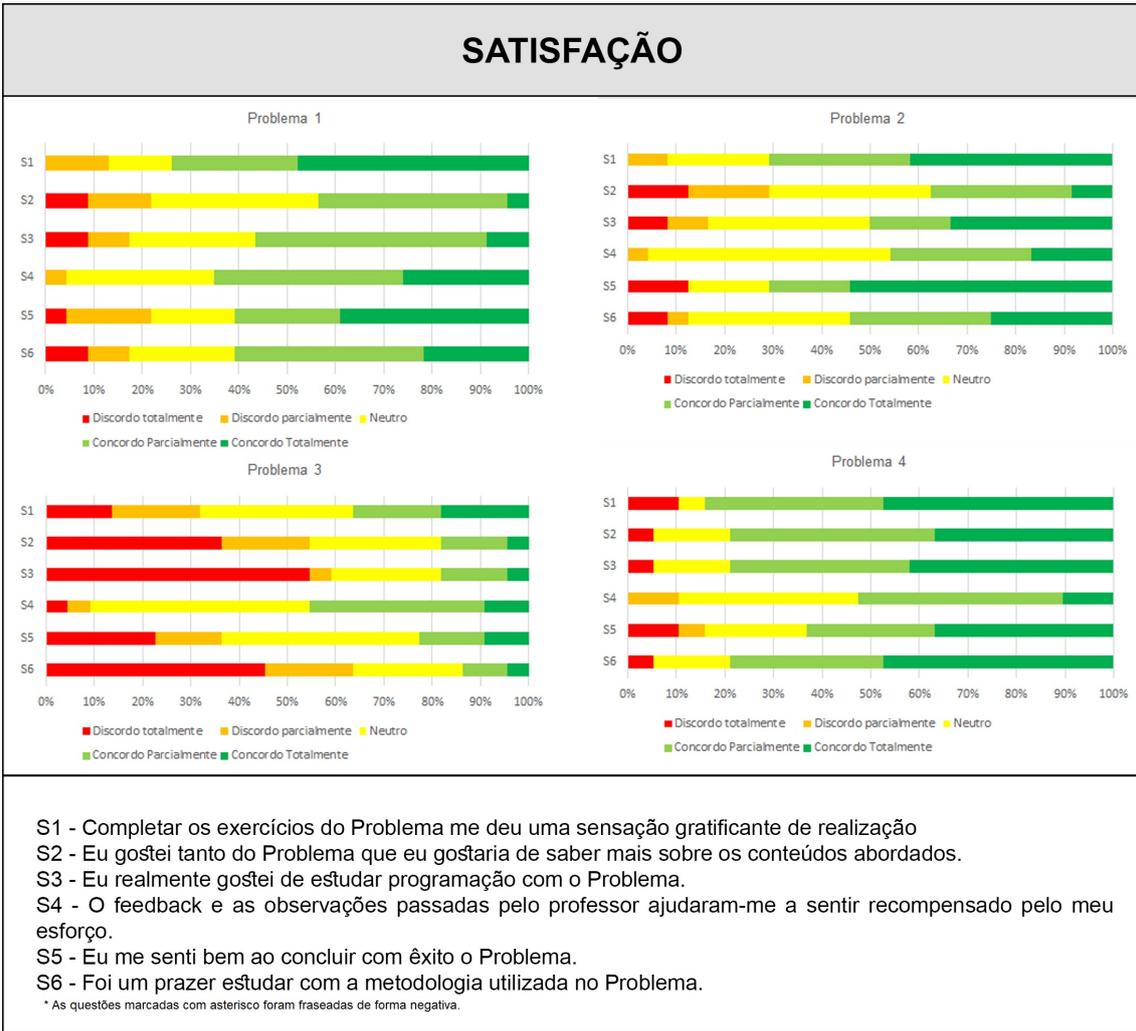
As principais características da motivação intrínseca são: a satisfação, o interesse, o desafio, a curiosidade e a novidade [Keller 1987]. Nossos resultados sugerem que a novidade influenciou em melhores níveis de Atenção, Relevância, Confiança e Satisfação nos Problemas 1, 2 e 4, e piores no Problema 3. Observamos que no Problema 1, houve a introdução de conceitos de POO, gerando uma novidade. No Problema 2, inseriu-se o conceito de testes, outra novidade considerada interessante pelos estudantes, pois agora teriam uma forma diferente de trabalhar. O Problema 4 trouxe o conceito de interface gráfica, proporcionando aos estudantes uma experiência estimulante, permitindo o *feedback* visual imediato. O Problema 3 não apresentava muitos conceitos novos, com exceção do conceito de árvore binária, os conceitos de POO eram similares e a dificuldade do problema era similar à dos anteriores.

Por outro lado, o domínio do problema, com a descrição de um cenário que apresente um contexto problematizador, escolhido a partir de um contexto real, torna o problema mais atraente e motivador. O domínio nos Problemas 1, 2 e 4 proporcionou uma identificação imediata dos estudantes. Principalmente no Problema 4, onde o domínio do problema tratava do desenvolvimento de um aplicativo similar a um já existente no mercado, que permitia o planejamento de viagens. Todavia, o domínio do Problema 3 não obteve a mesma aceitação. O domínio era desconhecido para grande maioria dos estudantes, tratando de um sistema gerenciador de carteira de ações. A maioria dos estudantes não considerou o problema relevante, gerando uma grande dificuldade motivacional.

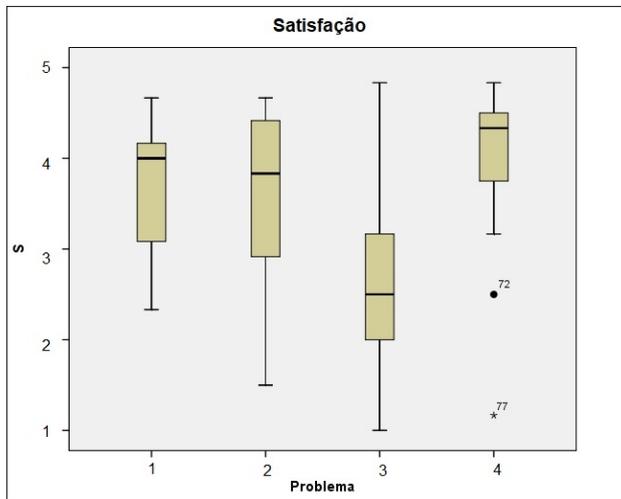
Finalmente, o problema deve ter a complexidade ideal. O problema não pode ser complexo demais, que impeça o entendimento dos conceitos, nem simples demais que impossibilite a reflexão e a discussão acerca do que deve ser aprendido. O problema deve ter a clareza e o tamanho necessários para incentivar os estudantes no desenvolvimento da solução do problema. As observações sugerem que a complexidade existente no Problema 3 influenciou nos baixos índices motivacionais em todas as dimensões do ARCS. Os estudantes consideraram o problema abstrato e confuso. O Problema 4 obteve a melhor avaliação dentre os problemas. Os estudantes acharam-no bem estruturado, com informações claras e objetivo bem definido, além de o considerarem mais intuitivo, por fazer alusão a uma aplicação existente.

### 5.2. Diferenças entre as dimensões do Modelo ARCS

Em respeito à Atenção, Keller (1987) argumenta que mesmo os melhores programas não obterão resultados positivos caso os estudantes não estejam motivados para aprender. Para isso, é necessário estabelecer um equilíbrio nas atividades do aprendiz que permita manter sua atenção. Portanto, é importante que se utilizem estratégias que incluam a variação de



**Figura 7. Resultados para a categoria Satisfação nos Problemas**



**Figura 8. Box-Plot dos escores de Satisfação nos Problemas**

ritmo ou estilo do material pedagógico, o uso do humor ou o envolvimento do estudante nas atividades. Os resultados sugerem que a atenção tem níveis relativamente altos nos Problemas 1, 2 e 4, e baixos no Problema 3.

Keller (1987) afirma que a Relevância consiste em fazer os estudantes perceberem a importância do que está sendo ensinado, e a utilização imediata destes ensinamentos devem fornecer respostas aos seus motivos e valores. As concepções de cada estudante certamente determinam os rumos por cujo intermédio seus objetivos podem ser alcançados. O fato de constatar relevância no assunto ensinado traz para o estudante a confiança de que ele está no caminho certo e que deve continuar a lutar para alcançar seus objetivos. A percepção da relevância foi relativamente alta nos Problemas 1, 2 e 4, e mais baixa no Problema 3. Porém, ainda sendo baixa no Problema 3, foi mais alta quando comparada às outras dimensões. Uma possível explicação para este grau de relevância é que os estudantes fazem uma disciplina de programação em um curso de computação, que é central neste curso e muito relacionada com a realidade profissional deles.

Na dimensão Confiança, foram obtidos os menores níveis em todos os problemas, quando comparadas às outras dimensões. Nos Problemas 1, 2 e 4, os resultados são mais neutros, e mais baixos no Problema 3. As prováveis justificativas seriam a complexidade dos conteúdos, o excesso de conceitos e as diversas competências que devem ser adquiridas para solucionar os problemas. Os estudantes enfrentam dificuldades ao lidar com a novidade dos conceitos de POO, Estruturas de Dados e Projeto de Sistemas.

Na dimensão Satisfação, foram alcançados níveis altos nos Problemas 1 e 2, baixos no Problema 3, e mais altos no Problema 4. A satisfação, segundo Keller (1987), é o resultado da avaliação cognitiva dos estudantes da equidade entre o esforço investido e os resultados percebidos ao final do processo de aprendizagem, a partir da interação com um dado objeto educacional. Deste modo, os resultados com dispersão ampla sugerem que nem todos vivenciam o sucesso ou a motivação da mesma maneira, nem gostam igualmente das suas experiências. Porém, os estudantes avaliaram que vale a pena continuar investindo seu esforço, confirmado nos Problemas 1, 2 e 4.

### 5.3. Lições aprendidas

O **cuidado na elaboração dos problemas** é relevante para a motivação dos estudantes. A qualidade dos problemas PBL está entre os maiores desafios, pois está diretamente vinculada ao aprendizado. O excesso de conceitos, a escolha do domínio do problema, a inserção de novidades e a complexidade exigida são aspectos que devem ser considerados.

O **excesso de conceitos** pode interferir no entendimento e no desenvolvimento da solução do problema. Bittencourt et al. (2013) mencionam haver uma grande variedade de conceitos que eram difíceis para os estudantes adquirirem. Eles afirmam que a dosagem de novos conceitos nos problemas propostos deve respeitar um processo gradual e a capacidade de assimilação dos estudantes.

A **escolha do domínio do problema** é relevante e deve introduzir fundamentos próximos à realidade dos estudantes, sendo uma estratégia para tornar o problema mais atraente e motivador. Um problema deve motivar os estudantes, principalmente ao inserir elementos próximos de suas realidades [Angelo et al. 2014]. E as questões iniciais dos problemas devem ser abertas, baseadas em conhecimentos prévios e/ou controversas, de forma a proporcionar discussão entre os aprendizes.

A **complexidade do problema** deve garantir que, com cooperação, os estudantes consigam solucionar o problema. Angelo et al. (2014) reiteram que os problemas devem ser complexos o bastante para que seja necessária a cooperação de todos os membros em sua solução. Ao mesmo tempo, descobrimos, neste trabalho, que a complexidade excessiva, como foi no problema 3, pode reduzir a motivação dos estudantes. Assim, é preciso encontrar um compromisso entre complexidade e motivação.

A **inserção de novidades** é essencial para manter os níveis de motivação. Keller (1987) afirma que não importa o quão bem sucedidas sejam algumas das estratégias de ensino, elas não permanecerão assim para sempre. Um erro cometido por professores e instrutores é que quando eles encontram uma estratégia que é altamente bem-sucedida, eles se exaltam e tendem a abusar dela. Cada nova estratégia tem um efeito inovador junto com qualquer nível mais profundo de conexão motivacional. O simples fato de ser novo pode estimular uma certa quantidade de interesse. Mas quando a novidade passar, a estratégia continuará a ser motivadora apenas se tiver uma conexão substancial e significativa com as exigências motivacionais dos alunos. Mesmo assim, eles se cansarão disso, porque o desejo de novidade é um aspecto da motivação humana.

## 6. Conclusões

Este trabalho apresentou um estudo de caso de uma abordagem baseada em PBL para ensino e aprendizagem de programação orientada a objetos durante um semestre letivo de um curso de Engenharia da Computação, onde foram avaliados os níveis de motivação dos estudantes utilizando o Modelo ARCS.

Os resultados apontam que os níveis de relevância foram relativamente altos em todos os problemas, possivelmente explicados por tratar-se de estudantes da área de TI. Os níveis de confiança foram um pouco mais baixos pelas prováveis dificuldades que os estudantes têm com os conceitos de programação orientada a objetos e as diversas competências a serem adquiridas. Os níveis de satisfação e atenção foram relativamente altos em três dos quatro problemas. O domínio dos Problemas 1, 2 e 4 foi considerado relevante pois estava mais próximo da realidade dos estudantes. O Problema 3 obteve os piores resultados e uma das possíveis explicações é a complexidade elevada e o domínio desconhecido. O Problema 4 alcançou os melhores resultados em todas as dimensões.

O cuidado na elaboração do problema é uma das etapas mais importantes em PBL. O domínio do problema, a complexidade dos conteúdos, o excesso de conceitos e introdução de novidades são questões essenciais. O domínio escolhido deve partir de um contexto real, que faça parte da vida dos estudantes, para que proporcione uma identificação, e assim, motive-os a prosseguir no desenvolvimento da atividade investigativa. A complexidade dos conteúdos e a quantidade de conceitos devem ter o tamanho ideal. A introdução de novidades é fundamental para a manutenção da motivação.

Em trabalhos futuros, faremos uma avaliação qualitativa dos dados que foram coletados das observações e entrevistas. Pretendemos analisar também a motivação nos módulos teóricos de Estruturas de Dados, Programação Orientada a Objetos e Projeto de Sistemas, que compõem o estudo integrado de Programação. Pretendemos também mensurar a aprendizagem nesta abordagem. Finalmente, a partir das lições aprendidas e da análise de dados realizada, devemos replicar este estudo de caso em outra turma do mesmo curso.

## Referências

- Angelo, M. F., Loula, A. C., Bertoni, F. C., and Santos, J. A. M. (2014). Aplicação e Avaliação do Método PBL em um Componente Curricular Integrado de Programação de Computadores. *Revista de Ensino de Engenharia*, 33(2):31–43.
- Barrows, H. and Tamblyn, R. (1980). *Problem based-learning: An approach to medical education*, volume 1. Springer Publishing Company.
- Bennedsen, J. and Caspersen, M. E. (2007). Failure rates in introductory programming. *SIGCSE Bull.*, 39(2):32–36.
- Bittencourt, R. A. and Figueiredo, O. A. (2003). O Currículo do Curso de Engenharia de Computação da UEFS: Flexibilização e Integração Curricular. In *Anais do XXIII Congresso da SBC*, pages 171—182, Campinas, São Paulo. SBC.
- Bittencourt, R. A., Rodrigues, C. A., and Cruz, D. S. S. (2013). Uma Experiência Integrada de Programação Orientada a Objetos, Estruturas de Dados e Projeto de Sistemas com PBL. In *XXXIII Congresso da SBC – XXI WEI*.
- Burton, P. J. and Bruhn, R. E. (2003). Teaching Programming in the OOP Era. *ACM SIGCSE Bulletin*, 35(2):111–114.
- Cheiran, J. F. P., de M Rodrigues, E., de S Carvalho, E. L., and da Silva, J. P. S. (2017). Problem-based learning to align theory and practice in software testing teaching. In *Proceedings of the 31st Brazilian Symposium on Software Engineering*.
- Cintra, C. S. and Bittencourt, R. A. (2015). Being a PBL Teacher in Computer Engineering : An Interpretative Phenomenological Analysis. In *FIE Conference*. IEEE.
- Delisle, R. (1997). *How to use problem-based learning in the classroom*. AscD.
- Jenkins, T. (2001). The motivation of students of programming. *SIGCSE Bull.*, 33(3):53–56.
- Jenkins, T. (2002). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, pages 53–58.
- Keller, J. M. (1987). Development and Use of the ARCS Model of Instructional Design. *Journal of instructional development*, 10(3):2–10.
- Keller, J. M. (2010). *Motivational Design for Learning and Performance: The ARCS Model Approach*. Springer US.
- Kölling, M. (1999). The problem of teaching object-oriented programming, part 1: Languages. *Journal of Object-Oriented Programming*, 11(8):8–15.
- O’Grady, M. J. (2012). Practical problem-based learning in computing education. *Trans. Comput. Educ.*, 12(3):10:1–10:16.
- Ray, W. S. (1964). *The Science of psychology: an introduction*. Macmillan.
- Santos, D. M. B., Pinto, G. R. P. R., Sena, C. P. P., Bertoni, F. C., and Bittencourt, R. A. (2007). Aplicação do Método de Aprendizagem Baseada em Problemas no Curso de Engenharia de Computação da Universidade Estadual de Feira de Santana. In *XXXV Congresso Brasileiro de Educação em Engenharia*.