

# Avaliação de problemas PBL em um componente curricular de um curso de Engenharia de Computação

Gabriel Antonio Pereira dos Santos Carneiro, David Moises Barreto dos Santos

Departamento de Ciências Exatas (DEXA) – Universidade Estadual de Feira de Santana  
Feira de Santana, Bahia - Brasil

[gabril4el@gmail.com](mailto:gabril4el@gmail.com), davidmbs@uefs.br

**Abstract.** *The problem itself used in a Problem-based Learning (PBL) method is crucial to its success. Therefore, this paper describes the construction of an analysis of PBL problems applied to a semester curriculum component-Integrator Module Programming at University Estadual de Feira de Santana. The main contributions of this work are to structure a review of problems, based on characteristics previously raised, and generating a ranking of them.*

**Resumo.** *O problema em si usado no método Problem-based Learning (PBL) é crucial para o seu êxito. Portanto, este trabalho descreve a análise de problemas PBL aplicados em um semestre no componente curricular Módulo-Integrador de Programação em curso de Engenharia de Computação. As principais contribuições deste trabalho são a estruturação de uma avaliação de problemas, baseada em características levantadas anteriormente, e a geração de um ranking deles.*

## 1. Introdução

No contexto histórico atual, a boa (in)formação de profissionais é essencial para a evolução tecnológica da sociedade. Logo, o processo de ensino-aprendizagem destes profissionais é alvo de grandes discussões a fim de que estes se preparem adequadamente, sobretudo, para situações reais do mercado de trabalho. Neste sentido destaca-se o *Problem Based-Learning* (PBL), também conhecido como Aprendizagem Baseada em Problemas (ABP). O PBL é um método de ensino caracterizado pelo uso de problemas da vida real, ou baseados na vida real, para estimular o desenvolvimento do pensamento crítico, habilidade na resolução de problemas e absorção de conceitos da área de conhecimento em questão [Ribeiro 2005, 32]. Por isso, enquanto o processo convencional de ensino-aprendizagem baseia-se no professor como principal agente, o PBL procura colocar o aluno como principal responsável pelo seu ensino-aprendizado na medida em que incentiva-o a discutir as soluções de tais problemas em grupos. [Pereira et al, 2007].

Santos e Angelo (2009) revela que “um dos aspectos mais importantes para o sucesso da metodologia PBL são os problemas apresentados aos alunos”, (p. 1) já que serão através deles que os professores irão conduzir a linha de aprendizagem a qual os alunos deverão seguir. Um problema mal preparado poderia prejudicar o aprendizado dos alunos ao invés de ajudá-los. Porém, ainda não existe um modelo padrão de

construção de problemas ou mesmo uma ferramenta padrão de avaliação. Ademais, a literatura não é rica em trabalhos que avaliem problemas e, quando ocorre, geralmente não estão relacionados com cursos de Computação ou mesmo de Engenharia [Hung, 2006; Marchais, 1999; Munshi et al, 2008].

Neste contexto, este trabalho tem como objetivo avaliar cinco problemas PBL aplicados em um componente curricular — Módulo Integrado (MI) de Programação — do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS). A análise foi realizada por um educando que cursou o MI – Programação, que é um dos autores deste artigo. Para tanto, tomamos como base o trabalho de Schmidt e Sockalingam (2011) no qual são apresentadas onze características para um problema PBL de acordo com a opinião de discentes. Em outras palavras, foram avaliados cinco problemas de acordo com estas onze características.

O principal resultado deste trabalho é um ranking dos cinco problemas analisados. Mais tarde professores do MI-Programação ou de componentes similares poderão utilizar os apontamentos aqui observados para construir problemas PBL com mais qualidade.

Para um melhor entendimento deste trabalho, organizamos ele como disposto a seguir. Na Seção 2, descrevemos o componente MI – Programação no qual os problemas foram aplicados enquanto na Seção 3 descrevemos as características que usamos como base para a construção da análise. Na Seção 4 está a análise e na última Seção encontra-se as considerações finais do trabalho.

## **2. MI – Programação**

O curso de Engenharia de Computação da UEFS está em funcionamento desde 2003. Em sua grade curricular atual, existem os Módulos-Integrados(MI), que consistem em um tutorial para integrar os conteúdos de um conjunto de componentes curriculares. Cada tutorial tem um tutor, o professor responsável pelo acompanhamento do grupo que, geralmente, é composto por 10 educandos.

O MI – Programação é ofertado no segundo semestre do curso e agrega os conteúdos dos seguintes componentes: Estrutura de Dados, Algoritmos e Programação II, Estrutura Discretas e Projeto de Sistemas. As ementas e carga-horária das disciplinas podem ser observadas na Tabela 1. Todas estes componentes possuem como pré-requisitos Algoritmos e Programação I e MI – Algoritmos, que visam a introdução básica à algoritmos e programação. Ademais, havia duas sessões tutoriais por semana.

A avaliação realizada aqui neste trabalho diz respeito aos problemas aplicados no semestre 2013.1. Por limitação de espaço, não será possível disponibilizar qualquer um dos problemas usados, porém eles podem ser encontrados na seguinte página: <http://sites.ecomp.uefs.br/mip-20131/home/tutorial>. A solução de todos eles deveria ser individual e estar em forma de um código-fonte (Java). Em alguns problemas também foi solicitado a entrega de um relatório individual para descrição do processo de criação e também descrição da solução. Em média, a duração do desenvolvimento da solução para cada problema era estipulada em quatro semanas.

Tabela 1. Ementa das disciplinas do MI - Programação

Disciplina	Ementa	Carga-Horária
<b>MI – Algoritmos</b>	Estudo aprofundado de metodologias de programação, modelos de algoritmos e estruturas de dados avançadas que compõem os programas de computador, além das estruturas matemáticas discretas subjacentes aos modelos estudados.	60 h
<b>Algoritmos e Programação II</b>	Princípios de orientação a objetos: objetos, classes, atributos, métodos, mensagens, encapsulamento, herança, composição, polimorfismo, interfaces, acoplamento, ligação dinâmica. Interfaces gráficas e programação orientada a eventos. Tratamento de exceções. Princípios de testes e depuração de programas. Paradigmas de linguagem de programação	30 h
<b>Estrutura de Dados</b>	Tipos abstratos de dados; independência entre especificação e implementação. Estruturas de dados básicas: tabelas, listas simples e encadeadas, pilhas, filas. Filas com prioridades. Árvores: terminologia e implementação. Árvores binárias, balanceamento de árvores binárias, árvores B. Conjuntos: conceituação, implementação, métodos de representação. Métodos de busca e ordenação. Hashing. Grafos orientados e não orientados	30 h
<b>Projeto de Sistemas</b>	Realização de requisitos e modelos conceituais através de padrões GRASP. Diagramas essenciais UML para projeto de software. Design patterns essenciais. Noções básicas de projeto arquitetural. Mapeamento de projeto para código. Testes de aceitação e descrição das funcionalidades gerais do sistema. Teste de Unidade.	30 h
<b>Estruturas Discretas</b>	Princípios de contagem, combinatória. Grafos e árvores: grafos não-orientados e orientados, árvores, árvores geradoras.	60 h

### 3. Características de problemas PBL

Como dito anteriormente, a base teórica para nossa análise está no trabalho de Schmidt e Sockalingam (2011), cujo objetivo geral é identificar características mais marcantes dos problemas no PBL segundo as percepções de alunos. Nessa seção descreveremos sucintamente as características levantadas pelos autores e o processo de como fizeram isso.

Para ranquear as características foram usadas reflexões dos alunos sobre o que eles consideravam como características de bons problemas. Os alunos respondiam a seguinte questão: “Qual é a sua percepção de um bom problema provocador para você e por quê? Você pode responder baseado em qualquer um dos problemas que você tem feito até agora”.

De posse das respostas foi usado um software de reconhecimento de palavra, o TexSTAT, que "gera uma lista de todas as palavras usadas em um documento e conta a frequência das palavras" (Schmidt et al, 2011). Agrupou-se as palavras de acordo com a semelhança semântica e identificou-se onze características a partir disso. Quanto maior a frequência das palavras, mais importante a característica. A Tabela 2 exhibe a relação entre palavras e características, a porcentagem da frequência das palavras usadas nas respostas e o ranking obtido a partir desta frequência.

Abaixo encontra-se uma descrição de que consiste cada uma das características:

1. Levar a objetivos de aprendizagem: todo problema PBL procura levar aos alunos mais conhecimento sobre determinados conteúdos, esses conteúdos são chamados de objetivos de aprendizagem;
2. Desencadear interesse: os problemas PBL têm que motivar os educandos a encontrarem uma solução. A situação-problema exposta e o grau de complexidade da solução estão ligados a este critério;

Tabela 2. Ranking de características segundo Schmidt e Sockalingam (2011)

O problema deve..	Palavras usadas pelos estudantes	Frequência das palavras	Ranking de importância
Levar a objetivos de aprendizagem	Aprender, questões e fatos	23.8%	1
Desencadear interesse	Interessante, gostar e capturar	11.5%	2
Ser do formato adequado	frase, sentença e imagem	10.9%	3
Estimula o raciocínio crítico	Pensamentos, ideias e lógica	10.2%	4
Promover aprendizado autodirigido	Pesquisar, explorar, equipamento	10%	5
Ser de clareza adequada	Obvio, claro, entender	7.3%	6
Ser de dificuldade apropriada	Fácil, dificuldade, difícil	7.1%	7
Permitir aplicação ou uso	Aplicar, mundo e uso	7%	8
Usar conhecimento prévio	Saber, lembrar, experiência	6.7%	9
Estimular elaboração	Elaborar, discussão, ideia genial	3.6%	10
Promover trabalho em equipe	Equipe, classe, conjunto	1.9%	11

3. Ser do formato adequado: este item está relacionado a apresentação do problema, textos, imagens e qualquer outro elemento que o compõe;
4. Estimular o raciocínio crítico: o problema deve ter um grau de dificuldade adequado, nem fácil demais, porque não é exercício, nem difícil demais, sob pena de desestimular os alunos a terem ideias e à julgá-las úteis;
5. Promover aprendizado autodirigido: deve levar o aluno a construir conhecimento fazendo com que ele seja o principal responsável pelo seu processo de ensino-aprendizagem;
6. Ser de clareza adequada: deve ser de fácil entendimento de modo a não gerar demasiadamente dúvidas não relacionadas ao processo de ensino-aprendizagem;
7. Ser de dificuldade apropriada: a dificuldade deve ser dosada de forma equilibrada;
8. Permitir aplicação ou uso: deve permitir que o conhecimento construído seja útil na futura profissão;

9. Usar conhecimento prévio: utilizar conceitos que os alunos conheceram em situações de aprendizado no passado;
10. Estimular elaboração: O problema deve fazer com que o aluno tenha ideias e hipóteses;
11. Promover trabalho em equipe: deve fazer com que os alunos dos tutoriais trabalhem em equipe.

É importante ressaltar que estas características estão interligadas entre si, uma influenciando e/ou impactando na outra. Para ilustrar, a complexidade (critério 7) incide diretamente no estímulo do raciocínio crítico (critério 4) e vice-versa. Além disso, vale mencionar que a importância de cada critério pode variar de acordo a experiência do educando no método PBL e até mesmo de instituição para instituição.

#### 4. Resultados e discussões

Buscando esquematizar a avaliação, atribuímos às características indicadas na seção anterior, critérios de avaliação da adequação do problema, os quais variavam de 1 a 5, onde 1 e 2 indicam baixa adequação, 3, uma adequação regular e 4 e 5, uma alta adequação. Como se pode notar, temos duas escalas para cada extremo — baixo, 1 e 2 e alta, 4 e 5 —, mas apenas para indicar um maior ou menor grau de adequação dentro destas escalas. O resultado está na Tabela 3.

Tabela 3. Escala atribuída a cada uma das características para cada problema

Características	P1	P2	P3	P4	P5
1. Leva para as questões de aprendizado	5	5	5	5	5
2. Desencadear interesse	3	3	4	4	5
3. Ser de formato adequado	4	2	3	4	4
4. Estimular o raciocínio crítico	5	5	5	5	5
5. Promover a aprendizagem auto-dirigida	3	4	4	4	4
6. Ser de clareza adequada	4	2	3	5	5
7. Ser de dificuldade apropriado	4	4	4	3	4
8. Permitir a aplicação ou uso	2	3	4	4	4
9. Referem-se a um conhecimento prévio	5	5	5	5	5
10. Estimular a elaboração	5	5	5	5	5
11. Promover o trabalho em equipe	5	5	5	5	5

Todos os problemas conseguiram cumprir bem a sua primeira função de levar os alunos aos objetivos de aprendizagem. Na segunda característica, desencadear interesse, dois problemas tiveram adequação regular (problemas 1 e 2), enquanto o restante, alta adequação. No problema 1 havia uma certa insegurança, pois não se sabia ao certo o que estava sendo feito. Esperava-se algo parecido com o que era solicitado no componente curricular anterior (pré-requisito), de introdução a algoritmos e programação. Um fator que pode ter influenciado essa nota foi a falta de conhecimento sobre a orientação a

objetos e testes de unidade. Já no problema 2 a pontuação se deve principalmente ao texto que era confuso, oprimindo o interesse de resolução por conta dos alunos. O último problema, em especial, desperta o interesse por se tratar de um software mais realista, exigindo interface gráfica e problema do caixa viajante.

Na terceira característica encontra-se uma falha grave: quatro (2, 3, 4 e 5) dos cinco problemas não possuíam referências para materiais ideais para consulta. O problema 2 teve pior avaliação nesse quesito, principalmente, devido ao excesso de informações trazidos pelo texto. Abaixo encontra-se um fragmento de texto:

[...]

Nesta primeira fase devemos considerar que há uma única praça de pedágio, responsável por controlar as operações. Basicamente a praça é composta pelos sistemas de Nível 0 (composto por controlador do laço magnético – para detectar a quantidade de eixos, leitores de RFID, câmeras de contexto e de OCR – para fotografar as placas). Na verdade, isso tudo só existe nas pistas automáticas, nas pistas de cobrança manual, há apenas o laço magnético, para auditoria. Esse negócio do laço magnético ser na saída da cabine é um super problema, pois a operadora pode errar na hora de contar a quantidade de eixos (imagine um caminhão em que o motorista jura que 4 eixos estão levantados, se ela acha que não, isso dá uma briga rs, mas se ela acha que sim e não está, cobram a diferença do salário dela, super injusto!).[...]

O terceiro problema também não se saiu muito bem, porém, em comparação com o segundo, teve uma evolução. Assim como no problema 2 existiam informações desnecessárias que confundem o aluno como a destacada abaixo:

[...]

Não há mais validação explícita de passagens, pois o cliente quer a validação no momento em que inserir a nova cobrança. *Só as passagens inválidas serão enviadas para o Nível 3, mas por enquanto não precisamos nos preocupar com essa integração.* Na verdade, o cliente nem nos passou direito a especificação do Nível 3, só sei que será na fase 7 do projeto. [...]

Ao ler este trecho, é questionado se realmente as passagens inválidas serão ou não enviadas para o nível 3.

Todos os problemas conseguem facilmente estimular o raciocínio. Eles promovem situações em que o educando pense muito no que é necessário e como realizá-lo, ou seja, os problemas proporcionaram chances para criar hipóteses e testá-las.

A promoção da aprendizagem autodirigida também teve um desempenho equilibrado. O primeiro problema teve um desempenho menor que os outros por se tratar de uma novidade recheada de complexidade, afinal ele introduz um novo paradigma de linguagem de programação.

Na sexta característica, ser de clareza adequada, três problemas obtiveram um bom desempenho: 1, 4 e 5. Estes explicavam o suficiente para um problema PBL, deixando claro os requisitos do programa teriam que ser desenvolvidos. Conforme já exemplificado, o problema 2 teve comparativamente o pior texto e isso o tornou confuso. No problema 3, embora tenha melhorado neste aspecto, ainda assim o texto era um pouco confuso se comparado aos demais.

Em relação a dificuldade, todos os problemas se saíram bem. Nem fáceis demais, nem difíceis demais. Aplicávamos conhecimento dos componentes teóricos de forma prática. Porém, no problema 4 tivemos que pensar muito numa forma de armazenamento de dados adequada para uma determinada classe. Isso custou alguns dias de buscas, teste de ideais até que chegássemos a uma solução.

As situações trazidas pelos problemas PBLs analisados aqui poucas vezes permitirão realmente aplicação ou uso, pois se tratam de softwares básicos apenas para o aprendizado de programação. O problema 1 traz uma situação que dificilmente poderá aparecer, já que o objetivo do programa é apenas rodar testes de unidades. Os problemas 2 e 3 lidam com o gerenciamento de pedágio, onde basicamente era exigido a criação um sistema que administrasse de maneira básica este tipo de negócio. A última situação (problema 4 e 5) está contextualizada em uma empresa de fretes que precisa de otimização de rotas em grafos ponderados e a interface gráfica, ou seja, chegando mais perto de softwares comerciais.

Todos os problemas atenderam bem as últimas características: utilizar conhecimento prévio, estimular a elaboração e promover o trabalho em equipe. Para solucionar os problemas os alunos utilizam toda a base de conhecimento construída ao cursarem os pré-requisitos dos componentes curriculares do MI. Além disso, o conhecimento utilizado em cada problema é praticamente acumulativo. Na Tabela 4 estão listados exemplos do conhecimento prévio necessário para resolução de cada problema.

Tabela 4. Conhecimento prévio utilizado por cada problema

<b>Problema</b>	<b>Conhecimento prévio necessário</b>
<b>1</b>	Noções de programação, lista encadeada
<b>2</b>	Noções de Programação, lista encadeada, Classes, Objetos
<b>3</b>	Noções de Programação, lista encadeada, Classes, Objetos Herança, Polimorfismo, Padrões de projeto, Diagrama de classes, Árvore.
<b>4</b>	Noções de Programação, lista encadeada, Classes, Objetos Herança, Polimorfismo, Padrões de projeto, Diagrama de classes, Arquivos, Exceções.
<b>5</b>	Noções de Programação, lista encadeada, Classes, Objetos Herança, Polimorfismo, Padrões de projeto, Diagrama de classes, Arquivos, Exceções, Interface gráfica.

Os problemas sempre tinham uma certa dificuldade de resolução, estimulando, por um lado, a efervescência de ideias e, por outro, a discussão em grupo dessas ideias. Muitas delas eram adotadas por todos do grupo, outras só por alguns — já que a solução em forma de produto deveria ser individual.

Finalmente, usando as notas atribuídas aos problemas em cada característica, ranqueamos os cinco problemas de duas formas: usando a média aritmética e a média ponderada. Esta última considera o ranking de cada característica, então o valor das escalas (de 1 a 5) terá como peso o valor inverso do ranking (peso da característica 1 é 11, da característica 2 é 10 e assim sucessivamente). O problema que tiver maior (menor) pontuação é classificado como melhor (pior) problema dentre os usados em MI – Programação na turma supracitada. O resultado é mostrado na Tabela 5.

Tabela 5. Média ponderada obtida para cada problema

Problema	Média aritmética	Média ponderada
1	4,1	4,0
2	3,9	3,7
3	4,3	4,1
4	4,5	4,4
5	4,6	4,6

Por apresentar os maiores valores em ambas as médias, o problema 5 pode ser considerado o melhor aplicado no MI – Programação da UEFS no semestre 2013.1. Já os dois primeiros problemas (1 e 2) deste componente podem ser considerados os que apresentaram comparativamente maiores limitações quanto à sua qualidade. Porém, ainda assim, é importante destacar que, considerando o todo, obtiveram um valor suficiente para serem considerados adequados.

## 5. Considerações finais

Este artigo apresentou uma análise de problemas PBL aplicados no MI – Programação da UEFS no período 2013.1, tendo como base características fundamentadas na perspectiva estudantil. Os principais aspectos negativos encontrados foram a apresentação do problema, em seu formato e texto, além da sua falta de aplicação direta ao mercado de trabalho (mas compreensível por se tratar de conceitos iniciais). Porém, isso não influenciou na qualidade como um todo; pelo contrário, todos estavam adequados à proposta do método PBL, até mesmo porque existiam outras características em avaliação como conhecimento prévio, trabalho em grupo, estimular raciocínio crítico, etc. As principais contribuições deste trabalho são a estruturação de uma avaliação de problemas PBL (em Computação) e a geração de um ranking deles.

É importante também relatar algumas limitações da pesquisa. Primeiro, por se tratar de um estudo com base na percepção de apenas um dos alunos do componente curricular, pode-se em certa medida haver divergências quanto ao restante da turma. Portanto, para futuras análises, é interessante ampliar o escopo, considerando uma amostragem significativa da turma. Segundo, aspectos como ser o primeiro problema ou o último são desconsiderados na análise. Talvez isso possa ter influenciado na qualidade à medida em que os tutores podem ir aprendendo com a experiência.

## 6. Referências

- Hung, W. (2006). The 3C3R Model: A Conceptual Framework for Designing Problems in PBL. *Interdisciplinary Journal of Problem- based Learning*, 1(1). Disponível em: <http://dx.doi.org/10.7771/1541-5015.1006>
- Marchais, J. E. Des. (1999) A Delphi technique to identify and evaluate criteria for construction of PBL problems, *Medical Education*, 33 (7), pp. 504–508.
- Munshi, Fadi F. M., Zayar, El E. S. A. and & Dolmans, Diana D. H.. (2008). Development and utility of a questionnaire to evaluate the quality of PBL problems. *South East Asian Journal of Medical Education*, Vol. 2, no 2, 2008.

- Pereira, C. F., Afonso, R. A., Santos, M. J., Araújo, C. A. L., Nogueira, M. (2007). Aprendizagem Baseada em Problemas (ABP) – Uma proposta inovadora para os cursos de engenharia. XIV SIMPEP 2007. Disponível em: [http://nogueira.eti.br/profmarcio/obras/publicado\\_1474.pdf](http://nogueira.eti.br/profmarcio/obras/publicado_1474.pdf)
- Ribeiro, Luis R. de C (2005). A Aprendizagem Baseada em Problemas (PBL): uma implementação na educação em engenharia na voz dos autores. Disponível em: : [http://www.btdt.ufscar.br/htdocs/tedeSimplificado/tde\\_arquivos/8/TDE-2005-05-16T12:29:32Z-668/Publico/TeseLRCR.pdf](http://www.btdt.ufscar.br/htdocs/tedeSimplificado/tde_arquivos/8/TDE-2005-05-16T12:29:32Z-668/Publico/TeseLRCR.pdf)
- Santos, J. A. M. & Angelo, M. F. (2009). Análise de Problemas Aplicados em um Estudo Integrado de Programação utilizando PBL. In XI Workshop sobre Educação em Computação – Anais do XXIII Congresso da Sociedade Brasileira de Computação, Bento Gonçalves – RS.WEI 2009.
- Schmidt, H. G., & Sockalingam, N (2011). Characteristics of Problems for Problem-Based Learning: The Students' Perspective. *Interdisciplinary Journal of Problem-based Learning*, 5(1). Disponível em: <http://dx.doi.org/10.7771/1541-5015.1135>