

**INOVAÇÃO NO ENSINO DE ENGENHARIA: o caso da disciplina de  
algoritmos e programação da PUC/PR**

[doi.org/10.22409/1807-5908.i74a144](https://doi.org/10.22409/1807-5908.i74a144)

**Dani Juliano Czelusniak <sup>1</sup>**

Pontifícia Universidade Católica do Paraná – PUC/PR

---

1 1 Contextus

ISSN 1807-5908

Editor Executivo: Martius Vicente Rodriguez y Rodriguez, DSc.

Avaliação: Double Blind Review

Recebido em 13/11/2017

Aceito em 17/04/2018

**Resumo:**

Com a “Era da Informação” e a conseqüente disponibilização dos mais diversos conteúdos didáticos na rede Internet, as instituições de ensino buscaram dinamizar seu processo de ensino, visando proporcionar aumento no desempenho da aprendizagem de seus estudantes. Nesse contexto, o objetivo deste trabalho é de apresentar o processo de reconstrução da disciplina de Algoritmos e Programação, presente nos cursos de Engenharia da Escola Politécnica da Pontifícia Universidade Católica do Paraná (PUC/PR), que utilizou principalmente a metodologia *Problem Based Learning* (PBL) aliada à metodologia de desenvolvimento de software *eXtreme Programming* (XP). O método de pesquisa foi o bibliográfico para apresentar as metodologias de ensino e desenvolvimento utilizadas e, estudo de caso, com a descrição da aplicação prática ocorrida em um semestre letivo no curso de Engenharia da PUC/PR. Como resultados, observou-se que os métodos mencionados apresentam bom potencial, pelo fato de proporcionar aos alunos ambiente dinâmico de aprendizado, enquanto estimula fortemente o trabalho em equipes.

**Palavras chave:** PBL; XP; Algoritmos.

**Abstract:**

With the "Information Age" and the consequent availability of the most diverse didactic contents in the Internet network, the educational institutions sought to streamline their teaching process aiming to increase the learning performance of their students. In this context, the goal of this work is to present the of reconstruction process of Algorithms and Programming course, from the Engineering courses of the Pontifical Catholic University of Paraná Polytechnic School, which used mainly the Problem Based Learning (PBL) and eXtreme Programming (XP) software development methods. The research method was bibliographic to present the teaching and development methodologies used, and a case study with the description of the practical application occurred in a semester of a Engineering course. As results, it was observed that the mentioned methods present good potential, because it provides the students with a dynamic learning environment, while strongly stimulating the work in teams.

## Introdução

Hoje na Escola Politécnica (e na grande maioria das universidades brasileiras) o ensino da disciplina de algoritmos e programação em cursos de engenharia segue o método tradicional de ensino, em que nas disciplinas o conteúdo é fragmentado e o professor leciona aulas expositivas e, individualmente, cada aluno, quando da aula prática em laboratório, sob a tutela do docente, implementa os conceitos e modelos estudados e ensaiados durante a aula teórica, em linguagem de programação.

Diversos autores da área sustentam que o processo de desenvolver software é equivalente a pintar um quadro; mesmo se tendo como base uma mesma obra, cada um dos envolvidos tecerá sua obra de forma diferente. Ou seja, o processo de construção de programas de computador possui altos teores conceituais e, exige igualmente, altos graus de cognição de parte de cada um dos alunos, tornando o processo de aprendizado diferente em cada pessoa.

Nesse contexto, as metodologias ativas de ensino<sup>1</sup> e as técnicas ágeis de desenvolvimento de software<sup>2</sup> surgem como ferramental que se mostra capaz de homogeneizar o conhecimento referente ao desenvolvimento de software dentre os alunos. Técnicas como por exemplo o trabalho em duplas<sup>3</sup>, em que em intervalos regulares de tempo os membros das duplas que estão trabalhando são trocados<sup>4</sup>, aliadas à exposição de conteúdos baseadas em problemas do cotidiano, permitem melhor aproveitamento do conteúdo como um todo por parte dos alunos através da troca de experiências, aprendizados e conhecimentos.

Face à complexidade inerente no aprendizado da disciplina de algoritmos e programação, justifica-se que o uso de metodologias ativas de ensino aliadas às técnicas ágeis

---

<sup>1</sup> Metodologias tais como a sala de aula invertida, aprendizagem baseada em projetos, instrução por pares, dentre outras.

<sup>2</sup> Técnicas desenvolvidas com o objetivo de proporcionar a criação de pequenos grupos produtivos de trabalho como *eXtreme Programming (XP)*, *Scrum*, *Coding Dojo*, dentre outras, ao mesmo tempo em que estimulam o compartilhamento do conhecimento entre os membros destes grupos.

<sup>3</sup> Programação em pares, ou no termo original em inglês *pair programming* proposto por Kent Back, idealizador da metodologia XP.

<sup>4</sup> Técnica da mesma metodologia XP já citada, intitulada *move people around*.

de desenvolvimento de software, que possibilitem maior interação entre os alunos, corrobora com as alíneas da Universidade em que se busca estimular o uso de metodologias ativas de ensino, pautadas na troca de experiências e conhecimentos, bem como no aprendizado através de atividades lúdicas voltadas à resolução dos problemas do cotidiano que, serão enfrentados pelos alunos, quando futuramente estiverem atuando em seus postos de trabalho.

Assim, o objetivo deste artigo é apresentar o processo utilizado na reconstrução da disciplina de algoritmos e programação da Escola Politécnica da Pontifícia Universidade Católica do Paraná (PUC/PR), a fim de possibilitar a utilização de metodologias ativas de aprendizado aliadas às melhores práticas das técnicas ágeis de desenvolvimento de software nas turmas, bem como a futura capacitação do corpo docente a fim de realizarem atividades de “desenvolvimento ativo ágil de software” no ensino.

### **Referencial Teórico**

Após a década de 1960, observou-se uma evolução de novas tecnologias, principalmente aquelas relacionadas com as tecnologias de informação e comunicação (TICs). Mais amplamente, na década de 1970, houve o aceleração do desenvolvimento, convergindo em um novo paradigma, que veio a ser denominado de a “Era Informacional” (CASTELLS, 2006). O desenvolvimento dessas tecnologias foi responsável por mudanças significativas nos meios de produção e, também, no cotidiano das pessoas. Lastres (2007, p. 188) aponta que a utilização das TICs não está restrita apenas às empresas e indústrias e aos “setores de ponta”, mas que essas tecnologias possuem um caráter mais amplo e transformaram o modo do ser humano aprender, pesquisar, produzir, trabalhar, consumir, divertir-se e exercer a cidadania.

Nesse contexto, no âmbito das relações ensino-aprendizagem, também se observa mudanças significativas atualmente, que trazem desafios para os profissionais do ensino em se adequar a essas novas realidades. Os discentes experimentam uma realidade em que muitas informações já estão disponíveis e que, por isso, a absorção delas é mais acelerada. Assim, novas propostas de ensino vêm sendo criadas para tornar este ambiente mais colaborativo e mais aberto a atuação da tecnologia, possibilitando que o educando possa buscar

conhecimentos também fora da sala de aula, de modo mais proativo e autônomo, interagindo com sua realidade social.

Essas novas propostas de ensino são conhecidas como metodologias ativas (Active Learning), que, segundo Bonnell e Eison (1991, p. 2), “em resumo, no contexto da sala de aula da faculdade, [...] envolve estudantes fazendo coisas e pensando nas coisas que estão fazendo”. Davis (1993, p. 53) afirma que, em se adotando estilos pedagógicos participativos, todos os alunos se beneficiam [...], pois podem aprender discutindo, escrevendo ou fazendo, inclusive os estudantes repetentes, em particular, que são menos propensos a tolerar palestras passivas. Eles aprendem mais falando efetivamente sobre questões do que apenas ouvindo: “eles querem interagir”.

Existem diversos tipos de operacionalização das metodologias ativas referidos pela literatura, como o PBL – Problem Based Learning, o Arco de Charles de Margueréz, a TBL – Team Based Learning, o círculo de cultura (GUIMARÃES et. al., 2016). Entende-se que uma linha comum entre todos é colocar o aluno diante de problemas e/ou desafios que façam com que seja despertado seu potencial intelectual; não somente sobre o conteúdo curricular trabalhado, mas que desenvolva no aluno capacidades de compreensão e análise crítica para a superação de desafios, integrando teoria e prática (PAIVA et. al., 2016).

Neste trabalho, o método que será tratado é o PBL – Problem Based Learning. A Aprendizagem baseada em problemas (PBL) é “um método de instrução caracterizado pelo uso de problemas da vida real para estimular o desenvolvimento de pensamento crítico e habilidades de solução de problemas e a aprendizagem de conceitos fundamentais da área de conhecimento em questão” (RIBEIRO, 2008). O método permite que os alunos resolvam problemas relacionados com as suas futuras profissões e os estimula a pesquisar, tornando-os capazes de aprender a aprender, a serem críticos e a tomarem decisões (BOROCHOVICIUS e TORTELLA, 2014).

Também, outra metodologia utilizada neste estudo foi a metodologia de construção de software *eXtreme Programming*, também conhecida como XP, que é um método de trabalho que preconiza principalmente o trabalho em equipes, a difusão do conhecimento sobre o

projeto entre os membros da equipe, o trabalho do time de desenvolvimento em duplas e rápido *feedback* entre as fases de *desing* e desenvolvimento de software.

Proposta por Kent Beck nos idos de 1997, preconiza como principais valores a serem cultivados dentro do time de desenvolvimento de software a comunicação entre as pessoas, simplicidade do código escrito no programa, curtos períodos de tempo entre os *feedbacks* dados entre as fases e micro-fases do desenvolvimento do projeto, coragem para trabalhar com os desafios diários impostos pelo projeto e, respeito mútuo entre os *stakeholders* envolvidos Back (2004).

Dentre as mais de dez práticas propostas pela metodologia XP, foram escolhidas para essa pesquisa somente três delas, a fim de permitir a aplicação do trabalho desenvolvido em PBL durante as fases de construção de software em laboratório de informática da instituição, são elas o Design Simples (*Simple Design*), Programação Pareada (*Pair Programming*) e a Refatoração (*Refactoring*), que são:

Design Simples: A simplicidade é um dos princípios básicos dessa metodologia. Quando do entendimento do problema a ser resolvido no formato de algoritmos, os estudantes são incentivados a resolverem o problema da forma mais simples o possível. Após os estudantes identificarem como o problema pode ser resolvido, partem para o desenvolvimento das estruturas em software que permitirão a melhoria estrutural do software como um todo como ajustar as mensagens apresentadas para o usuário, melhorar a entrada de dados, dentre outras.

Refatoração: Refatoração é um processo em que, os estudantes são incentivados a revisitar os programas já desenvolvidos na disciplina para fins específicos e tentar ajustá-los para atender a novas demandas. Por exemplo, foram desenvolvidas quatro atividades em sala de aula e laboratório onde os alunos escreveram programas para realizar operações matemáticas simples com dois números apresentando o resultado ao final. Em um exercício de refatoração, os alunos revisitaram esses exercícios e, construíram uma calculadora em software.

Programação em Pares e o *Moving People Around*: A programação em pares da XP preconiza que o desenvolvimento de software deve sempre acontecer a quatro mãos e um teclado, ou seja, uma dupla de estudantes por computador. Dessa forma, permite-se que os estudantes

troquem experiências durante o processo de desenvolvimento à medida em que cada um escreve um pouco do programa de computador que está sendo construído. Durante o desenvolvimento em intervalos regulares de tempo (no caso da disciplina, a cada quinze minutos) era efetuada a troca de um dos membros da dupla (*move people around*) que iria trabalhar com outro colega, forçando-o a compreender o código que fora inicialmente escrito sem sua participação na nova dupla. Quinze minutos depois acontecia uma nova rodada de duplas em que o membro original da dupla que havia escrito o código desde o início ia para outra dupla. Neste ponto, em todas as duplas não haviam membros que estivessem desde o início da redação daquele código fonte, obrigando ambos a entenderem a programação elaborada por outro colega, ajustá-la e finalizar o programa com os testes de funcionalidade antes da entrega. Esse exercício propicia uma homogeneização de conhecimento mais uniforme na sala de aula, no tocante à programação de computadores.

### **A Nova Proposta para a Disciplina de Algoritmos e Programação**

Durante as fases do planejamento da nova proposta de disciplina de algoritmos e programação, buscou-se satisfazer os três critérios de aprendizagem propostos por Saroyan e Amudsen (2004):

- Requerer o envolvimento estudantes com as atividades propostas, a fim de satisfazer os princípios da aprendizagem ativa.
- Estruturar modelo de ensino que foca em oportunizar a prática do aprendizado pelos estudantes, ao mesmo tempo em que os instiga a refletirem sobre seu aprendizado e receberem feedback.
- Apresentar variedade de atividades que se adaptem aos diferentes tipos de aprendizado.

*Planejar uma disciplina de acordo com a estratégia de design, focada nos resultados de aprendizagem e na essência do significado da disciplina nos contextos acadêmico e profissional (SAROYAN, 2004).*

Buscou-se desenhar na disciplina, mecanismos em que os estudantes deverão realizar atividades reflexivas e reativas que vão além de ouvir o professor e tomar notas. Pois são características desejáveis da aprendizagem ativa estar associada ao mundo real, ou seja, ajuda a desenvolver aprendizados que permanecem mesmo com o passar do tempo, manter o estudante motivado. Para isso, o processo de aprendizagem deve ser significativo para ele, propiciar ambiente onde o aluno esteja ativo e reflexivo e, também requerer a colaboração mútua dos estudantes.

Nessa nova proposta de disciplina, buscou-se manter o foco do trabalho dos alunos nos resultados de aprendizagem, que são suportados pelo conteúdo trabalhado durante o semestre. Para tal, fora elaborado um mapa mental para a disciplina, que é uma das ferramentas que permite observar a disciplina de acordo com seus conceitos estruturantes, direcionando o foco do trabalho nos resultados de aprendizagem desejados.

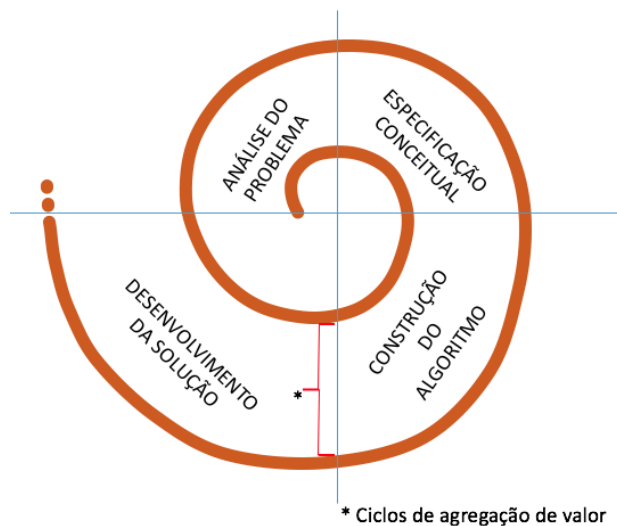


Figura 1 - Mapa mental proposto para a disciplina  
Fonte: O autor.

Esse mapa mental, inspirado em um *Espiral de Boehm*<sup>5</sup>, busca suprir a seguinte

<sup>5</sup> *Metamodelo* criado por Barry Boehm em 1988, amplamente utilizado em disciplinas de projeto de software, o qual permite acomodar diversos processos específicos de planejamento e desenvolvimento de software baseado em ciclos.



descrição e alcance dos conteúdos a serem trabalhados na disciplina, com foco na aprendizagem do aluno: “Essa disciplina faz parte do núcleo básico de todos os cursos de Engenharia da Escola Politécnica. Ela concentra-se na elaboração de algoritmos em pseudo-linguagem, especificação gráfica no formato de fluxogramas e construção de programas de computador em linguagem C. Ao final, os estudantes são capazes de analisar problemas interdisciplinares e especificar soluções que poderão ser implementadas em software.” Como o mapa é baseado em um espiral, entende-se que a cada ciclo de análise, especificação, construção e desenvolvimento há a melhoria contínua sobre o problema (tratada no item *refatoração* apresentado anteriormente), em que há um processo de melhoria contínua com a re-análise do problema, para adaptações a novos cenários e melhoria na geração de seus resultados.

Essa descrição conduziu aos seguintes resultados de aprendizagem:

- Aplicar conceitos fundamentais de algoritmos, na resolução de problemas básicos oriundos das disciplinas iniciais dos cursos de engenharia.
- Internalizar a premissa do algoritmo na criação do programa de computador.
- Converter algoritmos em instruções de programação em ambiente de construção de software.

A fim de alcançar os resultados de aprendizagem listados acima, foi elaborado um protótipo de processo de ensino e aprendizagem, aula a aula, iniciando a partir de uma questão norteadora narrada no formato de uma pergunta, que foi respondida com o encerramento da aula, embasada na teoria e práticas trabalhadas. Essa questão norteadora abordou um (ou um conjunto de) conteúdo presente na ementa da disciplina.

O protótipo de processo de abordagem contou com a estrutura descrita a seguir: sempre que possível, o trabalho foi iniciado retomando a resolução de um problema da aula anterior, a fim de concatenar a aula presente com a passada, permitindo ao aluno perceber a conexão e evolução dos conteúdos trabalhados. Então, fora explanado o problema inicial a ser tratado em aula, a fim de buscar formas (baseadas no conteúdo alvo) de resolvê-lo. Então, o

professor iniciou uma primeira explanação expositiva sobre o assunto, a fim de homogeneizar o conhecimento pertinente ao problema na turma. Após a explanação, os alunos, em grupos, tiveram um espaço de tempo a fim de buscar alternativas para a resolução do problema. Em seguida, o professor interveio retomando a palavra, iniciando uma segunda rodada expositiva a fim de complementar a teoria pertinente a aquele assunto, dando início em mais um tempo de resolução para os alunos, mas desta vez modificando a composição das equipes de alunos. Após esta segunda rodada de trabalho, o professor concluiu a aula resolvendo a atividade com os alunos e lançando uma atividade de fixação para ser realizada fora da sala de aula, que foi corrigida no início da próxima aula a fim de conectar o conteúdo tratado com o próximo assunto.

Foi elaborado estruturalmente, este protótipo de processo de desenvolvimento de aula abaixo pelo Quadro 1.

Detalhes	Tempo (minutos)
Detalhes Ementa	
Número do encontro	N/A
Questão norteadora	N/A
Conteúdo programático alvo	N/A
Resolução da atividade de fixação proposta ao final da aula anterior	00:10:00
Problema Inicial	00:05:00
1a. rodada expositiva	00:10:00
Atividade	00:15:00
2a. rodada expositiva	00:10:00
Continuação da atividade	00:15:00
Fechamento da atividade (c/ correção)	00:20:00
Explicação do enunciado da atividade de fixação	00:05:00
<b>Tempo total aproximado</b>	<b>01:20:00</b>

Quadro 1 - Protótipo Estrutural de Aula  
Fonte: o autor.

Nota-se que o tempo total aproximado é menor que o tempo total de aula em alguns minutos (10 minutos). Como esta é uma estrutura protótipo que foi ensaiada durante o semestre, sugere-se a manutenção de tal espaço temporal a fim de permitir alguma flexibilidade entre as fases, bem como o tempo para ligar equipamento audiovisual da sala de aula, bem como efetuar o preenchimento do controle de presença.

Como exemplo, o Quadro 2 apresentado na sequência demonstra a estrutura de uma

aula do curso de algoritmos sobre o conteúdo programático “variáveis e constantes”, presente na ementa do curso.

	Detalhes	Tempo (minutos)
Detalhes	<b>Número do encontro:</b> 3	N/A
	<b>Questão norteadora:</b> De que forma podemos manusear dados em um algoritmo?	N/A
Ementa	<b>Conteúdo programático alvo:</b> Como funcionam os tipos de dados, constantes, variáveis e atribuição.	N/A
	<b>Resolução da atividade de fixação proposta ao final da aula anterior:</b> resolução do algoritmo que mostra o passo a passo do processo de preparação de um pão de queijo.	00:10:00
	<b>Problema Inicial:</b> Precisamos armazenar as medidas da sala de aula para calcular sua área e volume. Quais as possibilidades que temos para resolvermos esse problema com algoritmos e programação?	00:05:00
	<b>1a. rodada expositiva:</b> explanação sobre variáveis e seus tipos.	00:10:00
	<b>Atividade:</b> Como armazenar os valores das medidas da sala nas variáveis de um algoritmo para efetuar o cálculo da área.	00:15:00
	<b>2a. rodada expositiva:</b> explanação sobre constantes e seu uso.	00:10:00
	<b>Continuação da atividade:</b> No algoritmo já iniciado, atribuir a altura do “pé-direito” da sala como constante, pois será um valor padronizado para todas as salas do andar e será utilizada para o cálculo do volume.	00:15:00
	<b>Fechamento da atividade (c/ correção):</b> Demonstrar o uso das variáveis e constantes em algoritmos no problema estudado, evidenciando a sua diferença.	00:20:00
	<b>Explicação do enunciado da atividade de fixação:</b> Construir algoritmo que permita calcular a área de um círculo qualquer, utilizando o valor de $\pi$ como constante.	00:05:00
	<b>Tempo total aproximado</b>	01:20:00

Protótipo do Processo de Abordagem

Quadro 2 - Exemplo de Protótipo Estrutural de Aula Preenchido

Fonte: o autor.

Além da estruturação da disciplina em si, compreendeu a abrangência deste projeto a identificação de conteúdos prontos em formatos diversos (textuais, áudio e vídeo) passíveis de uso na rede Internet, bem como, a geração do conteúdo que fora necessário *in loco*. Também foi necessário o projeto e criação dos itens de aprendizagem no AVA da PUCPR, a ferramenta Blackboard, para acompanhamento dos alunos e, a organização (e valoração) das atividades de fixação em formato que permita seu enquadramento como atividades de Trabalho Discente Efetivo (TDE). A fim de ilustrar de uma melhor forma a estrutura da disciplina desenvolvida durante o semestre, o Quadro 03 apresentado na sequência traz o conjunto de todas as questões norteadoras tratadas nos encontros semanais, alinhadas com

conteúdo programático alvo e os comandos de linguagem C envolvidos.

Semana de Aula	Questões Norteadoras	Conteúdo Programático Alvo	Comandos envolvidos (linguagem C)
1	Apresentação da disciplina e laboratórios.	N/A	N/A
2	O papel dos algoritmos no processo de resolução de problemas.	Sintaxe e semânticas básicas de uma linguagem de alto nível	Estrutura do programa, <i>printf</i>
3	Como manusear dados em um algoritmo.	Como funcionam os tipos de dados, constantes, variáveis e atribuição	Variáveis, constantes, <i>printf</i>
4	Elaborando operações de entrada e saída de dados simples em um algoritmo.	Uso básico de expressões aritméticas em processos de entrada e saída	Comandos <i>printf scanf</i> , operações e atribuição
5	Como testar uma condição em um algoritmo.	Controle de fluxo com expressões lógicas	Comandos <i>if else</i>
6	Como efetuar vários testes de condições em um mesmo algoritmo.	Controle de fluxo com múltiplas expressões lógicas	Comandos <i>if else if else</i>
7	Simplificando o algoritmo, utilizando testes caso a caso.	Controle de fluxo com múltiplas expressões lógicas	Comando <i>select case</i>
8	Exercícios e atividades de fixação.	Sintaxe, operações com variáveis e constantes e controle de fluxo	Comandos <i>printf scanf, if, else, elseif, select case</i> , operadores lógicos, aritméticos
9	<b>Avaliação da disciplina.</b>		
10	Como executar um bloco de comandos em um algoritmo até que uma condição seja atendida?	Laços de repetição com comando <i>enquanto</i>	Comandos <i>while e do/while</i>
11	Automatizando tarefas simples em um algoritmo: entendendo melhor as estruturas de repetição.	Laços de repetição com comando <i>para</i>	Comando <i>for</i>
12	Como trabalhar com conjuntos de valores de um mesmo tipo.	Estruturas de dados unidimensionais – Vetores	Comando <i>for</i> e <i>vetor[x]</i>
13	Se precisar armazenar muitos valores é possível? É sim, utilizando matrizes!	Estruturas de dados multidimensionais – Matrizes	Matrizes <i>matriz[i][j]</i> e <i>for</i>
14	Se precisar armazenar muitos valores é possível? É sim, utilizando matrizes!	Estruturas de dados multidimensionais – Matrizes	Matrizes <i>matriz[i][j]</i> e <i>for</i>
15	Quer simplificar o entendimento um algoritmo que ficou muito complexo? Divida-o em partes.	Modularização	<i>void funcao(void) / tiporetorno funcao (tipovar Nx, tipovarNy)</i>
16	Exercícios e atividades de fixação.		
17	<b>Avaliação da disciplina.</b>		
18	<b>Fechamento de notas e encerramento da disciplina.</b>		

Quadro 3 - Questões norteadoras do curso de Algoritmos e Programação

Fonte: o autor.

### **A Experiência em Sala de Aula (diário de bordo)**

A seguir, estão os detalhes do diário de bordo, em que foram tomadas notas, aula a aula, durante o semestre, no tocante ao andamento da disciplina.

Turmas: Eng. Civil A e B

Data: 21 a 24/2

Período (Manhã/Tarde/Noite): M

Tema da Aula: Apresentação da disciplina

Número de estudantes: 80% do total (de 60 em cada turma)

Como imaginei que seria esta aula?

Apresentação do plano de ensino, livros de apoio na biblioteca virtual, metodologias utilizadas no ensino e estrutura de laboratórios.

Como de fato a aula ocorreu?

N/A

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 02 a 3/3 - aulas práticas

Período (Manhã/Tarde/Noite): M

Tema da Aula: Aclimação com ambiente de programação

Número de estudantes: 80% do total (de 60 em cada turma) devido ao advento da semana do carnaval

Como imaginei que seria esta aula?

Os alunos puderam ter o primeiro contato com o ambiente de desenvolvimento em linguagem de programação no laboratório. Receberam noções de funcionamento do compilador e fizeram o primeiro programa que mostra a mensagem “Olá mundo! ?;-)” em linguagem C.

Como de fato a aula ocorreu?

A aula transcorreu sem anormalidades, porém os alunos acharam o ambiente de programação “difícil” devido ao número de detalhes necessário na programação.

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

---

---

Turmas: Eng. Civil A e B

Data: 07 a 10/3 - aulas práticas

Período (Manhã/Tarde/Noite): M

Tema da Aula: Introdução ao estudo de algoritmos

Número de estudantes: 100% do total (de 60 em cada turma)

Como imaginei que seria esta aula?

Aula expositiva, em que foram apresentados os detalhes de formalismo existentes na produção de um algoritmo em fluxograma e pseudo-código, bem como sua correlação com os comandos que serão desenvolvidos em laboratório.

Em laboratório, implementamos programas lúdicos capazes de demonstrar o passo a passo para fazer pães de queijo.

Como de fato a aula ocorreu?

A aula transcorreu sem anormalidades, nas aulas teóricas montamos as receitas em duplas em sala de aula. Na prática os alunos programaram as receitas na linguagem de programação, pena que alguns esqueceram de pegar a receita com o colega que estavam na outra divisão da turma, mas, mesmo assim, construíram seus programinhas.

O que mudaria? Por quê?

Nada, acredito que o objetivo de se aclimatar com as estruturas lógicas básicas de programação foi atendido.

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 07 a 10/3

Período (Manhã/Tarde/Noite): M

Tema da Aula: Como manusear dados em um algoritmo

Número de estudantes: 100% do total (de 60 em cada turma)

Como imaginei que seria esta aula?

Aula 50% expositiva e 50% estudo dirigido nas teóricas e práticas. Na teórica, no primeiro momento (cerca de 20 minutos) foi efetuada exposição sobre os conceitos de variáveis e constantes, demonstrando como se manuseia dados através de algoritmos. Em seguida exploramos juntos alguns problemas simples que deveriam fazer operações com dois valores.

Nas aulas práticas ocorreu seguiu-se o mesmo *modus operandi*, em que fiz uma explanação de uns 20 minutos correlacionando as estruturas vistas em sala de aula com os comandos em linguagem de programação no laboratório e resolvemos os mesmos exercícios, mas desta vez programando de acordo com os algoritmos desenvolvidos em sala.

Como de fato a aula ocorreu?

A aula transcorreu sem grandes anormalidades, porém os alunos sentiram bastante dificuldade com o ambiente de programação todo o tempo esqueciam os pontos-e-vírgulas do final das linhas, esqueciam aspas, trocavam parênteses por chaves, etc. Demonstrando que estavam desatentos no momento da programação, mesmo sabendo que nas aulas anteriores fora mencionada a questão da sintaxe e semântica dos comandos.

O que mudaria? Por quê?

Não há o que mudar ainda. O interessante ao término das aulas foi o fato dos alunos comentarem entre si que, se não prestassem atenção não conseguiriam acompanhar o ritmo das próximas aulas.

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 14 a 17/3

Período (Manhã/Tarde/Noite): M

Tema da Aula: PBL – Um algoritmo para ler notas e faltas.

Número de estudantes: 100% do total (em torno de 60)

Como imaginei que seria esta aula?



Aula em PBL. Lançado um problema para os alunos, em que o programa deveria ler as faltas de um aluno e o número de aulas do curso. Se o aluno não estivesse reprovado em faltas, ler as duas notas, calcular a média e dizer se o aluno estava em aprovado, em final ou reprovado por nota. O algoritmo foi desenvolvido em sala em nos formatos fluxograma e pseudo-código e, fora implementado em laboratório em duplas (pair programming).

Como de fato a aula ocorreu?

A aula teórica gerou bastante discussão entre os grupos, até que um grupo de alunos sugeriu que construíssemos também o cálculo da média final, que fizemos todos juntos no quadro. Nas aulas práticas, novamente, os alunos sentiram bastante dificuldade com o ambiente de programação todo o tempo esqueciam os pontos-e-vírgulas do final das linhas, esqueciam aspas, trocavam parênteses por chaves, etc. Mas algumas duplas já conseguiram desenvolver a aplicação praticamente inteira sem meu auxílio.

O que mudaria? Por quê?

Percebi nesta aula a completa ausência de raciocínio lógico por parte dos alunos. Sem contar que, na ânsia de terminar o programa e ver se estava rodando, muitas equipes simplesmente não liam o código que estavam escrevendo e cometiam os erros mais diversos. Talvez fossem necessárias algumas aulas expositivas de lógica booleana. No formato college isso estará resolvido, pela presença da disciplina de “Lógica para Engenharia\*” no primeiro período com 36h/a.

O que manteria? Por quê?

N/A

\*Informação do primeiro relatório quando a referida disciplina estava na grade. Essa foi abandonada e a disciplina de algoritmos está dentro das disciplinas “Modelagem matemática e computacional” I e II.

Turmas: Eng. Civil A e B

Data: 21 a 24/3

Período (Manhã/Tarde/Noite): M

Tema da Aula: PBL – O Estacionamento Rotativo

Número de estudantes: 100% do total (em torno de 60)

Como imaginei que seria esta aula?

Aula em PBL. Lançado um problema para os alunos, que devem construir o algoritmo (e posterior código fonte em laboratório em duplas) para automatizar a tarefa de cálculo de valor a pagar em um estacionamento.

Como de fato a aula ocorreu?

Passei as instruções iniciais e interferi o mínimo possível na resolução do problema pelos grupos. Visivelmente uma quantidade menor de alunos teve dificuldade com a organização do raciocínio. Colhi um fato interessante: enquanto eu estava em sala, eu era sempre requisitado pelos grupos, nem que fosse para eles a cada passo me perguntarem “... está certo isso professor?” Quando com a turma B realizando a mesma atividade, ausentei-me da sala dizendo que ia à secretaria e fiquei observando-os do corredor trabalhando. Quando retornei alguns minutos depois a resolução estava mais adiantada que a turma anterior em um mesmo período de tempo. Ou seja: eles ficam gastando tempo em confirmar o que eles já sabem comigo ao invés de desenvolver suas soluções, o que denota insegurança no processo criativo. Na aula prática tudo ocorreu bem, o desenvolvimento está mais encaminhado, na próxima semana vou começar a trocar os pares (move people around da metodologia eXtreme Programming) durante a execução da tarefa

O que mudaria? Por quê?

Talvez eu devesse passar atividades para eles e sair da sala, só voltar ao final da aula para colher o resultado na tentativa de deixá-los quem sabe, mais confiantes de suas capacidades e menos apegados ao “está certo professor?” Porém, pelos olhos dos alunos, pode parecer que eu estou fazendo pouco caso deles. Tenho que encontrar uma medida para trabalhar isso com segurança.

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 28 a 31/03

Período (Manhã/Tarde/Noite): M

Tema da Aula: PBL – Brincando com a matemática

Como imaginei que seria esta aula?

Contextualizei em sala a importância da utilização do algoritmo para o entendimento e resolução de problemas tanto lógicos quanto matemáticos, e apresentei o PBL em duas etapas, na primeira verificaríamos uma figura geométrica se é ou não triângulo e, se fosse, diríamos qual tipo de triângulo era (equilátero, isósceles ou escaleno). Na primeira eu resolvi partes junto com eles no quadro e, solicitei a opinião de alguns alunos sobre a forma de construção do algoritmo, fechando a primeira parte em seguida. Na segunda parte, o enunciado pedia que fosse calculados coeficientes de Bháskara. Expliquei como se utilizavam as funções matemáticas e os alunos o fizeram sozinhos com pouquíssimas interferências minhas.

Como de fato a aula ocorreu?

A aula ocorreu conforme mencionada acima. Os alunos já estão me perguntando menos e estão conseguindo resolver as atividades sozinhos tanto em sala quanto em laboratório. Induzi algumas equipes com maior dificuldade a conversarem com os membros de uma das equipes

que estava com o código praticamente pronto a fim de verem as diferenças nos códigos de ambos. Apliquei a técnica das trocas de duplas, no início os alunos reclamaram que não entendiam o código dos outros, mas depois perceberam que era mais fácil aprender com as resoluções dos outros e em seguida modificá-las para que se adaptassem ao seu enunciado de problema.

O que mudaria? Por quê?

N/A.

O que manteria? Por quê?

N/A.

---

Turmas: Eng. Civil A e B

Data: 04 a 07/04

Período (Manhã/Tarde/Noite): M

Tema da Aula: Listas de Exercícios para treinamento

Como imaginei que seria esta aula?

Aula de resolução simples de listas de exercícios em duplas.

Como de fato a aula ocorreu?

Os alunos resolveram em duplas os exercícios em sala de aula e no laboratório. Tira dúvidas individual durante a aula, alguns alunos optaram por resolver sozinhos, alegando que “na prova resolveriam sozinhos”, por isso iriam praticar naquele momento como tal.

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 11/04 – Semana da Páscoa Período (Manhã/Tarde/Noite): M

Tema da Aula:

Como imaginei que seria esta aula?

Como perdemos as aulas práticas devido ao feriado da Páscoa, montei um “pseudo PBL” para discutir com os alunos em sala de aula a resolução de 04 exercícios com níveis de dificuldades diferentes.

Como de fato a aula ocorreu?

Como apresentada acima. Inclusive tivemos menos alunos em sala, provavelmente devido ao feriado.

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

Turmas: Eng. Civil A e B

Data: 18 a 20/04 – Feriado Tiradentes Período (Manhã/Tarde/Noite): M

Tema da Aula: Estruturas de Controle

Como imaginei que seria esta aula?

Partimos do pressuposto que precisávamos construir uma pequena calculadora para resolver um problema de engenharia. Então, tentamos juntos (eu e os alunos) montar uma calculadora de quatro operações básicas com os comandos que eles já haviam visto em linguagem C.

Como de fato a aula ocorreu?

Logo no início da atividade os alunos já perceberam que as estruturas estudadas não seriam adequadas para a construção da calculadora, precisavam de algo “diferente”. Um dos alunos com o google na mão, pois tinha visto um amigo usando em outro momento lembrou do comando switch/case (que era o foco da aula) e conseguiu efetuar uma implementação bem espartana de seu funcionamento. Isso espalhou-se para as outras equipes rapidamente e, antes que usassem o comando de forma errônea, eu tomei as rédeas da aula explicando a forma correta de seu uso e, todos construíram suas calculadoras. Uma das turmas não conseguiu fazer a parte prática devido ao feriado, então, liberei para todos uma resolução em fonte para que utilizassem como base para estudo.

O que mudaria? Por quê?

Os alunos estão ficando mais “pro-ativos”, tentarei depois da avaliação passar para eles problemas e resoluções de cenários semelhantes para avaliar se conseguem aprender e resolver sem a interferência do professor.

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 25 a 28/04

Período (Manhã/Tarde/Noite): M

Tema da Aula: Avaliação e resolução de exercícios em laboratório

Como imaginei que seria esta aula?

A aula teórica fora utilizada para a prova da disciplina. Nas aulas práticas, reconstruímos em grupos os PBLs da calculadora e também fizemos um exercício para implantação de (simulado) uma máquina de transações eletrônicas financeiras, conhecida como TEF, ou ainda, carinhosamente, a “maquininha de pagamentos via cartões”.

Como de fato a aula ocorreu?

Os alunos montaram as duas resoluções sem maiores dificuldades. A grande maioria, na metade do tempo resolveu ambos os exercícios.

O que mudaria? Por quê?

Vou tentar montar um exercício em laboratório em equipes sem a minha presença e do monitor para ver se os alunos conseguem resolver sozinhos. Talvez eu também deva aumentar a complexidade dos exemplos.

O que manteria? Por quê?

N/A

Turmas: Eng. Civil A e B

Data: 02 a 05/05

Período (Manhã/Tarde/Noite): M

Tema da Aula: Vistas e resolução das provas.

Como imaginei que seria esta aula?

Aulas teórica e práticas foram utilizadas para as vistas de provas pelos alunos, também fizemos no laboratório os exercícios da avaliação com a finalidade de exercitar os conteúdos e identificar os principais erros. Montei grupos de alunos e incentivei-os a discutirem entre si as resoluções possíveis para as questões, para que um aprendesse com o erro dos outros.

Como de fato a aula ocorreu?

N/A

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 09 a 12/05

Período (Manhã/Tarde/Noite): M

Tema da Aula: Como executar um bloco de comandos em um algoritmo até que uma condição seja atendida?

Como imaginei que seria esta aula?

Voltamos nos exercícios resolvidos nas semanas anteriores, em que os alunos perceberam que, se precisassem executar o programa mais de uma vez era necessário rodá-lo manualmente quantas vezes fosse necessário. Então, utilizamos dos mesmos exercícios já prontos e modificamos sua estrutura para que fosse executado até que uma condição fosse atendida. Montamos contadores de tempo e melhoramos a calculadora montada em aulas passadas, para que após o resultado ela solicitasse novamente outra operação a ser realizada.

Como de fato a aula ocorreu?

Exatamente como descrito acima, o processo de aprendizado está um pouco lento nessa parte da matéria pois alguns alunos não conseguiram captar ainda a essência da estrutura de



controle de fluxo, mas acredito que com os exercícios para serem resolvidos com a monitoria o entendimento melhora.

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 16 a 19/05

Período (Manhã/Tarde/Noite): M

Tema da Aula: Automatizando tarefas simples em um algoritmo: entendendo melhor as estruturas de repetição

Como imaginei que seria esta aula?

Voltamos aos exercícios já prontos para melhorá-los de deixá-los mais atraentes do ponto de vista lógico-computacional. Muitos alunos riram de códigos montados por eles mesmos nas semanas anteriores, ouvi frases como “não acredito que construí com essa lógica imunda... rs, vou melhorar isso...”. Muitos dos alunos perceberam os pontos fracos nos próprios programas elaborados no início da disciplina e quase a totalidade já está conseguindo raciocinar com formalismo lógico nas modificações dos próprios códigos fonte e de colegas. Já vejo alunos terminando rapidamente as atividades e indo auxiliar os colegas.

Como de fato a aula ocorreu?

Conforme descrito acima.

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 23 a 26/05

Período (Manhã/Tarde/Noite): M

Tema da Aula: Como trabalhar com conjuntos de valores de um mesmo tipo

Como imaginei que seria esta aula?

Trabalhamos com conceitos de vetores. Para isso, reaproveitamos um exercício desenvolvido na semana passada, porém as equipes adotaram um programa já resolvido por outra equipe, a fim de estudar e entender a lógica de programação construída por outro colega. Então, transformamos os conjuntos de variáveis existentes em vetores para armazenamento de informações de um mesmo tipo. Os alunos conseguiram perceber que é mais fácil manusear informações complexas nesse tipo de estrutura, porém alguns estão com dificuldade de entender o posicionamento dos contadores de acesso à informação.

Como de fato a aula ocorreu?

No início da aula quando tratamos dos conceitos estruturantes de vetores, fiz uma sessão expositiva de cerca de 50 minutos. Os alunos ficaram extremamente entediados. Foi bem interessante perceber que, depois de utilizar os métodos de aprendizado baseado em problemas eles acham a aula “convencional” entediante. A monitora me falou que ouviu um grupo de alunos falando que é chato aula nesse formato, que eles já estão mais acostumados a

entender resolvendo os “probleminhas”.

O que mudaria? Por quê?

Vou tentar tirar a parte expositiva na próxima repetição da disciplina.

O que manteria? Por quê?

O estudo de caso com o exercício da aula anterior foi bem interessante, pois como é uma tratativa recente, os alunos lembram mais facilmente e ganhamos algum tempo que possibilitou resolver mais exercícios dentro da mesma aula.

---

Turmas: Eng. Civil A e B

Data: 03/05 a 02/06

Período (Manhã/Tarde/Noite): M

Tema da Aula: Se precisar armazenar muitos valores é possível? É sim, utilizando matrizes!

Como imaginei que seria esta aula?

Novamente, precisei fazer a aula teórica completamente expositiva, pois o conceito de matrizes confunde-se com o de vetores. Precisei chamar a estrutura do vetor e resolver com os alunos dois casos distintos para que eles conseguissem perceber a diferença. Muitos alunos se perderam no conteúdo na aula teórica e não conseguiram “enxergar” o funcionamento da matriz. Na aula prática demos continuidade no estudo com a linguagem de programação, o que facilitou muito o entendimento dos alunos. Então resolvemos diversos exercícios pequenos com estruturas semelhantes para que eles compreendessem o funcionamento da matriz, bem como o posicionamento dos elementos.

Como de fato a aula ocorreu?

Conforme descrito acima.

O que mudaria? Por quê?

Talvez nessa parte do conteúdo, quando se estiver abordando vetores e matrizes, seja mais interessante inverter a estrutura da aula, primeiro ver na prática na linguagem de programação e depois montar as construções lógicas em algoritmo em sala. Fica mais fácil do aluno entender com a linguagem de programação.

O que manteria? Por quê?

N/A.

---

Turmas: Eng. Civil A e B

Data: 06 a 09/06

Período (Manhã/Tarde/Noite): M

Tema da Aula: Resolução de problemas em sala e laboratório

Como imaginei que seria esta aula?

Utilizamos a aula para resolver uma série de enunciados de problemas de programação. Pedi que escolhessem da lista dois enunciados para que eu resolvesse com eles e então trabalhariam sozinhos. Então, deixei-os em laboratório para que discutissem entre eles os enunciados e as melhores soluções.

Como de fato a aula ocorreu?

Conforme descrito acima. O que me chamou atenção foi que os alunos estavam um tanto quanto “inquietos”. Fiz a chamada e cerca de um terço da turma debandou. Perguntei para os demais o que estava havendo, parece que havia uma prova de outra disciplina (física) próxima que eles não sabiam o assunto com a profundidade necessária.

O que mudaria? Por quê?

Pensei em ver o assunto e montar um algoritmo com eles para resolver o problema, mas como não consegui falar com a professora da disciplina a tempo não deu para alinhar a resolução. Talvez seja interessante montar alguns pontos de convergência entre as disciplinas para que os alunos trabalhem multidisciplinarmente.

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 13 a 16/06

Período (Manhã/Tarde/Noite): M

Tema da Aula: Primeiro estudo de caso aplicado do 2º bimestre.

Como imaginei que seria esta aula?

Fizemos o estudo de caso da primeira atividade avaliativa com rubrica de 4 pontos. Liberei o enunciado para os alunos na segunda-feira cedo no AVA, na terça-feira eles ficaram comigo e a monitora em sala de aula para entender como resolver o problema, em seguida liberei os alunos para trabalharem nos laboratórios. A entrega da solução desenvolvida foi marcada para o dia de quarta-feira no final da noite pelo AVA.

Como de fato a aula ocorreu?

N/A

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 20 a 23/06

Período (Manhã/Tarde/Noite): M

Tema da Aula: Segundo estudo de caso aplicado do 2º bimestre.

Como imaginei que seria esta aula?

Assim como na semana anterior, fizemos o estudo de caso da segunda atividade avaliativa com rubrica de 4 pontos. No mesmo formato, liberei o enunciado para os alunos no final da tarde de domingo no AVA, na terça-feira eles ficaram comigo e a monitora em sala de aula para entender como resolver o problema, em seguida liberei os alunos para trabalharem nos laboratórios. A entrega da solução desenvolvida foi marcada para o dia de quarta-feira no final da noite pelo AVA.

Como de fato a aula ocorreu?

N/A

O que mudaria? Por quê?

N/A

O que manteria? Por quê?

N/A

---

Turmas: Eng. Civil A e B

Data: 26 a 29/06

Período (Manhã/Tarde/Noite): M

Tema da Aula: Semana de encerramento do semestre

Efetuei a correção dos dois estudos de caso com os alunos, e também da atividade em TDE. Muitos alunos esqueceram de colocar os nomes dos colegas nas submissões dos trabalhos, o que me deu bastante trabalho para localizar os alunos que estavam “perdidos” sem nota. Então, quando esses me comunicaram que estariam na equipe de “fulano e beltrano”, pedi que esse aluno me explicasse como tinham trabalhado no desenvolvimento do referido trabalho para que ter a garantia de que ele realmente pertencia a essa ou aquela equipe.

Durante as conversas com as equipes, percebi que alguns poucos alunos foram “carregados” pelos demais, que mesmo trabalhando juntos não conseguiam entender partes pontuais dos códigos-fonte, que foram construídos por outros colegas. Na próxima repetição da disciplina vou manter os estudos de caso, porém penso em montar um banco de problemas com diferentes enunciados e mesmos níveis de dificuldade. Também vou utilizar equipes somente para atividades realizadas em aulas, para as atividades avaliativas pedirei que os alunos entreguem trabalhos individualmente para garantir que quem fez o trabalho o fez sozinho melhorando o desempenho do processo de avaliação.

## **Conclusão**

Quanto ao uso das metodologias ativas de ensino no contexto dos cursos básicos de programação, esse trabalho buscou efetuar um *blend* entre a metodologia utilizada por empresas de software XP (*eXtreme Programming*) com a metodologia PBL (*Problem Based Learning*) em proporções que variaram durante o semestre. Em alguns momentos, em que foram trabalhados conteúdos mais complexos da programação, como por exemplo, o tema “matrizes”, conforme explicita o diário de bordo daquela semana, fora necessário efetuar uma aula expositiva, dada a densidade do conteúdo e a técnica com estruturas de repetição

encadeadas para acessar os elementos de uma matriz de dados.

Do ponto de vista do aproveitamento do estudante, foi possível observar que provavelmente pelo formato do ensino básico e médio aplicado no país, o jovem assume um papel passivo em sala de aula, em que o que ele quer é que o professor explique o assunto *teatralmente* e ele resolva o problema da avaliação seguindo em frente. Com o uso de métodos ativos de ensino, o foco do processo de aprendizado é dirigido ao estudante e não ao professor, que nesse caso atua como um mero “moderador” do processo em sala de aula, instigando no estudante, a responsabilidade do estudo prévio e aprendizado.

Em algumas aulas, fora necessário o estudo prévio de materiais pelos alunos, o que praticamente não ocorreu, mesmo os materiais sendo fornecidos em um formato bastante resumido, o que intitulamos durante o curso de “*Five Minutes Paper*”. Estes foram materiais desenvolvidos pelo professor para uma leitura rápida antes de realizar as atividades curtas que deveriam ser resolvidas antes das aulas. Diversas vezes foi percebido que os alunos não leram os materiais. Então, ao iniciar a aula, quando pedia-se para que todos acessassem o AVA a fim de acessar os arquivos em formato PDF, que seriam utilizados na aula, solicitava também que lessem o *five minutes paper* relativo ao conteúdo, enquanto era realizada a chamada no início da aula. Acabou que se tornou uma *práxis* das aulas a leitura rápida do material durante a chamada no início da aula, o que se avaliou como positivo.

Com essa nova estrutura de aula, mesmo com a dificuldade com lógica apresentada pelos alunos, ainda foi possível observar um aumento significativo do desempenho dos alunos na disciplina de algoritmos e programação, pois a metodologia de ensino utilizada não foca somente em trabalhar com competências de programação de computadores em si, mas no desenvolvimento do senso de equipe e do compartilhamento de conhecimentos e experiências. Essa estrutura também pode ser adaptada para outros cursos e disciplinas, pois não depende de nenhuma ferramenta específica para seu uso.

Em suma, os métodos ativos de ensino proporcionam grande melhoria e diferencial no desempenho do estudante, se esse for estudante e não um aluno passivo em sala de aula. Quando incentivados ao trabalho, a grande maioria dos alunos aderiu à estrutura proposta para



a disciplina, porém, cerca de 10% a 15% da turma gerou bastante resistência ao método, pois sempre optavam por desenvolver as atividades e tarefas sozinhos ao invés de trabalhar em grupos. Houveram dois casos de alunos que, mesmo que se deliberasse equipes para trabalharem em sala, esses ainda assim realizavam sozinhos as atividades e, após a aula, efetuavam a entrega via mensagem privada do AVA (talvez por terem tido problemas com outras equipes no passado). Mas, sob o ponto de vista da estrutura colaborativa necessária à aula, talvez fosse necessário, quando da entrada no estudante no primeiro semestre do curso, desenvolver algum tipo de sinergia entre o alunado, a fim de quebrar o “tabu” de que a atividade solitária é mais aproveitada que a troca de experiências proporcionada pela atividade coletiva.

### Referências Bibliográficas

BECK, K. **Programação Extrema** (XP) Explicada. 2<sup>a</sup> Ed. Porto Alegre : Editora Bookman, 2004. 182 p.

BIGGS, J.; TANG C., **Teaching for Quality Learning at University**. What the Student Does. Third Edition, 2007. Disponível em: [http://www.umweltbildung-noe.at/upload/files/OEKOLOG%202014/2\\_49657968-Teaching-for-Quality-Learning-at-University.pdf](http://www.umweltbildung-noe.at/upload/files/OEKOLOG%202014/2_49657968-Teaching-for-Quality-Learning-at-University.pdf) Acesso em Out/2017.

BONNELL, C. C.; EISON, J. A. **Active learning**: Creating excitement in the classroom. Washington: George Washington University Press, 1991.

BOROCHOVICIUS, E.; TORTELLA, J. C. B. Aprendizagem baseada em problemas: um método de ensino-aprendizagem e suas práticas educativas. **Ensaio: Avaliação e Políticas Públicas em Educação**, v. 22, n. 83, p. 263-294, abr./jun. 2014.

CASTELLS, M. A. **A era da informação: economia, sociedade e cultura. A Sociedade em Rede**. v. 1. 9. ed. São Paulo: Paz e Terra, 2006.

DAVIS, B.G. **Tools for teaching**. San Francisco: Jossey-Bass, 1993.

DONALD, J., Clarifying Learning, in Saroyan A. and Amundsen, C., Rethinking Teaching in Higher Education, 2004.

GUIMARÃES, J. C. F. de. Formação docente: uso de metodologias ativas como processo inovador de aprendizagem para o ensino superior. **Anais da XVI Mostra de Iniciação Científica, pós-graduação, pesquisa e extensão do Programa de Pós-Graduação em**

**Administração** – UCS. Caxias do Sul: Universidade de Caxias do Sul, 2017.

LASTRES, H. M. M. Invisibilidade, injustiça cognitiva e outros desafios à compreensão da economia do conhecimento. **In: Informação e desenvolvimento: conhecimento, inovação e apropriação social.** Brasília: IBICT, UNESCO, 2007. p. 185–212.

PAIVA, M. R. F.; *et. al.* Metodologias ativas de ensino-aprendizagem: revisão integrativa. **SANARE**, Sobral, v. 15, n. 2, p. 145-153, jun.-dez., 2016.

RIBEIRO, L. R. C. Aprendizado baseado em problemas: uma implementação na educação em engenharia na voz dos atores. **Tese (Doutorado).** Programa de Pós-graduação em Educação. Universidade Federal de São Carlos. São Carlos: UFSCAR, 2005.

SAROYAN, A.; AMUNDSEN, C. **Rethinking teaching in higher education.** Sterling, Virgínia: Stylus, 2004.