Research, Society and Development, v. 9, n.1, e74911648, 2020
(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v9i1.1648

Metodologias Ativas de Aprendizagem: um relato de experiência nas disciplinas de programação e estrutura de dados

Active Learning Methodologies: an experience report in the courses of programming and data structure

Metodologías de Aprendizaje Activo: um informe de experiencia sobre programación y disciplinas de estrutura de datos

Recebido: 10/09/2019 | Revisado: 18/09/2019 | Aceito: 23/09/2019 | Publicado: 04/09/2019

Nara Martini Bigolin
ORCID: https://orcid.org/0000-0002-7566-2514
E-mail: narabigolin@hotmail.com
Universidade Federal de Santa Maria, Brasil
Sidnei Renato Silveira
ORCID: https://orcid.org/0000-0002-4506-8522
E-mail: sidneirenato.silveira@gmail.com
Universidade Federal de Santa Maria, Brasil
Cristiano Bertolini
ORCID: https://orcid.org/0000-0002-0183-2365
E-mail: cristiano.bertolini@ufsm.br
Universidade Federal de Santa Maria, Brasil
Iara Carnevale de Almeida
ORCID: https://orcid.org/0000-0003-3587-3883
E-mail: iara.carnevale.almeida@gmail.com
Centro Universitário de Maringá, Brasil
Marlise Geller
ORCID: https://orcid.org/0000-0002-9640-2666
E-mail: marlise.geller@gmail.com
Universidade Luterana do Brasil, Brasil
Fábio José Parreira
ORCID: https://orcid.org/0000-0002-8344-0380
E-mail: fabiojparreira.ufsm@gmail.com
Universidade Federal de Santa Maria, Brasil

Guilherme Bernardino da Cunha
ORCID: https://orcid.org/0000-0002-0972-9784
E-mail: guilherme@ufsm.br
Universidade Federal de Santa Maria, Brasil
Ricardo Tombesi Macedo
ORCID: https://orcid.org/0000-0001-7469-8446
E-mail: rmacedo1987@gmail.com
Universidade Federal de Santa Maria, Brasil

Resumo

O objetivo deste artigo é o de apresentar, por meio de um relato de experiências, as ações realizadas durante os processos de ensino de aprendizagem nas disciplinas envolvendo o estudo de algoritmos, lógica de programação e linguagens de programação no Curso de Bacharelado em Sistemas de Informação da UFSM (Universidade Federal de Santa Maria) – Campus Frederico Westphalen – RS. O relato apresenta as estratégias e os resultados obtidos a partir da aplicação de metodologias ativas de aprendizagem, colocando o aluno como centro do processo, sujeito ativo na construção do seu conhecimento.

Palavras-Chave: Processos de Ensino e de Aprendizagem, Lógica de Programação, Programação de Computadores.

Abstract

The objective of this paper is presents an experience report carried out during the learning-teaching processes in the courses involving the study of algorithms, programming logic, and programming language. in the Degree in Information Systems of Federal University of Santa Maria in the city of Frederico Westphalen - RS. The report presents the strategies and results, obtained from the application of active learning methodologies, placing the student as the center of the process, an active subject in the construction of his knowledge.

Keywords. Learning and Teaching Processes, Logic Programming, Computer Programming.

Resumen

El propósito del este artículo es presentra una descripción de las experiencias realizadas durante los procesos de enseñanza del aprendizaje en las disciplinas que involucran el estudio de algoritmos, lógica de programación y lenguajes de programación en el curso de

Licenciatura en Sistemas de Información de la Universidad Federal de Santa Maria en la ciudad de Frederico Westphalen – RS. El informe presenta las estrategias y los resultados obtenidos de la aplicación de metodologías de aprendizaje activo, colocando al alumno como el centro del proceso, sujeto activo en la construcción de sus conocimientos.

Palabras clave: procesos de enseñanza y aprendizaje, lógica de programación, programación informática.

1. INTRODUÇÃO

Os processos de ensino e de aprendizagem de computação não são de conhecimento geral da população, já que estes conteúdos não são estudados na Educação Básica, apesar dos esforços da SBC (Sociedade Brasileira de Computação) para incluir os conteúdos ligados ao pensamento computacional na BNCC (Base Nacional Curricular Comum) (SBC, 2017; SBC, 2018). Quando os alunos ingressam no Ensino Superior, na área de Informática, os mesmos se deparam com a lógica de programação e chega-se a um momento crítico, gerando um alto índice de desistências. A lógica de programação, bem como o estudo de linguagens de programação, exige um esforço real e o nível de dificuldade empregado é alto. Estudar a área de programação de computadores é um dos requisitos fundamentais para os cursos de Computação (Pereira & Rapkiewicz, 2004; Garlet, Bigolin & Silveira, 2018; Souza, Silveira e Parreira, 2018).

O ensino da programação de computadores não é fácil e, devido a isso, muitas universidades discutem com frequência seus currículos dos cursos da área de Computação, em busca de alternativas para diminuir o índice de evasões deste curso. É comum observarmos pesquisas que apontam o grande número de evasões neste curso, fato que tem relação com as dificuldades de aprendizagem (Castro *et. al*, 2003; Hoed, 2017).

O contexto deste artigo envolve os processos de ensino e de aprendizagem de lógica de programação (utilizando o ambiente *VisuAlg)* e de programação de computadores (aplicando a linguagem de programação C e o ambiente *Dev-C*). O relato de experiências envolve um Curso de Bacharelado em Sistemas de Informação da UFSM (Universidade Federal de Santa Maria) – campus Frederico Westphalen/RS (Apoio Informática, 2019; Slashdot Media, 2019).

O currículo do referido curso foi reformulado no ano de 2016, sendo que existem duas disciplinas iniciais que abordam os conteúdos relacionados à lógica de programação e à programação de computadores (as disciplinas de Programação e Estruturas de Dados I e II,

ofertadas nos 1º e 2º semestres do curso). Cada uma das disciplinas tem carga horária de 120 horas. Como o curso é ofertado no turno da noite, cada uma das disciplinas ocupa duas noites na semana. Sendo o curso noturno, ofertado em uma IES (Instituição de Ensino Superior) Pública Federal, a maioria dos alunos trabalha durante o dia e tempo pouco tempo para estudar (UFSM, 2019).

O objetivo deste relato de experiência, cujo o foco são as disciplinas que envolvem o estudo de algoritmos e programação (raciocínio lógico, desenvolvimento de algoritmos e estudo de linguagens de programação), dos semestres iniciais, é o de apresentar e discutir a aplicação de uma metodologia ativa de aprendizagem, que coloca o aluno como centro dos processos de ensino e de aprendizagem, sendo sujeito protagonista na construção do conhecimento (Brenelli, 2005; Franco, 2004; Vygotsky, 2007; Pereira *et. al*, 2017; Silveira *et. al*, 2019).

2. METODOLOGIAS ATIVAS DE APRENDIZAGEM

A aplicação de metodologias ativas de aprendizagem auxilia o professor em suas atividades e beneficia os alunos, aumentando a interação e a possibilidade de aprendizagem (Silveira *et. al*, 2019).

Dentro do contexto atual, onde existe um maior envolvimento das empresas e dos empregadores na formação acadêmica, progressiva massificação e a consequente heterogeneização dos estudantes (Zabalza, 2004), faz-se necessária uma forte ligação da teoria com a prática. A articulação teoria-prática encontra, na relação entre o ensino e o mundo do trabalho, sua forma principal de concretização. Cowan (2002), coloca que a competência dos alunos é aumentada, particularmente, por métodos ativos de aprendizado que desenvolvam interesses, habilidades e experiências prévias dos aprendizes. Além disso, a capacidade de lidar com dificuldades é desenvolvida, encorajando os alunos a encontrarem soluções para problemas que identificaram pessoalmente. Fica clara a importância da existência de atividades práticas, que permitam que os alunos, alicerçados na teoria, possam colocar a "mão na massa".

Com relação às metodologias ativas de aprendizagem, podem ser aplicadas diferentes estratégias, tais como: Aprendizagem baseada em Problemas (PBL – *Problem Based Learning*), Aprendizagem baseada em Projetos (*Project Based Learning*), Sala de Aula Invertida, Aprendizagem baseada em Jogos, Gamificação, entre outras (Bergmann, 2018; Silveira *et. al*, 2019).

Bergmann (2018) coloca que nas metodologias ativas de aprendizagem deve-se inverter a taxonomia proposta por *Bloom*, deixando o trabalho mais simples para ser feito em casa e o mais complexo para ser desenvolvido em sala de aula, por meio da interação com os colegas e com o professor ou tutor. A Figura 1 apresenta a Taxonomia de *Bloom.*

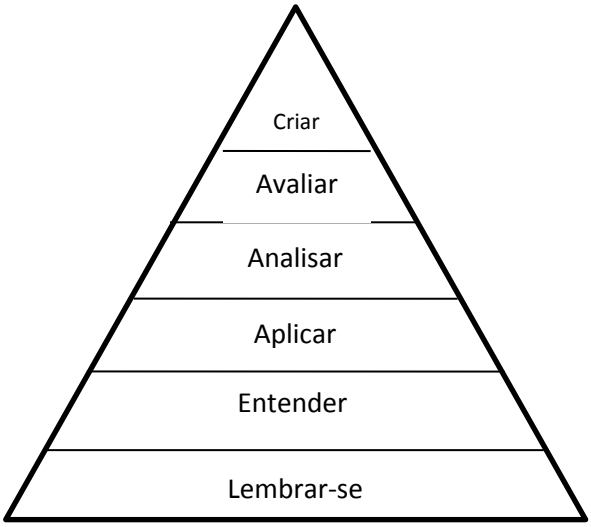

Figura 1 - Taxonomia de Bloom Fonte: (Adaptada de Bergmann, 2018)

De acordo com a Taxonomia de *Bloom* (Figura 1), a base da pirâmide envolve as atividades mais simples (lembrar-se, entender) e que podem ser realizadas em casa, no dever de casa. À medida que se sobe nos níveis da pirâmide, o nível de dificuldade vai aumentando. Estas tarefas mais complexas (aplicar, analisar, avaliar e criar) devem ser realizadas em sala de aula.

A Teoria de Aprendizagem principal que apoia o desenvolvimento das aulas de Programação e Estruturas de Dados I e II é a teoria construtivista proposta por Piaget (Franco, 2004; Pereira *et. al*, 2017; Silveira *et. al*, 2019). Na abordagem construtivista o aluno é visto como construtor do seu conhecimento. A construção do conhecimento possibilita que os alunos assimilem novos conhecimentos, a partir de conceitos já conhecidos. Essa construção envolve interação, estudo, experiência e erro. Neste sentido, os processos de ensino e de aprendizagem não podem envolver meramente atividades repetitivas, é preciso estimular os alunos a desenvolverem sua criatividade e interagirem com os colegas, com os professores e com materiais didáticos (Pereira *et. al*, 2017; Silveira *et. al*, 2019).

Segundo Piaget, o criador da teoria construtivista, o conhecimento não está no sujeito nem no objeto, mas ele se constrói na interação do sujeito com o objeto. Na medida em que o sujeito interage com os objetos é que ele produz a capacidade de conhecer e produz o próprio conhecimento (Brenelli, 2005; Franco, 2004; Pereira *et. al*, 2017; Silveira *et. al*, 2019). A

construção é realizada por meio de esquemas que cada pessoa já possui, ou seja, esquemas que foram construídos por meio da sua relação com o meio em que vive. Um esquema é um padrão de comportamento ou uma ação que se desenvolve com uma certa organização, e que consiste em um modo de abordar a realidade e conhecê-la. Existem esquemas simples, como o reflexo da sucção, presente após o nascimento, e há esquemas complexos, tais como os ligados às operações lógicas que emergem por volta dos sete anos de idade (tais como os esquemas necessários para que os alunos possam construir conhecimento sobre lógica de programação e programação de computadores).

Segundo Piaget, as chaves principais do desenvolvimento são a própria ação do sujeito e o modelo pelo qual esta ação se converte em um processo de construção interna, isto é, de formação dentro da mente de uma estrutura em contínua expansão, que corresponde ao mundo exterior (Pereira *et. al*, 2017; Silveira *et. al*, 2019).

2. O CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO DA UFSM/FREDERICO WESTPHALEN/RS

O Curso de Bacharelado em Sistemas de Informação da UFSM/Frederico Westphalen - RS foi criado no ano de 2009 e suas atividades iniciaram, efetivamente, no 2º semestre de 2010. O PPC (Projeto Pedagógico do Curso) foi reformulado no ano de 2016. Anteriormente, existiam quatro disciplinas, nos dois semestres iniciais, que tratavam dos conteúdos ligados à lógica de programação e à programação de computadores, sendo elas: no 1º semestre: Lógica e Algoritmo (60h) e Laboratório de Programação I (60h) e, no 2º semestre: Laboratório de Programação II (60h) e Estruturas de Dados (60h). Sendo assim, as disciplinas iniciais, voltadas ao estudo de programação, somavam 240h, mas eram 4 disciplinas separadas, que podiam ser ministradas por diferentes professores. Em 2016, o NDE (Núcleo Docente Estruturante) propôs uma reformulação curricular, criando duas disciplinas, cada uma com 120h. A 1ª disciplina – Programação e Estruturas de Dados I (120h) uniu os conteúdos das disciplinas de Lógica e Algoritmo e Laboratório de Programação I e a 2ª disciplina – Programação e Estruturas de Dados II, também com 120h, uniu os conteúdos das disciplinas de Laboratório de Programação II e Estruturas de Dados (UFSM, 2019).

Dessa forma, um mesmo professor assume cada uma das disciplinas, com 120h, e ministra o conteúdo sequencialmente, não havendo mais sobreposição e sobrecarga cognitiva, com acontecia anteriormente. O aluno, no currículo anterior, cursava Lógica e Algoritmo em uma noite da semana, utilizando o *VisuAlg* e, na disciplina de Laboratório de Programação I

programava na linguagem C, utilizando o ambiente *Dev-C*. Estes conteúdos, ministrados simultaneamente, acabavam confundindo os alunos e gerando uma sobrecarga cognitiva. Com a nova estrutura de disciplinas com 120h, o professor pode dividir a disciplina em duas partes, iniciando com a construção do pensamento computacional e lógica de programação e depois abordando a linguagem de programação C (UFSM, 2019).

Os conteúdos abordados nas disciplinas de Programação e Estruturas de Dados I contemplam: 1) Introdução e conceitos; 2) Lógica de programação; 3) Metodologias de projetos de programas; 4) Dados, expressões e algoritmos sequenciais; 5) Algoritmos estruturados; 6) Dados estruturados e 7) Modularização. A disciplina de Programação e Estruturas de Dados II aborda os seguintes tópicos: 1) Estruturas de dados lineares e encadeadas; 2) Árvores e 3) Grafos (UFSM, 2019).

4. METODOLOGIA APLICADA NAS DISCIPLINAS DE PROGRAMAÇÃO E ESTRUTURAS DE DADOS I E II

Métodos ativos de aprendizagem e motivação dos alunos são fatores que afetam o sucesso dos processos de ensino e de aprendizagem (Bergmann, 2018). O aprendizado de lógica de programação e de programação de computadores também é afetado por esses fatores. Definir e aplicar uma metodologia de aprendizagem adequada possibilitará que os alunos respondam positivamente, aumentado a chance de aprendizagem. Alguns alunos são muito visuais, alguns são mais auditivos e, outros, poucos atentos. Estas constatações são reforçadas na Teoria das Inteligências Múltiplas, destacando que as pessoas aprendem de formas diferentes, de acordo com diferentes habilidades.

Para Gardner, todos os indivíduos aprendem de formas diferentes e, para os professores, é preciso identificar diferentes modos de adquirir e representar conhecimento. Os indivíduos diferem na potencialidade de suas inteligências e, também, na forma como tais inteligências são invocadas e combinadas para executar diferentes tarefas, resolver problemas e progredir em diversas áreas (Gardner, 1994; Gardner, 1995; Gardner, 1998; Gardner, 2013).

Selecionar uma metodologia de aprendizagem específica ou uma forma de motivação, pode permitir que um aluno adquira habilidades de programação rápida e facilmente. Aplicando-se um método inadequado, ele pode encontrar dificuldades no aprendizado da programação. Por exemplo, o método tradicional de ensino, baseado em aulas meramente expositivas, não é adequado para os processos de ensino e de aprendizagem na área de programação de computadores.

Na abordagem construtivista, adotada neste relato de experiência, o professor deve produzir situações que favoreçam a compreensão dos alunos, de que existe um conflito entre sua ideia sobre um determinado fenômeno e a concepção cientificamente correta. Isto supõe a aplicação de uma metodologia educativa que apresente maiores dificuldades e complicações do que geralmente se quer reconhecer (Carretero, 2002). Esta metodologia não é a aula tradicional, expositiva. Isso não significa que não podemos e/ou não devemos utilizar a aula expositiva, mas que devemos utilizá-la cada vez menos. Demo (2005, p. 74) coloca que:

> "A aula expositiva pode ser resultado eminente do esforço de argumentação do professor, no qual mostra sua capacidade de elaboração e comunicação. Este tipo de aula precisa ser defendido e continua em alta. O problema é que predomina, de longe, a aula reprodutiva, fruto, geralmente, de professores mal preparados, desestimulados e cansados (...)".

Demo (2005) destaca alguns pontos importantes para que uma aula seja boa: 1) precisa ser elaborada e reconstruída: o professor precisa estudar continuamente e ministrar aula daquilo que produz; 2) o professor precisa pesquisar e elaborar seus conteúdos, ao invés de ficar apenas copiando os outros; 3) precisa ser atraente (ou pelo menos suportável); 4) não deve ser longa e 5) precisa ser envolvente.

A aula expositiva, ou aula reprodutiva é, segundo Demo (2004), o signo maior do instrucionismo, por meio da qual se 'repassa' conhecimento. Entretanto, como estamos falando em construção do conhecimento, esta prática é obsoleta, porque o cérebro é incapaz de repassar, ele interpreta e reconstrói (Becker, 2003 citado por Demo, 2004).

De acordo com uma classificação sobre os processos de aprendizagem (Jenkins, 2002), a aprendizagem é dividida em abordagem profunda e superficial. A aprendizagem profunda refere-se à compreensão de um tópico, enquanto a aprendizagem de superfície se concentra na memorização dos fatos. Analisando-se pelo ponto de vista da Taxonomia de *Bloom* (destacada anteriormente, na seção 2 deste artigo), a aprendizagem superficial é a base da pirâmide (Bergmann, 2018). O que queremos buscar é chegar ao topo da pirâmide, a criatividade. Na proposta de estudar programação de forma superficial, os processos de ensino e de aprendizagem podem ser usados para memorizar a sintaxe da linguagem de programação. As duas abordagens são importantes. Entretanto, a aprendizagem profunda é crucial para obter uma verdadeira compreensão da lógica de programação lógica e, consequentemente, uma verdadeira competência em programação (o topo da pirâmide da Taxonomia de *Bloom*).

Outro fator importante é a motivação, que afeta a eficiência da aprendizagem. O professor precisa estar motivado e motivar constantemente os alunos, certificando-se de que eles realmente se envolvam em tarefas que ele criou. Este é um aspecto destacado nas metodologias ativas de aprendizagem. A motivação envolve o relacionamento com os alunos, afetividade, comprometimento, engajamento, estimulando a interação entre alunos e professor e entre os alunos (Bergmann, 2018).

Neste contexto, criou-se uma metodologia ativa de aprendizagem para as disciplinas de Programação e Estruturas de Dados I e II, de forma proativa. A metodologia consiste em propor aos alunos que apresentem um seminário de um conteúdo ainda não apresentado pelo professor (nos moldes da *Problem Based Learning*) (Silveira *et. al*, 2019). Assim, antes de cada aula em que será abordado um conteúdo novo, os alunos apresentam sua percepção em relação àquele assunto no formato de um seminário. Durante a apresentação o aluno é questionado pelo professor sobre o assunto, para que o mesmo reflita sobre o que está apresentando. Esse procedimento é repetido para todos os grupos (o seminário pode ser em grupo ou individual). O uso de *slides* não é avaliado, apenas o conteúdo explicado pelo aluno. Cada aluno desenvolve o assunto de maneira integral. O questionamento do professor é utilizado para clarear o conteúdo para o aluno.

Os resultados apontam que, nas turmas em que esta metodologia ativa foi aplicada, houve um maior número de aprovações como apresenta o gráfico da Figura 2, onde são mostrados os números de aprovações e reprovações nas disciplinas iniciais de programação (Programação e Estruturas de Dados I e II), ofertadas nos 1º e 2º semestres do currículo do curso, no período compreendido entre 2014/2 e 2019/1 (números absolutos). Cabe destacar que, cada uma dessas disciplinas é ofertada somente uma vez por ano.

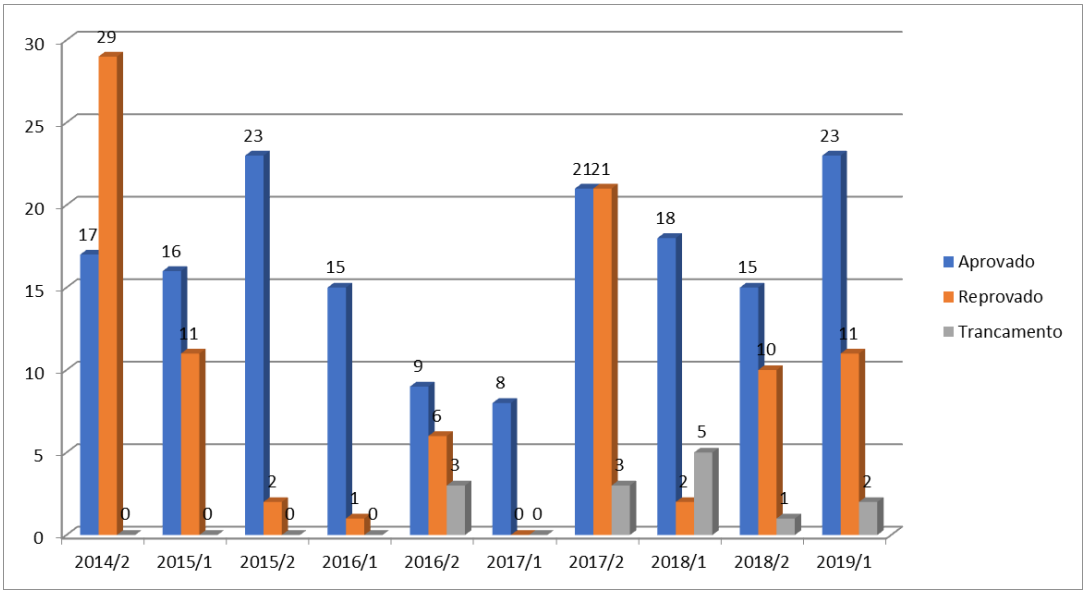

Figura 2 – Dados das turmas de 2014/2 a 2019/1 (Fonte: os autores, 2019)

A aplicação desta metodologia ativa de aprendizagem foi motivada a partir de 2014/2, pois houve um alto índice de reprovação neste semestre (como mostram os dados do gráfico apresentado na Figura 2). Nos semestres seguintes, o método foi utilizado e os resultados quanto à aprovação foram muito positivos. Em 2016/2 (a metodologia ativa não foi utilizada, e o número de reprovações voltou a aumentar. No semestre seguinte, a metodologia ativa voltou a ser utilizada, para verificar se a mesma realmente tinha impacto no percentual de aprovações. Mais uma vez se mostrou muito eficaz. No semestre 2017/2, a metodologia não foi aplicada novamente, e o número de reprovações voltou a subir. Nos 3 últimos semestres (2018/1, 2018/2 e 2019/1), a metodologia foi utilizada e o sucesso foi comprovado e, a partir de agora, será adotada nos semestres seguintes.

Os alunos, por meio da avaliação do processo acadêmico (parte integrante da avaliação institucional, realizada semestralmente) consideram a disciplina muita satisfatória. Mesmo os alunos que reprovam reconhecem que não executaram as atividades da maneira proposta, sendo esse o motivo da reprovação. A avaliação da disciplina pelos alunos é sempre positiva, alcançando índices superiores a 90%.

A programação é uma atividade que envolve abstração e pensamento lógico. Desta forma, nas disciplinas envolvendo o estudo de lógica e programação de computadores no Curso de Bacharelado em Sistemas de Informação, observa-se que a maior dificuldade é a de ministrar as disciplinas em turmas numerosas, com mais de 30 alunos e turmas muito heterogêneas. Em 2017/2 (conforme gráfico da Figura 2) a turma tinha 45 alunos e houve um grande percentual de reprovações e de desistências na turma (como mostram os dados, em

percentuais, do gráfico da Figura 3). Nesse semestre não foi possível fazer um trabalho motivacional e um acompanhamento individual. Em 2018/1 eram apenas 20 alunos na turma (descontados os trancamentos), não havendo nenhuma reprovação por nota. Os dados demonstram que, em turmas pequenas, o aproveitamento é muito melhor, pois o professor consegue fazer um trabalho mais interativo e individualizado (Bergmann, 2018).

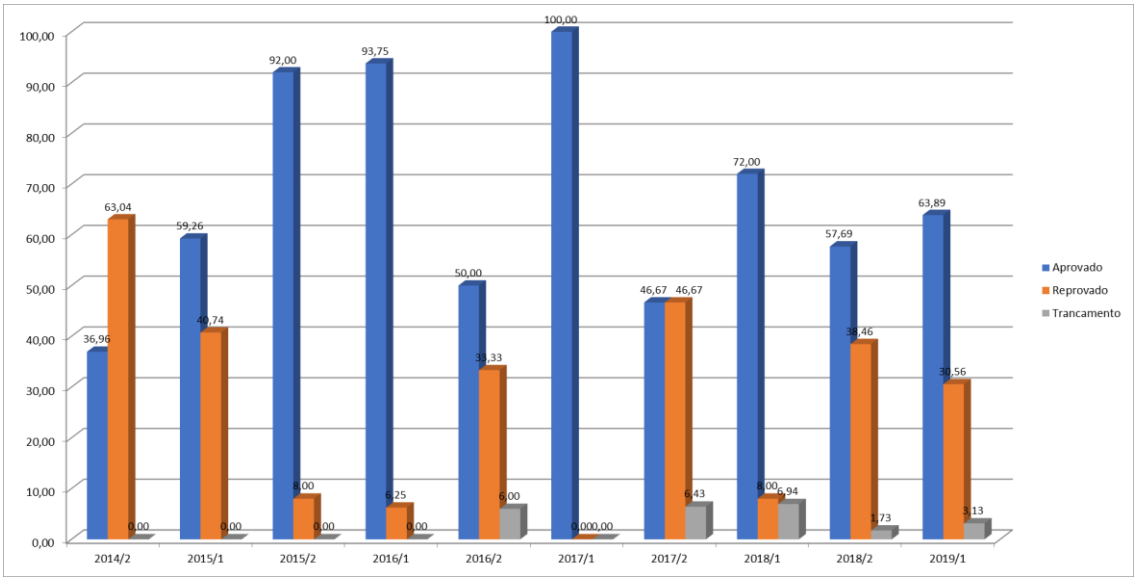

Figura 3 – Dados das turmas de 2014/2 a 2019/1 em Percentuais (Fonte: os autores, 2019)

As potencialidades identificadas no ensino de programação, conforme este relato de experiência, utilizando metodologias ativas de aprendizagem, envolvem a satisfação dos alunos com os resultados da sua aprendizagem, evidenciados na avaliação do processo acadêmico da disciplina. A aprendizagem, por meio dos seminários, torna-se significativa, ou seja, deixa de ser meramente repetitiva (modelo instrucionista) e torna-se uma construção onde o aluno é o protagonista.

Um dos principais desafios no ensino de programação de computadores no Curso de Bacharelado em Sistemas de Informação é o de atuar em turmas com um grande número de alunos, o que dificulta o acompanhamento individual, necessário para avaliar o ritmo de aprendizagem dos alunos, especialmente quando são aplicadas metodologias ativas de aprendizagem.

5 CONSIDERAÇÕES FINAIS

Apesar das dificuldades nas disciplinas iniciais de programação, o índice de evasão do Curso de Bacharelado em Sistemas de Informação encontra-se abaixo do limite da área de Computação, conforme trabalho apresentado por Hoed (2017). Sabe-se que a maior dificuldade dos alunos é com relação às disciplinas que envolvem o estudo de Lógica de Programação e Programação de Computadores (Garlet, Bigolin & Silveira, 2018; Souza, Silveira & Parreira, 2018).

Com relação à evasão, a média do Curso de Sistemas de Informação da UFSM/FW é de 14,95%, como mostram os dados do Quadro 1. Segundo Hoed (2017), nas Instituições de Ensino Superior Públicas brasileiras, a média da evasão em Cursos de Sistemas de Informação é de 15,1% ao ano. No ano de 2018 a taxa de evasão foi de 14,92% (menor do que nos anos de 2016 e 2017).

Quadro 1 – Dados da Evasão

	2010		2011		2012		2013	
Evadidos/ Cancelamentos	3 8,12%		8 10,39%		24 20%		24 24%	
	M	F	M	F	M	F	M	F
	2	1	5	3	19	5	20	4
	66,66%	33,33%	62,5%	37,5%	79,16 %	20,84%	83,33%	16,67 %

2014		2015		2016		2017		2018	
17 11,33%		18 11,25%		24 17,26%		28 17,28%		27 14,92%	
M	F	M	F	M	F	M	F	M	F
12	5	16	2	21	3	23	5	22	5
70,58%	29,42%	88,88 %	11,12%	87,5%	12,5%	82,14 %	17,86%	81,48%	18,52 %

Fonte: dos autores, 2019 (com base nos dados oriundos do SIE – Sistema de Informações Acadêmicas da UFSM)

Os conhecimentos da área de Computação, de acordo com a SBC (2018) podem ser organizados em 3 eixos: Pensamento Computacional, Cultura Digital e Mundo Digital. O pensamento computacional refere-se à capacidade de sistematizar, representar, analisar e resolver problemas por meio da construção de *algoritmos*. Este é o foco das disciplinas iniciais de programação de computadores segundo o PPC do curso (UFSM, 2019).

A partir do relato de experiência da metodologia aplicada nas disciplinas introdutórias de lógica e de programação de computadores, outros professores sentiram-se motivados a inovar em suas aulas, o que poderá representar a continuidade deste estudo em trabalhos futuros. Sendo assim, o Curso de Sistemas de Informação já organizou 11 edições do *Workshop de Qualificação Docente*, momento no qual os docentes se reúnem e discutem

sobre suas práticas pedagógicas e, recentemente, um grupo de docentes iniciou um projeto de pesquisa (um estudo de caso), envolvendo a aplicação de diferentes metodologias de aprendizagem em seu fazer pedagógico. O relato de experiências, aqui apresentado, limitou-se a abordar a aplicação de metodologias ativas de aprendizagem nas disciplinas iniciais de Programação e Estruturas de Dados. Como trabalhos futuros, propõem-se, entre outras possibilidades, o levantamento de outras ações pedagógicas que estão sendo realizadas no curso, bem como a inserção de metodologias ativas de aprendizagem em todas as disciplinas do mesmo.

Referências

Apoio Informática. *VisuAlg*. *Site da Ferramenta VisuAlg*, 2019. Disponível em: <https://www.apoioinformatica.inf.br/produtos/visualg>. Acesso em junho de 2019.

Bergmann, J. *Aprendizagem Invertida para resolver o Problema do Dever de Casa.* Porto Alegre: Penso, 3018.

Brenelli, R. P. *O Jogo como Espaço para Pensar:* a construção de noções lógicas e aritméticas. Campinas, São Paulo: Papirus, 2005.

Carretero, M. *Construtivismo e Educação*. Porto Alegre: Artes Médicas, 2002.

Castro, C. T., Castro Júnior, A., Meneses, C. B. M. e Rauber, M. Utilizando Programação Funcional em Disciplinas Introdutórias de Computação, In: *Anais do XI Workshop de Educação em Computação* – WEI, Campinas/SP, 2003.

Cowan, J. *Como ser um Professor Universitário Inovador*: reflexão na ação. Traduzido por Costa, R.C. Porto Alegre: Artmed, 2002.

Demo, P. *Aprendizagem no Brasil*: ainda muito por fazer. Porto Alegre: Mediação, 2004.

Demo, P. *Universidade, Aprendizagem e Avaliação:* horizontes reconstrutivos. 2. ed. Porto Alegre: Mediação, 2005.

Franco, R. K. *O Construtivismo e a Educação.* 4. ed. Porto Alegre: Medição, 2004.

Gardner, H. *Estruturas da Mente:* A Teoria das Inteligências Múltiplas. Porto Alegre: Artes Médicas Sul, 1994.

Gardner, H. *Inteligências Múltiplas:* a teoria na prática. Porto Alegre: Artes Médicas, 1995.

Gardner, H. *Inteligência:* Múltiplas Perspectivas. Porto Alegre: Artes Médicas, 1998.

Gardner, H. *Mentes que Lideram:* uma anatomia da Liderança. Campus Elsevier, 2013.

Garlet, D.; Bigolin, N. M.; Silveira, S. R. Ensino de Programação de Computadores na Educação Básica: um estudo de caso. *Resiget* – Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica. v. 9, n. 2, 2018. Disponível em: <http://periodicos.unifacef.com.br/index.php/resiget/article/view/1604>. Acesso em setembro, 2019.

Hoed, R. M. *Análise da Evasão em Cursos Superiores*: o caso da evasão em cursos superiores da área de computação. Brasília: UnB – Programa de Pós-graduação em Computação Aplicada. (Dissertação de Mestrado), 2017. Disponível em: <http://repositorio.unb.br/handle/10482/22575>. Acesso em março, 2018.

Jenkins, T. On the Difficulty of Learning to Program. *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, Lough-borough, UK, 2002, pp. 53-58

Pereira, J. C. R., Rapkiewicz, C. *O Processo de Ensino-Aprendizagem de Fundamentos de Programação:* Uma Visão Crítica da Pesquisa no Brasil, Anais do WEI RJES, 2004.

Pereira A. S.; Parreira, F. J.; Silveira, S. R.; Bertagnolli, S. C. Metodologia da Aprendizagem em EaD. Santa Maria: UFSM/NTE/UAB, 2017. Disponível em: <https://nte.ufsm.br/images/identidade_visual/Metodologiaaprendizagem.pdf>. Acesso em junho, 2018.

SBC. Sociedade Brasileira de Computação. *Referenciais de Formação em Computação:* Educação Básica, 2017. Disponível em: <http://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/1166-referenciais-de-formacao-em-computacao-educacao-basica-julho-2017>. Acesso em maio, 2019.

SBC. Sociedade Brasileira de Computação. Diretrizes para o Ensino de Computação Básica. *Documento Interno da Comissão de Educação Básica da SBC,* 2018.

Silveira, S. R.; Parreira, F. J.; Bigolin, N. M.; Pertile, S. L. Metodologia do Ensino e da Aprendizagem em Informática. Santa Maria: UAB/NTE/UFSM, 2019.

Slashdot Media. *Dev-C++. Site da IDE Dev-C++.* Disponível em: <https://sourceforge.net/projects/orwelldevcpp/>. Acesso em junho, 2019.

Souza, N. G.; Silveira, S. R.; Parreira, F. J. Proposta de uma Metodologia para Apoiar os Processos de Ensino e de Aprendizagem de Lógica de Programação na Modalidade de Educação a Distância. Revista ECCOM, v. 9, n. 18, 2018. Disponível em: <http://fatea.br/seer3/index.php/ECCOM/article/view/851/856>. Acesso em junho, 2018.

UFSM – Universidade Federal de Santa Maria. *Projeto Pedagógico do Curso de Bacharelado em Sistemas de Informação.* Disponível em: <https://www.ufsm.br/cursos/graduacao/frederico-westphalen/sistemas-de-informacao/>. Acesso em setembro, 2019.

Vyogtsky, L. *A Formação Social da Mente*. São Paulo: Martins Fontes, 2007.

Zabalza, M. A. *O Ensino Universitário:* seu cenário e seus protagonistas. Traduzido por Rosa, E. Porto Alegre: Artmed, 2004.

Porcentagem de contribuição de cada autor no manuscrito

Nara Martini Bigolin – 30%

Sidnei Renato Silveira – 10%

Cristiano Bertolini – 10%

Iara Carnevale de Almeida – 10%

Marlise Geller – 10%

Fábio José Parreira – 10%

Guilherme Bernardino da Cunha - 10%

Ricardo Tombesi Macedo - 10%