



XII SEMANA CIENTÍFICA UNILASALLE – SEFIC 2016
Canoas, RS – 17 a 21 de outubro de 2016

COMUNICAÇÃO ORAL

ISSN 1983-6783

DE FLUXOGRAMAS A LINGUAGENS DE ALTO NÍVEL: ESTRATÉGIAS PARA APRENDER PROGRAMAÇÃO

Fabília Py Tortelli Noronha, Douglas Vaz, Vera Lúcia Felicetti (orient.), Luciana Backes (coorient.)
Centro Universitário La Salle - UNILASALLE

Resumo

Este artigo discute estratégias de ensino de programação, levando em conta os estilos de aprendizagem dos alunos. Por meio de um relato de experiência de práticas utilizadas pelos autores, apresenta variadas formas de trabalhar conceitos de programação com alunos de ensino médio e superior.

Palavras-chave: *Aprendizagem, Programação.*

Área Temática: PPG em Educação

1. Introdução

O acesso cada vez mais rápido à informação, o acelerado desenvolvimento científico e tecnológico e o novo perfil do profissional para o mercado de trabalho geram movimentos de perturbação e transformação no processo de ensino desenvolvido pelas instituições educacionais.

Segundo Antunes (2010, p. 13), o papel da escola, entre os vários que a ela competem é o de “central estimuladora da inteligência”. Nessa perspectiva, o professor deve se colocar como um articulador entre o estudante e o conhecimento, de forma que o estudante aprenda a pensar, questionar e construir o conhecimento de forma ativa, não sendo apenas um mero receptor passivo de informações (BULGRAEN, 2010). Ainda, de acordo com Antunes (2010, p. 97-98),

[...] em síntese, o papel do novo professor é o de usar a perspectiva de como se dá a aprendizagem, para que, usando a ferramenta dos conteúdos postos pelo ambiente e pelo meio social, estimule as diferentes inteligências de seus alunos e os leve a se tornarem aptos a resolver problemas ou, quem sabe, criar ‘produtos’ válidos para seu tempo e sua cultura.

Outra possibilidade, é contextualizar a metodologia pedagógica às necessidades e formas de aprender dos alunos, de modo a configurar um ambiente de aprendizado relevante, tanto para os alunos, quanto para os professores.

Diante da problemática inicial, este artigo intenta realizar uma reflexão acerca dessas possibilidades, com o objetivo de criar um ponto de partida para discussões sobre as diferentes metodologias pedagógicas de programação, levando em consideração os estilos de aprendizagem dos alunos, que podem ter unicamente um estilo ou uma combinação deles.

Por meio de relatos e percepções dos autores deste texto, serão apresentadas estratégias pedagógicas para aprender programação, desenvolvidas com alunos de ensino médio integrado ao curso técnico de informática e de ensino superior de informática, utilizando desde fluxogramas até a linguagem de alto nível propriamente dita.



XII SEMANA CIENTÍFICA UNILASALLE – SEFIC 2016
Canoas, RS – 17 a 21 de outubro de 2016

COMUNICAÇÃO ORAL

ISSN 1983-6783

2. O ensino de linguagens de programação

O ensino de linguagens de programação em diversas instituições costuma seguir um roteiro pré-definido: através de livros, apostilas e tutoriais *online*, alunos e professores constroem seus primeiros códigos em programas computacionais desenvolvidos para esta finalidade. Um exemplo de primeiro programa que costuma ser criado pelos alunos, em diversas linguagens, é o conhecido “Hello World”, que aparece como exemplo na maioria dos livros e *sites* que ensinam programação.

Segundo Fontes e Silva (2008, p. 86),

Essa metodologia de ensino talvez fosse satisfatória quando os programas de computadores apresentavam uma interface muito simples com o usuário, por meio de telas com informações apenas sob a forma de textos, e os objetivos em termos de programação eram muito limitados. Isso, contudo, mudou. O aluno que estiver estudando sob essa metodologia não terá oportunidades de aprendizagem para desenvolver sua capacidade de compreensão e abstração do pensamento lógico-computacional, pois a literatura técnica disponível emprega exemplos prontos, distantes de sua realidade.

Busca-se, portanto, novas formas de ensinar programação, utilizando técnicas que levam em consideração os estilos de aprendizagem dos alunos, a partir dos estilos da família VAKT: visual, auditivo, cinestésico e tátil. Estilos de aprendizagem, são as preferências ou facilidades de cada indivíduo no que se refere a condições, modos, preferências de aprendizagem, etc. Segundo Kolb (1984 apud Bariani 1998, p. 37), um “[...] estilo de aprendizagem é derivado das interações entre o indivíduo e o ambiente”.

De acordo com Felicetti (2011),

Na maioria das instituições educacionais é dado o mesmo enfoque de ensino a todos os alunos, parece que todos aprendem e/ou têm as mesmas habilidades para aprender, porém cada ser humano é único, a sequência de DNA (*Desoxirribo Nucleic Acid*) comprova isso, logo cada indivíduo tem características e particularidades que são só dele, suas diferenças, seu modo de aprender, compreender e entender colaboram para sua aprendizagem, para sua formação, no contexto escolar ou fora dele. Assim, parece que o aluno não aprende, não por falta de habilidades para tal, mas sim por não conhecê-las, logo não pode utilizá-las. (FELICETTI, 2011, p. 78).

Ao combinar diversas formas de ensinar, aumentamos as chances de aprendizagem de um número maior de alunos. Dunn apud Felicetti aponta que “[...] uma estratégia que é extremamente eficaz para um estudante, pode ser apenas aceitável para outro, pode ser adequada para um aluno, não interessante para outro, pode ser irritante, agradável ou desastrosa”. (FELICETTI, 2011, p. 79). Nesse sentido, a pluralidade de representação do conhecimento, proporcionada por diferentes estratégias, objetiva potencializar o processo de aprendizagem e a construção do conhecimento de algoritmos.

2. 1 Introduções aos algoritmos

Logo no primeiro contato com a turma, é de costume perguntar aos alunos quem sabe definir um algoritmo. Após vários palpites, surgem alguns relacionados à área da matemática (pela proximidade com a palavra “logaritmo”) e, assim, definem um algoritmo como uma fórmula ou equação. Além desses palpites, outro que se aproximou do conceito exato foi de que se tratava de uma “receita”.

De acordo com Lopes (2010), um algoritmo é “[...] uma sequência finita e ordenada de passos (regras), com um esquema de processamento que permite a realização de uma tarefa

(resolução de problemas, cálculos etc.)”. Segundo a definição do autor, constatou-se que a ideia dos alunos sobre algoritmos tinha fundamento, visto que, foi explicado, que algoritmos não envolvem apenas aspectos computacionais. Como exemplo, cita-se: a receita de um bolo, um manual de instruções e a troca de uma lâmpada.

Outro ponto importante na definição de um algoritmo é que os alunos compreendam que duas perguntas precisam ser respondidas: a primeira se a resolução para determinado problema ou função possui uma sequência de instruções finitas e ordenadas. E, se ao final desta sequência, o objetivo foi atingido. Se ambas as respostas forem sim, teremos um algoritmo, seja ele computacional ou natural.

Essa etapa da disciplina privilegia os alunos que possuem mais facilidade de aprender através de diálogos e da construção do conhecimento de forma colaborativa, ou seja, alunos que possuem estilo de aprendizado auditivo, pois neste primeiro momento trabalha-se apenas com conversação e explanação de conceitos no quadro e projeção.

2.2 Jogos de raciocínio lógico

O foco das disciplinas de introdução aos conceitos de programação, como mencionado anteriormente, é voltado para a criação de algoritmos que no futuro serão codificados em diferentes linguagens. Por terem objetivos comuns, que contemplam a utilização de linguagens de alto nível, com maior grau de dificuldade, opta-se por introduzir o assunto de maneira que as dificuldades surjam de forma gradativa, começando, então, por formas mais simples de se desenvolver programação.

Nesse sentido, costuma-se incentivar o uso de jogos de raciocínio lógico, que estimulam a reflexão dos alunos para resolver determinado problema. Um exemplo de exercício de raciocínio lógico utilizado para introduzir o ensino de programação pode ser observado na figura abaixo:

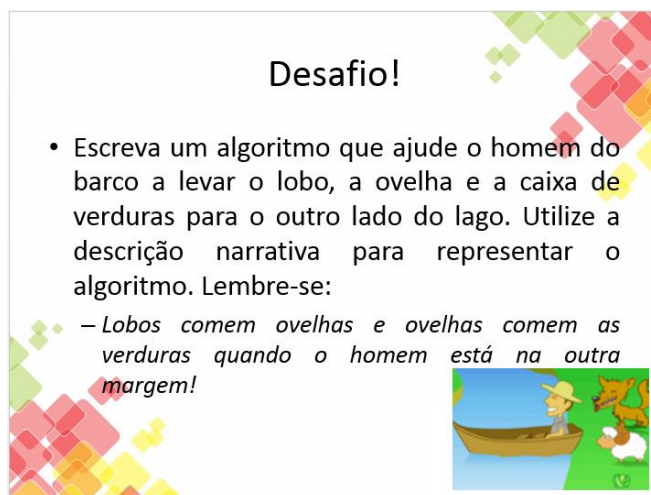


Figura 1: Primeiro desafio mostrado aos alunos. Fonte: autoria própria, 2014.

Após tentarem resolver o desafio por meio de texto, registrando no caderno ou bloco de anotações, é disponibilizada a versão digital do jogo para que continuem exercitando até completar o desafio. Nota-se que, um número maior de alunos consegue resolver o desafio com o auxílio do jogo no computador.

Após o jogo da travessia, outro semelhante é apresentado aos alunos: o jogo “Missionários e Canibais”, com proposta semelhante ao anterior. A regra deste jogo é atravessar todos os missionários para o outro lado, não os deixando, em momento algum, em menor número que os canibais. Este jogo obriga os alunos a refletir sobre o objetivo da atividade, criando estratégias e conseguir resolver o desafio através diversas tentativas. A figura 2, abaixo, mostra a tela principal do desafio.



Figura 2: Jogo “Missionários e Canibais”. Fonte: Racha Cuca, 2015.

A escolha de jogos de raciocínio lógico para introduzir conceitos básicos de programação veio da necessidade de apurar a análise e resolução de problemas por parte dos alunos, habilidade indispensável para a criação de algoritmos e estimulada a partir destes desafios. Para Vygotsky (1994, p. 65), o brincar e a interação com os jogos possibilitam às crianças aprenderem regras e sujeitarem-se às ações impulsivas pela via do prazer. Além de trazer uma história que contextualiza o conhecimento em questão, estabelecendo relação entre o concreto (situação) e o abstrato (conteúdo). Embora os alunos aqui mencionados já não se tratem mais de crianças, a ideia de Vygotsky é válida para tal grupo pois remete a infância e ao prazer de aprender de forma lúdica.

Em relação aos estilos de aprendizagem, este tipo de aprendizado está mais relacionado ao modo de aprender de alunos do estilo cinestésico, pois os mesmos aprendem executando, sendo capazes de guiar-se pela experiência motora e cognitiva. Segundo Felicetti (2011, p. 90) as atividades cinestésicas “[...] devem ser utilizadas em experiências iniciais de aprendizagem. Envolver todo o movimento do corpo e/ou experiência da vida real, exige dos participantes a ação de aprender de forma fácil e divertida.”

2.3 Fluxogramas

Utilizar fluxogramas para o ensino da lógica de programação tem o intuito de inserir os conceitos básicos de programação através de uma estrutura mais sintética e didática de visualizar o conhecimento. Deste modo, pretende-se privilegiar os alunos que possuem estilo de aprendizagem predominantemente visual, por intermédio de mapas conceituais do tipo fluxograma.

Um fluxograma, segundo Slack et al. (1997), é uma técnica de mapeamento que permite o registro de ações de algum tipo e pontos de tomada de decisão que ocorrem no fluxo real. De acordo com Tavares,

Ele organiza a informação de uma maneira linear. Ele é utilizado para mostrar passo a passo determinado procedimento, e normalmente inclui um ponto inicial e outro ponto final. Um fluxograma é normalmente usado para melhorar a performance de um procedimento. (TAVARES, 2007, p. 75).

Para Tavares (2007, p. 76), fluxogramas são fáceis de ler, pois as informações estão organizadas de uma maneira lógica e sequencial. Em informática, um fluxograma pode ser visto como um diagrama para a representação de um algoritmo.

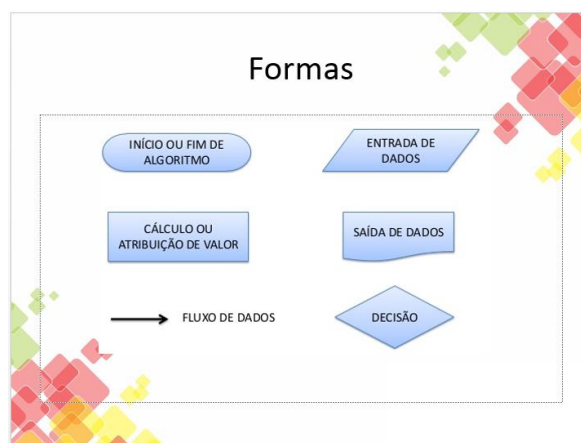


Figura 3: Formas geométricas utilizadas nos fluxogramas. Fonte: Autoria Própria, 2014.

Como mostra a figura 3, foram apresentadas aos alunos algumas formas geométricas que, ao serem unidas, podem resolver problemas lógicos, beneficiando os alunos que têm estilo de aprendizagem visual. Esses alunos aprendem sendo capazes de fazer uma imagem imediata do que estão visualizando e recebendo como informação.

Ao relacionar figuras a ações de um programa, o aluno considerado visual garante uma facilidade maior em criar sequências lógicas por meio do jogo de figuras, do que de outra forma, como escrevendo o código, por exemplo. Este método, quando impresso, facilita também a compreensão de alunos com estilo tátil e cinestésico, percebendo a construção do programa de maneira mais clara e objetiva.

2.4 Português Estruturado

Outra etapa do ensino de programação se dá com a aplicação da pseudolinguagem denominada Português Estruturado, também conhecida como Portugol. Fernandes (2011), define o Português Estruturado como uma forma especial de linguagem, bem mais restrita que o Português e com significados bem definidos para todos os termos utilizados nas instruções.

Apesar do Português Estruturado ser bastante simplificado, por trabalhar com pseudocódigos, possui todos os elementos básicos e uma estrutura semelhante à de uma linguagem típica de programação de computadores. Outro aspecto importante, é o fato de ser em português, o que facilita sua compreensão pelos alunos, pela familiaridade em relação a língua.

Essa pseudolinguagem é utilizada para a realização de exercícios e trabalhos. Na figura 5, abaixo, podemos visualizar a resolução feita pelos alunos em Português Estruturado de um mesmo algoritmo que anteriormente já foi desenvolvido em fluxograma.

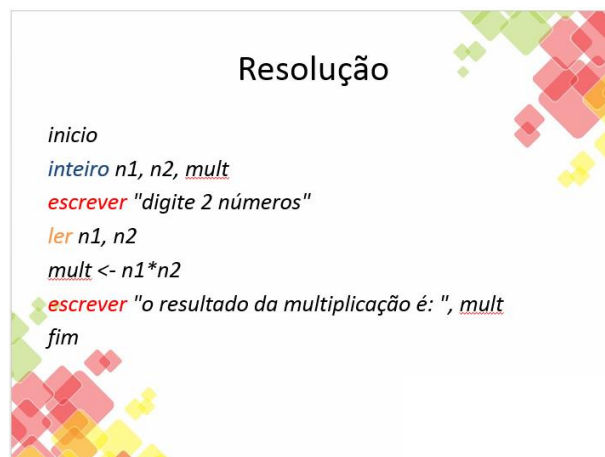


Figura 5: Resolução de exercício utilizando Português Estruturado. Fonte: Autoria Própria, 2014.

O Portugal, segundo Souza (2009, p. 1), permite que os alunos se concentrem inicialmente na solução do problema proposto sem ter que dominar a sintaxe de uma linguagem de programação de alto nível, mas ao mesmo tempo praticando conceitos como declaração de variáveis, linha de código, palavras-chave, comentários, dentre outros, que serão posteriormente transpostos com mais facilidade para ambientes reais de desenvolvimento. Nessa metodologia, assim como acontece com os fluxogramas, são privilegiados os alunos que têm maior facilidade visual.

2.5 Visualg

Segundo o site Apoio Informática¹, o VisuAlg é um software de grande utilidade para iniciantes em programação. Nele o estudante pode simular o que acontece na tela do computador, permitindo exercitar seus conhecimentos num ambiente próximo ao de uma linguagem de alto nível, através da execução dos comandos do Português Estruturado.

Abaixo, a figura 6 ilustra a interface inicial do VisuAlg.

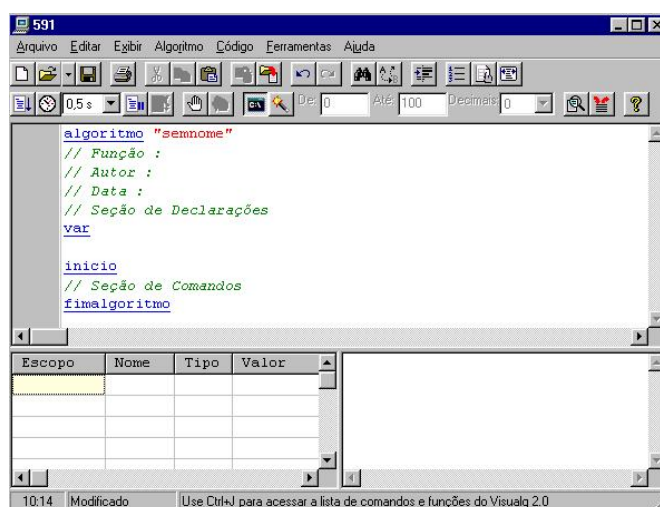


Figura 6: Interface do VisuAlg. Fonte: Apoio Informática

¹ <http://www.apoioinformatica.com.br>

Nesta etapa, o estudante utiliza o Português Estruturado, trabalhado anteriormente no caderno, e, agora, com o apoio do *software* tem a possibilidade de acompanhar a execução do algoritmo passo a passo. Assim, são privilegiados os alunos que têm maior facilidade de aprendizado visual e cinestésica provocada pela viabilidade de observar e colocar em prática o que antes era percebido de forma abstrata e estática no caderno.

3.6 SCRATCH

O Scratch é uma linguagem de programação visual que tem por objetivo tornar o aprendizado de programação mais fácil e mais divertido (MARJI, 2014).

Nesta etapa os alunos desenvolvem algoritmos baseados em blocos, isto é, ao invés de utilizar linhas de comandos os alunos arrastam e encaixam blocos coloridos que representam comandos, formando o algoritmo (programa), conforme mostra a figura 8.

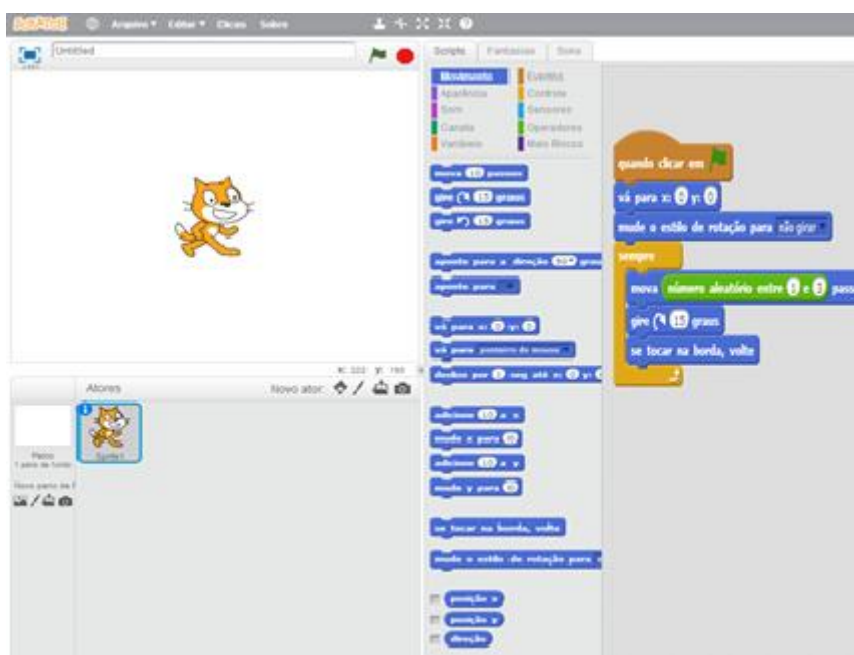


Figura 8: Scratch. Fonte: Capturado pela autora (Scratch)

Desta forma, são privilegiados os alunos que têm mais facilidade visual. De acordo com Felicetti (2011, p. 89) “Muitos alunos, quando visualizam têm uma melhor aprendizagem, efetivam um resumo, não com palavras, mas com imagens que veem quando leem, isso ajuda a pensar [...]”.

3.7 Linguagem Java

Após trabalhar algoritmos com as diferentes formas de aplicação utilizando desde jogos, fluxogramas, Português Estruturado, VisuAlg e Scratch os alunos já têm a base necessária para construir seus primeiros programas em uma linguagem mais avançada. No caso do curso superior, os alunos utilizaram linguagens mais avançadas em disciplinas específicas, já no curso técnico

integrado, a linguagem padrão é Java, linguagem de alto nível utilizada em todos os cursos técnicos e superiores de informática da instituição.

O *software* utilizado com a turma para desenvolver Java foi o Netbeans, pela sua apresentação mais amigável e pelas dicas dadas pelo programa em caso de erros de sintaxe. A figura 6 mostra a interface do Netbeans, utilizada pelos alunos.

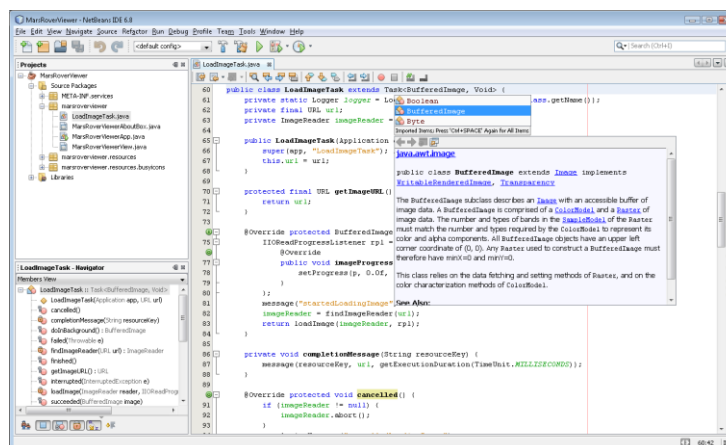


Figura 6: Tela principal do software Netbeans. Fonte: NetBeans.org, 2015.

Observou-se, a partir de então, facilidade no aprendizado da linguagem, pois os alunos já tinham capacidade de abstrair seus diversos conceitos e, também, atingiram o nível de raciocínio necessário para a criação de programas de baixa e média dificuldade, devido à preparação realizada anteriormente com outros estímulos e técnicas de aprendizagem. Assim, são privilegiados os alunos que têm maior facilidade de aprendizado de forma visual.

3. Metodologia

A problemática deste artigo surgiu da necessidade de se fazer uma análise das estratégias de ensino aplicadas no aprendizado de programação nos cursos técnico integrado e superior em informática do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - IFRS, dos *campi* Osório e Porto Alegre. Assim, promover um aprendizado em que o aluno atribua significado ao conteúdo estudado nas disciplinas básicas de programação, na tentativa de superar as dificuldades enfrentadas pelos iniciantes em programação.

A disciplina de Programação I e a disciplina de Lógica de Programação são componentes curriculares, respectivamente dos cursos técnico integrado e superiores da área de informática do IFRS. Nessas disciplinas são aplicadas as técnicas básicas e fundamentais para dar início ao aprendizado de programação, ou seja, construir algoritmos.

Para este relato de experiência utilizaremos as experiências dos autores nas turmas de primeiro ano do Curso Técnico em Informática Integrado ao ensino médio e dos primeiros semestres do curso superior Tecnólogo em Sistemas para Internet. Nesse período dos respectivos cursos, ocorre o primeiro contato dos alunos com algoritmos.

No curso técnico integrado a carga horária da disciplina é de 80 horas e contempla desde a introdução aos algoritmos até a codificação na linguagem de alto nível Java. Já no curso superior, a disciplina de introdução contempla somente algoritmos e linguagem de programação Java é trabalhada em uma disciplina específica no decorrer do curso.



XII SEMANA CIENTÍFICA UNILASALLE – SEFIC 2016
Canoas, RS – 17 a 21 de outubro de 2016

COMUNICAÇÃO ORAL

ISSN 1983-6783

Tanto no curso técnico quanto no curso superior, essas disciplinas têm como objetivos: desenvolver o raciocínio lógico por meio da construção e modelagem de algoritmos; desenvolver pequenos programas de computador; identificar erros e possíveis problemas relacionados à sintaxe e semântica e, no curso técnico integrado, compreender os fundamentos e codificar algoritmos na linguagem de programação Java.

Com o intuito de atingir estes objetivos, foram desenvolvidas metodologias pedagógicas que possibilitam, através da percepção e utilização de diferentes estratégias de ensino e aprendizagem, apreender conceitos relacionados ao ensino de lógica e a construção de algoritmos.

4. Considerações Finais

Ao utilizar diferentes metodologias pedagógicas para o ensino de programação, notou-se entusiasmo e interesse por parte das turmas em aprender a programar e desenvolver o senso lógico. Independentemente do nível acadêmico, obtiveram-se resultados promissores e relatos positivos dos alunos em relação à aplicação das diferentes formas utilizadas na abordagem de conceitos, no decorrer das disciplinas. Notou-se, durante a aplicação, correção e revisão de avaliações com os alunos, uma boa aplicação dos conceitos desenvolvidos.

Em contato informal com os docentes responsáveis por ministrar disciplinas precedentes às disciplinas de iniciação a programação, tanto no curso técnico, quanto no superior, foi relatada uma melhora nas capacidades de raciocínio lógico por parte dos alunos. Diante destes relatos e dos resultados obtidos a partir de avaliações e atividades com os estudantes, percebeu-se que os mesmos desenvolveram uma capacidade considerável para trabalhar com conceitos de lógica e programação, mostrando estarem prontos para aprofundar mais conceitos.

A partir do que foi observado e do que foi relatado, percebe-se a importância de:

- Criar metodologias pedagógicas diferenciadas, procurando desenvolver o raciocínio lógico e a capacidade de trabalhar em conjunto nos estudantes;
- Considerar na metodologia a ação cognitiva dos estudantes. Lembrando sempre, que o processo de aprendizagem tem uma conotação individual (o estudante precisa articular o novo conhecimento com os conhecimentos que já tem construídos para atribuir significado);
- Propiciar situações de interação entre os estudantes, entre os estudantes e o conhecimento e as diferentes tecnologias, por meio da mediação pedagógica;
- Articular variadas tecnologias no processo de ensino e de aprendizagem, a fim de proporcionar aos alunos uma pluralidade de estilos de aprendizagem;
- E, nunca esquecer, que cada aluno tem uma forma e um tempo específico para assimilar determinados conhecimentos.

Quando o professor se preocupa com essas condições, na elaboração da metodologia pedagógica, contemplando a participação dos alunos, tanto a partir de seus estilos, como a partir de suas curiosidades e perturbações, acreditamos que esse professor venha a contribuir para o desenvolvimento de processos de aprendizagem contextualizados com a realidade dos estudantes e, desse modo, os estudantes possam atribuir significado aos conhecimentos apreendidos. Assim, os estudantes podem encontrar facilmente, no seu cotidiano, onde utilizar as aprendizagens da escola.

Esta pesquisa indica que mais investigações necessitam ser feitas na área, para que novas formas de construção de conhecimentos, voltadas para disciplinas iniciais de linguagens de programação, possam ser descobertas. E, assim, possibilitar tanto aos professores, quanto aos alunos, formas de ensinar e aprender algoritmos que levem em conta o perfil dos estudantes e,



XII SEMANA CIENTÍFICA UNILASALLE – SEFIC 2016
Canoas, RS – 17 a 21 de outubro de 2016

COMUNICAÇÃO ORAL

ISSN 1983-6783

principalmente, seus estilos de aprendizagem. Desta maneira, proporcionar estratégias de aprendizagem de acordo com a preferência dos alunos.

Referências

APOIO INFORMÁTICA. **VisuAlg**. Disponível em: <<http://www.apoioinformatica.inf.br/produtos/visualg>>. Acesso em: 09 jan. 2016

ANTUNES, C. **As inteligências múltiplas e seus estímulos**. Campinas: Papyrus, 2010.

BARIANI, I. **Estilos Cognitivos de Universitários e Iniciação Científica**. Tese (Doutorado). Unicamp: Faculdade de Educação, 1998.

BULGRAEN, Vanessa C. **O papel do professor e a sua mediação nos processos de elaboração do conhecimento**. Revista Conteúdo, Capivari, v.1, n.4, 2010. Disponível em: <<http://www.conteudo.org.br/index.php/conteudo/article/viewFile/46/39>>. Acesso em: 09 dez. 2016
FELICETTI, Vera Lucia. **Comprometimento do estudante: um elo entre aprendizagem e inclusão social na qualidade da Educação superior**. Tese (Doutorado) – Faculdade de Educação, PUCRS. Porto Alegre: 2011, p.76-118.

FERNANDES, G. B. Programação utilizando Portugol. Disponível em: https://fernandesfmu.files.wordpress.com/2011/04/aula_05_introducao_ao_portugol_impressao.pdf. Acesso em: 29 de fevereiro de 2016.

FONTES, C. R. SILVA, F. **O ensino da disciplina linguagem de programação em escolas técnicas**. Disponível em: <http://www.cienciasecognicao.org/revista/index.php/cec/article/view/221>. Acesso em: 12 de dezembro de 2015.

LOPES, J. A. **O que é algoritmo?** Disponível em: <http://revistaescola.abril.com.br/matematica/pratica-pedagogica/algoritmo-611956.shtml>. Acesso em: 15 de dezembro de 2015.

MARJI, Majed. Aprenda a Programar com Scratch. Novatec Editora Ltda, 2014.

RACHA CUCA. **Missionários e Canibais**. Disponível em: <http://rachacuca.com.br/jogos/missionarios-e-canibais/> Acesso em: 12 de dezembro de 2015.

RACHA CUCA. **O lobo e a ovelha**. Disponível em: <http://rachacuca.com.br/jogos/o-lobo-e-a-ovelha/>. Acesso em: 12 de dezembro de 2015.

SCRATCH. Disponível em: <<https://scratch.mit.edu/>>. Acesso em: 09 jan. 2016

SLACK, N. CHAMBERS, S. HARLAND, C. HARRISON, A. JOHNSTON, R. - **Administração da Produção**. São Paulo: Editora Atlas, 1997.

SOUZA, C. **VisuAlg - Ferramenta de Apoio ao Ensino de Programação**. Disponível em: [http://www.uss.br/pages/revistas/revistateccen/V2N22009/Artigo VisuAlgSOUZA.pdf](http://www.uss.br/pages/revistas/revistateccen/V2N22009/Artigo%20VisuAlgSOUZA.pdf). Acesso em: 15 de dezembro de 2015.

TAVARES, R. Construindo mapas conceituais. Disponível em: <http://www.cienciasecognicao.org/pdf/v12/m347187.pdf>. Acesso em: 01 de março de 2016.

VYGOTSKY, L. S. **A formação Social da mente: o desenvolvimento dos processos psicológicos superiores**. Org. Michael Cole...[et al.]; tradução José Cipolla Neto [et al.], São Paulo: Martins Fontes, 1994.