

## Processo de Ensino e Aprendizagem de Algoritmos Integrando Ambientes Imersivos e o Paradigma de Blocos de Programação Visual

Érico M. H. Amaral<sup>1,2</sup>, Roseclea D. Medina<sup>3</sup>, Liane M. R. Tarouco<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Informática na Educação – Universidade Federal do Rio Grande do Sul (UFRGS)

90.040-060 – Porto Alegre – RS – Brasil

<sup>2</sup>Engenharia de Computação – Universidade Federal do Pampa (Unipampa)

96413-172 – Bagé – RS – Brasil

<sup>3</sup>Programa de Pós-Graduação em Informática – Universidade Federal de Santa Maria (UFSM) - 97105-900 – Santa Maria – RS - Brasil

ericoamaral@unipampa.edu.br; roseclea.medina@gmail.com;  
liane@penta.ufrgs.br

**Abstract.** *Difficulty of students in relation to learning algorithms is a subject that is renewed, either by the market demand in the area of Information Technology or the high dropout rates in introductory programming courses. Understanding these circumstances, this research focuses on proposing a different design for the learning process of algorithms, characterized by a pedagogical practice based on the integration of immersive environments and programming paradigms, differently from those commonly used in classroom. Results achieved with this proposal showed that the interaction and immersion features offered by the virtual world, combined with development tools in visual blocks, motivate and stimulate student's attention, making them as protagonists in the development of their knowledge of algorithms and programming.*

**Resumo.** *A dificuldade dos estudantes em relação à aprendizagem de Algoritmos é um assunto que se renova, seja pela demanda de mercado na área de Tecnologia da Informação ou pelos altos índices de evasão em disciplinas introdutórias de programação. Entendendo essas circunstâncias, esta pesquisa tem como foco propor uma concepção diferenciada para o processo de ensino de Algoritmos, caracterizada por uma prática pedagógica baseada na integração de ambientes imersivos a paradigmas de programação, diferentes dos comumente utilizados em sala de aula. Os resultados alcançados com esta proposta mostraram que as características de interação e imersão, proporcionadas pelo mundo virtual, aliadas a ferramentas de programação em blocos visuais, motivam e estimulam a atenção dos alunos, tornando-os protagonistas no desenvolvimento de seus saberes sobre algoritmos e programação.*

### 1. Introdução

Vive-se atualmente em um mundo de constantes mudanças sociais, culturais e tecnológicas, as quais impelem diferentes desafios à humanidade e, através destes, o indivíduo se desenvolve enquanto ser, em um ambiente adverso. Contudo, esta é uma tarefa que exige a cada dia, uma maior capacidade de adaptação ao meio, denotando

novas demandas para o cidadão do século XXI, o qual deve estar apto a reconhecer e lidar com estas mudanças. O uso das tecnologias é um dos grandes alicerces deste novo cenário, onde as pessoas estão imersas em um ambiente no qual a computação é cada vez mais ubíqua. Utilizar estes recursos tecnológicos para tomar decisões apropriadas no dia a dia exige do cidadão a faculdade de saber manipular diferentes sistemas, aplicando conceitos básicos de lógica de programação.

Ao apresentar esta nova concepção, percebe-se a importância do ensino e aprendizagem da Lógica de Programação e Algoritmos em todos os meios, principalmente em ambientes de ensino superior. É comum encontrar pelo menos uma disciplina relacionada com a computação, no currículo de cursos de nível superior, de áreas que não a da computação. Ao adotar-se esta linha de ação, relacionada ao ensino da Lógica e Programação, é possível perceber a dificuldade inerente à abstração e construção do raciocínio lógico dos estudantes, especialmente no que tange à forma de compreensão de esquemas algorítmicos. Para Santos e Costa (2006) o fato do estudante não conseguir, de forma simples, construir este raciocínio pode ser considerado como um dos principais fatores que corroboram para os altos índices de evasão nas disciplinas de programação e, também dos cursos superiores que possuem estas áreas como elementos curriculares.

A dificuldade na construção do Pensamento Computacional (PC) pode ser considerada uma das dificuldades inerentes à aprendizagem de programação, visto que Wing (2006) descreve o PC como um conjunto de competências e habilidades básicas e específicas, apresentadas pelo indivíduo, relacionadas à área de computação. Estas competências abrangem de forma geral a capacidade da utilização de recursos computacionais e estratégias algorítmicas para a resolução de problemas, por meio da abstração e decomposição lógica dos mesmos. O nível de complexidade, inserido na ação de descrever um algoritmo, é outro fator relevante. Para Wirth (1976) a construção de um programa é realizada a partir de formulações concretas destes algoritmos, com base em representações e estruturas específicas, as quais, para atender a um problema, devem respeitar uma determinada codificação (KNUTH, 1968).

Com base no exposto, este estudo buscou investigar estratégias mais adequadas para estimular os processos de ensino e de aprendizagem dos estudantes, com vistas ao desenvolvimento de algoritmos e programação. Como resultado propôs-se uma abordagem didática alternativa para estimular os processos de ensino e de aprendizagem, através de uma prática diferenciada, utilizando como instrumentos: um ambiente de blocos de programação visual (Scratch), para introdução à lógica e programação; uma linguagem de programação para mundos virtuais (*Linden Scripting Language/OpenSim*), a fim de motivar o estudante na construção de soluções que podem ser observadas em um ambiente realístico em 3D e; uma linguagem de programação tradicional (linguagem C), fazendo com que o estudante consiga codificar estruturas algorítmicas para solução de diferentes problemas.

## **2. Práticas pedagógicas no ensino de algoritmos**

A prática pedagógica possui uma ampla concepção, podendo ser considerada como uma ação social, calcada sobre objetivos, finalidades e conhecimentos, inserida no contexto da prática social, segundo Veiga (1992).

Ao se tratar do ensino na área de tecnologia é importante observar que os sistemas computacionais estão em constante mudança, aperfeiçoamento, o que atinge diretamente o contexto das práticas pedagógicas. Moran (2001) já afirmava que as mudanças são constantes na sociedade, países e instituições. Este contexto pode ser

nitidamente identificado no conteúdo desta pesquisa, em que diferentes práticas pedagógicas são adotadas por professores e experimentadas em estudos com o foco na aprendizagem de algoritmos, dentre as quais destacam-se: o uso de realidade virtual (BARBAS & LOPES, 2013), adoção de dispositivos móveis como ferramenta de apoio (BARCELOS, 2009), prática com o uso de animações, simulações e *web* (BECK *et al.* 2014), intervenções complexas (BROWN, 1992), ambientes interativos – projeto Alice (CONWAY E PAUSCH, 1997), ensino de programação por demonstração (FERREIRA *et al.* 2010), robótica como instrumento de aprendizagem (FINKE *et al.* 1996), utilização de redes sociais com prática em algoritmos (MIRANDA, 2014), entre outras.

### **3. Ensinar e aprender em um ambiente rico em tecnologia**

Sabe-se que ao utilizar recursos tecnológicos para construção de materiais didáticos, diversos elementos devem ser considerados, dentre os quais se destaca a Teoria da Carga Cognitiva (TCC), descrita por Sweller *et al.* (1998). Para o entendimento desta teoria é primordial o reconhecimento sobre o conceito da cognição humana (NUNES & GIRAFFA, 2003), que descreve a capacidade de um indivíduo em processar informações por meio de sua estrutura cognitiva, definida em: memória sensorial, memória de longa duração e memória de curta duração, segundo Tarouco & Santos (2009). Por sua vez a concepção de materiais educacionais deve ser estruturada através da composição sucessiva de objetos, disponibilizados de acordo com o seu nível de dificuldade, que segundo Wiley (2000), é a base para a chamada teoria da complexidade, a qual define que a aprendizagem deve ser organizada de forma hierárquica e crescente em relação a sua complexidade. Todavia, este fator pode ter um efeito negativo no processo de aprendizagem, levando a uma alta demanda no processo de interpretação dos conteúdos e, muitas vezes à distração do estudante, o que é descrito como Sobrecarga Cognitiva (TAROUCO, 2006).

Com o intuito de reduzir esta demanda, Lindermann (1983) e Tarouco (2006) apontam alguns cuidados na implementação de materiais educacionais, a fim de minimizar os impactos das diferentes informações contidas em um objeto de aprendizagem: simplicidade nos textos utilizados, padrões de cores atraentes e não cansativas formatações de textos e imagens de forma agradável, agrupamento das informações por similaridade e o uso de imagens e gráficos, para um modo visual com o foco na interpretação dos dados. Contudo, para que um determinado material didático não gere uma sobrecarga cognitiva, Mayer (2001) propõe em suas pesquisas um conjunto de princípios a serem seguidos: Princípio de Representação Múltipla, o qual descreve que estudantes apresentam um melhor entendimento ao combinar palavras e imagens; Princípio de Proximidade Espacial apontando para que o texto e imagens relacionadas apresentem uma proximidade espacial; Princípio da Não Divisão ou da Proximidade Temporal sugere que o texto e imagens sejam apresentados simultaneamente; Princípio das Diferenças individuais aponta para maior capacidade de determinados indivíduos no entendimento de certos conteúdos; Princípio da Coerência defende que o material didático deve ser simples e objetivo, apenas com informações pertinentes ao tema tratado.

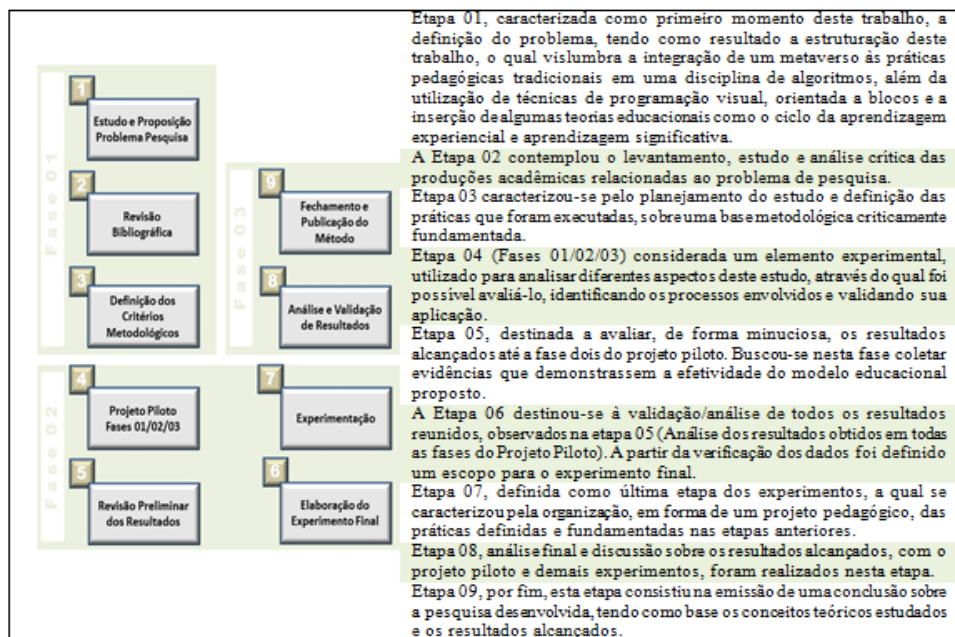
### **4. Materiais e Métodos**

Para o sucesso nesta pesquisa foi importante a definição de um método científico, que propiciasse um conjunto de ações sistemáticas, a fim de conferir segurança aos resultados alcançados (LAKATOS, 2007). De acordo com os métodos descritos na literatura, este estudo baseou-se em uma proposta indutiva, o qual utilizou o registro de

fatores singulares para a definição do conjunto de conclusões possíveis. A observação e identificação dos fenômenos, a partir dos experimentos pontualmente elaborados, concretizaram a caracterização deste estudo sobre uma metodologia indutiva científica (RUIZ, 2010). A pesquisa, com base no objeto de estudo, foi desenvolvida de forma descritiva, objetivando mapear as informações pertinentes ao processo de aprendizagem de algoritmos, considerando aspectos históricos, propondo um novo conceito para abordagem dos conteúdos e analisando os resultados alcançados.

Devido à concepção deste trabalho ser alinhada ao campo da educação, foi conveniente relacionar os procedimentos metodológicos enunciados, às práticas da metodologia de Pesquisa Baseada em *Design* (PBD), pois a mesma, segundo (RAMOS *et al.* 2009), está consolidada na integração da pesquisa ao desenvolvimento de interações educativas em contextos de aprendizagem reais, sendo reconhecida internacionalmente por estes aspectos.

Tendo a avaliação formativa durante o processo de aprendizagem, como alvo primordial, foi possível dividir as práticas desta pesquisa em 09 etapas, claramente definidas e descritas na Figura 01.



**Figura 01. Esquema de planejamento para a pesquisa**

Estas etapas contemplaram um conjunto de ações previamente avaliadas e definidas de acordo com a importância deste estudo, respeitando a metodologia científica adotada e parametrizando todas as ações do pesquisador.

## 5. Implementação

A partir do referencial estudado e da estrutura metodológica proposta foram concebidos e planejados um conjunto de experimentos, a fim de atender os requisitos desta pesquisa. Realizaram-se 4 experimentos, utilizando como grupo de estudos disciplinas de Algoritmos e Programação (60 horas) na Universidade Federal do Pampa, totalizando 240 horas de observações e coletas de dados, as quais contemplaram os 3 projetos piloto e o experimento final deste trabalho.

As 4 turmas elencadas para a participação da pesquisa foram selecionadas de maneira intencional, sendo relacionadas disciplinas com alunos de diferentes cursos superiores, incluindo estudantes de computação. A técnica de casos múltiplos permitiu

ampliar a efetividade dos resultados alcançados, pois possibilitou a replicação dos casos no segundo e terceiro projeto piloto e, também no experimento final, garantindo assim comparações deliberadas e contrastantes, reforçando os achados do estudo (YIN, 2002).

O primeiro caso (Projeto Piloto 01), implementado no semestre 2013/1, teve como objetivo principal identificar a viabilidade da utilização do mundo virtual Open Simulator (OS) como ferramenta de apoio na disciplina de algoritmos. Ao término deste caso foi possível apontar a efetividade na utilização dos recursos do OS como apoio ao processo de ensino e aprendizagem de Algoritmos, o ambiente utilizado é demonstrado na Figura 02.



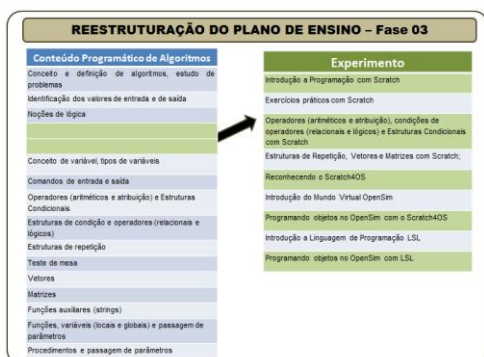
**Figura 02. Sala de aula para o ensino de vetores e matrizes no Open Simulator**



**Figura 03. Infográfico da nova estrutura para o plano de ensino (Modelo 1) – Projeto Piloto Fase 2**

A partir do reconhecimento da viabilidade de utilização dos ambientes imersivos realizou-se o projeto, análise e implementação do segundo caso (Projeto Piloto 02), durante o semestre de 2013/2. O desenvolvimento desta experiência esteve calcado na reorganização do plano de ensino da disciplina de Algoritmos e Programação (Figura 03), com o intuito de apresentar aos alunos, além dos conteúdos de lógica e da linguagem C, outros paradigmas que pudessem auxiliar na tarefa da formação do Pensamento Computacional e da lógica de programação. Desta forma, os conteúdos da disciplina tradicional foram condensados na primeira etapa do semestre e, novos temas como programação com a linguagem Scratch, Scratch4OS e LSL passaram a compor o final deste plano de ensino. A avaliação do caso ocorreu por meio da análise estatística, do desempenho acadêmico dos alunos (notas durante todo semestre) e, observação de instrumentos de pesquisa respondidos pelos mesmos, durante diferentes momentos.

O terceiro caso (Projeto Piloto 03) baseou-se em uma réplica Projeto Piloto 2, durante o semestre de 2014/1, contudo neste experimento o plano de ensino da disciplina teve os conteúdos organizados de maneira inversa ao caso 2 (Figura 04). O projeto de alteração, na ordem dos temas apresentados em sala de aula, foi motivado pelo interesse em saber qual disposição de conteúdos obteria um melhor rendimento dos alunos. Realizou-se a análise dos resultados do caso 3 da mesma forma que no semestre anterior, a replicação teve como intuito garantir a obtenção de repostas fidedignas, validando a comparação das informações coletadas.



**Figura 04. Plano de Ensino Adaptado (Modelo 2) – Projeto Piloto Fase 3**

PLANO DE ENSINO – Experimento Final		
Conteúdo Programático de Algoritmos	Avaliações	Conteúdo
Conceito e definição de algoritmos, estudo de problemas	Avaliação 01 Scratch Scratch4OS LSL	Avaliação 1.1 Identificação dos valores de entrada e de saída; Noções de lógica; Construção de aplicações simples com Scratch.
Identificação dos valores de entrada e de saída		Avaliação 1.2 Conceitos de variáveis; Operadores (aritméticos e atribuição); condições de operadores (relacionais e lógicos) e Estruturas Condicionais com Scratch.
Noções de lógica e Conceitos de variável		Avaliação 1.3 Estruturas de Repetição, Vetores e Matrizes com Scratch; Scratch4OS.
Introdução a Programação com Scratch		Avaliação 1.4 Programando com LSL.
Exercícios práticos com Scratch	Avaliação 02 Linguagem C	Conceito de variável, tipos de variáveis; Comandos de entrada e saída; Operadores (aritméticos e atribuição); Estruturas Condicionais; Estruturas de condição e operadores (relacionais e lógicos).
Operadores (aritméticos e atribuição), condições de operadores (relacionais e lógicos) e Estruturas Condicionais com Scratch		Avaliação 03 Linguagem C Estruturas de repetição; Teste de mesa; Vetores; Matrizes; Funções auxiliares (strings).
Estruturas de Repetição, Vetores e Matrizes com Scratch;		Avaliação 04 Linguagem C Funções, variáveis (locais e globais) e passagem de parâmetros; Procedimentos.
Reconhecendo o Scratch4OS		Avaliação 05 - Recuperação Scratch, Scratch4OS, LSL, Linguagem C Todo conteúdo da disciplina
Introdução do Mundo Virtual OpenSim		
Programando objetos no OpenSim com o Scratch4OS		
Introdução a Linguagem de Programação LSL		
Programando objetos no OpenSim com LSL		
Conceito de variável, tipos de variáveis na linguagem C		
Comandos de entrada e saída		
Operadores (aritméticos e atribuição) e Estruturas Condicionais		
Estruturas de condição e operadores (relacionais e lógicos)		
Estruturas de repetição		
Teste de mesa		
Vetores		
Matrizes		
Funções auxiliares (strings)		
Funções, variáveis (locais e globais) e passagem de parâmetros		
Procedimentos e passagem de parâmetros		

**Figura 05. Plano de Ensino e Atividades Avaliativas do Experimento Final**

O caso 4 (Experimento Final - 2014/2) foi projetado a partir do reconhecimento da efetividade dos casos anteriores. Embora os resultados alcançados com o caso 2 e 3 tenham ficado muito próximos, concluiu-se a partir da avaliação do desempenho acadêmico dos discentes, que a variação positiva nas médias dos alunos do terceiro caso possibilitou a tomada de decisão em relação a qual organização de conteúdo atendeu melhor a demanda de formação do Pensamento Computacional e capacidade de programação dos estudantes, culminando em um plano de ensino a ser adotado no experimento final (Figura 05). Desta forma foi realizado o estudo e planejamento do experimento final, o qual foi constituído de um conjunto de atividades formativas e avaliativas, implementadas durante a totalidade de um semestre letivo.

**Quadro 01. Práticas Formativas realizadas (Casos)**



**Figura 06. Infográfico sobre conteúdos do Modelo 2**

Casos	Título	Descrição
Caso01	Nivelamento	Atividade baseada em um instrumento de pesquisa, composto por questões abertas e objetivas abordando o conhecimento dos alunos sobre programação. Os alunos deveriam implementar códigos em linguagem C.
Caso02	Lógica	Atividade realizada com a finalidade de estimular o raciocínio lógico dos alunos, por meio da interação com a aplicações desenvolvidas em flash.
Caso03	Fluxograma	Os alunos deveriam construir fluxogramas para resolução de problemas com demandas diferenciadas.
Caso04	Scratch	Prática desenvolvida para explorar a capacidade dos alunos em implementar um programa simples, utilizando os recursos da linguagem Scratch.
Caso05	Interpretação	Exercício voltado para a análise e interpretação de dois requisitos em aplicações desenvolvidas com Scratch.
Caso06	Identificar Erros	Tarefa com propósito de estimular o senso crítico do aluno, sua aptidão em analisar programas prontos e de apontar a existência ou não de erros.
Caso07	Programação LSL	Uma prática voltada a interpretação de códigos desenvolvidos com a linguagem LSL.
Caso08	OpenSim	Implementação de aplicações para o Mundo Virtual.
Caso09	Programação com Linguagem C	Este exercício consistiu no desenvolvimento de aplicações com a Linguagem C, a partir de requisitos descritos em fluxogramas.
Caso10	Interpretação de Códigos em C	Caso composto por duas atividades de interpretação de códigos desenvolvidos em linguagem C, com a finalidade de identificar erros de sintaxe, lógica e classificação de níveis de acerto.

A metodologia adotada neste experimento, devidamente avaliada nas fases iniciais da pesquisa, foi definida sob uma estrutura didática organizada com a seguinte sequência: uma concepção inicial sobre lógica e raciocínio, construção de fluxogramas, programação com a linguagem Scratch, desenvolvimento com o Scratch4OS, construção de *scripts* para mundos virtuais com o LSL e, por fim, utilização da linguagem C, com uma programação estruturada e imperativa, a qual recebeu a denominação de Modelo 2, conforme apresentado no infográfico da figura 06.

Um conjunto de práticas formativas, denominadas de Casos, foram propostas em sala de aula, a fim de coletar subsídios e indícios da construção do Pensamento



Computacional e raciocínio lógico dos estudantes ao utilizar o Modelo 2, as quais estão apresentadas no Quadro 01.

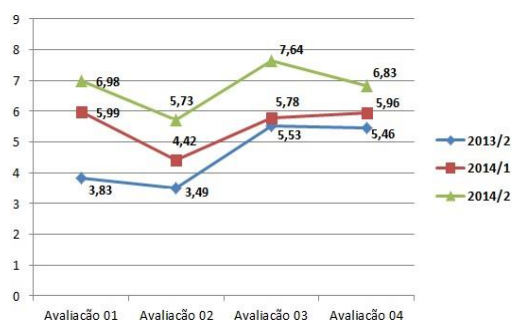
### 5.1. Resultado e Discussões

Entendendo a pertinência de validar a proposta metodológica defendida por esta pesquisa, realizou-se uma análise formal dos dados obtidos durante a última etapa dos experimentos, buscando através de procedimentos estatísticos fundamentar as observações realizadas, visto que as técnicas estatísticas basearam-se na coleta, organização e interpretação de informações sobre modelos específicos, segundo Kachigan (1986). Para a execução destas atividades foi utilizado um conjunto de instrumentos, com o objetivo de interpretar os dados numéricos manipulados e, desta forma obter uma coleção de resultados precisos, esclarecidos e devidamente fundamentados (SILVESTRE, 2007).

O Quadro 02 apresenta o universo de participantes deste estudo, com a composição de cada semestre letivo, apontando os dados de matrícula, evasão, frequência e aprovações. Os quantitativos neste quadro se equivalem em todos os campos, contudo destaca-se a evolução no número de aprovados no último período (2014/2). Salienta-se que neste caso, observou-se pontualmente a amostra de estudantes ativos até o final do semestre (com presença acima de 75% nas aulas), os quais obtiveram um percentual 91% de aprovações em 2014/2, corroborando com as conjecturas supostas por este estudo na utilização do Modelo 2.

**Quadro 02. Universo de alunos participantes de todas as etapas da pesquisa**

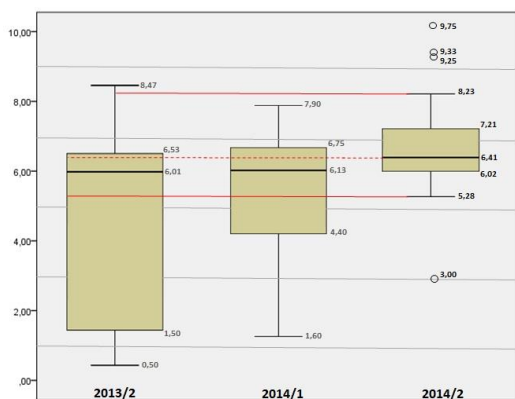
Matrículas x Evasão x Aprovações (2013/2 - 2014/1 - 2014/2)					
Semestre	Matrículas	Nunca compareceram	Evasão Geral	Frequentes	Aprovados
2013/2	60	27	39	21	13
2014/1	60	30	35	25	15
2014/2	60	24	38	22	20



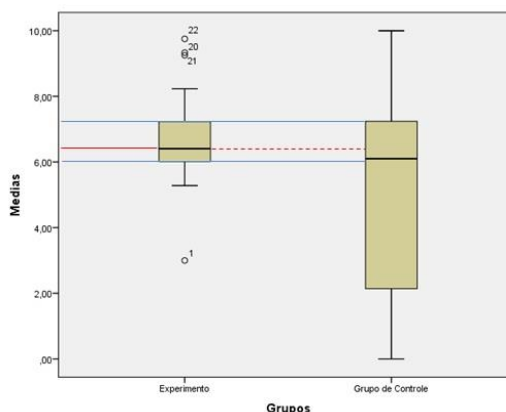
**Gráfico 01 Comparação entre médias das avaliações (2013/2, 2014/1 e 2014/2)**

Outra evidência encontrada da melhoria no desempenho dos estudantes é exposta ao analisar os resultados alcançados por estes, em atividades avaliativas realizadas nos respectivos semestres, conforme demonstrado no Gráfico 01.

A fim de comparar as médias dos alunos, de forma individual, utilizou-se um gráfico de BoxPlot (Gráfico 02). Destaca-se nesta representação dos *outliers* dos dados de 2014/2, referentes às notas de 3 acadêmicos (9,25/9,33/9,75), as quais apontaram o bom nível de habilidade de programação destes. O limite superior no BoxPlot de 2013/1 encontra-se acima dos demais semestres, contudo a mediana para este semestre é a menor observada, sobressaindo novamente 2014/2, com um valor de média 7,21 para o terceiro quartil e, 6,41 como mediana, sendo estes dados superiores aos demais períodos. Como informação relevante nesta observação, destaca-se o limite inferior denotado no segundo semestre de 2014, apontando um valor de 5,28, caracterizando que a predisposição para a melhora no desempenho ocorreu de forma linear e equânime, tanto pelos aprovados, quanto pelos discentes com notas reduzidas (abaixo da média).



**Gráfico 02. BoxPlot comparativo entre médias (2013/2, 2014/1 e 2014/2)**



**Gráfico 03. BoxPlot comparativo entre Grupo Experimento e Grupo de Controle**

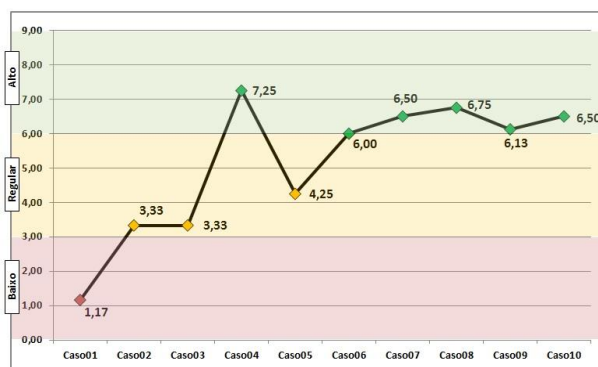
Com o intuito de inferir indutivamente as propriedades do universo observado, vislumbrou-se a obtenção de uma decisão estatística que apoiasse a afirmação da superioridade no desempenho acadêmico, constatado neste trabalho, em relação às notas dos alunos participantes das ações didáticas do Modelo 2 (denominados de Grupo Experimento com 22 alunos) sobre os demais discentes, os quais cursaram outras disciplinas de algoritmos, nesta mesma universidade, nos anos de 2013 e 2014, que compuseram o chamado Grupo de Controle (284 estudantes).

As constatações iniciais, com os testes de normalidade, determinaram que para a comparação das médias do Grupo de Controle e Experimento era possível utilizar tanto testes paramétricos (distribuições normais) quanto testes não paramétricos (distribuições diferentes de normais). A partir do reconhecimento formalizado dos padrões de distribuição das notas, referentes aos grupos, foram elencados três métodos para a comparação entre as médias dos alunos: um teste de variância (ANOVA), um teste paramétrico (Teste t) e um teste não paramétrico (Mann-Whitney). Obtendo-se como resposta, a diferença entre as médias, nos dois primeiros testes (ANOVA e Teste t). Contudo o teste de Mann-Whitney apontou uma superioridade da média do Grupo de Experimento sobre o Grupo de Controle (Gráfico 03).

Por fim, com o intuito de avaliar a capacidade de construção do Pensamento Computacional, dos alunos participantes no Experimento, propôs-se uma taxonomia baseada em dimensões de competências e habilidades, sendo estas mensuradas em 3 níveis diferentes (Quadro 03). Esta taxonomia foi aplicada sobre as 10 atividades formativas (Quadro 01) realizadas durante o semestre, obtendo-se como resultado a representação inferida no Gráfico 04.

**Quadro 03. Níveis de classificação da TAPC**

Intervalo Valores	Nível PC
1,00 – 2,99	Baixo
3,00 – 5,99	Regular
6,00 – 9,00	Alto



**Gráfico 04. Análise sobre a construção do Pensamento Computacional**



A utilização desta taxonomia permitiu validar o Experimento Final desta pesquisa, no que consente a construção do Pensamento Computacional, visto que ao observar os resultados alcançados com a análise dos Casos desenvolvidos pelos alunos, denota-se uma clara ascensão no que tange as capacidades e habilidades demonstradas durante a execução das tarefas propostas. Desta forma, é factível afirmar, que o método de aprendizagem Modelo 2, pode ser utilizado como uma ferramenta efetiva para o estímulo à construção do Pensamento Computacional por alunos em uma disciplina de algoritmos e programação.

## 6. Conclusões

A reconhecida dificuldade dos estudantes em construir o conhecimento sobre a lógica e programação de computadores é tema de diferentes estudos. O reconhecimento deste contexto e o levantamento teórico realizado nesta pesquisa contribuíram para a definição do modelo adotado, o qual envolveu a utilização de blocos de programação visual, devido à facilidade que ensejam para criar um programa uma vez que possibilitam trabalhar em um processo de selecionar, arrastar/soltar os blocos, compondo assim a solução final. Também envolveu a adoção de mundos virtuais, com os quais foi possível explorar aspectos motivacionais, de imersão e cognitivos comuns neste tipo de aplicação. Culminando com a capacitação para o desenvolvimento da programação com uma linguagem imperativa, neste caso específico a linguagem C, o que não poderia deixar de ser feito uma vez que a aprendizagem desta linguagem é um dos principais objetivos da disciplina de Algoritmos e Programação.

A efetividade do Modelo 2 pode ser comprovada pelas diferentes ações executadas, desde a inferência estatística das médias dos estudantes até a análise das capacidades de construção do Pensamento Computacional, permitiram concluir que a prática pedagógica proposta estimulou a construção do raciocínio lógico e das habilidades e competências necessárias para que o aluno iniciante consiga implementar soluções computacionais para problemas reais.

## Referencias

- Barbas, Maria; Lopes, Nuno. Introdução à programação:(re) construção de espaços educacionais em realidade aumentada. REVISTA DA UIIPS, p. 40, 2013.
- Barcelos, Ricardo; Tarouco, Liane; Bercht, Magda. O uso de mobile learning no ensino de algoritmos. RENOTE, v. 7, n. 3, p. 327-337, 2009.
- Beck Brondani, Matheus; Mozzaquatro, Patricia Mariotto; Antoniazzi, Rodrigo Luiz. Ambiente de simulação e animação para o ensino de programação. Revista Interdisciplinar de Ensino, Pesquisa e Extensão, v. 1, n. 1, 2014.
- Brown, A. Design experiments: theoretical and methodological challenges in creating complex interventions in classroom settings. The Journal of the Learning Science, v.2, n.2, p.141-178, 1992.
- Conway, M., J.; Pausch, R. "Alice: Easy to Learn Interactive 3D Graphics", Computer & Graphics, Volume 31, Issue 3, pages 58- 59, 1997.
- Ferreira, Cláudia; Gonzaga, Flávio; Santos, Rodrigo. Um Estudo sobre a Aprendizagem de Lógica de Programação Utilizando Programação por Demonstração. In: Anais do XVIII Workshop sobre Educação em Computação, XXX CSBC, Belo Horizonte, MG, Brasil. 2010.
- Finke, M.; Hommel, G.; Scheffer, T.; Wysotzki, F. Aerial robotics in computer science education. Computer Science Education, 7(2), 239-246, 1996.

- Kachigan, Sam Kash. *Statistical analysis: An interdisciplinary introduction to univariate & multivariate methods*. New York, NY, 1986.
- Knuth, D. E. *The Art of Computer Programming. Vol 1 – Fundamental Algorithms*. Addison-Wesley, 1968.
- Lakatos, Eva M.; Marconi, Marina de A. *Metodologia do Trabalho científico*. 7. ed. São Paulo: Atlas, 2007.
- Lindermann, Helena. Os sistemas do futuro e seus aspectos ergonômicos. In: *Boletim Dc. Porto Alegre* vol. 4, n. 1, 1983.
- Mayer, Richard. *Multimedia Learning*. Cambridge: Cambridge University Press. 2001.
- Miranda, Fabio Neves. "Facebook e Algoritmos: Um Estudo de Caso do Uso do Facebook como Suporte Pedagógico na Disciplina de Algoritmos e Técnicas de Programação do Curso de Sistemas de Informação em IES. Disponível em: <[http://www.faminasbh.edu.br/upload/downloads/20130627151026\\_83686.pdf](http://www.faminasbh.edu.br/upload/downloads/20130627151026_83686.pdf)>. Acesso em: 20 junho 2014.
- Moran, José M. Novos desafios na educação: a Internet na educação presencial e virtual. In: Porto, Tania M. E. (Org.). *Saberes e linguagens de educação e comunicação*. Pelotas: Ed. Universitária, 2001.
- Nunes, Marcelo; Giraffa, Lúcia. *A educação na ecologia digital*. PPGCC/FACIN, PUCRS, 2003.
- Ramos, Paula; Giannella, T. Rabetti, Struchiner, Miriam. *A Pesquisa Baseada em Design em Artigos Científicos sobre o Uso de Ambientes de Aprendizagem Mediados pelas Tecnologias da Informação e da Comunicação no Ensino de Ciências: Uma Análise Preliminar*. VII Encontro Nacional de Pesquisa em Educação em Ciências – Enpec, ISSN 21766940, Florianópolis, 2009.
- Ruiz, Alvaro J. *Metodologia Científica, Um guia para eficiência nos estudos*, Editora Atlas, 6ª Ed. 2010.
- Santos, Rodrigo Pereira; Costa, Heitor Augustus Xavier. *Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática*. Infocomp, *Journal of Computer Science*, v. 5, n. 1, 2006.
- Silvestre, Antônio. *Análise de dados e estatística descritiva*. Escolar editora, 2007.
- Sweller, John; Merrienboer, Jeroen; Paas, Fred. *Cognitive Load Theory and Instructional Design*. *Educational Psychology Review*, Vol. 10, No. 3, 1998
- Tarouco, Liane. *Alfabetização visual para a redução da sobrecarga cognitiva em material educativo digital*. *Ambientes hipermediáticos*, p. 37-48, 2006.
- Tarouco, L. M. R.; Dos Santos, P. M. E.; Ávila, B.; Grando, A. R.; Souza Abreu, C. *Multimídia Interativa: princípios e ferramentas*. *RENOTE*, v. 7, n. 1, 2009.
- Veiga, Ilma Passos Alencastro. *A prática pedagógica do professor de Didática*. 2. Ed. Campinas, Papirus, 1992.
- Wiley, David. *Learning Object Design and Sequencing Theory*. Dissertation. Brigham Young University, 2000.
- Wing, J. M. *Computational thinking*. *Communications of the ACM*, v. 49, n. 3, p. 33–35, 2006.
- Wirth, N. *Algorithms + Data Structures = Programs*. Prentice Hall, 1976.
- Yin, R. K. *Estudo de caso. Planejamento e métodos*. Porto Alegre, Ed. Artmed, 2002.