

# AMBIENTE COMPUTACIONAL PARA CRIANÇAS DAS SÉRIES INICIAIS APRENDEREM PROGRAMAÇÃO – NEWPROG+

Cleziel Franzoni da Costa<sup>1,2</sup>, Carpegieri Torezani<sup>1</sup>, Orivaldo de Lira Tavares<sup>1</sup>

<sup>1</sup>Programa de Pós-graduação em Informática  
Universidade Federal do Espírito Santo (UFES) – Vitória – ES – Brasil

<sup>2</sup>Instituto Federal de Educação, Ciência e Tecnologia Baiano (IFBAIANO) – Senhor do  
Bonfim – BA – Brasil

cleziel.costa@ifbaiano.edu.br, carpegieri@gmail.com,  
tavares@inf.ufes.br

**Abstract.** *This paper presents the Newprog+, a digital environment for elementary school children to learn computer programming. Learning computer programming exercises a variety of abilities in children, such as logical thinking and problem solving skills. Newprog+ has an activity editor for use by educators and a visual programming language and program evaluator for use by apprentices.*

**Resumo.** *Este artigo apresenta o Newprog+, um ambiente digital para crianças das séries iniciais aprenderem programação de computadores. Aprender programação de computadores exercita diversas habilidades nas crianças, como o raciocínio lógico e a capacidade de resolver problemas. O Newprog+ dispõe de um editor de atividades para ser usado pelos educadores e de uma linguagem visual de programação e um avaliador de programas para serem usados pelos aprendizes.*

## 1. Introdução

Este trabalho apresenta uma extensão do ambiente de aprendizagem de programação NewProg, o NewProg+. Esses ambientes dispõem de uma Linguagem Visual de Programação (LVP) para as crianças aprenderem programação de computadores. Essa LVP permite que os alunos construam algoritmos para a resolução de problemas. Além da LVP, o NewProg+ dispõe de um editor de atividades para os professores criarem as atividades de programação de computadores a serem desenvolvidas pelos alunos.

O NewProg é um software desenvolvido para auxiliar as crianças entre cinco e oito anos de idade na aprendizagem dos conceitos de programação de computadores. Segundo Fessakis, Gouli e Mavroudi (2013), aprender programação de computadores ajuda o aluno a desenvolver conhecimentos e habilidades em diversas áreas. O Newprog proporciona uma sensação de entretenimento, enquanto favorece o desenvolvimento de habilidades de programação que exercitam destrezas como o planejamento e a resolução de problemas, importantes em diversas situações do cotidiano do aprendiz.

O Newprog+ enriquece a LVP do Newprog, permitindo a construção de programas que, além de sequências de instruções, já disponíveis no NewProg, possam incluir testes condicionais e estruturas de repetição.

Fessakis, Gouli e Mavroudi (2013) elaboraram um estudo de caso, onde são levantados requisitos computacionais que podem ser utilizados para compor um ambiente onde crianças com faixa etária entre 5 e 8 anos de idade aprendam a programar. Sua pesquisa utilizou dois ambientes, o *Ladybug Maze* e o *Ladybug Leaf*. Estes ambientes possuem uma linguagem visual que se assemelham a LVP apresentada pelo NewProg+. Além do trabalho citado, existem outros projetos com finalidades similares as do Newprog+, como o CodeCombat, *Scratch*, CondeHunt, dentre outros. Cada um deles possui suas particularidades, no entanto, a proposta da LVP deste trabalho, além do ambiente para o professor editar atividades e para o aprendiz editar e testar a solução de cada atividade, ao mesmo tempo em que o percurso de construção das soluções fica registrado, para posterior análise do professor, justifica o desenvolvimento do ambiente aqui apresentado.

## 2. Revisão de Literatura

De acordo com Oliveira *et al.* (2014), os conhecimentos referentes às ciências da computação estão restritos aos alunos dos cursos superiores e técnicos das áreas afins. O autor ainda ressalta que o ideal seria que cada cidadão compreendesse pelo menos os princípios básicos dessa área. Em consequência disso, estudos apresentados pelo mesmo autor mostram uma previsão de uma grande demanda por profissionais das áreas de tecnologia até o ano de 2018, devido à escassez de profissionais qualificados.

Uma pesquisa realizada pela Folha de São Paulo aponta uma evasão de 28% dos alunos dos cursos de Matemática e de Ciências da Computação. Dentre os vários motivos que podem provocar essa evasão, segundo a mesma pesquisa, estão as dificuldades dos alunos com o desenvolvimento da lógica de programação e das diversas sintaxes presentes nas diferentes linguagens de programação (RIBEIRO, BRANDÃO e BRANDÃO, 2012).

Considerando isso, a aprendizagem de programação de computadores deveria ser promovida, em ambientes de programação especialmente construídos, desde as séries iniciais, visto que, segundo Castro, Castro e Menezes (2002), o processo de aprendizagem de programação de computadores envolve o desenvolvimento de habilidades como a capacidade de raciocínio lógico e de abstração.

Para Oliveira *et al.* (2014), a maioria dos ingressantes dos cursos de diversas áreas encontram dificuldades para aprenderem certos conteúdos referentes à Ciência da Computação, destacando-se as disciplinas que envolvem algoritmos, lógicas de programação e cálculo. Essas disciplinas propõem uma “nova forma de pensar” e requerem diversas habilidades que provavelmente não foram desenvolvidas por esse aluno no ensino regular, o que dificulta o processo e traz resultados diferentes do esperado para o aprendiz.

Tavares, Menezes e Nevado (2012) citam que o aluno reinicia suas tentativas de construir um programa de computador após tentativas anteriores infrutíferas, buscando uma solução correta para resolver o problema. O intuito do aluno é descobrir e corrigir

seus erros em um processo de tentativa e erro, até desenvolver uma solução correta. Esses autores ainda relatam que o processo de correção dos erros é fundamental para a aprendizagem dos alunos. Nesse processo é bem-vinda a existência de mecanismos de autorregulação, para apontar formas de aperfeiçoar a solução construída. Além disso, afirmam que a construção de uma solução correta não significa que o aprendiz já dominou o processo de construção de soluções para um dado problema. A aprendizagem sobre esse processo somente se consolida quando o aprendiz desenvolve outras soluções diferentes para o mesmo problema e reflete sobre o processo de construção. Quando o aprendiz tem oportunidade de comparar suas soluções com as dos seus colegas, a consolidação da aprendizagem pode ser acelerada.

Para Wangenheim, Nunes e Santos (2014), a programação de computadores é fundamental para a computação, tornando-se necessário o desenvolvimento da competência para criar programas. Os autores ainda afirmam que com o aprendizado de programação se estimula a aprendizagem do pensamento computacional, o que envolve um conjunto de conceitos computacionais, como a abstração, recursão e iteração, entre outros.

### **3. Metodologia**

Para o desenvolvimento do ambiente computacional proposto, este projeto de pesquisa foi elaborado de acordo com os seguintes passos: O primeiro consiste em conhecer o que realmente é importante para ser incorporado ao ambiente, de forma que seja alcançado o objetivo final de propor um ambiente para aprendizagem de programação para crianças, projetado com uma linguagem visual e com ícones e símbolos que são facilmente compreendidos pelas crianças das séries iniciais. Para isto, foi realizada uma revisão bibliográfica, com a finalidade de identificar os requisitos essenciais ao sistema proposto e aprimorar a sua versão anterior, com o objetivo de promover a aprendizagem de programação de computadores de forma descontraída e atrativa. O segundo passo foi construir a ferramenta proposta, que se trata de um sistema desenvolvido para a web.

Antes de iniciar a terceira etapa, foi preciso estabelecer parceria com pelo menos uma escola com o perfil adequado de alunos, com idades entre 5 a 8 anos, para iniciar os testes de implementação e validação do software. Com a escola selecionada, a terceira etapa foi implantar e validar o sistema, gerando relatórios para validar sua usabilidade e seus resultados. Essa validação foi realizada com os alunos da escola e com seus professores, que são os criadores das atividades no ambiente.

### **4. O Newprog+**

O NewProg+ é uma extensão e atualização do NewProg, um software que foi desenvolvido com o objetivo de auxiliar as crianças, na faixa etária de cinco até oito anos de idade, na aprendizagem de programação de computadores. O Newprog, segundo Torezani, Chagas e Tavares (2013), proporciona uma sensação de entretenimento, enquanto desafia a criança em atividades que envolvem planejamento e resolução de problemas. A Figura 1 ilustra uma atividade a ser desenvolvida por um aluno no ambiente do Newprog.

De forma geral, o Newprog+ se diferencia do Newprog principalmente pela inserção das estruturas de repetição e de testes condicionais na Linguagem Visual de Programação, além de um novo sistema de gerenciamento e edição de atividades.

Assim como o Newprog, o Newprog+ possibilita, ao aluno, diversos testes e desafios com problemas a serem resolvidos por eles. Nessas interfaces, além do cenário representado, existe um avatar que simula a execução das instruções do programa elaborado pelo aprendiz. Esse avatar é escolhido pelo aprendiz.



Figura 1. Atividade desenvolvida no Newprog

A Figura 2 ilustra a interface de interação com o aluno do Newprog+. A interface do Newprog+, na versão apresentada para os alunos, possui as seguintes características:



Figura 2. Interface de interação com o aluno no Newprog+

- Andar 1 vez: faz o avatar avançar uma posição (ou um quadro), no sentido da seta indicadora (para cima, para baixo, esquerda ou direita).

- Andar 2 vezes: similar ao “Andar 1 vez”, porém nesta opção o avatar avança duas posições (ou quadros) no sentido das setas.

- Girar: gira o avatar 90° à sua esquerda ou 90° à sua direita. Esse comando é essencial para auxiliar nas estruturas de decisão. Serve para o avatar se direcionar para o objeto ao qual deverá tomar alguma decisão. Quando a posição frontal do avatar não coincidir com o objeto a ser analisado, esse avatar deve ser direcionado ao objeto. No exemplo da Figura 2, o avatar deverá abrir uma porta que está à esquerda de sua localização frontal, porém poderia haver uma porta, por exemplo, tanto à esquerda, quanto à direita do avatar, ao mesmo tempo e assim, com o comando para gira-lo, ele saberá exatamente qual a porta deverá ser aberta.

Ainda no exemplo das portas, o avatar não conseguirá abrir uma porta que não esteja em frente a ele.

- Repetir: ao selecionar o símbolo de repetição (R1 na Figura 2) e inseri-lo na Sequência de Ações, todas as instruções encontradas dentro do quadro “Repetir: (R1)” são realizadas durante a execução da sequência de ações. Para repetir essas mesmas instruções, basta editar o campo que representa a quantidade de repetições, dentro do menu Repetir, assim como podemos observar na codificação apresentada na Figura 2 (Repetir 3 vez(es)). A estrutura de repetição aqui apresentada é a estrutura “for”, das linguagens de programação tradicionais. É possível incluir outros blocos de Repetições (R2, R3, etc), de acordo com a necessidade da atividade proposta pelo professor.

- Itens: representa um inventário contendo todos os itens que o avatar possui. O aluno utiliza esses itens para cumprir seus objetivos finais, como abrir uma porta, se ela estiver fechada, utilizar um “*jump*” para saltar sobre uma armadilha, entre outras possibilidades. Se o programador (aprendiz) guardar mais itens do que ele vai precisar na execução (avaliação) do programa, não haverá erro no código, mas sim um excesso de itens em seu inventário. Este excesso de itens representa mais código que o necessário para a realização de uma atividade. Assim, ao término da atividade, o professor responsável pode apresentar ao aluno uma sugestão de otimização do código desenvolvido. A captura dos itens pode ser comparada aos comandos de atribuição, das linguagens de programação tradicionais.

Para utilizar o item desejado, basta acioná-lo no momento certo, de acordo com a sequência de passos construída, como utilizar o item chave no momento que desejar abrir uma porta.

## 5. Arquitetura do Newprog+

O ambiente do Newprog+ é composto por três módulos: Módulo Administrador, Módulo Pedagogo e Módulo Aprendiz. O Módulo Administrador é responsável pela manutenção do sistema em geral, como o gerenciamento de usuários e de perfis, o Módulo Pedagogo é responsável pelo gerenciamento e administração das atividades propostas, assim como os tabuleiros e cenários que serão construídos para as atividades por eles propostas. O Módulo Pedagogo é o espaço onde os professores podem criar, editar e acompanhar as atividades a serem realizadas pelos alunos.

O Newprog+ mantém boa parte da arquitetura existente no Newprog, onde o Módulo Pedagogo recebe e armazena os dados fornecidos pelo Módulo Aprendiz. Além disso, gerencia as funções pedagógicas em geral, incluindo o gerenciamento das turmas e das atividades desenvolvidas pelos alunos.

Se comparado à arquitetura cliente-servidor, O módulo servidor foi subdividido nos módulos Administrador e Pedagogo. O Administrador é responsável pelas funções administrativas, tais como cadastros de alunos e professores. O Pedagogo é responsável pelas funções pedagógicas em geral, como a edição das atividades, e principalmente, pelo acompanhamento do estudante no processo de construção de soluções para cada atividade.

O Módulo Aprendiz foi projetado para o aprendiz desenvolver soluções (programas) para cada atividade. Neste módulo o aluno tem acesso ao seu espaço, onde é possível acessar as atividades e visualizar um histórico sobre seu progresso em cada uma delas.

Todos os módulos foram desenvolvidos para a web, com a linguagem ASP.Net e a linguagem de programação C#. Foi utilizado ainda a linguagem de programação JavaScript e recursos como JQuery, JSON e AJAX. O desenvolvimento do ambiente NewProg+ utilizou ainda a linguagem de marcação HTML5, configurações de folhas de estilo com CSS3 e o sistema de gerenciamento de banco de dados MySQL.

## 6. Experimentos e Resultados

Como a principal diferença do Newprog+ em relação ao Newprog é a inclusão das estruturas de repetição e dos testes de condição, os experimentos foram realizados com intuito de testar, principalmente, essas duas funcionalidades da ferramenta.

O NewProg+ foi testado durante o primeiro semestre de 2017, com trinta e três alunos, da faixa etária de sete e oito anos de idade, e dois professores, em uma escola de ensino fundamental. Com o Editor de Atividades, os professores criaram uma atividade com a proposta de analisar a capacidade de seus alunos para a resolução de problemas que demandavam soluções com repetição de blocos de comandos.

Vale ressaltar que a participação das crianças nas avaliações realizadas nesta pesquisa foi aprovada pelos seus respectivos responsáveis e os nomes e as imagens destas crianças foram preservados.

O desafio de cada estudante, em uma das atividades usada nos experimentos, era percorrer todo o labirinto, abrindo as portas existentes e ainda capturando todo o tesouro disponível sem cair nas armadilhas (Figura 2). Quando o estudante desenvolvia uma solução, ele era novamente desafiado a tentar construir uma nova solução “melhor” (um programa com menos comandos).

Para esta atividade, cada aluno precisa cumprir os seguintes desafios:

- Capturar todos os tesouros (condição que indica o fim do desafio, concretizando o sucesso da atividade em questão);
- Abrir todas as portas existentes:
  - Para cada porta é necessário um item chave;
- Não colidir com nenhuma armadilha (condição que define o término do desafio, acarretando insucesso);
- Executar o mínimo de passos possíveis, reaproveitando soluções desenvolvidas.

Na área administrativa, o professor cria e edita a atividade (Figura 3). Com o Editor de Atividades o professor edita sua atividade exatamente como desejar. No painel de edição existe o menu “Testes”, onde o professor adiciona os testes que devem ser realizados, com as respectivas condições de aceitação. Na atividade da Figura 3, o teste configurado prevê verificar se a porta está fechada. O professor pode configurar um item que determine a condição do teste como verdadeira, quando estiver em posse do avatar. Assim, no exemplo citado, se a porta está fechada, mas o avatar possui uma chave da porta, então a condição do teste torna-se satisfeita (verdadeira).



Figura 3. Painel para edição de atividade

A Figura 3 apresenta o painel usado pelo professor para edição de atividade. A Figura 2 apresentou uma tela com uma atividade proposta ao aluno para a edição de uma solução (programa). O aprendiz define uma sequência de ações a serem realizadas pelo avatar para solucionar a atividade proposta. Pode-se observar que nessa atividade existe o menu “Testes” (Figura 5). Nesse menu, o aluno deve adicionar os testes necessários para a realização da atividade.

No exemplo da Figura 4, o aluno precisa adicionar um teste para verificar se uma porta está fechada, definindo que “se encontrar uma porta fechada, precisa utilizar uma chave para abri-la” (configurado no menu “Testes”, figura 5). Quando o aluno define que precisa utilizar uma chave para abrir a porta, automaticamente é inserido um item “chave” = 0 em seu inventário. Assim, somente é possível abrir uma porta se o avatar possuir o item chave que tenha sido coletado em seu percurso pelo cenário.



Figura 4. Verificando se a porta está aberta

Na Figura 5 é possível observar o menu “Repetir”, onde existe a possibilidade de incluir uma quantidade configurável de comandos (no caso, máximo de 12 comandos). Esse menu é representado pelo rótulo **RI**. Quando esse símbolo é inserido na “Sequência de Ações”, todas as instruções existentes no menu “Repetir” são executadas



a quantidade de vezes que foram programadas (no exemplo da figura, os comandos serão repetidos 3 vezes).



**Figura 5. Menus de repetição e testes**

Na Figura 6, temos a sequência de ações, que equivale ao programa principal. Nesta solução, foi necessário incluir apenas dois comandos, visto que o segundo chama a estrutura de repetição R1.



**Figura 6. Menus de comandos a serem executados**

No experimento do ambiente, antes da realização da atividade, os professores explicaram e demonstraram aos alunos as funcionalidades do ambiente, assim como os objetivos da atividade. Cada atividade é criada e gerenciada pelo professor responsável. Durante o experimento, foi permitido a cada aluno realizar até cinco tentativas de programação de uma solução.

Após aplicar a atividade, obteve-se os resultados representados na Tabela 1:

- 1ª amostragem: 100% dos alunos envolvidos realizaram a atividade, porém apenas 67% deles conseguiram concluir a atividade com sucesso nesta primeira tentativa;
- 2ª amostragem (Aqui não foram incluídos os alunos que finalizaram a atividade na primeira amostragem, procedimento que ocorre também nas demais amostragens): Após uma segunda explicação sobre o objetivo da atividade, 82% obtiveram sucesso ao completar a atividade;
- 3ª amostragem: podemos constatar que os alunos que chegaram a esta amostragem ainda não estavam entendendo o objetivo da atividade. Após o esclarecimento das dúvidas, 50% deles conseguiram concluir a atividade;
- 4ª e 5ª amostragem: 3% destes alunos chegaram a esta amostragem, mostrando-se desmotivados e ainda sem entender o que era para ser realizado.



**Tabela 1. Resultados após a realização da atividade**

	<b>1<sup>a</sup></b>	<b>2<sup>a</sup></b>	<b>3<sup>a</sup></b>	<b>4<sup>a</sup></b>	<b>5<sup>a</sup></b>
	<b>Amostragem</b>	<b>Amostragem</b>	<b>Amostragem</b>	<b>Amostragem</b>	<b>Amostragem</b>
Alunos	100%	33%	6%	3%	3%
Sucesso	67%	82%	50%	0%	0%
Erros	33%	18%	50%	100%	100%

Os alunos que não obtiveram êxito nas duas primeiras amostragens avaliaram o ambiente do Newprog+ como fácil de ser utilizado, porém alegaram que faltou um pouco de compreensão sobre o real objetivo da atividade. Logo que entenderam o objetivo, em sua maioria, facilmente concluíram a atividade proposta.

De uma forma geral, os alunos e professores envolvidos acharam o ambiente divertido e fácil de ser utilizado e ainda afirmaram que voltariam a utilizar o Newprog+.

## **7. Avaliação de Aprendizagem**

Da mesma forma que o Newprog, o Newprog+ busca auxiliar as crianças no processo de aprendizagem de programação de computadores. Assim como no estudo elaborado por Torezani, Chagas e Tavares (2013), também observamos indicadores que comprovam o desenvolvimento da lógica de programação – primeiro passo para a aprendizagem de programação de computadores. Esses indicadores são: 1. correção de erros – após a identificação de problemas; 2. busca por soluções alternativas para a resolução de um mesmo problema; 3. autoria na criação das soluções e 4. reflexões sobre as soluções propostas. Pôde-se observar também uma atenção especial para solucionar as repetições propostas, de forma a reutilizar um bloco de instruções para resolver problemas diferentes e ainda a associação de objetos às soluções de outros problemas, como nos casos onde os passos deveriam levar o avatar a capturar alguns itens e abrir as portas.

Com a análise das amostragens, vimos um aumento significativo dos alunos que conseguiram construir uma solução para o problema em questão. Em alguns casos, durante uma mesma amostragem, esses alunos reconstruíam a solução anterior, pois julgavam ter encontrado uma melhor solução.

Com esta análise, podemos afirmar que o Newprog+ é capaz de auxiliar as crianças no desenvolvimento de habilidades de programação e de resolução de problemas. O Newprog+ possibilita, ainda, o planejamento de atividades que envolvam o reaproveitamento de soluções para mais de um problema.

## **8. Considerações Finais**

Este trabalho apresentou o Newprog+, atualização e continuação do software Newprog, proposto por Carpegieri Torezani, um software educacional para auxiliar as crianças, na faixa etária entre cinco e oito anos de idade, durante o processo de aprendizagem de programação de computadores.

A análise dos resultados obtidos mostra que o Newprog+ permite aos envolvidos o desenvolvimento de habilidades de programação e de resolução de problemas. O software também possibilita a criação de soluções utilizando estruturas de repetição e testes condicionais.

O Newprog+ permite ainda um acompanhamento pelo professor responsável, que, além de criar e propor as atividades desenvolvidas pelos alunos, é capaz de observar o histórico da sessão de cada estudante para analisar o percurso de aprendizagem do aluno.

Continuaremos usando e avaliando o ambiente NewProg+ em escolas de ensino básico, de modo a fazermos um estudo mais detalhado sobre os benefícios do desenvolvimento das habilidades de programação e os impactos delas no cotidiano escolar dessas crianças.

## 9. Referências

- CASTRO, T.H.C; CASTRO JÚNIOR, A.N.; MENEZES, C.S. Representando Padrões em um Ambiente de Apoio à Aprendizagem de Programação. *In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 2002. p. 558-561.
- OLIVEIRA, M.L. *et al.* Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. *In: XXXIV Congresso da SBC-XXII Workshop de Ensino de Computação, Brasília*. 2014.
- RIBEIRO, R.D.S.; BRANDÃO, L.; BRANDÃO, A. 2012. Uma visão do cenário Nacional do Ensino de Algoritmos e Programação: uma proposta baseada no Paradigma de Programação Visual. *In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 2012.
- TAVARES, O. L.; MENEZES, C.S.; NEVADO, R.A.: *Pedagogical architectures to support the process of teaching and learning of computer programming: In FIE2012-Frontiers in education conference, 2012.*
- TOREZANI, C.; CHAGAS, L. B. C.; TAVARES, O. D. L. NewProg - um ambiente para crianças aprenderem programação de computadores. *Anais do Congresso Brasileiro de Informática na Educação (Wie/Cbie), Campinas - SP, 23 Nov. 2013.* 140-149.
- WANGENHEIM, C. G. V., NUNES, V. R., SANTOS, G. D., 2014. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. *Revista Brasileira de Informática na Educação*, 22(03), p.115.
- FESSAKIS, G.; GOULI, E.; MAVROUDI, E. *Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. Computers & Education*, v. 63, p. 87-97, 2013.