

# O Pensamento Computacional no Ensino Médio: Uma Abordagem *Blended Learning*

Tancicleide C. S. Gomes<sup>1</sup>, Jeane C. B. de Melo<sup>1</sup>

<sup>1</sup>Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco (UFRPE), Recife – PE– Brasil

tancigomes@bsi.ufrpe.net, jeane.ufrpe@gmail.com

**Abstract.** *This paper presents a methodology based on the meaningful learning premises, which outlines a strategy for teaching computational thinking in a contextualized mode, incorporating blended learning techniques to promote more flexibility, as well as the presentation of content in a more attractive and ludic way.*

**Resumo.** *Este trabalho apresenta uma proposta metodológica baseada nas premissas de aprendizagem significativa, a qual delinea uma estratégia de ensino do pensamento computacional de modo contextualizado, incorporando técnicas blended learning para promover um cenário mais flexível, bem como a apresentação dos conteúdos de uma maneira mais atraente e lúdica.*

## 1. Introdução

As constantes transformações na sociedade e a introdução de sistemas computacionais cada vez mais complexos têm trazido consigo desafios que demandam novas abordagens para serem resolvidos. Neste contexto, o pensamento computacional desponta como uma forma de resolver problemas, conceber sistemas e compreender o comportamento humano que se baseia em conceitos fundamentais da Ciência da Computação. Sua aplicabilidade para resolução de problemas nos mais diversos campos do conhecimento o torna uma habilidade fundamental para todas as pessoas, não apenas para cientistas da computação, despontando como um requisito elementar para a formação básica dos profissionais de todas as áreas nos próximos anos [Wing 2006]:

Departamentos de RH que atualmente valorizam candidatos que estejam familiarizados com aplicações básicas como o pacote *Microsoft Office*, mudarão suas expectativas buscando currículos que incluam habilidades de análises estatísticas e raciocínio quantitativo. [Davies, Fidler e Gorbis 2012]

Diante deste panorama, diversas iniciativas de incorporar conceitos fundamentais de computação - conhecimentos de lógica de programação, estruturação de algoritmos e a abstração necessária à resolução de problemas computacionais - vêm sendo propostas para os currículos do ensino fundamental e médio [Fernandes e Menezes 2011, Marques *et al.* 2011, Rapkiewicz *et al.* 2006]. Entretanto, o processo de ensino-aprendizagem destes conteúdos é potencialmente desafiador, apresentando dificuldades em vários aspectos, quer seja sob a perspectiva do professor ou do aluno, gerando discussões acerca de métodos, técnicas e ferramentas, bem como aspectos cognitivos e pedagógicos envolvidos.

Pereira Júnior (2005) elenca, por exemplo, os principais desafios enfrentados pelo professor no processo de ensino-aprendizagem de lógica de programação. Um ponto

inicial seria identificar os pré-requisitos necessários para o desenvolvimento das competências de construção de algoritmos e programação, de forma a poder trabalhá-los. Esta discussão não é consensual em relação à inclusão de um domínio de habilidades matemáticas prévias ou ao menos integradas, ou a se restringir aos princípios básicos de lógica considerando uma disciplina de caráter introdutório. Outro aspecto a ser considerado seria identificar/reconhecer nos alunos tais habilidades prévias de forma a aproveitá-las melhor. Adicionalmente, pode-se trabalhar a capacidade de abstração do aluno na busca por possíveis soluções para os problemas ou mesmo na escolha das estruturas de dados.

Algumas dificuldades comumente enfrentadas pelos alunos no processo de ensino-aprendizagem destes conteúdos são: O desenvolvimento do raciocínio lógico em contraste com a prática de memorizar o conteúdo; a falta de motivação do aluno gerada pelo despreparo ou mesmo desânimo quando o mesmo encara a disciplina como um obstáculo difícil de ser superado; a dificuldade na interpretação do problema, antes mesmo da dificuldade de interpretação de algum tipo de representação e mesmo dificuldade na aplicação das suas habilidades prévias, gerando medo e frustração [Pereira Júnior 2004, Rapkiewicz *et al.* 2006].

Pesquisas têm sido desenvolvidas a fim de propor abordagens para lidar com tais aspectos, as quais podem ser classificadas em três vertentes: 1. **Ferramentas**, que incluem trabalhos que apresentam ferramentas computacionais visando facilitar o processo de ensino-aprendizagem; 2. **Estratégias**, que abrangem trabalhos que discutem estratégias de ensino e/ou de avaliação de competências; 3. **Ferramentas e Estratégias**, são trabalhos que expõem estratégias suportadas por ferramentas computacionais [De Paula, Piva Júnior e Freitas 2009].

O presente trabalho segue a linha proposta na terceira vertente, utilizando a abordagem *blended learning* no intuito de apresentar os conteúdos propostos de modo mais abrangente e flexível, consistindo na combinação de diferentes estilos de aprendizagem e modelos de ensino:

*Blended learning* é a cuidadosa integração de experiências de aprendizagem em sala de aula com experiências de aprendizagem online. Isto não é apenas encontrar a combinação certa de tecnologias [...]. É, inerentemente, repensar e redesenhar a relação de ensino aprendizagem. [...] Não é o suficiente apenas entregar o conteúdo antigo em um novo meio [Garrison e Kanuka 2004].

Assim, este trabalho traz uma proposta metodológica que delinea um contexto de aprendizagem significativa suportada por uma linguagem de programação visual para o desenvolvimento do pensamento computacional. A fundamentação teórica e a estruturação metodológica do curso são apresentadas na Seção 2. A Seção 3 relata a experiência desenvolvida, e as análises, considerações finais e diretrizes futuras, são discutidas na Seção 4.

## 2. Fundamentação Teórica e Metodologia

A lógica de programação permite realizar um encadeamento coerente de instruções em uma sequência lógica para a resolução de problemas, constituindo-se com um dos saberes elementares para o desenvolvimento do pensamento computacional. Todavia, frequentemente estes conteúdos são ensinados de forma isolada, sem que haja uma correlação com conceitos de outras áreas do conhecimento, ou mesmo sem levar em conta

uma contextualização para o aprendiz. Desta forma, uma vez que o educando não compreende a aplicabilidade dos conteúdos que lhe foram ensinados, o processo de aprendizagem é prejudicado e esta não se torna significativa.

A aprendizagem significativa é o processo através do qual uma nova informação (um novo conhecimento) se relaciona de uma maneira não arbitrária e substantiva com a estrutura cognitiva do aprendiz, transformando o significado lógico do material de aprendizagem em significado psicológico para o sujeito, permitindo descobrir e redescobrir outros conhecimentos, e o conhecimento prévio se modifica pela aquisição de novos significados, caracterizando assim, uma aprendizagem prazerosa e eficaz [Ausubel 1982, Pelizzari *et al.* 2001].

A não arbitrariedade diz respeito ao fato de que um novo conhecimento se relaciona com conhecimentos pré-existentes especificamente relevantes, os quais Ausubel (1982) classifica como *subsunçores*. Desta forma, novos conteúdos podem ser aprendidos significativamente à medida que outros conceitos relevantes estejam adequadamente claros e disponíveis na estrutura cognitiva do sujeito e funcionem como pontos de ancoragem aos primeiros. A *substantividade* significa que aquilo que é incorporado à estrutura cognitiva é a substância do novo conhecimento, das novas ideias, e não precisamente as palavras usadas para expressá-las, uma vez que o mesmo conceito ser expresso de diferentes maneiras, as quais são equivalentes em termos de significados [Pelizzari *et al.* 2001].

A metodologia utilizada propõe um ambiente de aprendizagem significativa através de uma apresentação contextualizada dos conteúdos de lógica de programação. O cenário é esboçado a partir de técnicas *blended learning*, expondo os conceitos de uma maneira mais flexível e atraente, mesclando diferentes estilos de aprendizagem e modelos de ensino.

O conceito de *blended learning* (BL), onde vários ambientes de aprendizagem e atividades são combinados, não é uma abordagem recente, pois educadores sempre combinaram diversos recursos, incorporando uma variedade de atividades em ambientes diferentes [Hogbood 2010]. No entanto, uma nova visão de BL enfatiza as contribuições oferecidas pelos artefatos tecnológicos digitais, que envolve tecnologias baseadas na web (*e.g.* sala de aula virtual ao vivo, aprendizagem adaptativa e/ou colaborativa, *streaming* de vídeo, áudio e texto), ferramentas tecnológicas (*e.g.* lousas digitais, computadores, câmeras) e o modelo tradicional a fim de atingir um objetivo educacional. Ou ainda, mesclando diversas abordagens pedagógicas (*e.g.* behaviorismo, construcionismo, construtivismo, cognitivismo) para produzir um resultado de aprendizagem ideal com ou sem tecnologias de ensino [Garrison e Kanuka 2004]. Seguindo a abordagem de *blended learning*, o cenário de aprendizagem foi delineado a partir de dois pilares: *classroom learning* e *online learning (e-learning)*.

**Classroom Learning.** No ambiente de sala de aula, as atividades buscaram integrar ludicidade a um processo de aprendizagem colaborativa com ênfase na criatividade, de modo que os estudantes pudessem trabalhar em equipe e resolver problemas que serviriam como prelúdios para os conceitos de computação posteriormente abordados. Os aspectos teóricos dos assuntos selecionados foram apresentados através de aulas expositivas, que incluíam demonstrações, sessões de análises de vídeos, discussões, e outras atividades lúdicas as quais não envolviam o uso de computadores ou artefatos tecnológicos.

A criatividade era incentivada mediante a proposição de problemas e desafios diversos, alguns destes abrangendo conceitos ainda não abordados em sala de aula. Como exemplo, pode-se citar o desafio que consistia em ‘decodificar’ um *QR Code*: os estudantes receberam um *QR Code* e uma *url* incompleta que levava a um texto base para resolver uma lista de exercícios. O conteúdo do *QR Code* era um arquivo de áudio (em inglês) com o complemento da *url*. Consequentemente, os estudantes puderam descobrir sem intervenções externas o que é um *QR Code* e quais suas funcionalidades, além de explorar o uso da língua inglesa, amplamente utilizada durante o curso.

Ainda neste contexto, jogos digitais foram empregados para facilitar a compreensão acerca do processo de resolução de problemas, pois são apontados como ferramentas eficientes, que divertem enquanto motivam, facilitam o aprendizado e aumentam a capacidade de retenção do conteúdo apresentado [Marques *et al.* 2011, Rapkiewicz *et al.* 2006]. A utilização dos jogos abrangeu o ensino de conceitos como instruções, sequências lógicas e algoritmos.

**Online Learning.** A aprendizagem *online* (*e-learning*) caracteriza-se pela apropriação de vários tipos de mídias eletrônicas e tecnologias da informação e comunicação na educação, de modo que a experiência de aprendizagem é suportada pelo computador. Neste direcionamento, os *blogs* e as redes sociais têm sido amplamente utilizados como objetos de aprendizagem, com foco na construção coletiva de significados, fomentada pelas interações sociais entre os sujeitos e os pares, de modo que a ênfase está em como os alunos podem interagir entre si e ensinar uns aos outros.

A partir desta premissa foi criado um espaço de aprendizagem virtual no ambiente *Redu* [redu.com.br], onde os conteúdos foram disponibilizados e administrados, de forma que os estudantes tinham acesso irrestrito ao material do curso (mesmo de assuntos ainda não abordados): tutoriais, questionários, desafios, material de estudo, além de permitir a interação entre os alunos e a instrutora. O *Tumblr* [tumblr.com] foi utilizado para que cada estudante criasse o seu *portfólio* no formato de um *blog*. E, para abranger tópicos específicos, foram disponibilizadas vídeo-aulas no *Vimeo* [vimeo.com] que ensinavam passo a passo a realização de alguma atividade desafio ou expandiam algum conceito abordado em sala de aula.

Adicionalmente, dispositivos móveis foram incorporados ao longo do curso, os quais permitiram explorar conceitos introdutórios de computação móvel, e, no uso da ferramenta *App Inventor*, desenvolver aplicações móveis.

### **3. Pensamento Computacional: Uma experiência de ensino**

Visando aplicar a metodologia proposta foi desenvolvido o minicurso “Lógica de Programação com *App Inventor for Android*”. Este minicurso foi realizado durante o ano de 2012, em uma escola da rede pública estadual e ofertado a estudantes dos 1ºs e 2º anos do ensino médio, tendo sido atendidos 41 estudantes. A escolha em ofertar o minicurso aos estudantes do ensino médio buscou levar em consideração os saberes prévios dos estudantes, sobretudo em relação à língua inglesa, cujo conhecimento era necessário para a utilização da ferramenta. Adicionalmente, o minicurso serviria como um suporte pré-universidade para explorar conceitos introdutórios para estudantes que tivessem interesse em ingressar em cursos relacionados à Computação.

O minicurso consistiu dos seguintes módulos: Módulo 01. *Definições e conceitos elementares*, que abrangeu a compreensão dos conceitos fundamentais de computação e

sua relevância na resolução de problemas na sociedade; Módulo 02. *Desenvolvendo o Raciocínio Lógico*, que abordou a resolução de problemas de diversos domínios; Módulo 03. *Meus primeiros aplicativos*, que englobou a criação de aplicativos utilizando a linguagem de programação visual *App Inventor*. Um detalhamento mais abrangente das atividades desenvolvidas em cada módulo é apresentado nas subseções a seguir.

### **3.1. Módulo 1: Definições e Conceitos Elementares**

Neste módulo buscou-se apresentar alguns fundamentos teóricos sobre computação e informática, e identificar processos e situações cotidianas permeadas pelas ações destes agentes, como, por exemplo, no processamento de informações. Um dos principais objetivos consistiu em demonstrar a relevância da aplicação do pensamento computacional na resolução de problemas da sociedade, bem como proporcionar ao aluno uma visão contextualizada da computação dentro do seu espaço social, e, desta forma, trazer significados aos conteúdos apresentados e discutidos em sala de aula.

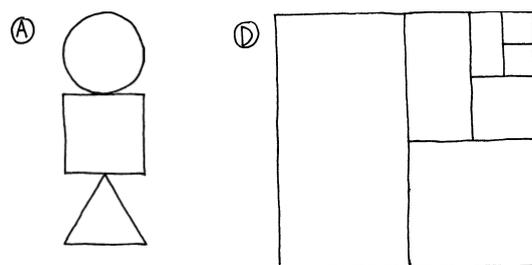
Inicialmente, foram propostas atividades onde os estudantes pudessem identificar, conceituar, analisar e discutir sobre o uso de artefatos computacionais na sociedade, bem como seus impactos positivos e negativos. Com este objetivo foi realizada uma sessão de análise do filme *Controle Absoluto* (2008), onde os estudantes deveriam enumerar circunstâncias de uso da computação durante o filme e discutir os seus impactos.

Em um segundo momento, os estudantes descreveram sobre os conceitos apresentados (*e.g.* O que é computação? O que é informática? O que é abstração?). As respostas dadas pelos alunos foram discutidas em sala, deste modo, eles puderam sugerir complementos às respostas dos colegas e elaborar uma definição a partir dos discursos coletivos, levando-os a construir o conhecimento destes conteúdos de uma forma substantiva. As discussões acerca da informação e o uso do computador como agente para o seu processamento foram realizadas com base no texto *cartoon* “O que é informação?” [Gonick 1984], onde os estudantes puderam identificar e compreender a transmissão, o armazenamento e o processamento da informação.

Estes conteúdos eram necessários a fim de servir como um nivelamento e prover um embasamento inicial, uma vez que nenhum dos participantes havia tido alguma experiência com programação, e estes não possuíam conhecimento formal acerca de informática e internet, de modo que, inicialmente, alguns apresentavam dificuldades com atividades básicas como navegar na Web, realizar *download* e instalação de programas.

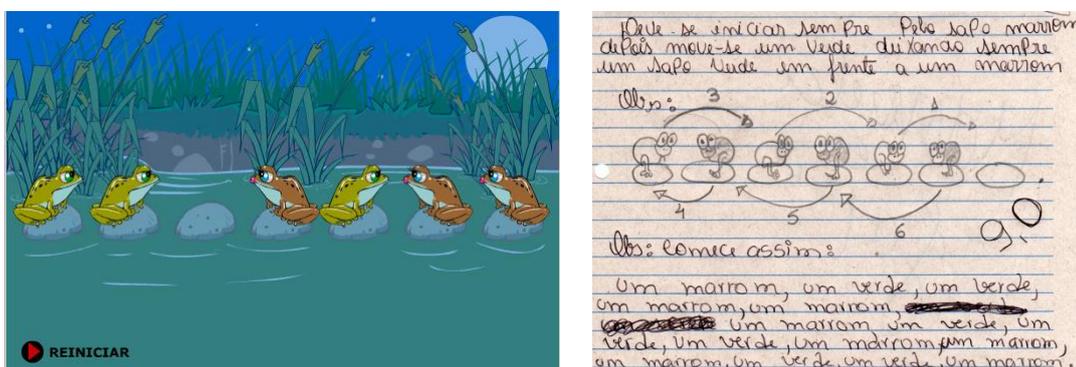
### **3.2. Módulo 2: Desenvolvendo o Raciocínio Lógico**

Conceitos de algoritmos foram introduzidos através da atividade lúdica “Seguindo Instruções – Linguagens de programação” [Bell, Witten e Fellows 2007]. Neste exercício, os estudantes foram convidados a “programar” os seus colegas em uma atividade onde deveriam fornecer, de forma tão clara quanto possível, instruções para eles pudessem desenhar algumas figuras. Depois de fornecidas as instruções, a figura original era revelada e os resultados eram comparados a partir dos seguintes parâmetros: 1. Muito semelhante, 2. Pouco Semelhante e 3. Diferente, sendo que os dois primeiros pontuavam positivamente, dados os resultados obtidos (Figura 01). O objetivo principal desta atividade foi provocar reflexões iniciais sobre a necessidade de precisão e clareza no desenvolvimento de algoritmos, direcionando os alunos a compreenderem a necessidade de expressar as soluções de modo que todos pudessem entender.



**Figura 01. Exemplos de figuras utilizadas na atividade “Seguindo Instruções – Linguagens de Programação”.**

Em outro momento, a fim de consolidar estes conceitos, os estudantes foram divididos em duplas e convidados a jogar: Canibais e Missionários, *Let it Flow*, *LightBot* e Sapos (Figura 02). Após jogarem, as duplas deveriam descrever instruções precisas, sucintas e claras para vencer cada jogo, e, para este fim, eles anotaram a pontuação obtida e o tempo gasto para jogar.



**Figura 02. Jogo Sapos e exemplo de instruções fornecidas pelos estudantes**

Estas instruções, que poderiam ser frases e/ou desenhos, foram trocadas com outros pares, a partir das quais, os colegas deveriam obter mesma pontuação e despende o mesmo tempo. Também deveriam pontuar (0-10) a instrução obtida, além de indicar possíveis dificuldades encontradas e/ou proporem uma nova solução para o problema, caso percebessem que esta poderia ser otimizada. Estas atividades buscavam evidenciar falhas comuns no uso da linguagem natural para descrição de resolução de problemas com o propósito de ratificar a necessidade do uso de linguagens de programação.

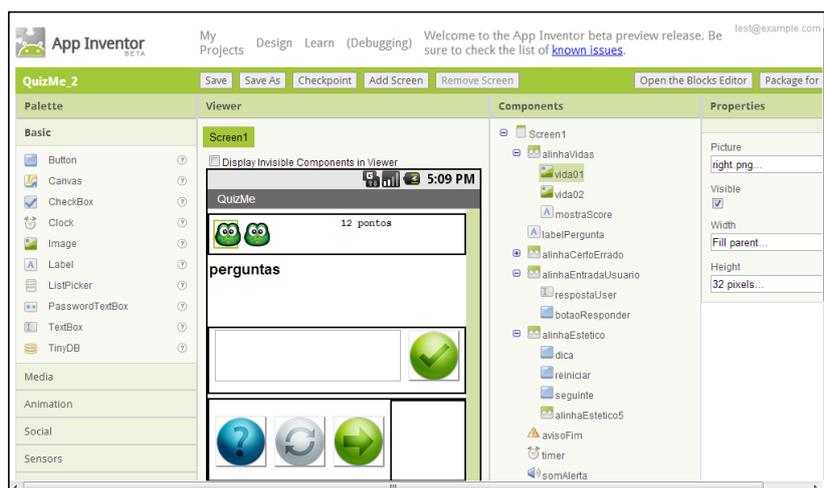
Ao longo do uso dos jogos, observou-se ainda uma deficiência por parte dos estudantes na estruturação do raciocínio lógico para resolução de problemas. Então, a fim de exercitar e aprimorar estes conceitos foi realizada uma mini maratona. Durante os dois dias da oficina os estudantes resolveram questões da modalidade Iniciação da Olimpíada Brasileira de Informática da SBC [2011]. Inicialmente os estudantes puderam realizar as atividades em grupo e discutir sobre as respostas encontradas. Aqueles exercícios nos quais os estudantes apresentavam dúvidas eram resolvidos pela instrutora, incorporando uma breve introdução ao estudo dos conectivos lógicos (*e*, *ou não*). Posteriormente, os estudantes realizaram os testes individualmente, apresentando uma significativa melhora em relação as atividades anteriores.

### 3.3. Módulo 3: Meus Primeiros Aplicativos

Os conceitos acerca de entrada e saída, variáveis, estruturas de controle, estruturas de

dados (lista), funções, dentre outros, foram apresentados a partir do desenvolvimento de aplicativos para *Android* na ferramenta *App Inventor* [appinventor.mit.edu/].

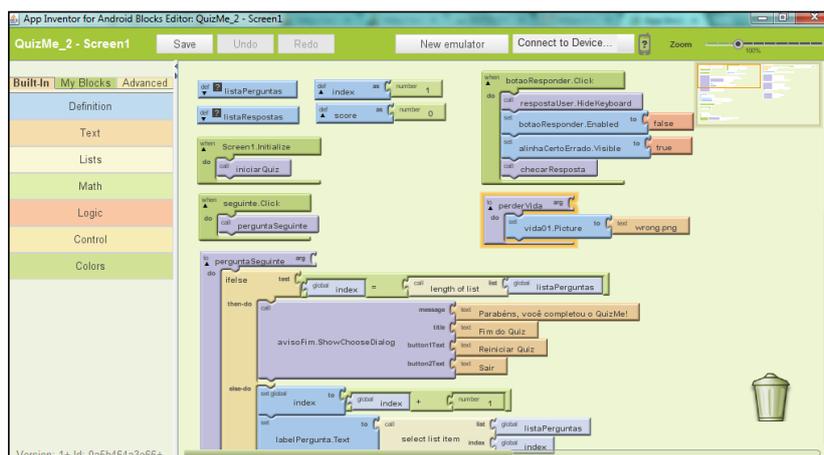
O *App Inventor* é uma linguagem de programação visual cujo maior diferencial é permitir que os usuários possam criar aplicativos para dispositivos móveis *Android* sem necessitar apropriar-se da sintaxe de uma linguagem de programação tradicional. A criação de um aplicativo neste ambiente consiste em uma atividade dividida em dois passos: 1. Criar a interface do usuário para o aplicativo, 2. Definir como o aplicativo se comporta. Estas duas tarefas são realizadas separadamente em duas janelas. A janela *App Inventor Designer* (Figura 03) é onde são adicionados os componentes, como botões, figuras, animações, caixas de texto, dentre muitos outros.



**Figura 03. Ferramenta App Inventor: Janela App Inventor Designer**

A janela *Blocks Editor* (Figura 04) por sua vez, permite ao usuário programar os componentes definidos na janela *App Inventor Designer*, encaixando blocos de código pré-definidos de modo similar ao *Scratch* [scratch.mit.edu/]. A medida que os aplicativos são desenvolvidos eles podem ser testados no emulador *Android* (que é acoplado ao *Blocks Editor*) ou mesmo podem executar o aplicativo em tempo real em um aparelho *Android* que esteja conectado ao computador, sem a necessidade de realizar *download* do aplicativo para o *smartphone*. Este é um aspecto especialmente importante porque permite os aplicativos sejam desenvolvidos e testados mesmo que o usuário não disponha de um *smartphone Android*.

Neste ambiente, os estudantes puderam compreender os conceitos básicos de programação, na medida em que desenvolviam aplicativos como: 1. *HelloDroid*, que exibe na tela alguma mensagem digitada pelo usuário, a fim de apresentar o ambiente, 2. *MoleMash*, um jogo simples onde o usuário deve acertar uma toupeira que se move aleatoriamente pela tela, 3. *MusicPlayer*, um tocador de música, 4. *QuizMe*, um jogo de perguntas e respostas com um *timer*.



**Figura 04. Ferramenta App Inventor: Janela Blocks Editor**

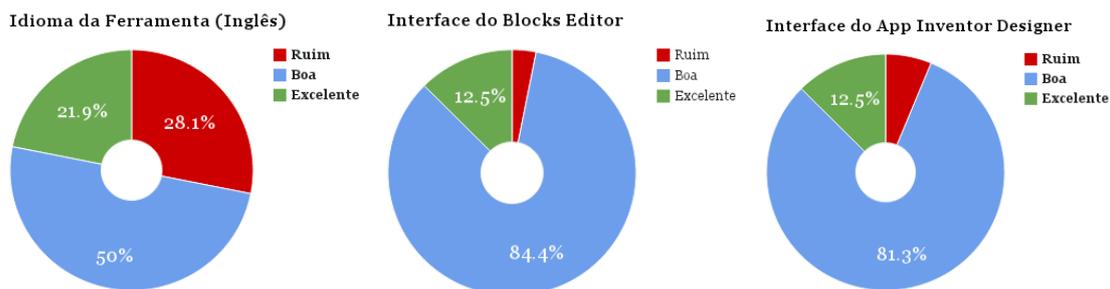
Em princípio, os aplicativos foram desenvolvidos passo a passo em sala de aula. Posteriormente os blocos de código eram apresentados incompletos ou com erros (Figura 04), a fim de capacitar o aluno a ler e entender um algoritmo e indicar onde um determinado problema está ocorrendo, bem como revisá-lo e melhorá-lo, propondo as alterações e/ou os complementos necessários.

Finalmente, os estudantes apresentaram uma solução (um aplicativo), para alguns problemas propostos que estendiam e combinavam os conceitos explanados em sala de aula, sendo esta atividade realizada em dupla ou individualmente.

#### 4. Conclusão

No intuito de avaliar a eficácia da metodologia aplicada no curso ofertado, além do grau de interesse e aprendizagem, a ferramenta utilizada, bem como as dificuldades encontradas, os estudantes responderam um questionário avaliativo. A experiência de aprendizagem de lógica de programação foi classificada como 'boa' por 71.9% dos estudantes, e como 'excelente' por 28.1%, inclusive, ao fim do curso, 75% dos participantes afirmaram ter interesse em prosseguir com estudos em Computação. Em relação às atividades desenvolvidas, 100% dos estudantes alegaram que o uso dos jogos digitais, das brincadeiras e desafios contribuíram de modo significativo ou muito significativo para a aprendizagem dos conceitos.

As dificuldades de aprendizagem residiram essencialmente sobre três pilares: 1. Leitura e interpretação de texto, 2. Matemática básica, 3. Idioma da ferramenta. Durante a realização das atividades envolvendo raciocínio lógico, os estudantes demonstraram dificuldades na compreensão dos problemas apresentados e erros, envolvendo operações matemáticas básicas (e.g. adição, divisão, etc.). Estas dificuldades foram atenuadas depois da realização da mini maratona que abrangia estes conceitos a partir de exercícios da OBI. O idioma da ferramenta (Inglês) foi um desafio em particular, pois a ferramenta atualmente não dispõe de uma versão no idioma nativo dos estudantes. Entretanto, apenas 28.1% dos estudantes consideraram 'Ruim' a experiência de utilizar o ambiente em Inglês, em detrimento de 50% que classificou a experiência como 'Boa' e 21,9%, como 'Excelente' (Figura 05).



**Figura 05. Avaliação da experiência de uso do ambiente App Inventor pelos estudantes**

Neste aspecto, o uso de uma linguagem de programação visual foi especialmente importante porque os estudantes não necessitaram apropriar-se da sintaxe de uma linguagem de programação tradicional. Muito embora a ferramenta não esteja disponível em português, dois aspectos tornaram a criação do aplicativo mais intuitiva: 1. O fato de a interface de programação do aplicativo ser composta por blocos de código conectáveis, e 2. A interface do usuário ser projetada ‘arrastando e soltando’ componentes.

Adicionalmente, alguns aspectos da infraestrutura do laboratório de informática impactaram na fluidez da realização das atividades (*e.g.* a falta de internet): como a ferramenta utilizada era baseada na Web, visto que o *App Inventor Designer* é executado diretamente no navegador e os projetos são armazenados nos servidores do *App Inventor*, foi necessário buscar uma solução alternativa que pudesse ser executada sem o uso de internet. Assim, foi usada uma versão *off-line* do *App Inventor* que utiliza os arquivos fornecidos pelo MIT para executar um servidor pessoal do *App Inventor* localmente.

Não obstante, os estudantes demonstraram motivação, engajamento e interesse pelos conteúdos explanados, bem como uma apropriação significativa dos conceitos, conforme observado no decorrer da realização das atividades propostas durante o curso. Especialmente, o ambiente colaborativo, predominantemente permeado por desafios, estimulou os estudantes a aprenderem a trabalhar em equipe e resolverem de modo criativo e autônomo os problemas propostos.

A fim de difundir estes conhecimentos para um número maior de alunos pretende-se além de atender outras escolas, transpor, inclusive, os limites geográficos, através de um curso à distância. Devido a necessidade de recursos para a ferramenta no idioma nativo, está em desenvolvimento um guia de referência que abrange os principais tópicos desta, bem como a criação de aplicativos que utilizam suas principais funcionalidades, estes e outros recursos estarão disponíveis livremente para a comunidade científica.

## Referências

- Ausubel, D. P. (1982). *A aprendizagem significativa: a teoria de David Ausubel*. São Paulo: Moraes.
- Bell, T.; Witten, I.H. E Fellows, M. (2007) “Computer Science Unplugged”. <http://csunplugged.org>, Março
- Controle Absoluto. Direção: D.J. Caruso, Produção: Steven Spielberg. EUA: DreamWorks Pictures, 2008, 1 DVD.
- Davies, A.; Fidler, D. e Gorbis, M. (2012) “Future Work Skills to 2020”.

<http://www.iftf.org/futureworkskills2020>, Fevereiro.

- De Paula, L. Q., Piva Júnior, D. E Freitas, R.L. (2009) “A importância da leitura e da abstração no processo de formação do raciocínio lógico-abstrato”. In: Anais do XVII Workshop sobre Educação em Informática. Bento Gonçalves: RS.
- Fernandes, C. S.; Menezes, P. B. (2011) “Metodologia do Ensino de Ciência da Computação: Uma Proposta Para Criança”. In: Anais do Workshop de Informática na Escola. Fortaleza: CE.
- Garrison, D. R.; Kanuka, H. (2004). “Blended learning: Uncovering its transformative potential in higher education”. In: The Internet and Higher Education. p. 95-105.
- Gonick, L. (1984) Uma Introdução Ilustrada à Computação. Ed. Harper & Row do Brasil Ltda. São Paulo: SP.
- Hogbood, B. (2010) “Blended learning”. <http://www.learnnc.org/lp/pages/6722>, Abril.
- Marques, D. L., Costa, L. F. S., Silva, M. A. A., Rebouças, A. D. D. S. (2011) “Atraindo Alunos do Ensino Médio para a Computação: Uma Experiência Prática de Introdução à Programação utilizando Jogos e Python”. In: Anais do XVII Workshop sobre Educação Informática. Aracaju: SE
- Pelizzari, A., Kriegl, M. L., Baron, M. P., Finck, N. T. L., Dorocinski, S. I. (2001) “Teoria da Aprendizagem Significativa Segundo Ausubel”. In: Revista PEC, Curitiba, v.2, n.1, p.37-42.
- Pereira Júnior, J. C. R. e Rapkiewicz, C. E. (2004) “O Processo de Ensino e Aprendizagem de Algoritmos e Programação: Uma Visão Crítica da Literatura”. In: Anais do III Workshop de Educação em Computação e Informática do estado de Minas Gerais. Belo Horizonte: MG.
- Pereira Júnior, J. C. R., Rapkiewicz, C.E.; Delgado, C. e Xexeo, J.A.M. (2005) “Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio”. In: Anais do XIII Workshop de Educação em Computação. São Leopoldo: RS.
- Rapkiewicz, C. E., Falkembach, G., Seixas, L., Rosa, N. S., Cunha, V. V. Klemann, M. (2006) “Estratégias Pedagógicas no Ensino de Algoritmos e Programação Associadas ao Uso de Jogos Educacionais”. In: CINTED-UFRGS, V. 4 N° 2.
- SBC. (2011) “Olimpíada Brasileira de Informática”. <http://olimpiada.ic.unicamp.br/>, Abril.
- Wing, J. “Computational Thinking”. (2006) In: *Communications of the ACM*. v.49, n.3. p. 33-35.