

## ***App Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação***

**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.** *The challenges in teaching introductory programming have driven the search for methods and tools to support this process. In this context, there is the App Inventor for Android, a visual language, block-based, focused on building apps for Android mobile devices. This paper presents an approach that uses this characteristic to provide more engaging experiences of teaching and learning in introductory courses in logic programming.*

**Resumo.** *Os desafios no ensino introdutório de programação têm impulsionado a busca por métodos e ferramentas que auxiliem este processo. Neste contexto surge a App Inventor for Android, uma linguagem visual, baseada em blocos, voltada à criação de aplicativos para dispositivos móveis Android. O presente trabalho apresenta uma abordagem que se utiliza desta particularidade para propiciar experiências mais atrativas de ensino-aprendizagem em cursos introdutórios de lógica de programação.*

### **1. Introdução**

O processo de ensino de lógica de programação é permeado por diversos desafios que englobam desde as dificuldades dos alunos na compreensão do problema proposto e em questões relativas à abstração, até mesmo falta de consenso no que diz respeito aos requisitos prévios necessários para um desempenho satisfatório na aprendizagem [Falkembach et al. 2003].

Segundo Cristovão (2008), para que o aluno consiga se expressar adequadamente a fim de solucionar um problema é necessário que ele possua domínio da linguagem adotada. Delgado *et al.* (2004) por sua vez, enfatiza a formalização mais próxima da linguagem natural, assim o aluno teria mais facilidade para expressar uma solução. No entanto, as linguagens tradicionalmente utilizadas em cursos introdutórios de programação possuem uma sintaxe rebuscada, e nem sempre permitem que o educando foque na resolução do problema ao invés de preocupar-se com o aprendizado sintático da linguagem. Adicionalmente, os programas desenvolvidos a partir destas linguagens, em geral, abordam assuntos não contextualizados emoldurando situações de aprendizagem em que os estudantes não percebem a importância ou a aplicabilidade do conteúdo que aprendem, e sentem-se desmotivados a dar continuidade aos estudos.

Visando atenuar os impactos negativos no processo de ensino-aprendizagem de programação, diversas alternativas têm sido buscadas. Uma das principais vertentes nesta proposta indica que a prática inicial de ensino seja suportada por ferramentas computacionais, que permitam ao aluno experimentar, descobrir, testar e errar suas respostas [Cristovão 2008, Delgado *et al.* 2004]. Estas propostas apresentam situações de aprendizagem que envolvem diversos recursos, abrangendo a utilização de linguagens de

programação visual [França, Costa e Amaral 2012, Scaico *et al.* 2012a], o desenvolvimento de jogos por e/ou para alunos [Marques *et al.* 2011, Rapkiewicz *et al.* 2006], a robótica educacional [Silva, Silva e Melo 2011], aplicativos para dispositivos móveis [Scaico *et al.* 2012b], dentre outros.

Considerando os dispositivos móveis, especificamente os que dispõem da plataforma *Android*, algumas aplicações disponíveis envolvem fundamentos de lógica de programação, conceitos de estruturas de dados com disciplina de acesso tais como pilha, fila e algoritmos de ordenação. Um exemplo de aplicativo nesta classe é o *Sortko* [bit.ly/WLbrgC], o qual fornece aos alunos a opção de escolher um dentre os algoritmos de ordenação (*Bubble Sort*, *Insertion Sort*, *Heap Sort*, *Quick Sort*), gerando uma sequência aleatória de números, a ser ordenada mediante gestos específicos de interação. Durante o processo de ordenação, o *Sortko* informa aos estudantes sobre o seu sucesso na realização das *switches* (trocas de elementos), fornecendo-lhes mensagens para auxiliar o seu progresso, tornando a experiência de aprender algoritmos de ordenação mais intuitiva.

Desenvolvido para auxiliar a aprendizagem dos conceitos de pilha e fila, o aplicativo *Aestuu's Gem* [bit.ly/ZX2wnr] tem como personagem principal um monstrinho que deve coletar joias e pedras preciosas, depois deve empilhá-las ou enfileirá-las de acordo com o padrão de cores solicitado. Tem-se também o jogo em desenvolvimento *Castelo dos Enigmas* [Scaico *et al.* 2012b], que busca apresentar conceitos de programação, e objetiva ensinar as estruturas de uma linguagem de programação e os conceitos básicos para a elaboração de um algoritmo.

Na *web*, além dos amplamente utilizados *applets*, atualmente tem-se diversas plataformas que disponibilizam cursos e tutoriais em níveis introdutórios, intermediários e avançados, acerca de diversas linguagens de programação. Muitas destas são interativas e apropriam-se de elementos e mecânicas de *design* de jogos (*e.g.* modularização do curso em níveis, o uso de *badges*, *rankings* e premiações) como estratégias para fomentar o engajamento do usuário e tornar o processo de aprendizado mais divertido.

Um bom exemplo deste tipo de abordagem é a plataforma *Code Avengers* [codeavengers.com], que disponibiliza cursos *online* interativos que ensinam o básico de *web design*, especificamente *HTML5*, *CSS3* e *JavaScript*, cuja interface lúdica baseada em desafios é inspirada em um filme de sucesso. Similarmente, pode-se citar a *Code Academy* [codeacademy.com/], a *Code Racer* [coderace.me/] e a *Try Ruby* [tryruby.org/], todas estas plataformas dispõem de um ambiente interativo onde o usuário digita o código no IDLE e recebe o *feedback* acerca dos erros e acertos.

Considerando as linguagens de programação visual, dispõem-se de ambientes tais como *Kodu*, *Scratch* e *Snap!*. O *Kodu* é um software desenvolvido pela Microsoft, que permite aos usuários desenvolver jogos sem conhecimento prévio em programação. O usuário escolhe os *worlds* (cenários de jogos pré-definidos) e insere personagens. A programação dos personagens é feita através da combinação de movimentos e ações possíveis listados no *world*, apenas arrastando e soltando estes componentes pré-definidos na tela. Assim, a programação é orientada a eventos, de modo que uma ação tem a seguinte sintaxe: “*When <evento> ... do...*”. O usuário pode criar jogos completos em mundos virtuais 3D, além de poder compartilhá-los com outros usuários.

O *Scratch* [scratch.mit.edu/] é um ambiente interativo desenvolvido em 2007 pelo Instituto de Tecnologia de *Massachusetts* (MIT), onde a criatividade e a inteligência são peças-chave. Este *software* permite a criação de histórias interativas, animações, jogos, música e diversos outros tipos de arte onde o processo de construção de um *design* seja necessário, sendo amplamente utilizado no ensino de lógica de programação. O *Snap!*

[snap.berkeley.edu/] foi desenvolvido por Jens Mönig com contribuições de estudantes de Berkeley e de outras universidades. É uma reimplementação estendida do *Scratch*, sendo, portanto, uma linguagem visual de programação do tipo arrastar e soltar, que tem como diferencial permitir que os usuários criem seus próprios blocos. O *Snap!* também dispõe de listas e procedimentos, tornando-o adequado para uma introdução mais contundente em Ciência da Computação para o ensino médio ou mesmo para estudantes universitários. Atualmente dispõe de versões *betas* de pacotes adicionais para *Lego Mindstorms* e *Nintendo Wiimote*.

O *App Inventor for Android*, objeto de estudo do presente trabalho, é um ambiente visual de programação em blocos, o qual permite o desenvolvimento de aplicativos para dispositivos móveis *Android* de uma maneira consideravelmente simples, principalmente se comparada às linguagens de programação tradicionais. Seu ambiente gráfico possibilita o ensino de conceitos de lógica de programação de uma forma atraente e motivadora para estudantes do ensino médio e superior.

Além da proposta metodológica de ensino de programação através do *App Inventor for Android*, este trabalho apresenta também relatos de experiências da aplicação desta metodologia, nos quais os participantes aprendem lógica de programação enquanto desenvolvem aplicativos para dispositivos móveis, especificamente para a plataforma *Android*.

Uma descrição mais detalhada desta ferramenta é apresentada na Seção 2 deste artigo. Relatos de aplicações pedagógicas no ensino de lógica de programação no Brasil e no exterior compõem a Seção 3. A Seção 4 apresenta as considerações finais dos estudos realizados e trabalhos futuros.

## 2. O Ambiente *App Inventor*

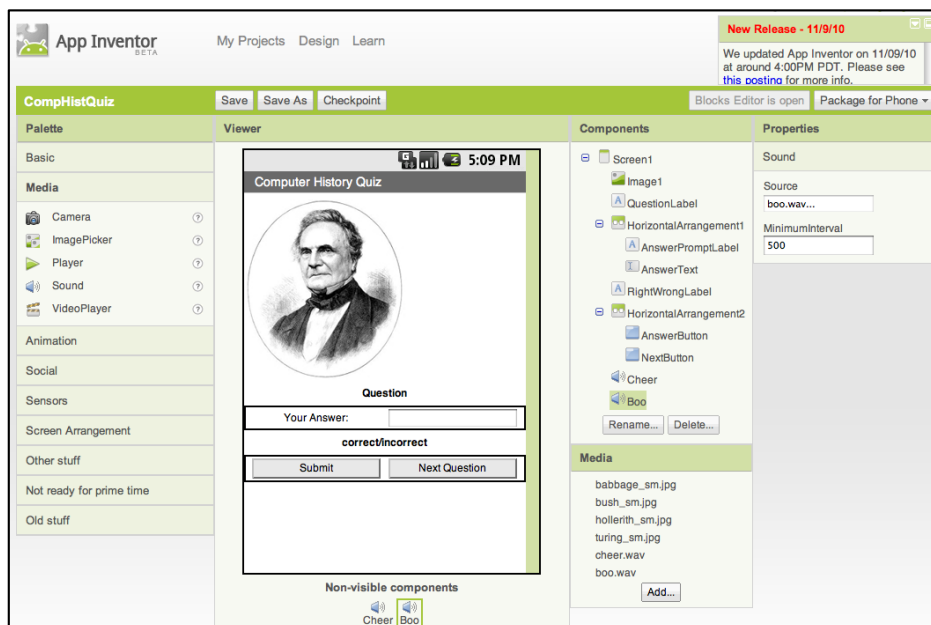
*App Inventor for Android* é uma linguagem visual de programação para a criação de aplicativos para dispositivos móveis baseados em *Android*, desenvolvida conjuntamente pelo Google e MIT. O *App Inventor* possui um ambiente de desenvolvimento baseado em blocos, entretanto, seu diferencial consiste em possibilitar aos usuários criar aplicações que incorporem serviços baseados na *web*, interação com redes sociais, leitura de códigos de barra, interação com sensores de orientação e geolocalização, e de funcionalidades como *text-to-speech* e reconhecimento de fala [MIT 2012].

Inspirado nas linguagens *Logo* e *Scratch*, o *App Inventor* propicia um ambiente de aprendizagem baseado no construcionismo, uma vez que permite aos estudantes criarem aplicações à medida que descobrem e exercitam sua criatividade, tornando o aprendizado mais lúdico. Segundo Abelson (2009), esta linguagem foi projetada com o objetivo de tornar o processo de ensino-aprendizagem dos fundamentos de computação mais atraente, visando provocar mudanças positivas nas experiências: “[...]tendo como alvo as pessoas e suas interações com os outros e com o mundo ao seu redor, em detrimento de cenários que incluem estudantes desmotivados, sentados em laboratórios escrevendo programas que ninguém jamais usará”.

A criação de aplicativos com o *App Inventor* é bastante intuitiva e não exige conhecimento prévio avançado em programação. Além disso, os aplicativos criados podem ser utilizados em quaisquer dispositivos com a plataforma *Android*. O *App Inventor* é baseado em **componentes**, os quais são manipulados por **eventos**, **métodos** e **propriedades** (e.g. *Button*: componente, *Button.Color*: propriedade,

*whenButton.Clicked*: evento). A programação de uma aplicação no *App Inventor* é orientada a eventos, ou seja, o comportamento dos componentes depende, em sua maioria, de eventos provocados pela interação do usuário com o aplicativo.

O desenvolvimento de uma aplicação nesta ferramenta é realizado através de duas janelas: *App Inventor Designer* e *Blocks Editor*. A janela *App Inventor Designer* é executada a partir do navegador e permite criar visualmente a interface do usuário, ao clicar e arrastar os componentes da *Palette*, tais como botões, caixas de texto, figuras, animações, sons, entre outros, para o *Viewer* (Figura 01).



**Figura 01. App Inventor Designer**

A janela *Blocks Editor* (Figura 02) por sua vez, permite controlar o comportamento dos componentes definidos na *App Inventor Designer*. Neste ambiente, o usuário encontra blocos conectáveis, que podem ser *eventos* ou *métodos*, em uma interface do tipo arrastar e soltar. Estes blocos operam *strings* e listas, realizam ações de controle (e.g. *if*, *else*, *foreach*, etc.) e operações matemáticas, entre outras funcionalidades. É possível executar o teste do aplicativo diretamente em um dispositivo *Android* (*smartphone* ou *tablet*) que esteja conectado ao computador ou através do emulador que acompanha o *Blocks Editor*.

O site oficial do *App Inventor* [MIT 2012] dispõe de uma ampla variedade de materiais de estudo e referência, que abrangem desde a instalação, a criação do primeiro aplicativo, bem como uma série de tutoriais e os respectivos 'códigos', criados por professores e estudantes de várias instituições, além de diversos relatos de experiência de uso do *App Inventor* nos mais variados contextos.

Atualmente, o *App Inventor* também dispõe do *MIT App Inventor Community Galery* [gallery.appinventor.mit.edu/], destinado a ser um repositório de aplicativos desenvolvidos por usuários de todo o mundo, onde é possível interagir com outros usuários, realizar *download* dos blocos de código e dos aplicativos criados. Há ainda uma competição anual que premia os autores dos melhores aplicativos.

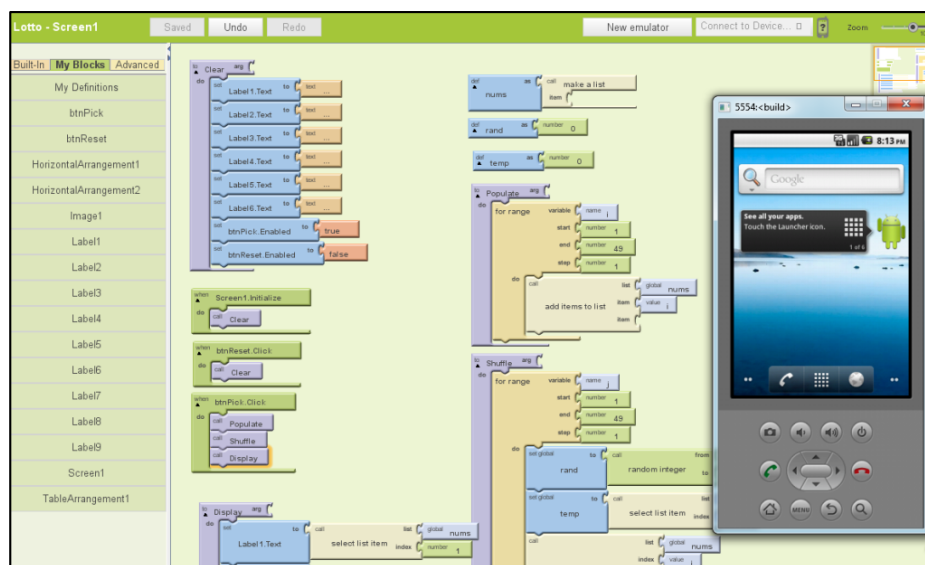


Figura 02. App Inventor: Janela Blocks Editor. Exemplo de Aplicativo e Emulador em Execução

### 3. Aplicações Pedagógicas no Ensino de Lógica de Programação

O *App Inventor* (AI) tem sido amplamente utilizado em diversas instituições de ensino. Inicialmente, o AI foi utilizado em projetos-piloto nos Estados Unidos, patrocinados pela *Google*, em instituições de ensino que tivessem interesse em explorar o seu potencial como uma plataforma de ensino para cursos introdutórios de programação no ensino médio e superior [Abelson 2010].

Atualmente têm-se diversos relatos sobre a utilização do *App Inventor* que abrangem desde o ensino fundamental, médio e superior, até mesmo a formação de professores. A aplicação do *App Inventor* em contextos educativos tem demonstrado o seu potencial didático-pedagógico, principalmente em cursos de programação voltado a iniciantes. Wolber (2010b) descreve a realização de um curso introdutório destinado a estudantes sem conhecimento prévio em programação usando a ferramenta *App Inventor*. Neste curso, os estudantes desenvolveram aplicativos que envolviam a resolução de problemas do cotidiano, por exemplo: 1. O aplicativo “*Android Where’s My Car*”, baseado em sistemas de geolocalização, permite encontrar onde o carro foi estacionado, 2. O “*Droid Muni*” que notifica sobre a chegada do próximo ônibus, 3. “*No Text While Driving*” é um aplicativo que impede que o usuário se distraia com o recebimento de novas mensagens enquanto dirige, silenciando os tons de alerta do *smartphone* e enviando SMS ao remetente informando que o usuário está dirigindo.

Estes aplicativos foram criados pelos alunos após poucas semanas de curso. Wolber (2010b) salienta que os estudantes mostraram-se muito mais motivados por estarem desenvolvendo aplicativos que poderiam ser instalados e executados em seus dispositivos Android, bem como demonstraram uma aprendizagem satisfatória dos conteúdos apresentados: “*Os aplicativos criados pelos meus alunos foram incríveis, especialmente dado que eles não tinham experiência anterior em programação [...]*”.

Outro exemplo de aplicação do *App Inventor* no ensino introdutório de programação foi realizado visando responder a pergunta: “*Would App Inventor be a suitable platform for bringing computational thinking to K-12?*”, Lake et. al (2011)

apresentam um relato de experiência que explorou diversas possibilidades de uso do *App Inventor* para uma introdução ao pensamento computacional com professores e estudantes do ciclo *K-12* dos Estados Unidos.

### **3.1. Explorando Possibilidades com o App Inventor: Relatos de Experiência**

Embora a plataforma esteja disponível apenas em inglês, este não tem demonstrado ser um fator restritivo para a sua aplicação em contextos educacionais no Brasil. A fim de explorar o potencial didático-pedagógico da ferramenta *App Inventor* como uma possibilidade para o ensino de lógica de programação, experiências de ensino têm sido feitas, objetivando, dentre outros, efetuar estudos de caso.

Inicialmente, realizou-se uma experiência de ensino de lógica de programação suportada por uma abordagem de *blended-learning* utilizando o *App Inventor*. Esta atividade extracurricular teve como público alvo estudantes dos 1<sup>os</sup> e 2<sup>os</sup> anos do ensino médio de uma escola da rede pública estadual [Gomes e Melo 2012, 2013c]. Além de atividades lúdicas e jogos digitais para explanar aspectos teóricos de computação, processamento da informação e raciocínio lógico, os estudantes puderam desenvolver aplicativos para Android utilizando a ferramenta *App Inventor*. Os aplicativos desenvolvidos pelos estudantes exploravam conceitos iniciais de programação, como variáveis, estruturas de controle, estruturas de decisão e listas. Os resultados expressaram, além de uma aprendizagem significativa dos conteúdos apresentados, uma possibilidade de atrair talentos para computação: 75% dos participantes manifestaram interesse em continuar os estudos em informática no nível superior.

Além da experiência de ensino descrita anteriormente, ao longo do ano de 2013, foram realizadas oficinas, nas quais os conteúdos eram apresentados com um enfoque mais prático, cujo público alvo passou a incluir também estudantes de instituições de ensino superior [Gomes e Melo 2013a, 2013b]. O conteúdo programático foi subdividido em quatro tópicos principais: 1. Introdução ao Desenvolvimento de Aplicativos para Dispositivos Móveis, 2. Linguagem de Programação Visual *App Inventor*, 3. Emulador & Testes, 4. Publicando os Aplicativos na Web. Nestas oficinas, os principais objetivos consistiam em: 1. Explorar conceitos introdutórios de computação móvel através da criação de aplicativos educacionais para dispositivos móveis *Android*, 2. Explorar os benefícios e técnicas de programação pelo usuário final para a aprendizagem de conceitos introdutórios de lógica de programação, 3. Promover experiências lúdicas de aprendizagem através da ferramenta *App Inventor*.

Em cada oficina, uma breve introdução ao desenvolvimento de aplicativos para dispositivos móveis foi apresentada, visando fornecer uma contextualização ao cenário de aprendizagem. Iniciando a fase prática, foram desenvolvidos quatro aplicativos que buscavam abranger os principais tópicos de lógica de programação: 01. *HelloDroid*, 02. *MiniPaint*, 03. *MoleMash*, 04. *QuizMe*. Os aplicativos, desenvolvidos ao longo das oficinas, tinham por objetivo uma apresentação lúdica e gradual dos conteúdos propostos, de modo a estender os conhecimentos adquiridos à medida que o participante avança no curso, conforme são descrição a seguir.

#### **3.1.1. Aplicativo *HelloDroid***

Assim como em linguagens tradicionais, a funcionalidade deste aplicativo é imprimir uma mensagem na tela, que, originalmente é a frase *Hello World*. Este aplicativo tem por

objetivo apresentar os principais recursos do ambiente *App Inventor* permitindo aos participantes familiarizar-se com sua interface. O aplicativo consiste de uma caixa de texto onde o usuário poderia digitar uma mensagem, a qual seria exibida na tela ao pressionar o botão “OK”.

### 3.1.2. Aplicativo *MiniPaint*

Um aplicativo de criação e edição de imagens simples, com recursos básicos, que permite ao usuário, por exemplo, riscar traços e pontos de cores variadas, importar uma imagem da galeria para edição, salvar o desenho criado no dispositivo, dentre outras funcionalidades. Com este aplicativo, os participantes puderam compreender conceitos de lógica de programação, tais como: constantes, variáveis e índices.

### 3.1.3. Aplicativo *MoleMash*

Este jogo simples simula uma toupeira percorrendo aleatoriamente um conjunto de pontos na tela, demarcados a partir do sistema cartesiano  $(x,y)$ , de modo que a cada vez que o usuário acerta a toupeira na tela, ele ganha pontos. A partir deste aplicativo, o participante estende o conhecimento aprendido durante a criação do aplicativo anterior, bem como, tem noções introdutórias de estruturas de controle, operações aritméticas e relacionais.

### 3.1.4. Aplicativo *QuizMe*

Este aplicativo consiste de um jogo de perguntas e respostas, com pontuação e *timer*. No desenvolvimento deste aplicativo, o participante aprofunda, dentre outros, o conhecimento sobre estruturas de controle e utiliza recorrentemente operações aritméticas (*e.g.* aumentar/ diminuir a pontuação em caso de acerto/erro), relacionais e lógicas (*e.g.* comparar a resposta do usuário com a resposta esperada).

Os blocos de código pré-definidos permitiram que os participantes tivessem maior liberdade criativa. À medida que os conteúdos eram ensinados e os aplicativos criados, os estudantes modificavam a aplicação, testavam novos blocos, ao passo que compreendiam melhor a lógica de resolução do problema e exploravam a ferramenta.

Na fase de Emulação e Testes, os participantes aprenderam a efetuar testes dos aplicativos utilizando o emulador, que já vem acoplado ao *App Inventor*, ou mesmo testar diretamente em seus *smartphones*, bem como realizar *download* de seus “códigos” (.rar). Com o objetivo de compartilhar alguns dos aplicativos criados durante as oficinas, foi criado um mural coletivo *online* utilizando a ferramenta *Lino.it* [en.linoit.com/] onde os participantes compartilharam através de um *QRCode*, criado com a aplicação *Unitag* [unitaglive.com/qrcode], um aplicativo escolhido por eles. Assim qualquer usuário poderá baixar o aplicativo apenas apontando a câmera do *smartphone* com um leitor de *QRCode*.

No decorrer das oficinas, pode-se observar o entusiasmo dos estudantes por terem a oportunidade de desenvolver aplicações que poderiam ser executadas em seus dispositivos, de modo que alguns passaram a conectar o aparelho ao computador para testar diretamente no *smartphone* as aplicações criadas.

## 4. Considerações Finais

O presente trabalho buscou apontar diretrizes para um melhor aproveitamento do potencial educativo dos dispositivos móveis, uma vez que novas possibilidades de ensino de programação envolvendo estas tecnologias são cruciais para delinear experiências de

aprendizagem mais atraentes, significativas e inovadoras.

Segundo Wolber (2010a), a possibilidade de criar aplicativos para dispositivos móveis, que possam atender necessidades reais, é significativamente importante para a motivação dos estudantes. Assim, o *App Inventor* proporciona um ambiente de aprendizagem significativa, permitindo que os usuários *transcendam* as fronteiras de execução, anteriormente limitada a um computador:

A chave de uma perspectiva de aprendizagem é a motivação: as pessoas estão significativamente mais motivadas para aprender conceitos de programação porque elas estão construindo algo divertido que elas podem usar em sua vida diária. Ensinando *App Inventor*, eu fiquei maravilhado em como os estudantes têm trabalhado arduamente - o nível de motivação deles foi significativamente maior do qualquer curso de caráter introdutório que eu já tenha lecionado.

Ao criar aplicativos que possam ser utilizados em seus *smartphones* e que incorporem funcionalidades específicas de dispositivos móveis como a mobilidade e o uso do sistema de geolocalização são desenhadas conexões reais entre o cotidiano do aprendiz e os conceitos aprendidos. Desta forma, os conceitos não apenas fazem sentido para os educandos, mas têm importância, pois a possibilidade de criar aplicativos reais os deixa mais motivados.

Além disso, a detecção intuitiva de erros conceituais cometidos ao longo do desenvolvimento no *App Inventor* possibilita apresentar os conteúdos a partir de uma abordagem construcionista, uma vez que a resolução lógica do problema, a concepção de solução do aluno, está representada através dos blocos de código conectados. A premissa construcionista ressalta o aprendiz como participante ativo, condutor do seu processo de aprendizagem: experimentando, descobrindo, errando e depurando, mediante o ciclo de descrever- executar – refletir – depurar. Desta forma, ao invés de apenas fornecer o conhecimento ao aluno e levá-lo a fornecer respostas, o computador recebe as instruções e operações que devem ser executadas para produzir as respostas desejadas pelo aluno [Almeida 2000].

Assim sendo, com o uso do *App Inventor*, o estudante além de criar softwares com conteúdos significativos, obtém também um *feedback* imediato às ações operadas sobre os blocos, podendo comparar os resultados obtidos com a solução proposta, de modo que o abstrato torna-se concreto e permitindo ainda que o estudante compreenda o porquê da solução proposta às vezes não gerar o resultado esperado [Almeida 2000]: “[...]O programa fornece importantes pistas sobre o pensamento do aluno, uma vez que o pensamento está descrito explicitamente e a resposta do computador permite comparar o previsto com o obtido.”.

A revisão da literatura apresentada permite constatar o potencial didático-pedagógico do *App Inventor*, despontando como uma possibilidade para auxiliar o ensino de lógica de programação e difundir o pensamento computacional, além de propiciar uma interessante abordagem para inserção das tecnologias móveis em contextos educativos. A interface gráfica de fácil uso possibilita aos estudantes explorar os conceitos básicos de programação e algoritmos de modo mais intuitivo, sem que estes precisem se ater a detalhes de sintaxe, e assim concentrarem-se na concepção da resolução dos problemas. Adicionalmente, os aplicativos desenvolvidos não se atêm a um teor estritamente pedagógico, são programas *reais*: podem ser instalados e executados em qualquer dispositivo móvel com a plataforma *Android* e estão inseridos nos contextos sociais dos



aprendizes, tornando a programação mais envolvente e a experiência motivadora.

No intuito de contribuir para a disseminação do uso desta ferramenta, está em desenvolvimento um guia de referência rápido em português, visando, dentre outros, atingir o público que não tem familiaridade com línguas estrangeiras. O guia abrange os principais tópicos para os iniciantes, bem como ideias e recursos para que professores possam usar esta ferramenta em sala de aula. Outra diretriz futura consiste na criação de um curso a distância, a fim de despertar o interesse naqueles que têm alguma curiosidade ou afinidade com tecnologias, atendendo a qualquer interessado que não necessariamente disponha de um conhecimento formal sobre este assunto, atraindo potenciais talentos para Computação. Adicionalmente, pretende-se criar um material de estudo abrangente, contendo diversos aplicativos e tutoriais descritos passo a passo.

## Referências

- Abelson, H. (2009) “*App Inventor for Android*”. <http://googleresearch.blogspot.com/2009/07/app-inventor-for-android.html>, Agosto.
- Abelson, H. and Friedman, M. (2010) “*App Inventor – A view into learning about computers through building mobile applications*”. <http://www.sigcse.org/sigcse2010/attendees/supportersessions.php>, Agosto.
- Almeida, M.E. (2000) “*Informática e formação de professores. Secretaria de Educação a Distância*”. ProInfo- Brasília: Ministério da Educação, SEED.
- Cristóvão, H. M. (2008). “*Aprendizagem de Algoritmos num Contexto Significativo e Motivador: Um Relato de Experiência*”. In: Anais do XXVII Congresso da Sociedade Brasileira de Computação. Belém do Pará: PA.
- Delgado, C.; Xexeo, J. A. M.; Souza, I. F., Campos, M., Rapkiewicz, C. E. (2004) “*Uma abordagem pedagógica para a iniciação ao estudo de algoritmos*”. In: Anais do XXIII Congresso da Sociedade Brasileira de Computação. Salvador: BA.
- Falkembach, G. A. M., Amoretti, M. S. M., Tarouco, L. R., Viero, F. (2003) “*Aprendizagem de Algoritmos: Uso da Estratégia Ascendente de Resolução de Problemas*”. 8º Taller Internacional de Software Educativo. Santiago: Chile.
- França, R. S. de; Silva, W. C. da; Amaral, H. J. C.do. (2012) “*Ensino de Ciência da Computação na Educação Básica: Experiências, Desafios e Possibilidades*”. In: XX Workshop sobre Educação em Computação. Curitiba: Paraná.
- Gomes, T., Melo, J. (2012). *Lógica de Programação com App Inventor: Um relato de experiência no nível médio*. In: Anais da XII Jornada de Ensino, Pesquisa e Extensão da Universidade Federal Rural de Pernambuco. Recife, PE.
- Gomes, T., Melo, J. (2013a). *App Inventor: Android para Não Programadores*. In: Anais do II Simpósio Brasileiro de Tecnologia da Informação. Recife: PE.
- Gomes, T., Melo, J. (2013b). *App Inventor: Android para Não Programadores*. In: Anais do Workshop de Inclusão Digital - XI Escola Regional de Informática Bahia, Alagoas e Sergipe. Aracaju: SE.
- Gomes, T., Melo, J. (2013c). *O Pensamento Computacional no Ensino Médio: Uma Abordagem Blended-Learning*. In: Anais do XXI Workshop de Educação em Computação – XXXIII Congresso da Sociedade Brasileira em Computação. Maceió,

AL

- Lake, P.; Lanerolle, T.; Limardo, N. ; Morelli, R. ; Uche, C. (2011) “Can Android App Inventor Bring Computational Thinking to K-12?” In: SIGCSE - Technical Symposium On Computer Science Education, Dallas – TX – EUA.
- 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
- MIT - Massachusetts Institute of Technology. (2012) “App Inventor for Android”, [appinventor.mit.edu](http://appinventor.mit.edu), Agosto.
- Morelli, R. (2012) “CPSC: Computing with Mobile Phones”, <http://tinyurl.com/rmorelli>, Março.
- Pereira Júnior, J., 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.
- Prensky, M. (2001) “Digital Natives, Digital Immigrants”. <http://www.marcprensky.com/writing/Prensky%20%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf> , MCB University Press, Agosto.
- 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.
- Scaico, P. D., Corlett, E. F., Paiva, L.F., Raposo, E. H. e Alencar, Y. (2012a) “Relato da Utilização de uma Metodologia de Trabalho para o Ensino de Ciência da Computação no Ensino Médio”. In: Anais do XVIII Workshop de Informática na Escola. Rio de Janeiro: RJ.
- Scaico, P.; Lopes, D.; Silva, M. A. de A.; Silva, J. C. da; Neto, S. V. M.; Falcão, E. de S. F. (2012b). “Implementação de um Jogo Sérioso para o Ensino de Programação para Alunos do Ensino Médio Baseado em m-learning”. In: Anais do XX Workshop sobre Educação em Computação. Curitiba: PR.
- Silva, T. S. C. da; Silva, A. S. C. da; Melo, J. C.B. de. (2011) “Ensino de Algoritmos a Nível Médio Utilizando Música e Robótica: Uma Abordagem Lúdica”. In: Anais do XIX Workshop sobre Educação em Computação. Natal: Rio Grande do Norte.
- Wolber, D. (2010a) “AppInventor.org”.<http://www.appinventor.org/course-in-a-box>, Agosto.
- \_\_\_\_\_. (2010b) “*App Inventor* and Real-World Motivation”. <http://cs.usfca.edu/~wolber/appinventor/wolberSigCse11.pdf> , Agosto.