

Introdução ao App Inventor

Prof. Sylvio Silveira Santos

Departamento de Sistemas de Informação – PUC Minas - Barreiro

Disciplina: Tópicos Especiais de Sistemas de Informação.

Programação Orientada a Eventos em Celulares com Sistema Operacional Android

<http://www.appinventor.mit.edu/>

Obs: Estas Notas de Aula estão baseadas no livro de David Wolber, Hal Abelson, Ellen Spertus & Liz Looney, “App Inventor – Create Your Own Android Apps”, editado pela O’Reilly, USA, 2011.

1. INTRODUÇÃO

O App Inventor foi originalmente criado no Google Labs, mas atualmente pertence ao MIT Labs, do Massachusetts Institute of Technology, USA, que vem empregando metodologias orientadas por eventos em diversos aplicativos de natureza educacional. O código do App Inventor é aberto, e qualquer pessoa pode criar um ambiente de desenvolvimento dele derivado.

Ambientes de desenvolvimento como o App Inventor constituem-se em uma modalidade recente para desenvolvimento de aplicativos voltados para sistemas móveis.

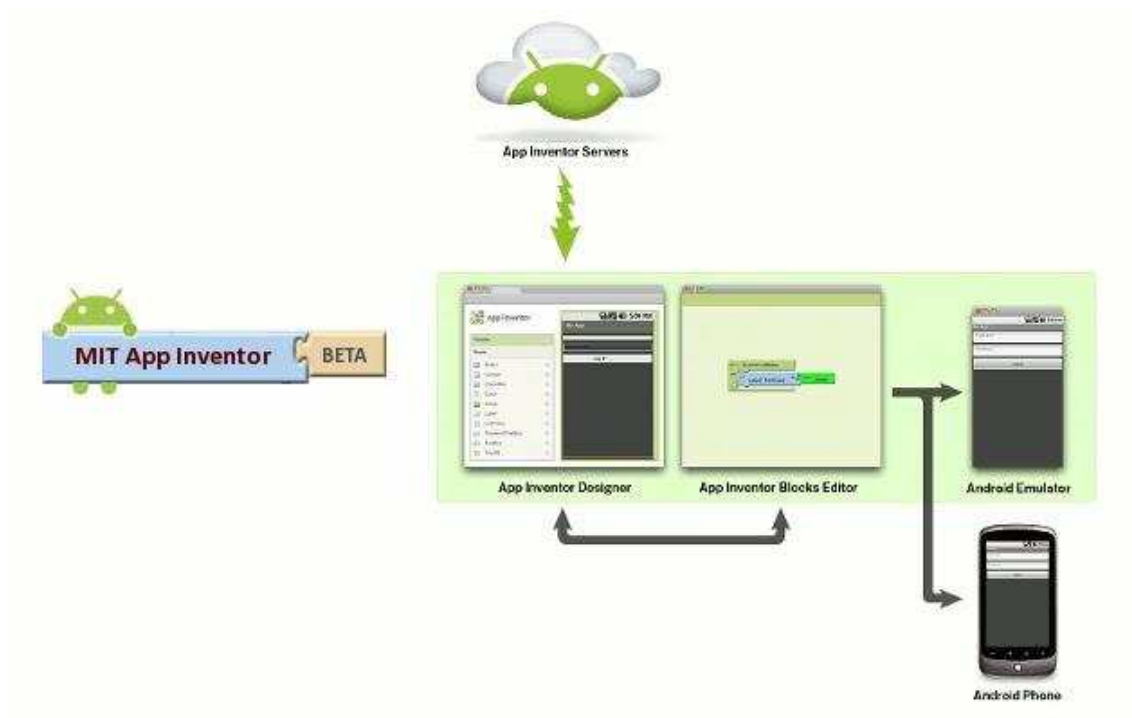


Figura 1 – Estrutura do ambiente de desenvolvimento do MIT App Inventor

Estes ambientes de desenvolvimento, destinados a facilitar a composição de aplicações envolvendo interfaces gráficas, deverão evoluir rapidamente para a criação de aplicativos

profissionais para tablets e celulares, da mesma forma que os ambientes de desenvolvimento profissionais existentes evoluíram nesta direção.

David Wolber, que se dedica a pesquisas práticas nesta área, confirma em seu blog que numerosos desenvolvedores profissionais tem trabalhado para acrescentar ao App Inventor novas rotinas capazes de operarem como blocos de encaixe, destinadas a dotar este ambiente de maior alcance profissional.

Para breve, é esperado que se ele seja integrado ao Google Market com um click do mouse, o que será bastante interessante para esta nova forma de desenvolver aplicativos.

Devemos dizer que o App Inventor está situado no limiar das modernas práticas de ensino da programação para dispositivos móveis e, tratando-se ainda de uma versão beta, deverá evoluir bastante no futuro, sendo que vale apenas investir um pouco de tempo nesta modalidade de programação visual, em particular através de interfaces amigáveis, orientadas para eventos.



Figura 2: Capa do Livro de D. Wolber, editado pela O`Reilly, em que nos baseamos para elaborar este tutorial.

Como outros programas do MIT Labs, Scratch (<http://scratch.mit.edu/>), e também o Star Logo (<http://education.mit.edu/starlogo/>) (destinado ao aprendizado em computação em paralelo e pertencente a outro grupo de pesquisas), assim como o GreenFoot, da Universidade de Kent – USA (<http://www.greenfoot.org/door>), para aprendizado de Java, o App Inventor é ainda um ambiente de desenvolvimento experimental, em versão beta.

No exemplo que se segue, original do livro acima mostrado, “App Inventor – Create Your Own Android Apps”, mostraremos como desenvolver um aplicativo divertido para Android, o qual, em lugar do clássico programa “Alô Mundo” de outros ambientes de programação, fará um gato miar quando o mouse tocar sua figura (emulador do telefone), ou quando o telefone for sacudido, no caso de uso real do celular.

O que será aprendido:

- Construir aplicativos selecionando componentes e então dizer-lhes o que fazer, quando ou como se comportar.

- Usar o *Component Designer* para selecionar componentes. Alguns componentes são visíveis no telefone e outros não o são.
- Adicionar mídia, sons e imagens, por upload de seu computador.
- Trabalhar com o Block Editor para montagem dos blocos lógicos e operacionais do programa, de modo que se tenha um comportamento ditado por suas propriedades.
- Estes componentes podem ser visualizados em tempo real, passo-a-passo, no telefone, na medida em que são construídos.
- Testar o aplicativo na medida que está sendo criado, a qualquer momento, por meio de um *live testing*, para se ter uma visão dinâmica total do que está sendo feito.
- Empacotar o aplicativo de modo a fazer o download para o telefone.

2. O AMBIENTE DO APP INVENTOR

Você poderá começar a usar o App Inventor usando as orientações de instalação fornecidas no site do MIT, em <http://www.appinventor.mit.edu/>

O aplicativo roda através de um navegador, mas alguns ajustes serão necessários, inclusive o download de alguns arquivos para seu desktop e adaptações para rodar o mesmo em seu celular.

O aplicativo App Inventor possui três partes, a saber:

- O *Component Designer*, mostrado na parte esquerda da figura 3, que roda em uma janela de seu navegador. É usado para selecionar componentes para a aplicação e ajustar suas propriedades.
- O *Blocks Editor*, que roda em uma janela separada do Component Designer. Pode ser posicionado ao lado do primeiro, se necessário. É usado para criar comportamento para os componentes, que são os blocos.
- Um telefone celular com sistema operacional Android. Caso não o tenha, poderá usar os emuladores que fazem parte da instalação completa do App Inventor. A figura 3 abaixo mostra todos os componentes do sistema.

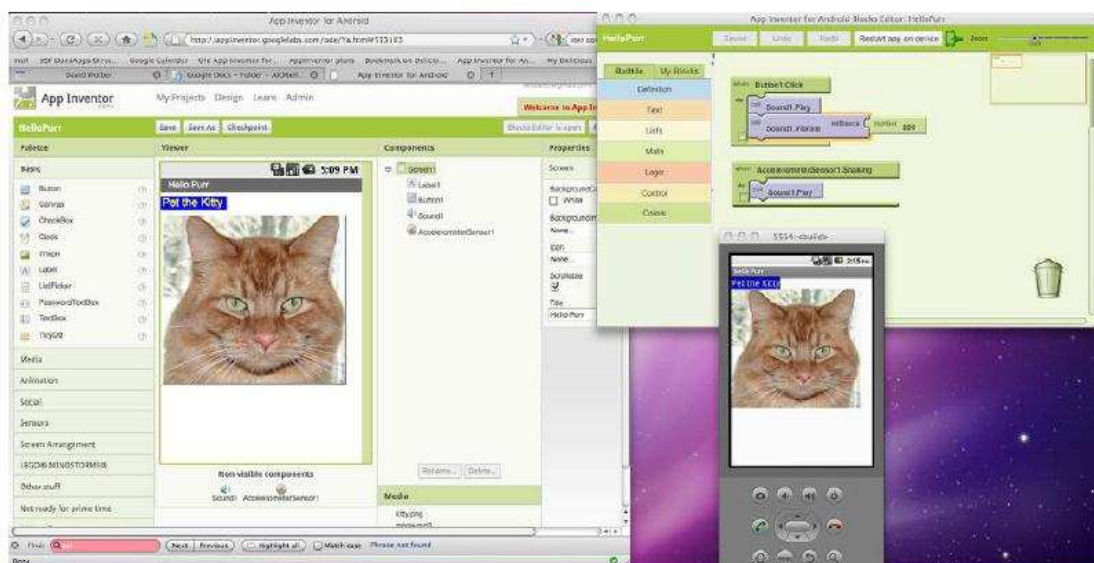


Figura 3 – Acima, o Component Designer, o Bloks Editor e um emulador do telefone com Sistema Operacional Android.

3. CONSTRUÇÃO DO APLICATIVO

Após ter instalado o App Inventor segundo as recomendações do MIT Labs, a primeira janela que se abre é a do Component Designer. Quando ela estiver aberta, clique no botão Blocks Editor, acima na janela. Se a sua instalação estiver correta, após chamar o Java Web Start, o editor de blocos será aberto em uma nova janela. Nele poderemos observar dois botões como abaixo mostrado (Figura 4).



Figura 4 – Você deverá plugar um telefone ao seu computador ou então abrir um emulador. Escolha *New emulator* neste segundo caso e, quando ele estiver operacional, click em *Connect to Device*, para que seu programa possa rodar nele.

Se tudo correr bem, seu ambiente de desenvolvimento deverá estar como na figura 3, porém vazio.

4. PROJETANDO OS COMPONENTES

A primeira coisa a usar é o *Component Designer*. Os componentes são os elementos que você irá combinar para desenvolver uma aplicação. Alguns são simples, como o componente *Label1*, que mostra textos na tela, ou *Button*, que você aperta para iniciar uma ação. Outros componentes são mais elaborados, como a gaveta (canvas) *Drawing*, que pode guardar imagens paradas ou animadas, um acelerômetro, um sensor de movimento capaz de detectar se você movimenta ou balança o telefone, ou componentes que recebem e enviam mensagens, tocam músicas e vídeo, obtém informações de web sites e assim por diante. Quando você abrir o Designer, ele aparecerá como na figura 4 abaixo.

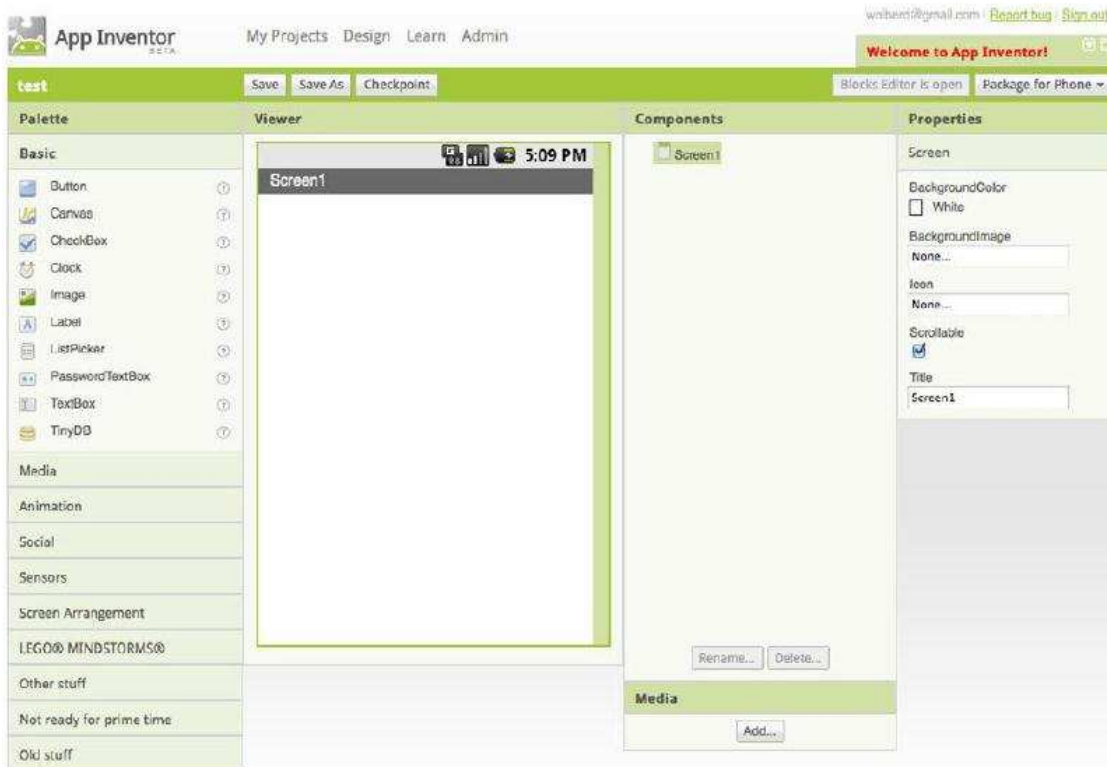


Figura 5 – O Component Designer do App Inventor

O Designer é dividido em várias áreas:

- Bem ao centro está o componente denominado *Viewer*. É o local onde você coloca os componentes e os arranja de modo a formar o aspecto visual do aplicativo. Ele mostra uma imagem aproximada de como ficará a aplicação no final. Por exemplo, um texto poderá ficar até bem comportado neste ambiente, mas poderá ficar quebrado no aplicativo real. Para acompanhar seu trabalho em tempo real, você deverá ter a conexão com seu celular já estabelecida ou então já ter escolhido o emulador.
- À esquerda do *Viewer* está a *Palheta*, que contém uma lista de componentes que você poderá escolher. A palheta é dividida em seções e, no ponto em que está na figura, as gavetas estão fechadas e não mostram seu conteúdo integral. Clicando nas gavetas, poderemos ver estes componentes. Observe que nos pontos de interrogação (?) temos uma ajuda completa sobre os componentes. Clicando por exemplo em **Twitter**, teremos a tela de ajuda:

*A non-visible component that enables communication with [Twitter](#). Methods are included to enabling searching (*SearchTwitter*) or logging into (*Authorize*) Twitter. Once a user has authorized their Twitter account (and the authorization has been confirmed successful by the *IsAuthorized* event), many more operations are available:*

- *Setting the status of the logged-in user (*SetStatus*)*
- *Directing a message to a specific user (*DirectMessage*)*
- *Receiving the most recent messages directed to the logged-in user (*RequestDirectMessages*)*
- *Following a specific user (*Follow*)*
- *Ceasing to follow a specific user (*StopFollowing*)*

- *Getting a list of users following the logged-in user (RequestFollowers)*
- *Getting the most recent messages of users followed by the logged-in user (RequestFriendTimeline)*
- *Getting the most recent mentions of the logged-in user (RequestMentions)*

You must obtain a Consumer Key and Consumer Secret for Twitter authorization specific to your app from http://twitter.com/oauth_clients/new

[More information](#)

- À direita da tela, onde está escrito *Components*, temos a lista destes componentes com suas propriedades. Observe que este ambiente está funcionando como serviço, SaaS, um dos elementos da **Computação em Nuvem**, como já vimos em aulas anteriores. Em meu caso em especial, o ambiente de desenvolvimento está registrado nos servidores do MIT, como se pode ver em <http://beta.appinventor.mit.edu/#1723267>, sendo 1723267 meu número de registro. Experimente transferir o componente Screen1 para a área de desenvolvimento e analise detalhadamente suas propriedades. A metodologia se parece muito com o desenvolvimento de interfaces no Visual Basic.
- Para desenvolver o aplicativo HelloPurr, iremos precisar de dois componentes visíveis, o componente Label1 com o texto “Aperte o Gato” (Pet the Kitt) e um componente Button que deverá conter a imagem de um gato. Nós também iremos precisar de um componente invisível do tipo “Sound”, que se encontra na gaveta “Media” e que saberá como produzir sons, no caso o miado do gato. Teremos também de selecionar um componente denominado “AccelerometerSensor”, situado na gaveta “Sensor”, e que irá vibrar quando o telefone for sacudido.

5 – CONSTRUINDO UM LABEL

O primeiro componente incorporado na palheta é *Label1*, que ocupa a 5ª posição dos componentes básicos (Gaveta *Basic*). Mude a cor de fundo do elemento clicando na propriedade *BackgroundColor*, selecionando *Blue*, azul, e mude a cor do texto para *Yellow*, amarelo. Aumente o tamanho da fonte para 20. Veja o resultado no emulador. O resultado destas alterações está mostrado na figura 6 abaixo:

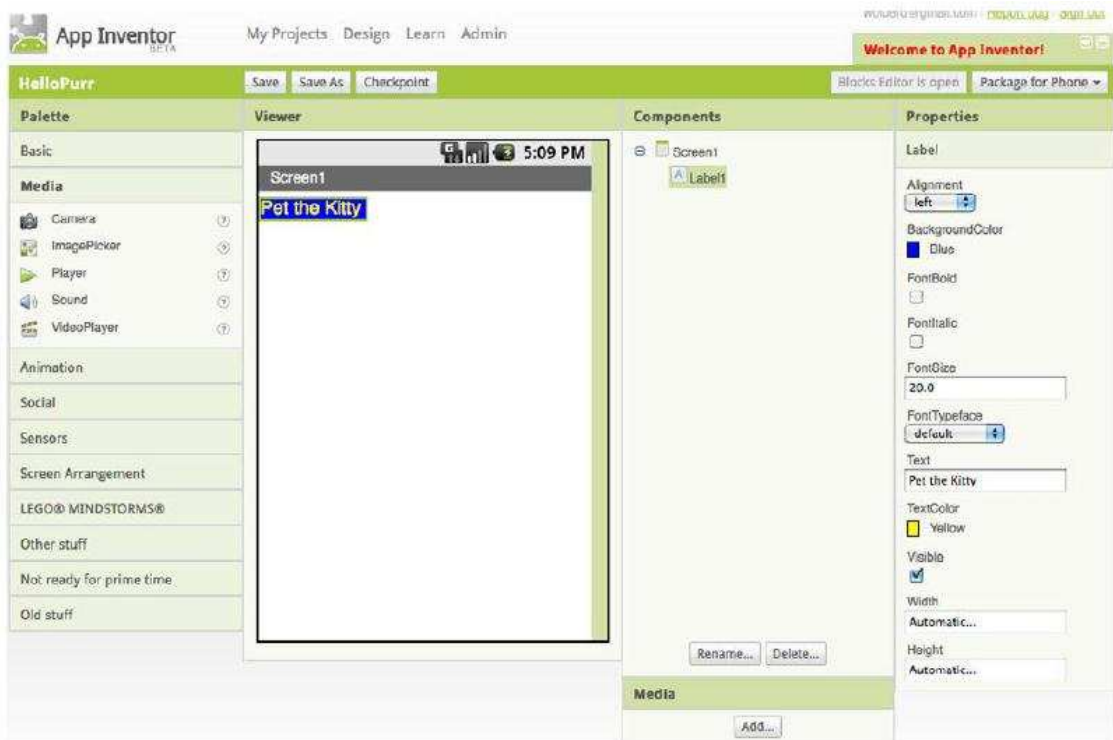


Figura 6 – O App Inventor mostrando o componente *Label*, após ajustes.

As operações realizadas até agora retratam o estado do sistema.

6 – CONSTRUINDO UM BOTÃO

A figura do gato é construída por meio de um botão normal, que em seguida é alterado para conter a imagem do gato. Para isto, vamos obter primeiro o botão, indo para a gaveta de componentes básicos e arrastando *Button* para a palheta. É este botão que usaremos para colocar nele a imagem do gato e fazer com que ela reaja ao ser clicada, fazendo “miau...”.

Para isto, baixe a imagem do gato indo ao site do autor do livro, David Wolber, em <http://examples.oreilly.com/0636920016632/>

Nos diretórios que surgem, baixe apenas o arquivo zipado, que contém todos os programas do livro. A pasta *CH01_HelloPurr* é a que nos interessa no momento, pois contém a imagem do gato, *kitty.png*, e o miado do mesmo, *meow.mp3*.

Vá ao aplicativo e, nas propriedades do botão instalado, clique em *Image* e procure por *kitty.png*. Faça o upload desta imagem. Ela deverá aparecer no botão conforme mostrado na figura 7 abaixo. Você poderá observar que, sobre a imagem, ainda está escrito “Pet the kitty”, texto que pode ser alterado ou deletado, como quiser.

7 – ADICIONANDO O SOM “MEOW”

Vamos adicionar o pequeno arquivo em mp3 com o som do miado do gato, que deverá ser ouvido quando clicarmos na imagem do mesmo. O procedimento é semelhante ao

anterior, mas deveremos ir para a gaveta *Media* da tela principal do App Inventor. Arraste o componente de som e coloque-o no *Viewer*, a tela principal.



Figura 7 – O aplicativo com um Label e um Botão, com a imagem escolhida.

Onde quer que você coloque este componente, ele ficará no local destinado aos componentes invisíveis, ao pé da tela do *Viewer*.

Componentes invisíveis como este participam do programa, mas não aparecem quando o mesmo é executado.

Clique em *Sound1* para ver as propriedades do componente. Ajuste o item *Source* para *meow.mp3*.

Agora, tanto *kitty.png* como *meow.mp3* deverão aparecer na seção *Media* do *Designer*.

A Tabela 1 nos mostra os componentes que foram usados até agora nesta aplicação:

Component type	Palette group	Name of component	Purpose
Button	Basic	Button1	Press to make the kitty meow.
Label	Basic	Label1	Shows the text "Pet the Kitty."
Sound	Media	Sound1	Play the meow sound.

Tabela 1 – Os componentes usados até agora no aplicativo HelloPurr.

8 – ADICIONANDO COMPORTAMENTO AOS COMPONENTES

Abra o Editor de Blocos na parte direita do menu superior. Examine o Editor de Blocos. É ele que dá vida ao aplicativo. Nós iremos dizer ao componente *Kitty Button* para emitir um som quando clicarmos nele.

9 – FAZENDO O GATO MIAR

Na parte esquerda da janela vemos os botões "Built-in" e "My Blocks". Clique em "My Blocks" e verá uma gaveta que se abre mostrando todos os seus componentes que foram criados no *Designer*: *Button1*, *Label1*, *Screen1* e *Sound1*.

Portanto, clique na gaveta para *Button1*. Seus componentes se abrem, mostrando uma seleção de blocos que você poderá usar para dizer ao botão o que ele deverá fazer, iniciando com **Button1.Click** ao alto, como mostrado na figura 8:



Figura 8 – Clicando em *Button1* podemos abrir todas as suas propriedades.

Clique no botão **Button1.Click**, arrastando-o para o espaço de trabalho. Observe atentamente este botão e verá que nele está escrito “**When Button1.Click do**”, em que a palavra **when** está em letra de tamanho menor que Button1.Click.

Blocos nos quais encontra-se escrita a palavra **when** são denominados “**Event Handlers**”, e especificam o que o evento deverá fazer quando um determinado evento acontece.

Em nosso caso, o evento desejado está relacionado à figura do gato, que na verdade é um botão. Desejamos que esta ação provoque um som, o miado do gato. A figura abaixo ilustra o processo:

```
When Button1.Click  
do  
    call.Sound1.Play
```

Podemos notar que o método *Click* atua sobre o objeto *Button1*, e o método *Play* sobre o objeto *Sound*. Teste sua aplicação em tempo real e verá o resultado da animação referente a este pequeno trecho de código.

Como complemento a esta altura, poderemos fazer o telefone vibrar e, ao mesmo tempo, o gato miar. Este fato está indicado na figura 9 abaixo.



Figura 8 – Acima, quando clicarmos na figura do gato no emulador, um som será ouvido.



Figura 9 – A figura 9 mostrando as duas chamadas: Sound1.Play e Sound1.Vibrate

A vibração do telefone é conseguida expandindo a área Sensors da palheta de componentes. Poderemos arrastar então o *AccelerometerSensor* para a o ambiente de composição, conforme indicado na figura 10, o que fará emissão do som quando o telefone for sacudido.

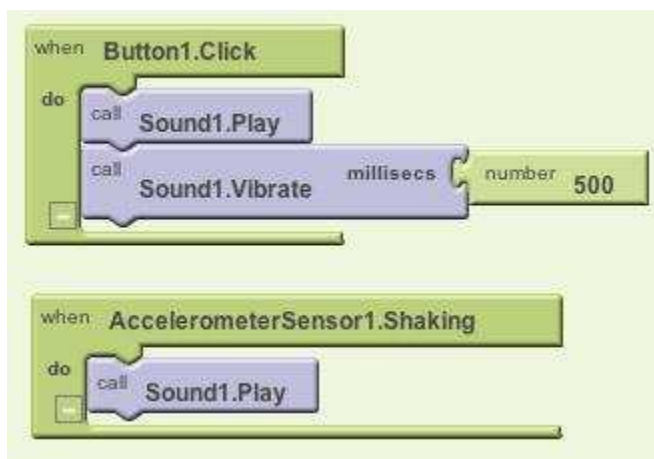


Figura 10: O acelerômetro, invisível no ambiente de composição, completa o projeto.

10 - EMPACOTANDO O PROJETO

O App Inventor é uma ferramenta de computação em nuvem, significando que seu aplicativo fica hospedado online em um servidor do MIT Labs à medida que você vai trabalhando. Portanto, se você fechar o App Inventor, o aplicativo estará lá quando você retornar, não sendo necessário salvar coisa alguma em sua máquina. Isto permite também testar seu programa no seu telefone (teste ao vivo). Mas se você desconectar seu telefone do App Inventor, o aplicativo irá parar, pois ele não foi instalado realmente.

Você poderá no entanto rodar o aplicativo em qualquer telefone, mesmo desconectado do App Inventor. Neste caso basta configurar seu telefone fora do Android Market.

Vá para *Settings* → *Applications*, marcando a caixa junto a *Unknown Sources* (encontre estes termos correspondentes em Português, se for o caso).

Vá então ao ambiente de desenvolvimento (*Designer*) e clique no botão superior, “*Package for Phone*”, e selecione “*Download to Connected Phone.*” Você verá as palavras “*Saving*” e “*Packaging*”, um processo que demora cerca de um minuto. Para compartilhar seu programa com alguém, clique em “*Package for Phone*” e selecione “*Download to this Computer*”. Isto criará um arquivo com a extensão *.apk* em seu computador, e você poderá fazer um upload dele de modo que fique acessível na Web.

Após isto, seu programa estará pronto para rodar no seu celular.

Referência:

David Wolber, Hal Abelson, Ellen Spertus & Liz Looney, “App Inventor – Create Your Own Android Apps”, O’Reilly Ed., USA, 2011.