

MIT APP INVENTOR

O que é?

O MIT App Inventor é uma ferramenta de programação baseada em blocos que permite que qualquer um, mesmo iniciantes, comecem a programar e construir aplicativos totalmente funcionais para dispositivos Android. Os recém chegados ao App Inventor podem ter seus primeiros aplicativos criados e rodando em uma hora ou menos, e podem programar aplicações mais complexas em um tempo significativamente menor do que com linguagens de programação mais tradicionais, baseadas em código.

Inicialmente desenvolvido pelo professor Hal Abelson e uma equipe do *Google Education*. App Inventor funciona como um Web Service administrado por membros do *MIT's Center for Mobile Learning*. MIT App Inventor possui uma comunidade mundial de quase 3 milhões de usuários, representando 195 países em todo o mundo. Mais de 100 mil usuário ativos semanalmente e que já construíram mais de 7 milhões de aplicativos para Android.

O App Inventor é uma ferramenta de código aberto que visa tornar a programação e criação de aplicativos acessíveis para uma grande variedade de públicos, como:

Educadores formais e informais que usaram MIT App Inventor para apresentar a programação aos seus alunos de Computação, os membros do clube de ciências, programas pós-escolares, oficinas, cursos, etc. Muitos educadores também começaram a usar o MIT App Inventor para desenvolver aplicações em apoio dos seus próprios objetivos instrucionais.

Governo e funcionários civis e voluntários que têm aproveitado o poder do MIT App Inventor para desenvolver aplicativos personalizados, muitas vezes na resposta a desastres naturais e necessidades comunitárias.

Os pesquisadores que utilizam MIT App Inventor para criar aplicativos personalizados para atender às suas necessidades de coleta e análise de dados em suporte de suas pesquisas em uma ampla variedade de campos desde médicos a social.

Entusiastas e empreendedores que têm uma idéia que eles querem transformar rapidamente em um aplicativo sem a curva de custo ou de aprendizagem que a criação de aplicativos mais tradicional implica.

Uma linguagem de blocos para smartphones

App Inventor é uma ferramenta de drag-and-drop (arrastar e soltar) visual para a construção de aplicações móveis na plataforma Android. Você pode projetar a interface do usuário (a aparência visual) de um aplicativo usando um construtor de interface gráfica de usuário baseada na web (GUI), assim como você também pode especificar o comportamento do app reunindo "blocos", como se você estivesse trabalhando em um quebra-cabeça.

O que você pode fazer com o App Inventor?

Play

Criar aplicativos para o seu telefone é divertido, e o App Inventor promove a exploração e descoberta. Basta abrir o App Inventor em um navegador web, conectar seu telefone, e começar a montar blocos como os da figura 1.2. Você pode ver imediatamente e interagir com o aplicativo que você está construindo no telefone.

Protótipos

Tem uma idéia para um aplicativo? Em vez de escrevê-la em um guardanapo ou deixá-lo de lado, construa um protótipo rápido. Protótipos são incompletos e não versões completas de suas idéias. Desta forma, o App Inventor pode servir como um guardanapo eletrônico para desenvolvimento de aplicativos móveis.

Construa aplicativos com utilidade pessoal

No estado atual do mundo de aplicativos móveis, nós estamos presos com os aplicativos que nos são dados. Quem não se queixou de um app e desejou que ele pudesse ser personalizado ou ajustado de alguma forma? Com o App Inventor, você pode construir um app exatamente como você quiser.

Desenvolver aplicativos completos

O App Inventor não é apenas um sistema de prototipagem ou uma interface desenhada para você poder construir aplicativos completos, aplicativos de uso geral. A linguagem fornece todos os principais fundamentos de programação na construção de blocos para sua aplicação, como loops e condicionais, mas em forma de blocos.

Ensinar e aprender

Se você está no ensino fundamental, ensino médio, ou universidade, o App Inventor é uma grande ferramenta de ensino e aprendizagem. É ótimo para a ciência da computação, mas também é um ótimo ferramenta para a matemática, física, empreendedorismo, e qualquer outra disciplina. O fundamental é que você aprende através da criação. Em vez de memorizar fórmulas, você constrói um aplicativo para, por exemplo, encontrar o hospital mais próximo (ou shopping). Achemos que o App Inventor pode ser uma grande ferramenta nas aulas de qualquer disciplina e em qualquer momento de sua vida.

Porque o App Inventor é a escolha certa

A maioria das pessoas diz que o App Inventor é fácil de usar por causa de seu visual, interface drag-and-drop. Mas o que isso significa? Por que é que o App Inventor é tão fácil de usar?

Você não precisa se lembrar de sentenças e escrever instruções

Uma das maiores fontes de frustração para programadores iniciantes vem de ir digitando os código e ter o computador lançando de volta mensagens de erro indecifráveis. Essa frustração desencoraja muitos iniciantes de programação antes mesmo chegar ao mais divertido, resolução de problemas lógicos.

Você escolhe a partir de um conjunto de opções

Com App Inventor, os componentes e os blocos são organizados em partes que estão prontamente disponíveis para você. Você programa encontrando um bloco que ajuda a especificar a funcionalidade que você quer construir e arrastando-o para o programa. Você não tem que se lembrar o que as instruções são ou ao que se referem.

Apenas alguns blocos ligam um ao outro

Em vez de castigar os programadores com mensagens de erro enigmáticas, a linguagem de blocos do App Inventor restringe você de fazer muitos erros. Por exemplo, se um bloco de função espera um número, você não pode ligar um texto. Isso não elimina todos os erros, mas com certeza ajuda.

Você lida diretamente com eventos

Linguagens de programação tradicionais foram projetados quando a programação era como trabalhar com receitas, ou conjuntos de instruções. Mas, com interfaces gráficas, e especialmente com aplicativos móveis onde os

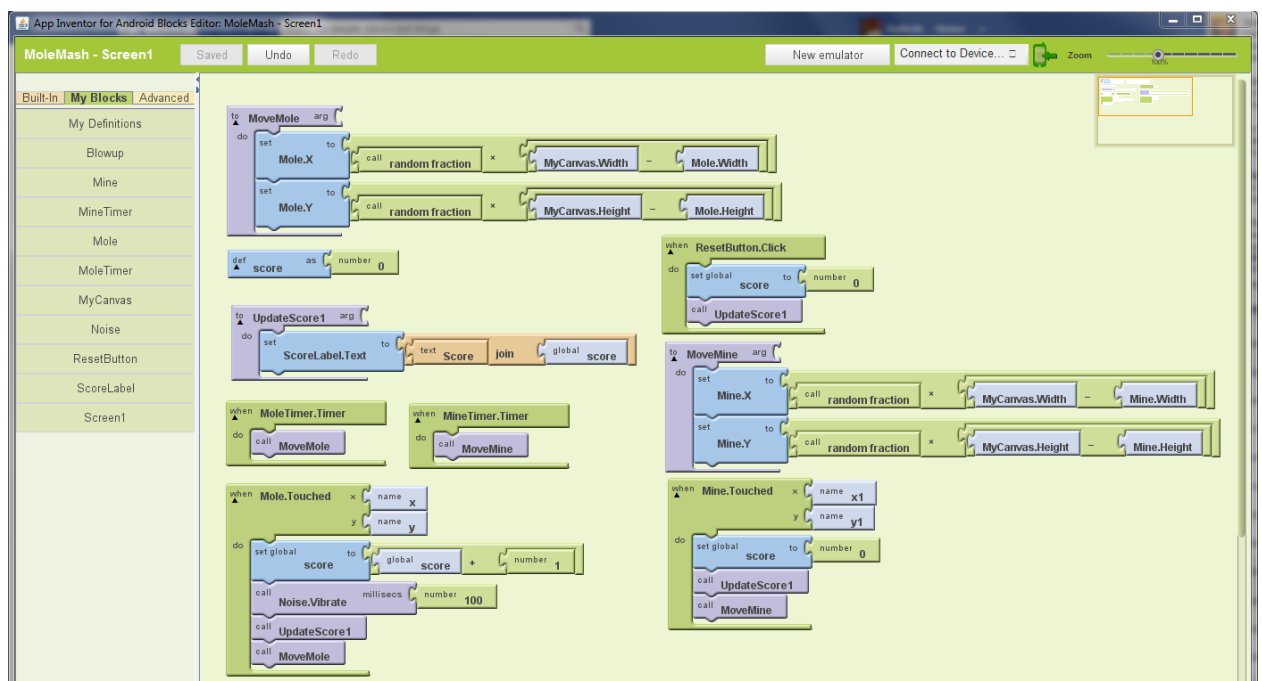
eventos podem acontecer a qualquer momento (por exemplo, receber uma mensagem de texto ou chamada telefônica), a maioria dos programas não são receitas, mas são em vez, conjuntos de manipuladores de eventos. Um manipulador de eventos é uma maneira de dizer: "Quando isso acontece, o app faz isso." Em uma linguagem tradicional, como Java, você tem que entender classes, objetos, e objetos especiais chamados "ouvintes", para expressar uma simples evento. Com App Inventor, você pode dizer: "Quando um usuário clicar neste botão..." ou "Quando um texto é recebido..." arrastando um "When" bloco.

Que tipos de aplicativos eu posso construir?

Você pode construir muitos tipos diferentes de aplicativos com o App Inventor. Use sua imaginação, e você poderá criar todos os tipos de aplicativos úteis.

Jogos

As pessoas muitas vezes começam com a construção de jogos como *Mole Mash* (Figura 1.3) ou aplicativos que deixam tirar fotos engraçadas. Conforme você avança, você pode construir suas próprias versões de jogos



mais complexos, como *Pac-Man* e *Space Invaders*.

Figura 1.3: Parte dos blocos para criação do jogo Mole Mash

Aplicativos Educativos

App Inventor não está limitada a jogos simples. Você também pode criar aplicativos que informam e educam. Você pode criar um app de quizz para ajudar você e seus colegas a estudar para uma prova, ou mesmo um aplicativo que permita que os usuários do seu aplicativo criar seus próprios quizzes.

Aplicativos de Localização

Por o App Inventor fornecer acesso a um sensor de GPS, você pode construir apps para saber onde você está. Você pode criar um aplicativo para ajudar a lembrar onde você estacionou seu carro, um aplicativo que mostra a localização dos seus amigos ou colegas em um evento ou conferência, ou o seu próprio app de tour personalizado da sua escola, universidade, local de trabalho, ou um museu.

Aplicações de alta tecnologia

Você pode criar aplicativos para escanear códigos de barra, falar, ouvir (reconhecer palavras), ouvir música, fazer música, reproduzir vídeo, detectar a orientação do telefone e de aceleração, tirar fotos, e fazer chamadas telefônicas. Smartphones são como canivetes para a tecnologia, e um grupo de engenheiros do Google dedicou-se a fazendo que a tecnologia seja fácil de controlar através App Inventor.

Aplicativos que controlam robôs

Você pode usar o telefone como um controle remoto para controlar robôs. O robô e o telefone podem se comunicar via Bluetooth, e os componentes de Bluetooth do App Inventor permitem que você crie aplicativos para isso.

Aplicativos Complexos

App Inventor reduz drasticamente a barreira de entrada para a programação e permite você construir aplicativos de alta tecnologia em questão de horas. A linguagem de blocos também fornece loops, condicionais, e outras lógicas de programação necessário para construir aplicativos com lógica complexa. Você vai se surpreender com o quão divertido tais problemas de lógica podem ser quando você está tentando construir um aplicativo.

Aplicativos habilitados para a Web

O App Inventor também fornece uma maneira para que seus aplicativos possam se comunicar com a Web. Você pode escrever aplicativos que puxam dados do Twitter ou um feed RSS, ou um Navegador de Livros da Amazon que permite verificar o custo online de um livro, digitalizando o seu código de barra.

Quem pode construir apps?

App Inventor está disponível gratuitamente para qualquer um usar. É possível usá-lo online (em vez de diretamente no seu computador) e é acessível a partir de qualquer browser. Você não precisa sequer de um telefone para usá-lo: você pode testar seus aplicativos em um emulador Android incluído. A partir de janeiro de 2011, havia dezenas de milhares de usuários App Inventor ativos e centenas de milhares de Apps.

App Inventor e sua Linguagem de Blocos

Como já foi dito antes, o App Inventor é uma ferramenta de construção de aplicativos Android através da junção de conjuntos de blocos que determinam como sua aplicação deve se comportar. Você define a lógica de seu aplicativo arrastando blocos específicos. Cada componente em seu projeto tem seu próprio conjunto de blocos específicos para seus próprios eventos, métodos e propriedades. Abaixo (figura 1.7) temos uma visão geral de todos os blocos de construção do Editor de Blocos disponíveis na ferramenta.



Figura 1.7: Visão geral dos blocos de construção

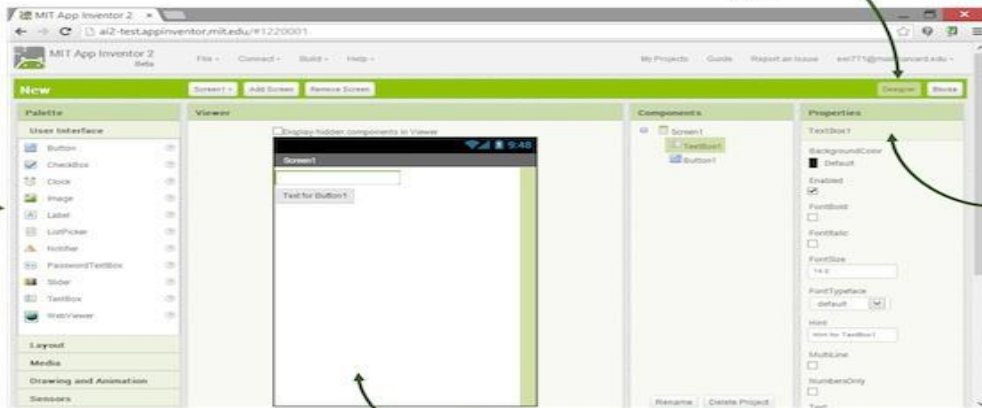
Designer e Editor de Blocos

O App Inventor consiste no Designer e Editor de Blocos. Estes são descritos em detalhe abaixo.

App Inventor Designer

Palette: Find your components and drag them to the Viewer to add them to your app.

Designer Button: Click from any tab to go to the Designer tab.



Properties: Select a Component in the Components List to change its properties (color, size, behavior) here.

Viewer: Drag components from the Palette to the Viewer to see what your app will look like.

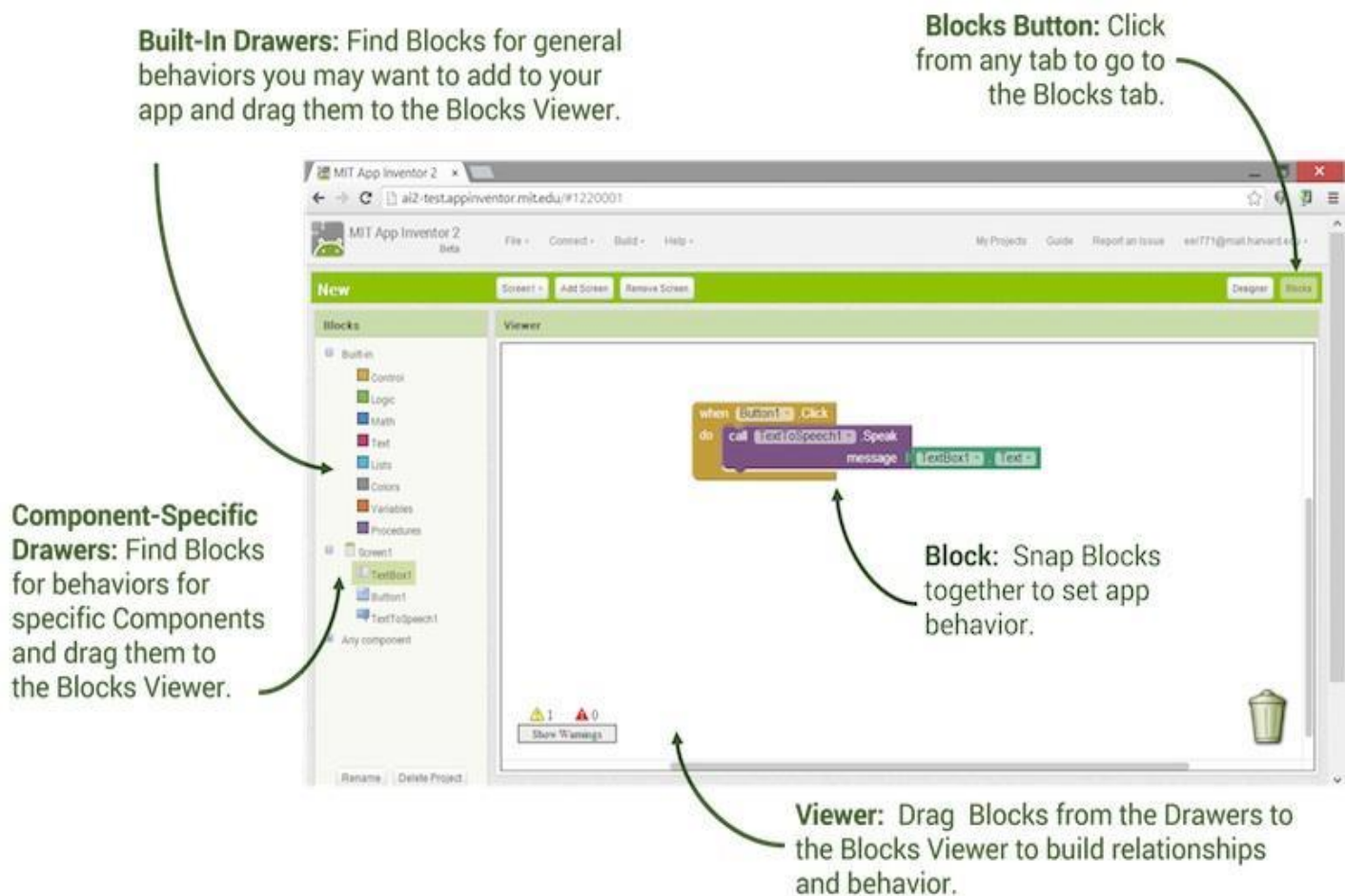
Figura 1.8: Interface de criação do design de aplicativos.

Pallete (Paleta): Encontre seu componentes e arraste-os para a Viewer para adicionar ao seu aplicativo.

Viewer (Espectador): Arraste componente da Paleta para o Espectador para formar seu app e ver como ele vai ficar.

Designer Button (Botão Designer): Caso você esteja em outra tela, clique em Designer Button para voltar à tela de design.

Properties (Propriedades): Selecione um componente na Lista de Componentes para alterar suas propriedades como, cor, tamanho, comportamento, etc.



App Inventor Blocks Editor (Editor de Blocos)

Figura 1.9: Interface para programar o comportamento do aplicativo, juntando blocos.

Built-in Drawers (Blocos Embutidos): Procure por blocos com comportamentos que você queira adicionar ao seu app e arraste-os para a Viewer.

Component-Specific Drawers (Componentes Específicos): Encontre blocos para comportamentos específicos e arraste-os para a Viewer.

Viewer (Espectador): Arraste blocos do para a Blocks Viewer para construir relações e comportamentos para seu aplicativo.

Blocks Button (Botão Blocos): Caso você esteja em outra tela, clique em Blocks Button para voltar à tela de construção de blocos.

Componentes – App Inventor para Android

A seguir descreveremos os componentes que você pode usar em App Inventor para construir seus aplicativos.

Cada componente pode ter métodos, eventos e propriedades. A maioria das propriedades pode ser alterada por aplicações - estas propriedades têm blocos que você pode usar para obter e definir os valores. Algumas propriedades não podem ser alteradas por aplicativos - estas só têm blocos que você pode usar para obter os valores, não defini-las. Algumas propriedades só estão disponíveis no Designer.

Dos componentes mostrados a seguir, muitas propriedades são comuns, então as veremos separadamente. Propriedades específicas serão comentadas nas áreas correspondentes de seu componente. As propriedades comuns são:

BackgroundColor

Retorna a cor de fundo do componente.

Enabled

Se definido, o usuário poderá fazer alguma ação com componente (clique, por exemplo).

FontBold

Se definido, o texto do componente é exibido em negrito.

FontItalic

Se definido, o texto do componente é exibido em itálico.

FontSize

Tamanho da fonte do texto do componente.

FontTypeface (designer apenas)

Tipo da fonte para o texto componente.

Height

Comprimento vertical do componente (-size y).

Image

Imagem a ser exibida no componente.

Shape (designer apenas)

Especifica a forma do componente (default, arredondado, retangular, oval). A forma não será visível se uma imagem está a ser exibida.

ShowFeedback

Especifica se um feedback visual deve ser mostrado quando o componente for clicado, como uma imagem no fundo.

Text

Texto a ser exibido no componente.

TextAlignment (designer apenas)

Orientação do texto do componente. Esquerda, centro ou direita.

TextColor

Cor de texto do componente.

Visible

Especifica se o componente deve estar visível na tela. O valor é true se o componente está visível e falso se estiver escondido.

Width

Comprimento horizontal do componente (-size x).

Hint

O texto que deve aparecer levemente em componentes que recebem texto do usuário, para fornecer uma dica sobre o que o usuário deve digitar.

➤ User Interface Components – Componentes da interface do usuário.

- **Button – Botão**

Botão com a capacidade de detectar cliques. Muitos aspectos da sua aparência podem ser mudados, assim como se é clicável (Enabled).

Eventos:

Click()

Usuário clicou e soltou o botão.

GotFocus()

Indica que o cursor move sobre o botão de modo que agora é possível clicar nele.

LongClick()

Usuário clicou e segurou o clique por um tempo.

LostFocus()

Indica que o cursor afastou-se do botão de modo que já não é possível clicar nele.

TouchDown()

Indica que o botão foi pressionado.

TouchUp()

Indica que um botão foi liberado.

• **CheckBox - Caixa de seleção**

Componentes CheckBox podem mudar seu estado de resposta quando o usuário o seleciona.

Um componente CheckBox gera um evento quando for selecionado. Existem muitas propriedades que afetam a sua aparência que podem ser definidas no Editor de Designer ou blocos.

▪ **Propriedades:**

Checked

True (verdadeiro) se a caixa está marcada, False (falso) caso contrário.

▪ **Eventos:**

Click()

Usuário clicou na caixa de seleção.

GotFocus()

Indica que o cursor move sobre a caixa de seleção de modo que agora é possível clicar nela.

LostFocus()

Indica que o cursor afastou-se do botão de modo que já não é possível clicar nele.

- **DatePicker – Seleccionador de data**

Um botão que, quando clicado, inicia um diálogo pop-up para permitir que o usuário selecione uma data.

- **Propriedades:**

- Day*

O dia do mês pra ser usado no DatePicker.

- Month*

O número do mês para ser usado no DatePicker.

- MonthInText*

Retorna o nome do mês em que foi passado usando o DatePicker, em formato textual.

- Year*

O ano para ser usado no DatePicker.

- **Eventos:**

- AfterDateSet()*

Evento que é executado depois que o usuário escolhe uma data no diálogo.

- GotFocus()*

Indica que o cursor move sobre o botão de modo que agora é possível clicar nele.

- LostFocus()*

Indica que o cursor afastou-se do botão de modo que já não é possível clicar nele.

- TouchDown()*

Indica que o botão foi pressionado.

- TouchUp()*

Indica que um botão foi liberado.

- **Métodos:**

- LaunchPicker()

Inicia o pop-up DatePicker.

- SetDateToDisplay (número do ano, número do mês, número do dia)

Permite ao usuário definir a data a ser exibida quando o DatePicker é exibido. Os valores válidos para o campo do mês são 1-12 e 1-31 para o campo do dia.

- **Image**

Componente para exibir imagens. A imagem a ser exibida, e outros aspectos da aparência da imagem, podem ser especificados no Designer ou no Editor de Blocks.

- **Propriedades:**

- Animation

Esta é uma forma limitada de animação que pode anexar um pequeno número de tipos de movimento para imagens. Os movimentos permitidos são ScrollRightSlow, ScrollRight, ScrollRightFast, ScrollLeftSlow, ScrollLeft, ScrollLeftFast, e Stop.

- Picture

Imagem a ser exibida.

- **Label - Etiqueta**

As Labels são componentes utilizados para mostrar texto. Suas propriedades podem ser definidas no editor de designer ou de blocos.

- **ListPicker – Recebedor de listas**

Um botão que, quando clicado, exibe uma lista de textos para que o usuário escolha entre eles. Os textos podem ser especificados através do Designer ou Editor de Blocks.

- **Propriedades:**

- Elements

Elementos da lista.

Obs: Apenas no editor de blocos.

ElementsFromString

Organização do elemento na lista.

Selection

O item selecionado. Quando alterado diretamente pelo designer, a propriedade SelectionIndex também é alterada para o primeiro item no ListPicker com o valor dado. Se o valor não for exibido, SelectionIndex será definido como 0.

SelectionIndex

O índice do item selecionado no momento, a partir de 1. Se nenhum item for selecionado, o valor será 0. Se for feita uma tentativa de defini-lo para um número menor do que 1 ou maior que o número de itens na ListPicker, SelectionIndex será definido para 0, e Selection será definido como texto vazio.

ShowFilterBar

Se definido, um Filtro de pesquisa será exibido na ListPicker.

Title

Título opcional exibido no topo da lista de opções.

ItemTextColor

A cor do texto dos itens do ListPicker.

ItemBackgroundColor

A cor de fundo dos itens do ListPicker.

▪ **Eventos:**

AfterPicking()

Evento a ser gerado após o ListPicker ser fechado.

BeforePicking()

Evento para ser gerado quando o botão do componente é clicado ou a lista é mostrada usando o bloco Abrir. Este evento ocorre antes da lista de itens a ser apresentada, e pode ser utilizado para preparar a lista antes de ser mostrada.

GotFocus()

Indica que o cursor move sobre o botão ListPicker de modo que agora é possível clicar nele.

LostFocus()

Indica o cursor afastou-se do botão ListPicker de modo que já não é possível clicar nele.

▪ **Métodos:**

Open()

Abre o ListPicker, assim como se o usuário clicar sobre ele.

• **ListView – Lista de visualização**

Este é um componente visível que permite colocar uma lista de elementos de texto em sua tela para exibir. Enquanto o ListPicker inicia em uma tela própria, o ListView tem seus elementos visualizados na tela em que é colocado.

Aviso: Este componente não irá funcionar corretamente nas telas que são de rolagem.

▪ **Propriedades:**

Elements

Elementos de texto que constituem a lista.

ElementsFromString

Organização do elemento na lista.

Selection

Retorna o elemento selecionado no último ListView.

SelectionIndex

O índice do item selecionado no momento, a partir de 1. Se nenhum item for selecionado, o valor será 0. Se for feita uma tentativa de defini-lo para um número menor do que 1 ou maior que o número de itens na ListView, SelectionIndex será definido para 0, e a seleção será definido como texto vazio.

ShowFilterBar

Se definido, um Filtro de pesquisa será exibido no ListView. True mostrará o filtro, False vai escondê-lo.

▪ **Eventos:**

AfterPicking()

Evento ser gerado após um elemento ser escolhido na lista.

- **Notifier - Notificador**

O componente Notifier exibe alertas diálogos, mensagens e alertas temporários. Não visível.

- **Propriedades:**

NotifierLength (designer apenas)

Especifica a duração de tempo em que o alerta é mostrado - ou "curta" ou "longa".

- **Métodos:**

DismissProgressDialog()

Descartar uma caixa ProgressDialog exibida anteriormente.

LogError (recebe um texto para a mensagem)

Exibe uma mensagem de erro no log do sistema Android.

LogInfo (recebe um texto para a mensagem)

Exibe uma mensagem informativa no log do sistema Android.

LogWarning (recebe um texto para a mensagem)

Exibe uma mensagem de aviso no log do sistema Android.

ShowAlert (recebe um texto para a notificação)

Exibir uma notificação temporária.

ShowChooseDialog (recebe texto para a mensagem, o título, o button1, o button2, e um valor lógico (true ou false) para cancelable)

Mostra uma caixa de diálogo com dois botões, a partir do qual o usuário pode escolher. Se cancelable receber "true", haverá um botão adicional CANCEL. A "escolha" do parâmetro para AfterChoosing será o texto do botão que foi pressionado, ou "Cancel" se o botão CANCEL foi pressionado.

ShowMessageDialog (recebe um texto para a mensagem, para o título, e para o buttonText)

Exibir um diálogo de alerta com um único botão que dispensa o alerta.

ShowProgressDialog(recebe um texto para a mensagem, para o título)

Mostra uma caixa de diálogo com um título e mensagem opcional (usar cadeias vazias, se eles não são desejados). Ele não pode ser cancelado pelo usuário, mas deve ser cancelado pelo Programa App Inventor usando o bloco DismissProgressDialog.

ShowTextDialog (recebe um texto para a mensagem, para o título e um valor lógico (true ou false) para cancelable)

Mostra uma caixa de diálogo onde o usuário pode digitar o texto, após o qual o evento AfterTextInput será gerado. Se cancelable receber "true", haverá um botão adicional CANCELAR. A "resposta" para AfterTextInput será o texto que foi digitado, ou "Cancelar" se o botão CANCELAR foi pressionado.

- **Eventos:**

AfterChoosing (retorna "choice" (escolha))

Evento que inicia após o usuário ter feito uma seleção para ShowChooseDialog.

AfterTextInput(retorna "response" (resposta))

Evento que inicia após o usuário responder a ShowTextDialog.

- **TextBox - Caixa de texto**

Os usuários digitam o texto em um componente da caixa de texto.

- **Propriedades:**

MultiLine

Se definido, então esta caixa de texto aceita várias linhas.

NumbersOnly

Se definido, então esta caixa de texto aceita somente números.

- **Eventos:**

GotFocus()

Evento gerado quando esse componente está em foco, tal como quando é selecionado pelo usuário.

LostFocus()

Evento gerado quando este componente já não está em foco, como quando o usuário toca em uma caixa de texto diferente.

- **Métodos:**

HideKeyboard()

Ocultar o teclado. Somente quando a propriedade MultiLine está definida. TextBox de linha única fecha o teclado quando o usuário pressiona a tecla Concluído.

RequestFocus()

Define a caixa de texto como ativa.

- **PasswordTextBox – Caixa de texto para senhas**

Os usuários digitam senhas em um componente da caixa de texto de senha, que pode esconder o texto que foi digitado na mesma.

- **Métodos:**

RequestFocus()

Define o PasswordTextBox ativa.

- **Eventos:**

GotFocus()

PasswordTextBox tornou-se o componente focalizado.

LostFocus()

PasswordTextBox não é mais o componente focalizado.

- **Screen - Tela**

Componente de nível superior que contém todos os outros componentes do programa.

- **Propriedades:**

AboutScreen

Informações sobre a tela. Aparece quando "Sobre este aplicativo" é selecionado a partir do menu do sistema. Use-o para informar aos usuários sobre o seu app. Em vários aplicativos de tela, cada tela tem sua própria informação AboutScreen.

AlignHorizontal

Alinhamento horizontalmente dos componentes na tela. As opções são: esquerda, centro e direita.

AlignVertical

Alinhamento vertical dos componentes na tela. As opções são: parte superior, centro e parte inferior.

AppName (designer apenas)

Este é o nome da aplicação instalado no visor do telefone. Se o AppName está em branco, ele será definido como o nome do projeto quando o projeto é construído.

BackgroundImage

A imagem de fundo da tela.

CloseScreenAnimation

A animação para fechar a tela atual e voltar à tela anterior. As opções válidas são padrão, fade, zoom, slidehorizontal, slidevertical, e nenhum

Icon (designer apenas)

Ícone da aplicação.

OpenScreenAnimation

A animação para mudar para outra tela. As opções válidas são padrão, fade, zoom, slidehorizontal, slidevertical, e nenhum

ScreenOrientation

A orientação do ecrã solicitado, especificado como um valor de texto. Valores utilizados são: paisagem, retrato, sensor, o usuário e não especificado.

Scrollable

Se definido, poderá haver um deslocamento vertical no ecrã.

Title

Título da aplicação.

VersionCode (designer apenas)

Um valor inteiro que deve ser incrementado cada vez que um novo arquivo Android Application Package (APK) é criado para o Google Play Store.

VersionName (designer apenas)

Uma sequência de caracteres que pode ser alterada para permitir que os usuários do Google Play Store possam distinguir entre diferentes versões do App.

- **Eventos:**

- BackPressed()*

Evento gerado quando o botão responsável pela função de VOLTAR for pressionado.

ErrorOccurred (recebe um componente para o campo component, um texto para functionName, um número para errorNumber, um texto para message)

Evento gerado quando ocorre um erro. Apenas alguns erros irão gerar essa condição. Para esses erros, o sistema irá mostrar uma notificação por padrão. Você pode usar esse manipulador de eventos para prescrever um comportamento de erro diferente do padrão.

- Initialize()*

Evento gerado quando a tela inicializar.

OtherScreenClosed (recebe um texto para otherScreenName, any result)

Evento disparado quando outra tela fechou e o controle voltou a esta tela.

- ScreenOrientationChanged()*

Evento gerado quando a orientação do ecrã mudar.

- **Slider - Deslizador**

Um Slider é uma barra de progresso que adiciona um “polegar arrastável”. Você pode tocar o polegar (thumb) e o arrastar para a esquerda ou para a direita para definir a posição polegar slider. Como o polegar Slider é arrastado, ele irá acionar o evento *PositionChanged*, relatando a posição do Slider. A posição indicada do polegar deslizante pode ser utilizada para atualizar dinamicamente outro atributo do componente, tal como o tamanho da fonte de uma caixa de texto ou o raio de uma esfera.

- **Propriedades:**

- ColorLeft*

A cor da barra para a esquerda do polegar.

- ColorRight*

A cor da barra para a direita do polegar.

- MaxValue*

Define o valor máximo do Slider. A alteração do valor máximo também redefine *Thumbposition* como o meio caminho entre o mínimo e o (novo) máximo. Se o

novo máximo for inferior ao mínimo atual, em seguida, mínimo e máximo irão ser definidos para este valor. Definir MaxValue redefine a posição do polegar para o meio caminho entre MinValue e MaxValue e sinaliza o evento PositionChanged.

MinValue

Define o valor mínimo do Slider. A alteração do valor mínimo também redefine Thumbposition como meio caminho entre o (novo) mínimo e o máximo. Se o novo mínimo for maior do que o máximo, mínimo e máximo irão ser definidos para este valor. Definir MinValue redefine a posição do polegar para o meio caminho entre MinValue e MaxValue e sinaliza o evento PositionChanged.

ThumbPosition

Define a posição do polegar do slider. Se este valor for maior que MaxValue, em seguida, ele será definido para o mesmo valor que MaxValue. Se esse valor for menor do que MinValue, em seguida, ele será definido para o mesmo valor que MinValue.

ThumbEnabled

Define se deve ou não exibir o thumb do slider.

- **Eventos:**

PositionChanged (retorna um número para thumbPosition)

Indica que a posição do ponteiro da barra mudou.

- **Spinner**

Um componente que exibe um pop-up com uma lista de elementos. Esses elementos podem ser definidos no Designer ou Editor de Blocks.

- **Propriedades:**

Elements

Retorna uma lista de elementos de texto do Spinner .

ElementsFromString

Define a lista de elementos do Spinner, passados em sequência separada por vírgulas.

Prompt

Texto com o título atual para a janela de Spinner.

Selection

Retorna o item selecionado.

SelectionIndex

O índice do item selecionado no momento, a partir de 1. Se nenhum item for selecionado, o valor será 0.

- **Eventos:**

AfterSelecting(text selection)

Evento chamado após o usuário seleciona um item na lista suspensa.

- **Métodos:**

DisplayDropdown()

Exibe a lista suspensa para a seleção, mesma ação como quando o usuário clica no botão rotativo.

- **TimePicker – Selecionador de tempo**

Um botão que, quando clicado, inicia um diálogo pop-up para permitir que o usuário selecione uma hora.

- **Propriedades:**

Hour

A hora da última definição usando o TimePicker. A hora está em um formato de 24 horas. Se o último conjunto de tempo foi 23:53, essa propriedade retornará 23.

Minute

O minuto da última hora definida usando o TimePicker

- **Eventos:**

AfterTimeSet()

Este evento é executado quando um usuário tiver definido a hora na caixa de diálogo pop-up.

GotFocus()

Indica que o cursor move-se sobre o botão de modo que agora é possível clicar nele.

LostFocus()

Indica que o cursor afastou-se do botão de modo que já não é possível clicar nele.

- **WebView – Visualizador web**

Componente para visualização de páginas da Web. A URL inicial pode ser especificada no Designer ou no Editor de Blocks.

- **Propriedades:**

CurrentPageTitle

Título da página visualizada no momento

CurrentUrl

URL da página visualizada no momento. Isto poderia ser diferente do URL inicial se novas páginas foram visitadas seguindo os links.

FollowLinks

Determina se deve seguir os links quando eles são batidos no WebView. Se você seguir os links, você pode usar o GoBack e GoForward para navegar no histórico do navegador.

HomeUrl

URL da página do WebView que deverá ser aberta inicialmente.

IgnoreSslError

Determina se ignora ou não os erros de SSL. Defina como true para ignorar erros. Use isto para aceitar certificados auto assinados a partir de sites.

PromptforPermission

Se definido, em seguida, solicitará que o usuário do WebView dê permissão para acessar a API de geolocalização. Caso contrário, a permissão é concedida.

UsesLocation (designer apenas)

Deve-se ou não dar a permissão ao aplicativo para usar a API de geolocalização Javascript.

WebViewString

Obtém Cordas do WebView, que é visível através de Javascript no WebView como objeto window.AppInventor.

- **Métodos:**

CanGoBack()

Retorna true se o WebViewer pode voltar na lista de histórico.

CanGoForward()

Retorna true se o WebViewer pode ir para a frente na lista de histórico.

ClearCaches()

Limpar os caches WebViewer

ClearLocations()

Limpar permissões de localização armazenadas.

GoBack()

Volte para a página anterior na lista de histórico. Não faz nada se não houver página anterior.

GoForward()

Vá em frente para a próxima página na lista do histórico. Não faz nada se não há nenhuma página seguinte.

GoHome()

Carrega o URL da página de origem. Isto acontece automaticamente quando a URL HOME é alterado.

GoToUrl (texto url)

Carregue a página no URL dado.

➤ Layout componentes

- **HorizontalArrangement – Arranjo Horizontal**

Use um componente arranjo horizontal para organizar os componentes lado a lado dentro dele.

- **VerticalArrangement – Arranjo Vertical**

Use um componente arranjo vertical para organizar os componentes um abaixo do outro dentro dele.

- **TableArrangement - Arranjo em Tabela**

Use de um componente arranjo em tabela para organizar os componentes em tabela. Numa TableArrangement, os componentes são dispostos numa malha de linhas e colunas, com não mais do que um componente visível em cada célula. Se vários componentes ocupar a mesma célula, apenas o último será visível.

- **Propriedades:**

- Rows

- O número de linhas na tabela.

- Columns

- O número de colunas na tabela.

- Componentes de mídia - Media Components

- **Camcorder - Câmera de vídeo**

Um componente para gravar um vídeo.

- **Eventos:**

- AfterRecording (clipe de texto)

- Indica que um vídeo foi gravado com a câmera e fornece o caminho para a imagem armazenada.

- **Métodos:**

- RecordVideo ()

- Grava um vídeo, em seguida, dispara o evento AfterRecording.

- **Camera**

Use um componente da câmera para tirar uma foto no telefone.

- **Propriedades:**

- UseFront

- Especifica se a câmera frontal deve ser usada (quando disponível). Se o dispositivo não tem uma câmera frontal, esta opção será ignorada e a câmera vai abrir normalmente.

- **Métodos:**

- TakePicture()

Abre a câmera do telefone para permitir que uma foto seja tirada.

- **Eventos:**

- AfterPicture (imagem do texto)

Chamado depois que a foto é tirada. O argumento de texto de imagem é o caminho que pode ser utilizado para localizar a imagem no telefone.

- **ImagePicker – Seleccionador de imagem**

Quando o usuário clica em um seccionador de imagem, a galeria de imagens do dispositivo é exibida, e o usuário pode escolher uma imagem.

- **Propriedades:**

Selection

Caminho para o arquivo que contém a imagem que foi selecionada.

- **Eventos:**

- AfterPicking()

Evento iniciado após a atividade do seccionador retornar seu resultado.

- BeforePicking ()

Evento iniciado quando o componente é clicado, mas antes da atividade do seccionador ser iniciada.

- GotFocus ()

Indica que o cursor move sobre o botão de modo que agora é possível clicar nele.

- LostFocus ()

Indica que o cursor afastou-se do botão de modo que já não é possível clicar nele.

- **Métodos:**

- Open()

Abre o seccionador, assim como se o usuário clicar sobre ele.

- **Player**

Componente multimídia que reproduz a vibração do telefone áudio e controles. Este componente é melhor para arquivos de som longos, tais como músicas, enquanto o componente sound é mais eficiente para arquivos curtos, como efeitos sonoros.

- **Propriedades:**

- IsPlaying*

Retorna se a mídia está reproduzindo.

- Loop*

Se definido, o player reproduzirá repedidas vezes.

- PlayOnlyInForeground*

Se definido, o player irá pausar a reprodução ao sair da tela atual; se falso (opção padrão), o player continuará reproduzindo mesmo que a tela atual feche.

- Source*

Arquivo de mídia que irá reproduzir no player.

- Volume*

Define o volume para um número entre 0 e 100

- **Eventos:**

- Completed()*

Inicia quando a mídia chega ao fim.

- OtherPlayerStarted()*

Este evento é iniciado quando outro player começou (e o atual player está tocando ou pausado, mas não parado).

- **Métodos:**

- Pause()*

Suspende o player se ele estiver reproduzindo.

- Start()*

Reproduz a mídia.

- Stop()*

Para a reprodução de mídia e procura o início da música.

Vibrate (número em milissegundos)

Vibra para o número especificado de milissegundos.

➤ **Sound**

Um componente multimídia que reproduz arquivos de som e, opcionalmente, vibra para o número de milissegundos (milésimos de segundo) especificado no Editor de Blocks.

▪ **Propriedades:**

MinimumInterval

O intervalo mínimo entre sons. Um novo som só irá iniciar quando o intervalo acabar.

Source

O nome do arquivo de som.

▪ **Métodos:**

Pause ()

Interrompe a reprodução do som se ele está sendo executado.

Play()

Inicia a reprodução do som.

Resume ()

Continua a reprodução do som depois de uma pausa.

Stop ()

Para de tocar o som se ele está sendo reproduzido.

Vibrate (número em milissegundos)

Vibra para o número especificado de milissegundos.

➤ **SoundRecorder – Gravador de som**

Componente multimídia que grava áudios.

▪ **Eventos:**

AfterSoundRecorded (som)

Inicia quando a captura de um som for terminada. Retorna o som capturado.

StartedRecording ()

Inicia quando o gravador é iniciado, e pode ser interrompido.

StoppedRecording ()

Inicia quando o gravador parou, e pode ser iniciado novamente.

▪ **Métodos:**

Play ()

Inicia a gravação.

Stop()

Para a gravação.

➤ **SpeechRecognizer:**

Use um componente de reconhecimento de voz para ouvir o discurso do usuário e converter o som falado em texto usando o recurso de reconhecimento de voz do Android.

▪ **Propriedades:**

Result

O último texto produzido pelo reconhecedor.

▪ **Métodos:**

GetText ()

Pede ao usuário para falar, e converte a fala em texto.

▪ **Eventos:**

AfterGetting (resultado texto)

Inicia após o reconhecimento de um texto. Retorna o texto.

BeforeGettingText ()

Inicia antes de o reconhecedor ser iniciado.

➤ **TextToSpeech – Texto para fala**

O componente TextToSpeech fala um determinado texto em voz alta.

▪ **Propriedades:**

AvailableCountries

Lista dos códigos de países disponível neste dispositivo para uso com TextToSpeech.

AvailableLanguages

Lista das línguas disponíveis neste dispositivo para uso com TextToSpeech.

Country

Código do país a ser usado para geração de fala. Isso pode afetar a pronúncia. Por exemplo, Inglês Britânico (GBR) vai soar diferente de US Inglês (EUA). Nem todo o código do país afetará todas as línguas.

Language

Define o idioma do TextToSpeech. Isso muda a maneira que as palavras são pronunciadas, não a verdadeira linguagem que é falada.

Pitch

Define o passo para TextToSpeech. Os valores devem estar entre 0 e 2, onde valores mais baixos irão baixar o tom de voz sintetizada e maiores valores irão aumentá-lo.

Result

Resultado do TextToSpeech.

SpeechRate

Define a velocidade da fala para o TextToSpeech. Os valores devem estar entre 0 e 2.

▪ **Eventos:**

AfterSpeaking (resultado boolean)

Evento iniciado após a mensagem ser falada.

BeforeSpeaking ()

Evento iniciado quando Speak é chamado, antes que a mensagem é falada.

- **Métodos:**

Speak (mensagem de texto)

Fala a mensagem dada.

➤ **VideoPlayer**

Um componente capaz de reproduzir vídeos multimídia.

Os arquivos de vídeo devem estar em 3GPP (.3gp) ou MPEG-4 (.mp4) formatos.

App Inventor for Android permite somente arquivos de vídeo com menos de 1 MB e limita o tamanho total de um aplicativo para 5 MB.

- **Propriedades:**

FullScreen

Tela cheia.

Source

O "caminho" para o vídeo. Normalmente, este será o nome do arquivo de vídeo, que deve ser adicionado no Designer.

Volume

Define o volume a um número entre 0 e 100.

- **Eventos:**

Completed()

Inicia quando o vídeo chega ao fim.

- **Métodos:**

getDuration ()

Retorna a duração do vídeo em milissegundos.

Pause ()

Pausa a reprodução do vídeo.

SeekTo (número ms)

Avança para o tempo solicitado (especificada em milissegundos) no vídeo.

Start()

Inicia a reprodução do vídeo.

➤ **YandexTranslate**

Use este componente para traduzir palavras e frases entre diferentes idiomas. Este componente precisa de acesso à Internet, uma vez que irá solicitar traduções para o serviço Yandex.Translate.

▪ **Eventos:**

GotTranslation (responseCode, tradução)

Evento disparado quando o serviço Yandex.Translate retorna o texto traduzido. Este evento também fornece um código de resposta para o tratamento de erros. Se o responseCode não é 200, então algo deu errado com a chamada, e a tradução não estarão disponíveis.

▪ **Métodos:**

RequestTranslation (languageToTranslateTo, textToTranslate)

Ao fornecer uma língua-alvo para traduzir a (por exemplo, 'es' para o espanhol, 'en' para Inglês, ou 'ru' para russo), e uma palavra ou frase para traduzir, este método irá solicitar uma tradução para o Yandex.

➤ **Drawing and Animation Components – Componentes de desenho e animação**

• **Canvas**

Um painel retangular bidimensional sensível ao toque, onde componentes sprites podem ser movidos. Qualquer local na tela pode ser especificado como um par de (x, y).

X é o número de pixels de distância da margem esquerda dos Canvas.

Y é o número de pixels de distância a partir da borda superior do Canvas.

▪ **Propriedades:**

LineWidth

A largura de linhas desenhadas na tela.

PaintColor

A cor em que as linhas são desenhadas

TextAlignment

Determina o alinhamento do texto elaborado por DrawText () ou DrawAngle () com respeito ao ponto especificado por esta ordem.

- **Eventos:**

Dragged (número startX, número startY, número PrevX, número prevY, número currentX, número currentY, draggedSprite boolean)

Evento iniciado quando o usuário faz um arrasto de um ponto (PrevX, prevY) para outro (x, y). O par (startX, startY) indica onde o usuário primeiro tocou a tela, e "draggedSprite" indica se um sprite está sendo arrastado.

Flung (número x, número y, número de velocidade, número da posição, número Xvel, número Yvel, flungSprite boolean)

Evento iniciado quando um gesto de arremessar (golpe rápido) é feito na tela: fornece a posição (x, y) do início do arremesso, em relação ao canto superior esquerdo da tela. Também fornece a velocidade (pixels por milissegundo) e a orientação (0-360 graus), bem como a velocidade x e a velocidade y do vetor do arremesso. O valor "flungSprite" é verdadeiro se um sprite foi localizado próximo ao ponto de partida do arremesso.

TouchDown (número x, número y)

Evento iniciado quando o usuário toca a tela não retira o dedo por determinado tempo: fornece a posição (x, y) do contato, em relação ao canto superior esquerdo da tela.

TouchUp (número x, número y)

Evento iniciado quando o usuário para de tocar a tela (levanta o dedo depois de um evento TouchDown): fornece a posição (x, y) do contato, em relação ao canto superior esquerdo da tela.

Touched (número x, número y, touchedSprite)

Quando o usuário toca a tela e, em seguida, imediatamente levanta o dedo: fornece a posição (x, y) do contato, em relação ao canto superior esquerdo da tela. TouchedSprite é true se o mesmo toque também tocou um sprite, e false caso contrário.

- **Métodos:**

Clear ()

Limpa qualquer coisa feita no Canvas, mas não qualquer cor de fundo ou imagem.

DrawCircle (número x, número y, número de raio)

Desenha um círculo (preenchido) nas coordenadas indicadas na tela, com o raio dado.

DrawLine (número x1, número y1, o número de x2, número y2)

Desenha uma linha entre as coordenadas indicadas.

DrawPoint (número x, número y)

Desenha um ponto nas coordenadas indicadas.

DrawText (texto , número x, número y)

Desenha o texto especificado com relação às coordenadas especificadas usando os valores das propriedades FontSize e TextAlignment.

DrawTextAtAngle (texto, número x, número y, número do ângulo)

Desenha o texto de partida especificado nas coordenadas especificadas no ângulo especificado usando os valores das propriedades FontSize e TextAlignment.

GetBackgroundPixelColor (número x, número y)

Obtém a cor do ponto especificado. Isso inclui o fundo e qualquer desembainhado de pontos, linhas ou círculos, mas não sprites.

GetPixelColor (número x, número y)

Obtém a cor do ponto especificado.

Save ()

Salva uma imagem deste canvas para armazenamento externo do dispositivo. Se ocorrer um erro, evento ErrorOccurred da tela será chamado.

SaveAs (nome do arquivo de texto)

Salva uma imagem deste canvas para armazenamento externo do dispositivo no arquivo chamado filename. Nome do arquivo deve terminar com um dos formatos: .jpg, .jpeg, ou .png, que determina o tipo de arquivo.

SetBackgroundPixelColor (número x, número y, número da cor)

Define a cor do ponto especificado. Isso é diferente de drawPoint por ter um argumento para a cor.

- **Ball**

Uma bola que pode ser colocado em um canvas, onde ela pode reagir a toques, interagir com outros sprites (ImageSprite s e outros Bolas) e a borda da tela, e mover-se de acordo com seus valores de propriedade.

- **Propriedades:**

- Enabled

Controla se a bola se move quando sua velocidade é diferente de zero.

- Heading

Retorna a orientação da bola em graus.

- Interval

O intervalo, em milissegundos, em que a posição da bola será atualizada. Por exemplo, se o intervalo é de 50 e a velocidade é de 10, então a bola moverá 10 pixels cada 50 milissegundos.

- PaintColor

Cor da bola.

- Raio

Raio da bola.

- Velocidade

A velocidade na qual a bola se move.

- X

A coordenada horizontal da borda esquerda do sprite, aumentando à medida que a bola move para a direita.

- Y

A coordenada vertical do topo do sprite, aumentando à medida que a bola se move para baixo.

- Z

Como a bola deve ser mergulhada em relação a outros sprites, com camadas de número mais alto na frente de camadas inferiores de numeração.

- **Eventos:**

- CollidedWith (outro componente)

Evento iniciado quando a bola colide com outro sprite.

- Dragged (número startX, número startY, número PrevX, número prevY, número currentX, número currentY)

Evento iniciado quando o usuário arrasta a bola (PrevX, prevY) para outro (x, y). O par (startX, startY) indica onde o usuário primeiro tocou a bola.

EdgeReached (número da borda)

Manipulador de eventos chamado quando o sprite atinge uma borda da tela. Borda aqui é representada como um número inteiro que indica uma das oito direções: norte (1), Nordeste (2), a leste (3), Sudeste (4), sul (-1), sudoeste (-2), oeste (-3) e noroeste (-4).

Flung (número x, número y, número de velocidade, número da posição, número Xvel, número Yvel)

Evento iniciado quando um gesto de arremessar (golpe rápido) é feito na bola: fornece a posição (x, y) do início do arremesso, em relação ao canto superior esquerdo da tela. Também fornece a velocidade (pixels por milissegundo) e a orientação (0-360 graus), bem como a velocidade x e a velocidade y do vetor do arremesso.

NoLongerCollidingWith (outro componente)

Evento indicando que um par de sprites não colidiu.

TouchDown (número x, número y)

Evento iniciado quando o usuário toca a bola e não retira o dedo por determinado tempo: fornece a posição (x, y) do contato, em relação ao canto superior esquerdo da tela.

TouchUp (número x, número y)

Evento iniciado quando o usuário para de tocar a bola (levanta o dedo depois de um evento TouchDown): fornece a posição (x, y) do contato, em relação ao canto superior esquerdo da tela.

Touched (número x, y número, touchedSprite)

Quando o usuário toca a bola e, em seguida, imediatamente levanta o dedo: fornece a posição (x, y) do contato, em relação ao canto superior esquerdo da tela. TouchedSprite é true se o mesmo toque também tocou um sprite, e false caso contrário.

▪ **Métodos:**

Bounce (número da borda)

Faz com que a bola de um salto (rebata). Para salto normal, o argumento da borda deve ser o único retornado pelo EdgeReached.

CollidingWith (outro componente)

Indica se uma colisão foi registrada entre este componente e o componente passado por parâmetro.

MoveIntoBounds ()

Se a bola excede os limites de uma área especificada, este método alinha o tamanho da bola com o da área.

MoveTo (número x, número y)

Mova a bola para o local especificado pelas coordenadas x e y.

PointInDirection (número x, número y)

Se a bola estiver em movimento ou em contato com a borda do canvas e esse método for chamado, a bola mudará sua direção, passando pelo ponto dado pelas coordenadas (x, y).

PointTowards (componente alvo)

Se a bola estiver em movimento ou em contato com a borda do canvas e esse método for chamado, a bola mudará sua direção, passando pelo componente dado por target.

- **ImageSprite**

Uma imagem que pode ser colocada em um canvas, onde ela pode reagir a toques, interagir com outros sprites (ImageSprites e Bolas) e a borda da tela, e mover-se de acordo com seus valores de propriedade.

- **Propriedades:**

- Picture

- A imagem do Sprite.

- Rotates

- Se definido, a imagem do sprite gira para coincidir com o título do Sprite. O sprite gira em torno do seu ponto central.

- **Eventos:**

- Os mesmos eventos do componente Ball.

- **Métodos:**

- Os mesmos métodos do componente Ball.

➤ Sensor Components - Componentes de sensores

• **AccelerometerSensor -**

Componente não visível que pode detectar a agitação e medir a aceleração aproximadamente em três dimensões (podendo entender a inclinação do dispositivo) usando unidades do Sistema internacional (SI) (m / s²).

▪ **Propriedades:**

xAccel

0 quando o telefone está em repouso sobre uma superfície plana, positivo quando o telefone está inclinado para a direita (ou seja, o lado esquerdo é elevado), e negativa quando o telefone está inclinado para a esquerda (ou seja, o seu lado direito é levantado).

yAccel

0 quando o telefone está em repouso sobre uma superfície plana, positivo quando seu fundo é levantado, e negativo quando seu topo é levantado.

zAccel

Igual a -9,8 (gravidade da Terra em metros por segundo quando o dispositivo está em paralelo ao chão com o visor virado para cima, 0 quando perpendicular ao chão, e 9,8 quando virado para baixo. O valor também pode ser afetado em acelerá-lo com ou contra a gravidade.

Available

Indica se o acelerômetro está disponível no dispositivo.

MinimumInterval

O intervalo mínimo entre shakes (sacudidas) do dispositivo.

Sensibility

Um número que codifica o quão sensível é o acelerômetro. As opções são: 1 = fraco, 2 = moderado, 3 = forte.

▪ **Eventos:**

AccelerationChanged (número xAccel, número yAccel, número zAccel)

Iniciado quando a aceleração do dispositivo é alterada, nas dimensões X, Y e/ou Z.

Shaking()

Inicia quando o dispositivo começa ou continua a ser sacudido.

- **BarcodeScanner - Leitor de códigos de barra**

Componente para leitura de QR Codes.

- **Propriedades:**

Result

Resultado da verificação anterior.

UseExternalScanner

Se definido, o App Inventor irá procurar e usar um programa de digitalização externo, como "Bar Code Scanner".

- **Eventos:**

AfterScan (resultado texto)

Iniciado após a leitura de um qr code.

- **Métodos:**

DoScan ()

Começa uma varredura de código de barras, utilizando a câmara. Quando a pesquisa estiver concluída, o evento AfterScan será gerado.

- **Clock - Relógio**

Componente não visível que fornece o instante no tempo usando o relógio interno no phone.It, e pode disparar um temporizador em intervalos definidos regularmente e executar cálculos de tempo, manipulações, e conversões.

- **Propriedades:**

TimerAlwaysFires

Se definido, dispara mesmo quando o aplicativo não está mostrando na tela.

TimerEnabled

Se definido, o cronômetro fica ativo.

TimerInterval

Intervalo entre eventos do cronômetro em ms (milissegundos).

- **Eventos:**

Timer()

Evento que dispara de acordo com a propriedade TimerInterval. Se esta estiver com a quantidade 1000 ms (1 segundo), então o evento disparará de 1 em 1 segundo.

- **Métodos:**

AddDays(Instante, número de dias)

Adiciona o número de dias passado por parâmetro, ao instante passado por parâmetro.

AddHours (Instante, número de horas)

Adiciona o número de horas passado por parâmetro, ao instante passado por parâmetro.

AddMinutes (Instante, número de minutos)

Adiciona o número de minutos passado por parâmetro, ao instante passado por parâmetro.

AddMonths (Instante, número de meses)

Adiciona o número de meses passado por parâmetro, ao instante passado por parâmetro.

AddSeconds (Instante, número de segundos)

Adiciona o número de segundos passado por parâmetro, ao instante passado por parâmetro.

AddWeeks (Instante, número de semanas)

Adiciona o número de semanas passado por parâmetro, ao instante passado por parâmetro.

AddYears(Instante, número de anos)

Adiciona o número de passado por parâmetro, ao instante passado por parâmetro.

DayofMonth (Instante)

Retorno o dia do mês no instante passado por parâmetro.

Duration (início, final)

Retorna a quantidade de milissegundos decorridos entre o início (start) e o final (end).

FormatDate (Instante, padrão de texto)

Retorna a data no instante passado por parâmetro no padrão especificado.

FormatDateTime (instante, padrão de texto)

Retorna a data e o horário no instante passado por parâmetro no padrão especificado.

FormatTime (Instante)

Retorna o horário no instante passado por parâmetro.

GetMillis (Instante)

O instante de tempo medido como milissegundos desde 1970.

Hour (Instante)

Retorna a hora no instante passado por parâmetro.

MakeInstant (texto de)

Retorna várias informações especificadas pelos formatos MM / DD / AAAA hh : mm : ss ou MM / DD / AAAA ou hh: mm

MakeInstantFromMillis (número de millis)

Um instante no tempo especificado pelos milissegundos desde 1970.

Minute (Instante)

Retorna os minutos no instante passado por parâmetro.

Month (Instante)

Retorna o mês do ano no instante passado por parâmetro, representado como um número de 1 a 12.

MonthName (Instante)

Retorna o nome do mês no instante passado por parâmetro.

Now ()

O instante atual do relógio do telefone.

Second (Instante)

Retorna os segundos no instante passado por parâmetro.

SystemTime ()

Tempo interno do telefone.

WeekDay (Instante)

Retorna o dia da semana no instante passado por parâmetro, representado como um número de 1 (domingo) a 7 (Sábado).

WeekdayName (Instante)

Retorna o nome do dia da semana no instante passado por parâmetro.

Year (Instante)

Retorna o ano no instante passado por parâmetro

- **LocationSensor – Sensor de Localização**

Componente não visível de fornecimento de informações de localização, incluindo longitude, latitude, altitude (se for suportado pelo dispositivo), e endereço.

- **Propriedades:**

Accuracy

Indica o nível de precisão do dispositivo, em metros.

Altitude

Altitude do dispositivo Android, se disponível.

AvailableProviders

Lista de fornecedores de serviços disponíveis, como o GPS ou de rede. Esta informação é fornecida como uma lista e em forma de texto.

CurrentAddress

Endereço físico do dispositivo Android.

DistanceInterval

Determina o intervalo mínimo de distância, em metros, que o sensor irá tentar usar para o envio de atualizações de localização.

Enabled

Se definido, as informações de localização estarão disponíveis.

HasAccuracy

Se definido, o dispositivo Android pode relatar seu nível de precisão.

HasAltitude

Se definido, o dispositivo Android pode relatar sua altitude.

HasLongitudeLatitude

Se definido, o dispositivo Android pode relatar longitude e latitude.

Latitude

Latitude do dispositivo Android.

Longitude

Longitude do dispositivo Android.

ProviderLocked

O dispositivo não irá alterar o fornecedor de serviços.

ProviderName

O prestador de serviços atual. O provedor provavelmente será GPS ou de rede.

TimeInterval

Determina o intervalo mínimo de tempo, em milissegundos, que o sensor irá tentar usar para o envio de atualizações de localização.

▪ **Eventos:**

LocationChanged (número latitude, número longitude, número de altitude)

Chamado quando o dispositivo Android relata um novo local.

StatusChanged (provedor de texto, estado do texto)

Chamado quando o estatuto do prestador de serviços sofre alterações.

▪ **Métodos:**

LatitudeFromAddress (LocationName)

Determina a latitude do endereço fornecido.

LongitudeFromAddress (LocationName)

Determina a longitude do endereço fornecido.

• **OrientationSensor – Sensor de Orientação**

Use um componente sensor de orientação para determinar a orientação espacial do telefone.

- **Propriedades:**

- Available

- Indica se o sensor de orientação está presente no dispositivo Android.

- Azimuth

- Retorna o ângulo de azimuth do dispositivo. 0 graus quando a parte superior do dispositivo está a apontar para o norte, 90 graus quando se está apontando para o leste, 180 graus quando se está apontando para o sul, 270 graus quando ele está apontando para o oeste, etc.

- Pitch

- Retorna o ângulo de inclinação do dispositivo.

- Roll

- Retorna o ângulo de rolagem do dispositivo.

- Magnitude

- Retorna um número entre 0 e 1 que indica o quanto o dispositivo está inclinado. Dá a magnitude da força que seria sentida por uma bola ao rolar sobre a superfície do dispositivo.

- Angle

- Retorna um ângulo que diz a direção em que o dispositivo está inclinado. Ou seja, ele informa o sentido da força que seria sentida por uma bola ao rolar sobre a superfície do dispositivo.

- **Eventos:**

- OrientationChanged (número azimuth, número de inclinação, número de rolo)

- Chamado quando a orientação muda.

- **ProximitySensor - Sensor de proximidade**

- Um componente que pode medir a proximidade de um objeto (em cm) em relação ao ecrã de visualização de um dispositivo.

- **Propriedades:**

- Available

- Relata se o dispositivo tem um sensor de proximidade

- Enable

- Se ativado, a aplicação poderá realizar tarefas com o sensor.

KeepRunningWhenOnPause

Se definido, ele irá manter sensor de proximidade mesmo quando o aplicativo não estiver visível.

Distance

Retorna a distância do objeto ao dispositivo.

MaximumRange

Informa o alcance máximo de ProximitySensor do dispositivo.

▪ **Eventos:**

ProximityChanged (número distância)

Chamado quando a distância (em centímetros) do objeto ao dispositivo sofre alterações.

➤ Social Components - Componentes Sociais

• **ContactPicker – Seleccionador de contatos**

Um botão que, quando clicado, exibe uma lista dos contatos para escolher entre.

▪ **Propriedades**

ContactName

O nome do contato.

EmailAddress

Endereço de e-mail principal do contato.

▪ **Eventos**

AfterPicking ()

Evento iniciado após uma seleção ser feita no ContactPicker.

BeforePicking ()

Evento iniciado quando o ContactPicker é aberto, mas antes que uma seleção seja feita.

GotFocus ()

Indica que o cursor move sobre o botão de modo que agora é possível clicar nele.

LostFocus ()

Indica que o cursor afastou-se do botão de modo que já não é possível clicar nele.

- **Métodos:**

Open()

Abre o selecionador, como se o usuário clicar sobre ele.

- **EmailPicker – Selecionador de e-mail**

Um EmailPicker é uma espécie de caixa de texto. Se o usuário começa a introduzir o nome ou endereço de e-mail de um contato, o telefone irá mostrar um menu suspenso de opções que completam a entrada.

- **Métodos:**

RequestFocus ()

Define o EmailPicker ativo.

- **Eventos:**

GotFocus ()

Indica que o cursor move sobre o botão de modo que agora é possível clicar nele.

LostFocus ()

Indica que o cursor afastou-se do botão de modo que já não é possível clicar nele.

- **PhoneCall - Telefonema**

Um componente não visível que faz uma chamada telefônica para o número especificado na propriedade PhoneNumber.

- **Propriedades**

PhoneNumber

Número do telefone.

- **Eventos**

IncomingCallAnswered (phoneNumber)

Evento indicando que uma chamada telefônica é respondida. phoneNumber é o número chamada telefônica.

PhoneCallEnded (status, phoneNumber)

Evento indicando que uma chamada de telefone foi encerrada. Se o status for 1, chamada de entrada é perdida ou rejeitada; se o status é 2, chamada é atendida antes de desligar; se o estado é 3, chamada de saída é desligou. phoneNumber é o número de telefone de entrada / saída.

PhoneCallStarted (status, phoneNumber)

Evento indicando que um telefonema foi iniciado. Se o status for 1, chamada de entrada está tocando; se o status é 2, chamada de saída é discado. phoneNumber é o número de telefone de entrada / saída.

- **Métodos:**

MakePhoneCall ()

Faz uma chamada de telefone através do número na propriedade PhoneNumber.

- **PhoneNumberPicker – Seleccionador de números de telefone**

Um botão que, quando clicado, exibe uma lista de números de telefone dos contatos para escolher entre.

- **Propriedades:**

ContactName

O nome do contato

EmailAddress

Endereço de e-mail do contato.

PhoneNumber

Número de telefone do contato.

- **Eventos:**

AfterPicking ()

Evento iniciado após uma seleção ser feita no ContactPicker.

BeforePicking ()

Evento iniciado quando o ContactPicker é aberto, mas antes que uma seleção seja feita.

GotFocus ()

Indica que o cursor move sobre o botão de modo que agora é possível clicar nele.

LostFocus ()

Indica que o cursor afastou-se do botão de modo que já não é possível clicar nele.

▪ **Métodos:**

Open ()

Abre o selecionador, como se o usuário clicar sobre ele.

• **Sharing - Compartilhamento**

Sharing é um componente não visível que permite o compartilhamento de arquivos e / ou mensagens entre seu aplicativo e outros aplicativos instalados em um dispositivo. O componente irá exibir uma lista dos aplicativos instalados que podem lidar com as informações fornecidas, e permitirá que o usuário escolha um para compartilhar o conteúdo com, por exemplo, um aplicativo de e-mail, uma rede sociais, um aplicativo de mensagens de texto, e assim por diante.

▪ **Métodos:**

ShareFile (arquivo)

Compartilha um arquivo através de qualquer aplicação capaz instalada no telefone, exibindo uma lista dos aplicativos disponíveis e permitindo que o usuário escolha um da lista. O aplicativo selecionado será aberto com o arquivo inserido nele.

ShareFileWithMessage (arquivo, mensagem)

Compartilha um arquivo e uma mensagem através de qualquer aplicação capaz instalada no telefone, exibindo uma lista de aplicativos disponíveis e permitindo que o usuário escolha um da lista. O aplicativo selecionado será aberto com o arquivo e mensagem inseridos nele.

ShareMessage (mensagem)

Compartilha uma mensagem através de qualquer aplicação capaz instalada no telefone, exibindo uma lista dos aplicativos disponíveis e permitindo que o usuário escolha um da lista. O aplicativo selecionado será aberto com a mensagem inserida nele.

- **Texting**

Um componente que irá, quando o SendMessage método for chamado, enviar a mensagem de texto especificado na mensagem de estabelecimento para o número de telefone especificado na propriedade PhoneNumber.

- **Propriedades:**

GoogleVoiceEnabled

Se definido, então SendMessage tentará enviar mensagens através de Wi-Fi usando o Google Voice. Isso requer que o app Google Voice esteja instalado e configurado no telefone ou tablet, com uma conta do Google Voice. Se GoogleVoiceEnabled é falso, o dispositivo deve ter um serviço de mensagens de texto, a fim de enviar ou receber mensagens com este componente.

Message

A mensagem a ser enviada quando o método SendMessage é chamado.

PhoneNumber

O número para o qual a mensagem vai ser enviada.

ReceivingEnabled

Se definido como (OFF), as mensagens não serão recebidas. Se definido como (Foreground), as mensagens serão recebidas quando o app estiver aberto. Se definido como (Always), as mensagens sempre serão recebidas.

- **Eventos:**

MessageReceived (número, MessageText)

Evento que é gerado quando uma mensagem de texto é recebida pelo telefone.

- **Métodos:**

SendMessage ()

Enviar uma mensagem de texto.

- **Twitter**

Um componente não visível que permite a comunicação com o Twitter.

- **Propriedades:**

ConsumerKey

A chave do consumidor para ser usado para autorização com o Twitter via OAuth.

ConsumerSecret

O segredo do consumidor para ser usado para autorização com o Twitter via OAuth

DirectMessages

Esta propriedade contém uma lista das mensagens mais recentes que citam o usuário logado. Inicialmente, a lista está vazia. Para configurá-la, o programa deve:

Chamar o método `Authorize`.

Aguardar o evento `IsAuthorize`.

Chamar o método `RequestDirectMessages`.

Aguardar o evento `DirectMessagesReceived`.

O valor dessa propriedade será então definido para a lista de mensagens diretas recuperadas (e manter esse valor até qualquer chamada subsequente para `RequestDirectMessages`).

Followers

Esta propriedade contém uma lista de seguidores do usuário logado. Inicialmente, a lista está vazia. Para configurá-la, o programa deve:

Chamar o método `Authorize`.

Aguardar o evento `IsAuthorize`.

Chamar o método `RequestFollowers`.

Aguardar o evento `FollowersReceived`.

O valor dessa propriedade será então definido para a lista de seguidores (e manter o seu valor até qualquer chamada subsequente para `RequestFollowers`).

FriendTimeline

Esta propriedade contém as 20 mensagens mais recentes dos usuários seguidos. Inicialmente, a lista está vazia. Para configurá-la, o programa deve:

Chamar o método `Authorize`.

Aguardar o evento `IsAuthorize`.

Especificar os usuários seguidos com um ou mais chamadas para o método `Follow`.

Chamar o método RequestFriendTimeline.

Aguardar o evento FriendTimelineReceived.

O valor dessa propriedade será então definido para a lista de mensagens (e manter o seu valor até qualquer chamada subsequente para RequestFriendTimeline) .

Mentions

Esta propriedade contém uma lista de menções (referências) do usuário logado. Inicialmente, a lista está vazia. Para configurá-la, o programa deve:

Chamar o método Authorize.

Aguardar o evento IsAuthorize.

Chamar o método RequestMentions.

Aguardar o evento MentionsReceived.

O valor dessa propriedade será então definido para a lista de menções (e manter o seu valor até qualquer chamada subsequente para RequestMentions).

SearchResults

Esta propriedade, que é inicialmente vazia, está definida para uma lista de resultados de pesquisa após o programa:

Chamar o método SearchTwitter.

Aguardar o evento SearchSuccessful.

O valor da propriedade será, então, o mesmo que o parâmetro para SearchSuccessful . Note que não é necessário chamar método Authorize antes de chamar SearchTwitter .

UserName

O nome de usuário do usuário autorizado. Vazio se não houver nenhum usuário autorizado.

• **Eventos:**

DirectMessagesReceived (mensagens de lista)

Este evento é gerado quando as mensagens recentes solicitadas através do RequestDirectMessages foram recuperadas. Uma lista de mensagens pode,

então, ser encontrada no parâmetro messages ou a mensagens de estabelecimento.

FollowersReceived (lista Seguidores2)

Este evento é gerado quando todos os seguidores do usuário logado solicitados através do RequestFollowers foram recuperados. A lista dos seguidores pode ser encontrada no parâmetro followers2 ou na propriedade Followers.

FriendTimelineReceived (lista linha do tempo)

Este evento é gerado quando as mensagens solicitadas através do RequestFriendTimeline foram recuperadas. O parâmetro timeline e a propriedade Timeline irão conter uma lista de listas, onde cada sub-lista contém uma atualização de status do formulário (mensagem username)

IsAuthorized ()

Este evento é gerado após o programa chamar o método Authorize, se a autorização foi bem sucedida. É também gerado após uma chamada para CheckAuthorized se já temos um token de acesso válido. Após este evento ter sido levantado, qualquer outro método para este componente pode ser chamado.

MentionsReceived (lista menciona)

Este evento é gerado quando as menções do usuário logado solicitadas através do RequestMentions foram recuperadas. Uma lista das menções pode ser encontrada no parâmetro mentions ou na propriedade Mentions.

SearchSuccessful (searchResults)

Este evento é gerado quando os resultados da pesquisa solicitada através do SearchSuccessful foram recuperados. A lista de resultados pode então ser encontrada no parâmetro results ou na propriedade Results.

▪ **Métodos:**

Authorize()

Redireciona o usuário para acessar o Twitter através do navegador da Web, usando o protocolo OAuth, se não tiver autorização.

CheckAuthorized ()

Verifica se já tem acesso, e em caso afirmativo, provoca o manipulador de eventos isAuthorized.

DeAuthorize ()

Remove a autorização do Twitter a partir desta instância do aplicativo em execução.

DirectMessage (usuário, mensagem)

Isso envia uma mensagem direta (privado) para o usuário especificado. A mensagem será aparada se exceder 160 caracteres.

Requisitos: Isso só deve ser chamado após o evento isAuthorized ser levantado, indicando que o usuário está registrado com sucesso no Twitter.

Follow(usuário)

Começa a seguir um usuário.

RequestDirectMessages ()

Requisita as 20 mensagens diretas mais recentes enviadas para o usuário logado. Quando as mensagens são recuperadas, o sistema irá levantar o evento DirectMessagesReceived e definir a propriedade DirectMessages à lista de mensagens.

Requisitos: Isso só deve ser chamado após o evento isAuthorized ser levantado, indicando que o usuário está registrado com sucesso no Twitter.

RequestFollowers ()

Obtém quem está seguindo você.

RequestFriendTimeline()

Obtém as mais recentes 20 mensagens na linha do tempo do usuário.

RequestMentions()

Solicita as 20 mais recentes referências ao usuário logado. Quando as referências são recuperadas, o sistema irá iniciar o evento MentionsReceived e definir a propriedade Mentions à lista de menções.

Requisitos: Isso só deve ser chamado após o evento isAuthorized ser levantado, indicando que o usuário está registrado com sucesso no Twitter.

SearchTwitter (texto)

Isto procura no Twitter o texto dado.

Requisitos: Isso só deve ser chamado após o evento isAuthorized ser levantado, indicando que o usuário está registrado com sucesso no Twitter.

StopFollowing (usuário)

Deixa de seguir um usuário.

Tweet (status)

Isso envia um tweet como o usuário conectado com o status especificado, que será cortado se exceder 160 caracteres.

Requisitos: Isso só deve ser chamado após o evento `isAuthorized` ser levantado, indicando que o usuário está registrado com sucesso no Twitter.

TweetWithImage (status, imagepath)

Isso envia um tweet como o usuário conectado com o status especificado e um caminho para a imagem a ser carregada, que serão cortados se exceder 160 caracteres. Se uma imagem não for encontrada ou for inválida, a atualização não será enviada.

Requisitos: Isso só deve ser chamado após o evento `isAuthorized` ser levantado, indicando que o usuário está registrado com sucesso no Twitter.

➤ **Storage Components – Componentes de Armazenamento**

• **File - Arquivo**

Componente não visível para armazenar e recuperar arquivos. Use este componente para escrever ou ler arquivos em seu dispositivo. O comportamento padrão é para gravar arquivos no diretório de dados privado associado à sua App. O companheiro escreve arquivos de / sdcard / ApplInventor / dados para facilitar a depuração. Se o caminho do arquivo começa com uma barra (/), em seguida, o arquivo é criado em relação ao / sdcard. Por exemplo, escrever um arquivo para /myFile.txt irá gravar o arquivo em /sdcard/myFile.txt.

▪ **Eventos:**

GotText (texto)

Evento indicando que o conteúdo do arquivo foi lido.

▪ **Métodos:**

AppendToFile (texto, fileName)

Acrescenta texto ao fim de um arquivo. Cria o arquivo se ele ainda não existir. Consulte o texto de ajuda sob `SaveFile` para obter informações sobre onde os arquivos são gravados.

Delete (fileName)

Exclui um arquivo de armazenamento. Prefixar o nome do arquivo com / para excluir um arquivo específico no cartão SD (por exemplo, /myFile.txt irá apagar o arquivo /sdcard/myFile.txt). Se o nome não começa com a /, em seguida, o arquivo localizado em armazenagem privada do programa será excluído. Iniciando o arquivo com // é um erro porque os arquivos de ativos não podem ser excluído.

ReadFrom (fileName)

Lê texto de um arquivo no armazenamento. Prefixar o nome do arquivo com / para ler a partir de um arquivo específico no cartão SD (por exemplo, /myFile.txt irá ler o arquivo /sdcard/myFile.txt). Para ler bens embalados com uma aplicação (também funciona para o Companion) iniciar o nome do arquivo com // (duas barras). Se um nome de arquivo não começar com uma barra, ele será lido a partir do armazenamento particular do aplicativo (por aplicativos em pacotes) e de / sdcard / ApplInventor / dados para o Companion.

SaveFile (texto, fileName)

Salva texto para um arquivo. Se o nome do arquivo começa com uma barra (/) o arquivo é gravado no sdcard (por exemplo, escrevendo para /myFile.txt irá gravar o arquivo em /sdcard/myFile.txt). Se o nome do arquivo não começar com uma barra, ele será gravado no diretório de dados privado do programa, quando isso não será acessível a outros programas no telefone. Há uma exceção especial para o companheiro AI onde esses arquivos são gravados em / sdcard / ApplInventor / dados para facilitar a depuração. Note-se que este bloco irá substituir um arquivo se ele já existe. Se você quiser adicionar conteúdo em um arquivo de usar o bloco de acréscimo.

• **FusionTablesControl**

Um componente não visível, que comunica com o Google Fusion Tables. Fusion Tables permitem armazenar, compartilhar, consulta e visualizar tabelas de dados; este componente permite que você consulte, crie e modifique estas tabelas.

Este componente utiliza a fusão V1.0 Tabelas API .

Aplicações usando o Fusion Tables deve autenticar com os servidores do Google. Há duas maneiras onde isto pode ser feito. A primeira maneira utiliza apenas uma chave de API que você (desenvolvedor) pode obter. Com esta abordagem os usuários finais também devem entrar para acessar uma tabela de fusão.

A segunda abordagem é a utilização do serviço de autenticação. Com essa abordagem, você cria credenciais e um especial "Conta de serviço E-mail Address" que permite que os usuários finais possam usar seus Fusion Tables sem efetuar login; a conta de serviço autentica todos os acessos.

- **Propriedades:**

- ApiKey

- Sua Chave API do Google. Veja acima para obter detalhes sobre a obtenção de uma chave de API.

- KeyFile

- Especifica o caminho do arquivo de chave privada. Este arquivo de chave é usado para obter acesso à API FusionTables através do serviço de autenticação.

- Question

- A consulta para enviar para o API Fusão Tables.

- Para os formatos de consulta legais e exemplos, consulte o manual de referência v1.0 API FusionTables .

- ServiceAccountEmail

- A conta de serviço do endereço de e-mail usado para autenticação Service.

- UseServiceAuthentication

- Indica se uma conta de serviço deve ser usada para autenticação.

- **Eventos:**

- GotResult (resultado)

- Indica que a consulta Fusion Tables concluiu o processamento e voltou com um resultado. O resultado da consulta será geralmente retornado no formato CSV, e pode ser convertido para a lista usando o formato "lista da tabela csv" ou "lista de linha CSV" blocos.

- **Métodos:**

- ForgetLogin ()

- Esqueça as credenciais de login do utilizador final. Não tem efeito sobre Authentication Service.

GetRows (tableID, colunas)

Obtém todas as linhas de uma tabela Fusão especificado. O campo tableID (obrigatório) é o id da tabela de fusão. O campo de colunas é uma lista separada por vírgula das colunas para recuperar.

GetRowsWithConditions (tableID, colunas, condições)

Obtém todas as linhas de uma tabela de fusão que cumprir determinadas condições. O campo tableID (obrigatório) é o id da tabela de fusão. O campo de colunas é uma lista separada por vírgula das colunas para recuperar. O campo de condições especifica quais linhas para recuperar da tabela (por exemplo, as linhas em que um valor de coluna em particular não é nulo).

InsertRow (tableID, colunas, valores)

Inserir uma linha na tabela Fusão especificada. O campo tableID é o ID da tabela de fusão. O campo de colunas é uma lista das colunas em que para inserir valores separados por vírgulas. O campo de valores especifica os valores a inserir em cada coluna.

SendQuery ()

Enviar a consulta para o servidor Fusion Tables.

- **TinyDB**

TinyDB é um componente não-visível que armazena dados para um aplicativo.

Aplicativos criados com o App Inventor são inicializados a cada vez que são executados. Isto significa que se um aplicativo define o valor de uma variável e, em seguida, o usuário fecha o aplicativo, o valor dessa variável não será lembrado na próxima vez que o aplicativo é executado. Em contraste, TinyDB é um persistente armazenamento de dados para o aplicativo. Os dados armazenados em um TinyDB estarão disponíveis cada vez que o aplicativo é executado. Um exemplo pode ser um jogo que salva a pontuação mais alta e recupera-lo cada vez que o jogo é jogado.

- **Métodos:**

ClearAll()

Limpar todo o armazenamento de dados no TinyDB.

ClearTag (tag)

Desmarque a entrada com a tag dada.

GetTags ()

Retorna uma lista de todas as marcas no TinyDB.

GetValue (tag, valueIfTagNotThere)

Recuperar o valor armazenado sob a tag dada. Se não existe essa tag, em seguida, retornar valueIfTagNotThere.

StoreValue (tag, valueToStore)

Armazenar o valor sob a tag dada. O armazenamento persiste no telefone quando o aplicativo for reiniciado.

- **TinyWebDB**

Componente não visível que se comunica com um serviço Web para armazenar e recuperar informações.

- **Propriedades:**

ServiceURL

O URL para o banco de dados com os quais o componente deve comunicar.

- **Eventos:**

GotValue (tagFromWebDB, valueFromWebDB)

Indica que uma solicitação servidor GetValue foi bem sucedido.

ValueStored ()

Evento indicando que um pedido do servidor StoreValue foi bem sucedido.

WebServiceError (mensagem)

Indica que a comunicação com o serviço Web sinalizou um erro.

- **Métodos:**

GetValue (tag)

Envia uma solicitação para o serviço da Web para obter o valor armazenado sob a tag dada. O serviço Web deve decidir o que retornar se não houver nenhum valor armazenado sob a tag. Este componente aceita tudo é devolvido.

StoreValue (tag, valueToStore)

Envia uma solicitação para o serviço da Web para armazenar o valor dado sob a tag dada.

➤ **Connectivity Components - Componentes de conectividade**

- **ActivityStarter**

Um componente que pode lançar uma atividade usando o método startActivity.

Atividades que podem ser iniciadas incluem:

Começando outro App Inventor para o Android app. Para fazer isso, primeiro descobrir a classe do outro aplicativo fazendo o download do código fonte e usando um explorador de arquivos ou descompacte o utilitário para encontrar um arquivo chamado "youngandroidproject / project.properties". A primeira linha do arquivo vai começar com "principais =" e ser seguida pelo nome da classe; por exemplo, principal = com.gmail.Bitdiddle.Ben.HelloPurr.Screen1 . (. Os primeiros componentes indicam que ele foi criado por Ben.Bitdiddle@gmail.com) Para fazer o seu ActivityStarter lançar este aplicativo, defina as seguintes propriedades:

Iniciando o aplicativo de câmera, definindo as seguintes propriedades:

Ação: android.intent.action.MAIN

ActivityPackage: com.android.camera

ActivityClass: com.android.camera.Camera

Execução de pesquisa na web. Assumindo que o termo que você deseja procurar é "vampire" (sinta livre substituir sua própria escolha), defina as propriedades para:

Ação: android.intent.action.WEB_SEARCH

ExtraKey: query

ExtraValue: vampiro

ActivityPackage: com.google.android.providers.enhancedgooglesearch

ActivityClass: com.google.android.providers.enhancedgooglesearch.Launcher

- **Propriedades:**

Action

Ação da atividade a ser lançada.

ActivityClass

Nome de classe da atividade a ser lançada.

ActivityPackage

Nome do pacote da atividade a ser lançada.

DataUri

URI passado para a atividade a ser lançada.

ExtraKey

Nome da chave de texto passado para a atividade.

ExtraValue

Valor de texto passado para a atividade.

Result

Valor devolvido pela atividade que está sendo iniciada.

ResultName

O nome usado para extrair o resultado retornado da atividade que está sendo iniciada.

ResultType

Escreva informações retornadas da atividade que está sendo iniciada.

ResultUri

URI (ou dados) informações retornadas da atividade que está sendo iniciada.

▪ **Eventos:**

AfterActivity (resultado)

Evento gerado após o retorno de ActivityStarter.

ActivityCanceled ()

Evento levantado se este ActivityStarter retorna porque a atividade foi cancelada.

▪ **Métodos:**

ResolveActivity()

Retorna o nome da atividade que corresponde a este ActivityStarter, ou uma string vazia se nenhuma atividade correspondente pode ser encontrada.

StartActivity ()

Comece a atividade correspondente a esta ActivityStarter.

- **BluetoothClient**

Componente cliente Bluetooth.

- **Propriedades:**

- AddressesAndNames

Os endereços e nomes de dispositivos Bluetooth emparelhados.

- Available

Se Bluetooth está disponível no dispositivo.

- CharacterEncoding

A codificação de caracteres para usar ao envio e recebimento de texto.

- DelimiterByte

O byte delimitador para usar quando passar um número negativo para o parâmetro numberOfBytes ao chamar ReceiveText, ReceiveSignedBytes, ou ReceiveUnsignedBytes.

- Enable

Se Bluetooth está ativado.

- HighByteFirst

Se os números de 2 e 4 bytes devem ser enviados e recebidos com o elevado (ou mais significativo) primeiro byte. Verifique a documentação para o dispositivo com o qual seu aplicativo estará se comunicando para a configuração adequada. Esta é também conhecida como big-endian.

- IsConnected

Se está conectado.

- Secure

Se invoca a SSP (emparelhamento seguro simples), que é suportado em dispositivos com Bluetooth v2.1 ou superior. Ao trabalhar com dispositivos Bluetooth embutidos, esta propriedade pode precisar ser definida como False. Para Android 2,0-2,2, esta configuração de propriedade será ignorada.

- **Métodos:**

- BytesAvailableToReceive ()

Devolve uma estimativa do número de bytes que pode ser recebido sem bloquear.

Connect (endereço)

Conecte-se com o dispositivo Bluetooth com o endereço especificado e o perfil de porta série (SPP). Retorna true se a conexão foi bem sucedida.

ConnectWithUUID (endereço, uuid)

Conecte-se com o dispositivo Bluetooth com o endereço especificado e UUID. Retorna true se a conexão foi bem sucedida.

Disconnect ()

Desligue a partir do dispositivo Bluetooth ligado.

IsDevicePaired (endereço)

Verifica se o dispositivo Bluetooth com o endereço especificado é emparelhado.

ReceiveSigned1ByteNumber ()

Receber um número de 1 byte assinado a partir do dispositivo Bluetooth ligado.

ReceiveSigned2ByteNumber ()

Receber um número de 2 bytes assinado a partir do dispositivo Bluetooth ligado.

ReceiveSigned4ByteNumber ()

Receber um número de 4 bytes assinado a partir do dispositivo Bluetooth ligado.

ReceiveSignedBytes (numberOfBytes)

Receber vários valores de bytes assinados a partir do dispositivo Bluetooth ligado. Se numberOfBytes é inferior a 0, ler até um valor de byte delimitador ser recebido.

ReceiveText (numberOfBytes)

Receber mensagens de texto a partir do dispositivo Bluetooth ligado. Se numberOfBytes é inferior a 0, ler até um valor de byte delimitador ser recebido.

ReceiveUnsigned1ByteNumber ()

Receba um número de 1 byte sem sinal a partir do dispositivo Bluetooth ligado.

ReceiveUnsigned2ByteNumber ()

Receber um número de 2 bytes sem sinal a partir do dispositivo Bluetooth ligado.

ReceiveUnsigned4ByteNumber ()

Receber um número de 4 bytes sem sinal a partir do dispositivo Bluetooth ligado.

ReceiveUnsignedBytes (numberOfBytes)

Receba vários valores de byte sem sinal a partir do dispositivo Bluetooth ligado. Se numberOfBytes é inferior a 0, ler até um valor de byte delimitador ser recebido.

Send1ByteNumber (número)

Envie um número de 1 byte para o dispositivo Bluetooth ligado.

Send2ByteNumber (número)

Envie um número de 2 bytes para o dispositivo Bluetooth ligado.

Send4ByteNumber (número)

Envie um número de 4 bytes para o dispositivo Bluetooth ligado.

SendBytes (lista)

Enviar uma lista de valores de bytes para o dispositivo Bluetooth ligado.

SendText (texto)

Enviar mensagens de texto para o dispositivo Bluetooth ligado.

- **BluetoothServer:**

Componente de servidor Bluetooth.

- **Propriedades:**

Available

Se o Bluetooth está disponível no dispositivo Android.

CharacterEncoding

A codificação de caracteres para usar ao envio e recebimento de texto.

DelimiterByte

O byte delimitador para usar quando passar um número negativo para o parâmetro numberOfBytes ao chamar ReceiveText, ReceiveSignedBytes, ou ReceiveUnsignedBytes.

Enable

Se o Bluetooth está ativado.

HighByteFirst

Se os números de 2 e 4 bytes devem ser enviados e recebidos com o elevado (ou mais significativo) primeiro byte. Verifique a documentação para o dispositivo com o qual seu aplicativo estará se comunicando para a configuração adequada. Esta é também conhecida como big-endian.

IsAccepting

Se este componente BluetoothServer está aceitando uma conexão de entrada.

IsConnected

Se uma conexão Bluetooth foi feita.

▪ **Eventos:**

ConnectionAccepted ()

Indica que uma conexão Bluetooth foi aceita.

▪ **Métodos:**

AcceptConnection (serviceName)

Aceitar uma conexão de entrada com o perfil de porta série (SPP).

AcceptConnectionWithUUID (serviceName, uuid)

Aceitar uma conexão de entrada com um UUID específico.

BytesAvailableToReceive ()

Devolve uma estimativa do número de bytes que pode ser recebido sem bloquear.

Disconnect ()

Desligue a partir do dispositivo Bluetooth ligado.

ReceiveSigned1ByteNumber ()

Receber um número de 1 byte assinado a partir do dispositivo Bluetooth ligado.

ReceiveSigned2ByteNumber ()

Receber um número de 2 bytes assinado a partir do dispositivo Bluetooth ligado.

ReceiveSigned4ByteNumber ()

Receber um número de 4 bytes assinado a partir do dispositivo Bluetooth ligado.

ReceiveSignedBytes (numberOfBytes)

Receber vários valores de bytes assinados a partir do dispositivo Bluetooth ligado. Se numberOfBytes é inferior a 0, ler até um valor de byte delimitador ser recebido.

ReceiveText (numberOfBytes)

Receber mensagens de texto a partir do dispositivo Bluetooth ligado. Se numberOfBytes é inferior a 0, ler até um valor de byte delimitador ser recebido.

ReceiveUnsigned1ByteNumber ()

Receba um número de 1 byte sem sinal a partir do dispositivo Bluetooth ligado.

ReceiveUnsigned2ByteNumber ()

Receber um número de 2 bytes sem sinal a partir do dispositivo Bluetooth ligado.

ReceiveUnsigned4ByteNumber ()

Receber um número de 4 bytes sem sinal a partir do dispositivo Bluetooth ligado.

ReceiveUnsignedBytes (numberOfBytes)

Receba vários valores de byte sem sinal a partir do dispositivo Bluetooth ligado. Se numberOfBytes é inferior a 0, ler até um valor de byte delimitador ser recebido.

Send1ByteNumber (número)

Envie um número de 1 byte para o dispositivo Bluetooth ligado.

Send2ByteNumber (número)

Envie um número de 2 bytes para o dispositivo Bluetooth ligado.

Send4ByteNumber (número)

Envie um número de 4 bytes para o dispositivo Bluetooth ligado.

SendBytes (lista)

Enviar uma lista de valores de bytes para o dispositivo Bluetooth ligado.

SendText (texto)

Enviar mensagens de texto para o dispositivo Bluetooth ligado.

StopAccepting ()

Parar de aceitar uma conexão de entrada.

- **Web - Rede**

Componente não visível que fornece funções para HTTP GET, POST, PUT e DELETE solicitações.

- **Propriedades:**

AllowCookies

Se os cookies de uma resposta devem ser guardados e utilizados em pedidos subsequentes. Os cookies são suportados apenas na versão Android 2.3 ou superior.

RequestHeaders

Os cabeçalhos de solicitação, como uma lista de sub-listas de dois elementos. O primeiro elemento de cada sub-lista representa o nome do campo de cabeçalho de solicitação. O segundo elemento de cada sub-lista representa os valores do campo de cabeçalho do pedido, ou um valor único ou uma lista que contém vários valores.

Responsefilename

O nome do arquivo em que a resposta deve ser salva. Se SaveResponse é verdadeiro e responsefilename está vazia, então um novo nome de arquivo será gerado.

SaveResponse

Se a resposta deve ser salvo em um arquivo.

Url

O URL para a solicitação da web.

- **Eventos:**

GotFile (url, responseCode, responseType, fileName)

Evento indicando que uma solicitação foi concluída.

GotText (url, responseCode, responseType, responseContent)

Evento indicando que uma solicitação foi concluída.

▪ **Métodos:**

BuildRequestData (lista)

Converte uma lista de sub-lists de dois elementos, representando pares nome e valor, para uma cadeia formatada como application / x-www-form-urlencoded tipo de mídia, adequado para passar para PostText.

ClearCookies ()

Limpa todos os cookies para este componente Web.

Delete ()

Realiza uma solicitação HTTP DELETE usando a propriedade Url e recupera a resposta.

Get()

Executa uma solicitação HTTP GET usando a propriedade Url e recupera a resposta.

HtmlTextDecode (htmlText)

Decodifica o dado valor de texto HTML. Entidades de HTML como & amp ;, & lt ;, & gt ;, & apos ;, e & quot; são alterados para &, <,>, ' , e ". Entidades como o & # xhhhh, e & # nnnn são alteradas para os caracteres apropriados.

JsonTextDecode (jsonText)

Decodifica o valor dado JSON codificado para produzir um valor AppInventor correspondente. Uma lista JSON [x, y, z] decodifica a uma lista (xyz), um objeto JSON com o nome A e B valor, (indicado como A: B entre chaves) descodifica a uma lista ((AB)), que é, uma lista contendo a lista de dois elementos (AB).

PostFile (caminho)

Executa uma solicitação HTTP POST usando a propriedade Url e dados do arquivo especificado.

PostText (texto)

Executa uma solicitação HTTP POST usando a propriedade Url e o texto especificado.

Os caracteres do texto são codificados usando codificação UTF-8.

PostTextWithEncoding (texto, codificação)

Executa uma solicitação HTTP POST usando a propriedade Url e o texto especificado.

Os caracteres do texto são codificados usando a codificação dada.

PutFile (caminho)

Executa uma solicitação HTTP PUT usando a propriedade Url e dados do arquivo especificado.

PutText (texto)

Executa uma solicitação HTTP PUT usando a propriedade Url e o texto especificado.

Os caracteres do texto são codificados usando codificação UTF-8.

PutTextWithEncoding (texto, codificação)

Executa uma solicitação HTTP PUT usando a propriedade Url e o texto especificado.

Os caracteres do texto são codificados usando a codificação dada.

UriEncode (texto)

Codifica o valor de texto determinado de modo que possa ser usada em um URL.

XMLTextDecode (XmlText)

Decodifica a string XML dada para produzir uma estrutura de lista. Consulte a documentação do App Inventor em "Outros temas, notas e detalhes" para obter informações.

Blocks – Blocos

A seguir descreveremos um conjunto de blocos utilizados na programação para app inventor. Estes blocos não são específicos de componentes, mas são muito importantes para o desenvolvimento de uma aplicação completa.

➤ **Control – Controle**

- If – Se

Neste bloco uma condição será avaliada, caso o programa declare esta condição como verdadeira, o conjunto de blocos dentro do 'If-Se' será executado.

- For each number from - Para cada número de

Executa os blocos na seção "do" para cada valor numérico no intervalo do início ao fim, passando um valor de cada vez. Use o nome da variável dada para se referir ao valor atual.

- For each item in list - Para cada item na lista

Executa os blocos na seção "fazer" para cada item na lista. Use dado nome de variável para se referir ao atual item da lista.

- While - Enquanto

Executa os blocos na seção "fazer" enquanto o teste é verdadeiro.

- If / Then / Else – Se / Então / Se não

Se a condição que está sendo testada for verdade, retornar o resultado da avaliação da expressão ligada ao slot 'then-return' caso contrário retornar o resultado da avaliação da expressão ligada ao slot 'else-return', no máximo, uma das expressões do entalhe retorno será ser avaliada.

- Do - Faz

Executa os blocos em "fazer" e retorna um comunicado. Útil se você precisa executar um procedimento antes de retornar um valor a variável.

- Evaluate but ignore result - Avaliar mas ignorar o resultado

Executa o bloco de código ligado e ignora o valor de retorno (se houver). Útil se necessidade de chamar um procedimento com um valor de retorno, mas não precisa do valor.

- Open another screen screenName - Abra outra ScreenName tela

Abre uma nova tela em um aplicativo de tela múltipla.

- Open another screen with start value – Abra outa tela com valor inicial

Abre uma nova tela em um aplicativo de tela múltipla e passa o valor inicial para a tela.

- Get Star Value - Obter Valor Inicial

Retorna os valores que foram transmitidos a esta tela quando foi iniciado, tipicamente por uma outra tela em um aplicativo de tela múltipla. Se nenhum valor foi aprovado, retorna o texto vazio.

- Close Screen - Fechar tela

Feche a tela atual.

- Close screen with value - Fechar tela com valor

Fechar tela atual e retorna um resultado para a tela que abriu que este.

- Close application - Fechar pedido

Feche todas as telas neste app e para o app.

- Get plain start next - Se planície começar no próximo

Retorna o texto simples que foi passado para esta tela quando foi iniciado por outro app. Se nenhum valor foi aprovada, retorna o texto vazio. Para vários aplicativos em tela, use obter o valor iniciar em vez de se começar.

➤ **Logic – Logico**

- True – Verdadeiro

Retorna a variável booleana verdadeiro.

- False – Falso

Retorna a variável booleana falso.

- Not – Não

Retorna verdadeiro se a variável for falsa e vice-versa.

- Equal - Igual

Testa se duas coisas são iguais. As coisas que estão sendo comparados podem ser quaisquer coisas, não apenas números. Os números são considerados para ser igual ao seu impresso a partir de cadeias, por exemplo, o número 0 é igual ao texto "0". Além disso, duas cadeias que representam números são iguais se os números são iguais, por exemplo "1" é igual a "01".

- And - E

Retorna verdadeiro se todas as entradas são verdadeiras.

- Or – Ou

Retorna verdadeiro se ao menos uma entrada for verdadeira.

➤ **Math – Matemática**

- Number - Número

Relate o número mostrado.

- Equal - Igual

Testa se dois valores indicados são iguais. Se assim for, ele retorna true; caso contrário, retorna falso. Este é o mesmo que o bloco na gaveta lógica e é repetido aqui por conveniência. Os argumentos podem ser qualquer coisa, não apenas números.

- Sum - Soma

Retorna a soma dos dois ou mais números.

- Decrease - Diminuir

Devolver a diferença dos dois números.

- Multiplication - Multiplicação

Devolver o produto dos dois ou mais números.

- Division – Divisão

Retorna o quociente entre 2 números.

- Power – Potencia

Retorna o primeiro número elevado à potência do segundo número.

- Random integer - Inteiro aleatório

Retorna um valor inteiro aleatório entre os valores indicados, inclusive. A ordem dos argumentos não importa.

- Random fraction – Fração aleatória

Retorna um valor aleatório entre 0 e 1.

- Random set seed to - Semente conjunto aleatório de

Utilize este bloco para gerar sequências repetitivas de números aleatórios. Você pode gerar a mesma sequência de números aleatórios, primeiro chamando semente conjunto aleatório com o mesmo valor. Isso é útil para programas de testes que envolvem valores aleatórios.

- Min / Max – Minimo / Maximo

Retorna o menor de um determinado conjunto de números.

Retorna o maior de um determinado conjunto de números.

- Square root (Sqrt) - Raiz quadrada

Retorna a raiz quadrada de um número.

- Absolete - Absoluto

Retorna o valor absoluto do número fornecido.

- Neg - negar

Devolve o negativo do número fornecido.

- Round - Arredondar

Arredonda o número fornecido para o número inteiro mais próximo e retorna o resultado. Se o número está a meio caminho entre dois números inteiros, arredondado para o número inteiro mesmo.

- Ceiling - teto

Retorna o menor número inteiro que é maior do que ou igual ao número dado.

- Floor - Andar

Calcula o maior inteiro que é menor ou igual ao número dado.

- Modulo - Modulo

Modulo (a, b) é o mesmo que o resto (a, b) quando a e b são positivos. Mais geralmente, módulo (a, b) é definido por qualquer um e b, de modo que $(\text{Floor}(a / b) \times b) + \text{módulo}(a, b) = a$. Por exemplo, módulo (11, 5) = 1, módulo (-11, 5) = 4, módulo (11, -5) = -4, módulo (-11, -5) = -1. Modulo (a, b) sempre tem o

mesmo sinal que b, enquanto a parte restante (a, b) sempre tem o mesmo sinal que a.

- Sin - Seno

Retorna o seno do número fornecido em graus.

- Cos - Cosseno

Retorna o cosseno do número fornecido em graus.

- Tan - Tangente

Retorna a tangente do número fornecido em graus.

- atan2 – Arco duplo

Retorna o arco tangente de y / x, y dado e x.

- Convets to degrees radian - Converter radianos em graus

Retorna o valor de grau no intervalo (0, 360) correspondente ao seu argumento radianos.

- Format as decimal - Como formato decimal

Formatar um número como um número decimal com um determinado número de casas após o ponto decimal. O número de lugares deve ser um número inteiro não negativo. O resultado é produzido por arredondamento do número (se houvesse muitos lugares) ou pela adição de zeros à direita (se não eram muito poucos).

- Is a number? - É um número?

Retorna verdadeiro se o objeto fornecido é um número, e false de outra forma.

- Convert number base 10 to hex - Converte um número base 10 para hexadecimal

Leva um inteiro positivo na base 10 e retorna a sequência de caracteres que representa o número em hexadecimal.

➤ **Text – Texto**

Contém uma sequência de texto.

- Join - Junte-se

Acrescenta o segundo dada sequência de caracteres para o primeiro. Para esta operação, e todas as operações de texto, os números podem ser tratados como texto. Por exemplo, usando um bloco de junção para juntar 1 mais 1 e 2 vezes resulta em 3 26 (2 juntou-se com 6).

- Length - comprimento

Retorna o comprimento da string dada.

- Is empty - Está vazio

Retorna true se o comprimento do texto é 0, caso contrário false.

- Compare texts – Comparar textos

- <

Informa se o primeiro argumento sequência de texto é alfabeticamente menor que a segunda cadeia de texto. Se duas cordas começam com os mesmos caracteres, a cadeia mais curta é considerada menor. Caracteres maiúsculos precedem os minúsculos.

- **text =**

Informa se as sequências de texto são idênticas, ou seja, têm os mesmos caracteres na mesma ordem.

- **text>**

Informa se o primeiro argumento sequência de texto é alfabeticamente maior que o segundo texto. Se duas cordas começam com os mesmos caracteres, a cadeia mais curta é considerada menor. Caracteres maiúsculos precedem os minúsculos.

- Trim - Aparar

Retorna uma cópia do seu texto com todos os espaços esquerda ou à direita removidos.

- Uppcase – Maiúsculas

Retorna uma cópia do seu argumento sequência de texto convertidos em maiúsculas.

- Start at - Começa às

Retorna a posição do caractere onde o primeiro caractere da peça aparece pela primeira vez no texto, ou 0 se não estiver presente. Por exemplo, a localização de “ana” em havana banana é 4.

- Contains - Contém

Retorna true se peça aparece no texto, caso contrário, retorna false.

- Split – Dividido

- **Split at first - dividir em primeiro**

Divide o texto dado em duas partes usando o local da primeira ocorrência de pelo como o ponto de divisão, e retorna uma lista de dois itens que consiste na peça antes de o ponto de divisão e a peça após o ponto de divisão. Splitting maçã, banana, cereja, comida de cachorro com uma vírgula como o ponto de divisão retorna uma lista de dois itens: a primeira é a maçã de texto ea segunda é a banana texto, cereja, comida de cachorro. Note que a vírgula após a maçã não aparecer no resultado, porque esse é o ponto de divisão.

- **Split at first of any - dividir a primeira de qualquer**

Divide o texto dado em uma lista de dois itens, usando o local de qualquer item na lista como o ponto de divisão.

- **Split - Dividido**

Divide texto em pedaços utilizando pelo que os pontos de divisão e produz uma lista dos resultados. Dividindo um, dois, três, quatro em, (vírgula) retorna a lista (um, dois, três, quatro). Dividindo uma batata, batata-dois, três batatas, quatro no -batata, retorna a lista (um, dois, três, quatro).

- **split at any - dividir a qualquer**

Divide o texto dado em uma lista, usando qualquer um dos itens em que o ponto de divisão, e retorna uma lista de resultados. Maça divisão, banana, cereja, dogfood com pelo como a lista de dois elementos cujo primeiro item é uma vírgula, e cujo segundo item é rry retorna uma lista de quatro itens: (maça ser de banana che dogfood).

- *split at spaces - dividida em espaços*

Divide o texto dado em qualquer ocorrência de um espaço, produzindo uma lista de peças.

- *Segment - Segmento*

Extraí parte do texto começando na posição início e continuando por caracteres de comprimento.

- *Replace all - Substitua tudo*

Retorna uma nova sequência de texto obtida pela substituição de todas as ocorrências da substring com a substituição.

- *Obfuscated text - Texto ofuscado*

Produz texto, como um bloco de texto. A diferença é que o texto não é facilmente detectável que incluem informações confidenciais, por exemplo, chaves de API. Aviso: Este prevê apenas muito baixo segurança contra adversários experientes.

➤ **Lists – Listas**

- *Create a empty list - Criar uma lista vazia*

Criar uma lista com todos os números de itens.

- *Make a list - Faça uma lista*

Cria uma lista dos blocos de dados. Se você não fornecer quaisquer argumentos, isso cria uma lista vazia, que você pode adicionar elementos à tarde.

- *Add items to list - Adicionar itens para a lista*

Adiciona os itens dado ao final da lista. A diferença entre este e anexar a lista é que acrescentar à lista leva os itens a serem acrescentados como uma única lista, enquanto adicionar itens à lista tem os itens como argumentos individuais.

- *Is in list? - Está na lista?*

Se coisa é um dos elementos da lista, retorna true; caso contrário, retorna falso. Observe que, se uma lista contém sublists, os membros das sublists não

são eles próprios membros da lista. Por exemplo, os membros da lista (1 2 (3 4)) são 1, 2, e a lista (3 e 4); 3 e 4 não são eles próprios membros da lista.

- Length of list - Comprimento da lista

Retorna o número de itens na lista.

- Pick random item - Pegar item aleatório

Escolhas um item de forma aleatória a partir da lista.

- Index in list thing - Índice coisa em lista

Encontre a posição de coisa na lista. Se não estiver na lista, retornar 0.

- Select list item - Selecione o item lista

Seleciona o item no índice indicado na lista dada. O primeiro item da lista é no índice 1.

- Insert list item - Inserir item da lista

Insere um item em uma lista na posição especificada.

- Replace list item - Substituir item da lista

Insere substituição na lista dada na posição de índice. O produto anterior em que a posição é removida.

- Remove list item - Remover item da lista

Remove o item na posição especificada na lista.

- Append to list - Acrescentar à lista

Adiciona os itens da segunda lista, para o fim da primeira lista.

- Copy list - Cópia da lista

Faz uma cópia de uma lista, incluindo a cópia de todos os sub-listas.

- Is a list? - É uma lista?

Se coisa é uma lista, retorna true; caso contrário, retorna falso.

- List to csv row list - Lista de linha csv lista

Interpreta a lista como uma linha de uma tabela e retorna um arquivo CSV (valores separados por vírgula) texto que representa a linha. Cada item na lista de linha é considerado como um campo, e é citado com aspas no texto CSV resultante. Os itens são separados por vírgulas. O texto linha retornada não tem um separador de linha no final.

- List to csv table list - Lista à mesa csv lista

Interpreta a lista como uma tabela em formato de linha principal e retorna um arquivo CSV (valores separados por vírgula) texto que representa a mesa. Cada item da lista em si deve ser uma lista representando uma linha da tabela de CSV. Cada item na lista de linha é considerado como um campo, e é citado com aspas no texto CSV resultante. No texto retornado, artigos em linhas são separados por vírgulas e linhas são separados por CRLF (\r\n).

- List from csv row text - Lista de linha csv text

Analisa um texto como um arquivo CSV (valores separados por vírgula) linha formatada para produzir uma lista de campos. É um erro para o texto da linha para conter nova linha unescaped campos dentro (de forma eficaz, várias linhas). Não há problema para o texto linha para terminar em uma nova linha única ou CRLF.

- List from csv table text - Lista da tabela csv text

Analisa um texto como um arquivo CSV (valores separados por vírgula) formatado tabela para produzir uma lista de linhas, cada uma das quais é uma lista de campos. As linhas podem ser separadas por novas linhas (\n) ou CRLF (\r\n).

- Lookup in pairs - Pesquisa em pares

Espera-se uma determinada chave em uma lista de (chave, valor) pares e retorna o valor associado. Retorna um padrão especificado se não é emparelhar com essa chave. Cada par muito ser uma lista de exatamente dois itens.

➤ Variables – Variáveis

- Initialize global name to - Inicializar nome global para

Este bloco é usado para criar variáveis globais. Leva em qualquer tipo de valor como um argumento. Ao clicar no nome vai mudar o nome dessa variável

global. As variáveis globais são utilizadas em todos os procedimentos ou eventos de modo que este bloco vai ficar sozinho.

As variáveis globais podem ser alteradas enquanto um aplicativo está sendo executado e pode ser referido e mudado a partir de qualquer parte do aplicativo, mesmo dentro dos procedimentos e manipuladores de eventos. Você pode renomear este bloco em qualquer momento e quaisquer blocos associados referentes ao nome antigo será atualizado automaticamente.

- Get - Obter

Este bloco fornece uma maneira de obter quaisquer variáveis que você possa ter criado.

- Set to - Definido como

Este bloco segue as mesmas regras que get. Apenas variáveis no escopo estarão disponíveis na lista suspensa. Uma vez que uma variável v é selecionada, o usuário pode anexar um novo bloco e dar novo valor a v.

- Initialize Local name to - in (do) - Nome local para inicializar - em (fazer)

Este bloco é um modificador que permite que você crie novas variáveis que são usadas apenas no processo de executar na parte DO do bloco. Desta forma, todas as variáveis do presente procedimento iram iniciar todos com o mesmo valor cada vez que o procedimento é executado. NOTA: Este bloco difere do bloco descrito abaixo porque é um bloco de DO. Você pode anexar declarações a ele. Demonstrações fazer as coisas. É por isso que este bloco tem espaço dentro de blocos de instrução para ser anexado.

Você pode renomear as variáveis de participação neste bloco, em qualquer momento e qualquer dos blocos correspondentes em outros lugares em seu programa que se referem ao nome antigo será atualizado automaticamente.

- Initialize Local name to - in (return) - Nome local para inicializar - in (regresso)

Este bloco é um modificador que permite que você crie novas variáveis que são usadas apenas no processo de executar na parte de retorno do bloco. Desta forma, todas as variáveis do presente procedimento iram iniciar todos com o mesmo valor cada vez que o procedimento é executado. NOTA: Este bloco difere do bloco descrito acima, porque é um bloco de retorno. Você pode anexar expressões a ele. Expressões retornar um valor. É por isso que este bloco tem uma tomada para ligar expressões.

Você pode renomear as variáveis de participação neste bloco, em qualquer momento e qualquer dos blocos correspondentes em outros lugares em seu programa que se referem ao nome antigo será atualizado automaticamente.

➤ **Procedures – Procedimentos**

O que é um procedimento?

Um procedimento é um conjunto de instruções que executam uma tarefa ou tarefas específicas. Ele também pode ser chamado de uma função. A receita para o pão de banana é um exemplo de um procedimento. O padeiro deve seguir as instruções passo-a-passo para produzir o pão.

Nós usamos procedimentos na App Inventor para criar novos blocos que podemos usar repetidamente e ocupam menos espaço do que todos os blocos utilizados no processo inicial. Se estamos usando os mesmos conjuntos de blocos mais de uma vez, estes blocos são chamados redundante.

O que é um procedimento em App Inventor?

Um procedimento App Inventor recolhe uma sequência de blocos em conjunto num grupo. Você pode então usar a sequência de blocos repetidamente chamando o procedimento. Se o procedimento tem argumentos, você especificar os argumentos usando blocos do nome. Quando você cria um procedimento, o App Inventor gera automaticamente um bloqueio de chamada e coloca-lo na gaveta minhas definições. Você usa o bloqueio de chamadas para chamar o procedimento.

Quando você cria um novo bloco de procedimento, o App Inventor escolhe um nome exclusivo automaticamente. Você pode clicar sobre o nome e tipo de mudá-lo. Nomes de procedimentos em um aplicativo devem ser exclusivos. App Inventor não vai deixar você definir dois procedimentos no mesmo aplicativo com o mesmo nome. Você pode renomear um procedimento a qualquer momento, enquanto você está construindo o aplicativo, alterando a etiqueta no bloco. App Inventor irá renomear automaticamente os blocos de chamada associados ao jogo.

Vamos construir um procedimento para fazer o trabalho dos blocos de códigos redundantes. Na App Inventor, você define um procedimento de uma forma semelhante a como você definir variáveis. Da gaveta Procedimentos, arrastar para fora ou um procedimento que bloquear ou ao procedimento de bloqueio de retorno. Utilize este último se o procedimento deve calcular algum valor e devolvê-lo. Depois de arrastar um bloco procedimento, você pode mudar o seu nome do "procedimento" padrão clicando a palavra "processo" e digitar o novo nome.

No tutorial MoleMash, você cria um procedimento chamado MoveMole para mover a toupeira para um local aleatório na tela.

Depois de criar um procedimento, um bloco é colocado na gaveta Procedimentos embutido que permite que você ligue para o seu procedimento.

Detalhes adicionais sobre blocos de procedimento do App Inventor podem ser encontradas na página Procedimento Blocks.

O que é um argumento?

Um argumento é uma entrada para o nosso procedimento. Alguns procedimentos exigem saber alguns pedaços de informações que mudam a forma como o processo é executado. Vejamos o exemplo de banana pão receita. Às vezes, o padeiro pode querer fazer dois pães de banana. Em seguida, o padeiro seria o dobro da quantidade de cada ingrediente chamado para na receita. Este casal ou 2 é um argumento para o nosso procedimento uma vez que este muda a quantidade de nossos ingredientes que se misturam. Quando você cria um procedimento, você pode usar o botão modificador para adicionar argumentos.

Ao passar o mouse sobre um argumento, você vai ver um get e set bloco aparecer. Arraste esses blocos em sua tela para usá-los.

Este procedimento leva dois argumentos: x e y. Em seguida, ele retorna o resultado da adição de x para y.