



**Você pode
aprender
tudo que
quiser.**



MASTERTECH



Estação Hack
São Paulo
from **facebook**

Aprenda a Programar em um Final de Semana



Bem vindo ao Aprenda a Programar em um Final de Semana! Este é um desafio que poucas pessoas aceitam, mas tenho certeza que se você topou fazer essa atividade com a gente é que chegou o seu momento de **Aprender a Programar**.

Neste Final de Semana cheio de atividades, você será desafiado várias vezes a entender conceitos nunca visto antes, e como uma língua que não falou, se deparar com notações que não imaginava que existiam. Não se preocupe! Todas as pessoas ao seu lado também nunca viram isso.

Quando criamos o Aprenda a Programar em um Final de Semana, nosso objetivo nunca foi formar programadores e programadoras profissionais, mas sim mostrar que você pode aprender a aprender de uma maneira diferente. Por isso utilizamos nossos Ninjas que são ex-alunos Mastertech e passaram **exatamente** pelas mesmas dificuldades que vocês. Eles sempre se surpreendem ao ver quantas pessoas estão com vontade e coragem de aprender essa linguagem nova.

Por isso, seja muito bem vindo, e traga com você muita vontade para Aprender a Programar em um Final de Semana junto com a Mastertech!

Felipe Barreiros

CEO Mastertech

Índice

Aprenda a Programar em um Final de Semana	2
Índice	3
Introdução à Programação	4
Programando o site no seu computador	6
Buscando temas/templates já prontos	6
Estúdio de Programação	7
Publicando seu site no GitHub	8
Criando uma conta no GitHub	8
O que é um repositório	8
Criando um repositório	8
Subindo os arquivos que você criou	9
Javascript, sua primeira linguagem de programação	10
Missão 1: Mandar uma mensagem para o usuário	11
Missão 2: Pegar valores que o usuário digitou	11
Missão 3: Fazer uma calculadora que soma valores.	12
Missão 4: Fazer uma calculadora, mais espertinha com condições.	13
Introdução ao AppInventor	16
Como funciona?	16
Criando seu primeiro Projeto no Ionic Creator	16
Missão 1: Criando seu primeiro Aplicativo	17
Missão 2: Criando uma calculadora simples	19
Como publicar uma app na loja :)	22
Como criar uma conta de desenvolvedor no Google Play	22
Como publicar um novo aplicativo	23

Introdução à Programação

Nesta primeira parte o objetivo é discorrer sobre os paradigmas de programação, mostrar as diferentes áreas de programação e iniciar a construção de nosso próprio site.

Back-End vs Front-End	<p>Front-End Celulares e WebApps - tudo que é visto no seu celular ou desktop Facebook, Waze, Google, Instagram...</p> <p>Back-End Bastidores do seu site, onde a inteligência seu aplicativo roda. Onde o Waze faz o cálculo das rotas, onde o Facebook identifica quais posts vai te mostrar.</p> <p>Banco de dados Onde todos os dados são armazenados. Pra quem trabalha com Excel, é justamente isso em uma escala maior. São bilhões de linhas e milhares de colunas que armazenam os dados. Exemplos:<ul style="list-style-type: none">- Todas as ruas do Google Maps- Todas as fotos já publicadas no Facebook- Todas as informações de trânsito no momento</p>
Caminho da URL	<p>O que acontece quando você abre o browser e digita um endereço, como google.com.br? O que acontece? Pra onde isso vai?</p> <p>Resolvendo com DNS Sua máquina precisa pegar esse endereço e traduzir em IP. Alguns já devem ter ouvido falar disso, mas o IP é como se fosse o CEP e o numero da sua casa. Com o IP eu consigo te identificar em qualquer lugar que você esteja no mundo.</p> <p>Esse é o poder do IP.</p> <p>Buscando o IP Depois de ter recebido o IP é hora de encontrar o servidor do Google. Como o servidor do Google não está na sua casa, o seu computador manda para o roteador. E com uma escala um pouco maior, existe um roteador da NET, e outro roteador da Embratel, e outro roteador que vai te jogar e te levar até o lugar físico onde a máquina do Google está.</p> <p>Olha que maluco: a internet não é nada mais do que vários computadores conectados uns aos outros. Inclusive os servidores do Google tem menor capacidade que o notebook que está na sua mesa.</p> <p>E tudo que vai, tem que voltar. Por isso que esse pacote volta e abre pra você o Google. É UM MILAGRE!!!!</p>

Ferramentas	<p>Visual Studio Code É um editor de texto que funciona como a melhor ajuda para quem está programando. Há várias funcionalidades presentes no vscode, até mesmo um terminal. Há outras opções de editores, como o Sublime Text, Atom, Brackets que desempenham a mesma função só que cada um tem um diferencial seja de interface ou comandos.</p> <p>GitHub Caso algum avalanche ataque esse prédio ou o Godzilla quebre os vidros e roube sua máquina, todo o código que você desenvolver vai estar salvo no GitHub. É como uma nuvem para armazenar seus códigos. Inclusive se vocês conhecem algum programador, e perguntar qual é seu GitHub, a galera vai pirar. Esse é o lugar é como um portfólio do desenvolvedor.</p> <p>Inspeccionador de elementos (chrome) Sabe quando você recebe ou olha um meme ou uma fake news e se pergunta como fizeram? Então boa parte pode ser respondida pelo uso no Inspect. Como viram em aula, através dele é possível olhar todo o código fonte de um site, ver suas requisições, e até mesmo alterar a versão local do site carregado (momento de zoar o amiguinho). Tirando isso, esta ferramenta auxilia muito na hora de estudar pois passando o mouse dá para ver qual é a div, o css aplicado....</p>
HTML, CSS e JavaScript	<p>Vamos lembrar do exemplo do Marty McFly:</p> <p>HTML é o corpo Ele dá a estrutura e marca a posição de cada item, seja ele o título (heading), parágrafo (paragraph), links (anchor) ou qualquer outro item do HTML. Os itens do HTML são representados por símbolos de menor e maior e sempre precisam ser fechados, como o exemplo abaixo:</p> <pre><h1>Esse é um título</h1> <p>Esse é um parágrafo</p></pre> <p>CSS é o estilo Responsável por deixar o site mais bonito e mais utilizável, o papel do CSS na construção do site é melhorar a usabilidade do site como um todo. A notação do CSS é diferente do HTML e segue como o exemplo abaixo:</p> <pre>p { background-color:red; color: green; font-family:Arial; }</pre> <p>JavaScript é a linguagem de programação Responsável por trazer dinamicamente informações do servidor ou dar animação para a página, o JavaScript pode ser utilizado também para fazer cálculos e executar a inteligência do seu site ou aplicativo.</p>

Codepen.io	<p>HORA DE SEREM DESENVOLVEDORES! (alternativa)</p> <p>Acesse http://codepen.io e clique em New Pen</p> <p>Nesta ferramenta tem 3 diferentes áreas apresentadas: HTML, CSS e JavaScript. Fique à vontade para dar seus primeiros passos no processo de desenvolvimento de um site utilizando os conceitos abordados.</p>
Inspecionar no Google Chrome	<p>Utilizando o Google Chrome você conseguirá ver a estrutura de todos os sites no mundo e ver como eles foram construídos.</p> <p>Esse é uma das principais vantagens de construir sites Web. Eles são completamente abertos e podemos aprender muito com eles.</p> <p>Ao acessar qualquer site, clique com o botão direito e vá em Inspecionar.</p> <p>Abaixo estão alguns sites para você fazer isso:</p> <ul style="list-style-type: none"> - https://linux.ime.usp.br/~cachutti - http://felipebarreiros.com - http://globo.com <p>Repare e procure os seguintes itens:</p> <ul style="list-style-type: none"> - h1 (Título 1) - h2 (Título 2) - img (Imagen) - ol ou ul (Listas Ordenadas ou Listas Não-Ordenadas) - li (Itens de Lista)

Estúdio de Programação

Hora de arregaçar as mangas e colocar-se de corpo e alma para o código. Muito do que vimos pela manhã será colocada em prova nesta tarde em um atividade que chamamos de **Estúdio**. Junto com os Ninjas vocês serão desafiados a criar o próprio site de vocês e dar forma àquilo que fará parte da sua apresentação final.

Abaixo estão algumas tags HTML que são importantes para esta atividade:

<code>Site Mastertech</code>	Âncora Essa tag de âncora é usada para redirecionar o usuário para outro site ou outra área da sua página. No exemplo ao lado, o usuário está sendo redirecionado para a página http://mastertech.tech
<code><p></code>	Parágrafo Exemplo simples de um parágrafo em HTML. Todo texto longo deve estar dentro de um e mantém a coesão entre o documento todo.
<code> Item 1 Item 2 Item 3 </code>	Lista Ordenada Essa lista ordenada irá apresentar os itens da maneira abaixo: <ol style="list-style-type: none">1. Item 12. Item 23. Item 3
<code> Item 1 Item 2 Item 3 </code>	Lista Não-Ordenada (Unordered List) Essa lista não-ordenada irá apresentar os itens da maneira abaixo: <ul style="list-style-type: none">• Item 1• Item 2• Item 3
<code></code>	Imagen Essa tag HTML exibe uma imagem localizada na pasta img e já configura o tamanho total dela com altura 100px e largura 300px
<code><h1>Mastertech</h1> <h2>Mastertech</h2> <h3>Mastertech</h3> <h1>Mastertech</h1> <h2>Mastertech</h2></code>	Títulos Os títulos são muito importantes para o usuário final e o mecanismo de busca do Google identificar quais são os principais recursos dentro do seu site. É por eles que muitos dos usuários se guiam já que eles definem de forma hierárquica de apresentação das informações dentro do site. Não há um limite de identificação de Títulos no HTML, mas é muito raro ver páginas utilizando tags depois de <h6>
<code>Essa mensagem é importante</code>	Strong Essa tag é utilizada para dar destaque a uma mensagem no texto. Antigamente era utilizada a tag (que ainda funciona) mas foi descontinuada já que diz mais claramente a função dela.
<code><hr></code>	Linha separadora Quando o <hr> é colocado em no HTML, é criada uma linha separadora na página. Muito utilizada como mecanismo de Design Visual para separar informações.
<code>
</code>	Quebra de Linha (Break Line) Quebra a linha, divide o texto em duas linhas. Muito utilizado dentro de parágrafos ou entre tags HTML

Publicando seu site no GitHub



O GitHub é a principal plataforma mundial de gerenciamento de código e a plataforma social de colaboração entre programadores. Lá você poderá encontrar diversos projetos de grandes empresas como Google, Microsoft e Facebook.

O GitHub é utilizado para compartilhamento de código e unir programadores que desejam trabalhar em conjunto em projetos específicos. Como a partir de agora somos programadores, vamos também usar essa ferramenta!

Criando uma conta no GitHub

1. Acesse github.com
2. Clique em **Sign up**
3. Siga os passos da página para criar seu login
4. Confirme seu endereço de e-mail clicando no link enviado para sua caixa de entrada
5. Crie seu primeiro repositório

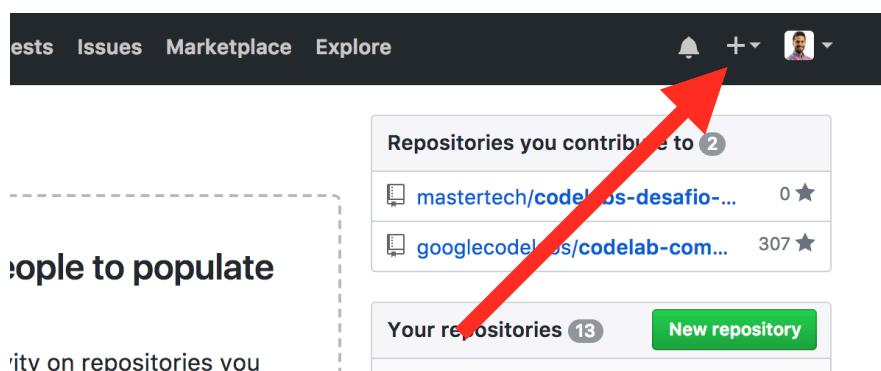
O que é um repositório

Repositório é como se fosse uma pasta disponível na nuvem que pode ser acessado de qualquer lugar e irá deixar seu código seguro. Além de deixar seu código seguro caso aconteça alguma coisa com seu computador, o repositório é capaz de salvar diferentes versões de seu código.

O versionamento do seu código é importante porque se você alterou algo e queria voltar a uma versão anterior, todas as informações estão salvas e de fácil acesso.

Criando um repositório

1. Acesse github.com
2. Faça o login no site
3. Clique no botão + no canto superior esquerdo e crie seu primeiro repositório



4. Dê o nome ao seu repositório

5. Mantenha ele como Público
6. **Importante:** Inicialize ele com o arquivo README
7. Clique no botão verde “Criar repositório”

Subindo os arquivos que você criou

1. Clique no botão Upload Files

The screenshot shows a GitHub repository page for 'felipebarreiros / teste-delete'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The latest commit was made 2 minutes ago by 'felipebarreiros' with the message 'Initial commit'. Navigation links include 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', and 'Insights'. Buttons at the bottom include 'Create new file', 'Upload files', 'Find file', and 'Clone or download'.

2. Arraste todas as pastas e arquivos do seu projeto
3. Espere todos os arquivos serem carregados
4. Faça o primeiro **commit** no seu repositório

The screenshot shows the 'Commit changes' dialog box. It contains a text area with 'Primeiro Upload' and a description field with 'Add an optional extended description...'. Below the text area are two radio buttons: one selected for committing directly to the master branch, and another for creating a new branch and starting a pull request. At the bottom are 'Commit changes' and 'Cancel' buttons. A large red arrow points downwards from the top of the dialog towards the 'Commit changes' button.

E para visualizar o seu site e mostrar para todo mundo o que você fez em aula basta ir em Settings, e descer a página até “Github Pages”. Vai aparecer como *None* e ai é só trocar por *masterbranch* e dar save. Vai aparecer um link de acesso ao seu site.

The screenshot shows the GitHub Pages settings page. It includes a message about GitHub Pages hosting personal, organization, or project pages. Below is a green success message: '✓ Your site is published at <https://gakiefer.github.io/DSOP-economize/>'. The 'Source' section shows 'master branch' selected. A dropdown menu is open, showing options: 'master branch' (selected), 'master branch /docs folder', and 'None'. The 'master branch' option is described as 'Use the master branch for GitHub Pages.' and 'master branch /docs folder' as 'Use only the /docs folder for GitHub Pages.' At the bottom are 'Save' and 'Cancel' buttons.

Javascript, sua primeira linguagem de programação

Vimos muita coisa bacana e até nosso site já fizemos, temos que agora desenvolver nosso Pensamento Computacional, mas o que é isso? Com certeza, essa é uma habilidade do século 21.

Pensamento Computacional: entender como se trabalha com passos computacionais, que nada mais são do que operações possíveis para um computador, como tratar condições dentro de um IF para conseguir o resultado que se quer do seu algoritmo (que nada mais é do que o nome bonito para esse conjunto de passos computacionais).

Javascript, que não tem nada a ver com Java, é uma linguagem de programação que foi criada como parte dos navegadores web para que scripts e programas pudessem ser executados sem passar pelo servidor, controlando o navegador por exemplo, realizando comunicação e alterando o conteúdo do documento exibido.

Antes de entrar nos detalhes da linguagem discutimos dois conceitos muito importantes:

1. VARIÁVEL:

- **Explicação para criança de 6 anos:** imagina que você tem um monte de sapatos e você gosta de guardá-los em caixas todos no mesmo armário. Para facilitar sua vida, você etiqueta as caixas com o nome do sapato para que você saiba exatamente onde está cada um deles. Variáveis são essas “coisas” que podemos guarda e dar um nome para elas para facilitar nossa vida.
- **Explicação mais técnica:** quando queremos guardar alguma informação seja ela em qualquer formato precisamos alocar um espacinho na memória do nosso computador e para recuperar damos um nome para essa espaço. Os principais tipos de variáveis são: Números, que podem ser inteiros (int) ou ponto flutuante (float); texto, que chamamos de string; e boolean que é o famoso true or false.

2. FUNÇÃO:

- **Explicação para criança de 6 anos:** quando você pede para sua mãe um milkshake, você não sabe bem quais são as tarefas que ela tem que executar para produzir aquele resultado e você também não está interessado em saber, você quer mesmo o milkshake. Fazer um milkshake, portanto, é uma função da sua mãe.
- **Explicação mais técnica:** quando trabalhamos com linguagens de programação temos algumas funções pré-definidas que não temos sempre acesso ao código ou sabemos exatamente como elas executam, mas sabemos que precisamos dar um certo tipo de informação para elas e vamos receber algo de volta. Por exemplo, se escrevemos **alert ("Oi, mundo!")**; sabemos que essa função abre um popup no navegador com a mensagem que eu passar para essa função que está dentro das aspas. Como ele faz isso? Não precisamos saber para usar! :)

Vamos escrever nossas primeiras linhas de código. E o primeiro desafio é saber onde escrever esse tal de Javascript. No nosso exemplo vamos colocar ele no mesmo arquivo HTML que estamos fazendo nosso site, só vamos precisar escrever dentro de uma tag `<script>`, dentro da tag `<head>`.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Exemplo de JavaScript</title>
5      <script type="text/javascript">
6
7          SEU CÓDIGO JAVASCRIPT VAI AQUI!
8
9      </script>
10 </head>
11 <body>
12
13     SEU SITE VAI AQUI!
14
15 </body>
16 </html>

```

Missão 1: Mandar uma mensagem para o usuário

Agora vamos trabalhar só no código Javascript que portanto deve estar dentro da tag `<script>`. Nosso primeiro desafio é conseguir rodar uma popup com uma mensagem logo que a gente abrir o arquivo. Para isso escreva a seguinte linha de código:

```
alert("Olá! Eu sei o que você fez no verão");
```

Atenção para a letra minúscula, as aspas e o ponto e virgula no final :)

Salve o arquivo (cheque se a extensão é `.html`) e abra no navegador (se você abrir o gerenciador de arquivos e clicar duas vezes vai conseguir abrir direto no navegador já).

Missão 2: Pegar valores que o usuário digitou

Agora queremos pedir valores para o usuário através de uma janela para isso vamos ter que criar nossa primeira variável. Fazemos isso usando a palavra `var` ou o `let` que foi inserido no Javascript ES6.

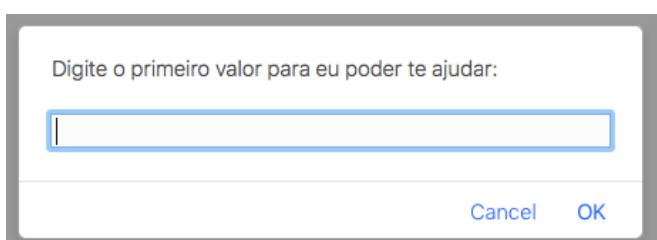
```
var valorDoPrimeiroTermo;
```

O nome dessa variável que criamos é `ValorDoPrimeiroTermo`. Podemos colocar qualquer coisa aí dentro. No nosso caso vamos colocar o valor que nosso usuário digitar usando a função `prompt`.

```
var valorDoPrimeiroTermo = prompt("Digite o primeiro valor para eu poder te ajudar: ");
```

Coloque essa linha aqui em cima no seu arquivo e abra. Algo parecido com isso deve aparecer.

A aparência depende do seu navegador :)



Missão 3: Fazer uma calculadora que soma valores.

Para fazer uma calculadora precisamos:

- pegar primeira valor com o usuário;
- pegar segundo valor com o usuário;
- realizar a soma;
- mostrar o resultado para o usuário.

Tente entender o código abaixo:

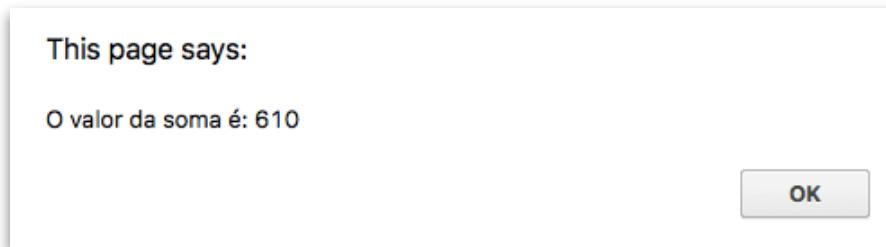
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Exemplo de JavaScript</title>
5     <script type="text/javascript">
6         // pegar os termos com o usuário
7         var valorDoPrimeiroTermo = prompt("Digite o primeiro valor para eu poder
8             te ajudar:");
9         var valorDoSegundoTermo = prompt("Digite o segundo valor para eu poder te
10            ajudar");
11
12         // realizar a soma
13         var valorSoma = valorDoPrimeiroTermo + valorDoSegundoTermo;
14
15     </script>
16 </head>
17 <body>
18
19 </body>
20 </html>
```

A grande nesse código é o **OPERADOR**.

OPERADOR: é usados para realizar as operações matemáticas básicas e gerar valores. Podem ser usados para comparar coisas também. Os operadores matemáticos são: + (adição), - (subtração), * (multiplicação) e / (divisão) e os de comparação são: > (maior), < (menor), == (igual), != (diferente), <= (menor ou igual), >= (maior ou igual).

DEU CERTO?! NÃO!

Olha aqui o que esse código gera se eu digital 6 como primeiro termo e 10 como segundo termos.



E sabe porquê? Nós estamos tratando os valores que os usuários escreveram como textos e não como números, então eles só juntaram os termos.

Precisamos contar que as nossas variáveis são números. Para isso usamos a função **Number()**:

```
// pegar os termos com o usuário
var valorDoPrimeiroTermo = Number(prompt("Digite o primeiro valor para eu
poder te ajudar: "));
var valorDoSegundoTermo = Number(prompt("Digite o segundo valor para eu
poder te ajudar"));
```

Agora confia que vai dar certo! Imagina que eu preciso fazer as 4 operações? Acho que você já está preparado para fazer isso, hein!? Olha o código aqui embaixo completinho:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Exemplo de JavaScript</title>
5     <script type="text/javascript">
6         // pegar os termos com o usuário
7         var valorDoPrimeiroTermo = Number(prompt("Digite o primeiro valor para eu
poder te ajudar: "));
8         var valorDoSegundoTermo = Number(prompt("Digite o segundo valor para eu
poder te ajudar"));
9
10        // realizar as operações
11        var valorSoma = valorDoPrimeiroTermo + valorDoSegundoTermo;
12        var valorSubtracao = valorDoPrimeiroTermo - valorDoSegundoTermo;
13        var valorMultiplicacao = valorDoPrimeiroTermo * valorDoSegundoTermo;
14        var valorDivisao = valorDoPrimeiroTermo / valorDoSegundoTermo;
15
16        // mostrar para o usuário
17        alert("O valor da soma é: " + valorSoma + "\n O valor da Sub é: " +
valorSubtracao + "\n O valor da Multiplicação é: " +
valorMultiplicacao + "\n O valor da Divisão é: " + valorDivisao);
18    </script>
19 </head>
20 <body>
21 </body>
22 </html>
```

Agora tá tudo funcionando né!? Tem só uma novidade aqui nesse código, o **\n**.

\n: esse \n significa newline e ele serve para pular linha assim como o **
** no HTML.

Leia o código com calma e perceba se você entendeu mesmo todas as linhas e não só copiou. Lembre-se que você passa mais tempo lendo código do que escrevendo. Sempre!

Missão 4: Fazer uma calculadora, mais espertinha com condições.

Agora queremos que a gente coloque uma condição para verificar algumas coisas que podem fazer nossa calculadora quebrar. Por exemplo, se o segundo termo for 0 vamos quebrar nossa calculadora já que não podemos ter divisão por 0, certo!?

Para fazer uma condição usamos o **IF**. E “dentro” desse **IF** colocamos a nossa condição. No nosso exemplo a condição é: **valorDoSegundoTermo == 0**. Então ficaria assim:

```
if (valorDoSegundoTermo == 0) {  
    valorDoSegundoTermo = Number(prompt("Coloque outra opção para o  
    segundo valor para eu poder te ajudar"));  
}
```

Outra possível verificação é ter certeza que o nosso usuário percebeu que está colocando números iguais para o primeiro e segundo termo. Nesse caso nossa condição seria: valorDoSegundoTermo == valorDoPrimeiroTermo. Então ficaria assim:

```
if (valorDoSegundoTermo == valorDoPrimeiroTermo) {  
    var resultado = confirm("Você tem certeza que quer manter os dois  
    número iguais?");  
}
```

Aqui criamos uma nova variável chamada **resultado** e usamos a função **confirm()**; que é uma maneira de pegar uma confirmação com o usuário. **Confirm** é como se fosse um **alert** que tem dois botões e eu consigo guardar a resposta do meu usuário. Se ele apertar OK, o valor de resultado é **true** e se ele clicar em CANCEL o resultado é **false**.

Então depois dessa confirmação precisamos tratar esse **resultado**. Se ele for false, precisamos pedir pra o usuário digitar novamente os valores, certo!? Para isso podemos fazer assim:

```
if (resultado == false) {  
    valorDoPrimeiroTermo = Number(prompt("Digite outro valor para o  
    primeiro termo para eu poder te ajudar: "));  
    valorDoSegundoTermo = Number(prompt("Digite outro valor para o  
    segundo termo para eu poder te ajudar: "));  
}
```

Com condições começamos a trabalhar lógica e muitas vezes a dificuldade de aprender a programar está no desenvolvimento da lógica e não tanto no código em si. E para aprender lógica só tem um jeito. Praticar, pensar, refazer até que você consiga desenvolver essa habilidade. Se proponha pequenos desafios e tente executar. Essa é uma ótima maneira de avançar.

Por exemplo, e comparações com strings? E se eu quisesse trabalhar com xarás (pessoas com nomes iguais). Como eu poderia fazer isso? E se eu quisesse validar um formulário? Como poderíamos fazer. Um bom caminho para quem quer avançar é usar essas duas referência aqui embaixo que são cursos iterativos muito bacanas:

Codecademy: <https://www.codecademy.com/learn/javascript>

W3Schools: <https://www.w3schools.com/js/>

E aí vai o código completão:

```

<!DOCTYPE html>
<html>
<head>
    <title>Exemplo de JavaScript</title>
    <script type="text/javascript">
        // pegar os termos com o usuário
        var valorDoPrimeiroTermo = Number(prompt("Digite o primeiro valor para eu poder te ajudar: "));
        var valorDoSegundoTermo = Number(prompt("Digite o segundo valor para eu poder te ajudar"));

        if (valorDoSegundoTermo == 0) {
            valorDoSegundoTermo = Number(prompt("Coloque outra opção para o segundo valor para eu poder te ajudar"));
        }

        if (valorDoSegundoTermo == valorDoPrimeiroTermo) {
            var resultado = confirm("Você tem certeza que quer manter os dois números iguais?");
        }

        if (resultado == false) {
            valorDoPrimeiroTermo = Number(prompt("Digite outro valor para o primeiro termo para eu poder te ajudar: "));
            valorDoSegundoTermo = Number(prompt("Digite outro valor para o segundo termo para eu poder te ajudar:"));
        }

        // realizar as operações
        var valorSoma = valorDoPrimeiroTermo + valorDoSegundoTermo;
        var valorSubtracao = valorDoPrimeiroTermo - valorDoSegundoTermo;
        var valorMultiplicacao = valorDoPrimeiroTermo * valorDoSegundoTermo;
        var valorDivisao = valorDoPrimeiroTermo / valorDoSegundoTermo;

        // mostrar para o usuário
        alert("O valor da soma é: " + valorSoma + "\n O valor da Sub é: " + valorSubtracao + "\n O valor da Multiplicação é: " + valorMultiplicacao + "\n O valor da Divisão é: " + valorDivisao);
    </script>
</head>
<body>
</body>
</html>

```

Introdução ao App Inventor

O App Inventor é uma boa solução para quem nunca programou, mas quer começar a criar aplicativos e iniciar no mundo da programação de aplicativos.

Foi criado pelo Google e hoje ele está sobre domínio do MIT (Massachusetts Institute of Technology) e está disponível apenas para Android.



MIT AI2 COMPANION

Disponível na Google Play Store. Baixe para testar seu app em seu celular.

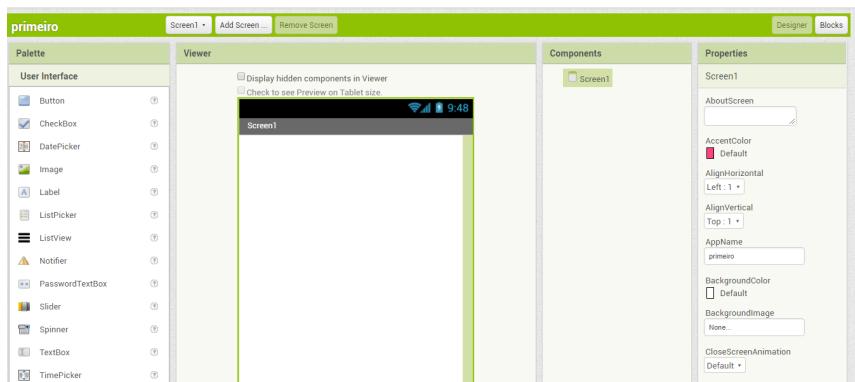
Para quem tem Iphone, há alguns apps com dicas e tutoriais.

Como funciona?

Toda criação do aplicativo é feita de forma visual, juntando peças com peças como se fosse um quebra-cabeça.

Para começar, você precisará acessar o App Inventor acessando <http://ai2.appinventor.mit.edu/>. É necessário ter uma conta Google.

Crie um novo projeto.



Assim que criar o projeto, aparecerá um menu lateral com os elementos que você poderá trabalhar para criar as telas do seu app, a tela em branco e no canto direito aparece o botão "Blocks" que é onde acontece a mágica, a parte de lógica por blocos.

Blockly é uma biblioteca de Javascript que funciona por blocos.

É possível trabalhar com variáveis, laços de repetição e condições comuns em Javascript mas desta forma é mais visual e para começo ajuda quando você não tem tanta facilidade com a sintaxe (ainda...)



The screenshot shows the Blockly-based script editor. It contains two main sections:

- Script for Button1.Click:**

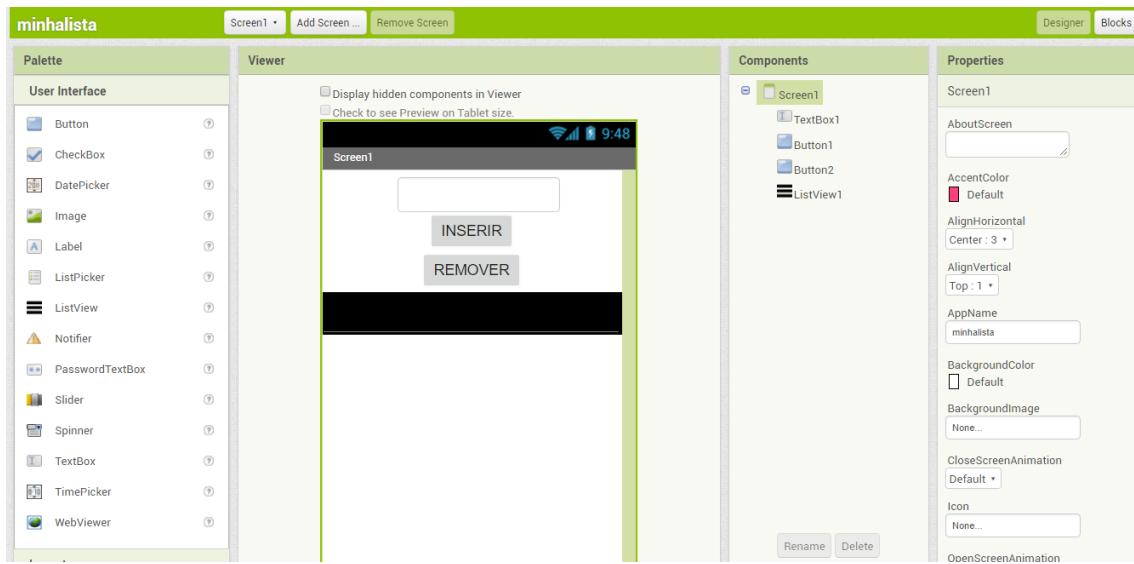
```
initialize global Lista to create empty list
initialize global TamanhoLista to 0
when Button1.Click
do
  set global TamanhoLista to get global TamanhoLista + 1
  insert list item list get global Lista
  index get global TamanhoLista
  item TextBox1.Text
  set ListView1.Elements to get global Lista
```
- Script for Button3.Click:**

```
when Button3.Click
do
  remove list item list get global Lista
  index ListView1.SelectionIndex
  set ListView1.Elements to get global Lista
```

At the bottom left, there are warning icons: a yellow triangle with '0' and a red circle with '0'. A 'Show Warnings' button is also present.

Missão 1: Criando seu primeiro Aplicativo

Neste exemplo vamos fazer um aplicativo que simule uma lista de tarefas. O usuário poderá inserir novas tarefas e removê-las da lista.



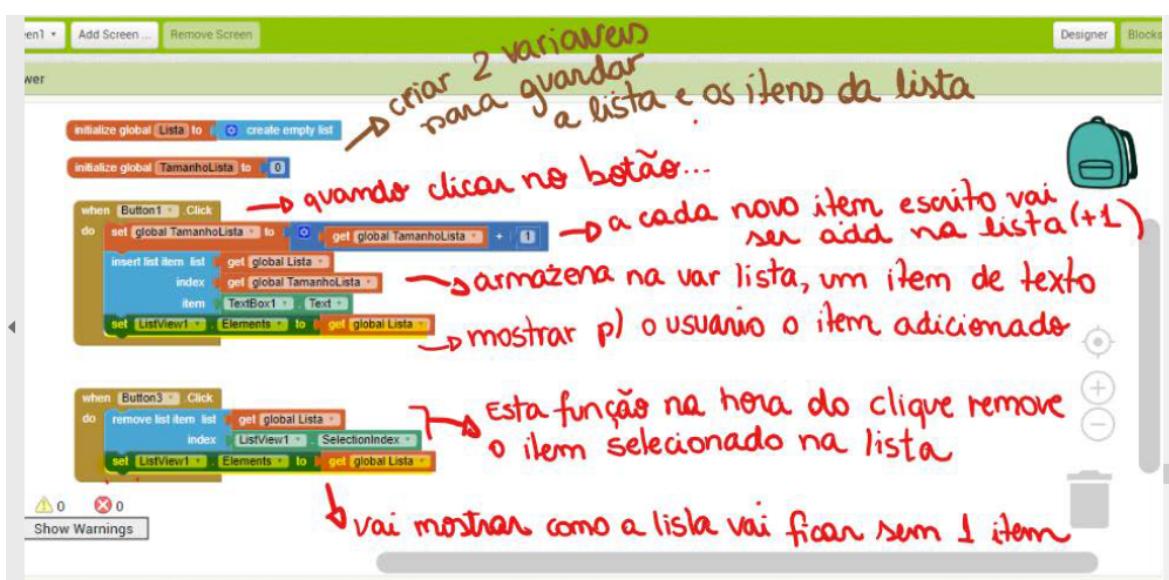
Vamos utilizar uma caixa de texto, onde o usuário poderá digitar suas tarefas, 2 botões de inserir e remover e uma área para lista aparecer que na tela é este retângulo preto, mas quando carregar em seu celular não aparece desta forma.

Há inúmeros itens que você pode testar para montar seu layout, a estrutura de como será seu aplicativo, inserir mídias, banco de dados, e até sensores.

Mas para que eu colocaria sensores? Quem lembra de um aplicativo que você podia acender e apagar uma vela, ou isqueiro? O princípio deste app é exatamente de um acelerômetro. Com o balançar ou sacudir é possível mudar a imagem/vídeo para "acesa" ou apagada".

No parte de Blocks é feita toda a lógica. Como colocado abaixo, cada peça serve para "falar" uma função a ser feita para acontecer uma ação.

É colocado o que deve ser feito na hora do clique do botão para inserir e armazenar os itens inseridos na lista, e no outro botão faria a função de remover aqueles itens selecionados.

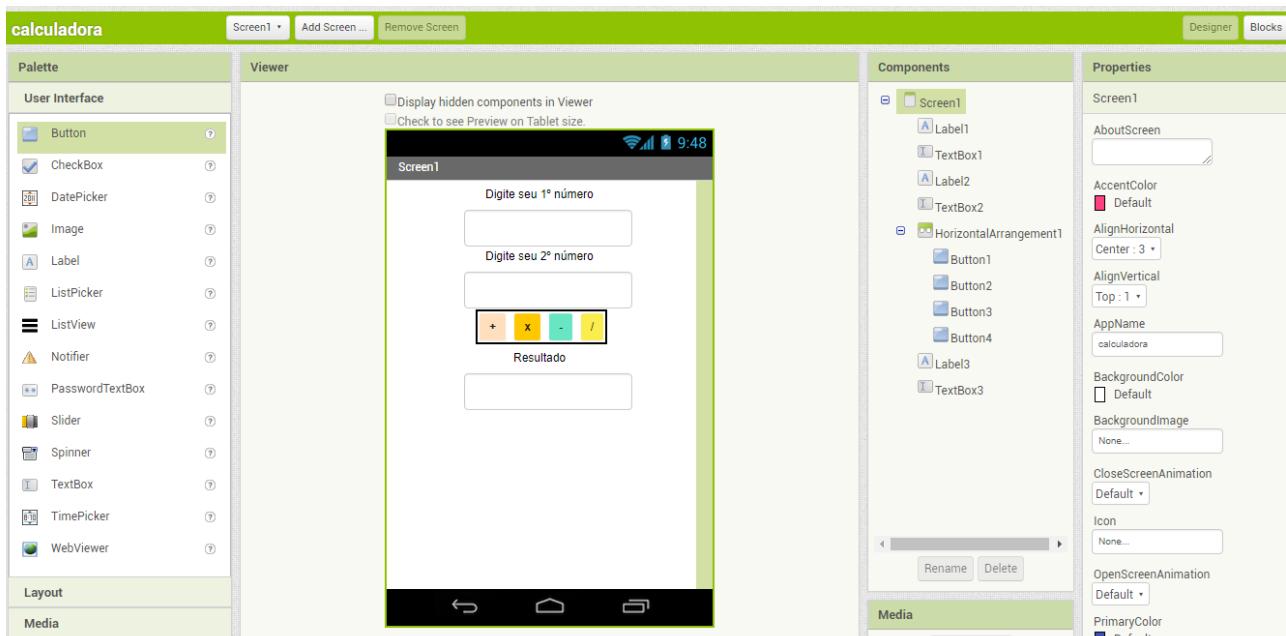


Missão 2: Criando uma calculadora simples

Neste novo projeto do App Inventor iremos criar uma calculadora simples, que através de duas entradas de texto, é capaz de somar os dois números e mostrar o resultado em uma caixa de texto.

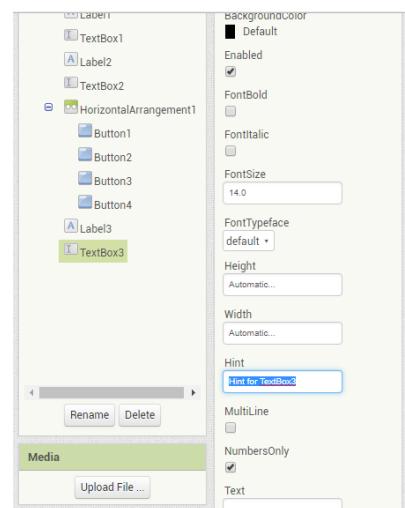
1. Comece criando um novo projeto em branco.
2. Arraste dois campos do tipo Caixa de Texto para a área central. Este tipo de campo aceita informações enviadas pelos usuários.
3. Em paralelo, você pode inserir Labels para colocar qual dado será inserido naquele campo.
4. Arraste 4 botões e altere cada texto para os símbolos de somar, multiplicar, subtrair e dividir. Assim ficará mais fácil para o usuário entender qual é o objetivo do botão.
5. Por fim, Você pode estilizar cada item inserido no app, de acordo com seu critério.

Nós fizemos de uma forma bem simples, mas fique à vontade para usar a criatividade.



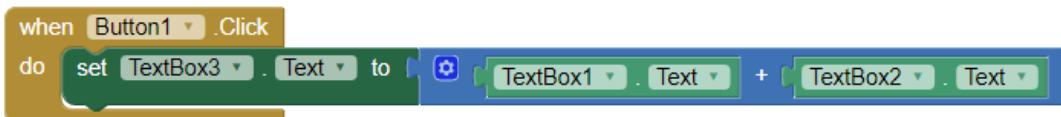
Todos os itens visuais já foram colocados no aplicativo.

Importante: nos itens de caixa de texto é necessário apagar o item "hint" que aparece no seu menu lateral direito e selecionar o item "numbersonly" pois trabalharemos só com números, e se você não fizer este processo, a aplicação entenderá como strings e fará concatenação ao invés de uma soma por exemplo.



Para o botão ter uma ação no momento em que apertarmos ele, precisamos criar uma função que executa a soma e coloca em **resultado** o resultado da soma :)

No código, essa disposição fica:

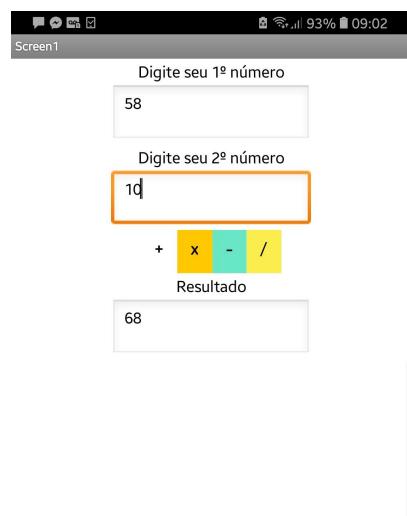


*Quando clicar no **botão 1** (botão +), fazer a função de aparecer na **caixa de texto 3**, o resultado da soma do valor que será colocado na caixa de texto 1 e da **caixa de texto 2**.*



A parte de Blockly ficaria desta forma acima, a função principal é a mesma mudando apenas qual operação matemática de acordo com cada botão ao clicar.

Super simples não? Agora vamos ver como ficaria no celular.



Como criar uma conta de desenvolvedor no Google Play

É fácil e rápido registrar e começar a distribuir seus aplicativos no Google Play, que é a loja de aplicativos Android, que vai possibilitar com que pessoas achem e baixem seu aplicativo, sem que você precise enviar um por um o seu .apk.

Antes de começar você precisa saber que há uma taxa de registro de US\$ 25 cobrados por uma conta de Console do desenvolvedor do Google Play, o que é uma taxa considerada barato uma vez que é cobrada apenas uma vez na sua vida e você pode publicar quantos aplicativos quiser. Só pra vocês compararem a Apple cobra \$99 por ano por uma conta de desenvolvedor.

Além da conta de desenvolver é necessário estar em um país compatível para distribuir aplicativos no Google Play (podem ficar tranquilos que o Brasil está nessa lista).

É bem simples tornar sua conta pessoal do Google em uma conta de desenvolvedor. Basta seguir o seguinte passo a passo:

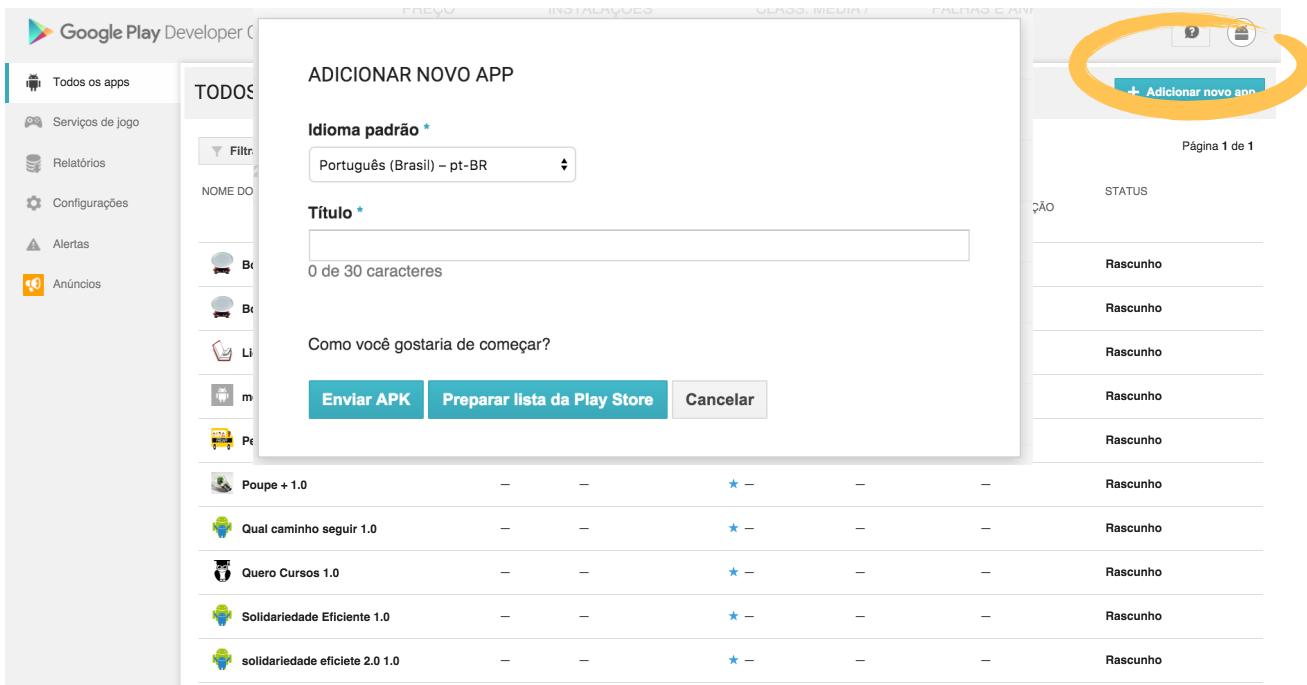
1. Faça login na Conta do Google que será usada como a proprietária da sua conta de desenvolvedor.
2. Acesse a página de inscrição no Console do desenvolvedor do Google Play para começar a se registrar: <https://play.google.com/apps/publish/signup/>
3. Durante o registro, será preciso aceitar o Contrato de distribuição dos desenvolvedores do Google Play para avançar. Se sua conta já violou esse contrato, não será possível se registrar como um Desenvolvedor do Google Play.
4. Pague a taxa de registro usando uma forma de pagamento válida da Carteira virtual do Google.
5. Preencha os detalhes da sua conta, incluindo seu Nome do desenvolvedor. Esse é o nome que é exibido na Google Play Store, então se você quer que apareça o nome do seu time é preciso alterar. Obs: A nossa conta é compartilhada então, nesse caso vamos manter o nome, ok!?

O processamento de seu registro do Console do desenvolvedor do Google Play poderá levar até 48 horas! Como você estão usando a conta compartilhada pode correr para a próxima sessão e publicar o seu aplicativo!

Como publicar um novo aplicativo

Agora que você já se registrou como desenvolvedor, é fácil fazer o upload de seu aplicativo para o Google Play!

Na tela inicial do Console do desenvolvedor do Google Play, selecione Adicionar novo aplicativo. Será solicitado que você preencha as informações a seguir sobre seu aplicativo! As primeiras informações que você vai ter que preencher são idioma e nome do aplicativo. Se você já tiver o arquivo .apk você já pode subir clicando em [\[Enviar APK\]](#), mas se você clicar em [\[Preparar lista da Play Store\]](#) você pode começar a completar as outras informações e depois subir o executável.



Aí vão alguns outros detalhes do material que você tem que produzir para publicar o seu app:

Arquivo APK:

- o tamanho máximo suportado por um único .apk é 50 MB.
- os nomes dos pacotes de aplicativos são exclusivos e permanentes. Por isso, escolha-os de forma sensata.
- arquivo .apk de rascunho do aplicativo: ao fazer o upload de um arquivo .apk, ele pode ser salvo como rascunho enquanto você edita outros aspectos da lista. O botão Salvar rascunho será exibido na caixa de diálogo após a conclusão do upload do APK. Se você deseja fazer o upload de um novo APK, sem publicá-lo no Google Play imediatamente, é possível salvar um novo APK como um rascunho a partir do modo "Simples" ou "Avançado".

Capturas de tela:

- são necessárias duas capturas de tela, e as seis capturas adicionais disponíveis são opcionais. O jeito mais fácil de fato é tirar print das telas do seu aplicativo pelo celular.

Detalhes da entrada:

- **Idioma:** para indicar o idioma de seu aplicativo. O idioma padrão é o inglês americano. Mais idiomas ficarão disponíveis à medida que telefones com tecnologia Android forem disponibilizados nesses idiomas. Além disso, você pode incluir traduções do título e da descrição do aplicativo para promovê-lo para usuários que falam outros idiomas. As traduções serão exibidas no Google Play.
- **Título:** o nome de seu aplicativo como deseja que ele apareça no Google Play. Você pode adicionar um por idioma.
- **Descrição:** a descrição visível de seu aplicativo no Google Play. Utilize o recurso de tradução a fim de traduzir sua descrição para outros idiomas. Há um limite de 4 mil caracteres para esse campo.
- **O que há de novo nesta versão?:** adicione notas sobre alterações específicas às versões publicadas do seu aplicativo.

- **Tipo de aplicativo:** o Google Play está dividido em dois tipos principais de aplicativos: Aplicativos e Jogos. Escolha um.
- **Categoria:** selecione uma categoria para seu aplicativo. Leia mais sobre categorias na própria página de publicação.
- **Locais:** são os locais em que você pode distribuir seus aplicativos. Nem todos os locais listados no momento têm usuários com dispositivos habilitados para Android. Você pode selecionar locais um por um ou escolher a opção.
- **Dados de contato:** você precisa escolher um canal de suporte para seu aplicativo: website, e-mail ou telefone. Os usuários do Google Play poderão ver essas informações. Você pode oferecer vários canais de suporte.