

#2 Ionic Framework

CUSTOMIZANDO E CRIANDO COMPONENTES

Por Fábio Rogério da Silva José
Última atualização em 06/2016

Sobre este Ebook

Continuando a série de ebooks sobre Ionic Framework, este material irá tratar de um fator importante, e muito particular, que está ligado diretamente com o campo de estudo *UX Design*. Cada aplicativo tem um público específico e o mesmo deve ser analisado com muito cuidado para que o usuário tenha a melhor experiência no aplicativo proposto.

Aprender sobre *UX Design* (Experiência do usuário) e *UI Design* (Interface do usuário) não está no escopo deste material, embora faremos uma pequena introdução, mas tenha em mente que este é um ponto de extrema importância para que seu aplicativo tenha interfaces elegantes e proporcione uma experiência agradável e satisfatória para o usuário final.

Para exemplificar e praticar, que é o objetivo central deste ebook, vamos criar algumas interfaces de aplicativos sem funcionalidades, ou seja, sem ações e eventos com o foco em ver as diferentes formas de customizar os componentes do Ionic e também criar componentes próprios.

Público alvo

Este ebook foi escrito para você que está começando no mundo do desenvolvimento de aplicativos móveis e que já leu o ebook [#1 Ionic Framework - Essencial](#). Não iremos abordar padrões de desenvolvimento e não vamos utilizar lógicas de programação complexas para facilitar o aprendizado e ter um público mais amplo, tendo em vista que iniciantes também no mundo da programação podem estar lendo este ebook.

Este material não deverá ser seu único ponto de referência e guia, pois existem outros ebooks, livros e blogs que falam sobre Ionic Framework e podem ampliar seu conhecimento.

Sobre o autor

Fábio Rogério da Silva José, conhecido como Fábio Rogério SJ nas redes sociais, é desenvolvedor de aplicações web e mobile desde quando o *Internet Explorer 7* era um pesadelo e desenvolver aplicativos híbridos com tecnologias web era um trabalho árduo, ou seja, desde 2007.

Atualmente Fábio Rogério trabalha com consultoria e treinamento em desenvolvimento de aplicações web e mobile utilizando tecnologias híbridas. E também é professor, de curso superior, onde ministra as disciplinas de desenvolvimento *frontend*, desenvolvimento de aplicativos móveis, web design, design de interação e lógica de programação.

Sumário

Introdução	3
Customizando listas	4
Fontes, plano de fundo geral e campos	9
Fontes	12
Criando badges e customizando botões	14
Customizando grid	19
Customizando formulários	25
Header transparente e scroll vertical	32
Ícones e novos componentes	38
Explore o universo web	45
Próximo passo	54

Introdução

As interfaces do usuário estão cada vez mais aprimoradas e estilizadas para garantir uma boa experiência com o usuário. Os usuários também estão mais exigentes e críticos com a navegabilidade e design das aplicações, existem vários relatos de aplicações que não atingiram o público que gostaria por não ter pensado na melhor experiência com o usuário.

Está fora do escopo deste ebook tratar diretamente sobre a melhor forma de se fazer um bom *UX Design* e também o *UI Design*, apenas vamos treinar e ver como é simples criar interfaces ricas e com poucas customizações nos componentes que o Ionic Framework já disponibiliza em seus framework.

É de extrema importância você estudar mais sobre a experiência do usuário, pretendo em breve escrever um ebook centrado neste assunto, mas por enquanto você pode ler materiais relacionados. Recomendo a leitura do material escrito por Griffiths da Google, "[Os Princípios de UX para Aplicativos em Dispositivos Móveis](#)" onde é comentado sobre vários pontos importantes que um aplicativo móvel precisa se preocupar em atender seu usuário.

Como foi explicado no ebook [#1 Ionic Framework - Essencial](#), os componentes do Ionic são elementos com seus estilos já programados, mas podemos fazer a customização que for necessária.

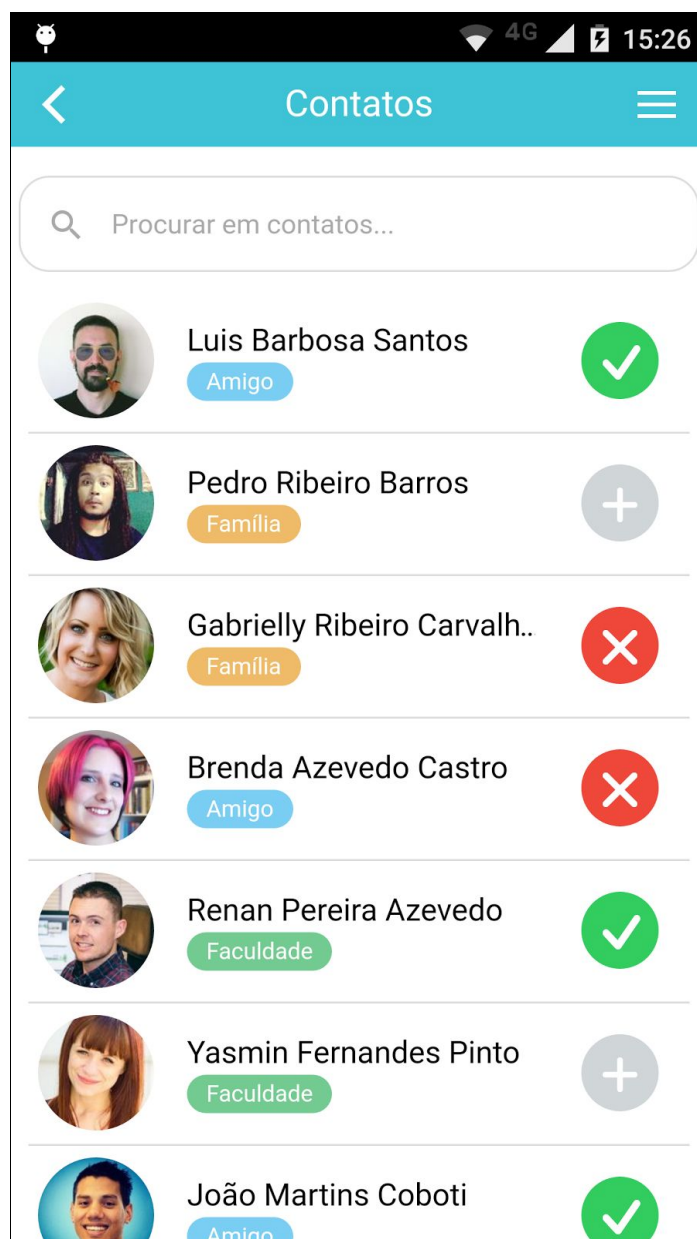
Neste material criei algumas interfaces simples para atendermos a customização dos principais componentes e comportamentos das interfaces no Ionic. Em muitos casos vamos utilizar uma linguagem mais explicativa para um melhor entendimento, porém tenham em mente que quanto menor ficar seu código mais performativo ele irá ficar.

A prática leva à perfeição, então implemente os exemplos e fique à vontade para estender os *layouts* aqui proposto com seu toque e criar outros diferentes.

Customizando listas

As listas em Ionic, na versão 1, são classes simples e limitadas em sua exibição, podemos criar listas com textos, imagens arredondadas e thumbnail, mas para uma lista mais rica em componentes visuais precisamos de algumas classes a mais em nossa folha de estilo.

Vamos criar uma lista de contatos com botões diferenciados:



Design: [Ayoub Elred](#) - Código fonte: [GitHub](#)

Crie um novo aplicativo com o nome appContatos:

```
ionic start appContatos blank
```

Primeira coisa vamos criar um subheader para que o campo de pesquisa fique sempre fixo no topo e apenas a lista sofra rolagem. Abaixo do seu header adicione um subheader:

```
<ion-header-bar class="bar-subheader ">
  <div class="item item-input item-search">
    <i class="icon ion-android-search placeholder-icon icon-search"></i>
    <input type="search" placeholder="Procurar em contatos...">
  </div>
</ion-header-bar>
```

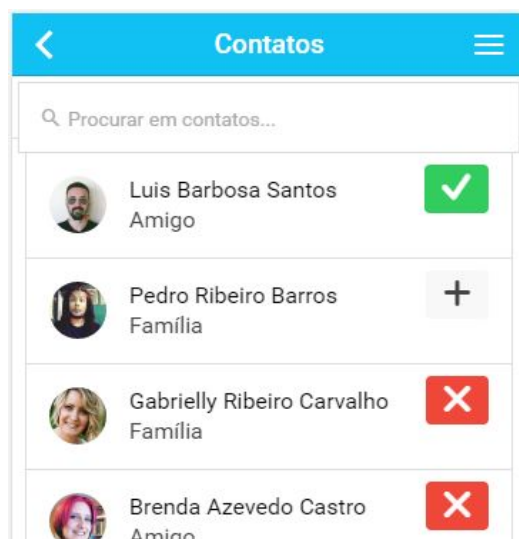
Na área de conteúdo (`ion-content`) adicione uma lista com avatar e botão do lado direito. Vamos adicionar a classe `list-custom` para facilitar a seleção no CSS desta nossa lista customizada. Vamos também adicionar um span com a classe `badge-custom`, para dar um comportamento diferente do badge que o Ionic propõem. Para gerar imagens de usuários utilizo o site <https://randomuser.me/>.

```
<ion-content class="padding">
  <ion-list class="list-custom">
    <ion-item class="item-avatar item-button-right">
      
      <h2>Luis Barbosa Santos</h2>
      <span class="badge-custom amigo">Amigo</span>
      <button class="button button-balanced">
        <i class="icon ion-checkmark-round"></i>
      </button>
    </ion-item>
    <ion-item class="item-avatar item-button-right">
      
      <h2>Pedro Ribeiro Barros</h2>
      <span class="badge-custom familia">Família</span>
      <button class="button button-stable">
        <i class="icon ion-plus-round"></i>
      </button>
    </ion-item>
    <ion-item class="item-avatar item-button-right">
      
      <h2>Gabrielly Ribeiro Carvalho</h2>
      <span class="badge-custom familia">Família</span>
      <button class="button button-assertive">
        <i class="icon ion-close-round"></i>
      </button>
    </ion-item>
    ...
  </ion-list>
</ion-content>
```

...

Crie mais itens para conseguirmos validar a rolagem da página.

O resultado utilizando apenas os componentes do Ionic é este:



Agora vamos a customização, primeiro vamos ajustar e customizar nosso header e subheader. no style.css adicione a classe abaixo:

```
.bar.bar-calm {
  border-color: #3ec3d6;
  background-color: #3ec3d6;
  background-image: none;
}
.bar-subheader {
  height: 70px;
  background-image: none;
}
.has-subheader {
  top: 110px;
}
```

Aqui alteramos a cor do tema do Ionic para outra cor, aumentamos o tamanho do subheader e ajustamos a área de conteúdo para respeitar o novo tamanho do subheader. Vamos ajustar o campo de pesquisa:

```
.item-search {
  border-radius: 16px;
  margin-top: 10px;
  margin-bottom: 10px;
}
.icon-search {
  margin-right: 10px;
```

```

        font-size: 20px;
    }

```

Deixamos as bordas arredondadas, alinhamos o campo dentro do espaço existente e aumentamos o tamanho do ícone do campo de pesquisa. Agora vamos a lista:

```

.list-custom .item {
    border-width: 0;
    border-bottom-width: 1px;
    border-top-width: 1px;
}
.list-custom .item:first-child {
    border-top-width: 0;
}
.list-custom button {
    margin-top: 8px;
    border-radius: 50%;
    width: 40px;
    height: 40px;
}
.list-custom button i {
    margin-left: -2px;
    margin-top: 4px;
}
.list-custom span {
    font-size: 12px;
}
.list-custom h2 {
    margin-left: 10px;
}
.list-custom .item-avatar img {
    top: 6px;
    left: 5px;
    max-width: 60px;
    max-height: 60px;
}

```

Primeiro removemos as bordas da lista e deixamos apenas as bordas superior e inferior, em seguida selecionamos o primeiro item da lista e removemos a borda superior. Para os botões aumentamos o tamanho, deixamos com as bordas arredondadas e ajustamos as margens do botão e do ícone. Para o span ajustamos o tamanho da fonte e do h2 ajustamos a margem e por fim a imagem de avatar ajustamos o posicionamento e também o tamanho. Agora vamos ajustar a cor do botão cinza para ficar mais visível:

```

.button.button-stable,
.button.button-stable:hover {
    background-color: #d0d5d8;
    color: #fff;
}
.button.button-stable.activated {

```



```
        background-color: #c4c9cd;
    }
```

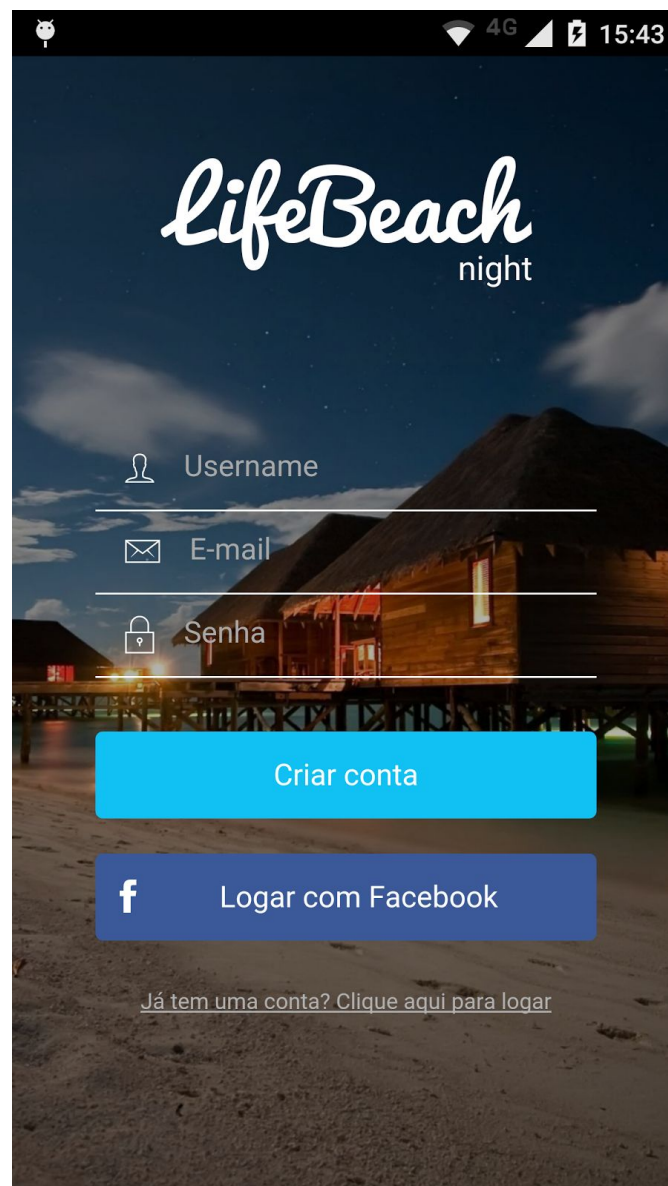
Alteramos a cor de fundo e a cor do ícone. Para concluir nosso app vamos adicionar as classes do badge customizado:

```
.badge-custom {
    background-color: #000;
    color: #fff;
    padding-left: 10px;
    padding-right: 10px;
    border-radius: 10px;
    float: left;
    margin-left: 10px;
}
.badge-custom.amigo {
    background-color: #7bcef3;
}
.badge-custom.familia {
    background-color: #efbb68;
}
.badge-custom.faculdade {
    background-color: #74cb92;
}
```

Em geral nosso novo componente destaca o span com cores diferentes para cada categoria de contato.

Fontes, plano de fundo geral e campos

Trabalhar com imagens de plano de fundo é sempre instigante e deixa nosso aplicativo com um aspecto de maioridade, mas lembre-se que a escolha de uma boa imagem é fundamental para um design rico. Neste exemplo a imagem é escura para destacar os campos com cor clara.



Design: [Ayoub Elred](#) - Código fonte: [GitHub](#)

Crie um novo aplicativo com o nome appLogin:

```
ionic start appLogin blank
```

Vamos primeiro criar nossa estrutura de elementos HTML, neste exemplo não teremos header:

```
<ion-pane class="login-custom">
  <ion-content>
    <h1>LifeBeach <span>night</span></h1>
    <ion-list class="list-inset">
      <ion-item class="item-input">
        <i class="icon ion-ios-person-outline placeholder-icon"></i>
        <input type="text" placeholder="Username">
      </ion-item>
      <ion-item class="item-input">
        <i class="icon ion-ios-email-outline placeholder-icon"></i>
        <input type="text" placeholder="E-mail">
      </ion-item>
      <ion-item class="item-input">
        <i class="icon ion-ios-locked-outline placeholder-icon"></i>
        <input type="text" placeholder="Senha">
      </ion-item>
      <ion-item class="item-button button-first">
        <button class="button button-block button-calm">Criar conta</button>
      </ion-item>
      <ion-item class="item-button">
        <button class="button button-block icon-left ion-social-facebook
button-positive">Logar com Facebook</button>
      </ion-item>
      <ion-item class="item-button">
        <button class="button button-block button-clear">
          Já tem uma conta? Clique aqui para logar
        </button>
      </ion-item>
    </ion-list>
  </ion-content>
</ion-pane>
```

Adicionamos a classe `login-custom` no `ion-pane`, que é o painel principal da aplicação, com isto conseguimos aplicar um fundo na tela inteira. O `h1` irá ser responsável pelo título do aplicativo, veja que dentro dele temos um `span`, pois apenas naquele pedaço de texto iremos estilizar de forma diferente.

Para o formulário adicionamos uma lista com itens de input e botões. Nosso aplicativo sem estilo está assim:

Vamos começar customizando o plano de fundo, no style.css adicione a classe:

```
.login-custom {
    background-color: #000;
    background-image: url(../img/fundo.png);
    background-size: cover;
    background-position: center;
}
```

Minha imagem de fundo está com transparência para poder fundir com a cor preto do próprio elemento, você pode usar outros recursos mais performáticos como o [background-blend-mode](#) do CSS3. Para deixar a imagem responsiva, ou seja, utilizar o espaço que tem disponível na tela, utilizamos o [background-size:cover](#). Vamos customizar a lista:

```
.list {
    width: 75%;
    margin: auto;
    margin-top: 60px;
}
.item {
    background: none;
    border:none;
}
.item-input {
    border-bottom: 1px solid #fff;
}
.item-input i {
    color: #fff;
    font-size: 25px;
}
.item-input input {
```

```
    font-size: 16px;
    margin-left: 10px;
    color: #fff;
}
.item-button {
    padding: 0;
}
```

Deixamos a lista ocupando 75% da tela e no centro, removemos o cor branca de fundo e ajustamos tamanho de fonte e posicionamento dos elementos da listagem. Agora vamos instalar fontes externas.

Fontes

O processo de utilizar fontes é similar o que fazemos para websites, basicamente é deixar acessível o arquivo da fonte (.woff, .ttf, etc) e criar um `@font-face` para declarar a fonte e suas propriedades.

O Google tem um serviço gratuito para fontes, o [Google Fonts](#), que tem mais de 800 fontes preparadas para utilizar na web.

Para agilizar o processo de baixar as fontes, criar o `@font-face` e incluir no projeto eu criei um módulo em NodeJS para realizar está tarefa, então vamos lá. Faça a instalação de forma global do módulo getfont:

```
npm install -g getfont
```

Para utilizar o módulo basta selecionar a fonte, ou as fontes, no site do Google Fonts que deseja utilizar no projeto e passar o link de *Embed* para o getfont. Para este *layout* estou utilizando a fonte Pacífico e a Roboto, então dentro da pasta do projeto digite o comando abaixo:

```
getfont "Pacífico|Roboto" --ionic
```

Com o parâmetro `--ionic` informado o módulo irá baixar a fonte e salvar dentro da pasta fonts, criar o font.css dentro da pasta css e alterar o index.html para adicionar o carregamento da fonte.

Saiba mais como utilizar o [getFont aqui](#).

O Roboto já tem instalado nos aparelhos com sistema operacional Android, mas para outras plataformas, incluindo no browser, precisamos instalar a fonte explicitamente.

Agora vamos alterar a fonte do título do nosso aplicativo, no style.css adicione as seguintes classes:

```

h1 {
  font-family: 'Pacifico';
  color: #fff;
  text-align: center;
  font-size: 50px;
  padding-top: 50px;
}
h1 span {
  font-family: 'Roboto';
  font-size: 18px;
  position: absolute;
  margin-top: 55px;
  margin-left: -35px;
}

```

Definimos a fonte Pacifico para o h1 com exceção do span interno, onde deixamos com a fonte Roboto e um posicionamento diferente.

Para concluir nosso app vamos customizar os botões:

```

.button-first {
  padding-top: 20px;
}
.button.button-positive {
  background-color: #3B5998;
}
.button-clear.button-clear {
  padding: 0;
  margin: 0;
  font-size: 12px;
  text-decoration: underline;
}

```

Alteramos a cor do tema azul do Ionic para ficar mais relativo com o azul do Facebook e ajustamos as margens dos botões.

Criando badges e customizando botões

Este próximo exemplo é simples mas didático para percebermos que em um mesmo *layout* podem existir diferentes estilos para os mesmos botões e utilizar conceitos, como o badges, para outras finalidades.



Código fonte: [GitHub](#)

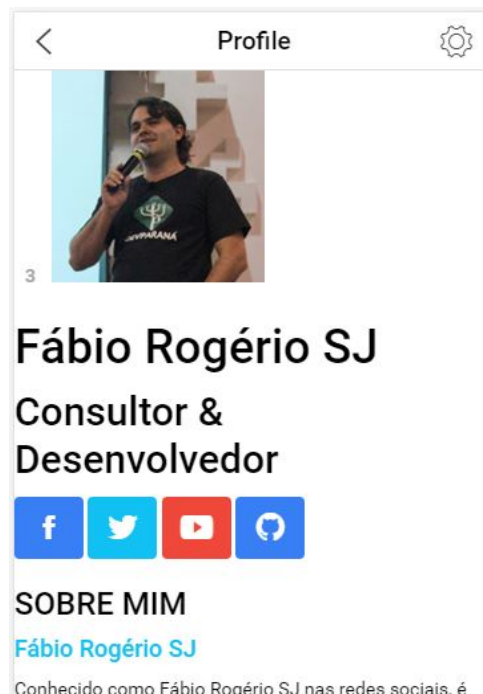
Crie um novo aplicativo com o nome appProfile:

```
ionic start appProfile blank
```

Na área de conteúdo adicione os seguintes elementos HTML:

```
<ion-content>
  <div class="top-profile">
    <div class="badge">3</div>
    
  </div>
  <h1 class="name">Fábio Rogério SJ</h1>
  <h2 class="prof">Consultor & Desenvolvedor</h2>
  <div class="sociais">
    <button class="button icon ion-social-facebook button-positive
button-facebook"></button>
    <button class="button icon ion-social-twitter button-calm"></button>
    <button class="button icon ion-social-youtube button-assertive"></button>
    <button class="button icon ion-social-github button-positive"></button>
  </div>
  <h3>SOBRE MIM</h3>
  <h4 class="calm">Fábio Rogério SJ</h4>
  <p>Conhecido como Fábio Rogério SJ nas redes sociais, é desenvolvedor de
aplicações web e mobile desde 2007. Hoje trabalha com consultoria e treinamento.</p>
  <h3>FALE COMIGO</h3>
  <div class="contato">
    <button class="button icon ion-ios-telephone button-assertive
button-small"></button> +55 44 9999-8888
  </div>
  <div class="contato">
    <button class="button icon ion-ios-email button-energized
button-small"></button> fabio.rogerio.sj@gmail.com
  </div>
  <div class="contato">
    <button class="button icon ion-ios-location button-balanced
button-small"></button> Av São Paulo, 345, Centro, Maringá-PR
  </div>
  <div class="contato">
    <button class="button icon ion-earth button-positive button-small"></button>
www.fabiorogerosj.com.br
  </div>
</ion-content>
```

Adicionamos alguns elementos de imagem, títulos e botões, nosso app sem customização está assim:



Vamos ao style.css e começar ajustando o topo da aplicação:

```
.bar {
    background:none;
}
.top-profile {
    text-align: center;
    margin-top: 20px;
}
.top-profile img {
    width: 120px;
    border-radius: 50%;
}
.top-profile .badge {
    background-color: #FF4B5A;
    width: 22px;
    height: 22px;
    color: #fff;
    border-radius: 50%;
    text-align: center;
    position: absolute;
    margin-left: 90px;
    margin-top: 5px;
}
```

Removemos o fundo da barra do topo, centralizamos o conteúdo do topo e criamos o componente badge. Agora vamos ajustar o título e subtítulo do topo:

```

.name {
  font-size: 18px;
  color: #4E4E4E;
  text-align: center;
  margin: 0;
}
.prof {
  font-size: 12px;
  color: #A5A5A5;
  text-align: center;
  margin: 0;
}

```

Aplicamos cores, tamanho de fonte e posicionamento aos títulos, agora vamos customizar os botões de redes sociais:

```

.sociais {
  text-align: center;
  margin-top: 10px;
}
.sociais button {
  border-radius: 50%;
  min-width: 45px;
  min-height: 45px;
  margin: 5px;
}
.sociais button.button.icon:before {
  font-size: 20px;
}
.button-positive.button-facebook {
  background-color: #5171A5;
}

```

Apenas deixamos redondos e alteramos tamanho e posicionamento dos botões. Para o botão do Facebook alteramos também sua cor. Agora vamos ajustar os demais títulos e parágrafo:

```

h3 {
  color: #909090;
  font-size: 14px;
  padding: 10px;
  margin-bottom: 0;
}
h4 {
  font-size: 14px;
  padding-left: 10px;
  padding-right: 10px;
  margin: 0;
}
p {

```

```
    color: #909090;
    font-size: 14px;
    padding-left: 10px;
    padding-right: 10px;
}
```

Ajustamos cores, tamanhos e posicionamentos dos demais títulos e dos parágrafos. Para concluir nosso app vamos ajustar os espaços dos botões de contato:

```
.contato {
    color: #484848;
    margin-top: 10px;
}
.contato button {
    margin-left: 10px;
    margin-top: -3px;
    margin-right: 10px;
}
```

Apenas alteramos cores e posicionamentos dos botões.

Customizando grid

As grids, diferentemente das listas, podem conter linhas e colunas. É bem comum utilizarmos elas para diagramar alguma apresentação estruturadas de dados e elementos. Neste aplicativos vamos utilizar apenas os recursos simples da grid que é linhas e comunas e com isso customizar seu conteúdo.



FábioCommerce - Status Pedidos		
Pedidos pendentes 24	Pedidos entregues 125	Pedidos cancelados 12
Valor pendente R\$ 26.600,00	Valor recebido R\$ 245.900,00	Valor total R\$ 272.500,00
SDS R\$ 26.600,00 ▲ 2%	WTD R\$ 24.000,00 ▼ 2%	JJD R\$ 672.000,00 ▲ 10%
Pedidos pendentes		
	Mario Silva 📅 12/02/2016 - 14:50	R\$ 2.345,00 >
	João Paula Moraes 📅 12/02/2016 - 15:12	R\$ 2.455,00 >
	Monica Fransoneze 📅 12/02/2016 - 15:50	R\$ 1.200,00 >

Design: [Amol Hadkar](#) - Código fonte: [GitHub](#)

Crie um novo aplicativo chamado appDashboard:

```
ionic start appDashboard blank
```

Como este aplicativo é mais extenso vamos adicionando os elementos HTML por partes. Na área de conteúdo vamos adicionar um título e três linhas com três colunas:

```
<ion-content>
  <h1>FábioCommerce - Status Pedidos</h1>

  <div class="row">
    <div class="col">
      <label>Pedidos pendentes</label>
      <h2 class="light">24</h2>
    </div>
    <div class="col">
      <label>Pedidos entregues</label>
      <h2 class="balanced">125</h2>
    </div>
    <div class="col">
      <label>Pedidos cancelados</label>
      <h2 class="assertive">12</h2>
    </div>
  </div>
  <div class="row valores">
    <div class="col">
      <label>Valor pendente</label>
      <h2 class="light"><span>R$</span> 26.600,00</h2>
    </div>
    <div class="col">
      <label>Valor recebido</label>
      <h2 class="light"><span>R$</span> 245.900,00</h2>
    </div>
    <div class="col">
      <label>Valor total</label>
      <h2 class="light"><span>R$</span> 272.500,00</h2>
    </div>
  </div>
  <div class="row valores">
    <div class="col">
      <label>Valor pendente <span><i class="icon ion-arrow-up-b balanced"></i>
2%</span></label>
      <h2 class="light"><span>R$</span> 26.600,00</h2>
    </div>
    <div class="col">
      <label>WTD <span><i class="icon ion-arrow-down-b assertive"></i>
2%</span></label>
      <h2 class="light"><span>R$</span> 24.000,00</h2>
    </div>
    <div class="col">
```

```

        <label>Valor total <span><i class="icon ion-arrow-up-b balanced"></i>
10%</span></label>
        <h2 class="light"><span>R$</span> 672.000,00</h2>
    </div>
</div>

```

Neste primeiro passo apenas adicionamos as linhas e colunas da nossa grid. Nas linhas de valores adicionamos a classe valores para ser estilizada de forma diferente da primeira. Agora vamos adicionar uma lista com avatar:

```

<h2 class="title-custom">Pedidos pendentes</h2>
<ion-list>
    <a class="item item-avatar">
        
        <div class="right">
            <div class="valor">R$ 2.345,00</div>
            <i class="icon ion-ios-arrow-right"></i>
        </div>
        <h2>Mario Silva</h2>
        <p><i class="icon ion-calendar"></i> 12/02/2016 - 14:50</p>
    </a>
    <a class="item item-avatar">
        
        <div class="right">
            <div class="valor">R$ 2.455,00</div>
            <i class="icon ion-ios-arrow-right"></i>
        </div>
        <h2>João Paula Moraes</h2>
        <p><i class="icon ion-calendar"></i> 12/02/2016 - 15:12</p>
    </a>
    <a class="item item-avatar">
        
        <div class="right">
            <div class="valor">R$ 1.200,00</div>
            <i class="icon ion-ios-arrow-right"></i>
        </div>
        <h2>Monica Fransoneze</h2>
        <p><i class="icon ion-calendar"></i> 12/02/2016 - 15:50</p>
    </a>
    <a class="item item-avatar">
        
        <div class="right">
            <div class="valor">R$ 5.980,00</div>
            <i class="icon ion-ios-arrow-right"></i>
        </div>
        <h2>José Luiz</h2>
        <p><i class="icon ion-calendar"></i> 12/02/2016 - 15:52</p>
    </a>
    <a class="item item-avatar">
        
        <div class="right">

```

```

        <div class="valor">R$ 800,00</div>
        <i class="icon ion-ios-arrow-right"></i>
    </div>
    <h2>Maria Vitória</h2>
    <p><i class="icon ion-calendar"></i> 12/02/2016 - 15:58</p>
</a>
</ion-list>
</ion-content>

```

Apenas adicionamos um classe para o título da listagem e uma lista com alguns itens. Nosso app ficou assim até agora:



No style.css vamos primeiro alterar o cor de fundo e o estilo do nosso título:

```

.pane {
    background-color: #3B3C3E;
}
ion-content h1{
    color: #fff;
    font-size: 20px;
    padding: 10px;
}

```

Deixamos nosso app com o fundo escuro e o título branco. Vamos agora selecionar os elementos da grid e aplicar alguns estilos para ter uma visualização mais coordenada destacando os itens mais importante:

```

.row {
    padding: 0;
}

```

```

.col {
  margin: 1px;
  margin-bottom: 6px;
  background-color: #525659;
  padding: 10px;
}
.row label{
  color: #9CA0A3;
  font-size: 12px;
}
.row label span {
  float: right;
  color: #fff;
  padding-top: 4px;
}
.row h2 {
  margin: 0;
  font-size: 25px;
}
.row h2 span {
  display: block;
  font-size: 14px;
  font-weight: 400;
}
.row.valores h2 {
  font-size: 18px;
  font-weight: 600;
}

```

Ajustamos cores, posicionamento e tamanhos dos elementos da grid. Agora vamos customizar a listagem para seguir as cores do nosso tema escuro:

```

.title-custom {
  color: #8C9093;
  font-size: 18px;
  padding-left: 10px;
}
.item {
  background:none;
  border-color: #474747;
}
.item h2 {
  color: #8C9093;
}
.item p {
  color: #8C9093;
}

```

Removemos a cor de fundo dos itens da lista e estilizamos o título da listagem. Para concluir nosso app vamos posicionar os valores da listagem do lado direito:


```
.right {  
  float: right;  
  color: #8C9093;  
  font-size: 18px;  
  font-weight: bold;  
  margin-top: 10px;  
}  
.right .valor {  
  float: left;  
  margin-right: 10px;  
}  
.right i {  
  font-size: 40px;  
  float: left; margin-top: -10px;  
}
```

A div com a classe right foi posicionado ao lado direto dos itens.

Customizando formulários

Os formulários sempre foram um desafio para designers de interface, não basta ter uma boa aparência, precisa ser prático e rápido, pois os usuários não gostam de perder tempo preenchendo muitas informações.

The screenshot shows a mobile application interface for 'PassagemApp'. At the top, there's a blue header with a hamburger menu icon, the app name 'PassagemApp', and a 'Próximo' button. Below the header, a blue curved bar indicates 'PASSO 1/3'. The main content area is white and contains four sections: 1. 'Tipo da viagem?' with three buttons: 'Ida e volta' (selected, blue), 'Somente ida', and 'Somente volta'. 2. 'Vai com quem?' with two buttons: 'Sozinho' (selected, blue) and 'Acompanhado'. 3. 'Qual sua rota?' with two input fields. The first field contains 'Maringá-PR' and has a location pin icon. The second field is empty with the placeholder text 'Informe o local de destino' and a location pin icon. 4. 'Quando vai ser?' with the label 'Data ida' and two input fields. The first field has a calendar icon, and the second field has a clock icon.

Design: [Angelique Calmon](#) - Código fonte: [GitHub](#)

Crie um novo aplicativo chamado appPassagem.

```
ionic start appPassagem blank
```

Um formulário tem vários componentes, então vamos dividir em partes a criação dos elementos HTML. Primeiro vamos adicionar o título e os elementos de seleção:

```
<ion-content>
  <h1>PASSO 1/3</h1>

  <div class="forms">
    <h3>Tipo da viagem?</h3>
    <div class="row radio-custom">
      <div class="col select">Ida e volta</div>
      <div class="col">Somente ida</div>
      <div class="col">Somente volta</div>
    </div>

    <h3>Vai com quem?</h3>
    <div class="row radio-custom">
      <div class="col select">Sozinho</div>
      <div class="col">Acompanhado</div>
    </div>

    <h3>Qual sua rota?</h3>
    <div class="item item-input">
      <input type="text" placeholder="Informe o local de partida"
value="Maringá-PR">
      <i class="icon ion-ios-location-outline"></i>
    </div>
    <div class="item item-input">
      <input type="text" placeholder="Informe o local de destino">
      <i class="icon ion-ios-location-outline"></i>
    </div>
```

Criamos um título e uma grid com algumas linhas e colunas, mas para este *layout* vamos estilizar a grid para se comportar de forma diferente, onde cada coluna será um campo de seleção. Agora vamos adicionar a grid com os campos de data:

```
<h3>Quando vai ser?</h3>
<h4>Data ida</h4>
<div class="row row-input">
  <div class="col">
    <div class="item item-input">
      <input type="date">
      <i class="icon ion-ios-calendar-outline"></i>
    </div>
  </div>
  <div class="col">
    <div class="item item-input">
```

```

        <input type="time">
        <i class="icon ion-ios-clock-outline"></i>
    </div>
</div>
</div>
<h4>Data volta</h4>
<div class="row row-input">
    <div class="col">
        <div class="item item-input">
            <input type="date">
            <i class="icon ion-ios-calendar-outline"></i>
        </div>
    </div>
    <div class="col">
        <div class="item item-input">
            <input type="time">
            <i class="icon ion-ios-clock-outline"></i>
        </div>
    </div>
</div>

```

E para concluir nossos elementos HTML vamos adicionar uma lista de checkbox:

```

<h3>Passeios adicionais</h3>
<ion-list class="list-custom">
    <ion-item class="item-checkbox item-checkbox-right">
        <label class="checkbox">
            <input type="checkbox">
        </label>
        Visita ao Corcovado
    </ion-item>
    <ion-item class="item-checkbox item-checkbox-right">
        <label class="checkbox">
            <input type="checkbox">
        </label>
        Castelo da Sapolientra
    </ion-item>
    <ion-item class="item-checkbox item-checkbox-right">
        <label class="checkbox">
            <input type="checkbox">
        </label>
        Lago Santo River Side
    </ion-item>
    <ion-item class="item-checkbox item-checkbox-right">
        <label class="checkbox">
            <input type="checkbox">
        </label>
        Visita ao Corcovado
    </ion-item>
    <ion-item class="item-checkbox item-checkbox-right">
        <label class="checkbox">

```

```

        <input type="checkbox">
      </label>
      Castelo da Sapolientra
    </ion-item>
    <ion-item class="item-checkbox item-checkbox-right">
      <label class="checkbox">
        <input type="checkbox">
      </label>
      Lago Santo River Side
    </ion-item>
  </ion-list>

</div>
</ion-content>

```

Na lista não vamos customizar muita coisa do padrão do Ionic, apenas ajustes nas cores. Nosso app ficou assim até agora:

The screenshot shows a mobile application interface for 'PassagemApp'. At the top, there is a blue header bar with a hamburger menu icon on the left, the text 'PassagemApp' in the center, and a 'Próximo' button on the right. Below the header, the main content area has a white background. It starts with the title 'PASSO 1/3' in large, bold, black letters. Below this is the section 'Tipo da viagem?' with three radio button options: 'Ida e volta', 'Somente ida', and 'Somente volta'. The next section is 'Vai com quem?' with two radio button options: 'Sozinho' and 'Acompanhado'. This is followed by 'Qual sua rota?' which contains two input fields with location icons on the right; the first field contains 'Maringá-PR' and the second is a placeholder 'Informe o local de destino'. The final section is 'Quando vai ser?' with a label 'Data ida' above a date picker component showing 'dd/mm/aaaa' and a time picker component showing '--:--'.

Para este *layout* vamos utilizar dois tipos de fontes diferentes, a Pacifico e Lato. Instale as fontes utilizando o getfont:

```

getfont ' <link
href="https://fonts.googleapis.com/css?family=Lato:300,400,900|Pacifico"
rel="stylesheet">' --ionic

```

No style.css vamos começar pelo header alterando a cor e a fonte utilizada:

```
.bar.bar-positive.bar-top {
    background-color: #00A5FF;
    background-image: none;
}
.bar-top .title {
    font-family: 'Pacifico', cursive;
}
```

Alteramos a cor azul do tema do Ionic para um azul mais vivo e alteramos a fonte do título do header para a fonte Pacifico. Agora vamos customizar os títulos do formulário:

```
ion-content {
    font-family: 'Lato', sans-serif;
}
ion-content h1 {
    font-family: 'Lato', sans-serif;
    background-color: #00A5FF;
    color: #fff;
    text-align: center;
    font-size: 18px;
    font-weight: 800;
    -webkit-border-bottom-right-radius: 50%;
    -webkit-border-bottom-left-radius: 50%;
    -moz-border-radius-bottomright: 50%;
    -moz-border-radius-bottomleft: 50%;
    border-bottom-right-radius: 50%;
    border-bottom-left-radius: 50%;
    padding: 20px;
}
ion-content h3 {
    font-weight: 300;
    font-size: 25px;
    text-align: center;
    color: #595959;
    font-family: 'Lato', sans-serif;
}
h4 {
    color: #595959;
    padding-left: 15px;
    font-family: 'Lato', sans-serif;
    margin: 0;
}
```

Primeiro definimos a fonte Lato para toda área de conteúdo do aplicativo depois customizamos o h1 para ter os cantos arredondados unindo-se com o header, e ajustamos o estilo dos outros títulos. Agora vamos customizar as colunas de nossa grid para se comportar como um componente de seleção:

```

.forms {
    padding: 20px;
}
.row {
    padding: 2px;
}
.col.select {
    background-color: #00A5FF;
    color: #fff;
    border-radius: 15px;
}
.radio-custom {
    background-color: #F2F2F2;
    border-radius: 15px;
    color: #999999;
}
.radio-custom .col {
    text-align: center;
    padding-top: 10px;
    padding-bottom: 10px;
    font-family: 'Lato', sans-serif;
}

```

Para não ficar colado nas laterais demos um espaçamento interno em todo formulário, e ajustamos as cores das colunas. Para a coluna que representa o item selecionado foi adicionado alguns estilos próprios para se diferenciar dos demais. Vamos customizar os campos de entrada de dados do formulário:

```

.item {
    border-radius: 15px;
    border-color: #BEBEBE;
    margin-bottom: 20px;
}
.item input {
    color: #00A5FF;
    font-family: 'Lato', sans-serif;
}
.item-input i {
    font-size: 25px;
    margin-right: 15px;
    color: #BEBEBE;
}
.item-input input {
    padding-right: 0;
}
.row-input {
    padding: 0;
}
.row-input .col {
    padding-top: 0;
}

```

```
padding-bottom: 0;  
}
```

Para os itens e campos ajustamos cores, espaçamento e tamanho de fonte. Para concluir vamos customizar a listagem dos checkbox:

```
.list-custom .item {  
  border-radius: 0;  
  border-left-width: 0;  
  border-right-width: 0;  
}  
.list-custom .item {  
  color: #00A5FF;  
  margin: 0;  
  margin-top: -1px;  
}  
.checkbox input:checked:before, .checkbox input:checked + .checkbox-icon:before {  
  background: #00A5FF;  
  border-color: #00A5FF;  
}  
.checkbox input:before, .checkbox .checkbox-icon:before {  
  border-color: #00A5FF;  
}
```

Ajustamos a lista para não ter bordas arredondadas como as demais e ajustamos o estilo do checkbox.

Header transparente e scroll vertical

A disposição criativa dos elementos trazem uma rica experiência visual para os aplicativos, estes estilos são simples de implementar mas podem fazer uma grande diferença quando bem aplicados, mas tenha cuidado pois se utilizado de forma incorreta a navegabilidade do aplicativo pode ser totalmente comprometida.



Design: [Andreea Nicolaescu](#) - Código fonte: [GitHub](#)

Crie um novo aplicativo chamado appPersona.

ionic start appPersona blank

Vamos aos elementos HTML, adicione no header uma classe para ser selecionado no CSS de forma mais fácil:

```
<ion-pane>
  <ion-header-bar class="bar-transparent">
    <button class="button button-icon ion-ios-arrow-left"></button>
    <h1 class="title"></h1>
    <button class="button button-icon ion-ios-heart-outline"></button>
  </ion-header-bar>
```

A classe `bar-transparent` será utiliza para deixar nosso header sem cor de fundo. Agora vamos a parte mais complexa, após o header adicione o início da área de conteúdo com seus elementos:

```
<ion-content class="no-header">
  <div id="preview" style="background-image: url(img/angelina_jolie3.jpg)">
    <div class="info-preview">
      <h2>Angelina Jolie, 40</h2>
      <h3 class="right">
        <i class="icon ion-image"></i>
        3/8
      </h3>
      <h3>
        <i class="icon ion-ios-location-outline"></i>
        Los Angeles, Califórnia, EUA
      </h3>
      <ion-scroll direction="x" class="wide-as-needed">
        <div class="photo" style="background-image:
url(img/angelina_jolie1.jpg);"></div>
        <div class="photo" style="background-image:
url(img/angelina_jolie2.jpg);"></div>
        <div class="photo selected" style="background-image:
url(img/angelina_jolie3.jpg);"></div>
        <div class="photo" style="background-image:
url(img/angelina_jolie4.jpg);"></div>
        <div class="photo" style="background-image:
url(img/angelina_jolie5.jpg);"></div>
        <div class="photo" style="background-image:
url(img/angelina_jolie6.jpg);"></div>
        <div class="photo" style="background-image:
url(img/angelina_jolie7.jpg);"></div>
        <div class="photo" style="background-image:
url(img/angelina_jolie8.jpg);"></div>
      </ion-scroll>
    </div>
  </div>
```

No `ion-content` adicionamos a classe `no-header` para que o Ionic ignore o espaço reservado para a barra superior fixa, como o header ficará transparente a imagem da área de conteúdo será visível. Criamos uma div com o id `preview` para ser mais fácil a seleção do elemento via CSS, perceba que as imagens são declaradas inline utilizando a propriedade `style`. Fazemos assim porque em um aplicativo real as imagens serão dinâmicas e estando no HTML podemos manipular pelo JavaScript, mas não se preocupe com isso neste momento. Agora vamos adicionar o restante do nossos elementos:

```
<h4>Sobre Angelina Jolie</h4>
```

```
<p>Angelina Jolie Pitt é uma atriz, dubladora, diretora, produtora, roteirista e ativista humanitária americana. Ela já foi premiada com um Óscar, dois Prêmios Screen Actors Guild, e três Prêmios Globo de Ouro.</p>
```

```
<h4>Filmes</h4>
```

```







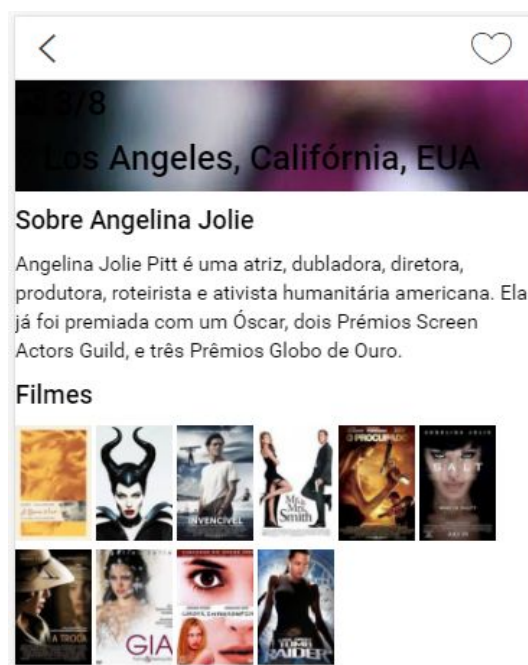



```

```
</ion-content>
```

```
</ion-pane>
```

Adicionamos apenas algumas imagens com a classe `filme`. Nosso aplicativo está assim:



Para este app vamos utilizar as fontes Lato e Poiret One, faça a instalação delas pelo getfont:

```
getfont '<link  
href="https://fonts.googleapis.com/css?family=Lato:300,400,900|Poiret+On  
e" rel="stylesheet">' --ionic
```

Agora em style.css vamos iniciar pelo header:

```
.bar-transparent {  
    background: none;  
}  
.bar-transparent button {  
    color: #fff;  
}
```

Apenas deixamos nosso header sem cor de fundo e deixamos nossos botões brancos. Agora vamos ajustar a área de preview:

```
#preview {  
    background-size: cover;  
    height: 60%;  
    background-position: center;  
    position: relative;  
}  
.info-preview {  
    width: 100%;  
    position: absolute;  
    bottom: 0;  
    text-shadow: 1px 1px 0 rgba(0,0,0,0.8);  
}  
.photo {  
    width: 70px;  
    height: 70px;  
    background-color: #000;  
    display: inline-block;  
    margin-left: 6px;  
    margin-bottom: 0;  
    background-size: cover;  
    background-position: center;  
    opacity: 0.8;  
}  
.photo.selected {  
    border: 2px solid #fff;  
    opacity: 1;  
}  
.photo:last-child {  
    margin-right: 6px;  
}
```

Estilizamos o plano de fundo da div preview para cobrir toda área útil e deixamos as fotos da galeria com tamanhos únicos, apenas a div que estiver com a classe selected terá um tratamento diferente com cor branca de borda e sem transparência. Neste momento a galeria de fotos ainda não teve o comportamento esperado, vamos ajustar seu scroll:

```
.wide-as-needed {
  overflow: scroll;
  white-space: nowrap;
  display: inline-block;
  width: 100%;
  margin-top: 10px;
}
.scroll {
  min-width: 100%;
}
```

Apenas ajustamos o scroll para se comportar de forma horizontal e expansiva. Para concluir vamos ajustar os demais títulos e os filmes:

```
ion-content {
  background-color: #F0F2F5;
}
ion-content h2 {
  font-family: 'Poiret One', cursive;
  color: #fff;
  font-weight: bold;
  margin: 0;
  margin-left: 20px;
}
h3.right {
  float: right;
  margin: 0;
  margin-right: 20px;
}
ion-content h3 {
  color: #fff;
  font-family: 'Lato', sans-serif;
  font-size: 16px;
  margin: 0;
  margin-left: 20px;
}
h4 {
  color: #2E3037;
  font-family: 'Lato', sans-serif;
  padding-left: 10px;
  padding-right: 10px;
  margin-bottom: 5px;
  margin-top: 20px;
}
```

```
p {  
  color: #2E3037;  
  font-family: 'Lato', sans-serif;  
  padding-left: 10px;  
  padding-right: 10px;  
}  
.filme {  
  margin-left: 10px;  
}
```

Aplicamos diferentes estilos para cada tipo de título e adicionamos uma pequena margem entre cada capa de filme apresentada.

Ícones e novos componentes

O Ionic disponibiliza uma coleção de ícones baseados em fonte bem interessante, mas em alguns momentos você pode precisar de outras fontes e vamos treinar esta necessidade. Também vamos criar um componente do zero. Escolhi um componente de timeline por ser aparentemente algo mais complexo, mas depois de implementado fica simples seu entendimento.



Design: [Daniel Kloppe](#) - Código fonte: [GitHub](#)

Crie um novo aplicativo chamado appTimeline:

```
ionic start appTimeline blank
```

Existem inúmeros sites que disponibilizam coleções de ícones baseado em fontes, os que mais utilizo são <http://flaticon.com>, <http://iconmonstr.com> e os ícones do [MaterialDesign](#).

Para este layout vamos utilizar os ícones do Material Design e a fonte Ubuntu do google fonts. Para instalar os dois podemos utilizar o getfont:

```
getfont  
'https://fonts.googleapis.com/icon?family=Material+Icons|Ubuntu:300,400,700' --ionic
```

Vamos começar adicionado um header com dois ícones do material design:

```
<ion-pane>  
  <ion-header-bar class="bar-assertive">  
    <button class="button button-clear">  
      <i class="material-icons">menu</i>  
    </button>  
    <h1 class="title"></h1>  
    <button class="button button-clear">  
      <i class="material-icons">chat_bubble_outline</i>  
    </button>  
  </ion-header-bar>
```

No site do material design é possível ver quais ícones podem ser utilizado. Vamos adicionar uma div com o id profile com algum elementos:

```
<ion-content>  
  <div id="profile">  
    <div class="photo">  
        
    </div>  
    <h2>Linus Torvalds</h2>  
    <h3>Criador do Linux</h3>  
    <div class="row">  
      <div class="col">1056 <span>Posts</span></div>  
      <div class="col">12k <span>Seguidores</span></div>  
      <div class="col">4533 <span>Seguindo</span></div>  
    </div>  
  </div>
```

E por fim vamos adicionar o elemento timeline que ainda não existe, mas vamos implementa-lo em breve:


```

<div class="timeline">

  <div class="event">
    <i class="material-icons mark">location_on</i>
    <div class="card padding">
      <p>Esteve No <b>Brasil</b> no evento <b>LinuxCon 2011</b>.</p>
      <div class="info">
        <i class="material-icons">access_time</i>
        12:00am
      </div>
      <div class="info">
        <i class="material-icons">date_range</i>
        18/11/2011
      </div>
    </div>
  </div>

  <div class="event">
    <i class="material-icons mark">call</i>
    <div class="card padding">
      <p>Ligou para <b>Bill Gates</b> falando sobre idéias para 2016.</p>
      <div class="info">
        <i class="material-icons">access_time</i>
        8:30am
      </div>
      <div class="info">
        <i class="material-icons">date_range</i>
        22/11/2011
      </div>
    </div>
  </div>

  <div class="event">
    <i class="material-icons mark">bookmark</i>
    <div class="card padding">
      <p>Foi marcado na foto de <b>Mike Flor</b>.</p>
      <p class="item-image">
        
      </p>
      <div class="info">
        <i class="material-icons">access_time</i>
        22:11pm
      </div>
      <div class="info">
        <i class="material-icons">date_range</i>
        12/12/2011
      </div>
    </div>
  </div>
</div>

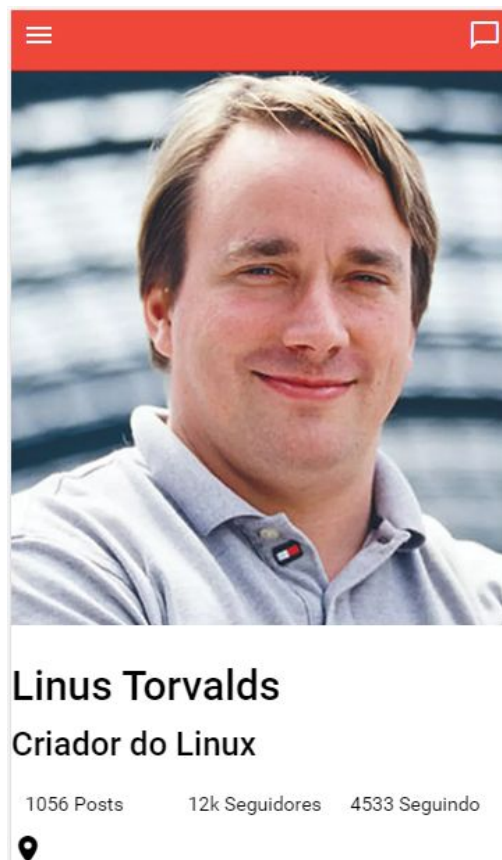
```

```

<div class="event">
  
  <div class="card padding">
    <p>Aceitou novo seguidor <b>Fábio Rogério SJ</b>.</p>
    <div class="info">
      <i class="material-icons">access_time</i>
      14:22pm
    </div>
    <div class="info">
      <i class="material-icons">date_range</i>
      15/12/2011
    </div>
  </div>
</div>
</div>
</ion-content>
</ion-pane>

```

Criamos uma div com a classe `timeline` e outras divs filhas com a classe `event`, neste caso cada evento é um item na listagem do componente. Até o momento nosso app está assim:



Agora vamos ao `style.css`. Primeiro customizamos os títulos e o header:

```

.pane, h1, h2, h3, p {
    font-family: Ubuntu;
}
h2 {
    color:#fff;
    margin:0;
}
h2 + h3 {
    color:#ffb2b9;
    margin:0;
    font-weight:800;
    font-size:22px;
}
.pane {
    background-color:#687586;
}

.material-icons {
    margin-top: 4px;
}
.bar.bar-assertive {
    border-color:#ff6861;
    background-color:#ff6861;
    background-image: none;
}

```

Definimos para os títulos que a fonte desejada é a Ubuntu, alteramos as cores e tamanhos e ajustamos o estilo do header. Vamos estilizar o profile:

```

#profile {
    background-color:#ff6861;
    color:#fff;
    text-align:center;
    text-shadow:1px 1px 2px #d2544d;
}
#profile .photo {
    background-color:#ed6b63;
    display:inline-block;
    border-radius:50%;
    padding:6px;
    padding-bottom:0;
}
#profile img {
    width:150px;
    border-radius:50%;
    border:6px solid #fff;
}

```

Aplicamos uma cor de destaque no fundo do profile e customizamos a imagem para ficar arredondada e com bordas de destaque. Vamos ajustar as colunas da grid para destacar os textos:

```
.row {
  margin-top:20px;
  padding:0;
}
.col {
  padding:10px;
  font-weight:800;
  font-size:20px;
}
.col span {
  display:block;
  font-weight:300;
  font-size:16px;
}
```

Aumentamos a fonte e ajustamos os posicionamentos. Vamos ao componente:

```
.timeline {
  padding-left:20px;
  padding-top:10px;
  background: #687586;
  background: linear-gradient(180deg, transparent, #5b6776, transparent);
  background-position: 45px;
  background-repeat: repeat-y;
  background-size: 8px 1px;
}
```

Primeiro adicionamos uma linha que servirá de base para a timeline. Vamos ajustar o componente card:

```
.card {
  border-radius:6px;
}
.card:before {
  content:"";
  width: 0;
  height: 0;
  border-top: 10px solid transparent;
  border-bottom: 10px solid transparent;
  border-right:10px solid #fff;
  position:absolute;
  margin-left:-20px;
}
```

Os cards ficaram com as bordas mais arredondadas e adicionamos uma seta para a esquerda. Vamos criar os estilos para as marcações:

```

.mark {
  background-color:#2bb9bb;
  border-radius:50%;
  padding:10px;
  border:4px solid #fff;
  color:#fff;
  float:left;
  margin-right:25px;
  margin-top:-5px;
}
.mark-profile {
  width:52px;
  border-radius:50%;
  border:4px solid #fff;
  float:left;
  margin-right:25px;
  margin-top:-5px;
}

```

Deixamos cada marcação redonda e posicionada na linha da timeline. Por fim ajustamos os elementos do card:

```

.event:last-child {
  background-color:#687586;
}
.card .info {
  color:#cccccc;
  font-weight:300;
  float:left;
  margin-right:10px;
}
.card .info i {
  font-size:14px;
  float:left;
}

```

Deixamos as informações menores e com uma cor de menos destaque.

Explore o universo web

Uma das vantagens de desenvolver aplicações híbridas é a possibilidade de utilizar qualquer coisa que a web permite, não dependendo das limitações da linguagem nativa. Neste último aplicativo vamos utilizar um código de terceiro para criar um aplicativo com interface rica.



Código original: [Kseso](#) - Código fonte: [GitHub](#)

Crie nosso último aplicativo deste material com o nome appFoods.

```
ionic start appFoods blank
```

Vamos utilizar a fonte Ubuntu, sendo assim siga os passos para realizar a instalação pelo getfont.

Quando você pretende usar um código de terceiro alguns ajustes são necessário tanto para melhorar a legibilidade do código quanto para ajustar margem, espaçamentos, cores, etc.

O código do gráfico, desenvolvido por [Kseso](#), foi ajustado para nosso aplicativo com os seguintes elementos HTML:

```
<ion-pane>
  <ion-header-bar class="bar-dark" align-title="center">
    <button class="button button-icon ion-navicon"></button>
    <h1 class="title">FOOD DATA</h1>
  </ion-header-bar>
  <ion-content>
    <h1 class="title-food">BAR DO ZÉ <i class="icon
ion-ios-wineglass-outline"></i></h1>
    <div class="donut-chart-block block">
      <h2>Consumo e avaliação</h2>
      <div class="donut-chart">
        <div id="porcion1" class="recorte"><div class="quesito bebida"
data-rel="21"></div></div>
        <div id="porcion2" class="recorte"><div class="quesito refeicao"
data-rel="39"></div></div>
        <div id="porcion3" class="recorte"><div class="quesito win"
data-rel="31"></div></div>
        <div id="porcionFin" class="recorte"><div class="quesito linux"
data-rel="9"></div></div>
        <p class="center-date">DOMINGO<br><span
class="scnd-font-color">01/05</span></p>
      </div>
      <ul class="os-percentages horizontal-list">
        <li>
          <p class="bebida os scnd-font-color">Bebida</p>
          <p class="os-percentage">21<sup>%</sup></p>
        </li>
        <li>
          <p class="refeicao os scnd-font-color">Refeição</p>
          <p class="os-percentage">39<sup>%</sup></p>
        </li>
        <li>
          <p class="porcao os scnd-font-color">Porção</p>
          <p class="os-percentage">9<sup>%</sup></p>
        </li>
      </ul>
    </div>
  </ion-content>
```

```

        <p class="lanche os scnd-font-color">Lanche</p>
        <p class="os-percentage">31<sup>%</sup></p>
    </li>
</ul>
</div>

<div class="bar-avalicacao">
    
    <div class="bar-base">
        <span class="n-avaliacao">19</span>
        <div id="bar1"></div>
    </div>
</div>

<div class="bar-avalicacao">
    
    <div class="bar-base">
        <span class="n-avaliacao">29</span>
        <div id="bar2"></div>
    </div>
</div>

<div class="bar-avalicacao">
    
    <div class="bar-base">
        <span class="n-avaliacao">13</span>
        <div id="bar3"></div>
    </div>
</div>

<div class="bar-avalicacao">
    
    <div class="bar-base">
        <span class="n-avaliacao">3</span>
        <div id="bar4"></div>
    </div>
</div>

</ion-content>
</ion-pane>

```

E o CSS:

```

.pane {
    background-color: #1E1B22;
    font-family: 'Ubuntu', sans-serif;
}
.bar.bar-dark {
    background-color: #1D1A20;
    background-image: none;
}

```



```

.bar.bar-dark .title {
    color: #FFFADA;
    font-family: 'Ubuntu', sans-serif;
}
.bar-dark .button,
.bar-dark .button:hover {
    color: #FFFADA;
}
h1.title-food {
    background-color: #2B2B2D;
    color: #FFFADA;
    font-weight: 300;
    padding: 10px;
    font-size: 30px;
    margin: 0;
    font-family: 'Ubuntu', sans-serif;
}
h1.title-food i {
    float: right;
    margin-right: 10px;
}
.donut-chart-block {
    background-color: #353238;
    color: #FFFADA;
    padding-top: 10px;
}
.donut-chart-block h2 {
    text-align: center;
    color: #FFFADA;
    font-weight: 300;
    margin-bottom: 10px;
    font-family: 'Ubuntu', sans-serif;
}
.horizontal-list {
    margin: 0;
    padding: 0;
    list-style-type: none;
}
.horizontal-list li {
    float: left;
}
.os-percentages li {
    width: 25%;
    border-left: 1px solid #3B363F;
    border-bottom: 1px solid #3B363F;
    text-align: center;
    background: #1C191F;
    color: #FFFADA;
}
.os {
    margin: 0;
}

```

```

        padding: 10px 0 5px;
        font-size: 15px;
    }
    .os.bebida {
        border-top: 4px solid #9528AA;
    }
    .os.refeicao {
        border-top: 4px solid #00AD9E;
    }
    .os.porcao {
        border-top: 4px solid #4FC4F6;
    }
    .os.lanche {
        border-top: 4px solid #FF6F51;
    }
    .os-percentage {
        margin: 0;
        padding: 0 0 15px 10px;
        font-size: 25px;
    }
    .donut-chart {
        position: relative;
        width: 200px;
        height: 200px;
        margin: 0 auto 20px;
        border-radius: 100%
    }
    p.center-date {
        background: #353238;
        position: absolute;
        text-align: center;
        font-size: 16px;
        top:0;left:0;bottom:0;right:0;
        width: 130px;
        height: 130px;
        margin: auto;
        border-radius: 50%;
        line-height: 35px;
        padding: 15% 0 0;
        font-weight: 300;
    }
    .center-date span.scnd-font-color {
        line-height: 0;
        font-size: 26px;
        font-weight: 600;
    }
    .recorte {
        border-radius: 50%;
        clip: rect(0px, 200px, 200px, 100px);
        height: 100%;
        position: absolute;

```

```

        width: 100%;
    }
    .quesito {
        border-radius: 50%;
        clip: rect(0px, 100px, 200px, 0px);
        height: 100%;
        position: absolute;
        width: 100%;
        font-family: monospace;
        font-size: 1.5rem;
    }
    #porcion1 {
        transform: rotate(0deg);
    }

    #porcion1 .quesito {
        background-color: #9528AA;
        transform: rotate(76deg);
    }
    #porcion2 {
        transform: rotate(76deg);
    }
    #porcion2 .quesito {
        background-color: #00AD9E;
        transform: rotate(140deg);
    }
    #porcion3 {
        transform: rotate(215deg);
    }
    #porcion3 .quesito {
        background-color: #4FC4F6;
        transform: rotate(113deg);
    }
    #porcionFin {
        transform: rotate(-32deg);
    }
    #porcionFin .quesito {
        background-color: #FF6F51;
        transform: rotate(32deg);
    }
    .nota-final {
        clear: both;
        color: #4FC4F6;
        font-size: 1rem;
        padding: 2rem 0;
    }
    .nota-final strong {
        color: #E64C65;
    }
    .nota-final a {
        color: #FCB150;
    }

```

```

    font-size: inherit;
}
/*****
BAR-CHART
*****/
.grafico.bar-chart {
    background: #3468AF;
    padding: 0 1rem 2rem 1rem;
    width: 100%;
    height: 60%;
    position: relative;
    color: #FFFADA;
    font-size: 80%;
}
.bar-chart [class^='eje-'] {
    padding: 0 1rem 0 2rem;
    bottom: 1rem;
}
.bar-chart .eje-x {
    bottom: 0;
}
.bar-chart .eje-y li {
    height: 20%;
    border-top: 1px solid #FFFADA;
}
.bar-chart .eje-x li {
    width: 14%;
    position: relative;
    text-align: left;
}
.bar-chart .eje-x li i {
    transform: rotate(-45deg) translate(-1rem);
    transform-origin: 30% 60%;
    display: block;
}
.bar-chart .eje-x li:before {
    content: '';
    position: absolute;
    bottom: 1.9rem;
    width: 70%;
    right: 5%;
    box-shadow: 3px 0 rgba(0,0,0,.1), 3px -3px rgba(0,0,0,.1);
}
.bar-chart .eje-x li:nth-child(1):before {
    background: #E64C65;
    height: 570%;
}
.bar-chart .eje-x li:nth-child(2):before {
    background: #11A8AB;
    height: 900%;
}

```

```

.bar-chart .eje-x li:nth-child(3):before {
  background: #FCB150;
  height: 400%;
}
.bar-chart .eje-x li:nth-child(4):before {
  background: #4FC4F6;
  height: 290%;
}
.bar-chart .eje-x li:nth-child(5):before {
  background: #FFED0D;
  height: 720%;
}
.bar-chart .eje-x li:nth-child(6):before {
  background: #F46FDA;
  height: 820%;
}
.bar-chart .eje-x li:nth-child(7):before {
  background: #15BFCC;
  height: 520%;
}
.bar-avalicacao {
  padding: 10px;
}
.bar-avalicacao img {
  width: 30px;
  float: left;
}
.bar-avalicacao .bar-base {
  background-color: #343137;
  height: 30px;
  margin-left: 40px;
}
.bar-avalicacao .bar-base div {
  background-color: #00D8C8;
}
#bar1 {
  height: 30px;
  width: 30%;
}
#bar2 {
  height: 30px;
  width: 45%;
}
#bar3 {
  height: 30px;
  width: 20%;
}
#bar4 {
  height: 30px;
  width: 5%;
}

```

```
.n-avaliacao {  
  position: absolute;  
  font-size: 20px;  
  font-weight: 600;  
  color: #1E1B22;  
  margin-top: 4px;  
  margin-left: 4px;  
}
```

Você pode analisar o que foi alterado comparando os dois códigos, o original de Kseso e o do aplicativo appFoods.

Embora os aplicativos aqui desenvolvidos sejam simples, podemos analisar que não temos limites no desenvolvimento com Ionic, precisamos utilizar a criatividade e analisar bem o público-alvo do app.

Este não deve ser o final de seus aplicativos de aprendizagem, crie novos, se inspire em *layouts* de designers, faça um redesign de apps no mercado.

Lembre-se que a prática leva à perfeição.

Próximo passo

Este ebook abordou a customização e criação de componentes, o próximo passo é adicionar interação em nossas interfaces com o JavaScript.

Veja o [#3 Ionic Framework - JavaScript](#)

