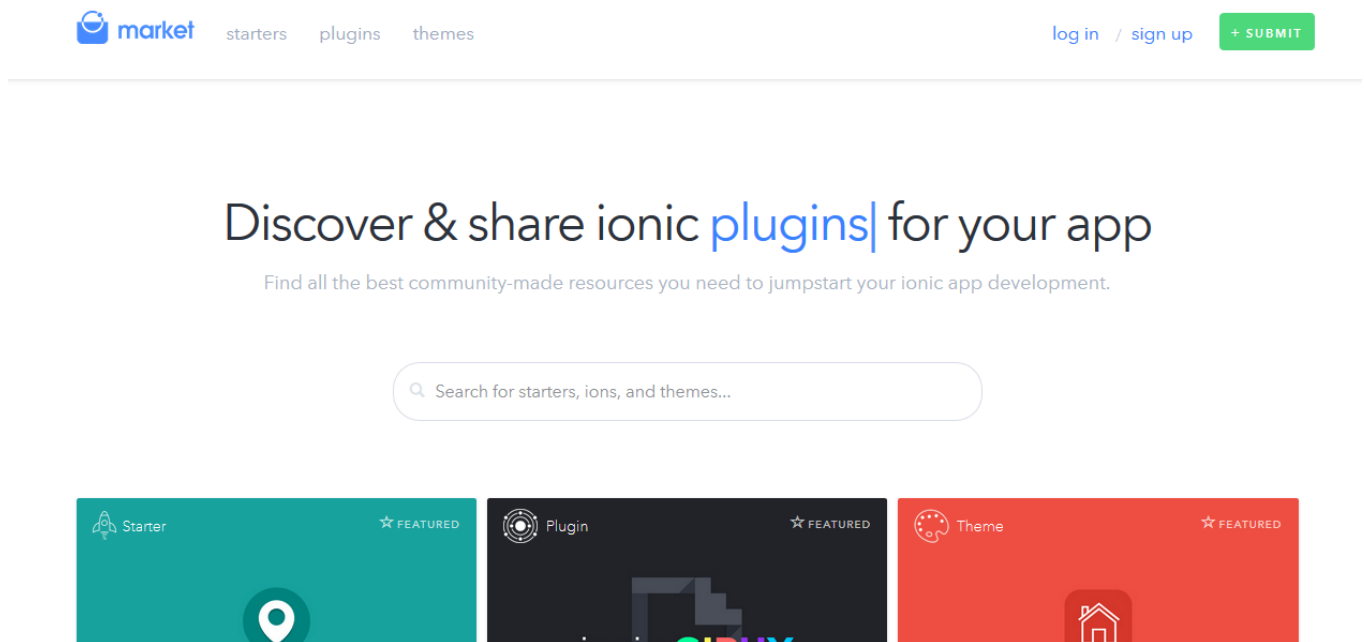


## Plugin Datepicker

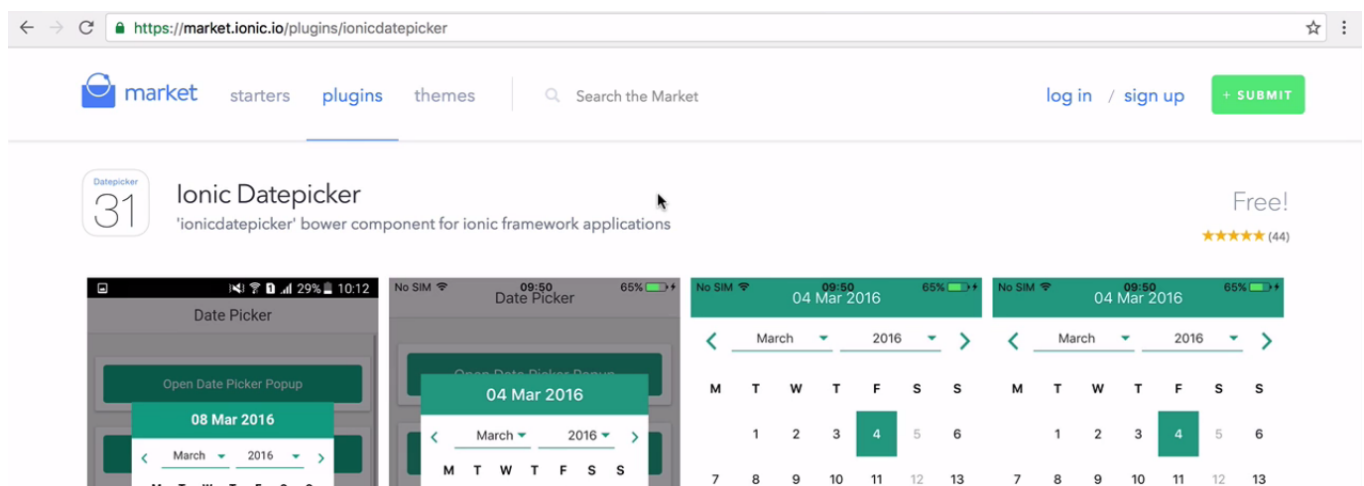
### Transcrição

O nosso cliente quer uma nova funcionalidade na nossa aplicação. Como estamos trabalhando com o agendamento do test drive, o usuário precisará visualizar a data em que realizará o agendamento. Logo, na tela "Estamos quase lá", adicionaremos um novo Input no qual surgirá um calendário para que o usuário escolha a data. Temos que tratar algumas regras de negócio que evitam a seleção de uma data errada. Nós podemos fazer isto manualmente ou usar um plugin. Ficaremos com a segunda opção.

Vamos encontrar o plugin no <https://market.ionic.io/> (https://market.ionic.io/) target=blank">Ionic Market. Além dos plugins, também encontraremos templates. Por exemplo, se você quiser criar uma aplicação para restaurante, você já encontrará alguns templates prontos.



Nós usaremos o plugin [ionicdatepicker](https://market.ionic.io/plugins/ionicdatepicker) (https://market.ionic.io/plugins/ionicdatepicker):



Ele abrirá um pop up com um calendário em que podemos seleccionar os meses e os anos, exatamente como queremos. Outra vantagem é que ele é gratuito, basta baixar e instalar. E logo abaixo, encontraremos a documentação que explicará

como instalá-lo, a forma de usá-lo e quais são os parâmetros. Temos uma `lib` parecida a que instalamos na câmera, mas felizmente poderemos testar no browser.

Seguiremos as instrução da documentação para realizar a instalação via `bower`. No Terminal, após verificarmos se estamos na pasta raiz, usaremos o seguinte comando:

```
AluraCar bower install ionic-datepicker --Salvaremos
```

Será feito o download das dependências. Em seguida, carregaremos a `lib` no arquivo `index.html` do nosso projeto. Iremos adicionar a nova `lib` abaixo da que é referente aos filtros. Adicionaremos o caminho para o script:

```
<script src="lib/ionic-datepicker/dist/ionic-datepicker.bundle.min.js"></script>
```

Já carregamos a `lib` no `index`, faremos o mesmo no `app.js`:

```
angular.module('starter', ['ionic', 'idf.br-filters', 'ngCordova', ionic-datepicker]) {  
  //  
}
```

Em seguida, vamos construir o Input no HTML e depois alteraremos o arquivo `finalizarPedido.controller.js`. No arquivo `finalizarpedido.html`, abaixo do `pedido.email`, adicionaremos um novo `label`:

```
<label class="item item-input item-stacked-label">  
  <span class="input-label">Data do Agendamento</span>  
  <input type="text" ng-focus="abrirPopupCalendario()">  
</label>
```

A função será chamada quando o campo ganhar o foco, por isso usamos o `ng-focus`. Depois, no `finalizarPedido.controller.js`.

```
angular.module('starter')  
.controller('FinalizarPedidoController', function($stateParams, $scope  
  , $ionicPopup, $state, CarroService, $ionicHistory, ionicDatePicker){  
  
  $scope.carroFinalizado = angular.fromJson($stateParams.carro);  
  
  $scope.pedido = {};  
  
  $scope.abrirPopupCalendario = function(){  
  
    var configuracoes = {  
      callback : function(data){  
        console.log(data);  
      }  
    }  
  }  
  ionicDatePicker.openDatePicker(configuracoes)  
}  
  
$scope.finalizarPedido = function(){
```

```
var pedidoFinalizado = {  
  params : {  
    carro : $scope.carroFinalizado.nome,  
    preco : $scope.carroFinalizado.preco,  
    nome : $scope.pedido.nome,  
    endereco : $scope.pedido.endereco,  
    email : $scope.pedido.email  
  }  
}
```

Como nós não tínhamos o objeto de `configuracoes`, criamos uma variável para ele. Para chamarmos a data, utilizamos o atributo `callback`.

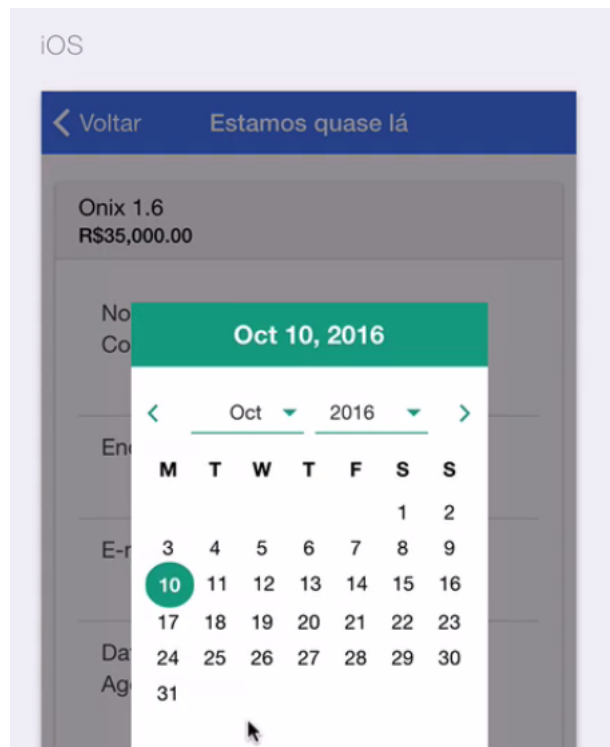
Vamos testar para ver como ficou até aqui. No Terminal, digitaremos:

```
AluraCar ionic serve -l
```

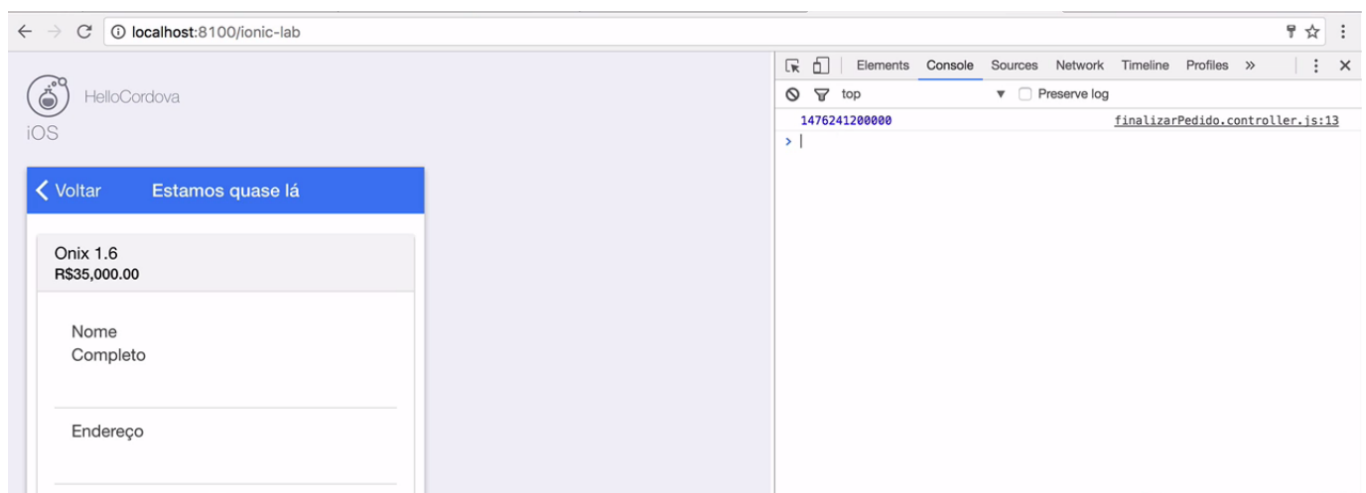
Nosso projeto começará a rodar e ao chegarmos na tela "Estamos quase lá", veremos o Input de agendamento.



Quando clicarmos para ganhar o foco, aparecerá o calendário.



No console, conseguiremos ver a data selecionada em milissegundos.



Agora, precisamos mostrar a data para o usuário. Para isto, usaremos o `ng-model` no input do `finalizar.pedido.html`.

```
<label class="item item-input item-stacked-label">
  <span class="input-label">Data do Agendamento</span>
  <input type="text" ng-model="dataSelecionada" ng-focus="abrirPopupCalendario()">
</label>
```

Criaremos a variável `dataSelecionada` no `finalizarPedido.controller.js`:

```
angular.module('starter')
.controller('FinalizarPedidoController', function($stateParams, $scope
, $ionicPopup, $state, CarroService, $ionicHistory, ionicDatePicker){

  $scope.carroFinalizado = angular.fromJson($stateParams.carro);

  $scope.pedido = {};
```

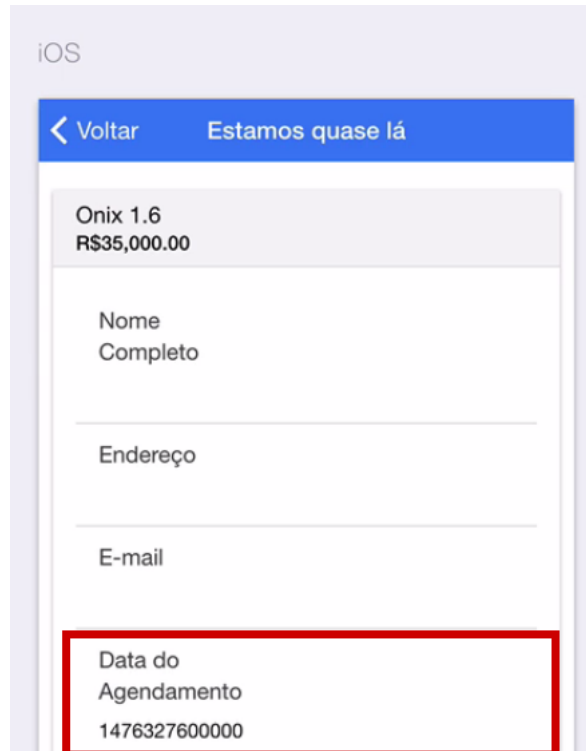
```
$scope.dataSelecionada;
```

```
$scope.abrirPopupCalendario = function(){  
  
    var configuracoes = {  
        callback : function(data){  
            $scope.dataSelecionada = data;  
        }  
    }  
    ionicDatePicker.openDatePicker(configuracoes)  
}
```

```
$scope.finalizarPedido = function(){  
  
    var pedidoFinalizado = {  
        params : {  
  
            carro : $scope.carroFinalizado.nome,  
            preco : $scope.carroFinalizado.preco,  
            nome : $scope.pedido.nome,  
            endereco : $scope.pedido.endereco,  
            email : $scope.pedido.email  
        }  
    }  
}
```

Substituímos o console, pela nova variável que passou a receber a `data` .

A data já poderá ser visualizada pelo usuário.



iOS

< Voltar Estamos quase lá

Onix 1.6  
R\$35,000.00

Nome  
Completo

Endereço

E-mail

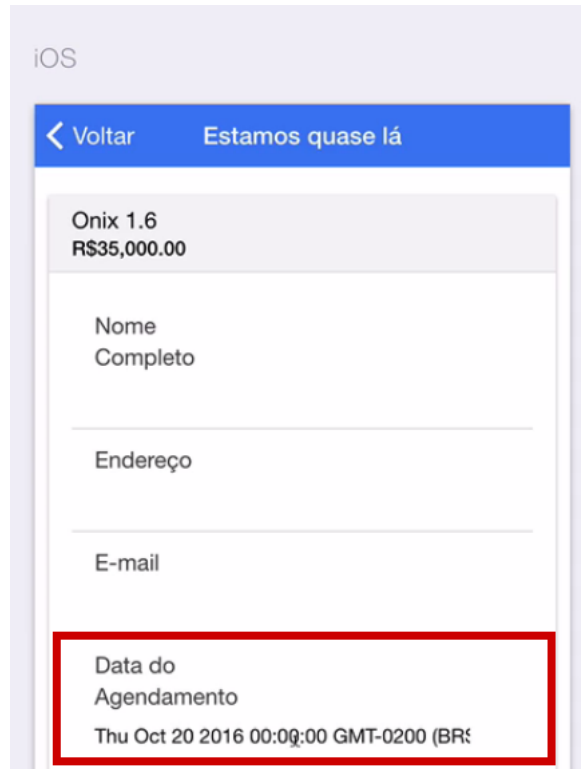
Data do  
Agendamento  
1476327600000

Mas o usuário vai ter dificuldade em entender a data. Não é o padrão que ele está acostumado nas aplicações do dia a dia. Vamos transformar em uma data de JavaScript. De volta ao arquivo `finalizarpedido.html` , adicionaremos o `new Date()` e passaremos a `data` como parâmetro.

```
$scope.abrirPopupCalendario = function(){

    var configuracoes = {
        callback : function(data){
            $scope.dataSelecionada = new Date(data);
        }
    }
}
```

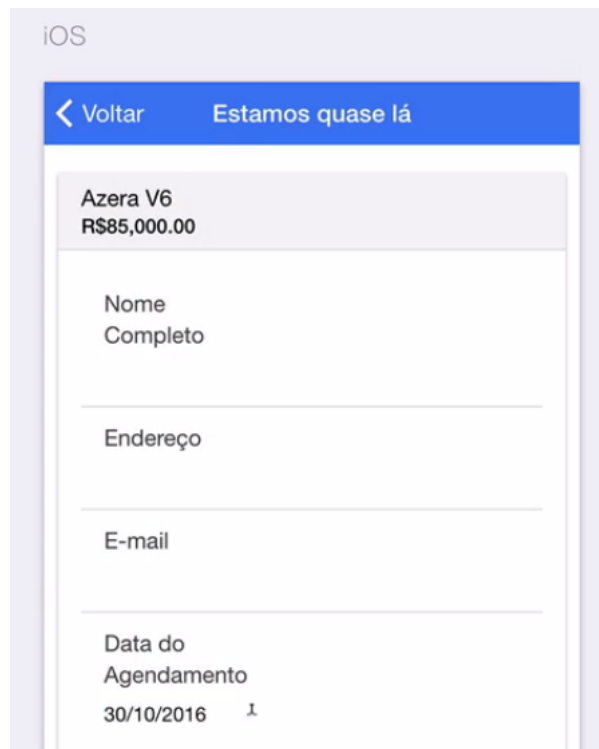
A data já será vista em outro formato.



Mas ainda não foi formatada... Podemos usar os filtros do Angular para fazer isto. No `finalizarpedido.html`, adicionaremos o filtro `date` com o padrão `dd (dia), MM (Mês), yyyy (ano)`:

```
<label class="item item-input item-stacked-label">
    <span class="input-label">Data do Agendamento</span>
    <input type="text" ng-model="dataSelecionada | date : 'dd/MM/yyyy' " ng-focus="abrirPopupCalendario"
</label>
```

A data será visualizada com o padrão que o usuário está habituado.



Nós podemos customizar o calendário para que os dias da semana fiquem em português. No `finalizaPedido.controller.js`, usaremos o `weekList` e passaremos como objeto os dias da semana `D` (domingo), `S` (segunda), `T` (terça), `Q` (quarta), `Q` (quinta), `S` (sexta) e `S` (sábado).

```
$scope.abrirPopupCalendario = function(){

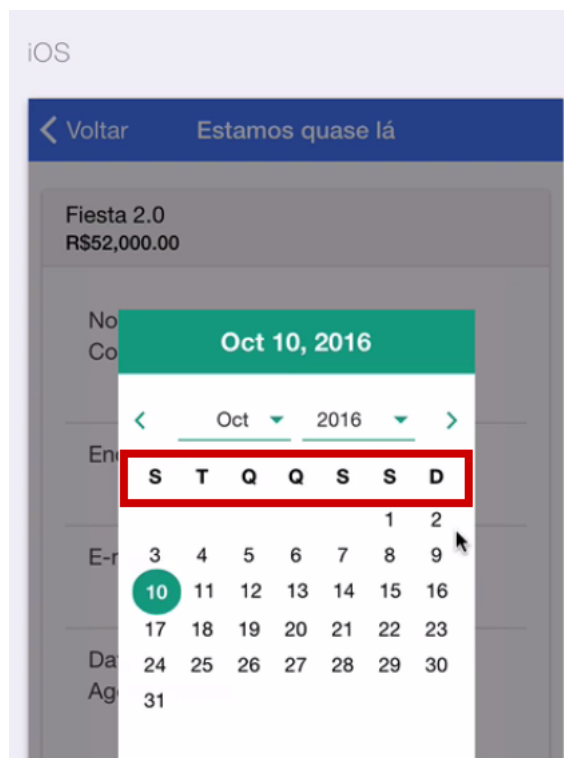
    var configuracoes = {
        callback : function(data){
            $scope.dataSelecionada = new Date(data);
        },

        weekList = ['D', 'S', 'T', 'Q', 'Q', 'S', 'S']
    }
}
```

Mas se testarmos, quando chegarmos na tela "Carro escolhido", ao clicarmos em "Finalizar pedido" teremos um erro, porque o `weekList` é um atributo e deve ser seguido pelo `:`.

```
weekList : ['D', 'S', 'T', 'Q', 'Q', 'S', 'S']
```

Fique atento a isto no seu código. Agora, quando chegarmos até a opção para o agendamento da data, veremos as alterações nos dias da semana no calendário.



Você pode fazer outras customizações no seu calendário.

Conseguimos formatar a data e os dias da semana estão em português.

Até a próxima!