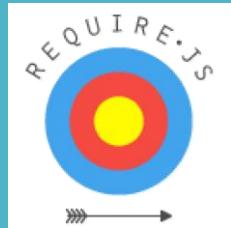




DURANDAL



***Knockout.***

B  
Bootstrap

# Introdução às tecnologias Web - ITW

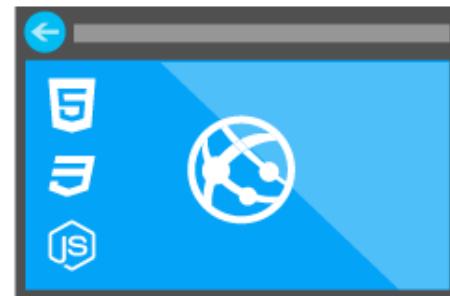
Aula 11 – ITW App

# Sumário

Primeiras noções do desenvolvimento de uma aplicação móvel utilizando HTML5 + CSS + Javascript



Single Page App's



A plataforma Durandal  
RequireJS, Knockout, Bootstrap



A aplicação ITW App



# Primeiras noções do desenvolvimento de uma aplicação móvel

Hoje em dia a produção de aplicações/jogos destinados a multiplataforma (telemóvel, tablet, computador, ...) /multisistema operativo (Android, IOS, Windows, ...) em dispositivos móveis está facilitada pela utilização das tecnologias do trinómio HTML5/CSS/Javascript.

Através da sua utilização é possível desenvolver aplicações não-nativas para dispositivos móveis mas que se comportam de modo muito similar às aplicações nativas com a vantagem de ser necessário desenvolver apenas uma versão de código.

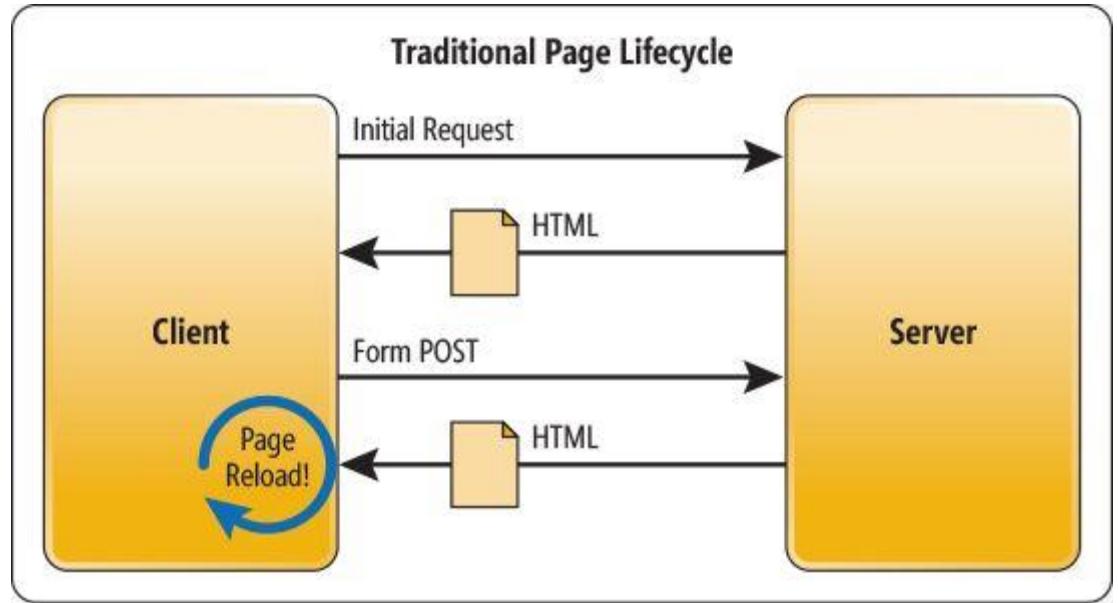
Ferramentas, como por exemplo, o phonegap fazem depois com que com a única versão do código possam ser geradas aplicações distintas destinadas aos diversos dispositivos/sistemas operativos.



Embora o desenvolvimento seja baseado em HTML, o desenvolvimento de aplicações deve ter um comportamento similar ao das aplicações nativas.

Isso implica numa forma de desenvolvimento distinto do utilizado, por exemplo, nos sítios e aplicações web.

Neste contexto é frequentemente utilizado o desenvolvimento numa arquitetura de **“aplicação de página única”** ou, no original, em inglês, SPA's – *Single Page Application's - Architecture*

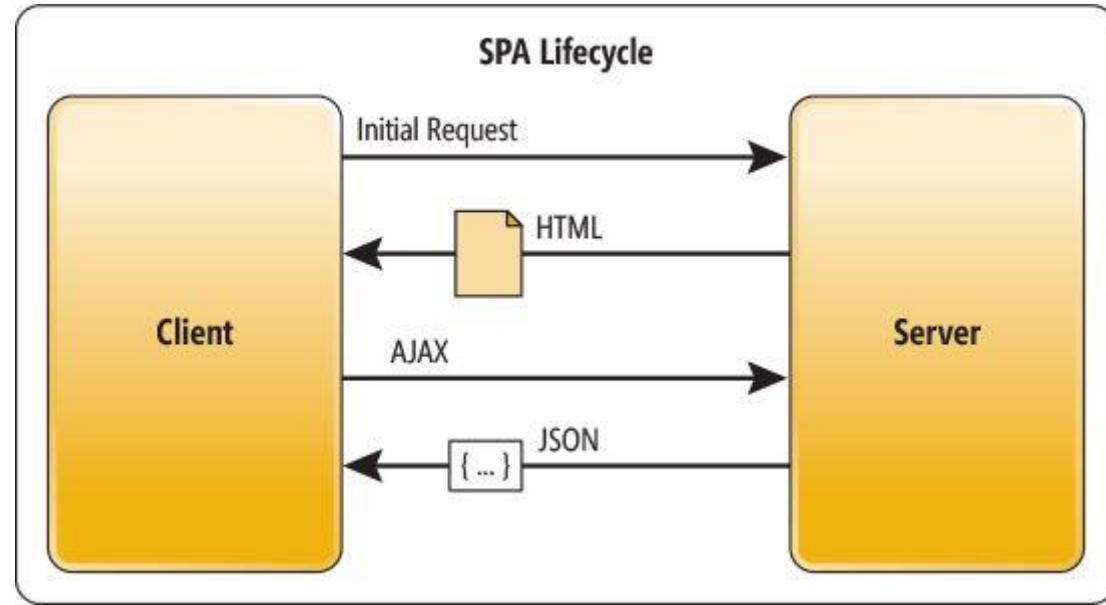


Nas **aplicações web tradicionais**, o cliente (browser) inicia a comunicação com o servidor, solicitando uma página.

O servidor processa então o pedido e envia o código HTML da página para o cliente.

O utilizador interage com a página com a página, por exemplo, selecionando hiperligações ou preenchendo um formulário com dados. Pro cada ação, um novo pedido é enviado para o servidor e o fluxo começa novamente....

O servidor processa o pedido e envia uma nova página para o navegador em resposta à nova ação solicitada pelo cliente.



**Em aplicações de página simples** (do inglês Single Page Application) tudo é carregado no browser após o pedido inicial

Interações subsequentes ocorrerão usando pedidos Ajax.

Isto significa que o navegador deve atualizar somente a parte da página que mudou; não há necessidade de recarregar a página inteira.

A abordagem SPA reduz o tempo dispendido pela aplicação para responder às acções do utilizador, o que resulta em uma experiência mais fluida.

E não tem desvantagens?

### Principais características de uma SPA

- Manipulação direta do DOM
- Navegação do hashtag
- Comunicações assíncronas feitas com o servidor
- Data databinding direto

# Objetivo?



# Ferramentas para o desenvolvimento de aplicações de página simples

# Frameworks para o desenvolvimento de SPA's

Google's AngularJS

Ember.js

Facebook's React

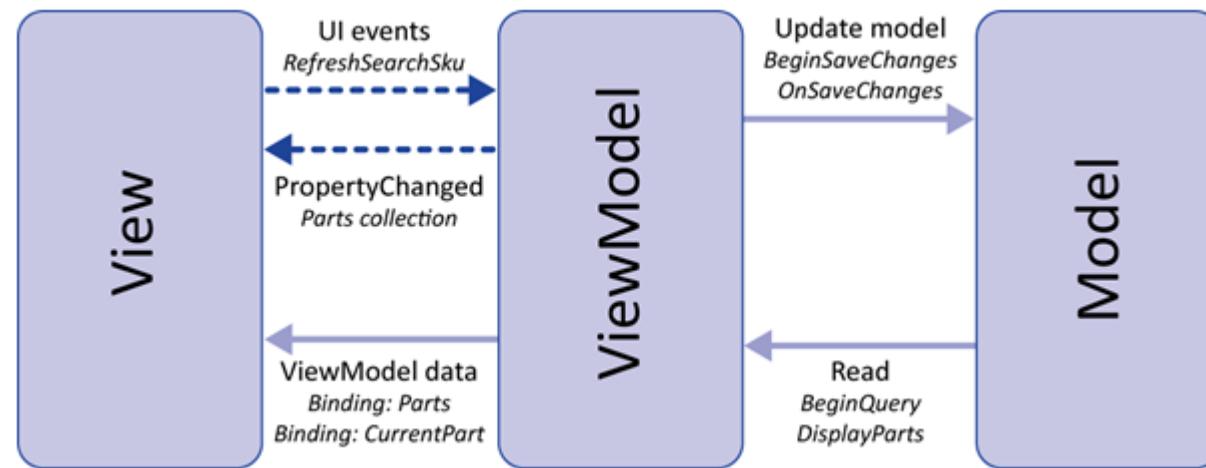
Knockout SPA

jquery.com

**Durandal**

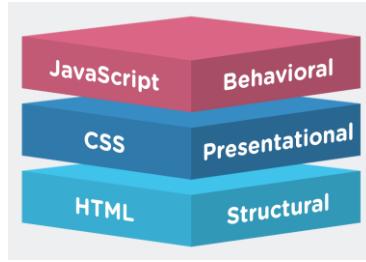
Aurelia

...



As plataformas de desenvolvimento de SPA's utilizam um modelo MVVM

# Cenários de desenvolvimento







# Durandal Features

Clean MV\* Architecture

JS & HTML Modularity

Simple App Lifecycle

Eventing, Modals, Message Boxes, etc.

Navigation & Screen State Management

Consistent Async Programming w/ Promises

App Bundling and Optimization

Use any Backend Technology

Built on top of jQuery, Knockout & RequireJS

Integrates with popular CSS libraries such as Bootstrap and Foundation

Make Your Own Templatable and Bindable Widgets

Fully Testable

# The 3 Pillars

jQuery

RequireJS

Knockout



# RequireJS – gestão & controlo das dependências



```
define(function (require) {
    var dependency1 = require('dependency1'),
        dependency2 = require('dependency2');

    return function () {};
});
```

The screenshot shows a code editor with two tabs: 'main.js' and 'flickr.js'. The 'main.js' tab contains the following RequireJS configuration and module definitions:

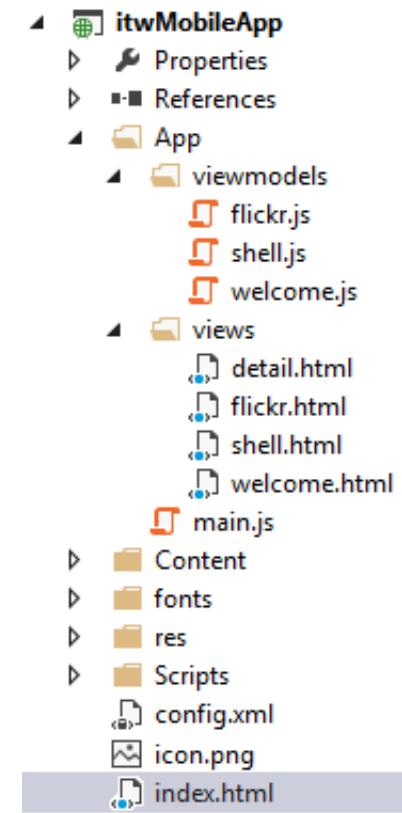
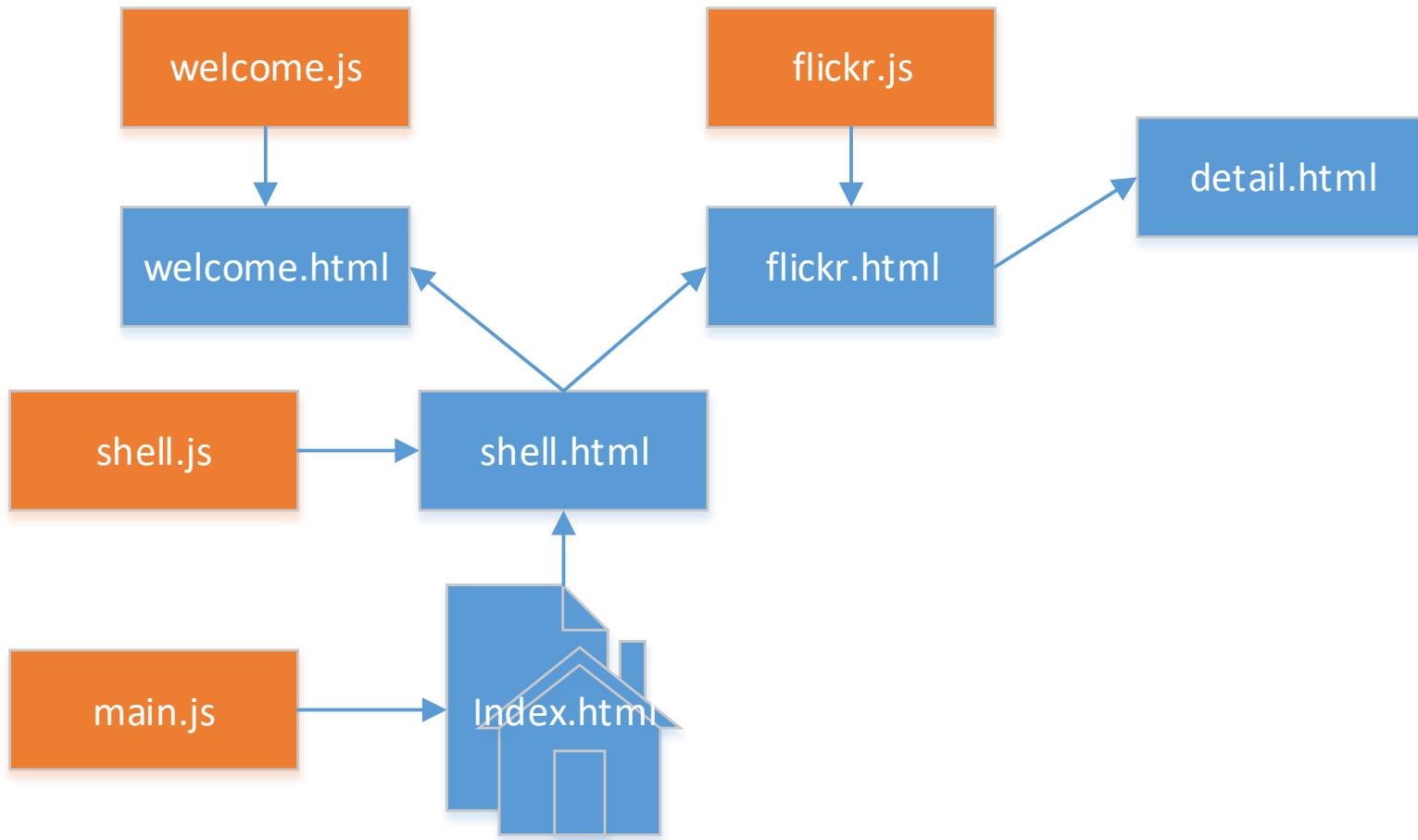
```
1 requirejs.config({
2     paths: {
3         'text':      '../Scripts/text',
4         'durandal':  '../Scripts/durandal',
5         'plugins':   '../Scripts/durandal/plugins',
6         'transitions': '../Scripts/durandal/transitions',
7         'amplify':   '../Scripts/amplify.min',
8         'moment':    '../Scripts/moment-with-locales.min',
9     },
10    shim: {
11        "amplify": { deps: ["jquery"], exports: "amplify" },
12        "bootstrap": { "deps": ['jquery'] },
13    }
14 });
15
16 define('jquery', function () { return jQuery; });
17 define('knockout', ko);
18
19 define(['durandal/system', 'durandal/app', 'durandal/viewLocator'], function (system, app, viewLocator){...});
```

The 'flickr.js' tab contains the following module definition:

```
1 define(['plugins/http', 'durandal/app', 'knockout'], function (http, app, ko) {
2     //Note: This module exports an object.
3     //That means that every module that "requires" it will get the same object instance.
4     //If you wish to be able to create multiple instances, instead export a function.
5     //See the "welcome" module for an example of function export.
6
7     return {
8         displayName: 'Flickr',
9         images: ko.observableArray([]),
10        activate: function (){},
11        select: function(item){},
12        canDeactivate: function (){}
13    };
14});
```

# A ITW App

# Arquitetura e organização da aplicação



# index.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>ITW App</title>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1" />
    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-status-bar-style" content="black" />
    <meta name="format-detection" content="telephone=no" />
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, target-densitydpi=medium-dpi" />
    <link rel="shortcut icon" href="icon.png" />
    <link href="Content/bootstrap.min.css" rel="stylesheet" />
    <link href="Content/font-awesome.min.css" rel="stylesheet" />
    <link href="Content/durandal.css" rel="stylesheet" />
    <link href="Content/myApp.css" rel="stylesheet" />
</head>
<body>
    <div id="applicationHost"></div> ←
    <script src="cordova.js"></script>
    <script src="Scripts/jquery-2.1.4.min.js"></script>
    <script src="Scripts/bootstrap.min.js"></script>
    <script src="Scripts/knockout-3.3.0.js"></script>
    <script src="Scripts/require.js" data-main="App/main"></script>
</body>
</html>
```

# main.js

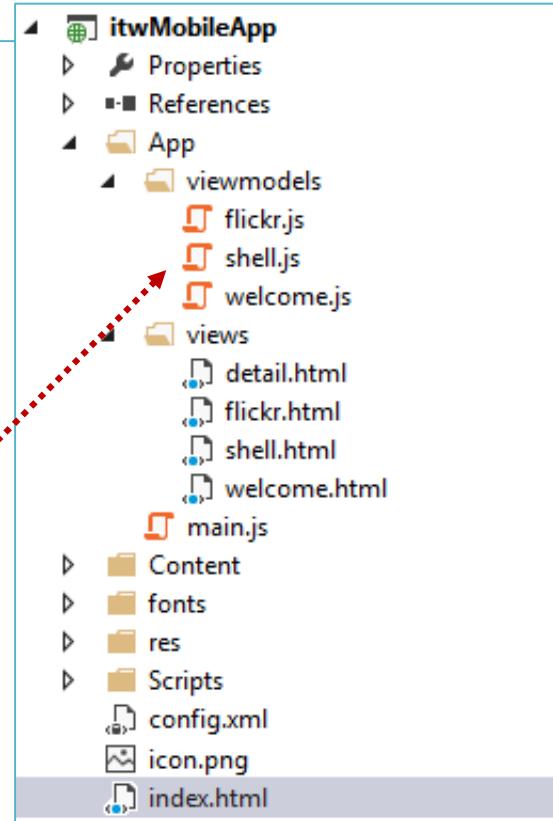
```
requirejs.config({
  paths: {
    'text': '../Scripts/text',
    'durandal': '../Scripts/durandal',
    'plugins': '../Scripts/durandal/plugins',
    'transitions': '../Scripts/durandal/transitions'
  }
});

define('jquery', function () { return jQuery; });
define('knockout', ko);

define(['durandal/system', 'durandal/app', 'durandal/viewLocator'],
function (system, app, viewLocator) {
  //>>excludeStart("build", true);
  system.debug(true);
  //>>excludeEnd("build");
  app.title = 'ITW Starter Kit';
  app.configurePlugins({
    router: true,
    dialog: true
  });

  app.start().then(function() {
    //Replace 'viewmodels' in the moduleId with 'views' to locate the view.
    //Look for partial views in a 'views' folder in the root.
    viewLocator.useConvention();

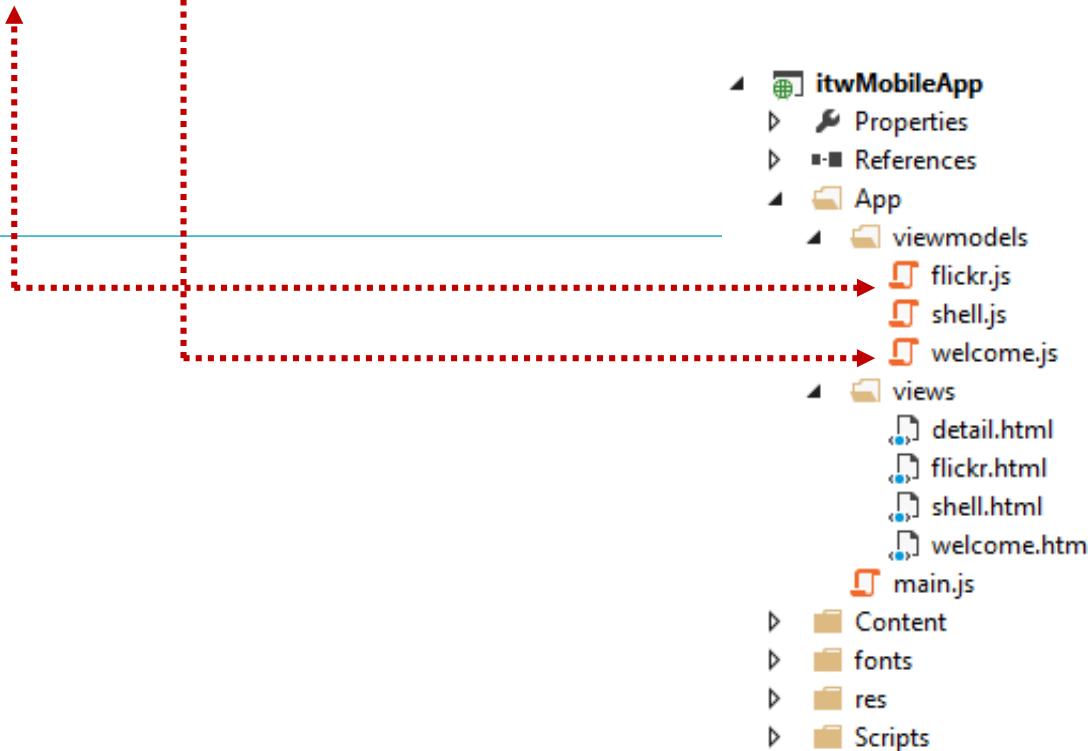
    //Show the app by setting the root view model for our application with a transition.
    app.setRoot('viewmodels/shell', 'entrance');
  });
});
```



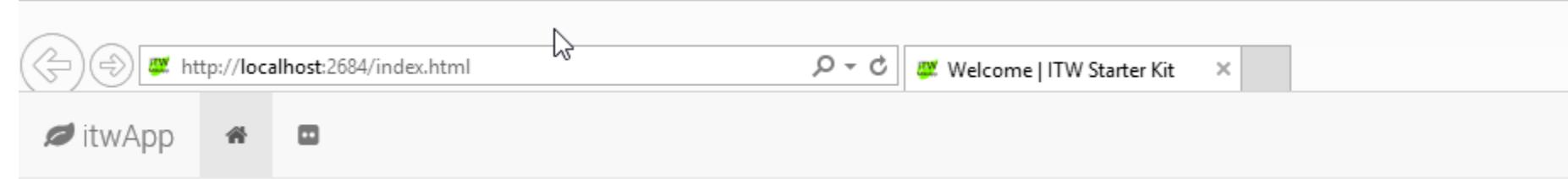
# shell.js

```
define(['plugins/router', 'durandal/app'], function (router, app) {
    return {
        router: router,
        activate: function () {
            router.map([
                { route: '/', title: 'Welcome', moduleId: 'viewmodels/welcome', nav: true, menu: '<i class="fa fa-home"></i>' },
                { route: 'flickr', moduleId: 'viewmodels/flickr', nav: true, menu: '<i class="fa fa-flickr"></i>' }
            ]).buildNavigationModel();

            return router.activate();
        }
    };
});
```



# shell.html



```
<div>
    <nav class="navbar navbar-default navbar-fixed-top" role="navigation">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
                <span class="sr-only">Toggle Navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">
                <i class="fa fa-leaf"></i>
                <span>itwApp</span>
            </a>
        </div>

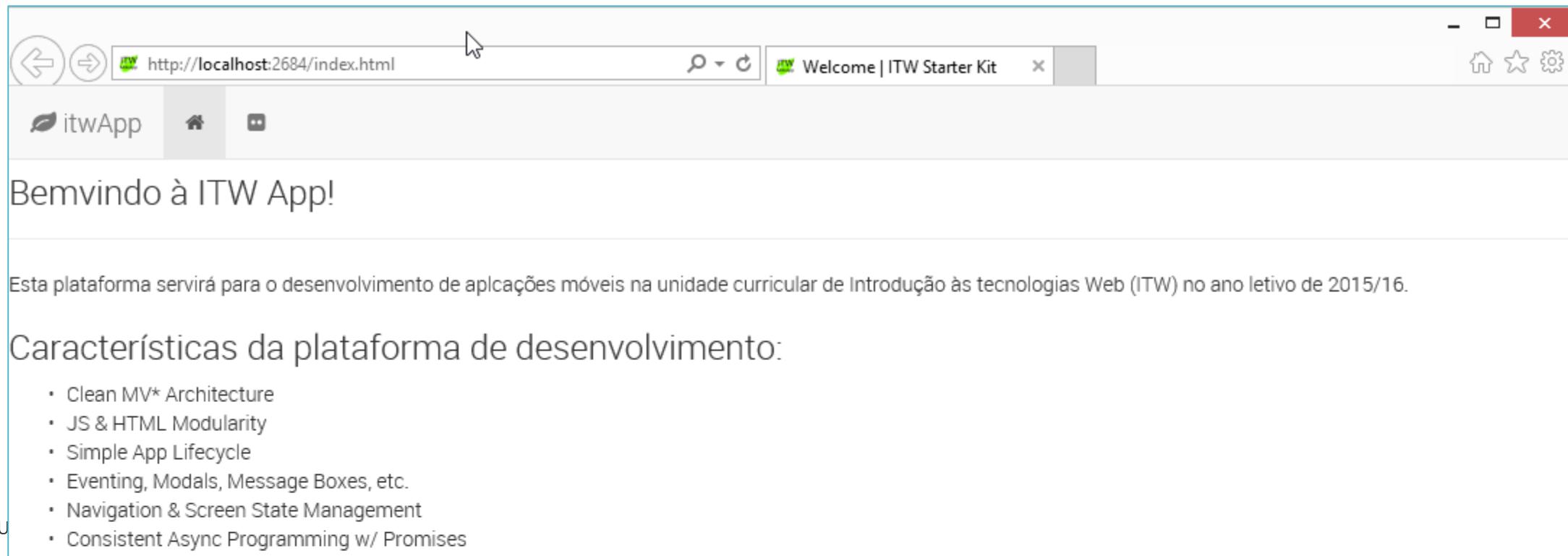
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav" data-bind="foreach: router.navigationModel">
                <li data-bind="css: { active: isActive }">
                    <a data-bind="attr: { href: hash, title: title }, html: menu"></a>
                </li>
            </ul>
        </div>
    </nav>
    <div class="page-host" data-bind="router: { transition:'entrance' }"></div>
</div>
```

# welcome.js

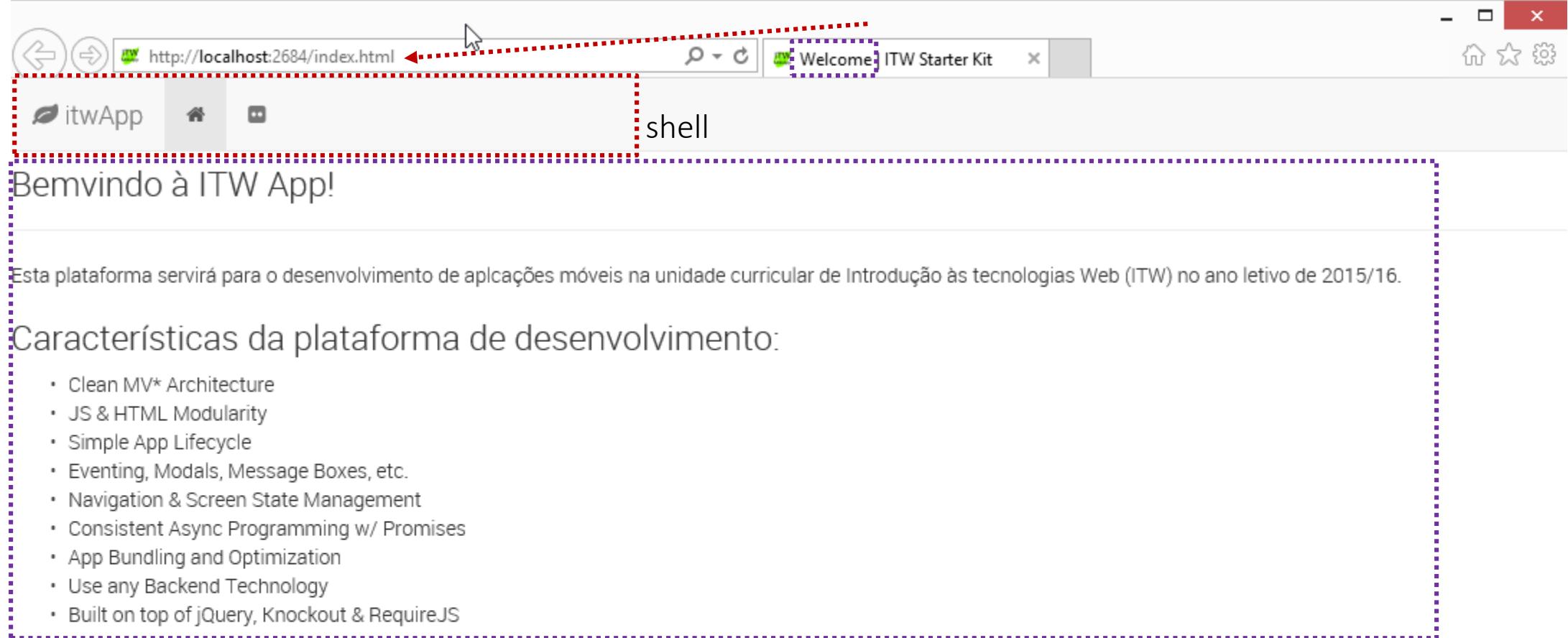
```
define(function() {
    return {
        displayName : 'Bemvindo à ITW App!',
        description : 'Esta plataforma servirá para o desenvolvimento de aplicações móveis na unidade curricular de Introdução às tecnologias Web (ITW) no ano letivo de 2015/16.',
        features : [
            'Clean MV* Architecture',
            'JS & HTML Modularity',
            'Simple App Lifecycle',
            'Eventing, Modals, Message Boxes, etc.',
            'Navigation & Screen State Management',
            'Consistent Async Programming w/ Promises',
            'App Bundling and Optimization',
            'Use any Backend Technology',
            'Built on top of jQuery, Knockout & RequireJS'
        ]
    };
});
```

# welcome.html

```
<section>
  <div class=" page-header"><h3 data-bind="text:displayName"></h3></div>
  <p data-bind="text:description"></p>
  <h3>Características da plataforma de desenvolvimento:</h3>
  <ul data-bind="foreach: features">
    <li data-bind="text:$data"></li>
  </ul>
</section>
```



The screenshot shows a web browser window displaying the content of the welcome.html file. The browser's address bar shows the URL `http://localhost:2684/index.html`. The title bar of the browser window says "Welcome | ITW Starter Kit". The main content area of the browser displays the rendered HTML. At the top, there is a header section with a logo (leaf icon), a home button, and a menu button. Below this, a large heading says "Bemvindo à ITW App!". A descriptive paragraph follows, stating: "Esta plataforma servirá para o desenvolvimento de aplicações móveis na unidade curricular de Introdução às tecnologias Web (ITW) no ano letivo de 2015/16." Underneath this, a section titled "Características da plataforma de desenvolvimento:" lists several bullet points: "Clean MV\* Architecture", "JS & HTML Modularity", "Simple App Lifecycle", "Eventing, Modals, Message Boxes, etc.", "Navigation & Screen State Management", and "Consistent Async Programming w/ Promises".



# flickrjs

```
define(['plugins/http', 'durandal/app', 'knockout'], function (http, app, ko) {
    //--- variáveis e funções globais -----
    var outsideData;
    function getDataFor(myLocation) {
        return http.jsonp('http://api.flickr.com/services/feeds/photos_public.gne',
            {
                tags: myLocation, tagmode: 'any', format: 'json', lang: 'pt-br'
            },
            'jsoncallback').then(function (response) {
                outsideData.images(response.items);
            });
    }
    //--- função específica do viewmodel -----
});
```

(continua...)

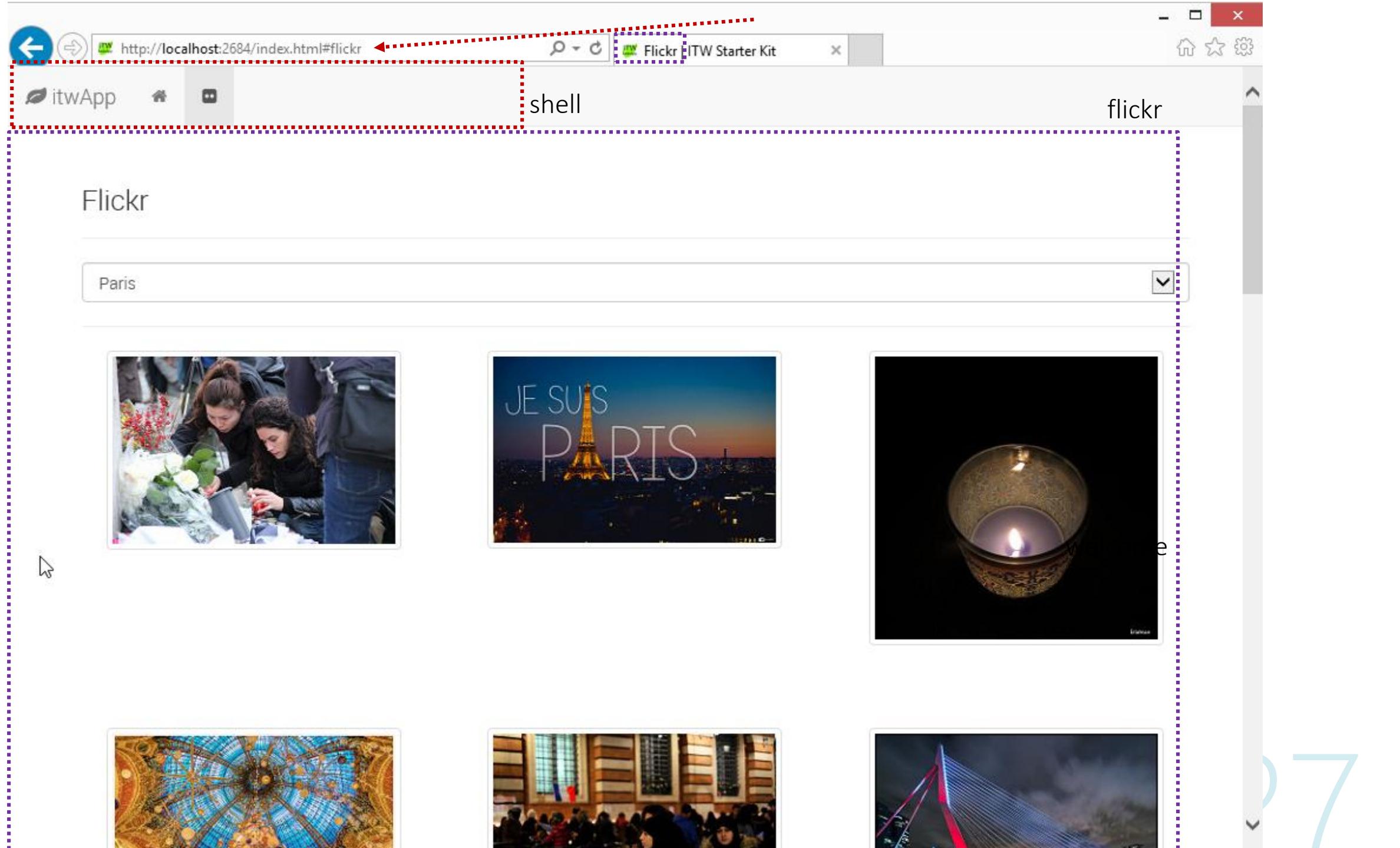
```

//--- função específica do.viewmodel -----
return {
    displayName: 'Flickr',
    images: ko.observableArray([]),
    activate: function () {
        //the router's activator calls this function and waits for it to complete before proceeding
    },
    compositionComplete: function () {
        outsideData = this;
        //--- regista uma função para responder, de futuro, quando
        //--- o utilizador mudar o nome da cidade selecionada
        $('select').change(function () {
            getDataFor($('select').val());
        });
        //--- fim
        if (this.images().length > 0) {
            return;
        }
        //--- vai buscar as images referentes à cidade selecionada incialmente
        getDataFor($('select').val());
    },
    select: function (item) {
        //the app model allows easy display of modal dialogs by passing a view model
        //views are usually located by convention, but you can specify it as well with viewUrl
        item.viewUrl = 'views/detail';
        app.showDialog(item);
    },
    splitTags: function (tags) {
        var arr = tags.split(' ');
        var retVal = "Tags: ";
        for (x = 0; x < arr.length; x++) {
            if (x > 0)
                retVal += " #" + arr[x] + ";";
            else
                retVal += "#" + arr[x] + ";";
        }
        return retVal;
    }
};
});

```

# flickr.html

```
<section>
  <div class="container">
    <div class=" page-header"><h3 data-bind="text: displayName"></h3></div>
    <select id="myDestination" class="form-control">
      <option>London</option>
      <option>Lisbon</option>
      <option>New York</option>
      <option>Paris</option>
    </select>
    <hr />
    <div class="row" data-bind="foreach: images">
      <div class="col-sm-6 col-md-4 flickrImage">
        <a href="#" data-bind="attr: { title: 'Title: ' + title + '\n' + $parent.splitTags(tags) }, click:$parent.select" class="small">
          <img class="thumbnail text-center" data-bind="attr: { src: media.m }" />
        </a>
      </div>
    </div>
  </div>
</section>
```



# detail.html

```
<div class="modal-content pictureDetail autoclose">
  <div class="modal-header">
    <h3>Details</h3>
  </div>
  <div class="modal-body">
    <p data-bind="html: description"></p>
  </div>
</div>
```



http://localhost:2684/index.html#flickr



Flickr | ITW Starter Kit



itwApp



# Flickr

Paris



## Details

Jose RL postou uma foto:



04/1

©20

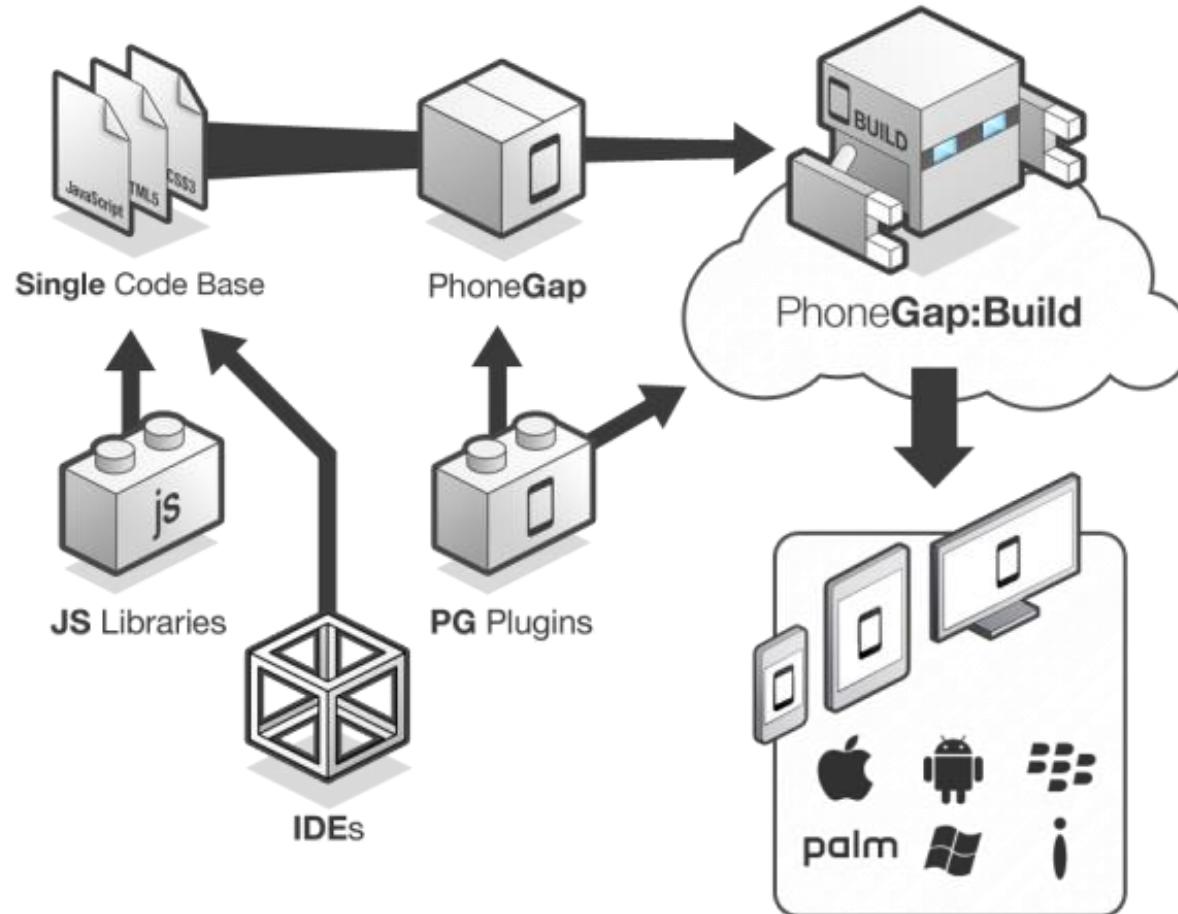
<http://www.flickr.com/photos/josrl/22423197083/>



9

# Como criar uma aplicação móvel

# Phonegap



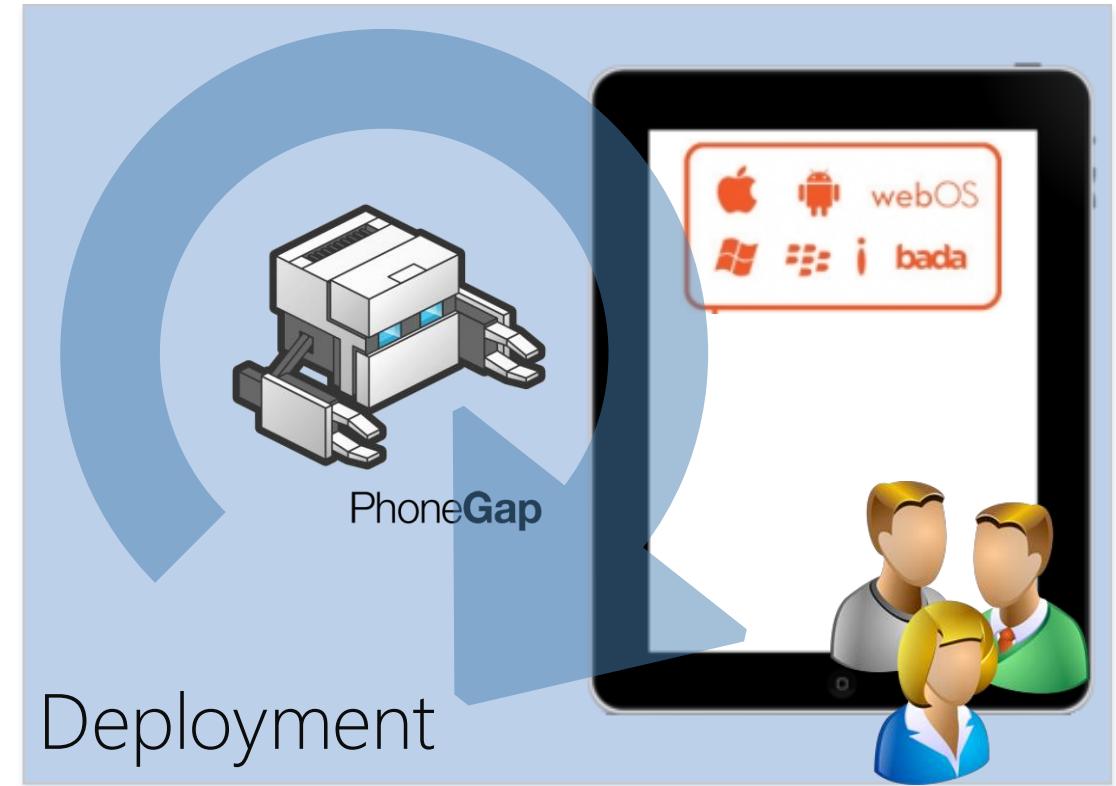
# Processo de desenvolvimento



Interfaces para os vários tipos de perguntas

Templating

```
# $IPRI: cookie.rb,v 1.16 2002/09/21 12:23:35 gotoyuzu Exp: $  
require 'time'  
require 'webrick/httputils'  
  
module WEBrick  
  class Cookie  
  
    attr_reader :name  
    attr_accessor :value, :version  
    attr_accessor :domain, :path, :secure  
    attr_accessor :comment, :max_age  
    #attr_accessor :comment_url, :discard, :port  
  
    def initialize(name, value)  
      @name = name  
      @value = value  
      @version = 0 ..... # Netscape Cookie  
      @domain = @path = @secure = @comment = @max_age =  
      @expires = @comment_url = @discard = @port = nil  
    end  
  end  
end
```



# GitHub

Repositório público com o projeto utilizado para fazer a criação da aplicação:

[https://github.com/joaquimsousapinto/ITW\\_App](https://github.com/joaquimsousapinto/ITW_App)

É possível obter uma cópia do projeto quer através de um ficheiro ZIP quer fazendo o clone na aplicação de desktop do GitHub

[\(https://desktop.github.com/\)](https://desktop.github.com/)

The screenshot shows a GitHub repository page for 'joaquimsousapinto / ITW\_App'. The repository has 2 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was made 5 hours ago. The repository contains files such as config.xml, icon.png, index.html, App, Content, Scripts, fonts, and res/icon. A message encourages adding a README. At the bottom right, there are buttons for 'Clone in Desktop' and 'Download ZIP', which are highlighted with a red dashed box.

joaquimsousapinto / ITW\_App

Application to support ITW classes — Edit

2 commits 1 branch 0 releases 1 contributor

Branch: master ITW\_App / +

joaquimsousapinto [001] - config.xml version update Latest commit 0299267 5 hours ago

App [000] Initial Setup 5 hours ago

Content [000] Initial Setup 5 hours ago

Scripts [000] Initial Setup 5 hours ago

fonts [000] Initial Setup 5 hours ago

res/icon [000] Initial Setup 5 hours ago

config.xml [001] - config.xml version update 5 hours ago

icon.png [000] Initial Setup 5 hours ago

index.html [000] Initial Setup 5 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

HTTPS clone URL https://github.com/

You can clone with HTTPS, SSH, or Subversion. Ⓜ

Clone in Desktop Download ZIP

# Adobe PhoneGap Build

## vista do criador da aplicação

The screenshot shows the Adobe PhoneGap Build application interface. At the top, there's a browser-like header with the URL <https://build.phonegap.com/apps>. Below it is a navigation bar with links for Apps, Plugins, Docs, Blog, FAQ, and Support. On the right side of the header, there's a user profile for "Joaquim" and the Adobe logo.

In the main area, there's a "Your apps" section with a button to "+ new app". To the left, there's a "Feedback" sidebar. The main content area displays the "ITW App" project. It includes a thumbnail for "ITW MobApp", a brief description in Portuguese about developing mobile applications for the ITW course, and build status for iOS, Android, and Windows. The iOS icon is highlighted with a red dashed rectangle. Below the icons are buttons for "Update code", "Rebuild all", and "Debug". A "Hydration Enabled" indicator is also present.

On the right side of the project card, detailed information is provided:

App ID	Version	Owned by
1767951	1.0.0.201511151...	jsp@ua.pt
PhoneGap (iOS / Android / Windows)		
cli-5.2.0 (3.9.1 / 41.1 / 3.8.1)		
Source		
<a href="https://github.com/joaquimsousapinto/ITW_App.git">https://github.com/joaquimsousapinto/ITW_App.git</a>		
Tag / Branch	Commit	Last built (10)
master	0299267	4 hours

A QR code is available for sharing, and a "public page" button is at the bottom right.

# PhoneGap App – página pública

(<https://build.phonegap.com/apps/1767951/share>)

A leitura de um QR Code permite ao dispositivo aceder diretamente à página de onde a aplicação pode ser instalada.



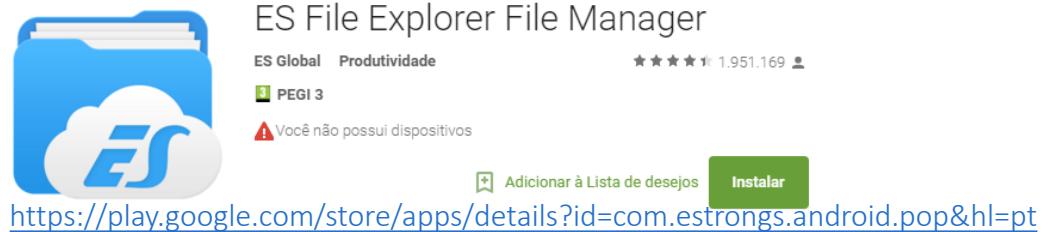
<https://play.google.com/store/apps/details?id=com.qrscan.barcodereader&hl=pt>

A screenshot of a web browser window titled "Adobe PhoneGap Build". The URL in the address bar is https://build.phonegap.com/apps/1767951/share. The page displays information for the "ITW App 1.0.0.201511151632". It includes a logo for "ITW MobApp", the version number, and a description: "Esta plataforma servirá para o desenvolvimento de aplicações móveis na unidade curricular de Introdução às tecnologias Web (ITW) no ano letivo de 2015/16.". Below this, there are download links for "ANDROID" (apk file) and "Windows Phone" (xap file), each accompanied by its respective platform icon. At the bottom right of the page is another QR code.

# Instalação da aplicação

A instalação da aplicação pode ser feita de dois modos distintos:

- executando diretamente a página apontada pelo QRCode;
- descarregando a aplicação para o seu computador, copiando a aplicação para o cartão de memória, por USB, e finalmente, através de uma aplicação de gestão de ficheiros, executando o ficheiro ".apk".



**ATENÇÃO!!!** Para que possa instalar e testar estas aplicações, em qualquer dos modo previamente apresentados, o telemóvel tem de estar em "**Debug Mode**". Esta configuração varia de telemóvel para telemóvel...

# Cópia do projeto Visual Studio na Dropbox

(<https://www.dropbox.com/s/p40zilr6jgearsn/itwMobileApp.zip?dl=0>)

