

# Introdução às tecnologias Web - ITW

## Aula 10 – Google API's (Maps & Charts)

# Sumário

Google API's  
Google Maps  
Google Charts



# O que é uma API? Para que serve?

Uma API (acrônimo do termo em inglês *Application programming interface*) consiste na especificação de um conjunto de rotinas e estruturas de dados estabelecidos por um desenvolvedor de software.

Através da utilização dessas rotinas e dessas estruturas de dados cria-se uma interface que permite a utilização dessas funcionalidades por aplicativos que não pretendem envolver-se nos detalhes da implementação do software, mas apenas usar seus serviços.

A Google disponibiliza várias APIs para permitir o acesso programático aos seus mais diversos serviços. Um deles é o “Google Maps” que permite a pesquisa e visualização de mapas.

São disponibilizadas duas APIs principais para aceder a este serviço:

## **Google Maps Embed API**

<https://developers.google.com/maps/documentation/embed/guide>

## **Google Maps JavaScript API v3**

<https://developers.google.com/maps/documentation/javascript/tutorial>

# Google Maps

## Embed API

A Embed API do serviço Google Maps utiliza um pedido simples, em HTTP, para devolver um mapa.

Este mapa é dinâmico e interativo e pode facilmente ser inserido numa página web, atribuindo a URL da API ao atributo “src” de um elemento “iframe”.

A sintaxe para a URL da API é:

<https://www.google.com/maps/embed/v1/MODO?key=CHAVE&PARÂMETROS>

MODO = “place” | “directions” | “search” | “view”

CHAVE = chave gratuita de acesso à API

PARÂMETROS = parâmetros específicos de cada modo ou parâmetros adicionais e opcionais

# Google Maps

## Embed API – Modo “Place”

MODO = “place” – mapa de um lugar

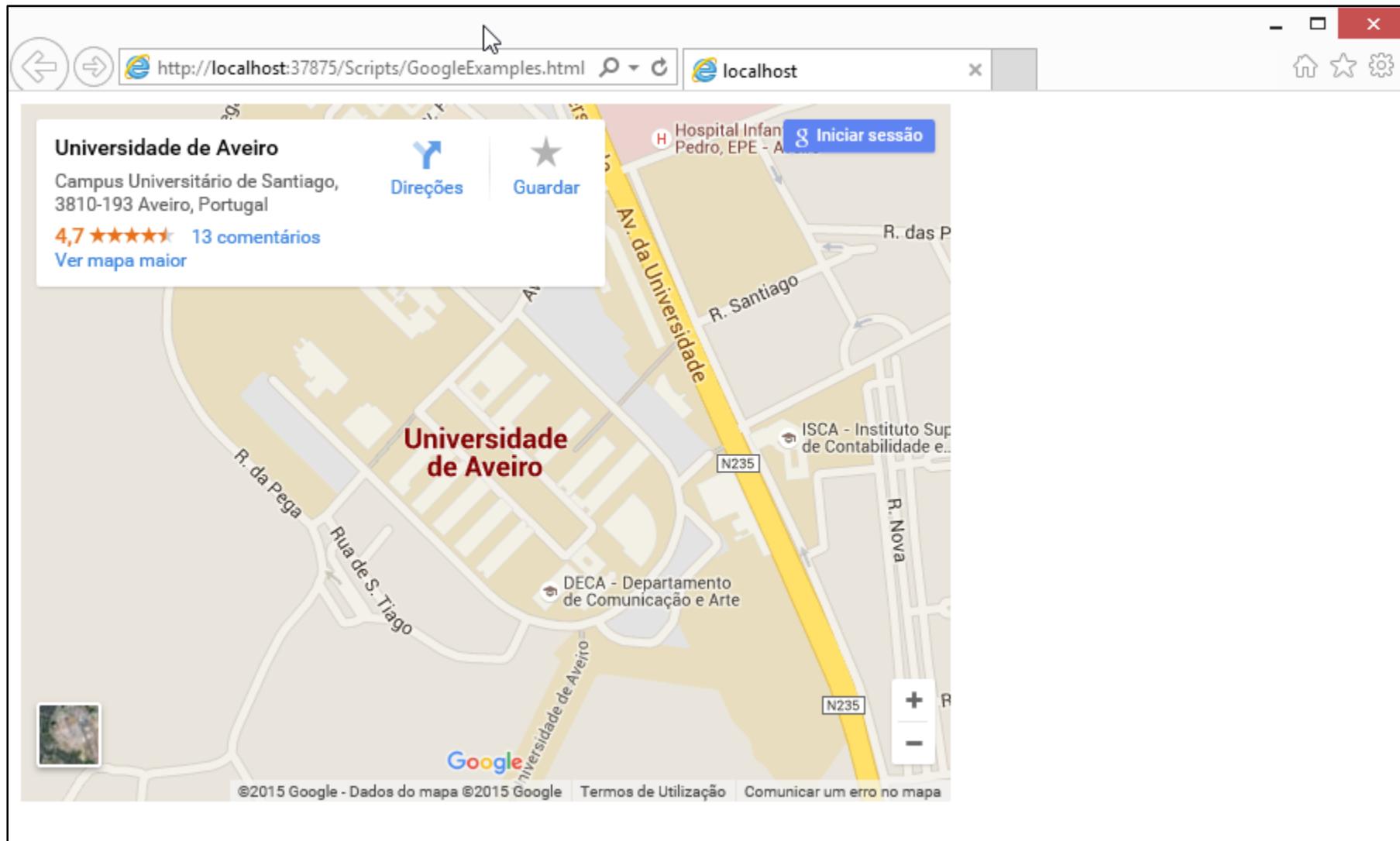
PARÂMETROS =

Obrigatórios

*q – especifica o lugar a pesquisar*

Exemplo:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <iframe width="600" height="450" frameborder="0" style="border:0"
        src="https://www.google.com/maps/embed/v1/place?key=AIzaSyDM0ZN1zzS6wu1KF0v1s6PARvuXmGI-paM&q=Universidade+Aveiro+Informatica"
        allowfullscreen></iframe>
</body>
</html>
```



# Google Maps

## Embed API – Modo “Directions”

MODO = “directions” – mapa com trajeto entre pontos

PARÂMETROS =

Obrigatórios

“origin” – origem do trajeto

“destination” – destino do trajeto

Opcionais

“waypoints” – pontos de passagem no trajeto

“mode” – modo de viajar (*driving, walking, bicycling, transit, flying*)

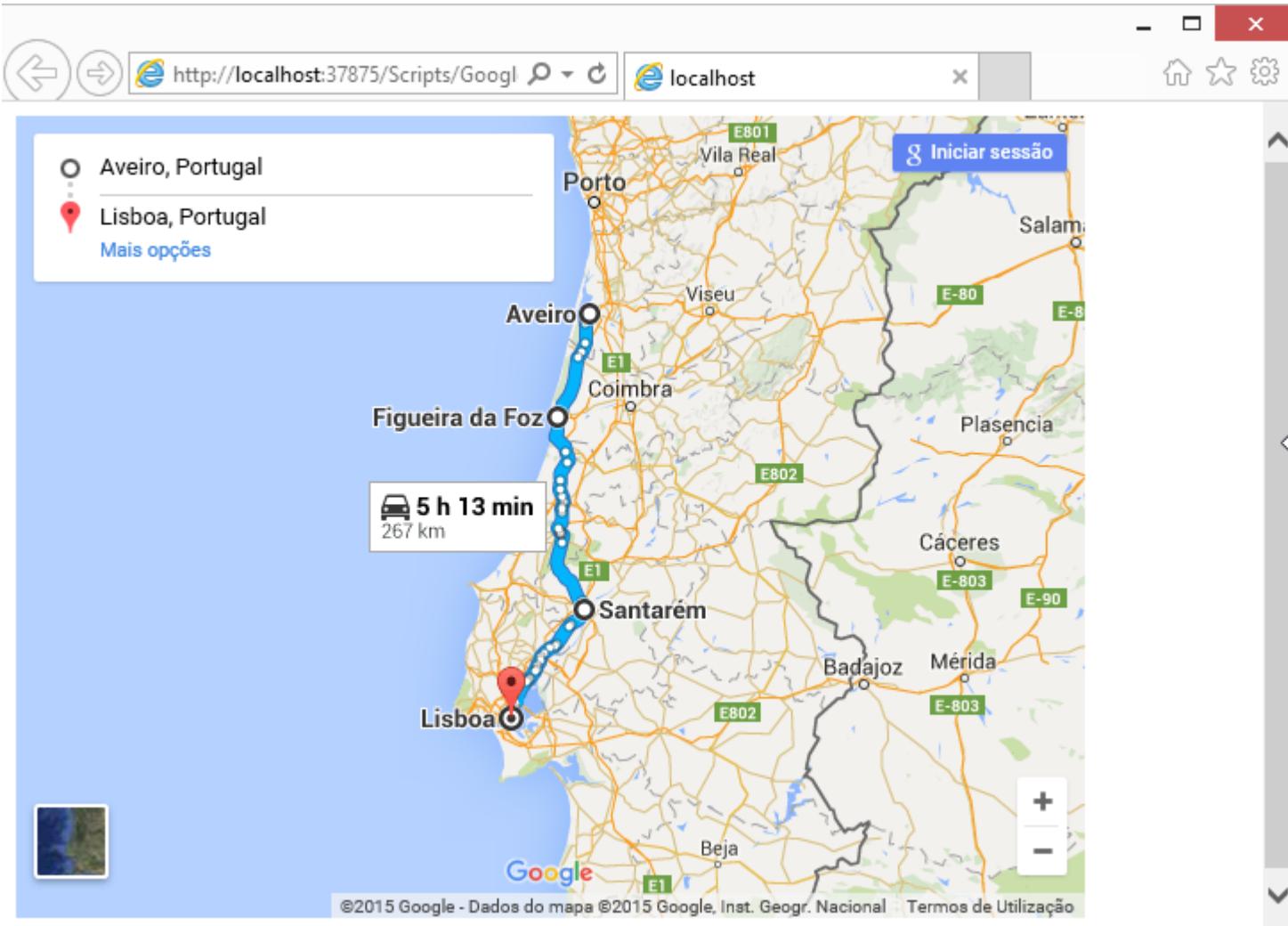
“avoid” – questões a evitar (*tolls, highways*)

“units” – unidades de medida do trajeto (*metric, imperial*)

Exemplo:

Aveiro → Lisboa, passando pela Figueira da Foz e por Santarém

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <iframe width="600" height="450" frameborder="0" style="border:0"
        src="https://www.google.com/maps/embed/v1/directions?key=AIzaSyDM0ZNlzzS6wulKF0v1s6PARvuXmGI-paM
            &origin=Aveiro
            &destination=Lisbon
            &waypoints=Figueira+Foz|Santarem
            &avoid=tolls"
        allowfullscreen></iframe>
</body>
</html>
```



# Google Maps

## Embed API – Modo “Search”

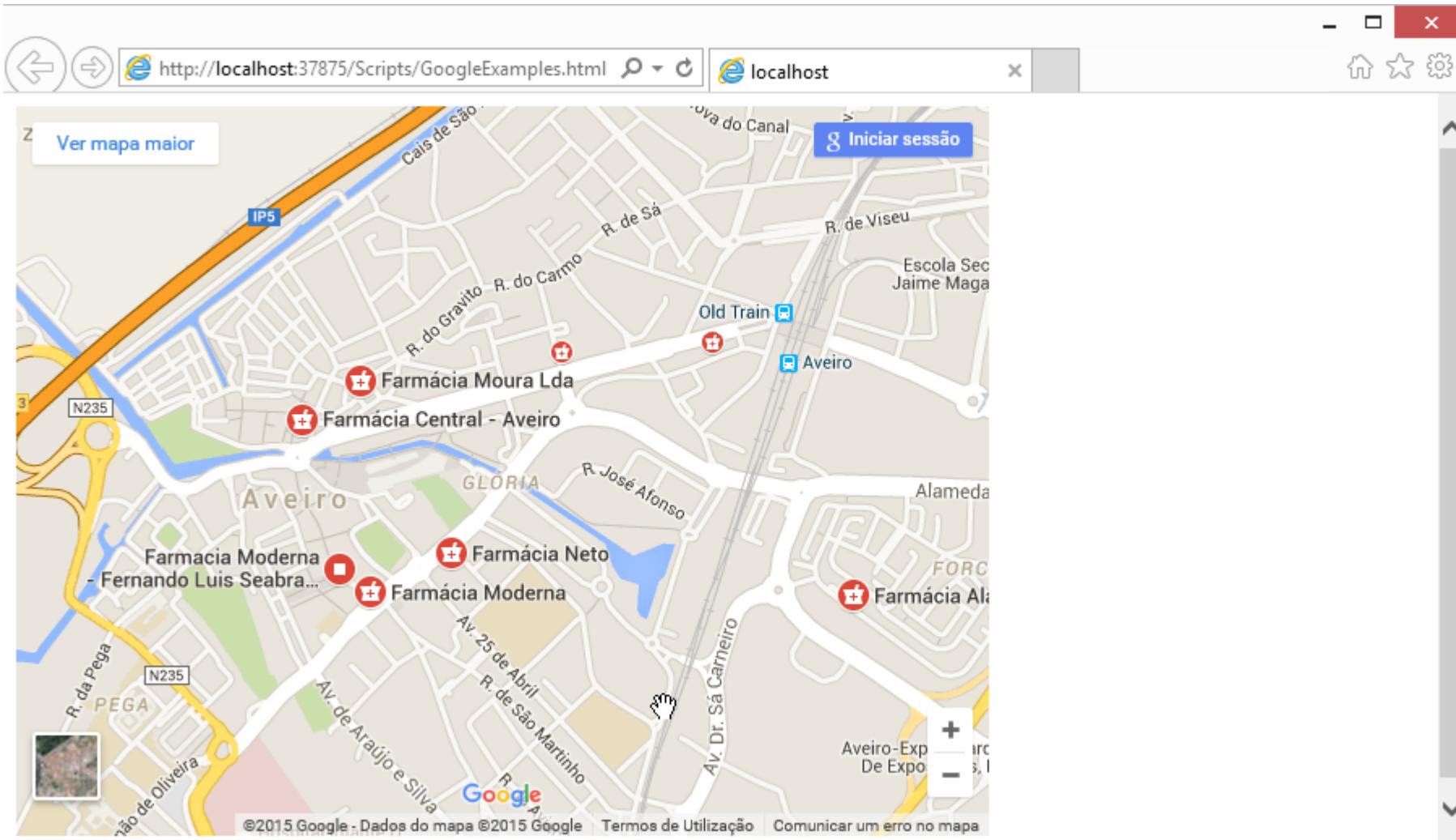
MODO = “search” – mapa com pesquisa de características num lugar

PARAMETROS =

*q – especifica o lugar e as características a pesquisar*

Exemplo:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <iframe width="600" height="450" frameborder="0" style="border:0"
        src="https://www.google.com/maps/embed/v1/search?key=AIzaSyDM0ZN1zZS6wulKF0v1s6PARvuXmGI-paM
        &q=Portugal+Aveiro+farmácias"
        allowfullscreen></iframe>
</body>
</html>
```



# Google Maps

## Embed API – Modo “Search”

MODO = “view” – mapa com centro num par de coordenadas

### PARÂMETROS

#### Obrigatório

“center” – *centro do mapa*

#### Opcionais

“zoom” – *nível de ampliação do mapa (0 - 21)*

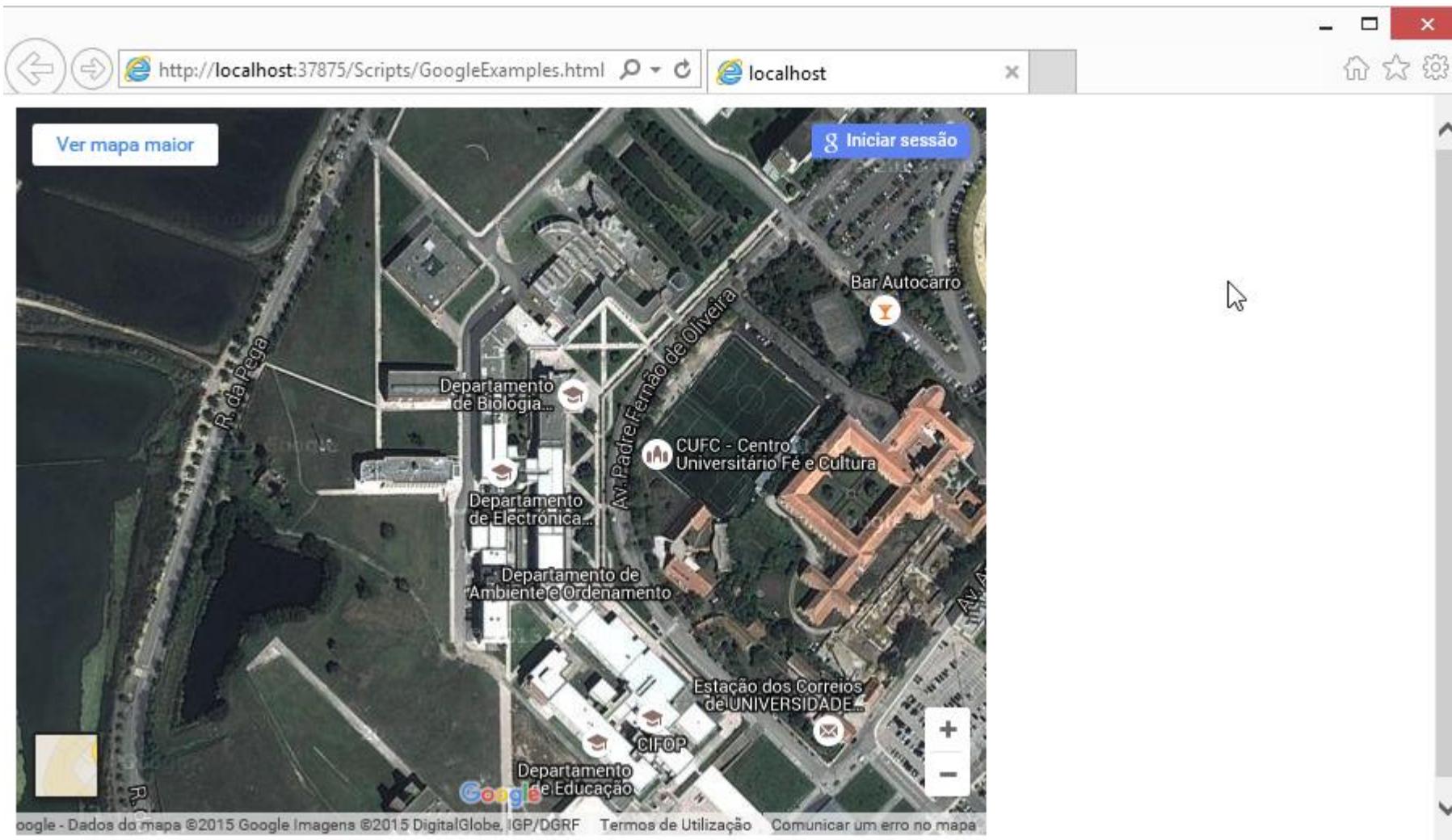
“maptype” – *tipo de mapa (roadmap, satellite)*

“language” – *linguagem a usar nos elementos da interface*

“region” – *mostra os limites apropriados, com base na sensibilidade geo-políticas (ccTLD – cc = código da região)*

## Exemplo:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <iframe width="600" height="450" frameborder="0" style="border:0"
        src="https://www.google.com/maps/embed/v1/view?key=AIzaSyDM0ZNlzZS6wulKF0v1s6PARvuXmGI-paM
            &center=40.633181,-8.6595
            &zoom=17
            &maptype=satellite"
        allowfullscreen></iframe>
</body>
</html>
```



# Google Maps

## JavaScript API v3

A **JavaScript API v3** do serviço Google Maps também realiza um pedido, em HTTP, para que seja devolvido um mapa dinâmico e interativo.

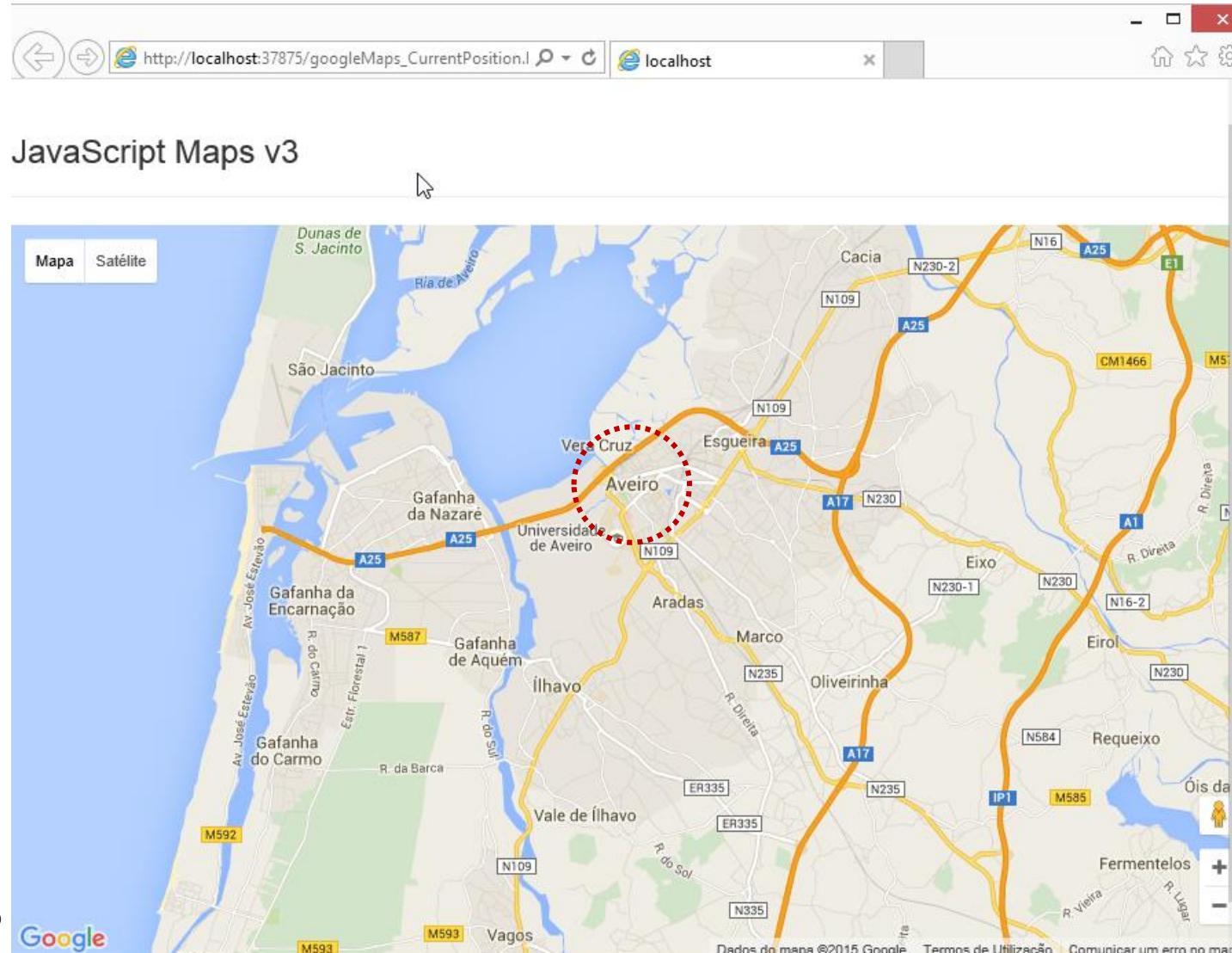
Que diferenças para a Embed API?

O mapa não necessita de ser colocado dentro de uma “iframe”; pode ser inserido em qualquer elemento HTML;

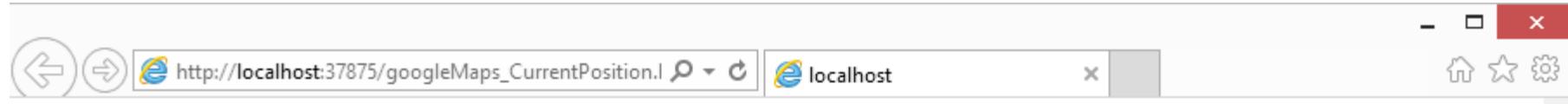
É possível um grau muito mais elevado de controlo e personalização do mapa.

```
<!DOCTYPE html>
<html>
<head>
    <link href="Content/bootstrap.min.css" rel="stylesheet" />
    <link href="Content/font-awesome.min.css" rel="stylesheet" />
    <style type="text/css">
        html, body { height: 100%; margin: 0; padding: 0; }
        #map { height: 80%; }
    </style>
    <script type="text/javascript">
        var map;
        function initMap() {
            map = new google.maps.Map(document.getElementById('map'), {
                center: { lat: 40.6413266, lng: -8.65363209999982 },
                zoom: 12
            });
        }
    </script>
</head>
<body>
    <div class="page-header">
        <h3>JavaScript Maps v3</h3>
    </div>
    <div id="map"></div>
    <script src="Scripts/jquery-2.1.4.min.js"></script>
    <script src="Scripts/bootstrap.min.js"></script>
    <script async defer src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDM0ZN1zZS6wulKF0v1s6PARvuXmGI-paM
        &callback=initMap">
    </script>
</body>
</html>
```

# Mapa para uma determinada coordenada



```
<!DOCTYPE html>
<html>
<head>
    <link href="Content/bootstrap.min.css" rel="stylesheet" />
    <link href="Content/font-awesome.min.css" rel="stylesheet" />
    <style type="text/css">
        html, body { height: 100%; margin: 0; padding: 0; }
        #map { height: 80%; }
    </style>
    <script type="text/javascript">
        var map;
        function initMap() {
            map = new google.maps.Map(document.getElementById('map'), {
                center: { lat: 40.6413266, lng: -8.653632099999982 },
                zoom: 12
            });
            if (navigator.geolocation) {
                navigator.geolocation.getCurrentPosition(
                    function (position) {
                        var pos = new google.maps.LatLng(position.coords.latitude, position.coords.longitude);
                        var infowindow = new google.maps.InfoWindow(
                            {
                                map: map, position: pos,
                                content: 'Estou aqui!<br/>Lat: ' + pos.lat().toString().substr(0, 8) +
                                    ' - Long: ' + pos.lng().toString().substr(0, 8)
                            }
                        );
                        map.setCenter(pos);
                    },
                    function () {
                        handleNoGeolocation(true);
                    }
                );
            }
        }
    </script>
</head>
```



## JavaScript Maps v3



# Exercícios

Crie uma página web, usando a Google Maps Embed API em conjunto com JavaScript, para permitir mostrar num mapa o seguinte:

Pesquisa de local;

Pesquisa de características no local;

Pesquisa de trajetos entre dois locais.

Use a biblioteca bootstrap para melhorar o aspeto da página que criou no exercício 1.

# Aspetto da página no exercício 1

Dica:

Para alterar o mapa dinamicamente, basta alterar o atributo "src" do elemento "iframe".

**Pesquisar em Portugal**



**Pesquisar:**

Por lugar:

Por características:

Por trajetos:

Origem:  Destino:

# Google Chart API

# O que é a Google Chart API?

A Google Chart API é uma ferramenta que permite criar um gráfico a partir de alguns dados e incorporá-lo em uma página da web.

A API do Google Chart cria uma imagem PNG de um gráfico a partir de dados e parâmetros de formatação enviados no pedido HTTP.

São suportados muitos tipos de gráficos e, fazendo a solicitação através de uma tag de imagem, é possível incluir o gráfico numa página da web.

Google charts bars - Pesquisa | Using Google Charts | Clique

https://developers.google.com/chart/interactive/docs/

Apps Which Programming... Conchas CSS & Fonts App Artigo Puzzles CRUD Web SJ ITW App Phonegap

Google Developers Charts Search rl8maio@gmail.com Sign out

Products > Charts

# Charts

HOME GUIDES REFERENCE SUPPORT

Overview

Hello, Charts!

- Quickstart
- Load the Libraries
- Prepare the Data
- Customize the Chart
- Draw the Chart

Chart Types

- Chart Gallery
- Annotation Charts
- Area Charts
- Bar Charts

## Using Google Charts

Google Charts provides a perfect way to visualize data on your website. From simple line charts to complex hierarchical tree maps, the [chart gallery](#) provides a large number of ready-to-use chart types.

The most common way to use Google Charts is with simple JavaScript that you embed in your web page. You load some Google Chart libraries, list the data to be charted, select options to customize your chart, and finally create a chart object with an id that you choose. Then, later in the web page, you create a <div> with that id to display the Google Chart.

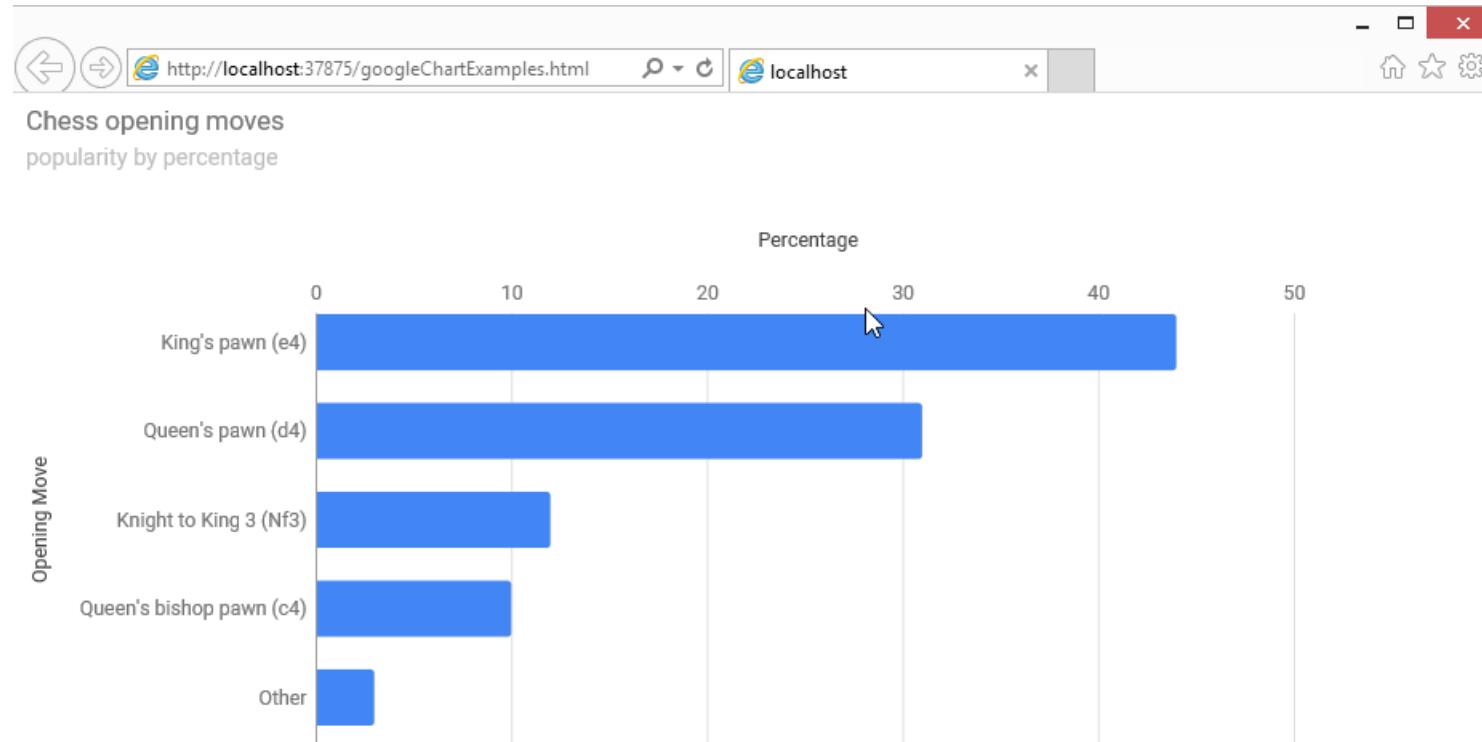
That's all you need to [get started](#).

★★★★★

25

# Google Charts

## Bars



```

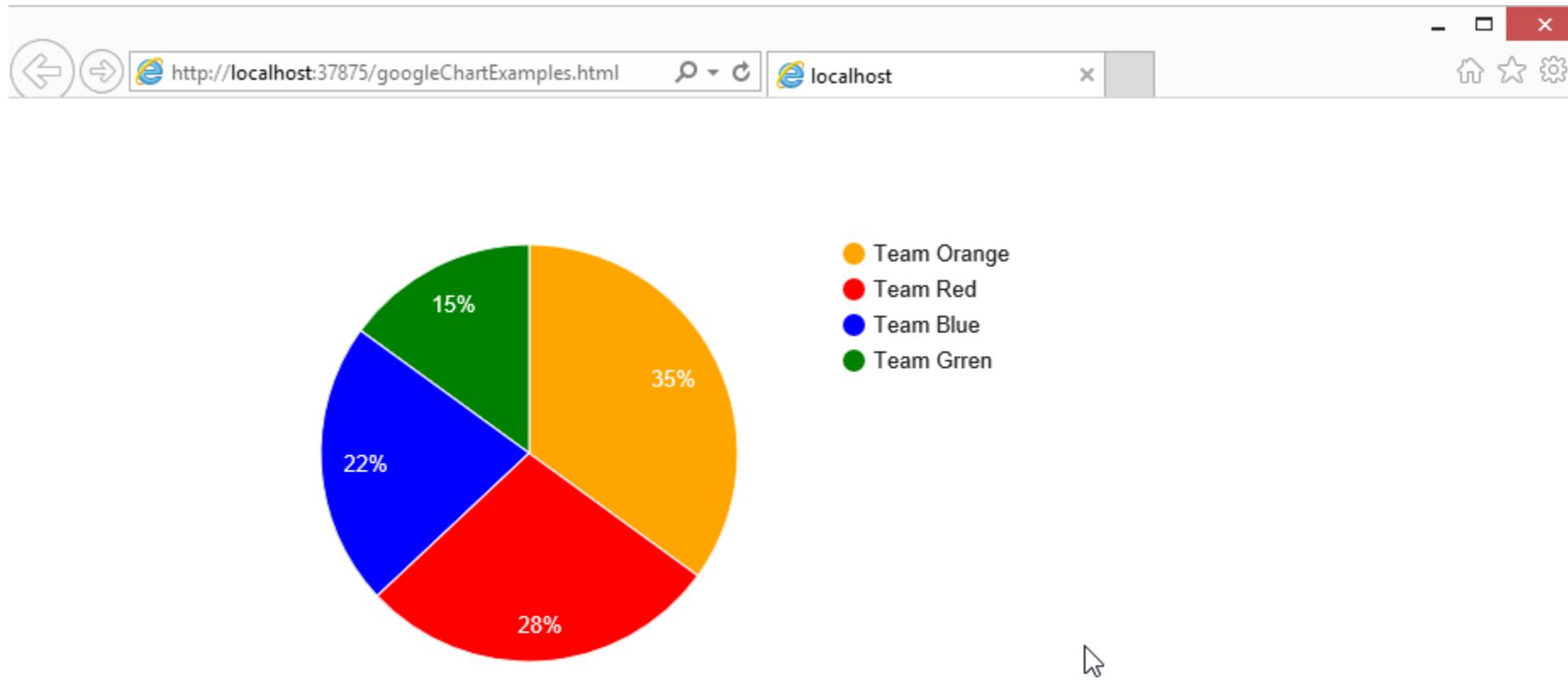
<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1.1", { packages: ["bar"] });
    google.setOnLoadCallback(drawStuff);
    function drawStuff() {
      var data = new google.visualization.arrayToDataTable([
        ['Opening Move', 'Percentage'],
        ["King's pawn (e4)", 44],["Queen's pawn (d4)", 31],["Knight to King 3 (Nf3)", 12],["Queen's bishop pawn (c4)", 10],['Other', 3]
      ]);
      var options = {
        title: 'Chess opening moves',
        width: 800,
        legend: { position: 'none' },
        chart: {
          title: 'Chess opening moves',
          subtitle: 'popularity by percentage'
        },
        bars: 'horizontal', // Required for Material Bar Charts.
        axes: {
          x: {
            0: { side: 'top', label: 'Percentage' } // Top x-axis.
          }
        },
        bar: { groupWidth: "90%" }
      };

      var chart = new google.charts.Bar(document.getElementById('top_x_div'));
      chart.draw(data, options);
    };
  </script>
</head>
<body>
  <div id="top_x_div" style="width: 800px; height: 400px;"></div>
</body>
</html>

```

# Google Charts

## Pie Chart



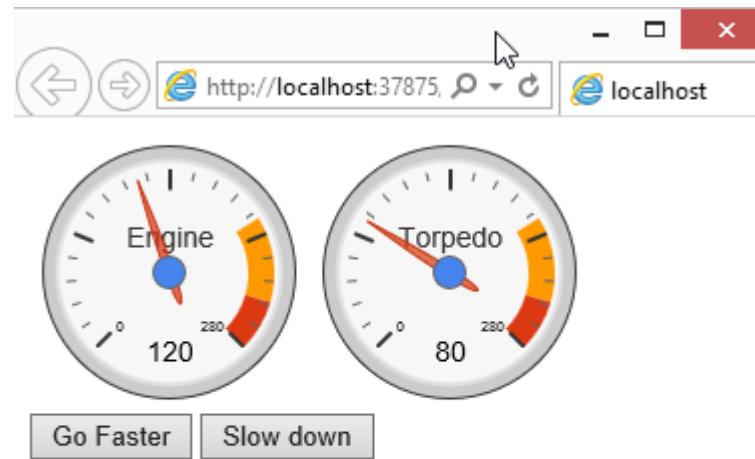
```

<html>
<head>
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>
    <script type="text/javascript">
        google.load("visualization", "1", { packages: ["corechart"] });
        google.setOnLoadCallback(drawChart);
        function drawChart() {
            var data = google.visualization.arrayToDataTable([
                ['Main National Teams', 'Percentage'],
                ['Team Orange', 35],
                ['Team Red', 28],
                ['Team Blue', 22],
                ['Team Grren', 15]
            ]);
            var options = {
                slices: {
                    0: { color: 'orange' },
                    1: { color: 'red' },
                    2: { color: 'blue' },
                    3: { color: 'green' }
                }
            };
            var chart = new google.visualization.PieChart(document.getElementById('NationalTeams'));
            chart.draw(data, options);
        }
    </script>
</head>
<body>
    <div id="NationalTeams" style="width: 800px; height: 400px;"></div>
</body>
</html>

```

# Google Charts

## Gauges



```
<html>
<head>
    <script type="text/javascript"
        src="https://www.google.com/jsapi?autoload={'modules':[{'name':'visualization','version':'1','packages':['gauge']}]}>
    </script>
    <script type="text/javascript">
        google.load('visualization', '1', { packages: ['gauge'] });
        google.setOnLoadCallback(drawGauge);

        var gaugeOptions = {
            min: 0, max: 280, yellowFrom: 200, yellowTo: 250,
            redFrom: 250, redTo: 280, minorTicks: 5
        };

        var gauge;
        function drawGauge() {
            gaugeData = new google.visualization.DataTable();
            gaugeData.addColumn('number', 'Engine');
            gaugeData.addColumn('number', 'Torpedo');
            gaugeData.addRows(2);
            gaugeData.setCell(0, 0, 120);
            gaugeData.setCell(0, 1, 80);

            gauge = new google.visualization.Gauge(document.getElementById('gauge_div'));
            gauge.draw(gaugeData, gaugeOptions);
        }

        function changeTemp(dir) {
            gaugeData.setValue(0, 0, gaugeData.getValue(0, 0) + dir * 25);
            gaugeData.setValue(0, 1, gaugeData.getValue(0, 1) + dir * 20);
            gauge.draw(gaugeData, gaugeOptions);
        }
    </script>
</head>
<body>
    <div id="gauge_div" style="width:280px; height: 140px;"></div>
    <input type="button" value="Go Faster" onclick="changeTemp(1)" />
    <input type="button" value="Slow down" onclick="changeTemp(-1)" />
</body>
</html>
```

# Exercícios

Experimente o exemplo dos diversos tipos de gráficos e veja as diversas propriedades que cada um suporta.

No “Pie Chart” remova a legenda lateral e inclua um título no gráfico (não na página!!!!)

No “Gauge” garanta que os valores não excedem (abaixo de zero ou acima do máximo) quando se varia a velocidade (para cima e para baixo)