

A light gray world map serves as the background. It features several location pins: a large blue pin with a black dot in the center of Europe, and several smaller red pins with white dots scattered across Europe, the Middle East, and East Asia. In the top left corner, there are five smaller blue pins with white dots clustered together.

# WEBINAR BIGDATA - WEBMAPPING

*Big Data – Visualização em Dados em mapas digitais*

*31 de janeiro – 19 horas*

*Ricardo José Vieira Baptista*



# OBJETIVOS

ORADOR: RICARDO BAPTISTA

[ricardo.baptista@my.istec.pt](mailto:ricardo.baptista@my.istec.pt)

- **conhecer a informação geográfica** no seu detalhe (dados *vetoriais* e dados *rasters*)
- **conhecer as ferramentas** de visualização de informação geográficas
- **desenvolver mapas em páginas web** através da framework javascript (JS) “*Leaflet*”
- **personalizar os mapas interativos** com pontos de interesse, descrição, e outros elementos.
- utilizar **diversos tipos de camadas:** *OpenStreetMaps*, e outros serviços e mapas disponíveis (WFS).

# AGENDA

- Introdução à informação geográfica
- *Web mapping*: conceito e aplicações
- Componentes do *Web mapping*
  - HTML
  - Programação Javascript (JS)
  - Bibliotecas de mapas e API's
  - Serviços de dados/ mapas web
- Exercícios Tutoriais

# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

- Os mapas têm um **leque variado de utilizações** desde sempre.
- Desde o início, a **tecnologia tem influenciado a criação** de mapas e como os mapas são utilizados. À medida que a tecnologia avança, os processos de cartografia e de mapeamento de informação evoluíram paralelamente com ela, desde mapas desenhados manualmente em papiros até mapas interativos na web.



*Mapa do passado*

# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

- **A segunda metade do século 20** foi um ponto de viragem para a cartografia.
- Os computadores foram tornando-se numa ferramenta cada vez mais universal, bem como a **fotografia aérea, imagens de satélite e deteção remota** mudaram a forma como os dados espaciais são obtidos.
- Aparecem os **Sistemas de Informação Geográfica (SIG)**, com os objetivos de capturar, armazenar, manipular, gerir e visualizar os dados espaciais ou geográficos.
- Com o evoluir da Internet e da Web, os mapas disponibilizados nos SIG começaram a **transferir-se da área de trabalho para a Web**, e desta forma os grandes fornecedores de SIGs começaram a fazer as primeiras estruturas para os mapas on-line.

# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

80% da informação digital está associada a uma localização geográfica!

The screenshot shows a Google search interface. The search bar contains 'museu soares dos reis' and indicates approximately 136,000 results. The left sidebar includes navigation options like 'Tudo', 'Imagens', 'Vídeos', 'Notícias', and 'Mais'. Below the search results, there is a map of Porto, Portugal, with a red pin marking the location of the National Museum Soares dos Reis. To the right of the map is a location card for 'National Museum Soares dos Reis' with details such as the address 'R. de Dom Manuel II 56, 4050 Porto', phone number '223 393 770', and a rating of 10 reviews. Below the map and location card, there are search results for 'Museu Nacional Soares dos Reis - Informações Úteis' and 'Museu Soares dos Reis'.

# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

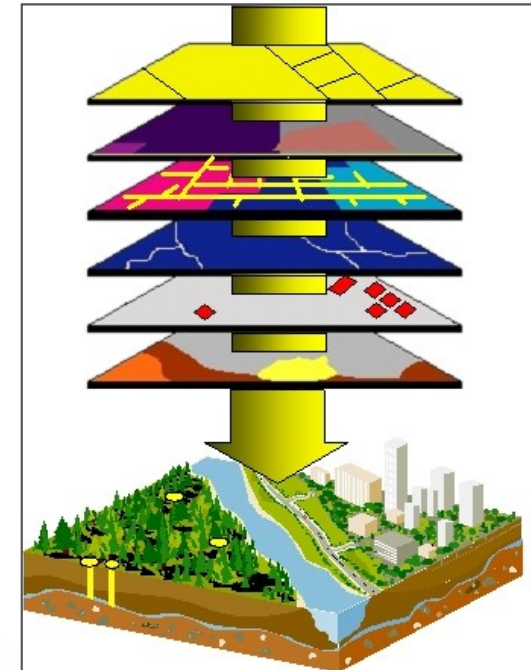
- **Mas** o mapeamento de SIG **não é uma tarefa fácil**.
- Requer muitas tecnologias do lado do servidor, padrões geoespaciais e protocolos, juntamente com as suas implementações.
- Requer **a compreensão** do que são dados geoespaciais, projeções de mapas e outros conhecimentos de como:
  - recolher os dados,
  - exibir os dados,
  - generalizar os dados em escalas específicas,
  - colocar rótulos no mapa,
  - configurar um servidor que disponibilizará os mapas,
  - usar uma base de dados espacial, entre outras questões.



# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

*Informação Geográfica = Representação abstrata da realidade*

- Informação gráfica
- Atributos alfanuméricos.

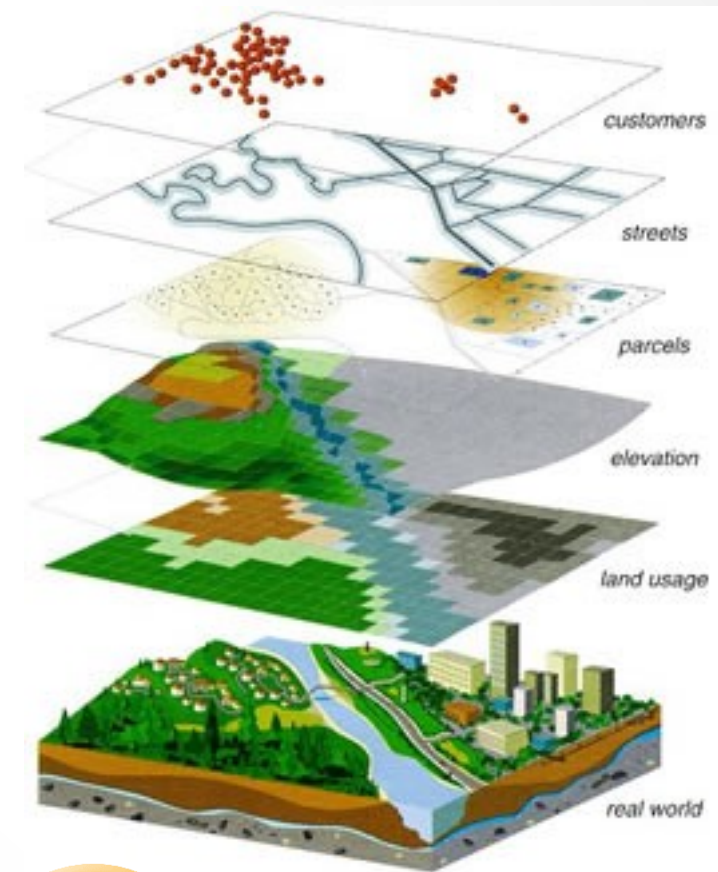
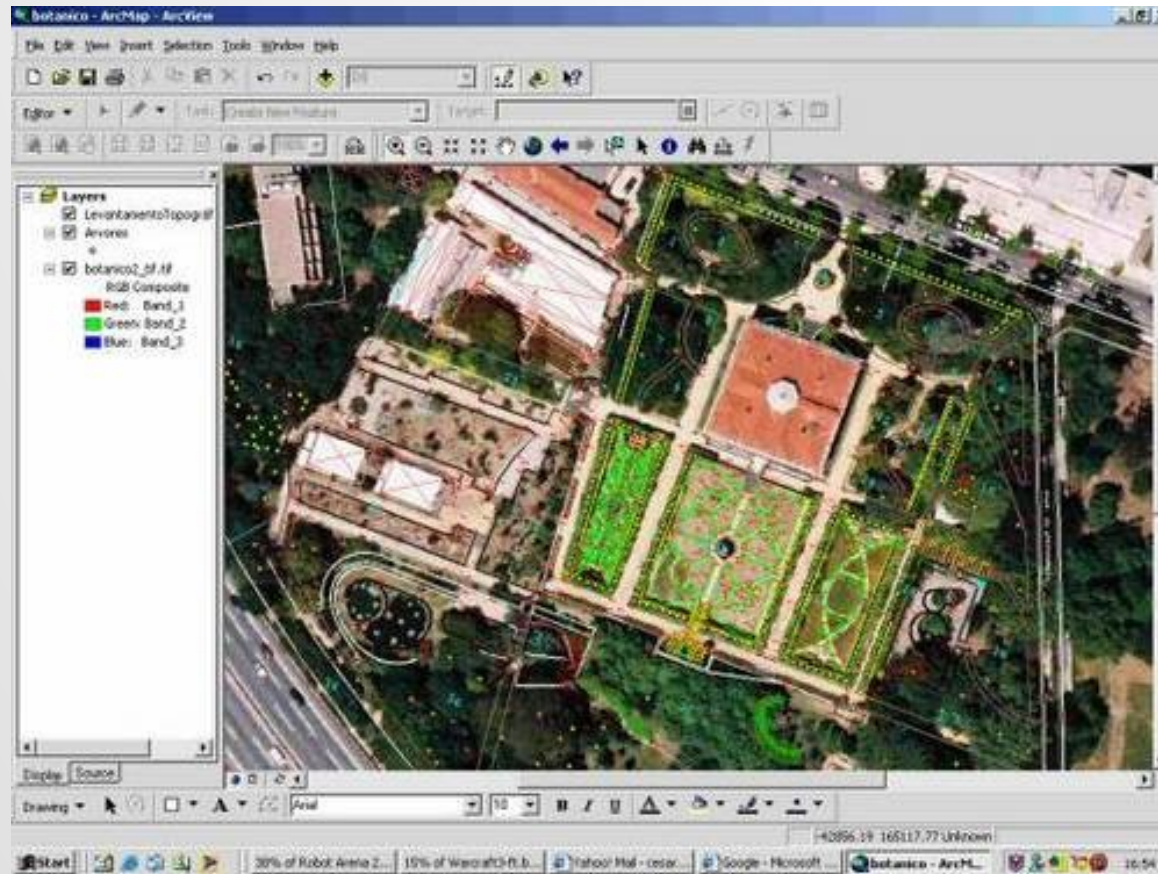


- Organização por temas geográficos.



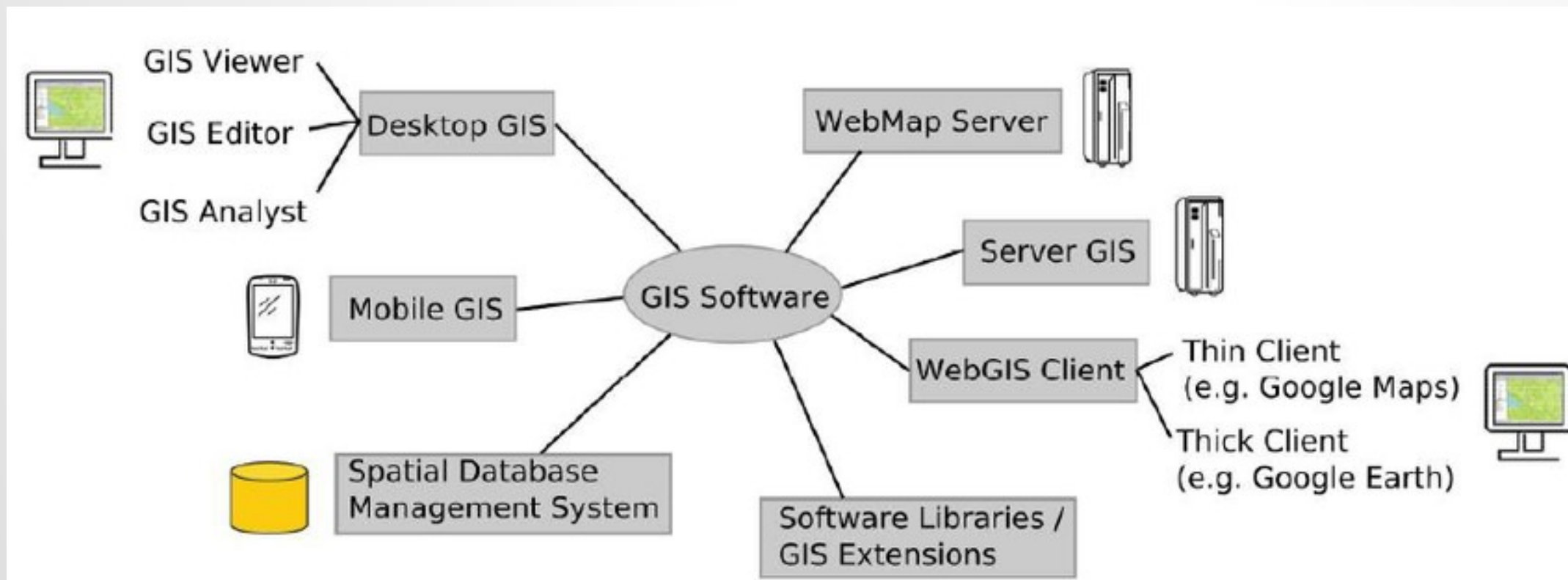
# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

## Software SIG



# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

## Software SIG (Arquitetura)



# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

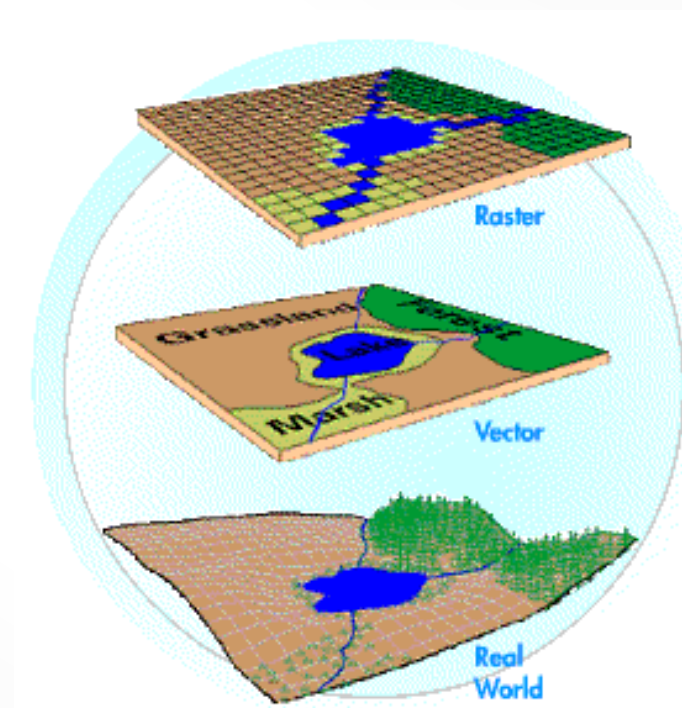
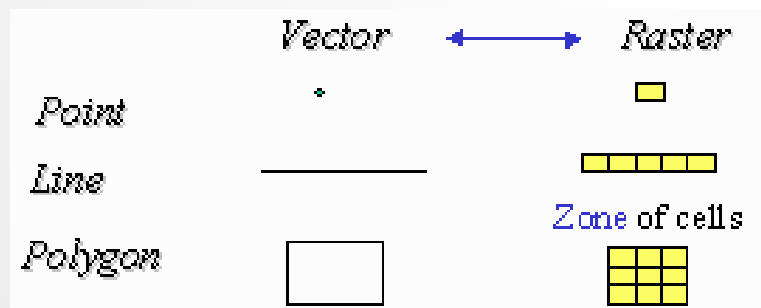
## Modelos de Dados Geográficos

- Modelo Raster

- A informação é representada por células de uma grelha uniforme, com determinada resolução.

- Modelo Vetorial

- A informação é representada por coordenadas definindo pontos, linhas ou áreas.



# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

- O **GPS** é um sistema de posicionamento geográfico que nos dá as coordenadas de um lugar na Terra, desde que tenhamos um recetor de sinais de GPS.
- Este sistema foi desenvolvido pelo Departamento de Defesa Americano para ser utilizado com fins civis e militares.
- O GPS trabalha **em qualquer condição atmosférica**, em **qualquer lugar no mundo**, **24 horas por dia**, e **não é cobrada nenhuma taxa** para se usar o GPS.

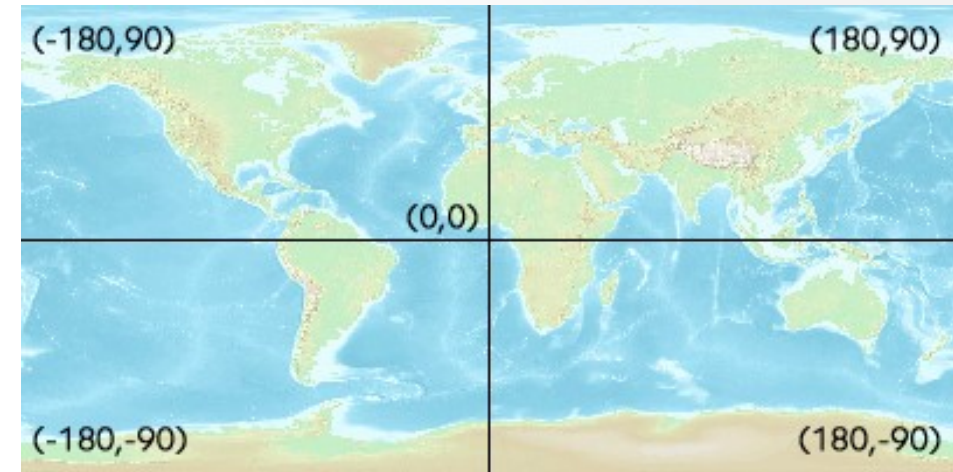


Constelação de satélites GPS na sua órbita em torno da Terra.



# A EVOLUÇÃO DO MAPEAMENTO ONLINE

- A nossa posição sobre a Terra é referenciada em relação ao **equador** e ao **meridiano de Greenwich** e traduz-se por três números: a latitude, a longitude e a altitude.



- **Latitude:** é a distância ao Equador medida ao longo do meridiano de Greenwich. Esta distância pode variar **entre 0° e 90° para Norte ou para Sul**. Por exemplo, Lisboa está à latitude de 38° 4'N.
- **Longitude:** é a distância ao meridiano de Greenwich medida ao longo do Equador. Esta distância pode variar **entre 0° e 180° para Leste ou para Oeste**. Por exemplo, Lisboa está à longitude de 9° 8'W.

# A EVOLUÇÃO DO MAPEAMENTO ONLINE

- Os primeiros mapas da Web geralmente **mostraram apenas uma imagem de mapa única** e muito pequena.
- Naquele momento, a panorâmica foi implementada movendo um passo, geralmente por metade do tamanho do mapa, em uma das oito direções possíveis da bússola - N, NW, W, SW, S, SE, E, NE.
- Nesta fazer para conseguir uma imagem inteiramente nova precisaria de uma **renderização no servidor do mapa**, sendo este **carregado/transmitido na rede** e depois **processado pelo navegador**.

# A EVOLUÇÃO DO MAPEAMENTO ON-LINE

- O Google transformou o mundo do mapeamento de cabeça para baixo quando introduziu o **Google Maps** a partir de 2005. O Google disponibilizou também um conjunto de scripts, para que os utilizadores pudessem colocar os **mapas do Google nos seus próprios sites** e **adicionar seus próprios dados** ao mapa.
- De repente, os mapas on-line tornam-se populares; e para adicionar um mapa ao seu site, não precisa de ser um cartógrafo ou um especialista em SIG. Tudo hoje reflete-se num mapa: orientação, pontos de interesse, etc.



# INTRODUÇÃO À INFORMAÇÃO GEOGRÁFICA

- Com o aparecimento da “**Web 2.0**” e as tecnologias como **HTML5**, **CSS3** e **SVG**, tornou fácil a criação de conteúdo interativo, e os mapas interativos estão vivendo um novo renascimento.



*Mapa do futuro?*

# DADOS DO MAPA

- Inicialmente, os mapas on-line iniciais começaram baseados em **conjuntos de dados SIG** e **suas base de dados espaciais** (geodatabases). Muitas pessoas não tinham acesso a esses dados, além que os custos de acesso eram elevados.
- Os **mapas disponibilizados pela Google** e os dados geográficos do projeto **OpenStreetMap (OSM)** mudaram isso. A base de dados do Google é privada e vem com restrições, contudo o OSM inspirou-se no conceito de Wikipédia, como um **projeto colaborativo para criar um mapa gratuito do mundo**.
- O OpenStreetMap é construído por uma comunidade de mapeadores voluntários, que contribuem e mantêm os dados espaciais.

# DADOS DO MAPA

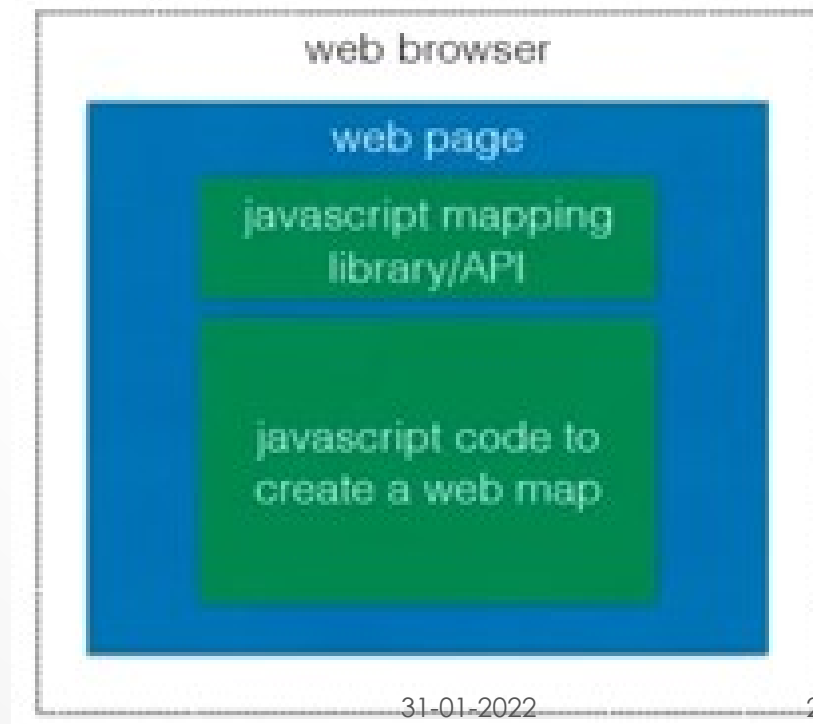
- Em suma, a maior revolução que contribuiu para a fácil criação de mapas foi a solução ao problema relacionado com a manutenção de um servidor de mapas e a base de dados espacial. **Os dados hoje estão disponíveis de várias formas.**
- *Plataformas de webmapping*
  - Google, MapQuest, Microsoft e Here (comerciais)
  - Mapeamento de código aberto baseado no OpenStreetMap (OpenLayer e Leaflet)
  - MapBox e CartoDB, utilizam o OSM mas não são gratuitos (oferecem curadoria dos dados)

# WEBMAPPING

- O que é um **aplicativo de mapeamento web (webmapping)** e como ele funciona?
  - É uma aplicação construída usando tecnologias web (HTML / CSS/ Javascript) e biblioteca de mapeamento Javascript
- Componentes do Webmapping:
  - Web page
  - Linguagem de programação Javascript
  - Biblioteca de mapeamento e APIs

# COMPONENTES DO WEBMAPPING

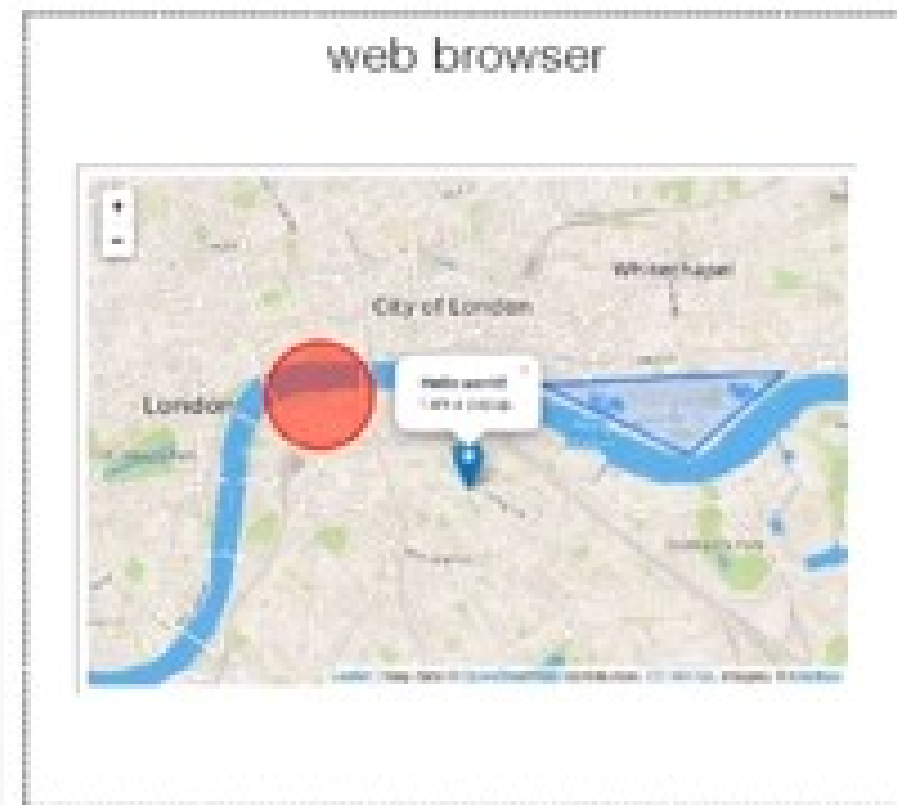
- Componentes do Webmapping:
  - Paginas Web
  - Linguagem de programação Javascript
  - Biblioteca de mapeamento e APIs



# COMPONENTES DO WEBMAPPING

- Uma página da web é um documento de texto simples que usa etiquetas HTML para definir estrutura e elementos de uma página.
- Uma página da Web criada usando apenas HTML **é um documento estático**

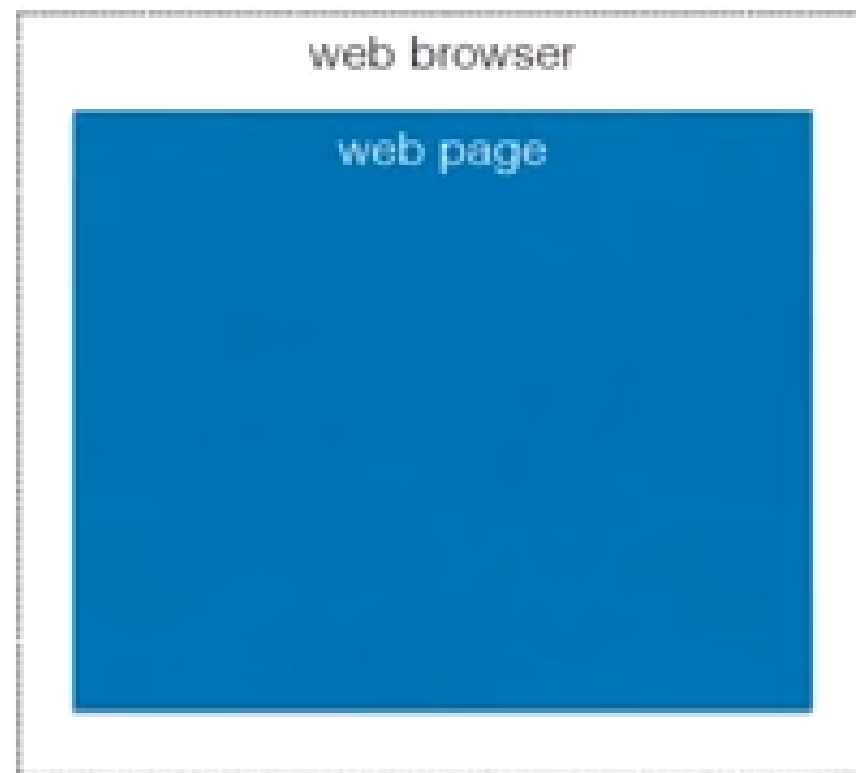
- Página web



# COMPONENTES DO WEBMAPPING

- Um **ficheiro de texto** (com extensão .htm e html)
- Escrita na linguagem de marcadores HTML5
- Uso de **tags** (anotações) para definir um novo elemento na página Web
- O **navegador interpreta** as anotações e torna-as visuais
- Uma página web só com HTML é estática

- Página web

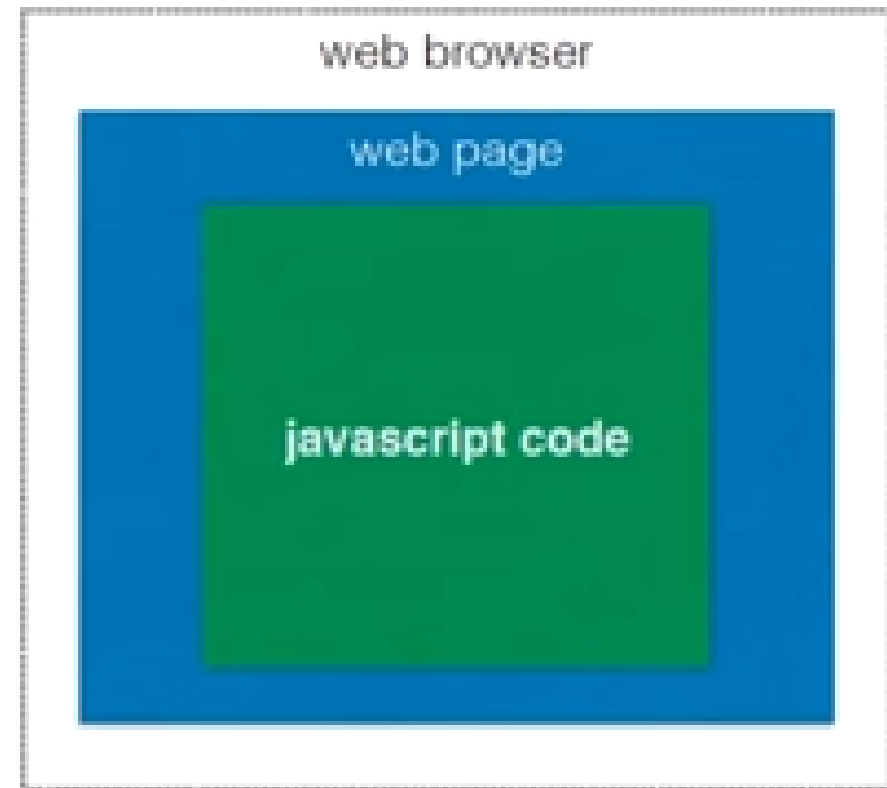




# COMPONENTES DO WEBMAPPING

- É uma linguagem de programação interpretada por praticamente todos os browser Web
- É colocado numa página da web dentro de uma etiqueta especial (<script>...</script>)
- E quando uma página é carregada, o navegador da Web executa-o.
- Para criar um mapa na web, é precisa usar um pouco de javascript e biblioteca de mapeamento

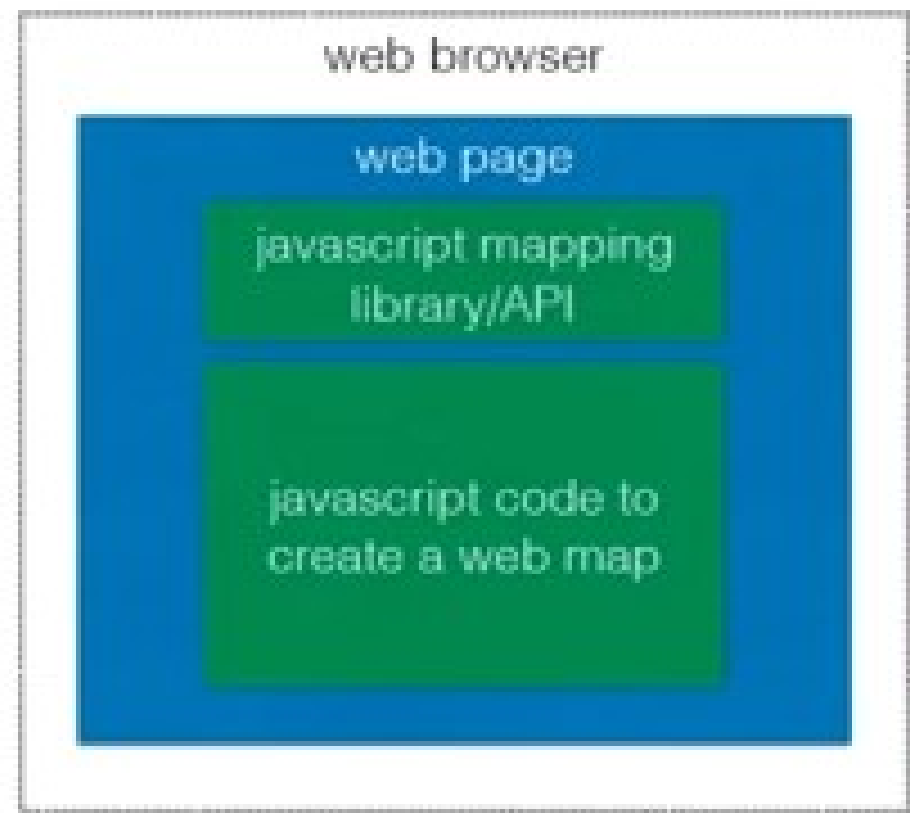
- Javascript



# COMPONENTES DO WEBMAPPING

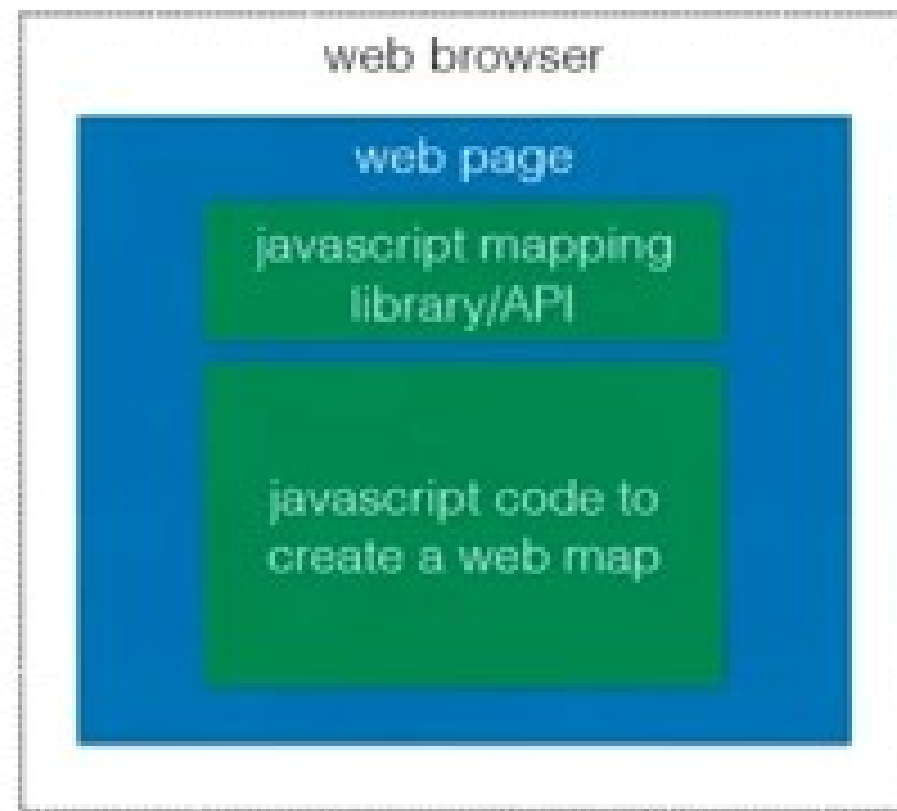
- Um conjunto de código javascript que **promove o interação com o mapa na página**
- **Não precisa saber como funciona** (framework)
- A única coisa que **precisa de precisa saber é como usá-lo**
- Necessário saber como **usar as bibliotecas de específicas, API** (Application Programming Interface)

- Biblioteca de mapeamento



# COMPONENTES DO WEBMAPPING

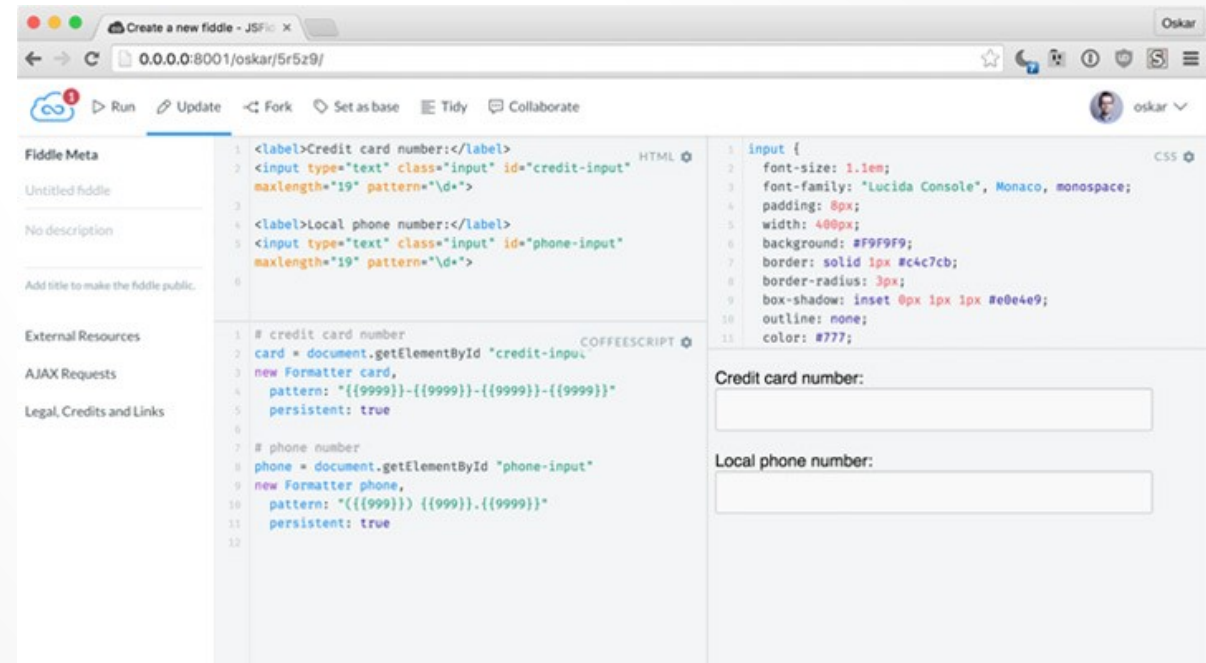
- Pode usar bibliotecas/frameworks com qualquer mapa ou serviços de dados quer grátis/aberto e comercial
  - Precisa de aceder/alocar os seus dados (usando um ficheiro geojson ou dados de um servidor dados e base de dados)
  - Ex. *Open Layers* e *Leaflet*
- Biblioteca de mapeamento



# FERRAMENTAS PARA CONSTRUIR WEBMAPPING

- **jsFiddle** permite aos programadores trabalhar com os três elementos principais do desenvolvimento da Web.
- É uma poderosa aplicação para prototipagem rápida e testes.
- Fornece um ambiente de desenvolvimento personalizado para testar o seu código **JavaScript, HTML e CSS** diretamente no seu navegador.

- <https://jsfiddle.net/>



# FERRAMENTAS PARA CONSTRUIR WEBMAPPING

## Framework JS Leaflet

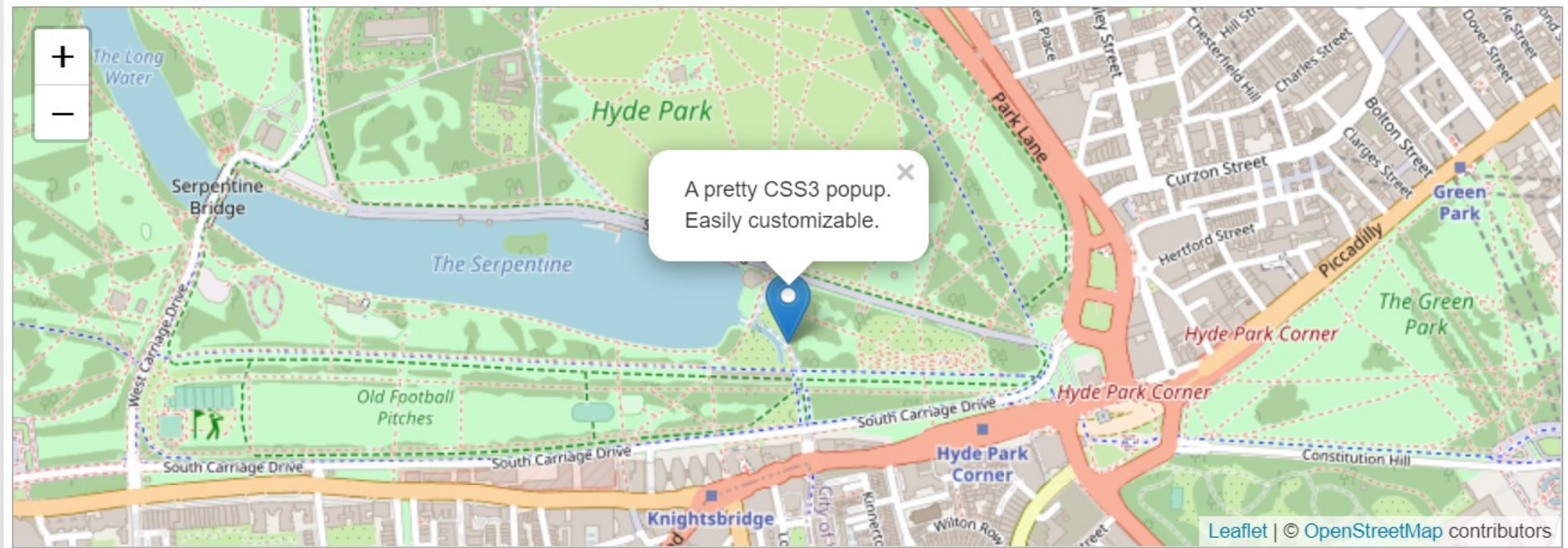
- **Biblioteca de código aberto** que nos permite utilizar mapas web simples, interativos e leves.
  - **Permite-lhe utilizar** camadas tais como camadas de Tiles, WMS, Marcadores, Popups, camadas vetoriais (pontos, polilinhas, polígonos, etc.), sobreposições de imagens e GeoJSON.
- <https://leafletjs.com/>
- Pode interagir com os mapas arrastando o mapa, fazendo **zoom** (por duplo clique ou, scroll de roda), usando o teclado, usando a **manipulação de eventos, e arrastando os marcadores.**
  - **É suportado por navegadores** como o Chrome, Firefox, Safari 5+, Opera 12+, IE 7-11 no ambiente de trabalho e, navegadores como Safari, Android, Chrome, Firefox para telemóveis.



# FERRAMENTAS PARA CONSTRUIR WEBMAPPING

Framework JS Leaflet

<https://leafletjs.com/>



# EXERCÍCIOS TUTORIAIS

- **ExTutorial 1:** Estrutura base do ficheiro HTML
- **ExTutorial 2:** Introdução dos estilos e scripts da biblioteca JS Leaflet
- **ExTutorial 3:** Introdução do *container (div)* para o mapa centrado num ponto
- **ExTutorial 4:** Apresentação de um marcador no mapa
- **ExTutorial 5:** Apresentação de vários marcadores no mapa
- **ExTutorial 6:** Adicionar linha no mapa
- **ExTutorial 7:** Adicionar polígono no mapa
- **ExTutorial 8:** Adicionar legenda ao dados
- **ExTutorial 9:** Adicionar um *popup* informativo sobre o mapa/ seletor de apresentação das coordenadas
- **ExTutorial 10:** Carregar um ficheiro GeoJson (*polígonos*)
- **ExTutorial 11:** Carregar vários ficheiros GeoJson para o mapa (pontos)
- **ExTutorial 12:** Criar um mapa de calor (heatmap)
- **ExTutorial 13:** Criar um mapa by API



# EXERCÍCIOS TUTORIAIS

Download de ficheiros de apoio e pdf Apresentação

- <https://tinyurl.com/56zbxuch>

# EXERCÍCIO TUTORIAL 1

## Estrutura base de uma *web page*

• Webmap\_app1.html

- Tags

- **<html> ... </html>** para indicar todo o ficheiro
- **<head> ... </head>** para configurações e outras indicações invisíveis no navegador
- **<body> ... </body>** para os conteúdos visíveis no navegador

```
1 <!DOCTYPE html>
2 <head>
3     <title>Primeira aplicação Webmapping -Leaflet</title>
4     <meta charset="UTF-8">
5 </head>
6 <body>
7
8 </body>
9 </html>
```

*Abrir num navegador web...*

# EXERCÍCIO TUTORIAL 2

## Tags

- **<title> ... </title>**
- **<meta ... >**
- **<link ... >**: chamada de ficheiro externo do tipo de estilos css
- **<script> ... </script>**: chamada de ficheiro externo do tipo de scripts javascript
- **<style> ... </style>**: definição de estilo específico na própria página

- Webmap\_app2.html

```
1  <!DOCTYPE html>
2  <head>
3      <title>Primeira aplicação Webmapping</title>
4      <meta charset="UTF-8">
5      <link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css" crossorigin=""/>
6      <script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js" crossorigin=""></script>
7
8      <style>
9          #mapDIV{
10             height: 500px;
11             width: 500px;
12             border: solid 1px ■black;
13         }
14     </style>
15 </head>
16 <body>
17 |
18 </body>
19 </html>
```

Voltando a abrir num navegador web... Qual o resultado?

# EXERCÍCIO TUTORIAL 3 - MAPA

Tags

- basemap

- Webmap\_app3.html

```
17 <body>
18   <div id='mapDIV'>i</div>
19   <script>
20     let map = L.map(document.getElementById('mapDIV'),
21     { center: [41.15073,-8.63247],
22       zoom: 13
23     });
24     let basemap =L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png',
25     {
26     });
27     basemap.addTo(map);
28   </script>
29 </body>
30 </html>
```

*Voltando a abrir num navegador web... Qual o resultado?*

# EXERCÍCIO TUTORIAL 3 - MAPA

## Tags

- `<h2> Mapa do Porto </h2>`
- `<div id="..." > ...</div>`
- `<script> ... </script>`:  
script com funções *Leaflet* que permite criar o mapa associado a um elemento HTML a partir do DOM:
  - Center
  - Zoom

- Webmap\_app3.html

```
17 <body>
18   <div id='mapDIV'>i</div>
19   <script>
20     let map = L.map(document.getElementById('mapDIV'),
21     { center: [41.15073,-8.63247],
22     |   zoom: 13
23     });
24     let basemap =L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png',
25     |   {}
26     );
27     basemap.addTo(map);
28   </script>
29 </body>
30 </html>
```

*Voltando a abrir num navegador web... Qual o resultado?*

# EXERCÍCIO TUTORIAL 3 - MAPA

Tarefa:

- Verificar com **outro ponto** para o centro do mapa (*através de novo ponto retirado do Google Maps*)
- Limitar o zoom máximo e zoom mínimo do mapa (níveis de zoom)
  - minZoom: 8,
  - maxZoom: 14

- Webmap\_app3.html

```
17 <body>
18   <div id='mapDIV'>i</div>
19   <script>
20     let map = L.map(document.getElementById('mapDIV'),
21     { center: [41.15073,-8.63247],
22       zoom: 13
23     });
24     let basemap =L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png',
25     {
26     });
27     basemap.addTo(map);
28   </script>
29 </body>
30 </html>
```

*Voltando a abrir num navegador web... Qual o resultado?*

# EXERCÍCIO TUTORIAL 4 - MARCADOR

## Definir o marcador/ponto

- Definir as propriedades de ambos

**marker:** [long/lat]

*Voltando a abrir num navegador web... Qual o resultado?*

- Webmap\_app4.html

```
17 <body>
18   <div id='mapDIV'>i</div>
19 <script>
20   let map = L.map(document.getElementById('mapDIV'),
21   { center: [41.15073, -8.63247],
22     zoom: 13
23   });
24   let basemap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png',
25   {
26     });
27   basemap.addTo(map);
28   // Create point feature for Casa da Música
29   var myDataPoint = L.marker([41.158799, -8.631102]).addTo(map);
30 </script>
31 </body>
32 </html>
```



# EXERCÍCIO TUTORIAL 5 - MARCADORES

- Criar marcadores para as seguintes capitais europeias:
  - Roma: [41.8829, 12.4850]
  - Madrid: [40.4299, -3.70221]
  - Londres: [51.5254, -0.127722]
  - Lisboa: [38.7258, -9.13996]
- Utilizem como base o exercício anterior, webmap\_app4.html
- Crie o mapa com o centro definido em Madrid

- Webmap\_app5.html

```
17 <body>
18   <div id='mapDIV'>i</div>
19   <script>
20     let map = L.map(document.getElementById('mapDIV'),
21     { center: [41.15073, -8.63247],
22       zoom: 13
23     });
24     let basemap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png',
25     {
26     });
27     basemap.addTo(map);
28     // Create point feature for Casa da Música
29     var myDataPoint = L.marker([41.158799, -8.631102]).addTo(map);
30   </script>
31 </body>
32 </html>
```

# EXERCÍCIO TUTORIAL 6 – CRIAR LINHA

- Criar uma linha no mapa com a ponte da Arrábida (Porto):
- Vetor de pontos  
[[41.149303, -8.640834],  
[41.143943, -8.639520]]

*Voltando a abrir num navegador web... Qual o resultado?*

```
17 <body>
18   <div id='mapDIV'>i</div>
19   <script>
20     let map = L.map(document.getElementById('mapDIV'),
21     { center: [41.15073, -8.63247],
22     |   zoom: 13
23     });
24     let basemap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png',
25     |   {});
26     basemap.addTo(map);
27     // Create point feature for Casa da Música
28     var myDataPoint = L.marker([41.158799, -8.631102]).addTo(map);
29     // Create line feature for Arrabida Bridge, style and add to map
30     var myDataLine = L.polyline([
31     |   [41.149303, -8.640834],
32     |   [41.143943, -8.639520]],
33     |   {color: 'red', weight: 10}).addTo(map);
34   </script>
35 </body>
36 </html>
```

# EXERCÍCIO TUTORIAL 7 – CRIAR POLÍGONO

- Criar uma área delimitada no mapa:
- Vetor de pontos

```
[[41.151671, -8.632632],  
[41.151323, -8.632128],  
[41.150851, -8.630465],  
[41.150733, -8.630508],  
[41.150665, -8.630615],  
[41.150289, -8.630894],  
[41.150095, -8.631313],  
[41.150762, -8.633780],  
[41.151671, -8.632632]]
```


*Voltando a abrir num navegador web... Qual o resultado?*

- Webmap\_app7.html

```
24 let basemap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png',  
25   {});  
26   basemap.addTo(map);  
27   // Create point feature for Casa da Música  
28   var myDataPoint = L.marker([41.158799, -8.631102]).addTo(map);  
29   // Create area feature for Campus FLUP, style and add to map  
30   var myArea = L.polygon([[41.151671, -8.632632],  
31     [41.151323, -8.632128],  
32     [41.150851, -8.630465],  
33     [41.150733, -8.630508],  
34     [41.150665, -8.630615],  
35     [41.150289, -8.630894],  
36     [41.150095, -8.631313],  
37     [41.150762, -8.633780],  
38     [41.151671, -8.632632]],  
39     {color: 'blue', weight: 4}).addTo(map);  
40   </script>  
41 </body>  
42 </html>
```

# EXERCÍCIO TUTORIAL 8 – LEGENDA

- Criar legendas sobre os elementos criados:
- Webmap\_app8.html
- bindPopup

```
40 
41 // Bind popup to data point object
42 myDataPoint.bindPopup("<h3>Casa da Música</h3><p>Porto, Portugal <br>
43 |   Informações sobre a Casa da Música.</p>");
44 // Bind popup to line object
45 myDataLine.bindPopup("Ponte da Arrábida");
46 // Bind popup to area object
47 myArea.bindPopup("Campus FLUP");
48 </script>
49 </body>
50 </html>
```



*Voltando a abrir num navegador web... Qual o resultado?*

# EXERCÍCIO TUTORIAL 9 – CRIAR POPUP

- Criar legendas sobre os elementos criados:

- L.popup
- função onMapClick(e)

- Webmap\_app9.html

```
54 
55
56
57
58
59 v
60
61
62
63
64
65
66
67 
```

```
// Create an Empty Popup
var popup = L.popup();

// Write function to set Properties of the Popup
function onMapClick(e) {
    popup
        .setLatLng(e.latlng)
        .setContent("Clicou no mapa em " + e.latlng.toString())
        .openOn(map);
}

// Listen for a click event on the Map element
map.on('click', onMapClick);
```

*Voltando a abrir num navegador web... Qual o resultado?*

# EXERCÍCIO TUTORIAL 10 – CARREGAR UM FICHEIRO GEOJSON PARA O MAPA

- Criar uma camada (vetor) através da informação existente num ficheiro GeoJSON
- `<script src="..."></script>`
- `L.geoJSON(freg_layer, {  
    color: '#800000',  
    weight: 2,  
    dashArray: '12 8 12',  
});`
- `bindTooltip`

- Webmap\_app10.html

```
7 <script src="freg_porto_polig.geojson"></script>  
8 ...  
9  
10 var counties= L.geoJSON(freg_layer, {  
11     color: '#800000',  
12     weight: 2,  
13     dashArray: '12 8 12',  
14 });  
15 counties.bindTooltip('freguesias');  
16 counties.addTo(map);  
17 </script>
```



# EXERCÍCIO TUTORIAL 10 – CARREGAR UM FICHEIRO GEOJSON PARA O MAPA

- Criar uma camada (vetor) através da informação existente num ficheiro GeoJson:

- `<script src="..."></script>`

```
- L.geoJSON(freg_layer, {  
    color: '#800000',  
    weight: 2,  
    dashArray: '12 8 12',  
});
```

- `bindTooltip`

- Webmap\_app10.html

```
7      <script src="freg_porto_polig.geojson"></script>  
8      ...  
9      var counties= L.geoJSON(freg_layer, {  
10         color: '#800000',  
11         weight: 2,  
12         dashArray: '12 8 12',  
13     });  
14     counties.bindTooltip('freguesias');  
15     counties.addTo(map);  
16 </script>
```

# EXERCÍCIO TUTORIAL 10 – CARREGAR UM FICHEIRO GEOJSON PARA O MAPA

- Criar uma camada (vetor) através da informação existente num ficheiro GeoJson:
  - dicionário (key : value)
- freg\_porto\_polig.geojson

```
1 let freg_layer = {"type":"FeatureCollection","features":[{"type":"Feature","id":0,  
  "geometry":{"type":"Polygon","coordinates":[[[-8.6359383599551691,41.159758970376053],[-8.  
  6360441650789195,41.160008104196933],[-8.6360848774385719,41.160128043392639],[-8.  
  6361563424467462,41.160264505393776],[-8.6362491056637438,41.160398259456876],[-8.  
  6365361037060779,41.160674512828024],[-8.6369358610719811,41.161035176893343],[-8.  
  6371166659479641,41.16119807752311],[-8.6374146015983957,41.161470588119769],[-8.  
  6377473228308812,41.161757438912126],[-8.6377533198756868,41.16176245506567],[-8.  
  6380438049656405,41.162026804499668],[-8.6383864370421826,41.162316041134027],[-8.  
  6386130358005282,41.162451994705272],[-8.6390186015545858,41.162743744562803],[-8.
```

# EXERCÍCIO TUTORIAL 10 – CARREGAR UM FICHEIRO GEOJSON PARA O MAPA

- Criar uma camada (vetor) através da informação existente num ficheiro GeoJSON
- `<script src="..."></script>`
- `L.geoJSON(freg_layer, {  
    color: '#800000',  
    weight: 2,  
    dashArray: '12 8 12',  
});`
- `bindTooltip`

- Webmap\_app10.html

```
7 <script src="freg_porto_polig.geojson"></script>  
8 ...  
9  
10 var counties= L.geoJSON(freg_layer, {  
11     color: '#800000',  
12     weight: 2,  
13     dashArray: '12 8 12',  
14 });  
15 counties.bindTooltip('freguesias');  
16 counties.addTo(map);  
17 </script>
```

# EXERCÍCIO TUTORIAL 11 – CARREGAR VÁRIOS FICHEIROS GEOJSON PARA O MAPA

- Criar outras camadas (vetor) através da informação existente noutros ficheiros GeoJson:
  - paragens.geojson
  - ts\_areas\_conservacao\_pontos.geojson
- Webmap\_app11.html

# EXERCÍCIO TUTORIAL 11 – CARREGAR VÁRIOS FICHEIROS GEOJSON PARA O MAPA

- Criar opção de outras camadas (vetor) através da informação existente noutros ficheiros GeoJson:

```
<script src="paragens.geojson">  
</script>
```

```
<script  
src="ts_areas_conservacao_po  
ojson"></script>
```

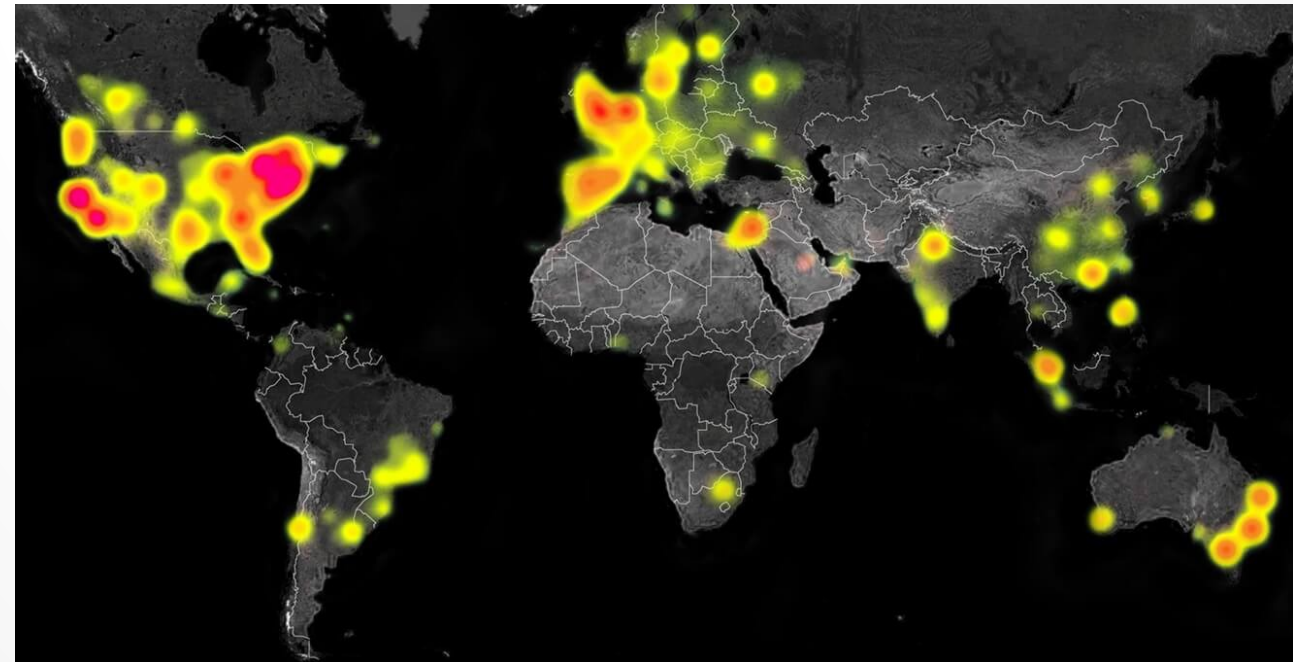
- Webmap\_app11.html

```
10 <script src="paragens.geojson"></script>  
11 <script src="ts_areas_conservacao_pontos.geojson"></script>  
12  
13 <script>  
14 var osm = L.tileLayer("http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {});  
15 var baserelief = L.tileLayer('https://tile.opentopomap.org/{z}/{x}/{y}.png', {});  
16  
17 var baseMaps = {  
18 |   "OpenStreetMap": osm,  
19 |   "Opentopomap": baserelief  
20 };  
21  
22 var paragens_pnpg = L.geoJSON(paragens_pnpg).addTo(map);  
23 var areas_pnpg = L.geoJSON(ts_areas_conservacao_pnpg).addTo(map);  
24  
25 var overlays = { //add any overlays here  
26 |   "Paragens": paragens_pnpg,  
27 |   "Areas Conservações": areas_pnpg  
28 };  
29  
30 L.control.layers(baseMaps,overlays, {position: 'topright',  
31 | | | selectedBaseLayerIndex: 1}).addTo(map);  
32 </script>
```

# EXERCÍCIO TUTORIAL 12 – MAPA DE CALOR

- Criar um mapa de calor
  - Biblioteca: leaflet-heat.js
- ```
<script src="leaflet-heat.js"> </script>
```

- Webmap\_app12.html



# EXERCÍCIO TUTORIAL 12 – MAPA DE CALOR

- Criar um mapa de calor
- Webmap\_app12.html

```
10     <style>
11         #map { width: 800px; height: 600px; }
12         body { font: 16px/1.4 "Helvetica Neue", Arial, sans-serif; }
13         .ghbtns { position: relative; top: 4px; margin-left: 5px; }
14         a { color: #0077ff; }
15     </style>
16 </head>
17 <body>
18     <div id="map"></div>
19     <script src="js/leaflet-heat.js"></script>
20     <script src="http://leaflet.github.io/Leaflet.markercluster/example/realworld.10000.js"></script>
21     <script>
22     var map = L.map('map').setView([-37.87, 175.475], 12);
23     var tiles = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
24         attribution: '© <a href="http://osm.org/copyright">OpenStreetMap</a> contributors',
25     }).addTo(map);
26     addressPoints = addressPoints.map(function (p) { return [p[0], p[1]]; });
27     var heat = L.heatLayer(addressPoints).addTo(map);
28 </script>
```



# EXERCÍCIO TUTORIAL 13 – MAPA BY API

- Criar um mapa a partir da API

- Webmap\_app12.html

```
<div id="map"></div>
<script>
$.getJSON("https://data.medicare.gov/resource/xubh-q36u.geojson?state=ND",
function(data) {
  var hospitals = L.geoJSON(data, {
    pointToLayer: function(feature, latlng) {
      return L.marker(latlng, {
        icon: L.icon({
          iconUrl: 'images/hospital.png',
          iconSize: [24, 24],
          iconAnchor: [12, 12],
          opacity: 0.5
        })
      }).bindTooltip(
        feature.properties.hospital_name
        + '<br>' + feature.properties.city
        + '<br>' + feature.properties.zip_code
      )
    }
  }).addTo(map)
})
</script>
```



WEBINAR  
BIG DATA -  
WEBMAPPING

ORADOR: RICARDO BAPTISTA

*ricardo.baptista@myistec.pt*

Obrigado!