

1. ArcMap

Inicialmente foi escolhido o ArcGis ArcMap para manipulação e edição dos mapas. O ArcGis oferece uma versão de testes (<http://desktop.arcgis.com/en/arcmap/>) gratuita por dois meses. Com esta versão é possível manipular os dados que geram os mapas, permitindo acrescentar, editar ou excluir valores da tabela de atributos.

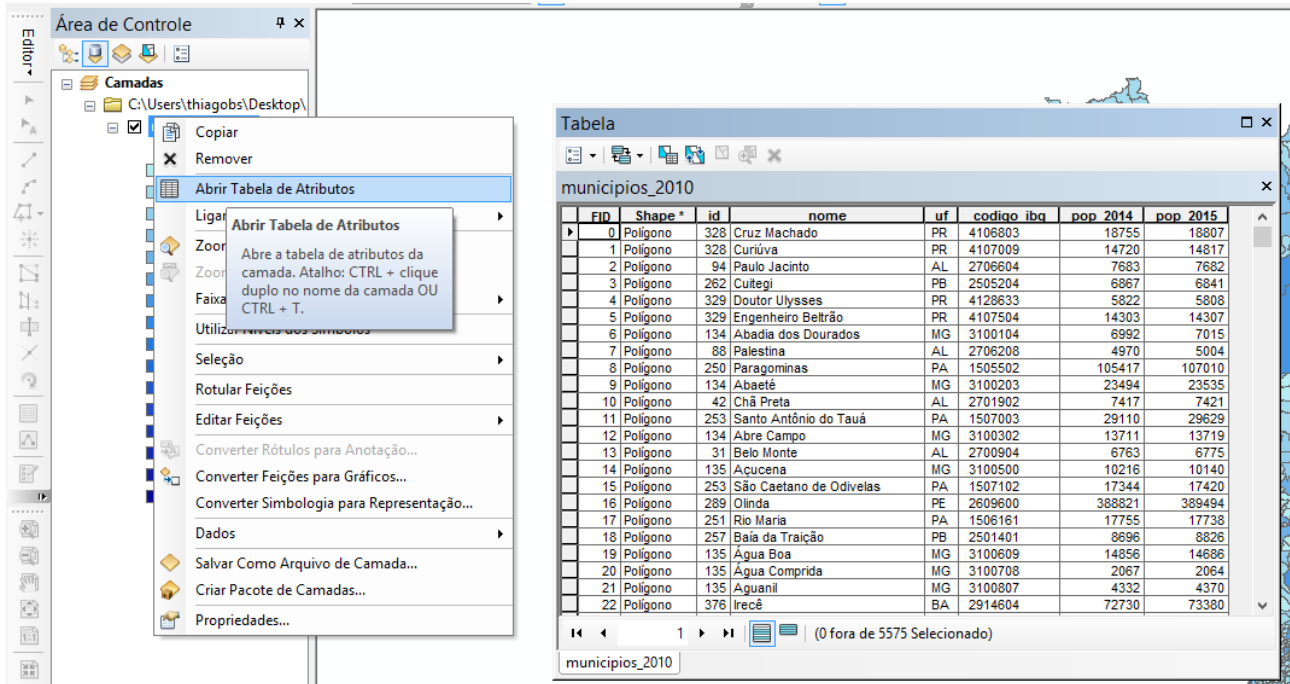


Figura 1 - Exemplo Tabela de Atributos ArcMap

Com a tabela de atributos completa com todos os dados necessários para criação dos mapas é preciso convertê-la em um arquivo no formato GeoJSON, que será utilizado posteriormente. Para isso, utilizamos o serviço online do ArcGis (<https://www.arcgis.com/home/signin.html?returnUrl=http%3A%2F%2Fdesktop.arcgis.com%2Fen%2Farcmap%2F>), que nos permite exportar o mapa criado com sua tabela de atributos para diversos formatos, incluindo o GeoJSON. Este serviço online também é disponibilizado de forma gratuita para testes por dois meses.

Para editar os valores da tabela de atributos, entre em *Editor* -> *Iniciar Edição*.

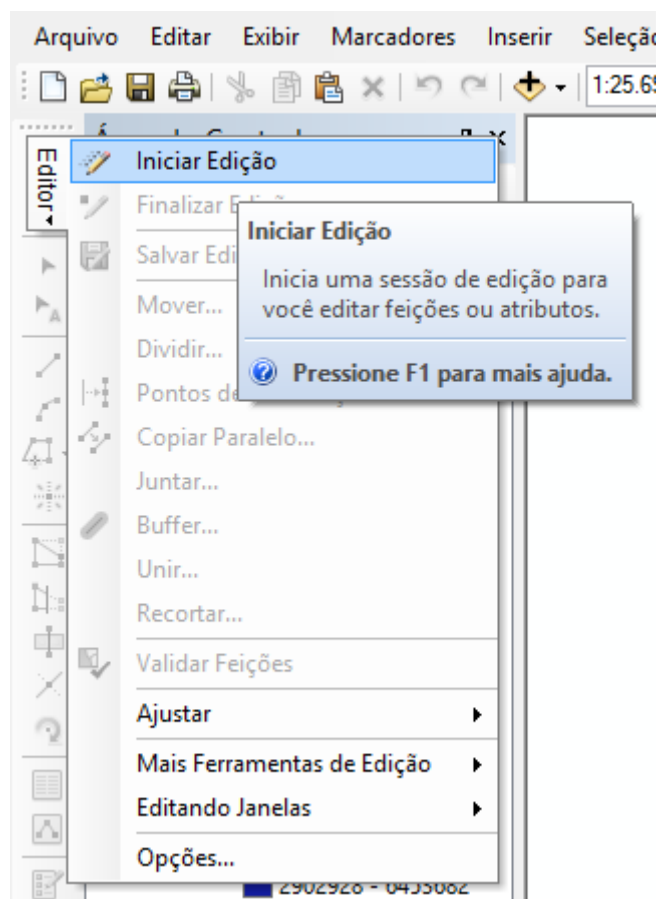


Figura 2 - Iniciar edição na tabela de atributos

A importação da tabela de atributos para o ArcGis Online é feita dentro do próprio software ArcMap, é preciso apenas fazer *login* com a conta do ArcGis Online no ArcMap.

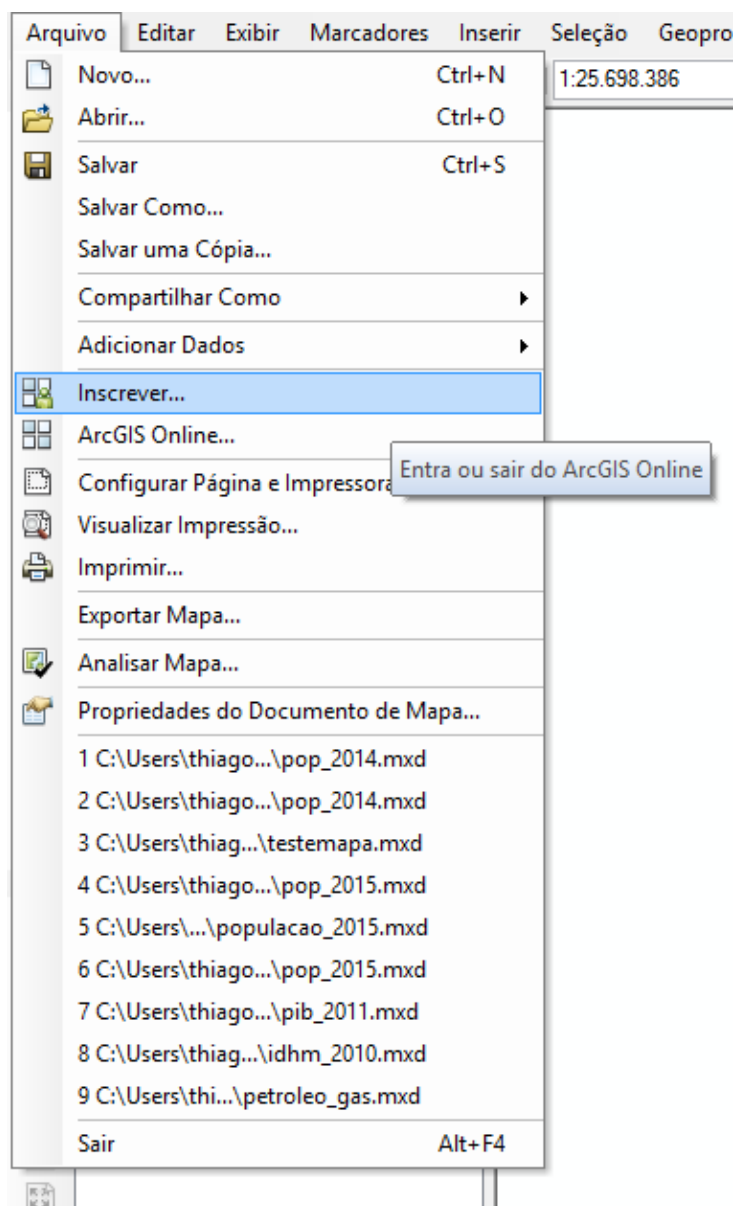


Figura 3 - Login no ArcGis Online pelo ArcMap

Com o login efetuado, basta compartilhar o mapa como um serviço que será publicado no ArcGis Online.

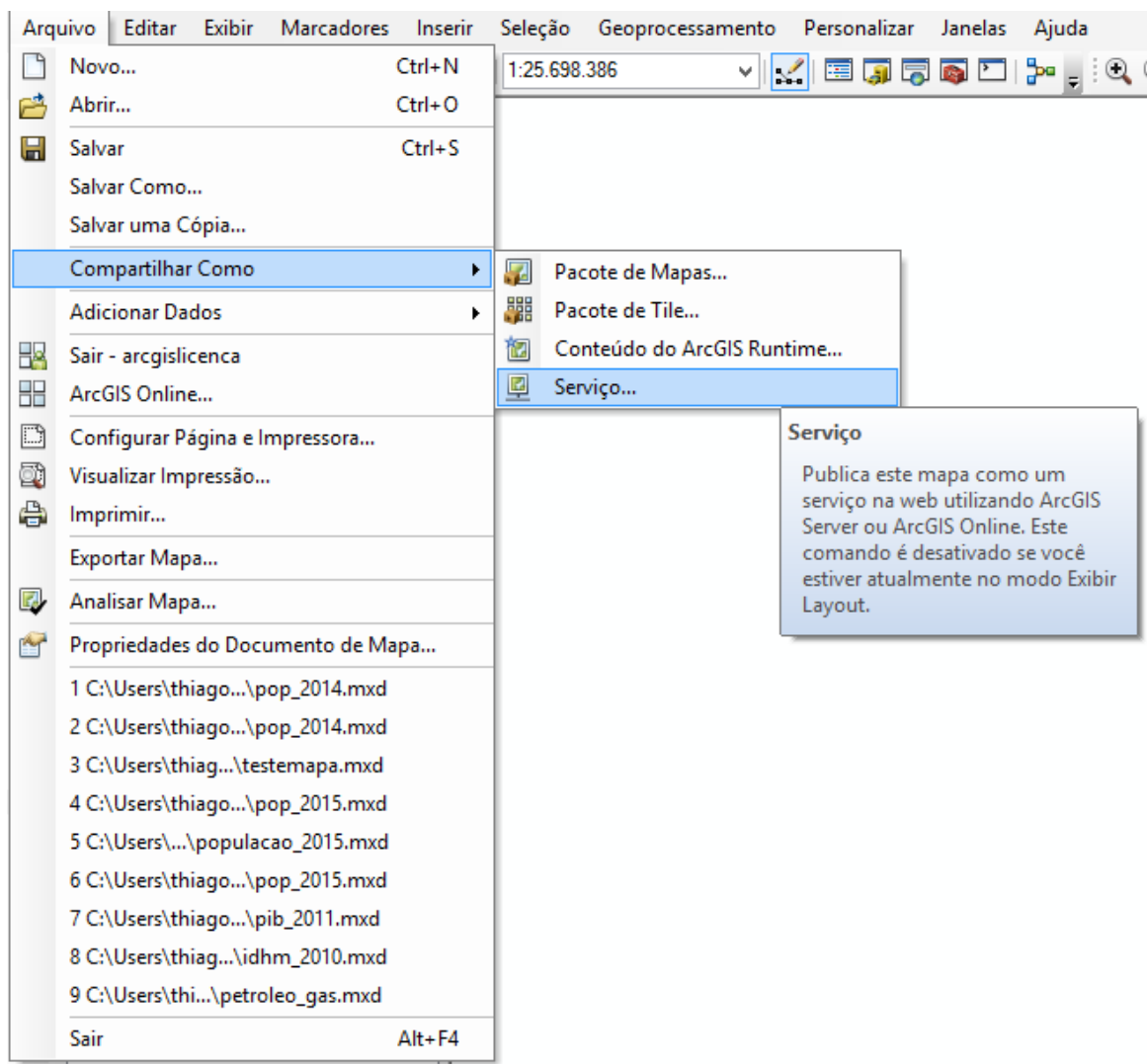


Figura 4 - Compartilhar mapa como serviço

Feito isso, selecionar *Publicar um serviço* e *Avançar*.

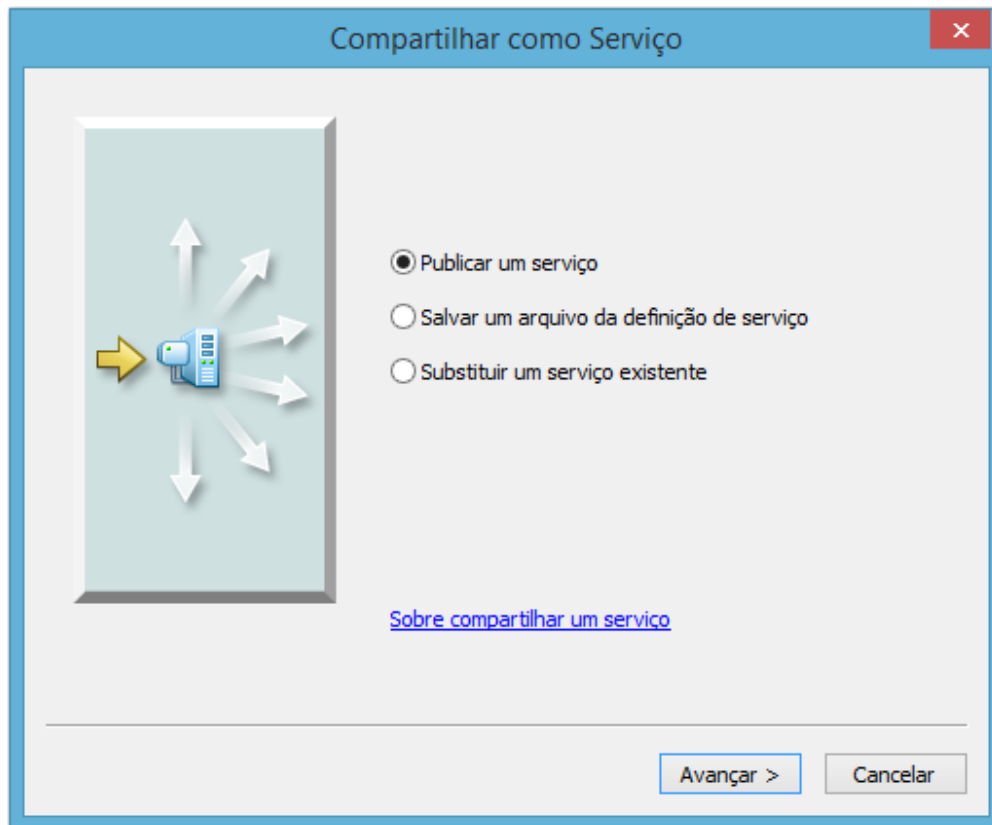


Figura 5 - Primeira etapa para publicar o mapa

Fornecer um nome para o serviço e selecionar uma conexão. A conexão este relacionada a conta criada no ArcGis Online.

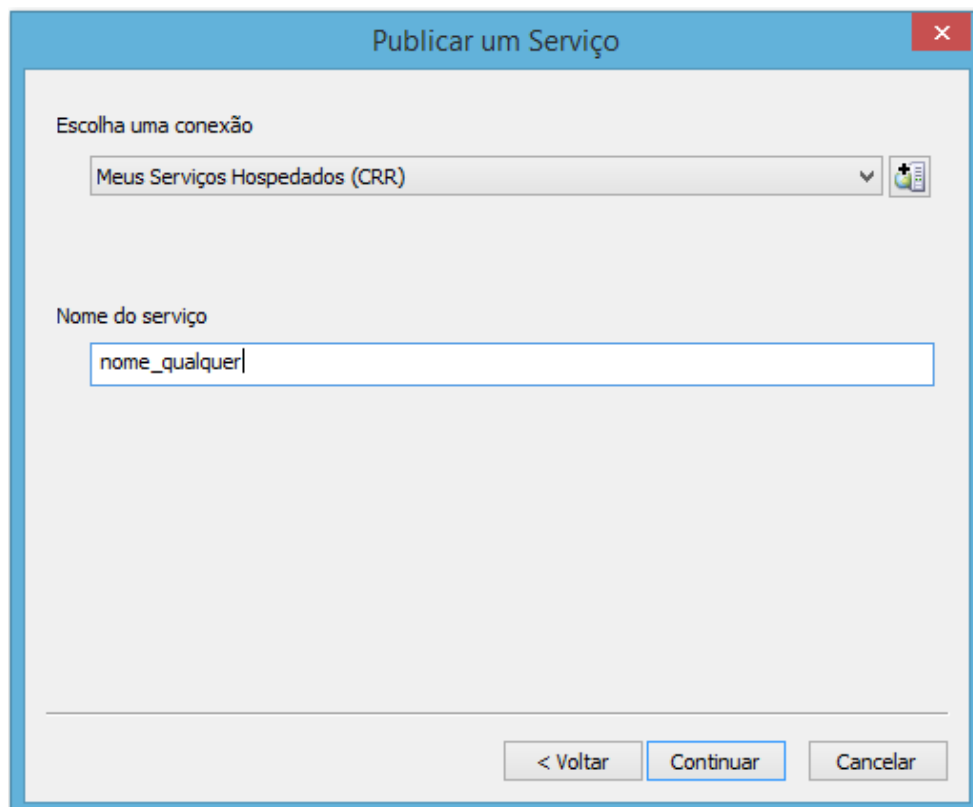


Figura 6 - Selecionar nome e conexão

Então configure os parâmetros e outras informações necessárias para a publicação.

Antes de publicar o serviço, é importante verificar se não possui nem um erro ou restrição no mapa (clique em *Análise*). Caso isso ocorra, é preciso corrigir todos os erros antes de publicar. Se tudo estiver correto, clique em *Publicar* e aguardar o mapa ser publicado no ArcGIS Online.

Editor de Serviço

Conexão: Meus Serviços Hospedados Nome do Serviço: nome_qualquer

Importar Analisar Visualizar Publicar

Parâmetros

Recursos

Feature Access

Descrição do Item

Compartilhamento

Parâmetros

Efeito Cascata

Efeito Cascata: Melhor

Efeito Cascata de Texto: Forçar

Propriedades

Número máximo de registros retornados pelo servidor: 10000

Avançado...

Figura 7 - Publicação do serviço

Após a publicação, o mapa poderá ser encontrado na aba *Meu Conteúdo* no ArcGIS Online.

Mapas Cena Grupos Meu Conteúdo Minha Organização

ArcGIS

do

+ Adicionar Item Criar Excluir Mover Alterar Proprietário

Título	Tipo	Modificado	Compartilhado
nome_qualquer	Service Definition	16/03/2017	Não Compartilhado
nome_qualquer	Feature Layer (hospedado)	16/03/2017	Todos

1 - 2 of 2 resultados

Figura 8 - Mapa publicado no ArcGIS Online

Ao entrar no mapa publicado, clicar na opção *Exportar Dados* -> *Exportar para GeoJSON*.

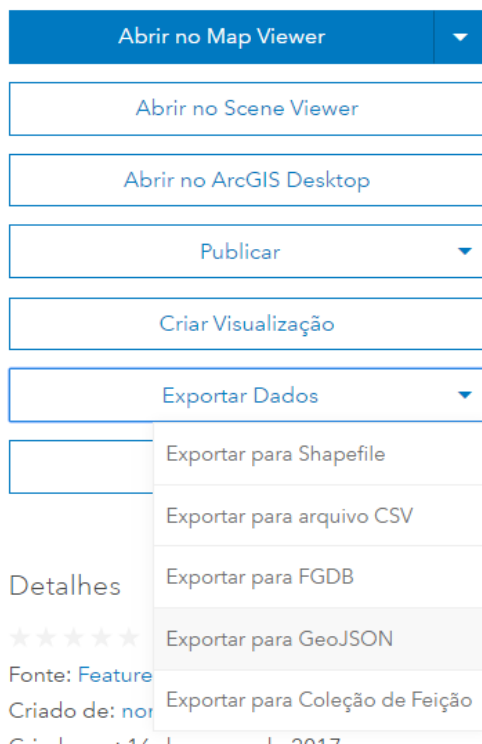


Figura 9 - Exportar para GeoJSON

Feito isso, preencha os dados necessários para a exportação e clique em *Exportar*.

A screenshot of a dialog box titled 'Exportar para GeoJSON' with a close button (X) in the top right corner. The dialog contains four input fields: 'Título:' with the value 'GeoJSON'; 'Tags:' with a tag 'GeoJSON x' and a link 'Adicionar tags'; 'Resumo:' with the value 'Exportando para GeoJson'; and 'Salvar na pasta:' with a dropdown menu showing 'arcgislicenca'. At the bottom right, there are two buttons: 'Exportar' (highlighted in blue) and 'Cancelar'.

Figura 10 - Preencher dados e exportar

Com o arquivo GeoJSON publicado é só fazer o download e salvá-lo. Este arquivo será muito importante para a criação do mapa online.

2. Banco de Dados

O banco de dados foi criado em MySQL a partir de uma tabela do Excel. Este banco de dados é utilizado para realizar as consultas das buscas realizadas pelo usuário no site.

	A	B	C	D	E	F	G	H	I
1	UF	COD_UF	COD_MUNICIP	COD_IBGE	NOME_MUNICIP	POP_2014	PIB_2011	IDHM_2010	IDHM_R
2	RO	11	15	1100015	ALTA FLORESTA D'OESTE	25652	349876793.00	0.641	0.657
3	RO	11	23	1100023	ARIQUEMES	102860	1569918349.00	0.702	0.716
4	RO	11	31	1100031	CABIXI	6424	117418847.00	0.65	0.65
5	RO	11	49	1100049	CACOAL	86556	1287814774.00	0.718	0.727
5565	GO	52	21858	5221858	VALPARAÍSO DE GOIÁS	150005	1155260881.00	0.746	0.733
5566	GO	52	21908	5221908	VARJÃO	3813	35865073.00	0.687	0.698
5567	GO	52	22005	5222005	VIANÓPOLIS	13343	194360646.00	0.712	0.71
5568	GO	52	22054	5222054	VICENTINÓPOLIS	8053	141712.42	0.684	0.707
5569	GO	52	22203	5222203	VILA BOA	5371	96377685.00	0.647	0.619
5570	GO	52	22302	5222302	VILA PROPÍCIO	5520	126267941.00	0.634	0.629
5571	DF	53	108	5300108	BRASÍLIA	2852372	164482129.00	0.824	0.863

Figura 11 - Trecho da tabela no Excel

Para fazer a exportação dos dados da tabela foi utilizado o plugin *MySQL for Excel* (<https://www.mysql.com/why-mysql/windows/excel/>). Este plugin permite exportar os dados do Excel para o MySQL sem que precisamos fazer isso manualmente.

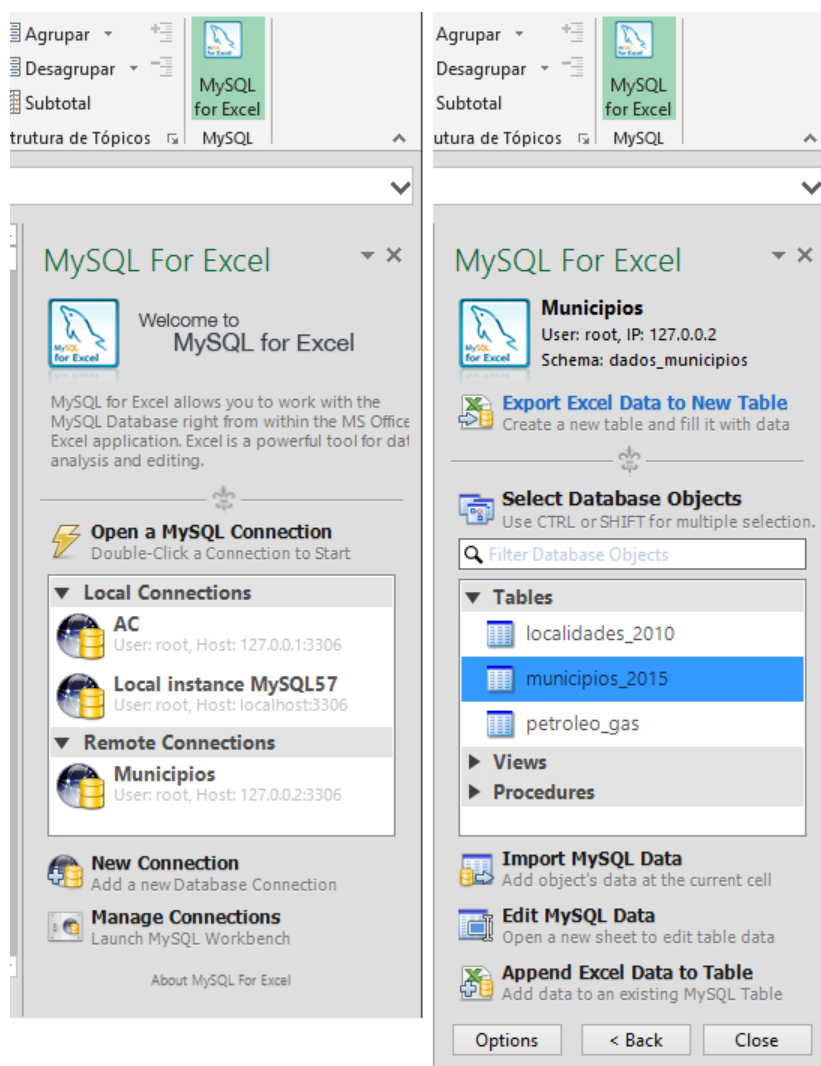
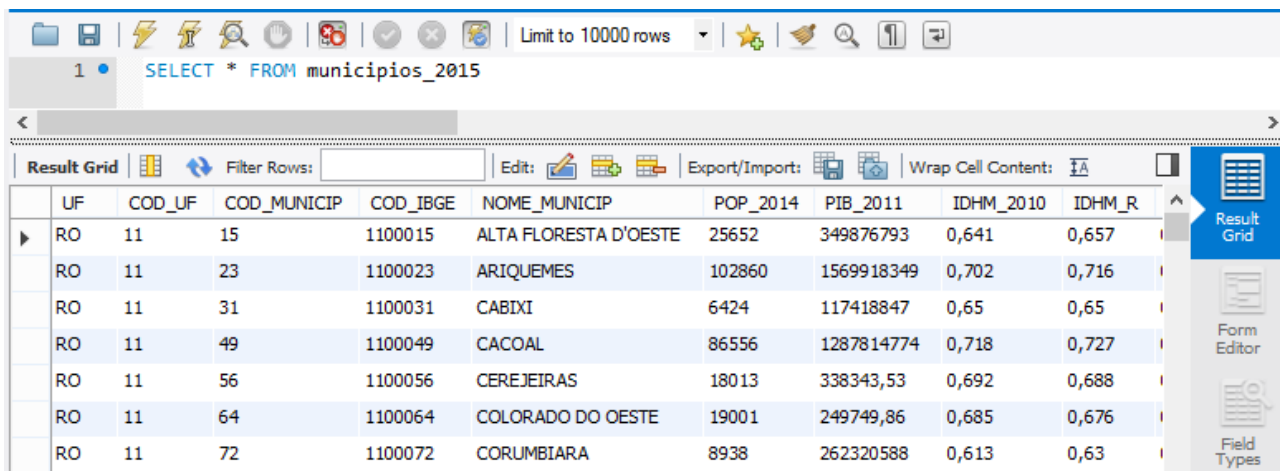


Figura 12 - Exemplo plugin MySQL for Excel

Com o banco criado, é possível realizar consulta e conectá-lo na aplicação web.



	UF	COD_UF	COD_MUNICIP	COD_IBGE	NOME_MUNICIP	POP_2014	PIB_2011	IDHM_2010	IDHM_R
▶	RO	11	15	1100015	ALTA FLORESTA D'OESTE	25652	349876793	0,641	0,657
	RO	11	23	1100023	ARIQUEMES	102860	1569918349	0,702	0,716
	RO	11	31	1100031	CABIXI	6424	117418847	0,65	0,65
	RO	11	49	1100049	CACOAL	86556	1287814774	0,718	0,727
	RO	11	56	1100056	CEREJEIRAS	18013	338343,53	0,692	0,688
	RO	11	64	1100064	COLORADO DO OESTE	19001	249749,86	0,685	0,676
	RO	11	72	1100072	CORUMBIARA	8938	262320588	0,613	0,63

Figura 13 - Select realizado no banco

Após a criação do banco, foi preciso conectá-lo a aplicação web. A conexão com o banco foi feita na linguagem PHP, utilizando a classe PDO (PHP Data Object) para isso.

```
1 <?php
2 // Parametro recebido do javascript
3 $coluna = $_POST['coluna']; // Onde pesquisar (coluna do banco)
4
5
6 // Estabelece conexão com o banco
7 try {
8     // PDO - PHP Data Object - Prepared Statements
9     $connect = new PDO('mysql:host=127.0.0.2;dbname=dados_municipios;charset=utf8mb4', 'user', 'user');
10 } catch(PDOException $ex) {
11     echo ($ex->getMessage());
12 }
13
14
15 // Consulta o banco
16 $sql = 'SELECT COD_IBGE, NOME_MUNICIP, UF, ' . $coluna . ' FROM municipios_2015'; // Query
17 $db = $connect->prepare($sql); // prepara a consulta
18 $db->execute(); // executa a consulta
19
20
21 // envia os resultados da consulta para o javascript que chamou
22 while ( $row = $db->fetch() ) {
23     echo $row['COD_IBGE'] . '#' . $row['NOME_MUNICIP'] . '#' . $row['UF'] . '#' . $row[ ($coluna) ] . '#';
24 }
25 ?>
```

Figura 14 - Código PHP para conexão com banco

Para chamar esta função no código JavaScript, foi utilizado Ajax com JQuery para poder acessar o PHP de forma assíncrona, sem a necessidade de recarregar a página a cada consulta do usuário.

```

691 // ***** Acessa o banco para retornar todos os resultados *****
692
693 // Variáveis globais
694
695 var flag = 1; // verificar se o usuário está realizando a mesma busca
696 // Isso evita que o usuário busque a mesma coisa várias vezes em sequência
697 // (evita acessar o bd e retornar o mesmo resultado várias vezes)
698
699 var content = ""; // conteúdo digitado pelo usuário para pesquisa
700
701 var _coluna = "IDHM_2010"; // Variável que será passada para o php
702 // Representa o nome da coluna no bd que será consultada
703
704 connect();
705
706 function connect(){
707
708     if( flag != 0 ){
709
710         flag = 0;
711
712         // Ajax para chamar php de forma assíncrona
713         $.ajax({
714             url: 'banco_todos.php', // função php que conecta com o banco
715             method: "post", // método usado para passar os parâmetros
716             data: {coluna: _coluna }, // parâmetros passados para o php
717             complete: function (response) {
718                 // Chama a função para construir a tabela passando os resultados
719                 fncMunicipios(response.responseText);
720             },
721             error: function () {
722                 // Caso ocorra algum erro na conexão
723                 console.log('Error');
724             }
725         });
726     }
727 }
728 // *****

```

Figura 15 - Exemplo Ajax para conexão com banco

Pesquisar

Por exemplo: São Paulo, MG, 2109056

Código	Município	Estado	IDHM
1100015	ALTA FLORESTA D'OESTE	RO	0.641
1100023	ARIQUEMES	RO	0.702
1100031	CABIXI	RO	0.65
1100049	CACOAL	RO	0.718
1100056	CEREJEIRAS	RO	0.692
1100064	COLORADO DO OESTE	RO	0.685
1100072	CORUMBIARA	RO	0.613
1100080	COSTA MARQUES	RO	0.611
1100098	ESPIGÃO D'OESTE	RO	0.672
1100106	GUAJARÁ-MIRIM	RO	0.657
1100114	JARU	RO	0.689
1100122	JI-PARANÁ	RO	0.714
1100130	MACHADINHO D'OESTE	RO	0.596
1100148	NOVA BRASILÂNDIA D'OESTE	RO	0.643
1100155	OURO PRETO DO OESTE	RO	0.682
1100189	PIMENTA BUENO	RO	0.71
1100205	PORTO VELHO	RO	0.736

Total de resultados: 5570

Dados de 2010

Figura 16 - Exemplo tabela com resultados

3. Leaflet

O Leaflet (<http://leafletjs.com/>) é uma biblioteca JavaScript para a elaboração de mapas interativos online, que nos fornece uma base para criação dos nossos mapas.

Primeiro é preciso importar os arquivos CSS e JavaScript do Leaflet para podermos usá-los.

```
25      <!-- JavaScript Leaflet -->
26      <script src="leaflet.js" type="text/javascript"></script>
27      <!-- CSS Leaflet -->
28      <link rel="stylesheet" href="leaflet.css" type="text/css" >
29
```

Figura 17 - Importação do Leaflet

Para criar este mapa é bem simples, basta ter uma variável que receberá o mapa do Leaflet e tudo o que aplicarmos a ele utilizando o método `.addTo(map)`.

O *basemap* refere-se ao estilo do mapa utilizado no Leaflet. Vários estilos diferentes podem ser encontrados em <https://leaflet-extras.github.io/leaflet-providers/preview/> e podem ser trocados caso haja necessidade.

Foi utilizado o tutorial *Interactive Choropleth Map* (<http://leafletjs.com/examples/choropleth/>) para a criação das outras funcionalidades do mapa.

```
126      // cria um novo mapa
127      map = L.map('map', {fullscreenControl: true }).setView([-15, -55], 4);
128
129      // Seleciona o basemap
130      L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
131          attribution: '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a> | &copy; <a
132              target="_blank" href="http://www.inatel.br/crr/">CRR</a> Inatel',
133              minZoom: 4, maxZoom: 13, unloadInvisibleTiles: true, updateWhenIdle: true
134      }).addTo(map);
```

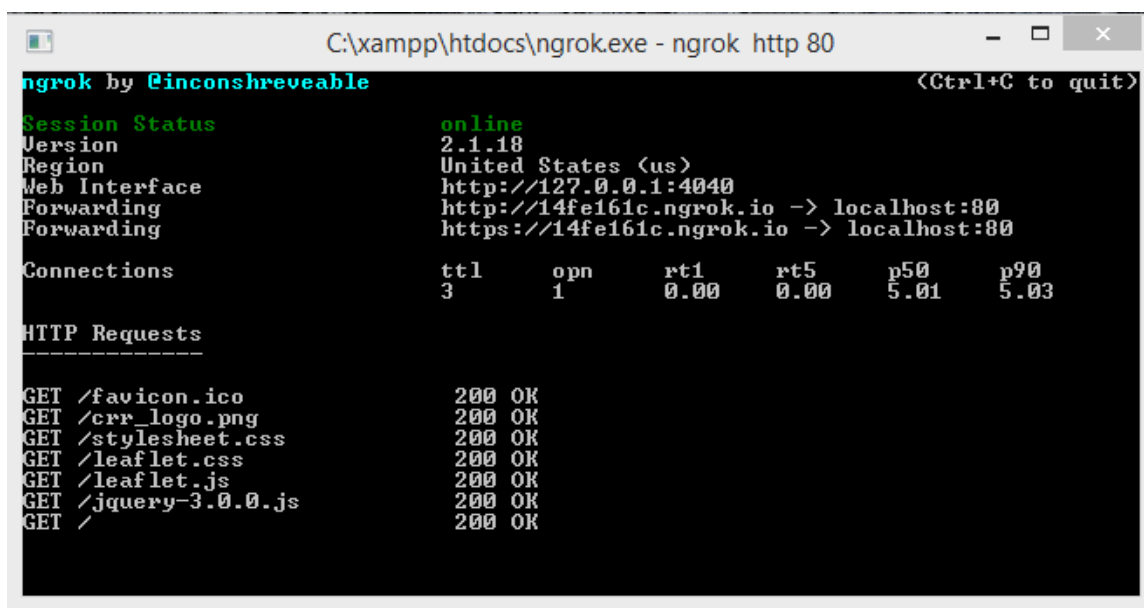
Figura 18 - Criação mapa Leaflet

A documentação completa do Leaflet pode ser encontrada em <http://leafletjs.com/reference-1.0.3.html> para consulta e entendimento das funcionalidades do Leaflet.

4. Testando a aplicação

Para realizar alguns testes na aplicação foi utilizado o Ngrok (<https://ngrok.com/>), que cria uma URL para um servidor local no computador, permitindo acessar remotamente a aplicação e testá-la. O comando do Ngrok para criar essa URL é `ngrok http 80`.

Para criar um servidor PHP localmente no computador, foi utilizado a ferramenta Xampp (https://www.apachefriends.org/pt_br/index.html), executando o módulo do servidor Apache na porta 80.



```
C:\xampp\htdocs\ngrok.exe - ngrok http 80
ngrok by @inconshreveable <Ctrl+C to quit>

Session Status      online
Version             2.1.18
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://14fe161c.ngrok.io -> localhost:80
                    https://14fe161c.ngrok.io -> localhost:80

Connections         ttl    opn    rt1    rt5    p50    p90
                   3      1      0.00   0.00   5.01   5.03

HTTP Requests
-----
GET /favicon.ico           200 OK
GET /css/logo.png         200 OK
GET /stylesheet.css        200 OK
GET /leaflet.css           200 OK
GET /leaflet.js            200 OK
GET /jquery-3.0.0.js       200 OK
GET /                      200 OK
```

Figura 19 - URL criada no Ngrok

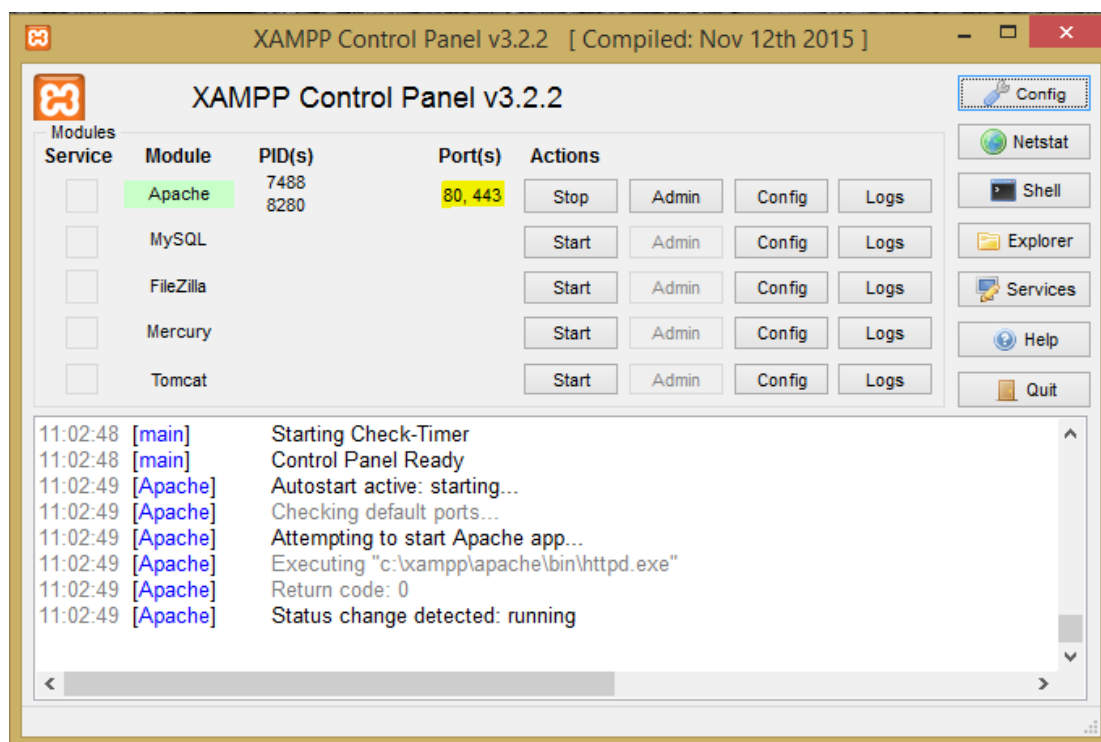


Figura 20 - Xampp executando Apache

5. Observações

Todos os códigos feitos e os relatórios estão disponíveis em uma pasta compartilhada no Dropbox

(https://www.dropbox.com/sh/8jrpkywzqd2pe/AACT3sET_fmoRpqswFqjwn3Ka?dl=0) e no Github (<https://github.com/tbsouza/CRR>).