

4 – Shapefiles próprios

Aula 06

Sumário

Shapefiles próprios

ArcGis developers

Hosted layers

Aplicação com hosted layers

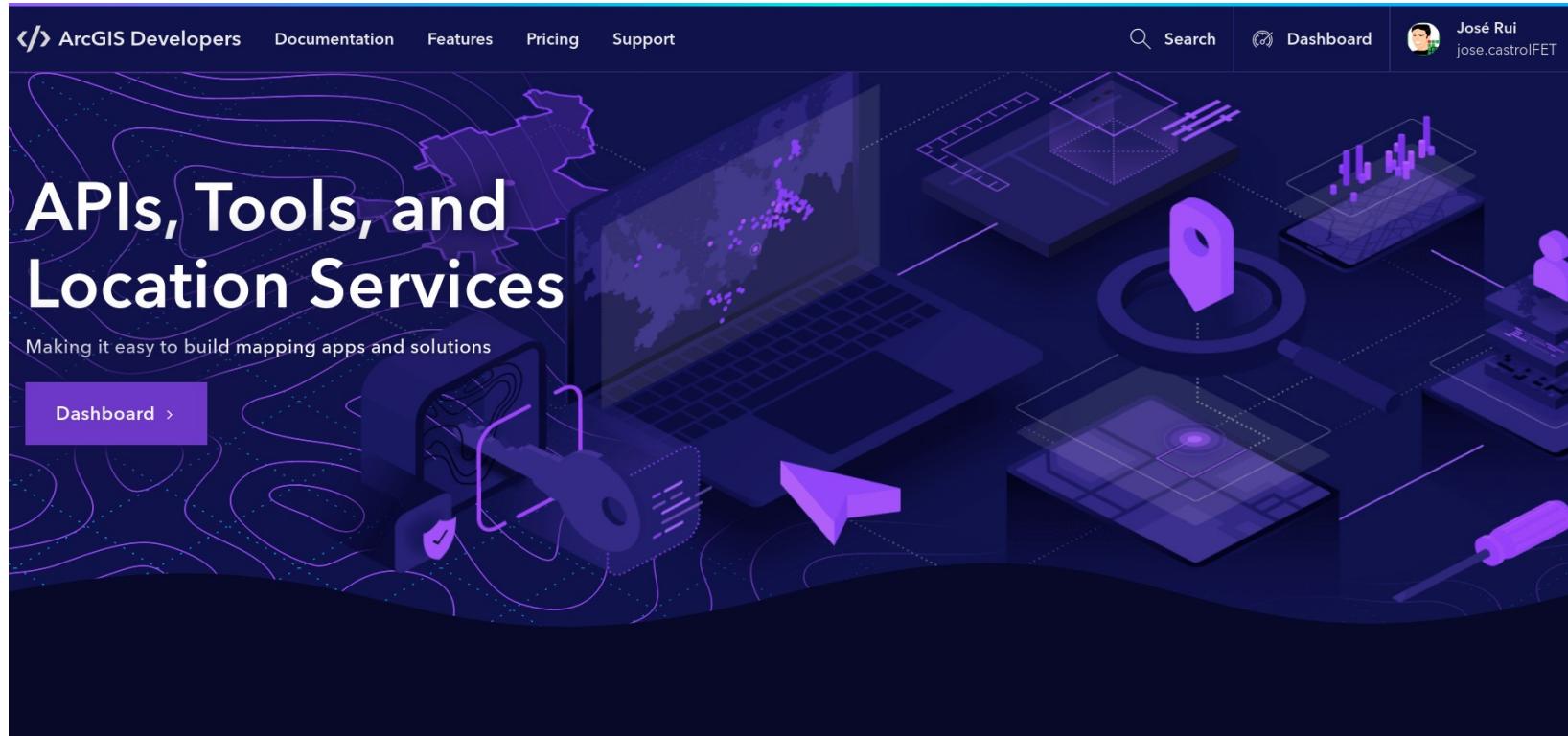
- Layout
- MainActivity

Acessando dados da tabela de atributos

- callout



ArcGIS Developers



ArcGIS Developers

The screenshot shows the ArcGIS Developers website's "Resources & Support" page. At the top, there is a dark blue header with the "ArcGIS Developers" logo, navigation links for Documentation, Features, Pricing, and Support, a search bar, and a sign-in button. Below the header is a large purple banner with the title "Resources & Support" and a subtitle: "Get help, technical support, and discover resources to guide you to success as a developer with ArcGIS."

API & SDK Support

Select your product for API and SDK-specific support resources. Read articles and guides, get in touch with fellow developers, access free training, share code snippets, and more.



ArcGIS API for
JavaScript



ArcGIS Runtime
SDK for Android



ArcGIS Runtime
SDK for iOS



ArcGIS Runtime
SDK for Java



ArcGIS Runtime
SDK for .NET



ArcGIS Runtime
SDK for Qt



ArcGIS API for
Python



ArcGIS Experience
Builder (Developer
Edition)



ArcGIS Web
AppBuilder
(Developer
Edition)



ArcGIS Arcade



esri

Cadastro ArcGIS Developers



ArcGIS Developers Documentation Features Pricing Support Search Dashboard José Rui jose.castrolFET

Welcome, José

Dashboard API keys OAuth 2.0 Layers Downloads Usage Account

Get Started

Guide **Mapping APIs and location services**

Tutorial **Create a mapping app**

Tutorial **Access geocoding and routing services**

ArcGIS Runtime API for Android

v100.10.0 · January 27, 2021

Doc and tutorials

Sample code

Blog - 27 de janeiro de 2021

Welcome to ArcGIS Runtime 100.10

API keys (1)

Manage

Use API keys to access location services and premium content in your applications. Learn more about using API keys to access services in the ArcGIS Developer Guide.

Default API Key | Modified 26 minutes ago

View Usage Edit API Key

Services Allowed Referrers

Account

Usage Billing Account Settings

Subscription

Type	?	Essentials
Renewal Date	?	28/03/2021

Pay as you go

Cadastro ArcGIS Developers

The screenshot shows the ArcGIS Developers dashboard. At the top, there is a navigation bar with links for ArcGIS Developers, Documentation, Features, Pricing, Support, Search, Dashboard, and a user profile for José Rui (jose.castroFET). Below the navigation bar, a welcome message "Welcome, José" is displayed, followed by a "Dashboard" tab which is currently selected. The main content area features a "Get Started" section with a yellow callout containing the text: "Cada um tem o seu. É usado como parâmetro lá no app". To the right of this, there is a section for "ArcGIS Runtime API for Android" (v100.10.0 · January 27, 2021), which includes links for "Doc and tutorials", "Sample code", and "Blog". In the bottom left, there is a "API keys" section with a sub-section for "Default API Key" (Modified 26 minutes ago), which has a yellow arrow pointing to it. This section also includes "View Usage" and "Edit API Key" buttons. In the bottom right, there is an "Account" section with tabs for Usage, Billing, and Account Settings, and a "Subscription" section showing "Type: Essentials" and "Renewal Date: 28/03/2021". A "Pay as you go" button is also present.

Cada um tem o seu. É usado como parâmetro lá no app

API keys (1)

Default API Key | Modified 26 minutes ago

View Usage Edit API Key

Allowed Referrers

Account

Usage Billing Account Settings

Subscription

Type: Essentials

Renewal Date: 28/03/2021

Pay as you go

Importar ShapeFile para **hosted layers**

Abrindo shapefiles

Importar ShapeFile para **hosted layers**

- Parks and Open Space (Shapefile)
- **.zip** de todos os arquivos que o compõe

Documentation / Mapping APIs and location services

The screenshot shows a documentation page for "Import the Parks and Open Spaces Shapefile". The left sidebar has sections for Data management, Hosted feature layers, Hosted vector tile layers, Hosted image tile layers, and Tutorials. Under APIs, there are links for Add a feature layer, Add a vector tile layer, Add an image tile layer, Query a feature layer (SQL), Query a feature layer (spatial), Style a feature layer, Display a pop-up, Edit feature data, Tools, Import data as a feature layer, Create a new feature layer, Publish a vector tile layer, and Access feature layer data. The main content area title is "Import the Parks and Open Spaces Shapefile". It describes the process of importing polygon data with attributes to create a new feature layer. Step 1: Click Layers > Import data to upload your next file. Step 2: Upload the Parks and Open Space.zip file by either dragging and dropping or selecting a file. Step 3: Set details in the Item Details pane, including Title: Parks and Open Space, Tags: LA Parks, and Description: Parks and open spaces in Mahou Riviera. Step 4: At the bottom, click Create layer to create the new Parks and Open Space feature layer and feature service. Step 5: In the item page, click the Settings tab. Step 6: Under Layer access (Sharing), ensure that People with access is set to Public (authentication not required). The right sidebar lists Import data as a feature layer steps: Prerequisites, Steps, Download the data, Import the Trailheads CSV file, Find the Trailheads ID and URL, Find the Trailheads ID and URL, Import the Trails GeoJSON file, Find the Trails ID and URL, Find the Trails ID and URL, Import the Parks and Open Spaces Shapefile, Find the Parks and Open Spaces ID and URL, Find the Parks and Open Spaces ID and URL, and What's next? Estimated time is 10 minutes.

Import the Parks and Open Spaces Shapefile

The Parks and Open Spaces Shapefile contains polygon data with attributes. Importing the file will create a new polygon feature layer in a feature service.

Developer dashboard ArcGIS Online

- 1 Click **Layers** > **Import data** to upload your next file.
- 2 Upload the *Parks and Open Space.zip* file by either:
 - Dragging and dropping box the file on the **Drop a file here** area.
 - Or, clicking the **Select file** button to browse to the file.
 - Click **Shapefile** > **Upload file**.
- 3 Once the file is uploaded, set the following details in the **Item Details** pane:
 - **Title:** Parks and Open Space
 - **Tags:** LA Parks The Tags property cannot be left blank.
 - **Description:** Parks and open spaces in Mahou Riviera.
- 4 At the bottom, click **Create layer** to create the new Parks and Open Space feature layer and feature service.
- 5 In the item page, click the **Settings** tab.
- 6 Under **Layer access (Sharing)**, ensure that **People with access** is set to **Public (authentication not required)**.

Import data as a feature layer

Prerequisites

Steps

Download the data

Import the Trailheads CSV file

Find the Trailheads ID and URL

Find the Trailheads ID and URL

Import the Trails GeoJSON file

Find the Trails ID and URL

Find the Trails ID and URL

Import the Parks and Open Spaces Shapefile

Find the Parks and Open Spaces ID and URL

Find the Parks and Open Spaces ID and URL

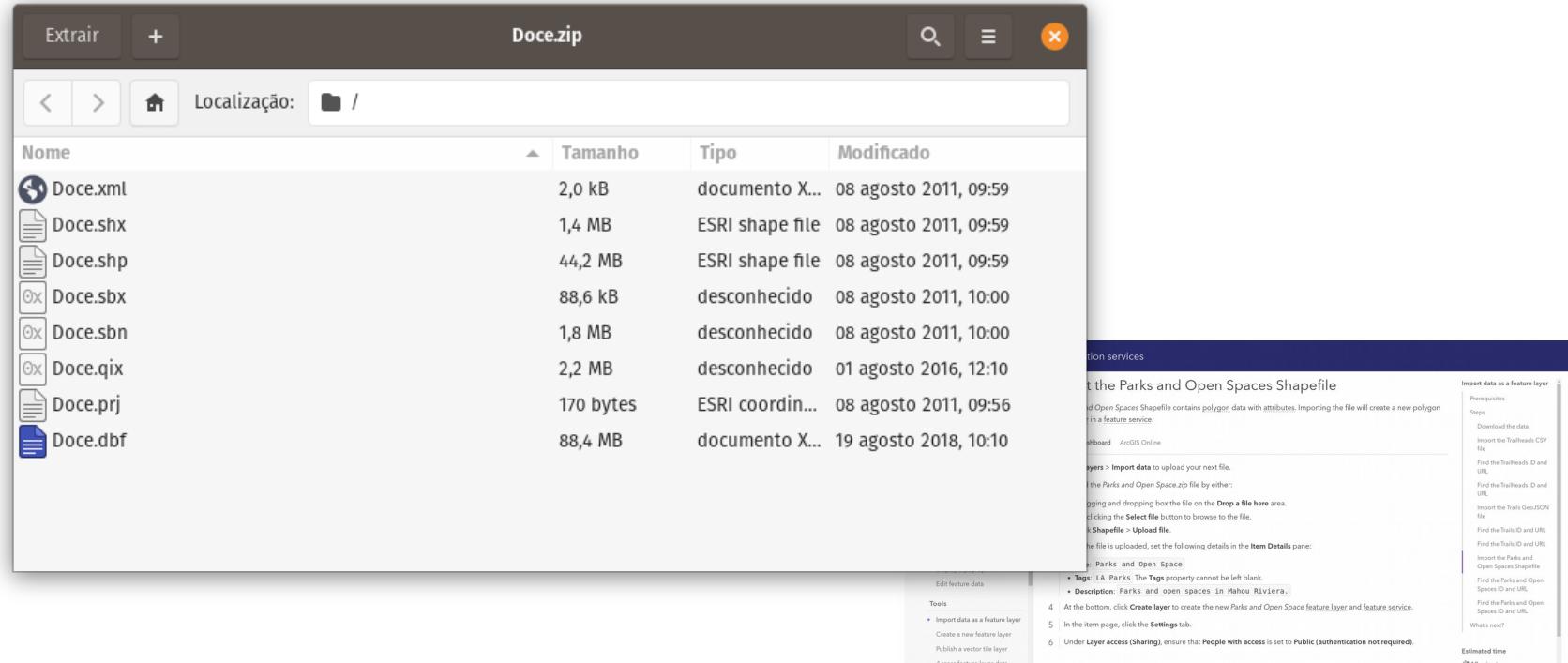
What's next?

Estimated time

~10 minutes

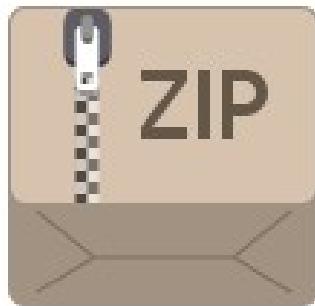
Importar ShapeFile para hosted layers

- Parks and Open Space (Shapefile)
 - **.zip** de todos os arquivos que o compõe

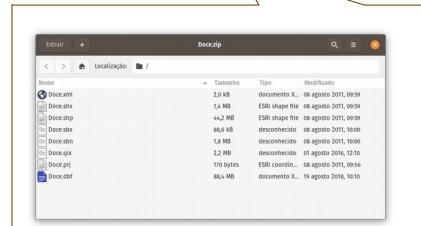


Importar ShapeFile para hosted layers

- Parks and Open Space (Shapefile)
- **.zip** de todos os arquivos que o compõe



Doce.zip



Documentation / Mapping APIs and location services

Import the Parks and Open Spaces Shapefile

The Parks and Open Spaces Shapefile contains polygon data with attributes. Importing the file will create a new polygon feature layer in a feature service.

Developer dashboard ArcGIS Online

- 1 Click **Layers > Import data** to upload your next file.
- 2 Upload the Parks and Open Space.zip file by either:
 - Dragging and dropping the box on the **Drop a file here** area.
 - Or, clicking the **Select file** button to browse to the file.
 - Click **Shapefile > Upload**.
- 3 Once the file is uploaded, set the following details in the **Item Details** pane:
 - **Title:** Parks and Open Space
 - **Tags:** LA Parks The Tags property cannot be left blank.
 - **Description:** Parks and open spaces in Mahou Riviera.
- 4 At the bottom, click **Create layer** to create the new Parks and Open Space feature layer and feature service.
- 5 In the item page, click the **Settings** tab.
- 6 Under **Layer access (Sharing)**, ensure that **People with access** is set to **Public (authentication not required)**.

Import data as a feature layer
Prerequisites
Steps
Download the data
Find the Parks and Open Spaces Shapefile
Import the Parks and Open Spaces ID and URL
Import the Trailsheads ID and URL
Import the Trailsheads CSV file
Find the Trails ID and URL
Find the Trails ID and URL
Import the Trails CSV file
Find the Parks and Open Spaces ID and URL
Find the Parks and Open Spaces ID and URL
What's next?

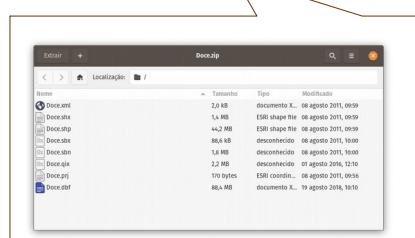
Estimated time 20 min

Importar ShapeFile para hosted layers

- Parks and Open Space (Shapefile)
 - **.zip** de todos os arquivos que o compõe



Doce.zip



Documentation / Mapping APIs and location services

Import the Parks and Open Spaces Shapefile

The Parks and Open Spaces Shapefile contains polygon data with attributes. Importing the file will create a new polygon feature layer in a feature service.

Developer dashboard ArcGIS Online

1 Click **Layers > Import data** to upload your next file.

2 Upload the Parks and Open Space.zip file by either:

- Dragging and dropping box the file on the **Drop a file here** area.
- Or, clicking the **Select File** button to browse to the file.
- Click **Shapefile > Upload file**

3 Once the file is uploaded, set the following details in the **Item Details** pane:

- **Title:** Parks and Open Space
- **Tags:** LA Parks. The Tags property cannot be left blank.
- **Description:** Parks and open spaces in Mahou Riviera.

4 At the bottom, click **Create layer** to create the new Parks and Open Space feature layer and feature service.

5 In the item page, click the **Settings** tab.

6 Under **Layer access (Sharing)**, ensure that **People with access** is set to **Public** (authentication not required).

Import data as a feature layer
Preerequisites
Steps
Download the data
Import the Trailsheads CSV file
Find the Trailsheads ID or URL
Find the Trailsheads ID or URL
Import the Trails GeoJSON file
Find the Trails ID or URL
Find the Trails ID or URL
Import Parks and Open Spaces Shapefile
Find the Parks and Open Spaces ID or URL
Find the Parks and Open Spaces ID or URL
What's next?

Shapefile carregado

- Uma vez carregado, disponível para acesso (web, mobile)
 - Item ID e Layer URL: serão passados como parâmetros em nossas aplicações.

Dashboard / Layers

Feature Layer (hosted)
Private (Authentication required)

Item ID: [Copy](#)

Created 25 de fevereiro de 2021
Modified 25 de fevereiro de 2021

Layer URL: <https://services3.arcgis.com/HVJannk9zTiwmsU1/arcgis/rest/services/Doce/FeatureServer> [Copy](#)

Overview Usage Settings

Goiânia
Uberlândia
Ribeirão Preto
Belo Horizonte
Seguro
Vitória
Campos dos Goytacazes

Open in [Map](#)

Esri, HERE, Garmin, FAO, NOAA, USGS Powered by Esri

Access and authentication

+ Create new API key [Edit sharing settings](#)

Access to this layer can be public or private and is managed through sharing settings. You can authenticate with an API key (read only) or an ArcGIS identity. Learn more about managing access to layers in the [security and authentication guide](#).

Dashboard API keys OAuth 2.0 Layers Downloads Usage Account

Create your first layer
Create a feature layer to store geographic data in ArcGIS Platform

Access and display your layer data
Display your feature layer data on a map

Manage your layer data
Set the properties and capabilities for a hosted feature layer.

Developer Guide
Mapping APIs and location services
Learn how to work with maps, scenes, and layers
Maps
Scenes
Data

Shapefile carregado

- Uma vez carregado, disponível para acesso (web, mobile)
 - Item ID e Layer URL: serão passados como parâmetros em nossas aplicações.

The screenshot shows the ArcGIS Platform dashboard with a loaded feature layer. The map displays the Bacia do Doce region in purple, with major cities like Belo Horizonte, Uberlândia, and Vassouras labeled. A yellow callout box highlights the layer's details: "Carregada a Bacia do Doce" (Loaded the Bacia do Doce), "Criada e editada no Qgis" (Created and edited in Qgis), and "Atenção para os sistema de coord. e proj." (Attention to coordinate system and projection). Another yellow box says "Enjoy it!".

Dashboard / Layers

Dashboard API keys OAuth 2.0 Layers Downloads Usage Account

Item ID: [Copy](#)

Created 25 de fevereiro de 2021
Modified 25 de fevereiro de 2021

Layer URL: <https://services3.arcgis.com/HVJannk9zTiwmsU1/arcgis/rest/services/Doce/FeatureServer> [Copy](#)

Overview Usage Settings

Goiânia Seguro

Uberlândia

Belo Horizonte

Vassouras

Ribeirão Preto

Campos dos Goytacazes

hosted feature layer.

Access and authentication

+ Create new API key [Edit sharing settings](#)

Access to this layer can be public or private and is managed through sharing settings. You can authenticate with an API key (read only) or an ArcGIS identity. Learn more about managing access to layers in the [security and authentication guide](#).

Developer Guide

Mapping APIs and location services

Learn how to work with maps, scenes, and layers

Maps

Scenes

Data

Carregada a Bacia do Doce

- Criada e editada no Qgis
- Atenção para os sistema de coord. e proj.

Enjoy it !

Usando em nossos Apps

Layout

- Adicionar uma **MapView**
 - Que carregará o mapa semelhante ao que fizemos com o **OPEN_STREET_MAP**

Layout

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Mapa de regionalização de vazão"/>
        <com.esri.arcgisruntime.mapping.view.MapView
            android:id="@+id/mapView"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```



Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3  
4  
5     @Override  
6     protected void onCreate(Bundle savedInstanceState) {  
7         super.onCreate(savedInstanceState);  
8         setContentView(R.layout.activity_main);  
9         // create a new map to display in the map view with a streets basemap  
10        mMapView = findViewById(R.id.mapView);  
11    }
```

Classe *MainActivity*

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     @Override  
6     protected void onCreate(Bundle savedInstanceState) {  
7         super.onCreate(savedInstanceState);  
8         setContentView(R.layout.activity_main);  
9         // create a new map to display in the map view with a streets basemap  
10        mMapView = findViewById(R.id.mapView);  
11    }
```

Carregada a Bacia do Doce

- Criar atributo **FeatureLayer**
- Link com o servidor **ArcGis Developers**

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps and other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

Atributo FeatureLayer

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps and other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

Precisamos da nossa chave única

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps and other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

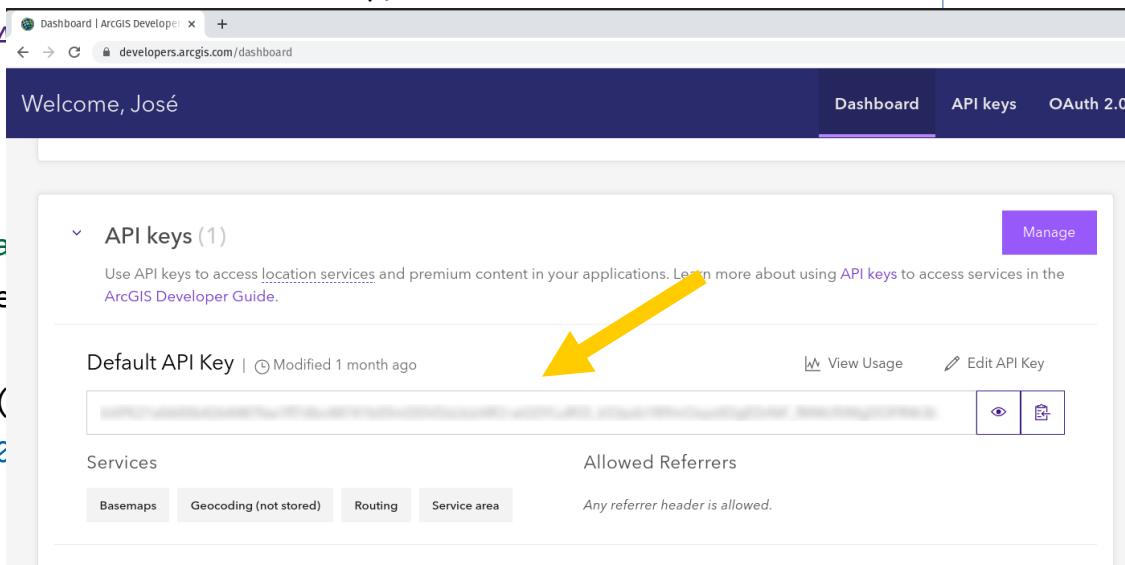
Precisamos da nossa chave única

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapview;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps and other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ArcGISWorldTopographic);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable =  
12            // create a feature layer to display the shapefile  
13            this.shapefileFeatureLayer = new FeatureLayer(  
14                // add the feature layer to the map  
15                this.mMapView.getMap().getOperationalLayers();  
16                this.mMapView.setViewpoint(new Viewpoint(-26  
17            })
```

Precisamos da nossa chave única



Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapview;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps and other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

Escolhemos o tipo do mapa

- Neste caso escolhi um mapa topográfico



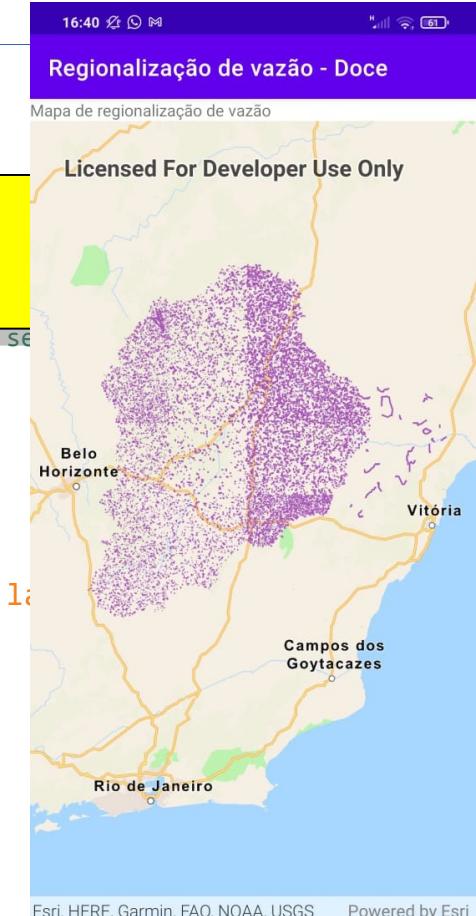
Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapview;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps from other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

Escolhemos o tipo do mapa

- Se fosse o mapa **Open_street_view** ficaria assim



Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps and other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

Precisamos informar qual é o shapeFile (feição)

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps and other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

Precisamos informar qual é o shapeFile (feição)

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

Para isso usar a classe
ServiceFeatureTable

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to acc  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

E é aqui que informamos o link
do nosso shapefile

(Sua conta pode ter vários. Quantos? Vai depender do seu plano R\$)

<https://services3.arcgis.com/HVJannk9zTiwmsU1/arcgis/rest/services/Doce/FeatureServer/0>

Feature Layer (hosted)
Private (Authentication required)
Item ID: d41ccc32cc8f471eacb5dc9b42eb3439

Created 25 de fevereiro de 2021
Modified 25 de fevereiro de 2021

Layer URL: <https://services3.arcgis.com/HVJannk9zTiwmsU1/arcgis/rest/services/Doce/FeatureServer/0>

Overview Usage Settings



Esri, HERE, Garmin, FAO, NOAA, USGS

```
7 ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");
8 ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);
9 this.mMapView.setMap(map);
10 // load the shapefile with a local path
11 ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");
12 // create a feature layer to display the shapefile
13 this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);
14 // add the feature layer to the map
15 this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);
16 this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));
17 }
```

E é aqui que informamos o link
do nosso shapefile

(Sua conta pode ter vários. Quantos? Vai depender do seu plano R\$)

<https://services3.arcgis.com/HVJannk9zTiwmsU1/arcgis/rest/services/Doce/FeatureServer/0>

Feature Layer (hosted)
Private (Authentication required)
Item ID: d41ccc32cc8f471eacb5dc9b42eb3439

Created 25 de fevereiro de 2021
Modified 25 de fevereiro de 2021

Layer URL: <https://services3.arcgis.com/HVJannk9zTiwmsU1/arcgis/rest/services/Doce/FeatureServer/0>

Overview Usage Settings



E é aqui que informamos o link
do nosso shapefile

(Sua conta pode ter vários. Quantos? Vai depender do seu plano R\$)

```

7  ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");
8  ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);
9  this.mMapView.setMap(map);
10 // load the shapefile with a local path
11 ServiceFeatureTable shapefileFeatureTable = new ServiceFeatureTable("link hosted layer");
12 // create a feature layer to display the shapefile
13 this.shapefileFeatureLayer = new FeatureLayer(shapefileFeatureTable);
14 // add the feature layer to the map
15 this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);
16 this.mMapView.setViewpoint(new Viewpoint(-20.0,
17 }

```

<https://services3.arcgis.com/HVJannk9zTiwmsU1/arcgis/rest/services/Doce/FeatureServer/0>

Atenção

- Precisamos colocar **/0** no final do link

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4  
5     private void setupMapHidrografia() {  
6         //authentication with an API key or named user is required to access basemaps and other location services  
7         ArcGISRuntimeEnvironment.setApiKey("AAPK21a0600b4264487faa1ff7d");  
8         ArcGISMap map = new ArcGISMap(BasemapStyle.ARCGIS_TOPOGRAPHIC);  
9         this.mMapView.setMap(map);  
10        // load the shapefile with a local path  
11        ServiceFeatureTable shapefileFeatureTable = new ServiceFeat  
12        // create a feature layer to display the shapefile  
13        this.shapefileFeatureLayer = new FeatureLayer(shapefileFeat  
14        // add the feature layer to the map  
15        this.mMapView.getMap().getOperationalLayers().add(shapefileFeatureLayer);  
16        this.mMapView.setViewpoint(new Viewpoint(-20.0, -41.0, 2500000.0));  
17    }
```

<https://services3.arcgis.com/HVJannk9zTiwmsU1/arcgis/rest/services/Doce/FeatureServer/0>



Finalmente,

- Adiciona-se no **mapView**
- Seta as **coordenadas e escala**

Classe MainActivity

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3  
4  
5     @Override  
6     protected void onCreate(Bundle savedInstanceState) {  
7         super.onCreate(savedInstanceState);  
8         setContentView(R.layout.activity_main);  
9         // create a new map to display in the map view with a streets basemap  
10        mMapView = findViewById(R.id.mapView);  
11        setupMapHidrografia();  
12    }
```

Pronto

Agora é só *chamar*



App funcionando



- Assim como qualquer outro mapa é possível
 - Dar zoom in, zoom out
 - Se posicionar neste mapa com a hidrografia
 - Navegar por ela

- Porém
 - Não conseguimos acessar as informações deste mapa
 - Que estão na tabela de atributos

App funcionando



- Assim como qualquer outro mapa é possível
 - Dar zoom in, zoom out
 - Se posicionar neste mapa com a hidrografia
 - Navegar por ela
- Porém
 - Não conseguimos acessar as informações deste mapa
 - Que estão na tabela de atributos

E ai???



Acessando os dados do mapa

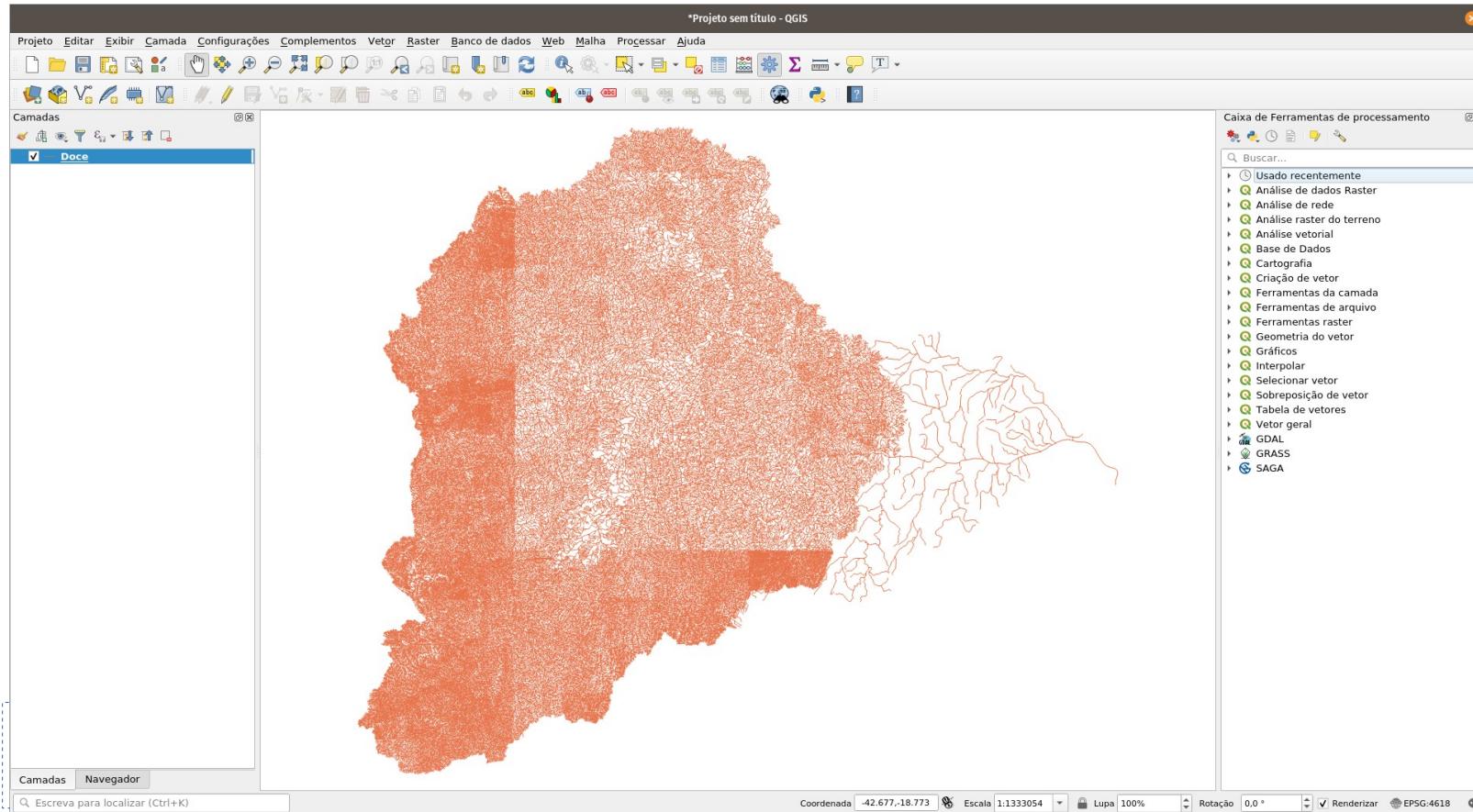
Classe Callout

- Responsável por fazer acesso à tabela de atributos

Acessando os dados do mapa

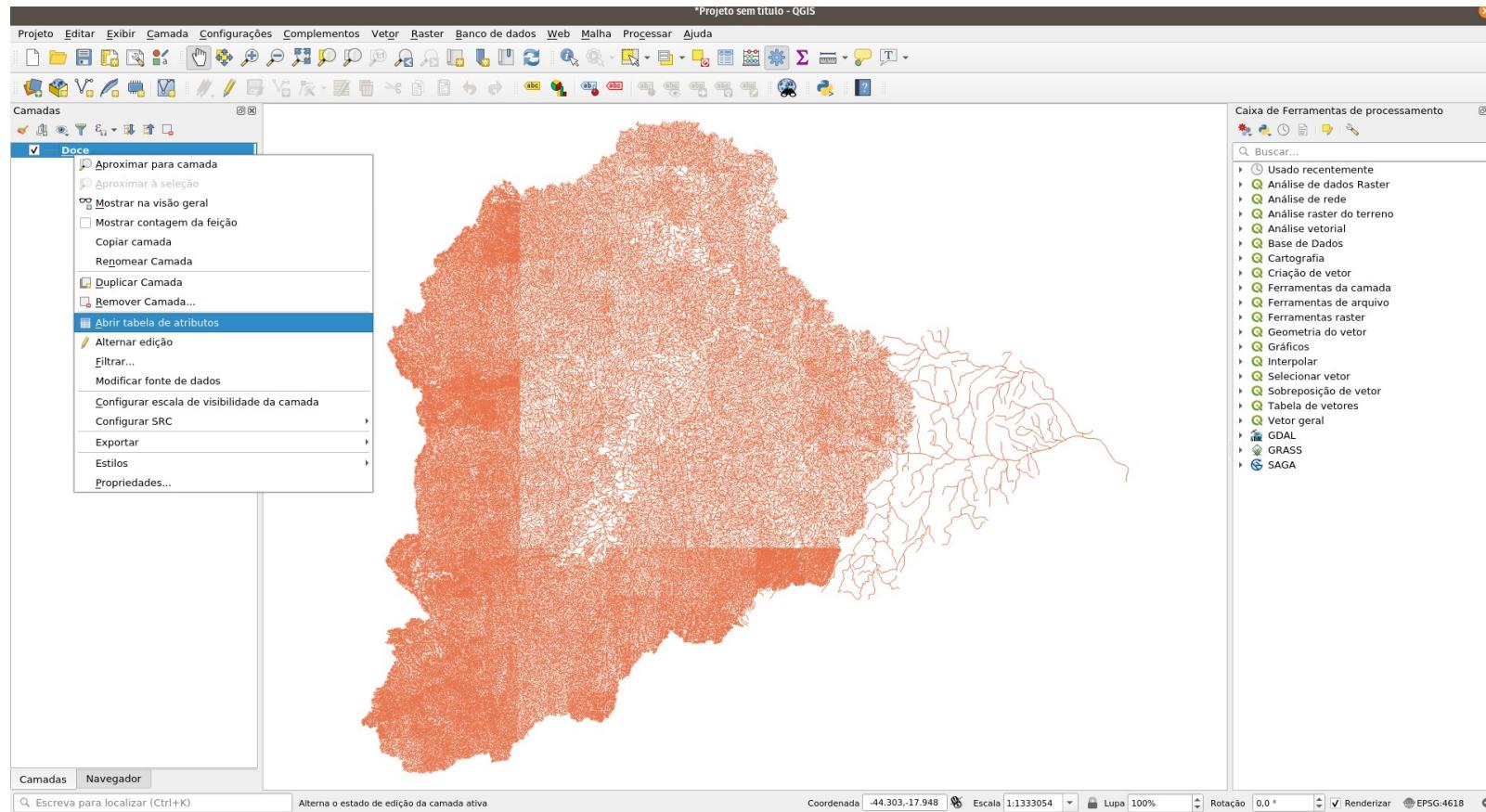
Classe Callout

- Responsável por fazer acesso à tabela de atributos



Classe Callout

- Responsável por fazer acesso à tabela de atributos



Classe Callout

- Responsável por fazer acesso à tabela de atributos

The screenshot shows the QGIS application interface. The title bar reads "Projeto sem título - QGIS". The main window displays a table titled "Doce :: Feições de totais: 181095, filtrado: 181095, selecionado: 0". The table has columns: cobacia, dedominial, nuareacon, nuareamont, noriocomp, nucomprido, Qmld, Q95, Q90, and Q710. The first row is highlighted with a green border. On the left, the "Camadas" (Layers) panel shows the "Doce" layer selected. A context menu is open over this layer, with the option "Abrir tabela de atributos" (Open attribute table) highlighted with a blue selection bar. Other options in the menu include "Aproximar para camada", "Aproximar à seleção", "Mostrar na visão geral", "Mostrar contagem da feição", "Copiar camada", "Renomear Camada", "Duplicar Camada", "Remover Camada...", "Alternar edição", "Filtrar...", "Modificar fonte de dados", "Configurar escala de visibilidade da camada", "Configurar SRC", "Exportar", "Estilos", and "Propriedades...". At the bottom of the interface, there are buttons for "Camadas" (Layers), "Navegador" (Browser), and a search bar with placeholder text "Escreva para localizar (Ctrl+K)".

	cobacia	dedominial	nuareacon	nuareamont	noriocomp	nucomprido	Qmld	Q95	Q90	Q710
1	7649454552	Não Federal	0,13711269	0,13711269	sem nome	0,54033	0,00086720...	6,99274719...	0,00012477...	5,07316953...
2	764945451	Não Federal	0,3083725	2,86521335...	sem nome	0,81801	0,01737731...	0,00146125...	0,00260734...	0,00106012...
3	7649454551	Não Federal	0,05849211	2,20266196...	sem nome	0,22845	0,01327443...	0,00112335...	0,00200442...	0,00081498...
4	76492467	Não Federal	0,30845463	4,40416062...	Córrego Ap...	5,46948000...	0,02881177...	0,00224612...	0,00400778...	0,00162953...
5	7649454553	Não Federal	0,06196631	2,00705716...	sem nome	0,24448	0,01205995...	0,00102359...	0,00182642...	0,00074261...
6	764945457	Não Federal	0,0064078	1,77448075...	sem nome	0,05512	0,01063886...	0,00090498...	0,00161477...	0,00065655...
7	7649246911	Não Federal	0,00700426	3,54083163...	Córrego Ap...	5,46948000...	0,02242546...	0,00180582...	0,00322215...	0,00131010...
8	764945459	Não Federal	0,03137053	1,56809840...	sem nome	0,20677	0,00935746...	0,00079973...	0,00142696...	0,00058019...
9	76494547	Não Federal	0,0381388	0,80766918	sem nome	0,17086	0,00476913...	0,00041191...	0,00073497...	0,00029883...
10	7649246913	Não Federal	0,13522934	3,36117495...	Córrego Ap...	5,46948000...	0,02131687...	0,00171419...	0,00305866...	0,00124363...
11	76494549	Não Federal	0,44589671	0,44589671	sem nome	1,02862000...	0,00262951...	0,00022740...	0,00040576...	0,00016498...
12	7649246915	Não Federal	0,35588451	2,98117659...	Córrego Ap...	5,46948000...	0,01859084...	0,00152040...	0,00271287...	0,00110303...
13	7648437692	Não Federal	0,09311562	0,09311562	sem nome	0,3217	0,00058339...	4,74889662...	8,47352142...	3,44527794...
14	76498291	Não Federal	0,16672057	1,22158470...	Córrego da ...	9,28357000...	0,00734673...	0,00062300...	0,00111164...	0,00045198...
15	764843762	Não Federal	0,14085173	0,14085173	sem nome	0,66769	0,00080480...	7,18343823...	0,00012817...	5,21151401...
16	764843764	Não Federal	0,1216413	0,1216413	sem nome	0,43727	0,00073491...	6,2037063e...	0,00011069...	4,5007281e...
17	764843766	Não Federal	0,25588784	0,25588784	sem nome	0,62976	0,00157576...	0,00013050...	0,00023285...	9,46785008...
18	76498293	Não Federal	0,19036787	0,8156186	Córrego da ...	9,28357000...	0,00485753...	0,00041596...	0,00074221...	0,00030177...
19	764843761	Não Federal	0,0128544	1,12636674...	sem nome	0,13064	0,00692381...	0,00057444...	0,00102499...	0,00041675...
20	76498297	Não Federal	0,02128567	0,28666268	sem nome	0,11317	0,00166642...	0,00014619...	0,00026086...	0,00010606...

Classe Callout

- Responsável por fazer acesso à tabela de atributos

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4     private Callout mCallout;  
5  
6     private void setupMap_callout() {  
7         this.setupHidrografia();  
8         mCallout = mMapView.getCallout();  
9  
10  
11  
12}
```

Classe Callout

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4     private Callout mCallout;  
5  
6     private void setupMap_callout() {  
7         this.setupHidrografia();  
8         mCallout = mMapView.getCallout();  
9     }  
10      
11    }  
12 }
```

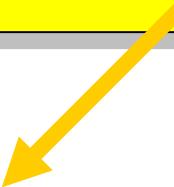
Agora
Precisamos vincular uma
ação de pesquisa no **touch** (clique)

Classe Callout

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4     private Callout mCallout;  
5  
6     private void setupMap_callout() {  
7         this.setupHidrografia();  
8         mCallout = mMapView.getCallout();  
9  
10        mMapView.setOnTouchListener(new DefaultMapViewOnTouchListener(this, mMapView) {  
11            @Override  
12            public boolean onSingleTapConfirmed(MotionEvent e) {  
13                ...  
14                }  
15            });  
16        }  
17    }
```

Agora
Precisamos vincular uma
ação de pesquisa no **touch** (clique)



Classe Callout

MainActivity

```
1 public class MainActivity extends AppCompatActivity {  
2     private MapView mMapView;  
3     private FeatureLayer shapefileFeatureLayer;  
4     private Callout mCallout;  
5  
6     private void setupMap_callout() {  
7         this.setupHidrografia();  
8         mCallout = mMapView.getCallout();  
9  
10    mMapView.setOnTouchListener(new DefaultMapViewOnTouchListener(this, mMapView) {  
11        @Override  
12        public boolean onSingleTapConfirmed(MotionEvent e) {  
13            ...  
14        }  
15    });  
16 }
```

Agora
Precisamos vincular uma
ação de pesquisa no **touch** (clique)



Zoom1

Classe Callout

```
mMapView.setOnTouchListener(new DefaultMapViewOnTouchListener(this, mMapView) {  
    @Override  
    public boolean onSingleTapConfirmed(MotionEvent e) {  
        // remove any existing callouts  
        if (mCallout.isShowing()) {  
            mCallout.dismiss();  
        }  
        // get the point that was clicked and convert it to a point in map coordinates  
        final Point screenPoint = new Point(Math.round(e.getX()), Math.round(e.getY()));  
        // create a selection tolerance  
        int tolerance = 10;  
        // use identifyLayerAsync to get tapped features  
        final ListenableFuture<IdentifyLayerResult> identifyLayerResultListenableFuture;  
        identifyLayerResultListenableFuture = mMapView.identifyLayerAsync(shapefileFeatureLayer, screenPoint,  
                           tolerance, false, 1);  
        identifyLayerResultListenableFuture.addDoneListener(() -> {  
            });  
        return super.onSingleTapConfirmed(e);  
    }  
});
```



Zoom1

Classe Callout

```
mMapView.setOnTouchListener(new DefaultMapViewOnTouchListener(this, mMapView) {  
    @Override  
    public boolean onSingleTapConfirmed(MotionEvent e) {  
        // remove any existing callouts  
        if (mCallout.isShowing()) {  
            mCallout.dismiss();  
        }  
        // get the point that was clicked and convert it to a point in map coordinates  
        final Point screenPoint = new Point(Math.round(e.getX()), Math.round(e.getY()));  
        // create a selection tolerance  
        int tolerance = 10;  
        // use identifyLayerAsync to get tapped features  
        final ListenableFuture<IdentifyLayerResult> identifyLayerResultListenableFuture;  
        identifyLayerResultListenableFuture = mMapView.identifyLayerAsync(shapefileFeatureLayer, screenPoint,  
                           tolerance, false, 1);  
        identifyLayerResultListenableFuture.addDoneListener(() -> {  
            });  
        return super.onSingleTapConfirmed(e);  
    }  
});
```

Sobreescrer o método
OnSingleTapConfirmed();



Zoom1

Classe Callout

```
mMapView.setOnTouchListener(new DefaultMapViewOnTouchListener(this, mMapView) {  
    @Override  
    public boolean onSingleTapConfirmed(MotionEvent e) {  
        // remove any existing callouts  
        if (mCallout.isShowing()) {  
            mCallout.dismiss();  
        }  
        // get the point that was clicked and convert it to a point  
        final Point screenPoint = new Point(Math.round(e.getX()), M  
            // create a selection tolerance  
        int tolerance = 10;  
        // use identifyLayerAsync to get tapped features  
        final ListenableFuture<IdentifyLayerResult> identifyLayerResultListenableFuture;  
        identifyLayerResultListenableFuture = mMapView.identifyLayerAsync(shapefileFeatureLayer, screenPoint,  
            tolerance, false, 1);  
        identifyLayerResultListenableFuture.addDoneListener(() -> {  
            });  
        return super.onSingleTapConfirmed(e);  
    }  
});
```

IdentifyLayerResult

Será o responsável por receber os dados



Zoom1

Classe Callout

```
mMapView.setOnTouchListener(new DefaultMapViewOnTouchListener(this, mMapView) {  
    @Override  
    public boolean onSingleTapConfirmed(MotionEvent e) {  
        // remove any existing callouts  
        if (mCallout.isShowing()) {  
            mCallout.dismiss();  
        }  
        // get the point that was clicked and convert it to a point  
        final Point screenPoint = new Point(Math.round(e.getX()), M  
            // create a selection tolerance  
        int tolerance = 10;  
        // use identifyLayerAsync to get tapped features  
        final ListenableFuture<IdentifyLayerResult> identifyLayerResultListenableFuture;  
        identifyLayerResultListenableFuture = mMapView.identifyLayerAsync(shapefileFeatureLayer, screenPoint,  
            tolerance, false, 1);  
        identifyLayerResultListenableFuture.addDoneListener(() -> {  
            });  
        return super.onSingleTapConfirmed(e);  
    }  
});
```

- Passamos este parâmetros para o **MapView**
- Ou seja, vinculamos ao componente da tela

A magnifying glass icon with a blue circular lens and a red handle, enclosed in a dashed red rectangular border. Below the icon, the text "Zoom1" is written.

Classe Callout

```
mMapView.setOnTouchListener(new DefaultMapViewOnTouchListener(this, mMapView) {  
    @Override  
    public boolean onSingleTapConfirmed(MotionEvent e) {  
        // remove any existing callouts  
        if (mCallout.isShowing()) {  
            mCallout.dismiss();  
        }  
        // get the point that was clicked and convert it to a point  
        final Point screenPoint = new Point(Math.round(e.getX()), M  
            // create a selection tolerance  
        int tolerance = 10;  
        // use identifyLayerAsync to get tapped features  
        final ListenableFuture<IdentifyLayerResult> identifyLayerResultListenableFuture;  
        identifyLayerResultListenableFuture = mMapView.identifyLayerAsync(shapefileFeatureLayer, screenPoint,  
            tolerance, false, 1);  
  
        identifyLayerResultListenableFuture.addDoneListener(() -> {  
            });  
            return super.onSingleTapConfirmed(e);  
    }  
});
```

Finalmente

- Faremos uma função lambda para **pegar** de fato os **dados e manipulá-los**



Zoom1

```
identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());

        for (GeoElement element : identifyLayerResult.getElements()) {
            Feature feature = (Feature) element;
            // create a map of all available attributes as name value pairs
            Map<String, Object> attr = feature.getAttributes();
            //cada Key é uma coluna da sua tabala de atributos
            Set<String> keys = attr.keySet();
            for (String key : keys) {
                //para cada hashmap, pegamos o valor da coluna por meio da KEY
                //Lembrando que KEY deve ser 'único'
                Object value = attr.get(key);
                // append name value pairs to text view
                calloutContent.append(key + " | " + value + "\n");
            }
        }
    }
}
```



```

identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());
    }

    for (GeoElement element : identifyLayerResult.getElements()) {
        Feature feature = (Feature) element;
        // create a map of all available attributes as name value pairs
        Map<String, Object> attr = feature.getAttributes();
        //cada Key é uma coluna da sua tabala de atributos
        Set<String> keys = attr.keySet();
        for (String key : keys) {
            //para cada hashmap, pegamos o valor da coluna por meio da KEY
            //Lembrando que KEY deve ser 'único'
            Object value = attr.get(key);
            // append name value pairs to text view
            calloutContent.append(key + " | " + value + "\n");
        }
    }
}

```

IdentifyLayerResult

- Possui os dados da tabela de atributos, referente ao clique do usuário.
- Um registo da tabela de atributos



```

identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
    }
}

```

Doce : Feições de totais: 181095, filtrado: 181095, selecionado: 0

	cobacia	dedominial	nuareacont	nuareamont	noriocomp	nucomproprio	Omlid	O95	O90	O710
1	7649454552	Não Federal	0,13711269	0,13711269	sem nome	0,54033	0,00086720...	6,99274719...	0,00012477...	5,07316953...
2	7649454541	Não Federal	0,3083725	2,86521335...	sem nome	0,81801	0,01737731...	0,00146125...	0,00260734...	0,00106012...
3	7649454551	Não Federal	0,05849211	2,20266196...	sem nome	0,22845	0,01327443...	0,00112335...	0,00200442...	0,00081498...
4	76492467	Não Federal	0,30845463	4,40416062...	Córrego Ap...	5,46948000...	0,02881177...	0,00224612...	0,00400778...	0,00162953...
5	7649454553	Não Federal	0,06196631	2,00705716...	sem nome	0,24448	0,01205995...	0,00102359...	0,00182642...	0,00074261...
6	7649454547	Não Federal	0,0064078	1,77448075...	sem nome	0,05512	0,01063886...	0,00090498...	0,00161477...	0,00065655...
7	7649246911	Não Federal	0,00700426	3,54083163...	Córrego Ap...	5,46948000...	0,02242546...	0,00180582...	0,00322215...	0,00131010...
8	7649454549	Não Federal	0,03137053	1,56809840...	sem nome	0,20677	0,00935746...	0,00079973...	0,00142696...	0,00058019...
9	76494547	Não Federal	0,0381388	0,80766918	sem nome	0,17086	0,00476913...	0,00041191...	0,00073497...	0,00029883...
10	7649246913	Não Federal	0,13522934	3,36117495...	Córrego Ap...	5,46948000...	0,02131687...	0,00171419...	0,00305866...	0,00124363...
11	76494549	Não Federal	0,44589671	0,44589671	sem nome	1,02862000...	0,00262951...	0,00022740...	0,00040576...	0,00016498...
12	7649246915	Não Federal	0,35588451	2,98117659...	Córrego Ap...	5,46948000...	0,01859084...	0,00152040...	0,00271287...	0,00110303...
13	7648437692	Não Federal	0,09311562	0,09311562	sem nome	0,3217	0,00058339...	4,74889662...	8,47352142...	3,44527794...
14	76498291	Não Federal	0,16672057	1,22158470...	Córrego da ...	9,28357000...	0,00734673...	0,00062300...	0,00111164...	0,00045198...
15	764843762	Não Federal	0,14085173	0,14085173	sem nome	0,66769	0,00080480...	7,18343823...	0,00012817...	5,21151401...
16	764843764	Não Federal	0,1216413	0,1216413	sem nome	0,43727	0,00073491...	6,2037063e...	0,00011069...	4,5007281e...
17	764843766	Não Federal	0,25588784	0,25588784	sem nome	0,62976	0,00157576...	0,00013050...	0,00023285...	9,46785008...
18	76498293	Não Federal	0,19036787	0,8156186	Córrego da ...	9,28357000...	0,00485753...	0,00041596...	0,00074221...	0,00030177...
19	764843761	Não Federal	0,0128544	1,12636674...	sem nome	0,13064	0,00692381...	0,00057444...	0,00102499...	0,00041675...
20	76498297	Não Federal	0,02128567	0,28666268	sem nome	0,11317	0,00166642...	0,00014619...	0,00026086...	0,00010606...

Mostrar todas as feições

IndetifyLayerResult

- Possui os dados da tabela de atributos, referente ao clique do usuário.
- Um registro da tabela de atributos



```
identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());

        for (GeoElement element : identifyLayerResult.getElements()) {
            Feature feature = (Feature) element;
            // create a map of all available attributes as name value pairs
            Map<String, Object> attr = feature.getAttributes();
            //cada Key é uma coluna da sua tabala de atributos
            Set<String> keys = attr.keySet();
            for (String key : keys) {
                //para cada hashmap, pegamos o valor da coluna por meio da KEY
                //Lembrando que KEY deve ser 'único'
                Object value = attr.get(key);
                // append name value pairs to text view
                calloutContent.append(key + " | " + value + "\n");
            }
        }
    }
}
```

Cria-se

- Dinamicamente uma **TextView** para apresentamos na tela do **App**



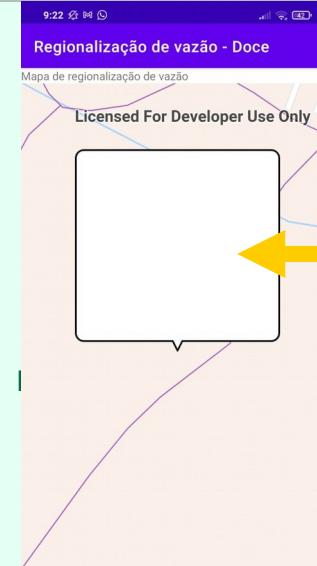
```

identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());

        for (GeoElement element : identifyLayerResult.getElements()) {
            Feature feature = (Feature) element;
            // create a map of all available attributes as name value pairs
            Map<String, Object> attr = feature.getAttributes();
            //cada Key é uma coluna da sua tabala de atributos
            Set<String> keys = attr.keySet();
            for (String key : keys) {
                //para cada hashmap, pegamos o valor da coluna por meio da
                //Lembrando que KEY deve ser 'único'
                Object value = attr.get(key);
                // append name value pairs to text view
                calloutContent.append(key + " | " + value + "\n");
            }
        }
    }
}

```

- Cria-se**
- Dinamicamente uma **TextView** para apresentamos na tela do **App**



Ao clicar num ponto do rio abrirá um **popup** (TextView)
(Com as informações daquele ponto)

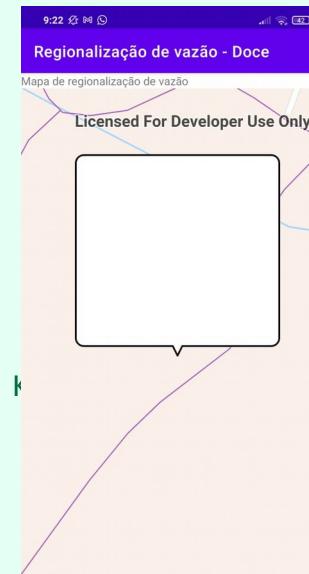
```

identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());
    }

    for (GeoElement element : identifyLayerResult.getElements()) {
        Feature feature = (Feature) element;
        // create a map of all available attributes as name value pairs
        Map<String, Object> attr = feature.getAttributes();
        //cada Key é uma coluna da sua tabala de atributos
        Set<String> keys = attr.keySet();
        for (String key : keys) {
            //para cada hashmap, pegamos o valor da coluna por meio da key
            //Lembrando que KEY deve ser 'único'
            Object value = attr.get(key);
            // append name value pairs to text view
            calloutContent.append(key + " | " + value + "\n");
        }
    }
}

```

- Parâmetros do **TextView**



```

identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());

        for (GeoElement element : identifyLayerResult.getElements()) {
            Feature feature = (Feature) element;
            // create a map of all available attributes as name value pairs
            Map<String, Object> attr = feature.getAttributes();
            // cada Key é uma coluna da sua tabela de atributos
            Set<String> keys = attr.keySet();
            for (String key : keys) {
                // para cada hashmap, pegamos o valor da coluna por meio da KEY
                // Lembrando que KEY deve ser 'único'
                Object value = attr.get(key);
                // append name value pairs to text view
                calloutContent.append(key + " | " + value + "\n");
            }
        }
    }
}

```

- Vamos iterar sobre as colunas de um determinado registro(GeoElement) da Tabela de atributos



```

identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());

        for (GeoElement element : identifyLayerResult.getElements()) {
            Feature feature = (Feature) element;
            // create a map of all available attributes as name value pairs
            Map<String, Object> attr = feature.getAttributes();
            // cada Key é uma coluna da sua tabela de atributos
            Set<String> keys = attr.keySet();
            for (String key : keys) {
                // para cada hashmap, pegamos o valor da coluna por meio da KEY
                // Lembrando que KEY deve ser 'único'
                Object value = attr.get(key);
                // append name value pairs to text view
                calloutContent.append(key + " | " + value + "\n");
            }
        }
    }
}

```

- Vamos iterar sobre as colunas de um determinado registro([GeoElement](#)) da Tabela de atributos

Doce : Feijões de totais: 181095, filtrado: 181095, selecionado: 0

	id	cobraria	defomilial	nucreacet	nucreasment	noricomio	nucomario	Omid	OVS	O710
2	764945451	Não Federal	0.3083725	2.886521355	sem nome	0.81801	0.01737731...	0.00146125...	0.00260734...	0.0106012...
3	7649454551	Não Federal	0.05849211	2.2026196...	sem nome	0.22845	0.0137443...	0.0112335...	0.00200442...	0.0081498...
4	76492467	Não Federal	0.30845463	0.40416062...	Córegos Ap...	5.46948000	0.02881177...	0.00224612...	0.00400778...	0.0162953...
5	7649454553	Não Federal	0.06196631	2.00705716...	sem nome	0.24448	0.0125995...	0.0102539...	0.00182642...	0.0074261...
6	764945457	Não Federal	0.0604078	1.7448075...	sem nome	0.05512	0.01053886...	0.00909489...	0.00161477...	0.0066555...
7	7649246911	Não Federal	0.0700426	3.54083163...	Córegos Ap...	5.46948000	0.0224254...	0.018058...	0.0032215...	0.0013103...
8	7649454559	Não Federal	0.03137053	1.56809840...	sem nome	0.20677	0.00935746...	0.00079973...	0.00142696...	0.0058019...
9	764945457	Não Federal	0.0381388	0.80766918	sem nome	0.17086	0.00476913...	0.00041191...	0.00073497...	0.0029883...
10	7649246913	Não Federal	0.13522934	3.3617495...	Córegos Ap...	5.46948000	0.02131687...	0.0017149...	0.00305866...	0.00124963...
11	764945459	Não Federal	0.44589671	0.44589671	sem nome	1.02862000...	0.00262951...	0.00027240...	0.00040576...	0.0016498...
12	7649246915	Não Federal	0.35588451	2.9817659...	Córegos Ap...	5.46948000	0.01859084...	0.00152040...	0.00271287...	0.00110303...
13	7648437692	Não Federal	0.09311562	0.09311562	sem nome	0.3217	0.00058339...	4.74889662...	8.47352142...	3.45527794...
14	76498291	Não Federal	0.16672057	1.22158470...	Córegos da ...	9.28357000...	0.000734673...	0.00062300...	0.00111164...	0.0045198...
15	764843762	Não Federal	0.14085173	0.14085173	sem nome	0.65769	0.00080480...	7.1834582...	0.00012817...	2.2151401...
16	764843764	Não Federal	0.1216413	0.1216413	sem nome	0.43727	0.00073491...	6.2037063...	0.00011069...	4.5072781...
17	764843766	Não Federal	0.25588784	0.25588784	sem nome	0.62976	0.00157576...	0.00013050...	0.00023285...	9.46785008...
18	76498293	Não Federal	0.19036787	0.18516186	Córegos da ...	9.28357000...	0.0048573...	0.00049156...	0.0074221...	0.00303177...
19	764843761	Não Federal	0.0128544	1.12636674...	sem nome	0.13064	0.00692381...	0.00057444...	0.00102499...	0.0041675...
20	76498297	Não Federal	0.02128567	0.28666268	sem nome	0.13137	0.00166642...	0.00014619...	0.00026086...	0.00010606...

Mostrar todas as feijões .

```
identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
```

- Usaremos um **hashMap**, onde teremos
 - **Key:** nome da coluna
 - **Valeu:** valor da coluna

```
    .ACK);  
    .  
    .enabled(true);  
    .ew.SCROLLBARS_INSIDE_INSET);  
    . ScrollingMovementMethod());
```

```
for (GeoElement element : identifyLayerResult.getElements()) {  
    Feature feature = (Feature) element;  
    // create a map of all available attributes as name value pairs  
    Map<String, Object> attr = feature.getAttributes();  
    //cada Key é uma coluna da sua tabala de atributos  
    Set<String> keys = attr.keySet();  
    for (String key : keys) {  
        //para cada hashmap, pegamos o valor da coluna por meio da KEY  
        //Lembrando que KEY deve ser 'único'  
        Object value = attr.get(key);  
        // append name value pairs to text view  
        calloutContent.append(key + " | " + value + "\n");  
    }  
}
```

Dado - Feixes de totais 181095, filtrado: 181095, selecionado: 0										
	coabia	defominal	nuareazet	nuaramont	nuaricoma	nuaricomo	Omid	Q95	Q90	Q710
1	7649454552	Não Federal	0.13711269	0.13711269	sem nome	0.54033	0.00086720	6.99274719	0.00012477	5.70136953
2	764945451	Não Federal	0.3087325	0.2651335	sem nome	0.81801	0.01377371	0.00146125	0.0260734	0.0106012
3	764945451	Não Federal	0.05849211	2.2062165	sem nome	0.22845	0.01377443	0.0011235	0.0200442	0.0081498
4	76492467	Não Federal	0.30845683	4.40416062	Córegos Ap.	0.54948000	0.02811177	0.00224612	0.04000798	0.0162953
5	7649454553	Não Federal	0.0169631	2.0075710	sem nome	0.24484	0.01205995	0.00125995	0.0182642	0.0047461
6	764945457	Não Federal	0.0064078	1.7744075	sem nome	0.05012	0.0163886	0.0099498	0.0161477	0.0066555
7	7649246911	Não Federal	0.0070042	3.54083163	Córegos Ap.	0.54948000	0.02242546	0.00180582	0.0322215	0.0131010
8	764945459	Não Federal	0.0317051	1.5681940	sem nome	0.20677	0.00935748	0.0099797	0.0142699	0.0059819
9	764945457	Não Federal	0.0381384	0.8766918	sem nome	0.17086	0.0064409	0.0004119	0.0073497	0.0029883
10	7649246913	Não Federal	0.13522934	3.36117495	Córegos Ap.	0.54948000	0.02131687	0.0017149	0.0305866	0.0124363
11	764945459	Não Federal	0.44598671	0.44598671	sem nome	1.0286200	0.0026951	0.0027450	0.00404728	0.0016498
12	7649246915	Não Federal	0.35885481	2.9811765	Córegos Ap.	0.54948000	0.01859084	0.0015204	0.00721287	0.0011030
13	7648437369	Não Federal	0.09311562	0.09311562	sem nome	0.3217	0.0058339	4.74889642	8.47352142	3.4452797
14	76482921	Não Federal	0.16672057	1.22138470	Córegos da ...	0.28357000	0.00734473	0.00062350	0.0111164	0.0045198
15	764843762	Não Federal	0.10480517	0.10480517	sem nome	0.567867	0.00080480	7.1834832	0.00012817	5.2151401
16	764843764	Não Federal	0.1216413	0.1216413	sem nome	0.43727	0.00073491	6.2307363	0.00011069	4.5007281
17	764843766	Não Federal	0.25588784	0.25588784	sem nome	0.62976	0.00157576	0.00013055	0.00228528	4.96785008
18	76482923	Não Federal	0.19038787	0.19038786	Córegos da ...	0.28357000	0.00734476	0.00062350	0.0074221	0.00301717
19	764843761	Não Federal	0.0128544	1.12636674	sem nome	0.13064	0.00692381	0.00057444	0.00102499	0.00041675
20	76482927	Não Federal	0.02128567	0.28666268	sem nome	0.11137	0.00166442	0.00014619	0.0026086	0.00010606

```

identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());
    }
}

```

- Iteramos sobre as chaves(**key**) "únicas"
- E finalmente montamos nossa saída

```

        layerResult.getElements()) {
    element;
    able attributes as name value pairs
        ture.getAttributes();
}

```

```

//cada Key é uma coluna da sua tabala de atributos
Set<String> keys = attr.keySet();
for (String key : keys) {
    //para cada hashmap, pegamos o valor da coluna por meio da KEY
    //Lembrando que KEY deve ser 'único'
    Object value = attr.get(key);
    // append name value pairs to text view
    calloutContent.append(key + " | " + value + "\n");
}
}

```

Doc : Feijões de totais: 181095, filtrado: 181095, selecionado: 0

	codbar	defominal	nusareacet	nusareamot	noriconomo	nucomorio	Omid	095	098	0710
2	764945451	Não Federal	0.3083725	2.886521355	sem nome	0.54033	0.00086720...	6.99274719...	0.00012477...	5.07316953...
3	7649454551	Não Federal	0.05849211	2.20266196...	sem nome	0.22845	0.01377443...	0.011235...	0.00200442...	0.00081498...
4	76492467	Não Federal	0.30845463	4.40416062...	Córegos Ap...	5.46948000...	0.02881177...	0.00224612...	0.00400778...	0.0162953...
5	7649454553	Não Federal	0.06196631	2.00705716...	sem nome	0.24448	0.01259959...	0.0102359...	0.00182642...	0.00074261...
6	764945457	Não Federal	0.0064078	1.77440705...	sem nome	0.05512	0.01053886...	0.0009498...	0.00161477...	0.00065555...
7	7649246911	Não Federal	0.00700426	3.54083163...	Córegos Ap...	5.46948000...	0.0224254...	0.0018058...	0.0032215...	0.001310310...
8	7649454559	Não Federal	0.03137053	1.56809840...	sem nome	0.20671	0.00935746...	0.00079973...	0.00142696...	0.00058019...
9	764945457	Não Federal	0.0381388	0.80766918	sem nome	0.17086	0.00476913...	0.00041191...	0.00073497...	0.00029883...
10	7649246913	Não Federal	0.13522934	3.3617495...	Córegos Ap...	5.46948000...	0.0211687...	0.0017419...	0.00305866...	0.00124963...
11	764945459	Não Federal	0.44589671	0.44589671	sem nome	1.02862000...	0.00262951...	0.00022740...	0.00040576...	0.00016498...
12	7649246915	Não Federal	0.35588451	2.98117659...	Córegos Ap...	5.46948000...	0.01859084...	0.00152040...	0.00271287...	0.00110303...
13	7648437692	Não Federal	0.09311562	0.09311562	sem nome	0.3217	0.00058339...	4.74889662...	8.47352142...	3.44527794...
14	76498291	Não Federal	0.16672057	1.22158470...	Córegos da ...	9.28357000...	0.00734673...	0.00062300...	0.0011164...	0.0045198...
15	764843762	Não Federal	0.14085173	1.14085173	sem nome	0.66769	0.00080480...	7.1834582...	0.00012817...	2.5151401...
16	764843764	Não Federal	0.1216413	0.1216413	sem nome	0.43727	0.00073491...	6.2037063...	0.00011069...	4.5072781...
17	764843766	Não Federal	0.25588784	0.25588784	sem nome	0.62976	0.00157576...	0.00013050...	0.00023285...	9.46785008...
18	76498293	Não Federal	0.19036787	0.18516186	Córegos da ...	9.28357000...	0.0048573...	0.00041596...	0.00074221...	0.00030177...
19	764843761	Não Federal	0.0128544	1.12636674...	sem nome	0.13064	0.00692381...	0.00057444...	0.00102499...	0.00041675...
20	76498297	Não Federal	0.02128567	0.28666268	sem nome	0.13117	0.00166642...	0.00014619...	0.00026086...	0.00010606...

Mostrar todas as feijões.

```

identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());
    }
}

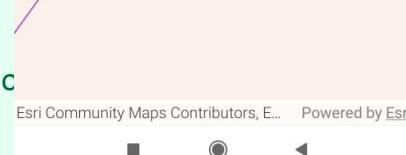
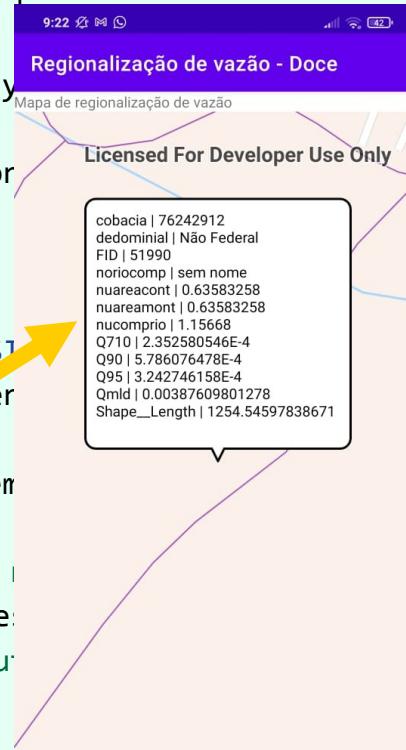
```

- Iteramos sobre as chaves(**key**) "únicas"
- E finalmente *montamos nossa saída*

```

//cada Key é uma coluna da sua tabala de atributos
Set<String> keys = attr.keySet();
for (String key : keys) {
    //para cada hashmap, pegamos o valor da coluna
    //Lembrando que KEY deve ser 'único'
    Object value = attr.get(key);
    // append name value pairs to text view
    calloutContent.append(key + " | " + value + "\n");
}

```



	cobacia	dedominial	nuareacomp	nuareamont	noricompri	nuocomprido	Qmid	Q95	Q90	Q710	
7649454552	Não Federal	0.13711269	0.13711269 sem nome	0.54033	0.00086720...	6.99274719...	0.00012477...	5.07316953...			
7649454551	Não Federal	0.3083725	2.88652135...	sem nome	0.81801	0.01737731...	0.00146125...	0.00260734...	0.00106012...		
7649454551	Não Federal	0.05849211	2.20266196...	sem nome	0.22845	0.0137443...	0.0011235...	0.00200442...	0.00081498...		
76492467	Não Federal	0.30845463	4.40416062...	Córrego Ap...	5.46948000...	0.02881177...	0.00224612...	0.00400778...	0.00162953...		
764945553	Não Federal	0.06196631	2.00705716...	sem nome	0.24448	0.0125995...	0.0010239...	0.00182642...	0.00074261...		
7649454557	Não Federal	0.0064078	1.77447805...	sem nome	0.05512	0.01063886...	0.00090498...	0.00161477...	0.00065555...		
7649246911	Não Federal	0.00700426	3.54083163...	Córrego Ap...	5.46948000...	0.02242546...	0.0018058...	0.00322215...	0.00131010...		
7649454559	Não Federal	0.03137053	1.56809840...	sem nome	0.20671	0.00935746...	0.00079973...	0.00142696...	0.00058019...		
76494545	Não Federal	0.0381388	0.80766918	sem nome	0.17086	0.00476913...	0.00041191...	0.00073497...	0.00029883...		
7649246913	Não Federal	0.13522934	3.3617495...	Córrego Ap...	5.46948000...	0.02131687...	0.00171419...	0.00305866...	0.00124963...		
11	76494549	Não Federal	0.44589671	0.44589671 sem nome	1.02862000...	0.00262951...	0.00027240...	0.00040576...	0.00016498...		
12	7649246915	Não Federal	0.35598451	2.98117659...	Córrego Ap...	5.46948000...	0.01859084...	0.00152040...	0.00271287...	0.00110303...	
13	7648437692	Não Federal	0.09311562	0.09311562 sem nome	0.3217	0.00058339...	0.47488966...	8.47352142...	3.44527794...		
14	76498291	Não Federal	0.16672057	1.22158470...	Córrego da ...	9.28357000...	0.00734673...	0.00062300...	0.00111164...	0.0045198...	
15	764843762	Não Federal	0.14085173	0.14085173 sem nome	0.66769	0.00080480...	7.1834382...	0.00012817...	2.15151401...		
16	764843764	Não Federal	0.1216413	0.1216413 sem nome	0.43727	0.00073491...	6.2037063...	0.00011069...	4.5072721...		
17	764843766	Não Federal	0.25588784	0.25588784 sem nome	0.62976	0.00157576...	0.00010350...	0.00023285...	9.46785008...		
18	76498293	Não Federal	0.19036787	0.18516186 Córrego da ...	9.28357000...	0.0048573...	0.00041596...	0.00074221...	0.00030177...		
19	764843761	Não Federal	0.0128544	1.12636674... sem nome	0.13064	0.00692381...	0.00057444...	0.00102499...	0.00041675...		
20	76498297	Não Federal	0.02128567	0.28666268 sem nome	0.13137	0.00166642...	0.00014619...	0.00026086...	0.00010606...		

Mostrar todas as feições

```

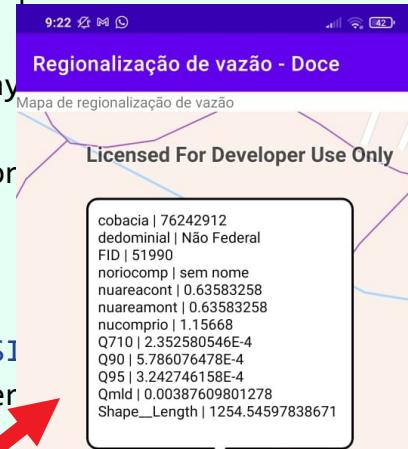
identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());
        calloutContent.setMovementMethod(new ScrollingMovementMethod());

        for (GeoElement element : identifyLayerResult.getFeatures()) {
            Feature feature = (Feature) element;
            // create a callout
            Map<String, String> keys = feature.getAttributes();
            // cada hashmap tem uma key
            Set<String> keysSet = keys.keySet();
            for (String key : keysSet) {
                //para cada hashmap, pegamos o valor da chave
                //Lembrando que KEY deve ser 'único'
                Object value = attr.get(key);
                // append name value pairs to text view
                calloutContent.append(key + " | " + value + "\n");
            }
        }
    }
}

```

Atenção

Estas imagens são meramente ilustrativas
Não representam o mesmo ponto



Esri Community Maps Contributors, E... Powered by Esri

ID	cobacia	dedominial	nuareacomp	nuareamont	noriocomp	nuocomprio	Omid	O95	O98	O710
7649454552	Não Federal	0.13711269	0.13711269 sem nome	0.54033	0.0096720...	6.99274719...	0.00012477...	5.07316953...		
7649454551	Não Federal	0.05849211	2.20266196...	sem nome	0.22845	0.0137443...	0.001235...	0.00200442...	0.00081498...	
76492467	Não Federal	0.30845463	4.40416062...	Córrego Ap...	5.46948000...	0.02881177...	0.00224612...	0.00400778...	0.00162953...	
7649454553	Não Federal	0.06196631	2.00705716...	sem nome	0.24448	0.0125995...	0.0010259...	0.00182642...	0.00074261...	
7649454557	Não Federal	0.0604078	1.77448075...	sem nome	0.05512	0.01063886...	0.00090498...	0.00161477...	0.00066555...	
7649246911	Não Federal	0.00700426	3.54083163...	Córrego Ap...	5.46948000...	0.0224526...	0.0018058...	0.0032215...	0.00131010...	
7649454559	Não Federal	0.03137053	1.56809840...	sem nome	0.20677	0.00935746...	0.00079973...	0.00142696...	0.00058019...	
76494547	Não Federal	0.0381388	0.80766918	sem nome	0.17086	0.00476913...	0.00041191...	0.00073497...	0.00029883...	
7649246913	Não Federal	0.13522934	3.3617495...	Córrego Ap...	5.46948000...	0.0211687...	0.00171491...	0.00305866...	0.00124963...	
11	76494549	Não Federal	0.44589671	0.44589671 sem nome	0.02862000...	0.00262951...	0.00022740...	0.00040576...	0.00016498...	
12	7649246915	Não Federal	0.35588451	2.98117659...	Córrego Ap...	5.46948000...	0.01859084...	0.00152040...	0.00271287...	0.00110303...
13	7648437692	Não Federal	0.09311562	0.09311562 sem nome	0.3217	0.00058339...	4.74889662...	8.47352142...	3.44527794...	
14	76498291	Não Federal	0.16672057	1.22158470...	Córrego da ...	9.28357000...	0.00734673...	0.00062300...	0.00111164...	0.0045198...
15	764843762	Não Federal	0.14085173	0.14085173 sem nome	0.66769	0.00080480...	7.1834382...	0.00012817...	2.15151401...	
16	764843764	Não Federal	0.1216413	0.1216413 sem nome	0.43727	0.00073491...	6.2037063...	0.00011069...	4.507281...	
17	764843766	Não Federal	0.25588784	0.25588784 sem nome	0.62976	0.00157576...	0.00010350...	0.00023285...	9.46785008...	
18	76498293	Não Federal	0.19036787	0.18516186 Córrego da ...	9.28357000...	0.0048573...	0.00041596...	0.00074221...	0.00030177...	
19	764843761	Não Federal	0.0128544	1.12636674...	sem nome	0.13064	0.00692381...	0.00057444...	0.00102499...	0.00041675...
20	76498297	Não Federal	0.02128567	0.28666268 sem nome	0.13137	0.00166642...	0.00014619...	0.00026086...	0.00010606...	

Mostrar todas as feições .

```
identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        IdentifyLayerResult identifyLayerResult = identifyLayerResultListenableFuture.get();
        // create a textView to display field values
        TextView calloutContent = new TextView(getApplicationContext());
        calloutContent.setTextColor(Color.BLACK);
        calloutContent.setSingleLine(false);
        calloutContent.setVerticalScrollBarEnabled(true);
        calloutContent.setScrollBarStyle(View.SCROLLBARS_INSIDE_INSET);
        calloutContent.setMovementMethod(new ScrollingMovementMethod());

        for (GeoElement element : identifyLayerResult.getElements()) {
            Feature feature = (Feature) element;
            // create a map of all available attributes as name value pairs
            Map<String, Object> attr = feature.getAttributes();
            //cada Key é uma coluna da sua tabala de atributos
            Set<String> keys = attr.keySet();
            for (String key : keys) {
                //para cada hashmap, pegamos o valor da coluna por meio da KEY
                //Lembrando que KEY deve ser 'único'
                Object value = attr.get(key);
                // append name value pairs to text view
                calloutContent.append(key + " | " + value + "\n");
            }
        }
    }
}
```

Para finalizar

- Um último zoom

```
identifyLayerResultListenableFuture.addDoneListener(() -> {
    try {
        ...
        for (GeoElement element : identifyLayerResult.getElements()) {
            Feature feature = (Feature) element;
            // create a map of all available attributes as name value pairs
            Map<String, Object> attr = feature.getAttributes();
            //cada Key é uma coluna da sua tabala de atributos
            Set<String> keys = attr.keySet();
            for (String key : keys) {
                //para cada hashmap, pegamos o valor da coluna por meio da KEY
                //Lembrando que KEY deve ser 'único'
                Object value = attr.get(key);
                // append name value pairs to text view
                calloutContent.append(key + " | " + value + "\n");
            }
            // center the mapview on selected feature
            Envelope envelope = feature.getGeometry().getExtent();
            mMapView.setViewpointGeometryAsync(envelope, 200);
            // show callout
            mCallout.setLocation(envelope.getCenter());
            mCallout.setContent(calloutContent);
            mCallout.show();
        }
    } catch (Exception e1) {
        Log.e(getResources().getString(R.string.app_name), "Select feature failed: " + e1.getMessage());
    }
});
```

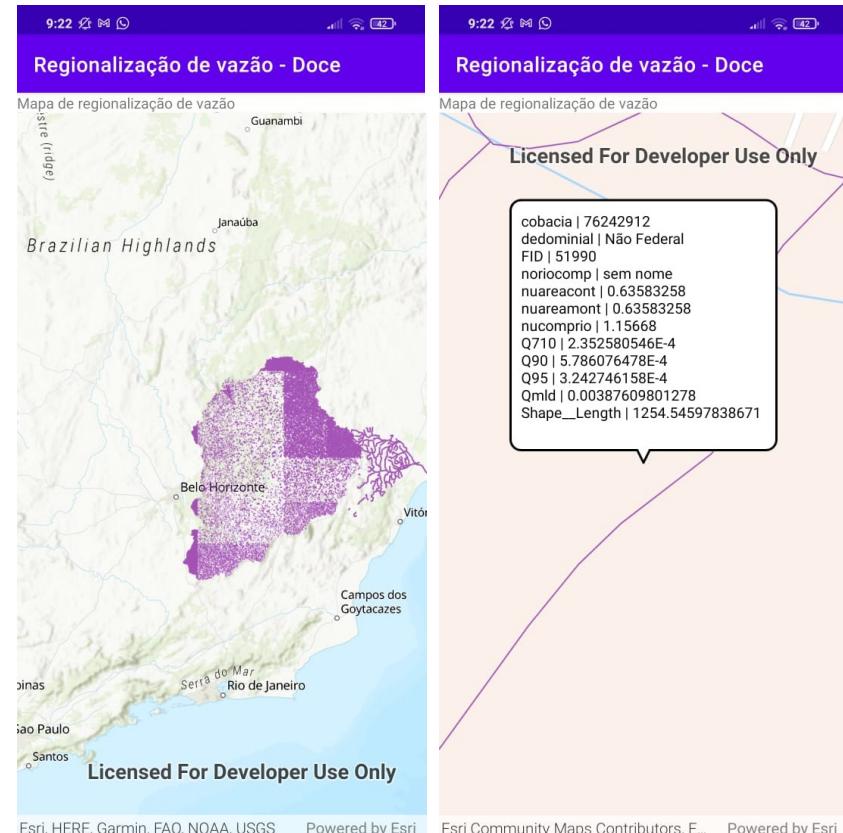
Para finalizar
• Um último zoom

Na verdade é o trecho de código que não coube .
Referente a renderização do **popup**



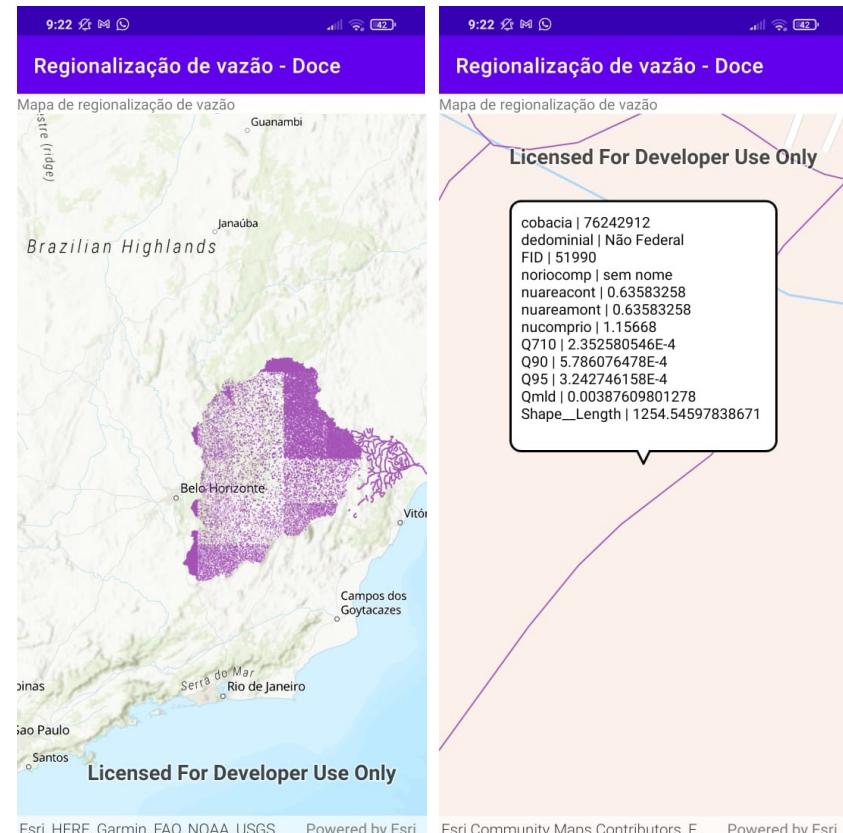
Comentários finais

- Com o **hosted Layer** da ArcGis developers
 - Pode-se subir qualquer mapa e acessá-lo
 - Via celular ou website



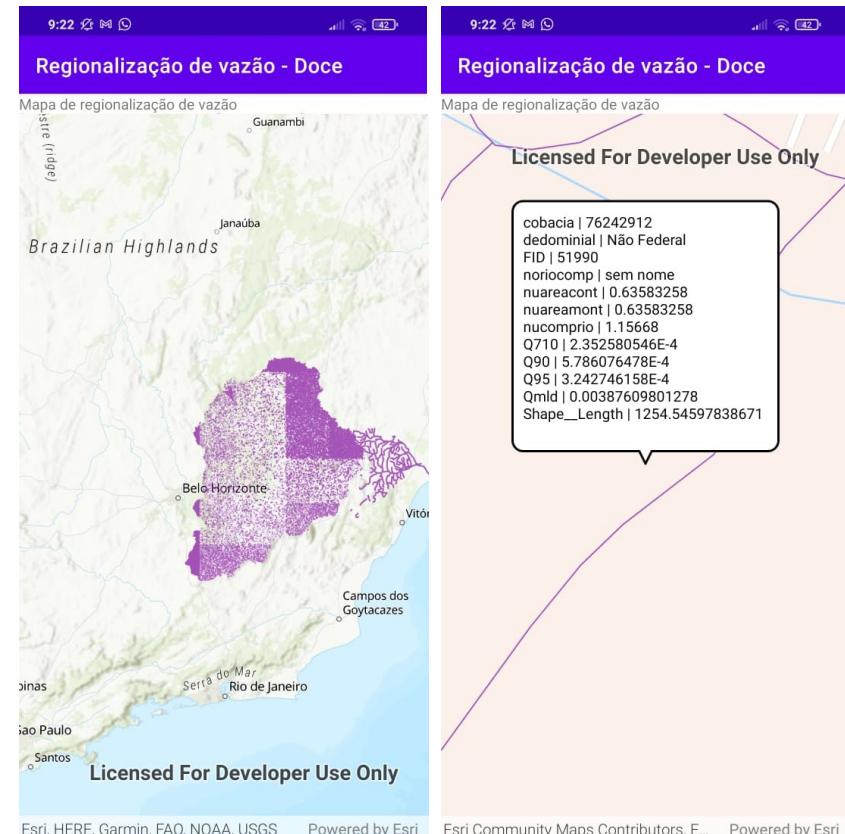
Comentários finais

- Com o **hosted Layer** da ArcGis developers
 - Pode-se subir qualquer mapa e acessá-lo
 - Via celular ou website
- Lembre-se serviços como este:
 - **Não** fazem o mapa para você



Comentários finais

- Com o **hosted Layer** da ArcGis developers
 - Pode-se subir qualquer mapa e acessá-lo
 - Via celular ou website
- Lembre-se serviços como este:
 - **Não** fazem o mapa para você
- Para melhorar a visualização dos dados
 - Ou fazemos via programação
 - Ou alteramos o nome das colunas dentro do Qgis, ArcGIS, etc...



Comentários finais

- Com o **hosted Layer** da ArcGis developers
 - Pode-se subir qualquer mapa e acessá-lo
 - Via celular ou website
- Lembre-se serviços como este:
 - **Não** fazem o mapa para você
- Para melhorar a visualização dos dados
 - Ou fazemos via programação
 - Ou alteramos o nome das colunas dentro do Qgis, ArcGIS, etc...
- O céu é o limite
 - Acesse a documentação sempre. Enjoy it!

