SmartData Construcción

Documentación

Visualizaciones

SmartData Construcción
Documento producido por PostData
Agosto del 2019
CLHM

Introducción

Este documento tiene por objetivo, realizar una revisión de cómo se construye cada visualización a nivel del frontend (lo que ve el usuario).

Sobre el framework de visualización

Para la aplicación en el backend se escogió el lenguaje de programación Python, tanto Flask para el servidor, como Jupyter Notebook para el trabajo con los datos. Para el frontend se utilizó JavaScript en el entorno React, que aloja módulos de visualización desarrollados utilizando como base RechartsJS[1]. Esto permite que se pueda construir modularmente una arquitectura compleja pero altamente personalisable.

Para SmartData se han construido 7 módulos de visualización:

- KpiGRaph: que muestra cifras particulares en el tiempo.
- TimeSerie: que muestra una serie de tiempo con o sin recorte territorial.
- TreemapGraph: muestra como se distribuyen cantidades en un total para un momento dato.
- RegionTreemapGraph: muestra como se distribuyen cantidades en el tiempo y en el territorio.
- StackedGraph: muestra como se distribuyen cantidades en un total para un momento dato.
- RegionStackedGraph: muestra como se distribuyen cantidades en el tiempo y en el territorio.
- MapGraph: muestra la distribución espacial de casos.

Cada uno de estos gráficos son alimentados por dos fuentes principales:

1) Jupyter Notebooks: cómo se menciona en ‘Documentación: procesamiento de datos’, cada dataset para cada indicador, viene configurado de forma automática para ser leído por los módulos de visuailización. Por ende basta con aplicar ‘Run All Cells’ y los datos ya poseen el formato adecuado.

[1] Ver http://recharts.org/en-US/

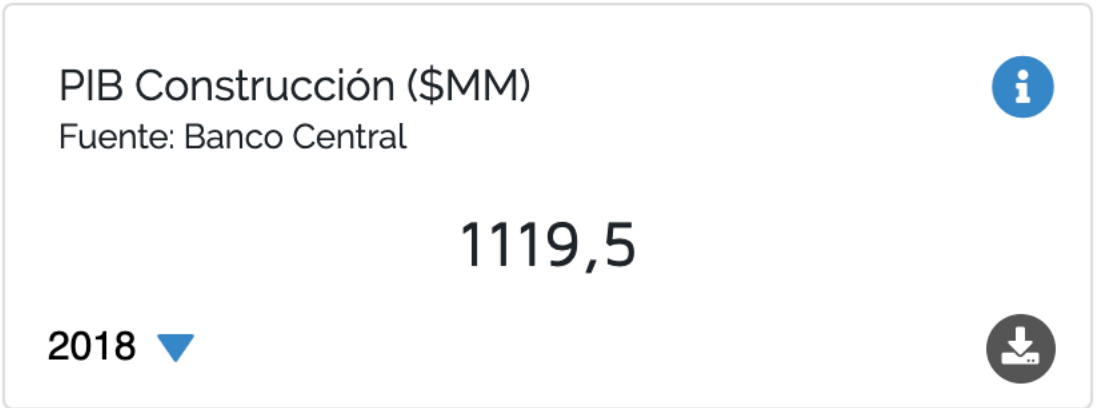
2) Files: existe un archivo particular en el backend que permite configurar el nombre con el cual cada modulo de visualización recibirá los datos en formato Json.

```
MapGraph.js   files.py
src > backend > smartdata > modules > api > config > files.py > ...
1   import os
2
3   DATA_S3_BUCKET = 'smartdata-demo'
4   DATA_S3_PATH = 'https://{}.sfo2.digitaloceanspaces.com/'.format(DATA_S3_BUCKET)
5   if os.environ['LOCAL_DATA_FILES']=="1":
6       DATA_S3_PATH = '/'
7
8
9   DATA_FILES = [
10      {
11          "name": "PIB_Construccion_MM",
12          "files": [
13              {"label": "PIB Construcción ($MM)", "name": "data/Destacado_01_PIB_Construccion.json" }
14          ]
15      },
```

Este archivo configura luego como las visualizaciones llaman a los datos. Basta con agregar ese nombre como variable en cada vista de sección y las visualizaciones cargan los datos.

```
<div className="col-lg-9">
    <div className="card-deck mb-3">
        <KpiGraph kpi_name='PIB_Construccion_MM'></KpiGraph>
        <KpiGraph kpi_name='PIB_Construccion_PER'></KpiGraph>
    </div>
```

De la forma que muestra la imagen anterior, se construye el KPI del PIB de la Construcción:



Cada módulo trabaja de la misma forma. Todos poseen un props[2] que toma el nombre del Dataset, el nombre del tipo de modulo de visualización y construye la visualización.

Subir archivo a Digital Ocean + Darle permiso Everyone can READ + Darle nombre en Files + Llamar datos y Módulo = Visualización.

Las vistas para los módulos de visualización utilizados en esta versión de SmartData, son:

```
<KpiGraph kpi_name='PIB_Construccion_MM'></KpiGraph>
<TimeSerie kpi_name='Permisos_Edificacion_Acumulados_m2'></TimeSerie>
<StackedGraph kpi_name='CEV1' isAllYears={true}></StackedGraph>
<RegionStackedGraph kpi_name='Participacion_regional_MM_total_pais_anio'></RegionStackedGraph>
<MapGraph kpi_name='LEED_CES'></MapGraph>
```

Cada una de estas vistas posee los diseños y la información adicional para cada módulo, desde el procesamiento de los datos[3].

[2] Ver https://reactjs.org/docs/components-and-props.html

[3] Ver Documentación: Procesamiento de los Datos