

# Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Branch: master

datosDataton2019 / README.md

Find fileCopy path

mariotorres Update README.md

6e99a32 · 15 hours ago

2 contributors

117 lines (85 sloc) | 5.37 KB

RawBlameHistory

# Dataton2019 - Datos

Las bases de datos que estarán disponibles para este año serán las de tres de los seis sistemas de la [Plataforma Digital Nacional](#):

- Sistema 2 (S2): Sistema de los Servidores públicos que intervengan en procedimientos de contrataciones públicas
- Sistema 3 (S3): Sistema nacional de Servidores públicos y particulares sancionados
- Sistema 6 (S6): Sistema de Información Pública de Contrataciones (incluyendo EDCA y sus extensiones)

## Servidores públicos que intervienen en contrataciones (S2)

Nombre del archivo	Formato	Descripción
s2-bulk.zip	JSON Array	

## Servidores públicos y particulares sancionados (S3)

Nombre del archivo	Formato	Descripción
s2-servidores-bulk.zip	JSON Array	Servidores públicos sancionados
s2-particulares-bulk.zip	JSON Array	Particulares sancionados

## Contrataciones (S6)

Son los datos abiertos de las compras públicas del Gobierno Federal en formato OCDS, [Open Contracting Data Standard](#) (A.K.A [EDCA](#) = Estándar de Contrataciones Abiertas).

### Fuente de datos

Plataforma Digital Nacional: <https://plataformadigitalnacional.org/contrataciones>

### Periodo comprendido

2017-2019

### Formato abierto de los datos

Json

### Datos

Primero se partió del conjunto de datos de contrataciones (S6). Este conjunto de datos tiene cientos de variables, las cuales corresponden al formato OCDS pero en su implementación en México.

Nombre del archivo	Formato	Descripción
contrataciones_arr.json.zip	JSON Array	Datos de contrataciones públicas, organizados conforme al estandar <a href="#">OCDS</a> (A.K.A. "EDCA")

### Construcción de subconjuntos de Datos

Para poder ejemplificar el uso de grafos en los datos de contrataciones, se extrajeron del conjunto original los **Instituciones compradores** (buyers) y los **Proveedores/contratistas** (suppliers) para crear dos subconjuntos de datos adicionales.

Finalmente se creó otro subconjunto de datos adicional usando las siguientes variables:

- ocid
- buyer
- parties
- tender.procurementMethod
- tender.title
- contracts

Los subconjuntos generados son los siguientes:

Archivo	Formato	Descripción
buyers.json.zip	JSON Array	Instituciones compradoras
tenderers_suppliers.json.zip	JSON Array	Proveedores/contratistas
cp.json.zip	JSON Array	Varias variables de contrataciones_arr.json.zip

### Importación de Datos

Instrucciones para importar los datos en una instancia local de [MongoDB](#):

```
# Descomprime los datos:
unzip datos_ejemplo.json.zip

# Importa los datos a MongoDB:
mongoimport -d dataton2019 -c datos_ejemplo --jsonArray datos_ejemplo.json
```

### Consultas básicas de los Datos

Consultas básicas en MongoDB (línea de comandos):

```
# Ingresa a la línea de comandos de MongoDB:
mongo

// Seleccionar base de datos dataton2019
use dataton2019;

// Contar registros
db.datos_ejemplo.count();

// Buscar y mostrar el primer registro en la colección
db.datos_ejemplo.findOne();
```

### Fuente

### Construcción de una Base de datos de grafos

El objetivo de este Datatón es usar grafos para modelar los datos de contrataciones públicas, de servidores públicos que intervengan en procedimientos de contrataciones públicas y de servidores públicos y particulares sancionado con el fin de encontrar posibles redes de corrupción.

Para hacer esto, se construyó una Base de datos usando [Neo4j](#) y generada a partir de los conjuntos de datos de contrataciones que mencionamos arriba.

Hacer esto permite trazar las relaciones entre las Instituciones de Gobierno (buyers), los procesos de contrataciones y los participantes (tenderers, suppliers, etc.).

Archivo	Formato	Comentarios
oc_graph.dump	Neo4j dump	Base de datos de grafos

Instrucciones para construir la base de datos:

```
# Ingresa al directorio de neo4j y detén la base de datos
$neo4j-home> bin/neo4j stop

# Restaura la base de datos
$neo4j-home> bin/neo4j-admin load --from=/path/to/oc_graph.dump --database=graph.db --force
```

A continuación se presenta una consulta a la base de datos para su representación gráfica:

```
$ MATCH p=(*)-->(*) RETURN p LIMIT 5000
```

¿Qué cosas interesantes puedes observar?