

## Ejercicio 1: Programar cliente REST para carga masiva de datos

Para la realización del ejercicio se tomó como base el ejemplo proporcionado, en donde se analizó la estructura del documento de carga y luego el fichero csv propuesto.

### Supuestos fichero de carga:

- Se consideró que el formato genérico de un fichero de carga se compone de dos columnas:
  - 1) Datetime
  - 2) Unidad de medida. Su valor determina si corresponde a una "electricityConsumption" cuando se informa kWh o "electricityKiloVolsAmpHours" en otros casos.
- Cada fichero informara mediciones de un tipo específico.

### Estructura Json de carga:

Dado que los únicos datos informados en el fichero de carga son la fecha/hora de medición y su valor, se optó por harcodear el resto de los datos en el documento.

Cada medición será reflejada en un nuevo documento embebido en la key "measurements"(lista de documentos).

```
amon_measures = {
    "measurements": [ Items a cargar ] ,
    "meteringPointId": "c1810810-0381-012d-25a8-0017f2cd8888",
    "readings":
    [
        {"type": "electricityConsumption", "period": "INSTANT", "unit": "kWh"}
    ],
    "deviceId": "c1810810-0381-012d-25a8-0017f2cd8888"
}

Estructura de los items a cargar
{
    "timestamp": Excel[datetime],
    "type": "electricityConsumption",
    "value": Excel[kWh]
}
```

### Características fichero de carga data.csv:

- Contiene las mediciones del 01/01/2013 al 31/12/2013.
- Cantidad de columnas: 2 ( Datetime, kWh)
- Cantidad de filas: 8.760

Analizando el fichero se detectó que hay 481 casos de filas con kWh nulo. Por lo tanto, la **cantidad de registros a cargar** es de **8.279**. En una etapa posterior se debería analizar el motivo por el cual estos valores no fueron informados o si representan algún tipo de error.

## Proceso carga de datos

El proceso de carga comprende los siguientes pasos:

- 1) Leer fichero csv.
- 2) Identificar el tipo de medición y lectura a partir de la unidad de medida especificada en el nombre de la segunda columna del csv.
- 3) Definir la estructura del documento amon\_measures\_in.
- 4) Leer cada una de las filas del fichero, validando que tanto la fecha con la medición estén informados.
- 5) Generar el documento "measure" para cada medición y agregarlo en la lista de la key "measurements".
- 6) Informar el tamaño del documento amon\_measures\_in (a modo de control).
- 7) Informar la cantidad de filas leídas y cargadas del fichero (a modo de control).
- 8) Invocar POST con el documento amon\_measures\_in .
- 9) Informar duración del POST ( a modo de control).

A continuación se presenta como evidencia las capturas con los resultados obtenidos:

### PUSH datos fichero csv

```
C:\Users\Carolina\Desktop\BeeData\Desa>python push_file_measures_cs.py
Total size of the document:820069
Read rows:8762
Load rows:8279
<Response [200]>
Start:Mon Aug 26 13:33:05 2019 End:Mon Aug 26 13:33:11 2019
```

- Duración de la ejecución: 6 segundos.
- Cantidad de registros cargados concuerdan con los detallados.

### Ejecución GET

\_Cantidad de registros antes de la inserción: 33.215

```
{...}
ks': {'self': {'href': 'amon_measures_measurements', 'title': 'amon_measures_measu
measurements'}, 'last': {'href': 'amon_measures_measurements?page=167', 'title': 'l
ast page'}, 'parent': {'href': '/', 'title': 'home'}, 'next': {'href': 'amon_mea
sures_measurements?page=2', 'title': 'next page'}}}, '_meta': {'max_results': 200
, 'total': 33215, 'page': 1}}
Inicio:Mon Aug 26 13:32:33 2019 Fin:Mon Aug 26 13:32:35 2019
```

Luego: 41.494 (8.279 registros insertados)

```
{...}
', 'title': 'amon_measures_measurements'}, 'last': {'href': 'amon_measures_measu
rements?page=208', 'title': 'last page'}, 'parent': {'href': '/', 'title': 'home
'}, 'next': {'href': 'amon_measures_measurements?page=2', 'title': 'next page'}}
, '_meta': {'max_results': 200, 'total': 41494, 'page': 1}}
Inicio:Mon Aug 26 13:35:31 2019 Fin:Mon Aug 26 13:35:33 2019
```

**Código entregado: push\_file\_measures\_cs.py**