# **INFORME EJECUTIVO**

## **DATA PIPELINE PARA DATOS DE TRANSFERENCIA PORTUARIA Y JSON**

El presente informe resume el proceso de elaboración de un data pipeline, conectándose al API REST disponibilizado por el Observatorio Logístico[[1]](#footnote-1).

Para el desarrollo de este, se utilizó Python 3, por medio del IDE Spyder, a través del sistema Anaconda. Los *packages* necesarios se detallan a continuación. Éstos vienen por defecto en el sistema Anaconda y su gestor de *packages*.

* requests
* simplejson
* pandas
* csv

Para efectos de evaluar conocimiento, se realizaron las siguientes funciones:

1. Realizar un *request* de la data a procesar en formato json.
2. Transformar json a un *Data Frame*, para posteriores análisis.
3. Realizar *data cleaning*.
4. Realizar funciones de agregación de datos.
5. Generar json de data agregada con metadata y csv de data *raw* para realizar posteriores análisis solicitados.

Hay que destacar que se detectaron ciertas irregularidades, las cuales fueron abordadas en diferentes niveles.

Primero, se revisó la data obtenida por la API y la presente en los reportes anuales de la DIRECTEMAR, y hay diferencias de valores. Como ejemplo se puede citar el puerto de Punta Patache en sus transferencias del año 2016 (presentadas en el Informe del 2017 en el caso de DIRECTEMAR). Al comparar las fuentes de información, se detectó que los datos de cabotaje no están registrados en la API, lo que genera una diferencia anual cercana a las 170.000 toneladas. Se requiere hacer un análisis más exhaustivo de las diferencias de información.

Por otro lado, están registrados, existían datos cuyo campo de “REGIÓN” tenía valor 0, los que correspondían a compra de naves. Estos registros se dejaron, para ser eliminados en el caso de requerirse.

Por último, se observó que los campos vacíos de la data entregada por la API, la metadata los define como texto. Esto se corrigió en el código y el posterior json proporcionado.

Para el producto final, se utilizó el mismo estándar de formato json proporcionado por la API, consultando también la documentación de Junar. El json obtenido por el código de Python se encuentra en la misma carpeta bajo el nombre “toneladas\_transferencia\_agregado.json”. En la [Sección 4](https://junar.github.io/docs/es/_sections/04-vistas.html#ejemplos-de-salida-de-los-datos) de la documentación de la API se puede ver un ejemplo del archivo. También puede encontrar un ejemplo en la carpeta bajo en nombre “toneladas\_transferencia\_agregado\_ejemplo.txt”.

También se generará un archivo csv para poder realizar análisis de la información.

Ambos archivos son generados automáticamente en el mismo directorio en donde se encuentra el código ejecutable. Tan solo se debe ejecutar el archivo para que funione.

1. https://datos.observatoriologistico.cl [↑](#footnote-ref-1)