

REUSO GASTO PAPEL EN EL SECTOR PÚBLICO

Resumen:

En este desarrollo, se implementó un sistema para obtener datos relacionados con las órdenes de compra de organismos registrados en ChileCompra que tienen gastos en papel de impresión. Se utilizó la API pública de Mercado Público para obtener los códigos de los organismos, así como para realizar consultas de órdenes de compra en un rango de fechas determinadas. El proceso incluyó el uso de Python en un entorno Jupyter Notebook, donde se aplicaron filtros de palabras clave y se gestionó la obtención de datos con métodos optimizados para manejar la gran cantidad de consultas. Los datos recopilados se utilizaron para generar un conjunto de datos filtrado, que luego fue visualizado en un dashboard en Tableau con diferentes vistas y análisis geográficos.

Objetivos

1. Obtener información de gasto de papel para impresión: Recopilar datos de órdenes de compra relacionadas con papel de impresión de los organismos registrados en ChileCompra.
2. Optimizar la recolección de datos: Implementar un sistema de consultas eficientes utilizando la API de Mercado Público, aplicando filtros de palabras clave y gestionando adecuadamente las múltiples solicitudes a través de reintentos controlados.
3. Filtrar y organizar los datos: Aplicar un conjunto de filtros a las órdenes de compra para seleccionar solo aquellas relacionadas con papel, descartando otros tipos de materiales, y generar un conjunto de datos estructurado.
4. Visualización interactiva: Crear un dashboard en Python (Dash) y Tableau que permita visualizar los gastos en papel de los distintos organismos, con análisis geográficos y temporales.

Implementación

Se obtuvo un conjunto de datos derivados de las órdenes de compra de los 900 organismos registrados en ChileCompra filtrando aquellos gastos relacionados con papel de impresión.

Ticket utilizado: "F8537A18-6766-4DEF-9E59-426B4FEE2844"

Endpoint para la obtención de los códigos:

<https://api.mercadopublico.cl/servicios/v1/Publico/Empresas/BuscarComprador?ticket=F8537A18-6766-4DEF-9E59-426B4FEE2844>

Captura de su visualización en la web

Pretty-print ☒

```
{
  "Cantidad": 900,
  "FechaCreacion": "2024-09-03T23:05:10.9376388Z",
  "listaEmpresas": [
    {
      "CodigoEmpresa": "7086",
      "NombreEmpresa": "AGENCIA CHILENA DE COOPERACION INTERNACIONAL PARA EL DESARROLLO"
    },
    {
      "CodigoEmpresa": "1224636",
      "NombreEmpresa": "AGENCIA CHILENA DE EFICIENCIA ENERGETICA"
    },
    {
      "CodigoEmpresa": "7193",
      "NombreEmpresa": "AGENCIA DE PROMOCION DE LA INVERSION EXTRANJERA"
    },
    {
      "CodigoEmpresa": "7212",
      "NombreEmpresa": "Agencia Nacional de Inteligencia"
    },
    {
      "CodigoEmpresa": "1824441",
      "NombreEmpresa": "ASOCIACION DE MUNICIPIOS METROPOLITANOS PARA LA SEGURIDAD CIUDADANA"
    },
    {
      "CodigoEmpresa": "1806837",
      "NombreEmpresa": "Asociación para el desarrollo productivo del turismo Alto del Carmen"
    }
  ],
}
```

Explicación del código desarrollado y sus funciones:

Nombre del script: dataset_zero_paper.ipynb

Para este desarrollo se utilizó Jupyter Notebook en Python 3, que es una aplicación web la cual permite crear y compartir documentos con código, texto y visualizaciones, ideal para análisis de datos y documentación.

Para ejecutarlo, se necesitan los siguientes datos de entrada:

```
keywords = ["papel", "resma", "imprimir", "impresión", "fotocopia", "carta",  
"oficio", "impresora"] # Lista de palabras clave a buscar en el nombre de la  
orden de compra para el filtro.
```

```
csv_name = "dataset_2024.csv" # Nombre del archivo CSV a generar.
```

```
tini = "22062024" # Fecha de inicio de búsqueda.
```

```
tend = "25062024" # Fecha fin de búsqueda.
```

```
ticket = "F8537A18-6766-4DEF-9E59-426B4FEE2844"
```

```
endpoint_institutions =
```

```
"https://api.mercadopublico.cl/servicios/v1/Publico/Empresas/BuscarComprador  
?" # Endpoint para búsqueda por códigos de organismos (son 900).
```

```
endpoint_oc =
```

```
"https://api.mercadopublico.cl/servicios/v1/Publico/ordenesDeCompra.json?" #  
Endpoint para obtener las órdenes de compra.
```

Comentarios:

El formato de las fechas de inicio y fin de búsqueda debe ser en este formato ddmmyyyy.

"22062024" representa una fecha en un estilo compacto y sin separadores.

Significado:

22 representa el día del mes.

06 representa el mes del año.

2024 representa el año completo.

Así que "22062024" corresponde al 22 de junio de 2024.

Este formato debe respetarse para que las consultas se realicen con éxito.

Para más información, visita la siguiente URL de ChileCompra:

<https://api.mercadopublico.cl/modules/api.aspx>, donde se puede utilizar el ticket de prueba o solicitar uno haciendo clic en el botón naranja llamado “¡¡¡Participa!!!”.



Luego se genera el rango de fechas con la función: **def generate_dates(tini, tend)**, la cual genera las fechas entre la fecha inicial y final ingresadas para la búsqueda de las órdenes de compra.

Se realiza una búsqueda por cada organismo en las fechas generadas para seleccionar las órdenes de compra que se encuentren en estado “aceptada” y que en su nombre contengan dos o más palabras clave para ser consideradas como órdenes de compra que incluyan papel.

Para obtener los datos asociados del servidor, se desarrolló la función: **def response(url, max_retries=2, delay=2)**, la cual tiene como parámetro la URL y dos parámetros estáticos adicionales. Estos se añadieron debido a los problemas de conexión con el servidor por la cantidad de consultas realizadas de forma simultánea.

Considerar que son 900 instituciones, y si queremos obtener una semana de datos, serían $900 \times 7 = 6,300$ consultas realizadas.

max_retries: Define el número máximo de intentos que la función realizará para hacer la solicitud a la URL en caso de errores. Si la solicitud falla (por ejemplo, debido a un error de red o un problema con el servidor), la función intentará realizar la solicitud nuevamente hasta max_retries veces, en este caso, 2.

delay: Especifica el tiempo de espera en segundos entre intentos fallidos. Si un intento falla, la función espera delay segundos antes de volver a intentar realizar la solicitud, en este caso, 2 segundos.

Cada registro seleccionado se guarda en un archivo CSV, el cual se actualiza continuamente con los nuevos registros mediante la siguiente función:

def csv(namefile, dataset) #Como parámetro se observa el nombre de archivo ingresado como dato de entrada y el conjunto de datos a guardar.

Los datos a guardar son los siguientes:

- Institución Padre #Nombre de organismo
- Dirección
- Comuna
- Región
- Coordenadas Geográficas
- Rubro de gasto #Nombre de la orden de compra
- Periodo #mes/año
- Gasto #Valor total de gasto

Para el **Periodo**, se desarrolló una función llamada **date_format(date)**, la cual recibe como parámetro la fecha de aceptación de la orden de compra y la transforma al formato de mes/año.

Por ejemplo, si la fecha ingresada es "2024-09-15", la función devolverá "Septiembre/2024".

Para obtener el campo **Coordenadas Geográficas**:

Se obtienen las coordenadas geográficas a partir de la dirección o el nombre de la institución con la función: **def latlong(inst, address, retries=5, backoff_factor=0.5)**
En esta función se emplea la librería geopy, que realiza intentos de consulta y aplica un backoff exponencial para manejar errores transitorios, como problemas de red o servidores sobrecargados.

Se observó que hay bastantes órdenes de compra sin dirección asociada. Se intentó resolver esto utilizando las librerías geocoder, requests con OpenStreetMap, geopy y urllib, pero sin éxito.

En algunos casos, donde la dirección existía pero no era encontrada, se guardaron las coordenadas como (0,0).

Los demás datos asociados se obtienen del detalle de la orden de compra una vez verificado que en el nombre se encuentran las palabras clave utilizadas en el filtro correspondiente. En este caso, se realiza otra consulta al servidor con el código de la orden de compra asociada.

Ejemplo:

<https://api.mercadopublico.cl/servicios/v1/Publico/ordenesDeCompra.json?codigo=3613-142-AG24&ticket=F8537A18-6766-4DEF-9E59-426B4FEE2844>

Donde se puede observar todos los datos relacionados a la orden de compra específica con su código correspondiente.

Una vez obtenidos todos los datos de los meses correspondientes de enero a octubre de 2024, se realizó una revisión manual y se desarrolló un notebook con un nuevo filtro para el CSV generado, donde se descartaron registros por tipos de papel entre otros para obtener resultados más precisos.

El notebook tiene por nombre: **analysis_with_filter_keywords.ipynb**

De esta manera se obtiene el conjunto de datos filtrado y listo para ser utilizado en la creación del dashboard en Tableau.

Se desarrolló un notebook llamado **create_json_for_dashboard.ipynb**, en el cual se obtiene un archivo JSON con el ranking top 10, tanto en modo ascendente como descendente, de instituciones y municipalidades con gastos de papel. Este ranking se organiza por año, e incluye el gasto asociado por mes y sus coordenadas geográficas. El propósito de este notebook es generar datos limpios que puedan ser utilizados directamente para el desarrollo del dashboard.

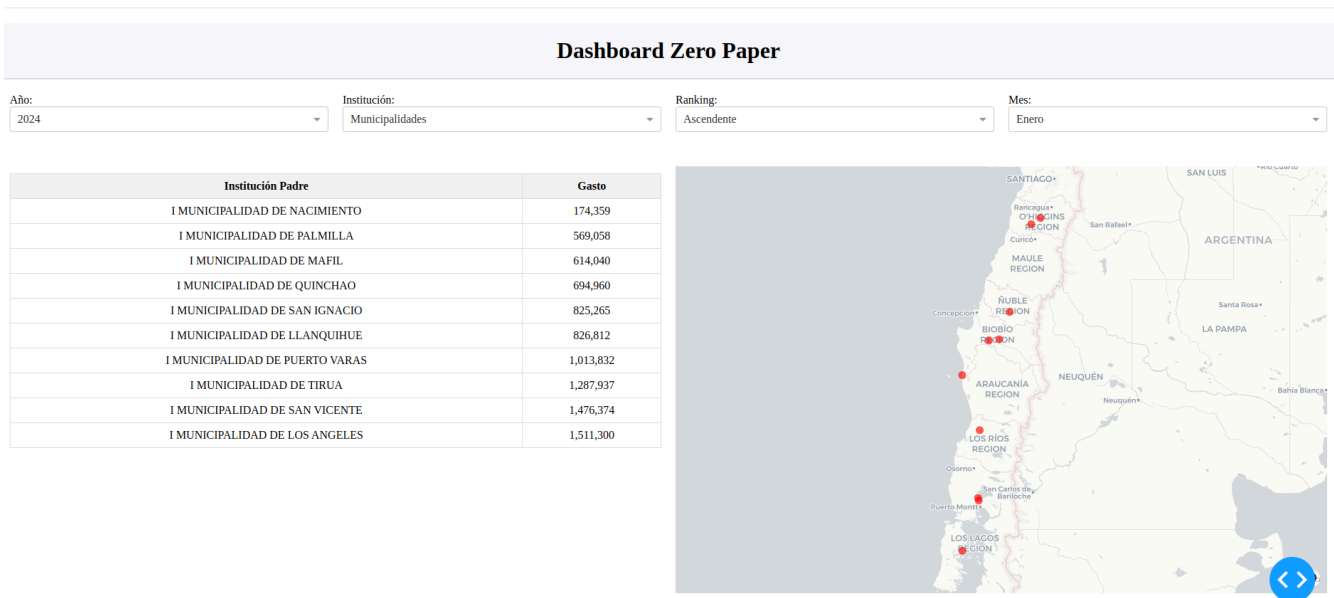
El notebook que genera el dashboard se llama **dashboard_zero_paper.ipynb**. Este notebook, utilizando el archivo JSON obtenido y la librería Dash en Python, permite visualizar lo siguiente:

- Dropdowns para la selección de año, institución, ranking y mes.
- Una tabla que muestra las 10 instituciones de acuerdo con los valores seleccionados en los dropdowns.
- Un mapa asociado a las coordenadas de las instituciones mostradas en la tabla, con una posición fija en Chile para mejorar su visualización.

Los valores de gastos mostrados corresponden a la suma de los gastos mensuales de cada institución.

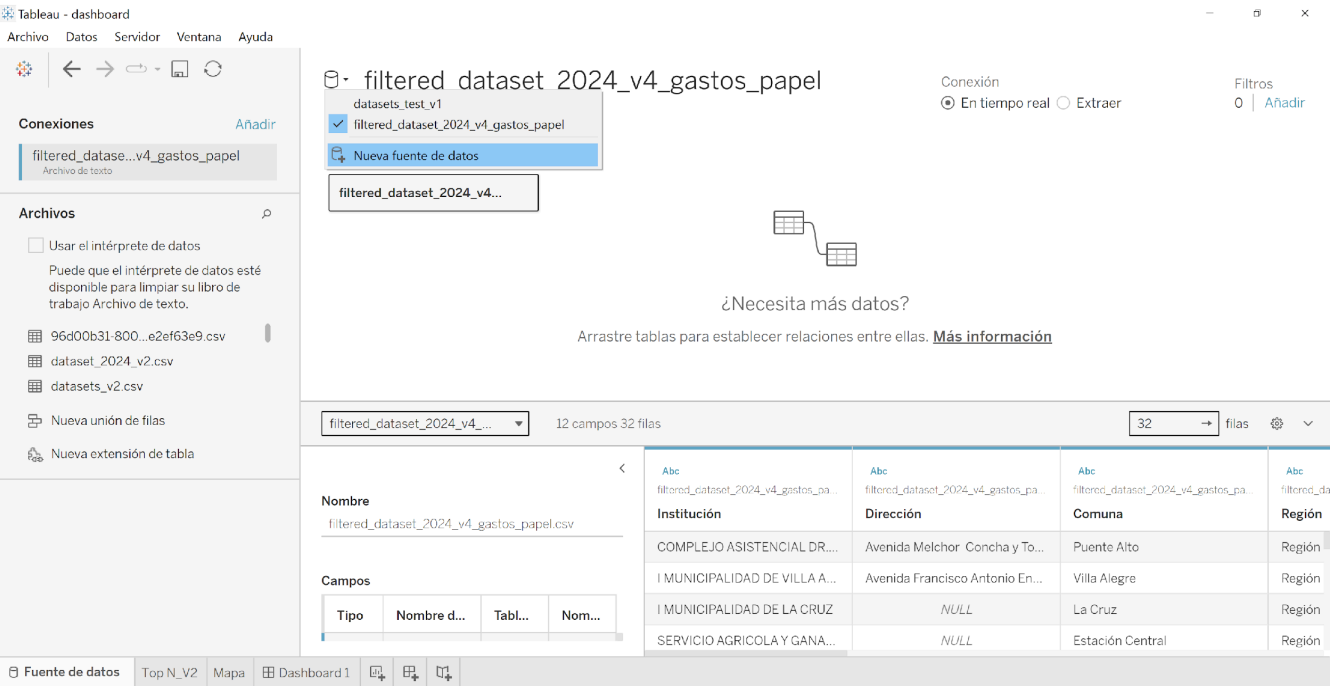
Se aplica un filtro para que aquellas instituciones cuya suma de gastos sea menor a 1,000 no aparezcan en la tabla, con el fin de obtener datos más precisos y representativos.

Imagen respectiva al dashboard generado



Dashboard en Tableau:

Se creó un libro de trabajo con el archivo csv filtrado como fuente de datos.



Luego se crearon 3 vistas:

La primera (Top N_V2): Muestra una selección por el tipo de institución la cual puede ser:

- Municipales
- Otros

Además de la selección de orden ascendente o descendente para mostrar el orden de los gastos de papel de 10 instituciones.

Imagen respectiva

Columnas		
Filas		
Institución		
Ranking		
Institución	Ranking	
I MUNICIPALIDAD DE VILLA ALEGRE	1	3.824.125
I MUNICIPALIDAD DE CHIMBARONGO	2	1.681.470
I MUNICIPALIDAD DE NANCAGUA	3	1.200.382
I MUNICIPALIDAD DE SAN ROSENDO	4	1.101.345
I MUNICIPALIDAD DE PENALOLEN	5	684.488
I MUNICIPALIDAD DE LONQUIMAY	6	610.560
I MUNICIPALIDAD DE TALCA	7	599.881
I MUNICIPALIDAD DE RANCAGUA	8	464.100
I MUNICIPALIDAD DE QUINTERO	9	456.960
I MUNICIPALIDAD DE PICHIDEGUA	10	406.980

Ranking top 10

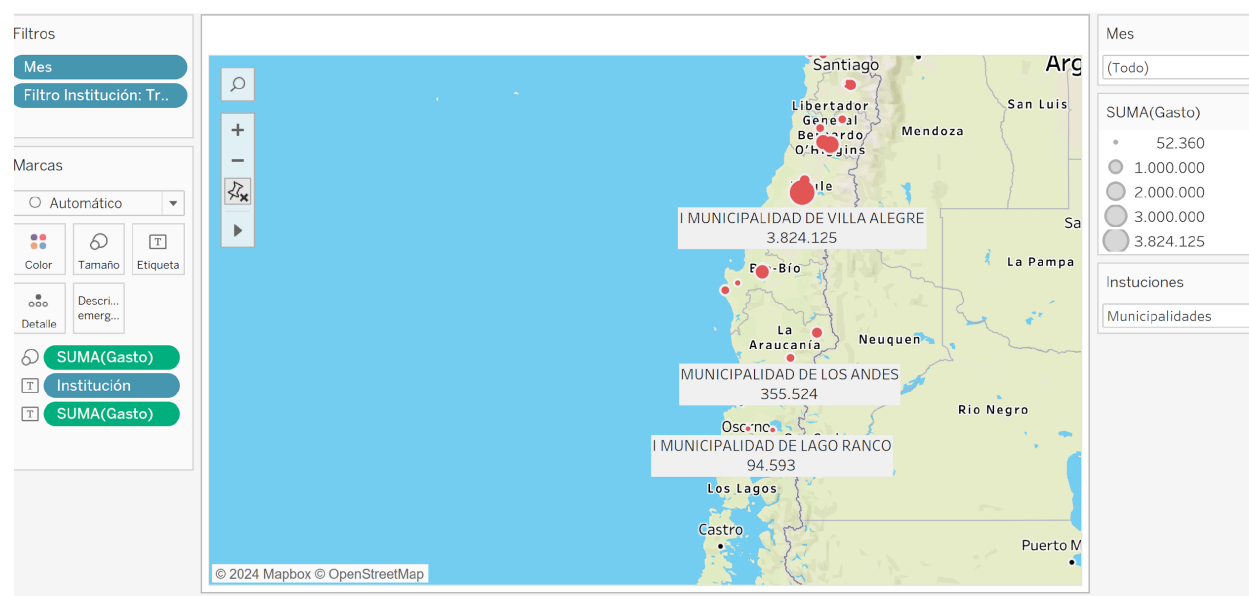
Descendente

Instuciones

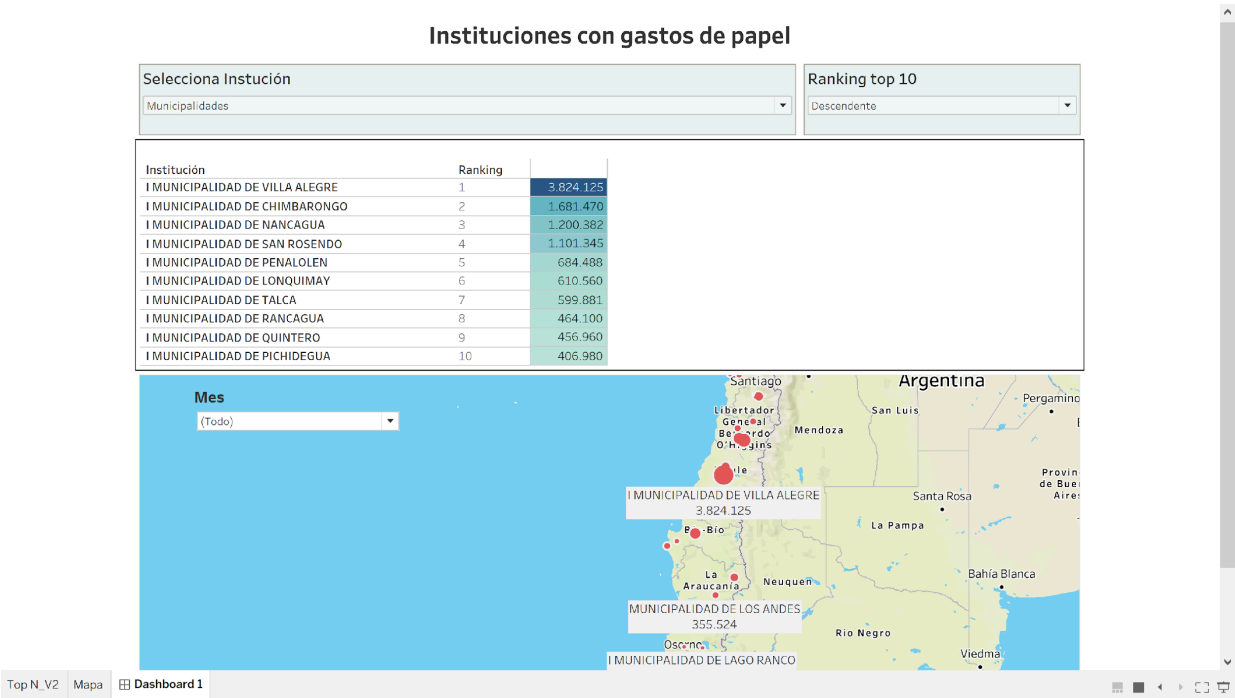
Municipalidades

En la segunda vista, denominada **Mapa**, se incluye un selector de mes que muestra los meses disponibles en los datos del archivo CSV. Según la selección realizada, el mapa muestra las coordenadas geográficas correspondientes a las instituciones asociadas a ese mes.

Imagen respectiva



Finalmente, se creó una tercera vista, que es un **dashboard** que integra todo lo mencionado anteriormente



Comentarios:

- Se observan nombres bastante variados en las órdenes de compra relacionadas con gastos de papel, ya que el nombre asociado a cada orden, al ser ingresado manualmente, puede presentarse de diversas formas o estilos de escritura.
- No existe un formato específico para los nombres de las municipalidades, por lo que pueden aparecer de diferentes maneras (por ejemplo: "Municipalidad", "Ilustre Municipalidad", "I Municipalidad", "I. Municipalidad").
- Existen bastantes registros ingresados sin dirección.
- No se dispone del valor total asociado a cada ítem dentro del nombre de la orden de compra. Por lo tanto, si existen dos o más palabras clave que indican un gasto en papel de impresión, se toma el valor total de la orden, aunque esta incluya otros ítems no relacionados, ya que en la orden de compra solo se muestra el valor total.
- Si tuviéramos un endpoint para obtener los datos por institución dentro de un rango de fechas, la obtención de datos sería mucho más rápida, ya que actualmente solo se cuenta con un endpoint que permite la consulta por día.
- Los errores asociados a las búsquedas se deben a que no se encuentran órdenes de compra en la fecha solicitada, lo que provocará que se muestren los intentos fallidos, además de errores por peticiones simultáneas. Debido a esto, el tiempo de ejecución es considerablemente largo.
- Se consideró el uso de hilos y de Curl para la obtención de datos. Con el uso de hilos, se obtenía una menor cantidad de datos y se presentaban errores recurrentes debido a las peticiones simultáneas, por lo que se descartó esa opción. En cambio, con Curl, el tiempo de ejecución fue similar y la obtención de datos no mejoró.

Ejemplo de error en Jupyter Notebook:

Attempt -> 2

Error 500: Unable to retrieve the information for url->

<https://api.mercadopublico.cl/servicios/v1/Publico/ordenesDeCompra.json?fecha=02012024&estado=aceptada&CodigoOrganismo=7086&ticket=F8537A18-6766-4DEF-9E59-426B4FEE2844>.

Si tiene alguna duda puede realizar una consulta desde el navegador, sólo necesita el código del organismo por el que desea consultar y aparecerán las órdenes de compra asociadas a la fecha ingresada.

Ejemplo :

Institución: I MUNICIPALIDAD DE CONCEPCION

Código: 87382

Desde el navegador:

<https://api.mercadopublico.cl/servicios/v1/Publico/ordenesDeCompra.json?fecha=09012024&estado=aceptada&CodigoOrganismo=87382&ticket=F8537A18-6766-4DEF-9E59-426B4FEE2844>

Con cUrl desde una terminal:

curl -X GET

"<https://api.mercadopublico.cl/servicios/v1/Publico/ordenesDeCompra.json?fecha=09012024&estado=aceptada&CodigoOrganismo=87382&ticket=F8537A18-6766-4DEF-9E59-426B4FEE2844>"

Impacto del reuso:

Este proyecto proporciona una herramienta poderosa para mejorar la visibilidad y el control sobre los gastos en papel de impresión realizados por organismos públicos en Chile. La automatización de la recolección y filtrado de datos no solo optimiza el proceso de obtención de información, sino que también permite identificar oportunidades de reducción de costos y mejorar la eficiencia en la gestión de compras. Al facilitar el análisis visual de los datos mediante un dashboard interactivo, se potencia la capacidad de los tomadores de decisiones para identificar tendencias y tomar medidas basadas en evidencia. Además, este enfoque puede replicarse para otros tipos de productos y servicios, incrementando el potencial de análisis en diferentes áreas del gasto público.