# Toolboxes ArcGIS Pro

*Departamento de Ciencia de Datos – Sello Rojo*

*Última Actualización –> 28/08/2023*

## Introducción

En el vasto mundo de los sistemas de información geográfica, la eficiencia y simplicidad en la gestión de datos geoespaciales es esencial. Presentamos un toolbox diseñado para ArcGIS Pro, compuesto por siete scripts de Python, cuya finalidad es optimizar y simplificar el manejo de información geoespacial. Estas herramientas han sido desarrolladas pensando en el usuario, facilitando procesos complejos y maximizando la eficiencia en cada tarea.

## Índice de los Archivos

* Actualizacion\_Secuencia.py
* Agregar\_Centro\_De\_Distribucion.py
* Optimizacion\_Agenda.py
* Optimizacion\_Agenda\_Espacial.py
* Sistema\_Recomendacion.py
* Transformacion\_De\_Formato.py
* Visualizacion\_Datos.py
* VRP\_ArcGis.py

## Objetivos

### Objetivo General:

Con este conjunto, se busca que la persona que haga uso del toolbox pueda enfocarse en el análisis y la interpretación, dejando la parte técnica y operativa en manos de las soluciones automatizadas, para así poder ahorrar tiempo y recursos.

### Objetivos Específicos:

* Automatizar los procesos geoespaciales de cada UNE.
* Crear un estándar para los procesos geoespaciales que se pueden hacer con el toolbox.

## Desarrollo

### Actualizacion\_Secuencia.py

Con este script se realiza una actualización de la secuencia de los clientes que están en cada UNE, arroja un Excel con la actualización optima generada por el proceso y el VRP de VROOM.

### Agregar\_Centro\_De\_Distribucion.py

En base a datos manuales que se le dan dentro de ArcGIS, en base a lo que se le da se agrega al mapa para poderlo visualizar y hacer uso de él en diferentes procesos.

### Optimizacion\_Agenda.py

Se genera una optimización de la agenda de las rutas (es decir de los días en los que se visita a cada cliente en base a su frecuencia. Este proceso se realiza de manera aleatoria realizando una cierta cantidad de iteraciones buscando minimizar la desviación estándar entre días de cada ruta.

### Optimizacion\_Agenda\_Espacial.py

Se realiza una optimización de la agenda de cada ruta, haciendo que la cantidad de clientes por día se busque homogenizar de manera espacial, es decir, que estén cerca el uno del otro.

Este proceso se realiza con un proceso usando los datos geoespaciales de los clientes y un proceso de clustering y ciertas reglas para poder realizar el proceso necesario.

### Transformacion\_De\_Formato.py

Este proceso solo hace cambios en el formato de una capa para poder hacer uso de ella en otros procesos en caso de no ser compatible.

### Visualizacion\_Datos.py

Este script realiza el proceso de carga de datos desde la base de datos y busca mostrar los clientes y sus datos en el mapa para saber qué clientes tiene cada UNE.

Así mismo, se puede usar para procesos de Maestro de Clientes, ya que la tabla de atributos de la capa contiene todos los datos. Así mismo, arroja varios archivos csv en los que vienen dichos datos y el porcentaje de cada ruta y UNE de la cantidad de clientes que sí tienen coordenada.

### VRP\_ArcGis.py

Este proceso realiza el VRP usando la herramienta nativa de ArcGIS para este tipo de problemas en vez de usar la herramienta externa de VROOM. Arroja todas las capas con los resultados de dicho proceso dentro del mapa.

## Conclusiones y Discusiones

Todos estos procesos se realizaron para poder optimizar el tiempo de proceso para la optimización de las UNE y rutas, así como poder ver y manipular de forma más fácil los procesos geoespaciales que se puedan requerir.

El tiempo de procesamiento depende totalmente del hardware con el que se cuente, por lo que se recomienda contar con tiempo en caso de que no se cuente con un equipo de computo bueno, ya que casi todos los scripts se procesan de manera local.

Para el caso del proceso de *Actualizacion\_Secuencia.py* se necesita estar conectado a la red local de sello rojo, ya que hace uso del VRP de VROOM que se encuentra en una imagen dentro del servidor.

## Anexos

* Carpeta *Scripts*:
  + En esta carpeta se encuentran todos los archivos ‘.py’ en los que se encuentran todos los procesos que ejecuta el toolbox.
* *Requirements.txt*:
  + Archivo ‘.txt’ en el que están todas las librerías que se necesitan para el uso de los scripts.
* *Toolbox\_Sello\_Rojo.atbx*:
  + Archivo ‘.atbx’ en el cual se juntan todos los scripts; contiene los parámetros y metadata de cada proceso.