


Curso de

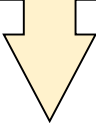
ParaView

05a. Filtración de los datos

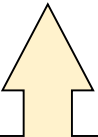
Michael Heredia Pérez
mherediap@unal.edu.co

Universidad Nacional de Colombia
Sede Manizales

La **visualización** puede caracterizarse como el proceso de transformar datos producidos de experimentos y simulaciones hasta el punto en que puedan ser **interpretados y analizados**.



Visualization can be characterized as a process of transforming raw data produced from experiments or simulations until it takes a form in which it can be interpreted and analysed. The visualization pipeline introduced in Section 1.2 formalizes this concept as a data flow paradigm where a pipeline is set up of sources, filters, and sinks (collectively called pipeline modules or algorithms). Data *flows* through this pipeline, being transformed at each node until it is in a form where it can be consumed by the sinks. In previous chapters, we saw how to ingest data into ParaView (Chapter 2) and how to display it in views (Chapter 4). If the data ingested into ParaView already has all the relevant attribute data, and it is in the form that can be directly represented in one the existing views, then that is all you would need. The true power of the visualization process, however, comes from leveraging the various visualization techniques such as slicing, contouring, clipping, etc., which are available as filters. In this chapter, we look at constructing pipelines to transform data using such filters.



El verdadero poder del proceso de visualización viene del aprovechamiento de las **diferentes técnicas de visualización** como *slicing, contouring, clipping*, etc, las cuales estás disponibles como filtros.

Entendiendo los filtros – *filters*

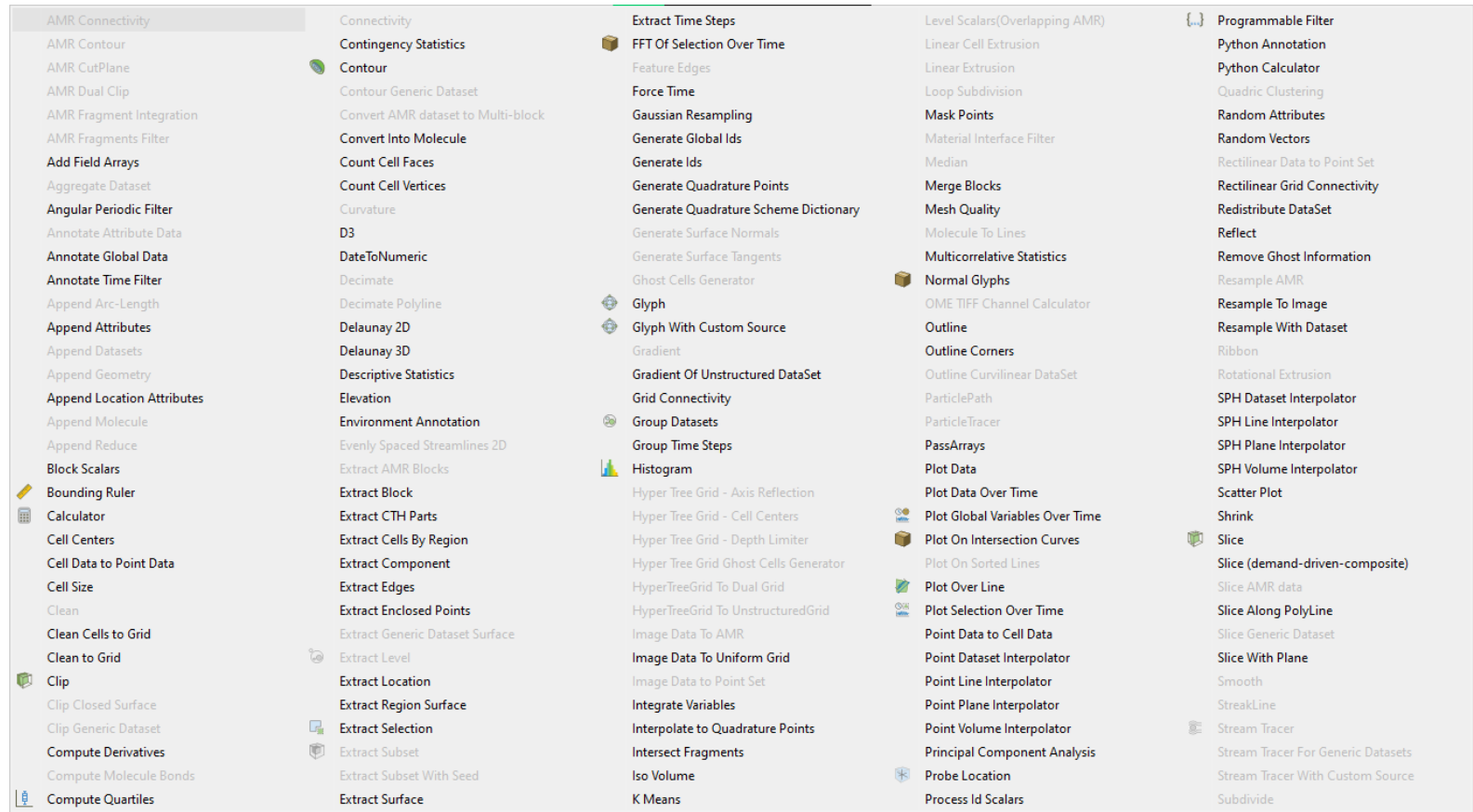
- ❖ Módulos que tienen un ingreso y salida de información.



- ❖ Tiene múltiples entradas y salidas y ambas cantidades son fijas, y cada input puede aceptar múltiples conexiones.
- ❖ Los filtros definen si una entrada de información puede aceptar una o varias entradas conectadas.
- ❖ Al igual que en los *readers*, las propiedades en el filtro nos permiten controlar el algoritmo de filtrado. Las propiedades disponibles dependen del filtro en sí.

Creando filtros en ParaView

Están disponibles en el **menú *Filters***, organizados y categorizados, y si no están activos es porque la fuente activa no produce información del tipo esperado para el filtro.



Conexiones

Múltiples conexiones

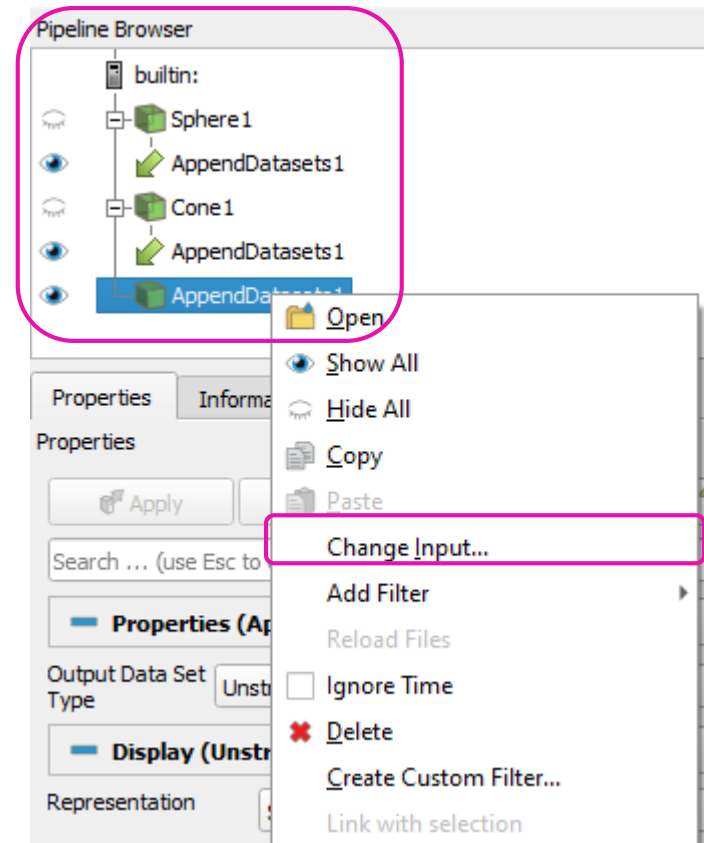
La fuente activa se conectará a la primera entrada del filtro. Para añadir a otras entradas se debe contar con el filtro *Append Datasets*.

Múltiples fuentes de entrada

Los filtros que solo presentan una fuente de entrada se deben de componer, por ejemplo, *Resample with Dataset*.

Cambiar las conexiones de entrada

Es posible modificar las fuentes de un filtro mediante la ventana *Change Inputs*.



Cambios en la función ResampleWithDataset()

A la función `ResampleWithData()` le cambiaron los argumentos de entrada desde la versión 5.7 (<https://blog.kitware.com/paraview-5-7-0-release-notes/>); sin embargo, el manual de la versión 5.8 the ParaViewGuide-5.8.1.pdf no está actualizada.

```
31 # Si miran en el manual de la versión 5.8.1, encontrarán que la función tiene
32 # los siguientes key arguments: INPUT y SOURCE.
33 remuestreo_con_datos = ResampleWithDataset(Input = esfera, Source = ondícula)
34
35 # Pero esto es una desactualización, ya que desde la versión 5.7 fueron cambiados a:
36 remuestreo_con_datos = ResampleWithDataset(SourceDataArrays = esfera, DestinationMesh=ondícula)
37
38 # Input --> SourceDataArrays
39 # Source --> DestinationMesh
```

Algunos comentarios



Cada filtro trae sus propias variables a modificar, y el procedimiento es el mismo que con los demás módulos: desde el **panel de propiedades**.



Tendremos cuatro grupos de filtros:

- Filtros para subgrupos de datos (*subsetting data*).
- Filtros para modificación geométrica (*geometric manipulation*).
- Filtros para muestreo (*sampling*).
- Filtros para manipulación de atributos (*attribute manipulation*).