

---



Curso de

# ***ParaView***

## 03a. Cómo entender los datos

Michael Heredia Pérez  
[mherediap@unal.edu.co](mailto:mherediap@unal.edu.co)

Universidad Nacional de Colombia  
Sede Manizales

# El modelo VTK

---

To use ParaView effectively, you need to understand the ParaView data model. ParaView uses VTK, the Visualization Toolkit, to provide the visualization and data processing model. This chapter briefly introduces the VTK data model used by ParaView. For more details, refer to one of the VTK books.

Para un correcto uso del programa es necesario entender cómo funciona el software, no simplemente cuáles botones oprimir.

## Data Object

Es el tipo de dato principal, puede ser tan sencillo como grillas o mallas de elementos finitos, o tan complejo como árboles o grafos.

Se compone de dos elementos

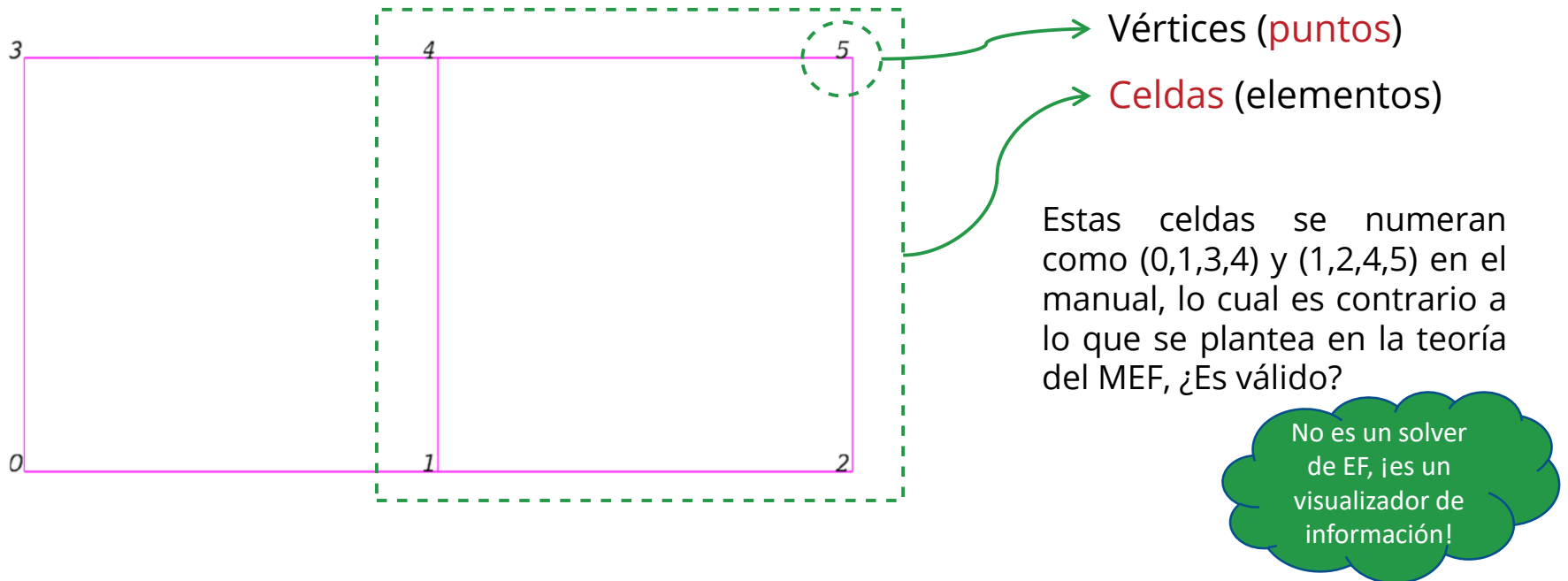
Mallas

Atributos

Topología

Geometría

# Sobre las Mallas



- Siendo los puntos y las celdas las estructuras de visualización, la información estará localizada allí.
- Se habla de **conectividad** como el mapeo que hay desde las celdas a los vértices.
- Se definen por la topología y las coordenadas espaciales de los vértices.

## Sobre los Atributos

---

Son **arreglos de información** que definen los valores discretos de un **campo** sobre la malla.

→ Velocidad, presión, esfuerzos, temperatura..... CUALQUIERA.

→ ¿Tienen forma? No, se asume dentro de ParaView según sus dimensiones.

Point-centered data



Atributos definidos en los puntos mediante interpolación según la documentación del formato VTK.

Cell-centered data

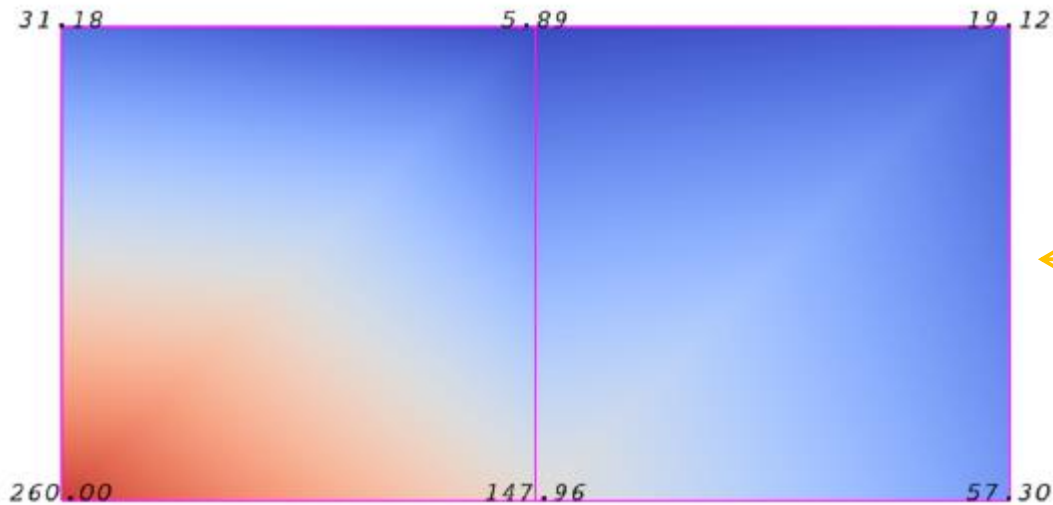


Atributos definidos en la celda constantes en ella.

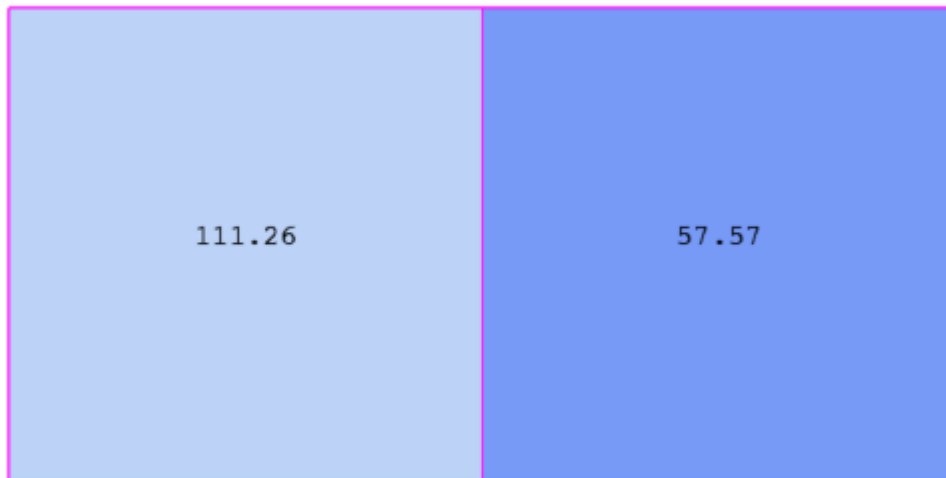
Se puede convertir de celda a punto mediante filtros

“Cell data to point data”  
filter

## Sobre los Atributos

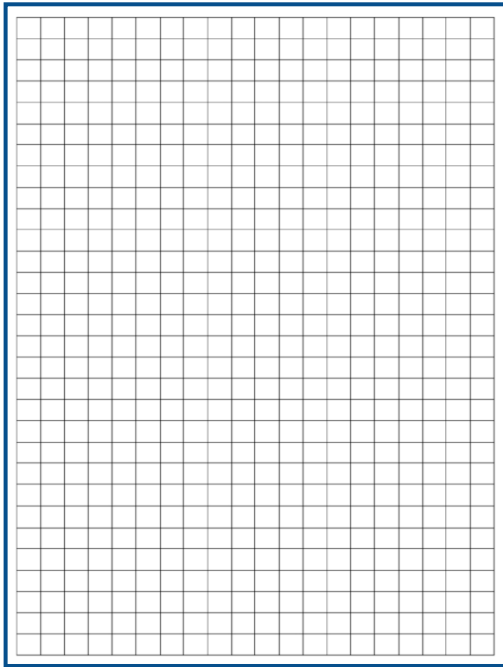


← Point-centered data



← Cell-centered data

# Grilla rectilínea uniforme



Uniform Rectilinear Grid o *Image Data*.

Las más eficientes en almacenamiento y con algoritmos más optimizados.

Sus celdas son del mismo tipo, el cual es definido por la dimensión del conjunto de datos (*extents*), y puede ser:

- Vértices (0D)
- Líneas (1D)
- Píxeles o *pixels* (2D)
- Volúmenes o *voxels* (3D)

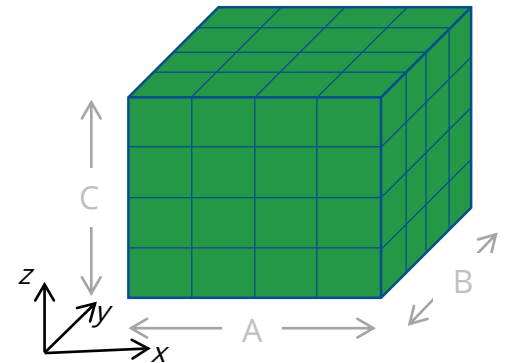
Topología y coordenadas implícitas.

Emplea:

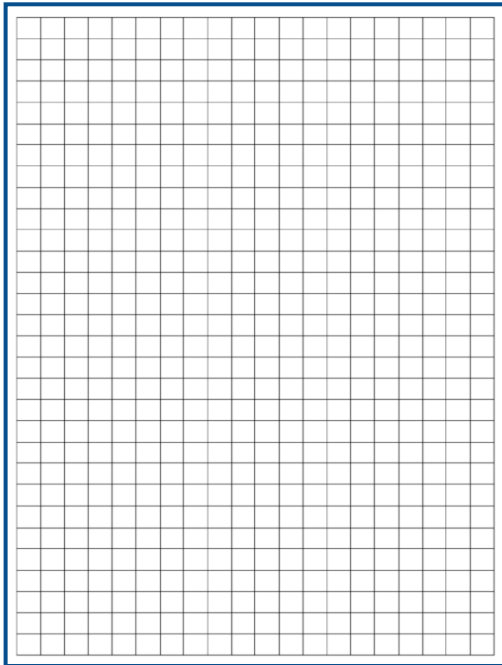
*Extents* Límites en cada dirección, i.e.  $(0, A_{n-1})(0, B_{n-1})(0, C_{n-1})$

*Origin* Posición del punto con índices  $(0,0,0)$

*Spacing* Distancia entre cada punto



# Grilla rectilínea uniforme



Uniform Rectilinear Grid o *Image Data*.

Las coordenadas se de la forma:

$$coord = origen + índice * distancia$$

$$idx\_flat = k * (npts_x * npts_y) + j * npts_x + i$$

El formato VTK permite el almacenamiento en índice plano.

Topología y coordenadas implícitas.

Emplea:

*Extents* Límites en cada dirección, i.e.  $(0, A_{n-1})(0, B_{n-1})(0, C_{n-1})$

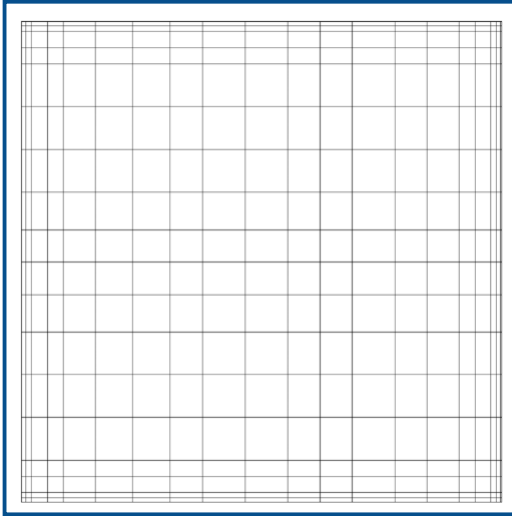
*Origin* Posición del punto con índices  $(0, 0, 0)$

*Spacing* Distancia entre cada punto

Índice plano en función de los índices  $(i, j, k)$ .

# Grilla rectilínea

---



*Rectilinear Grid.*



Topología implícita y coordenadas semi-implícitas.



Emplea:

- *Extents*
- Arrays de memoria en dirección  $x$ ,  $y$ ,  $z$ . para mejor ahorro computacional:

$$nptsx + nptsy + nptsz \neq nptsx * nptsy * nptsz$$



Coordenadas:  $coord = (coord\_array_x(i); coord\_array_y(j); coord\_array_z(k))$

$$idx\_flat = k(npts_x * npts_y) + j * npts_x + i$$

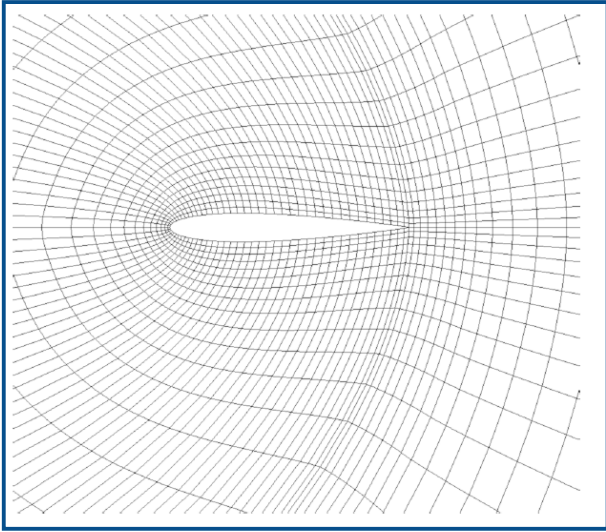


Celdas del mismo tipo, determinando por las dimensiones del conjunto de datos (*extents*), desde 0D hasta 3D.



# Grilla curvilinea

---



*Curvilinear grid.*



Topología implícita y coordenadas explícitas.



Emplea:

- *Extents*
- Arrays de puntos-coordenadas: posición de cada uno de los vertices de manera explícita



Coordenadas:  $coord = vector\_coord(idx\_flat)$

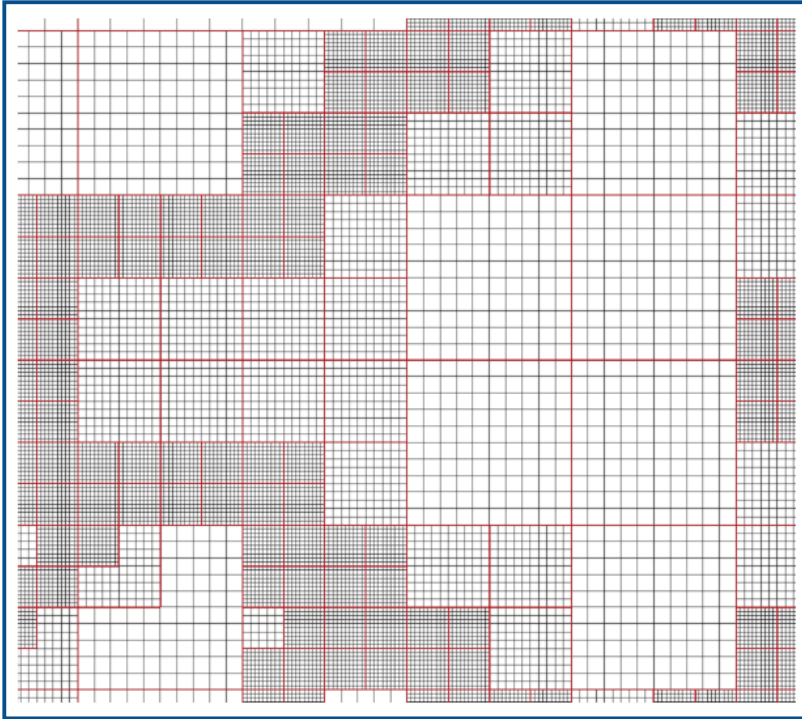
$$idx\_flat = k(npts_x * npts_y) + j * npts_x + i$$



Celdas del mismo tipo, determinando por las dimensiones del conjunto de datos (*extents*), desde 0D hasta 3D. 2D se llaman quad o cuádruples y 3D se llaman Hexaedros.

# Conjunto de datos RMA

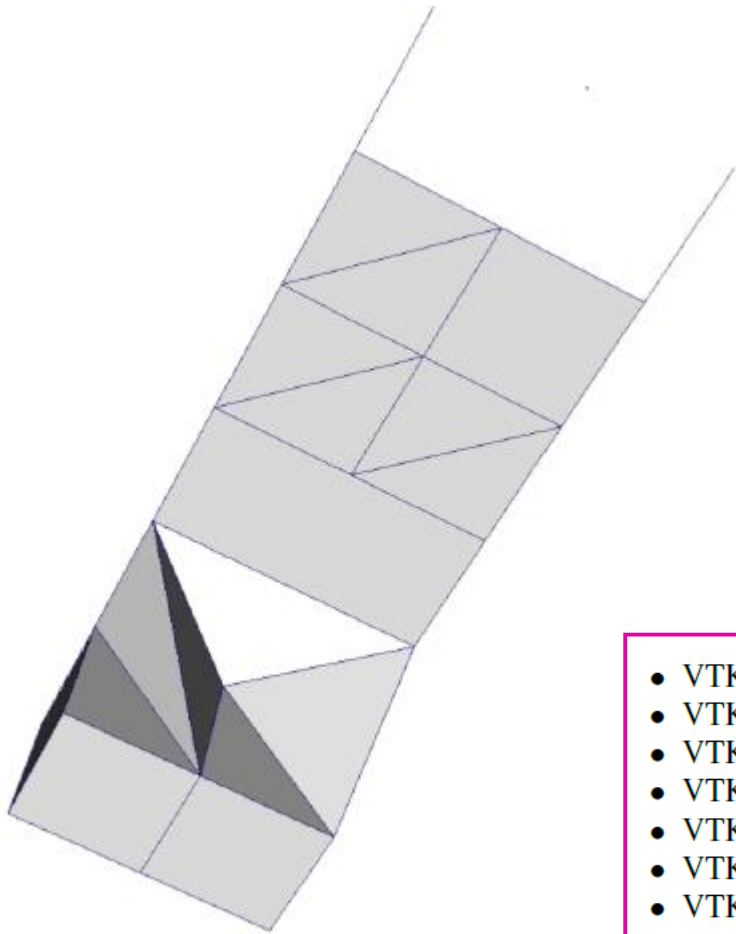
---



- ❖ *AMR dataset*: Adaptive Mesh Refinement.
- ❖ Colección de grillas rectangulares agrupadas según incrementos de coeficientes de refinamiento (*decreasing spacing*).
- ❖ Enmascara subregiones de la grilla rectilinear mediante arrays de bytes, lo cual permite que VTK pueda soportar el solapamiento (*overlapping*).

Es un conjunto  
complejo de  
datos

# Grillas no estructuradas



Unstructured grids.

La topología y las coordenadas se almacenan de manera explícita.

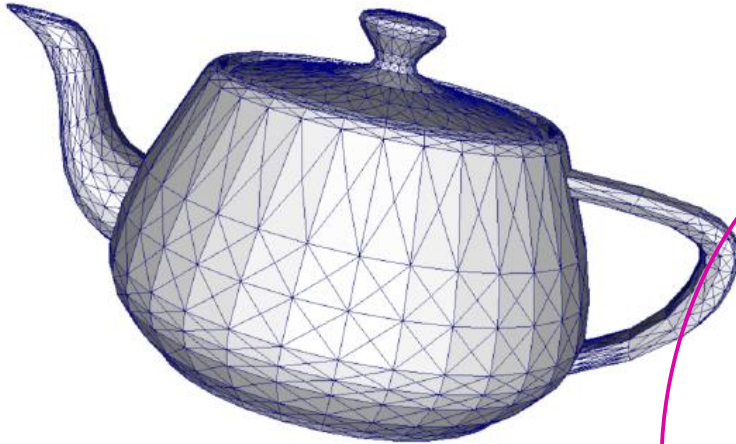
Altísimo consume de memoria.

Soporta diferentes tipos de celdas según la estructuración de VTK.

- VTK\_EMPTY\_CELL
- VTK\_VERTEX
- VTK\_POLY\_VERTEX
- VTK\_LINE
- VTK\_POLY\_LINE
- VTK\_TRIANGLE
- VTK\_TRIANGLE\_STRIP

- VTK\_POLYGON
- VTK\_PIXEL
- VTK\_QUAD
- VTK\_TETRA
- VTK\_VOXEL
- VTK\_HEXAHEDRON
- VTK\_WEDGE

# Grillas poligonales



▀ *Polygonal grid o polydata.*

▀ Grillas no estructuras eficientes para el renderizado.

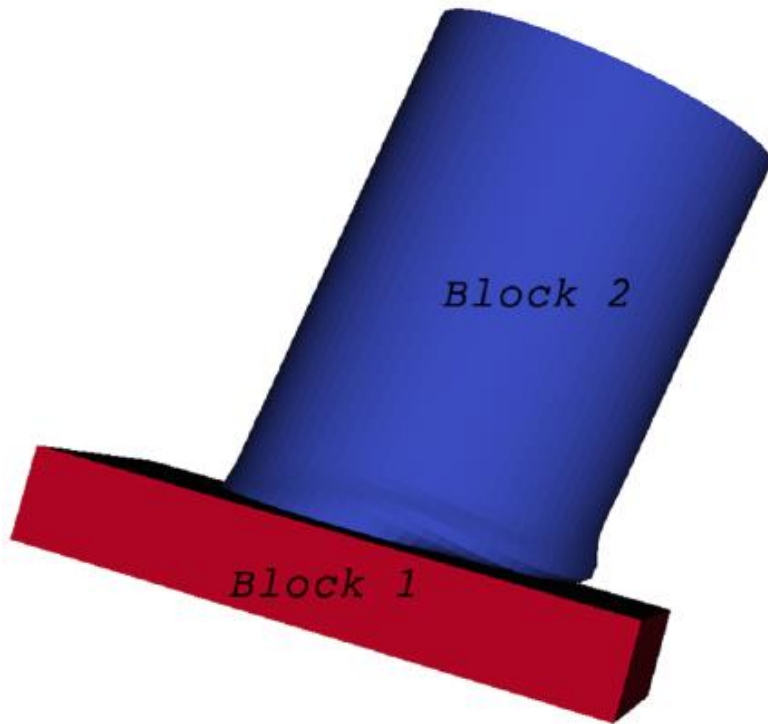
▀ Consiste de **celdas**:

- 0D: vertices y polivértices.
- 1D: líneas y polilíneas.
- 3D: polígonos y franjas triangulares

Hay filtros para convertir de grilla no estructurada a polidata y viceversa.

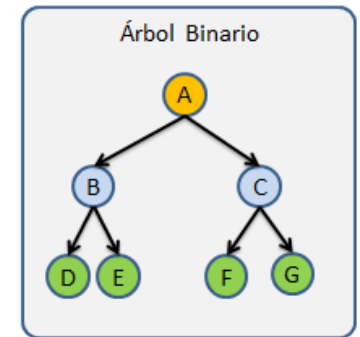
En VTK solo existe el concepto de celda y de punto, por lo que una celda se entenderá de diversas formas según sus dimensiones.

# Multibloques



❖ *Multiblock dataset.*

❖ Árboles de memoria donde los nodos son conjuntos de datos simples.



❖ Son conjuntos de datos relacionados ya sea dentro o fuera de ParaView.

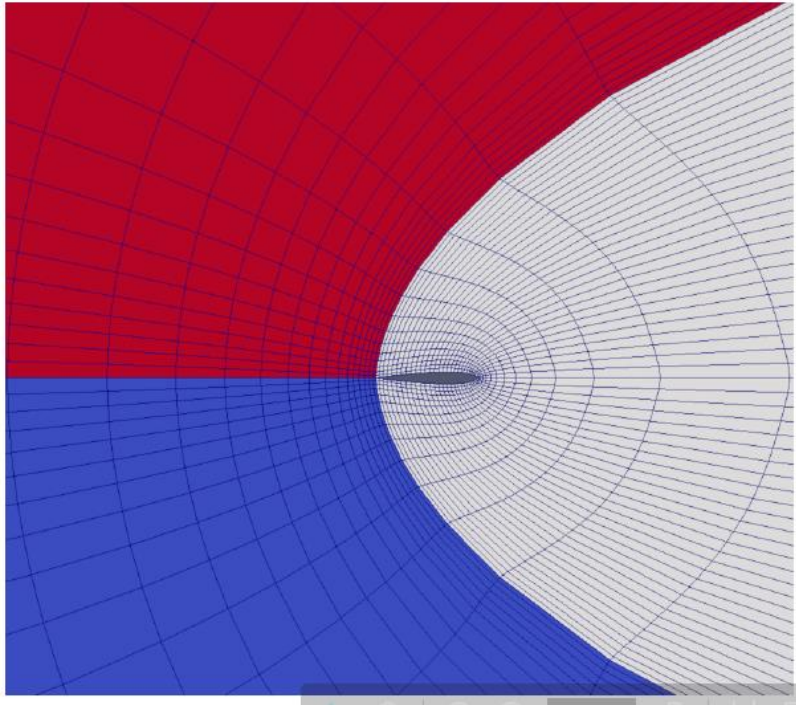


❖ Los nodos pueden tener diferentes atributos entre sí.

❖ Es un conjunto de datos simples (todos except el *ARM*).

# Conjuntos de datos multipieza

---



- *Multipiece dataset.*
- Agrupan conjuntos de datos simples que hacen parte de un mismo conjunto con los mismos atributos.
- No se puede crear internamente en ParaView.
- Se utilizan para recopilar conjuntos de datos producidos por una simulación paralela sin tener que unir las mallas.

Es un conjunto simple de datos

# Tablas

	Author	Affiliation			Alma Mater	Categories	Age	Coolness
0	Biff	NASA			Ole...	Jazz; Ro...	27	0.6
1	Bob	Bob's	Supermarket		Ole...	Jazz	54	0.3
2	Baz	Bob's	Supermarket		TVI	Food	16	0.3
3	Bippity	Oil	Changes	'R'	TVI	Food	23	0.2
4	Boppity	Oil	Changes	'R'	Home	Food; A...	34	0.25
5	Boo	Oil	Changes	'R'	Princeton	Automobiles	27	0.7

- ❖ *Tables.*
- ❖ Información en formato tabular compuesto de filas y columnas.
- ❖ Los filtros aplicables a las tablas generan tablas.
- ❖ Se pueden convertir a puntos o mallas estructuradas mediante filtros.
- ❖ Se pueden cargar con formato separación de coma .csv.

"Table to Point" y  
"Table to Structured  
Grid"

---





Obtener información en paraview

---

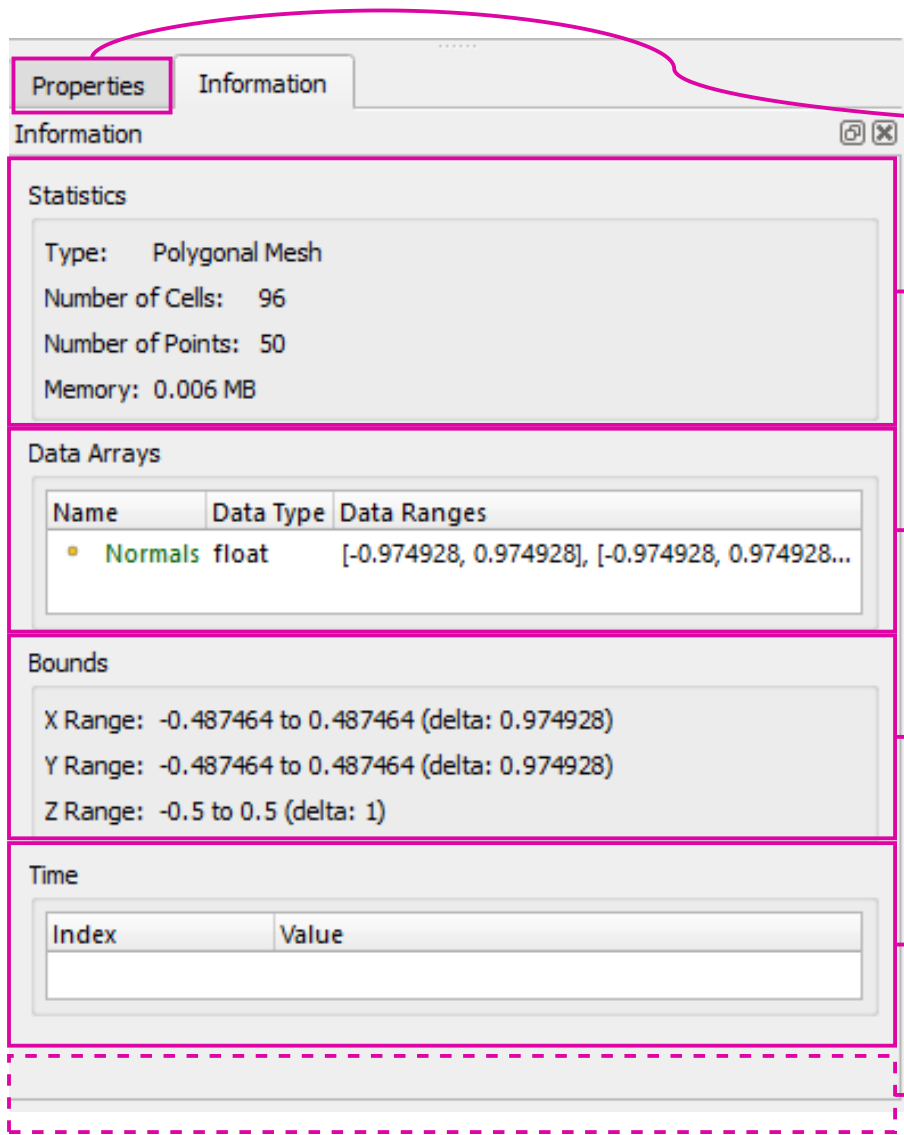


## Panel de información – *Information panel*

---

-  Muestra la información producida por la fuente activa, la cual será visible una vez se presione el botón Apply.
-  Si se tiene una serie de datos (datos temporales), se mostrará el estado escogido, es decir, el momento  $t$  o estado en el cual se está trabajando.
-  Toda la información de este panel se puede copiar y pegar para llevarlo a un documento, script, etc.
-  Las casillas de información se mostrarán en función del tipo de módulo que esté activo en el pipeline.

## Panel de información – *Information panel*



The screenshot shows the 'Information' panel of a software application. It has two tabs: 'Properties' and 'Information'. The 'Information' tab is active, displaying several sections: 'Statistics', 'Data Arrays', 'Bounds', and 'Time'. The 'Statistics' section shows 'Type: Polygonal Mesh', 'Number of Cells: 96', 'Number of Points: 50', and 'Memory: 0.006 MB'. The 'Data Arrays' section contains a table with one row: 'Normals' (float) with ranges [-0.974928, 0.974928]. The 'Bounds' section shows X, Y, and Z ranges. The 'Time' section has an empty table with 'Index' and 'Value' columns. A dashed pink box highlights the bottom of the panel.

Para *Readers* con información sobre el archivo que esté abierto.

Resumen del conjunto de datos.

Arreglo de puntos, celdas o campos disponibles e información al respecto.

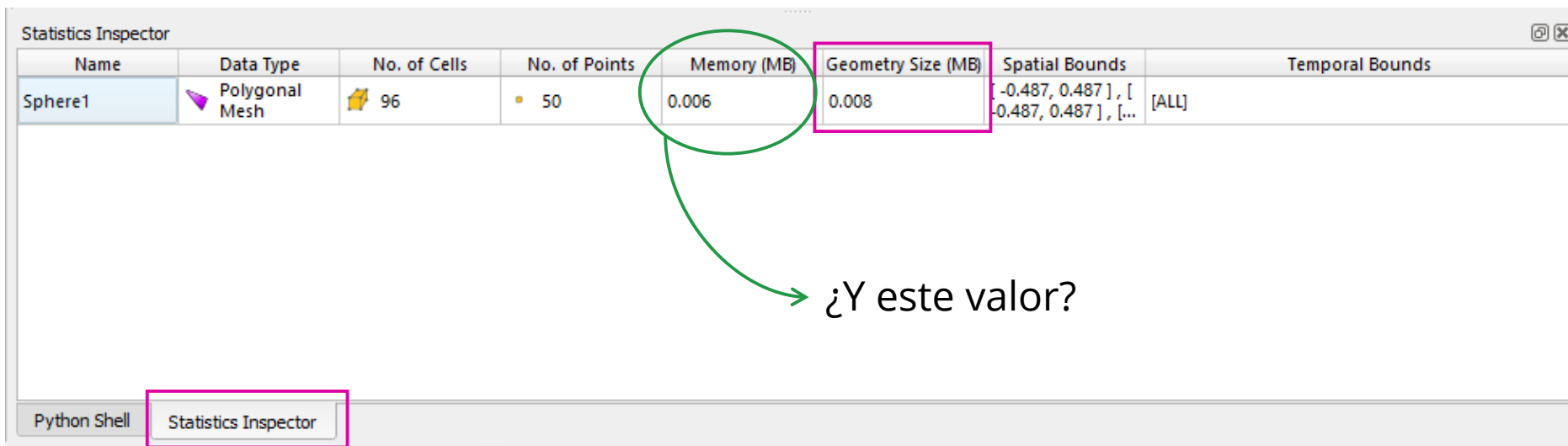
Límites espaciales para conjuntos de datos en coordenadas cartesianas 3D. NO disponible para tablas.

*Readers*. Información temporal para series de datos.

*EXTENDS*, cuando el dato lo permite.

## Panel de estadísticas – *Statistics inspector panel*

- Información producida por TODOS los módulos del pipeline, no solo la fuente activa; se activa desde el menú de *View*.
- Se encuentran las mismas opciones del panel de información y se adiciona el *Geometry Size*, el cual indica cuánta memoria se necesita para transformar el conjunto de datos usados en la vista activa. Por ejemplo, para renderizar un conjunto 3D como una superficie en la vista 3D, la malla superficial del objeto se extrae como *polydata*, y *Geometry Size* representa la memoria necesaria para este *polydata* con las mismas consideraciones que en el panel de información.





Name	Data Type	No. of Cells	No. of Points	Memory (MB)	Geometry Size (MB)	Spatial Bounds	Temporal Bounds
Sphere1	Polygonal Mesh	96	50	0.006	0.008	[-0.487, 0.487], [-0.487, 0.487], [...]	[ALL]

¿Y este valor?

# Memoria

---

El almacenamiento mostrado en los paneles anteriores no es exacta, ya que:

-  El tamaño no incluye la cantidad de memoria necesaria para construir las estructuras de datos para almacenar las matrices de datos. Si bien en la mayoría de casos esto es insignificante en comparación con las matrices de datos, puede no ser trivial, especialmente cuando se trata de conjuntos de datos compuestos profundamente anidados.
-  Varios filtros, como *Calculator* y *Shrink*, simplemente pasan las matrices de datos de entrada, por lo que no hay necesidad de espacio para esa información que queda. Sin embargo, los números de tamaño de memoria no toman esto en consideración.