

# Propiedades de la información

Dr. Luis Miguel de la Cruz Salas



# Introducción



# Datos abiertos

- Datos a los que se puede acceder, usar y compartir.
  - ¿Cómo se pueden usar una vez que se tienen? ¿Qué se puede hacer con estos datos?
  - No es equivalente a datos gratuitos.
- Gobierno, industria y sociedad (educación).
  - Se genera valor a partir de los datos y se pueden resolver muchos problemas.
- El internet ya es parte de la infraestructura de la sociedad, **los datos abiertos es el siguiente paso.**



# Datos abiertos: características

- Están disponible en línea (internet)
  - A través de diferentes tipos de plataformas.
  - Debe ser sostenible.
- Estos datos deben estar licenciados para su uso.
  - En cualquier ámbito, incluyendo propósitos comerciales.
  - La licencia entre más simple mejor! (creative commons attribution license).
- El costo de recuperación (cuando lo hay) no debe ser excesivo.
- Deben actualizarse continuamente.



# Datos abiertos: características

- Estos datos deben ser fácilmente usables
  - Formato de almacenamiento común y entendible para humanos y que pueda ser manipulado por máquinas.
- El formato debe ser estándar y de alta calidad.
  - En términos prácticos, legales, técnicos y sociales.
    - **Legales:** proteger información sensitiva, preservar los derechos de los dueños de los datos.
    - **Prácticos:** de dónde vienen los datos, cuál es el contexto, ...
    - **Técnicos:** formatos de los datos y los canales por los que se pueden obtener.
    - **Sociales:** construir comunidades, soporte económico, discusión, ...



# Formatos, estructura y distribución

- Formato.
  - Seleccionar un formato que sea común, que esté disponible y que sea entendible.
  - Usar .CSV como formato adicional.
  - Usar tantos formatos como sea necesario
  - Prepararse para la obsolescencia.





# Formatos, estructura y distribución

- Estructura de los datos
  - Tabular
  - Jerárquicos (relaciones, redes, grafos)
  - Espaciales (georeferenciados)
- Distribución
  - Tamaño
  - Frecuencia de actualización.
  - Terminología que se usa para que sean entendibles.
  - Datos en tiempo real





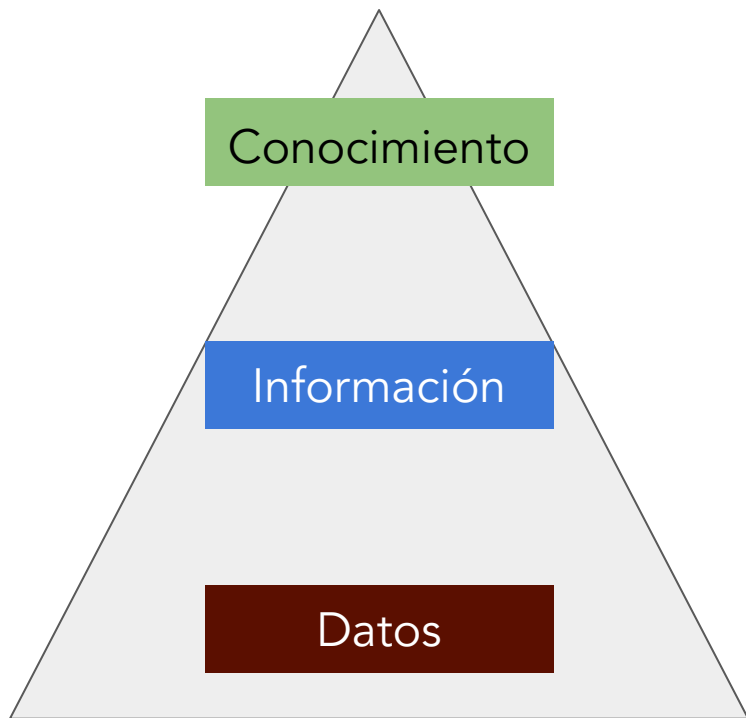
## En resumen

- Los datos abiertos deben:
  - Estar disponibles a través de una plataforma en internet.
  - Deben tener una licencia que permita usarlos en cualquier ámbito.
  - Deben ser reusables, estar bien estructurados y en un formato entendible.
  - No deben estar atados a un software particular.
- Lo anterior permitirá crear nuevos códigos de análisis y herramientas para explorar esos datos.





# De los datos la conocimiento



- Antes de comenzar a contar una historia, los datos crudos deben limpiarse, formatearse, analizarse.
  - El 80% del tiempo se usa en preparar los datos.
  - [OpenRefine](#)
- Posteriormente se comienza el análisis para generar información.
- Finalmente, esta información junto con la experiencia genera conocimiento.



# Análisis de datos



# Análisis de datos

- Los datos útiles son los que se analizan mediante diversas técnicas.
  - Estas técnicas dependen del tipo de información que se haya recopilado.
- El análisis puede ser:
  - **Cualitativo**: los datos se presentan de manera verbal, se basa en la interpretación y se obtienen mediante entrevistas abiertas, grupos de discusión o de observación y se utilizan patrones en toda la fase de recolección de datos. (véase por ejemplo escala de [Likert](#)).



# Análisis de datos

- (cont.):
  - **Cuantitativo**: se presentan de forma numérica y se basa en resultados tangibles.

	Tipos de análisis	Tipo de preguntas
Cualitativo	Se centra en la opiniones, actitudes, creencias, etc.	¿Por qué? ¿Cómo? ¿Qué tan satisfecho?
Cuantitativo	Se centra en los datos duros e información que pueda contabilizarse	¿Cuántos? ¿Quién? ¿Dónde? ¿Frecuencia?



# Los datos y las escalas de medición





# Los datos

- Representación de atributos o variables que describen hechos.
- Su análisis y procesamiento los transforman en información.
- Lo anterior requiere de la comparación de los datos entre sí y con respecto a referencias conocidas.
- Así mismo, la comparación necesita de escalas de medición para situar cada valor que tomen los datos.



# Las escalas de medición

- Entender lo que significan los datos.
  - Clasificar cada variable según su escala de medición.
  - **Escala de medición**: relación entre los valores que se asignan a las variables
  - **Variable**: cualquier cantidad que puede ser medida.
    - Población de estudiantes, nacionalidad, género, carrera, facultad, calificaciones, ...
- Cuatro escalas <sup>1</sup>:
  - Nominal, ordinal, de intervalo y de razón.

<sup>1</sup>Stevens, Stanley. (1946). On the Theory of Scales of Measurement. Science, New Series, Vol. 103, No. 2684, pp. 677-680. American Association for the Advancement of Science.



# Las escalas de medición

- Propiedades del sistema numérico asociadas con las escalas de medición <sup>2</sup>:
  - **Identidad**: cada número tiene un significado particular.
  - **Magnitud**: los números tienen un orden inherente ascendente o descendente.
  - **Intervalos iguales**: las diferencia entre números en cualquier punto de la escala son las mismas (la diferencia entre 10 y 20 es la misma que entre 100 y 110).
  - **Cero absoluto**: el punto cero en la escala de medición representa la ausencia de la propiedad que se estudia.

<sup>2</sup> Stevens, Stanley. (1957). On the Psychological Law. Psychological Review 64, Pp. 153-181. American Psychological Association. USA.





# Escala nominal

- Útiles para clasificar observaciones.
- El dato identifica el nombre de un atributo.
- No hay un orden. No es posible la comparación ni las operaciones aritméticas.
- Ejemplo:
  - Raza de gatos: siamés, angora, persa, esfinge, ...
  - Genotipo (inf. genética): AA, AT, AG
  - Género: Femenino o Masculino.
- Propiedades: **identidad**.



# Escala ordinal

- El dato identifica el nombre de un atributo.
- Tiene sentido el orden o la jerarquía, por lo que es posible la comparación, pero no se sabe la diferencia entre un dato y otro.
- Las operaciones aritméticas no tienen sentido.
- Ejemplo:
  - ¿Qué tan contento estás con la clase de Vis. de la Info.?:
    - 1. Muy descontento, 2. Descontento, 3. Equis, 4. Contento, 5. Muy contento.
- Propiedades: **identidad y magnitud**.



# Escala de intervalo

- Se puede establecer orden entre sus valores y hacer comparaciones de igualdad, así como medir la distancia existente entre cada valor de la escala.
- El valor cero de la escala no es absoluto, sino un cero arbitrario.
  - No refleja ausencia de la magnitud medida, por lo que las operaciones aritméticas de multiplicación y división no están bien definidas.
- Propiedades: **identidad, magnitud e igual distancia.**



# Escala de intervalo

- Ejemplos:
  - Hora del día: la diferencia entre las 10 am y 11 am, es la misma que entre las 5 pm y 6 pm.
  - Temperatura: la diferencia entre 25 °C y 30 °C es la misma que entre 5 °C y 10 °C.



# Escala de razón

- Nivel de medición más completo con las mismas propiedades que la escala de intervalos.
- Posee el cero absoluto, el cual representa la ausencia total de la magnitud que se está midiendo.
- Se puede realizar cualquier operación lógica (ordenamiento, comparación) y cualquier operación aritmética.
- Ejemplos:
  - longitud, peso, distancia, ingresos, precios.





# Resumen

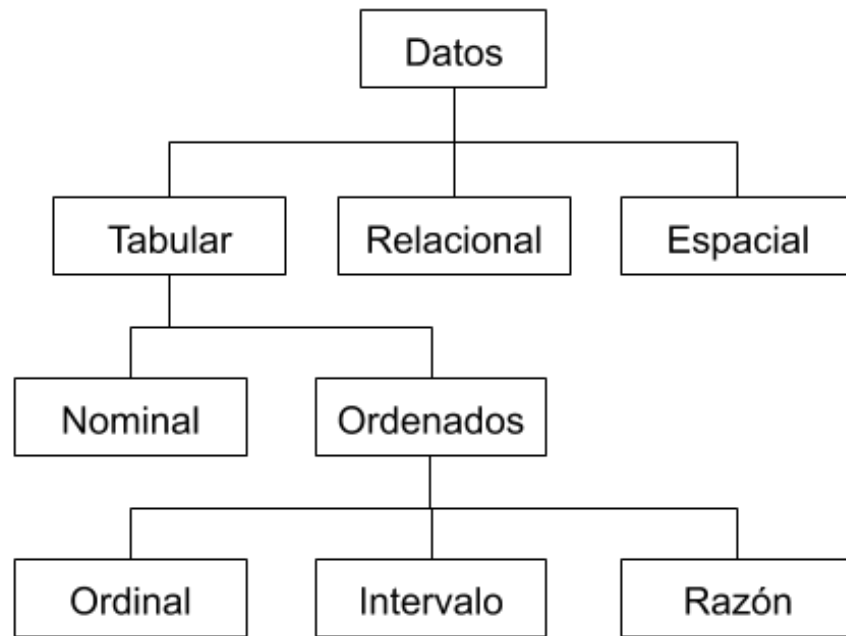
Escala de medición	Propiedad sistema numérico	Operación matemática	Operación estadística	Ejemplos
Nominal	Identidad	Contar	Frecuencia Moda	Género Genotipo
Ordinal	Magnitud	Ordenar	Mediana Rango	Educación Dureza de mat.
Intervalo	Distancia	Suma y resta	Mediana Varianza	Calificaciones Temperatura
Razón	Cero absoluto	Multiplicación División	Coeficiente de Variación	Peso, Longitud, Pesos, Precio

Tabla acumulativa. Las propiedades de una escala incluyen todas las propiedades de la escala anterior

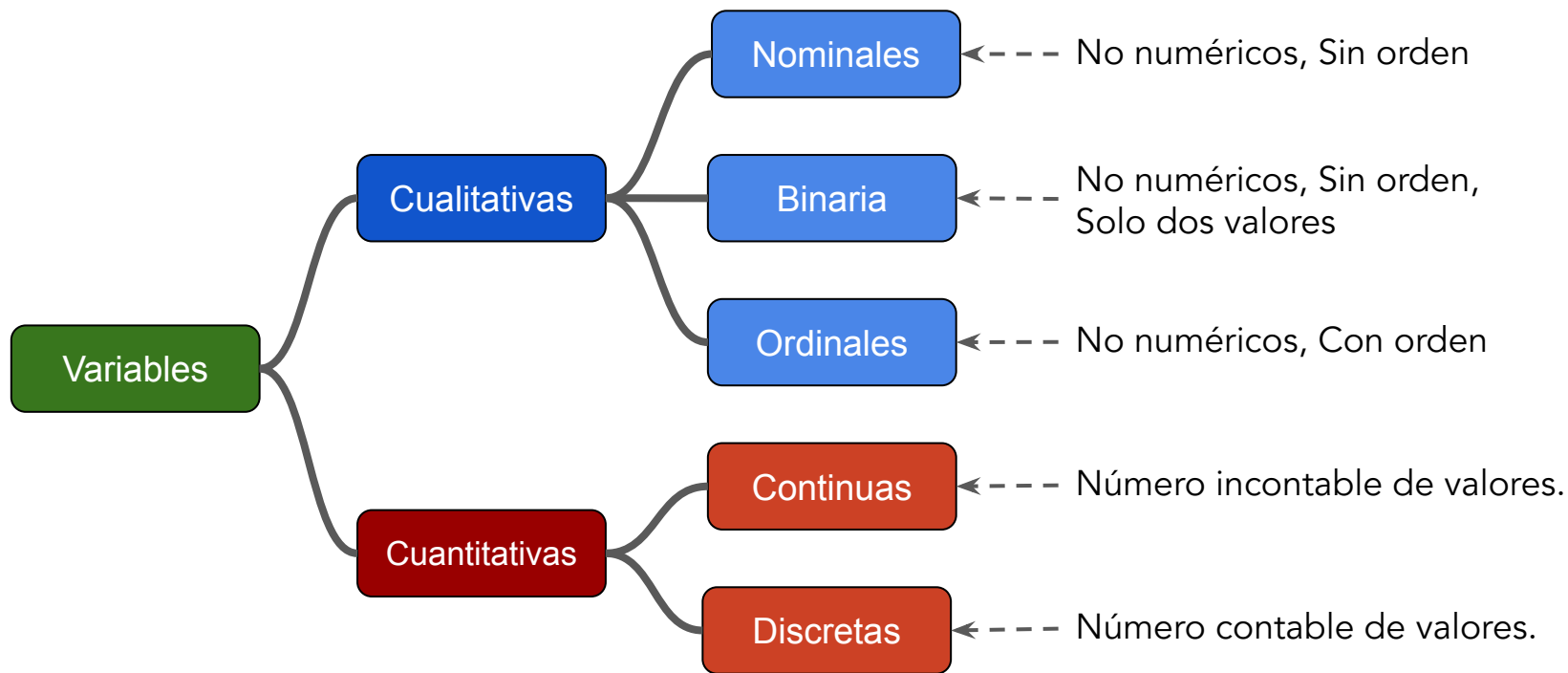


# Taxonomías

- 1D
- Series de tiempo
- 2D
- 3D
- nD
- Árboles
- Textos, documentos
- Mapas



# Variables estadísticas





# Ejemplos

1. Número de alumnos en una clase de licenciatura.
2. Estatura de 30 personas en una sala de espera.
3. Tipo de medallas que otorgan en las olimpiadas (oro, plata, bronce).
4. Número de páginas de la novelas de Gabo.
5. Grado de satisfacción (muy satisfecho, satisfecho, regular, insatisfecho, muy insatisfecho).
6. Respuesta a la pregunta: ¿Tiene casa propia?: Si, No
7. Estado Civil: Soltero, Casado, Divorciado, Unión libre, Viudo.
8. Tiempo requerido para responder las llamadas en un *call center*.
9. Lugar que ocupa un nadador en una competencia.
10. Da un ejemplo de una variable cuantitativa nominal.



# Respuestas

1. Cuantitativa, discreta.
2. Cuantitativa, continua.
3. Cualitativa, ordinal
4. Cuantitativa, discreta.
5. Cualitativa, ordinal.
6. Cualitativa, binaria
7. Cualitativa, nominal
8. Cuantitativa, continua.
9. Cualitativa, ordinal.
- 10.



# Metadatos



# Metadatos

- Término acuñado en la década de los 60 <sup>1</sup>.
- Significado etimológico: “más allá de los datos”
  - Meta (*μετα*): después de o más allá de
  - *Datum*: datos
- Conjunto de datos que describen el contenido informativo de recursos electrónicos o digitales.
  - Proporcionan la información mínima necesaria para identificar un recurso (información descriptiva sobre el contexto, calidad, condición y características de los datos).

<sup>1</sup> Término acuñado por Jack Myers en los 60s, Caplan, Priscilla (1995). *You call it corn, we call it syntax-independent metadata for document like objects*. The Public Access Computer Systems Review, v. 4, n. 6. <http://epress.lib.uh.edu/pr/v6/n4/capl6n4.html>.

# Funciones de los metadatos

- Resumir el significado de los datos
- Permitir la búsqueda
- Determinar si el dato es el que se necesita
- Prevenir ciertos usos
- Recuperar y usar una copia del dato
- Mostrar instrucciones de cómo interpretar un dato
- Obtener información sobre las condiciones de uso (derechos de autor)
- Aportar información acerca de la vida del dato
- Ofrecer información relativa al propietario/creador
- Indicar relaciones con otros recursos
- Controlar la gestión

# Importancia de los metadatos

**Incrementan la accesibilidad:** la existencia de un conjunto de metadatos que describan correctamente uno o varios objetos aumenta la posibilidad de acceder a ellos; hacen posible la búsqueda de información en múltiples colecciones a la vez; por medio del mapeo entre sistemas heterogéneos es posible consultar, con una única ecuación de búsqueda, bases de datos que utilicen diferentes sistemas de metadatos para describir sus objetos.

**Disminución del tráfico en la Red:** al indizar la representación del objeto, y no el objeto en sí, no requiere demasiado ancho de banda para hacer las búsquedas o generar los índices.



# Importancia de los metadatos

**Expandir el uso de la información:** ya que facilitan la difusión de versiones digitales de un único objeto.

**Control de versiones:** no sólo en lo que se refiere a gestionar la vida de un objeto, sino también en lo que tiene que ver con su difusión, es decir: generar diferentes metadatos con distintas cantidades de información sobre un mismo objeto con el fin de distribuirla a un público heterogéneo.

**Aspectos legales:** los metadatos permiten establecer claramente las restricciones de explotación, informar sobre los derechos de autor, control del uso de todo o una parte del objeto, método de pago por su disfrute, controlar el acceso a información restringida.



# Importancia de los metadatos

**Preservación del objeto original:** las búsquedas a través del Web son, en la actualidad, un proceso de equiparación (matching) entre los términos de la consulta y los del documento. Si esa equiparación no se produce (bien sea por un problema en la forma de definir la petición, bien porque esa información sí se encuentra pero bajo otro concepto que lo describe), el documento no se recuperará. Para estas autoras la utilización de metadatos junto al uso de lenguajes controlados permitiría aumentar la precisión en la mayoría de búsquedas en Internet.





# Ejemplos

- Existen “lenguajes” que permiten especificar la sintaxis en la que se definen las estructuras de los metadatos.
  - También proveen de las especificaciones semánticas necesarias (significado de las expresiones sintácticas).
- Metalenguajes.
  - SGML (*Standard Generalized Markup Language*)
    - Estándar que sirve para especificar lenguajes de marcado (*markup*).
    - Los docs se crean con base en su estructura, no en su apariencia.
    - Se puede interpretar cualquier documento con base en su DTD.
    - Ejemplos: HTML, XML, OED, ...





# HTML

```
<HTML>
<HEAD>
<TITLE>El super título </TITLE>
</HEAD>
<BODY>
<!-- Este es un comentario -->
<H1>Un subtítulo. </H1>
<P> <IMG SRC= "./Huevo_cocido.jpg" WIDTH=100 ALIGN= "MIDDLE " ALT= "Huevo cocido">
<P>Empezamos creando un párrafo con información
que puede ser útil para la descripción del tema en cuestión ... <br> me cambio de línea y continuo.
Necesito <STRONG> resaltar </STRONG> esta parte.
<P>Aquí comienzo el texto verdadero ...
<H3>Los subtítulos pueden se de varios tamaños </H3>

<UL>
<LI> Se pueden construir listas
<LI> con varios elementos.
</UL>

Y puedo hacer referencia a otros documentos <A HREF= "https://www.unam.mx/">UNAM</A>.
</BODY>
</HTML>
```






El super título - Mozilla Firefox

El super título

gmc.geofisica.unam.mx/pru 120%

# Un subtítulo.



Empezamos creando un párrafo con información que puede ser útil para la descripción del tema en cuestión ...  
me cambio de línea y continuo. Necesito **resaltar** esta parte.

Aquí comienzo el texto verdadero ...

## Los subtítulos pueden se de varios tamaños

- Se pueden construir listas
- con varios elementos.

Y puedo hacer referencia a otros documentos [UNAM](#).



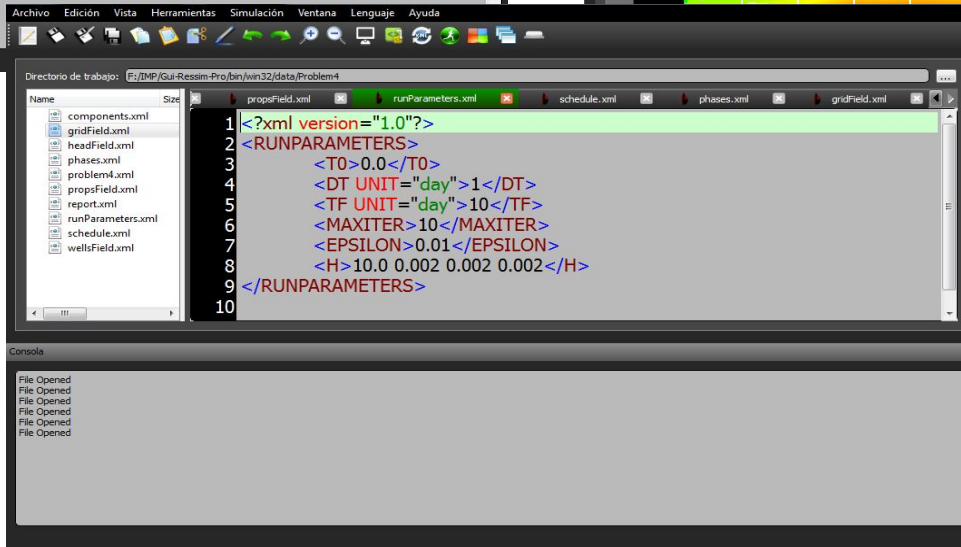
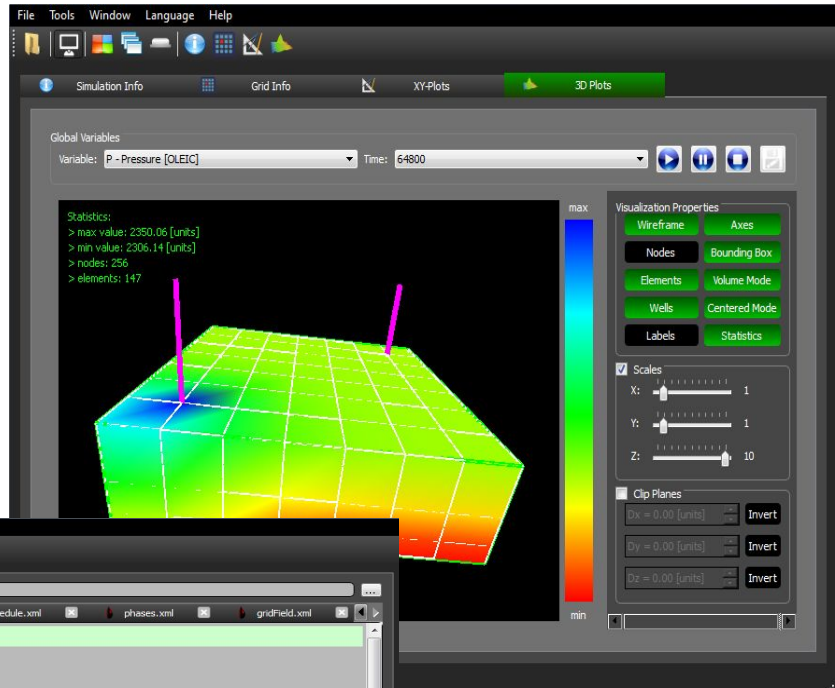
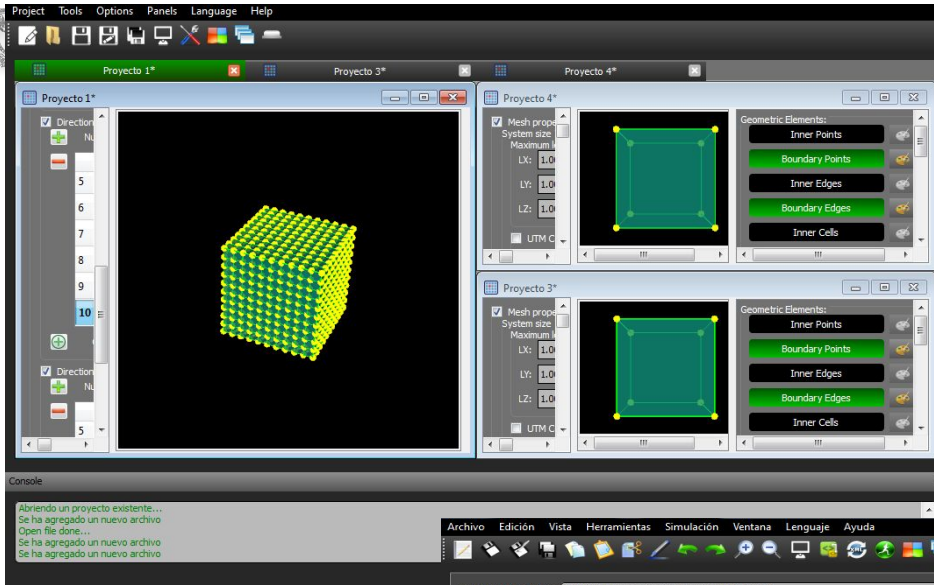
# XML

- eXtensible Markup Language.
- Diseñado para:
  - almacenar y transportar datos.
  - ser entendido por los humanos y las computadoras.

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend! </body>
</note>
```



- Tutorial: <https://www.w3schools.com/xml/default.asp>





# Ejemplos de sistemas de metadatos

- BibTeX
  - Bibliografías basadas en LaTeX.

```
@article{10.1145/2835175,  
  author = {Cruz, Luis M. De La and Ramos, Eduardo},  
  title = {General Template Units for the Finite Volume Method in Box-Shaped Domains},  
  year = {2016},  
  issue_date = {August 2016},  
  publisher = {Association for Computing Machinery},  
  address = {New York, NY, USA},  
  volume = {43},  
  number = {1},  
  issn = {0098-3500},  
  url = {https://doi.org/10.1145/2835175},  
  doi = {10.1145/2835175},  
  journal = {ACM Trans. Math. Softw.},  
  month = aug,  
  articleno = {Article 1},  
  numpages = {32},  
  keywords = {Generic programming, natural convection, two-phase flow}  
}
```



# Ejemplos de sistemas de metadatos

- Dublin Core
  - Modelo gestionado por la DCMI (*Dublin Core Metadata Initiative*) constituido por elementos para autores y editores de documentos electrónicos para facilitar la creación de registros de metadatos.
  - La OAI-PMH (*Open Archive Initiative-Protocol for Metadata Harvesting*) desarrolla y promueve estándares de interoperabilidad para facilitar la difusión eficiente de contenidos en Internet.





# Metadatos Dublin Core

1. [dc:title](#) Título del recurso
2. [dc:identifier](#) Dirección URL  
(Uniform Resource Locator)  
válida con acceso al recurso
3. [dc:creator](#) Autor del recurso
4. [dc:type](#) Tipo de material
5. [dc:date](#) Fecha de publicación
6. [dc:source](#) Fuente
7. [dc:rights](#) Licencia del material
8. [dc:publisher](#) Editor
9. [dc:language](#) Idioma
10. [dc:description](#) Descripción o  
resumen
11. [dc:format](#) Formato
12. [dc:subject](#) Tema o materia
13. [dc:coverage](#) Cobertura espacial o  
temporal
14. [dc:contributor](#) Colaborador
15. [dc:relation](#) Relación





# Formatos



# Formatos

- Comunes:
  - CSV (*Comma Separated Values*)
  - JSON (*JavaScript Object Notation*)
  - YAML (*YAML Ain't Markup Language*)
  - XML (*eXtensible Markup Language*)
- Científicos.
  - [HDF5](#) (*Hierarchical Data Format*)
  - RESQML, PRODML, WITSML ([Energistics](#) ).
  - [NetCDF](#) (*Network Common Data Form*)



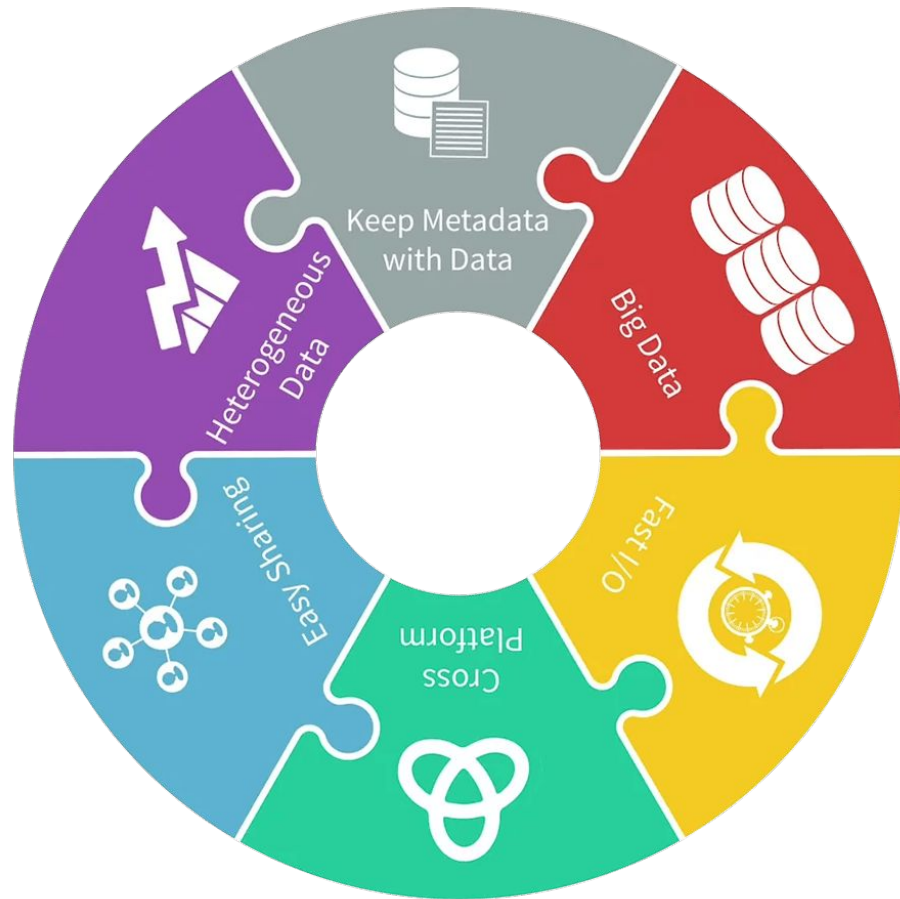
# HDF5

- *Hierarchical Data Format version 5*
- Es un formato abierto que soporta conjuntos de datos grandes, complejos y heterogéneos.
- Utiliza una estructura similar a directorios de archivos, lo que permite organizar los datos dentro de un archivo de muchas maneras.
- Contiene la definición de metadatos para hacer de este un formato auto descriptivo.
- Toma ideas de HDF4 y NetCDF.



# HDF5

- ¿Quién usa HDF5?
  - Astronomía.
  - Dinámica de Fluidos Computacional (CFD).
  - Ciencias de la tierra.
  - Ingeniería.
  - Finanzas.
  - Genómica.
  - Medicina.
  - Física.



[hdfgroup: benefits wheel](http://hdfgroup.org/benefits/wheel)



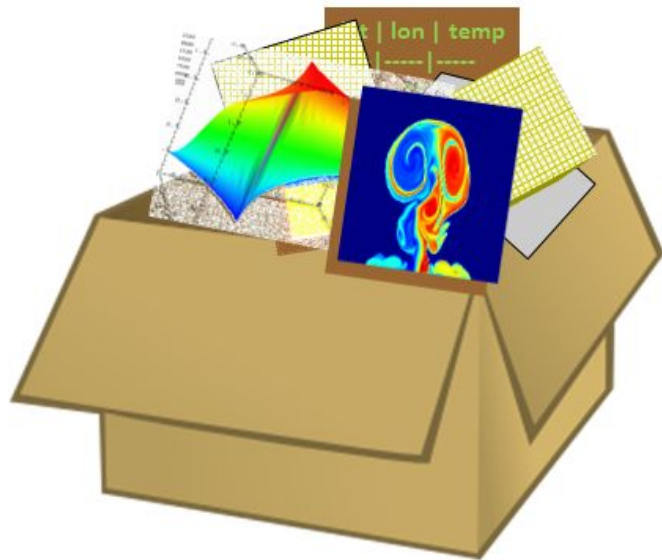
Descripción de HDF5 ([link](#))





# HDF5: ¿Qué es?

- *Formato* para almacenar los datos.
- Un *modelo de datos* para organizar y acceder a la información desde una aplicación.
- *Software* para trabajar con este formato (bibliotecas, interfaces para distintos lenguajes y herramientas).



Un archivo en formato HDF5 (*object*) se puede pensar como un contenedor (*group*) que mantiene una variedad de objetos de datos heterogéneos (*datasets*).



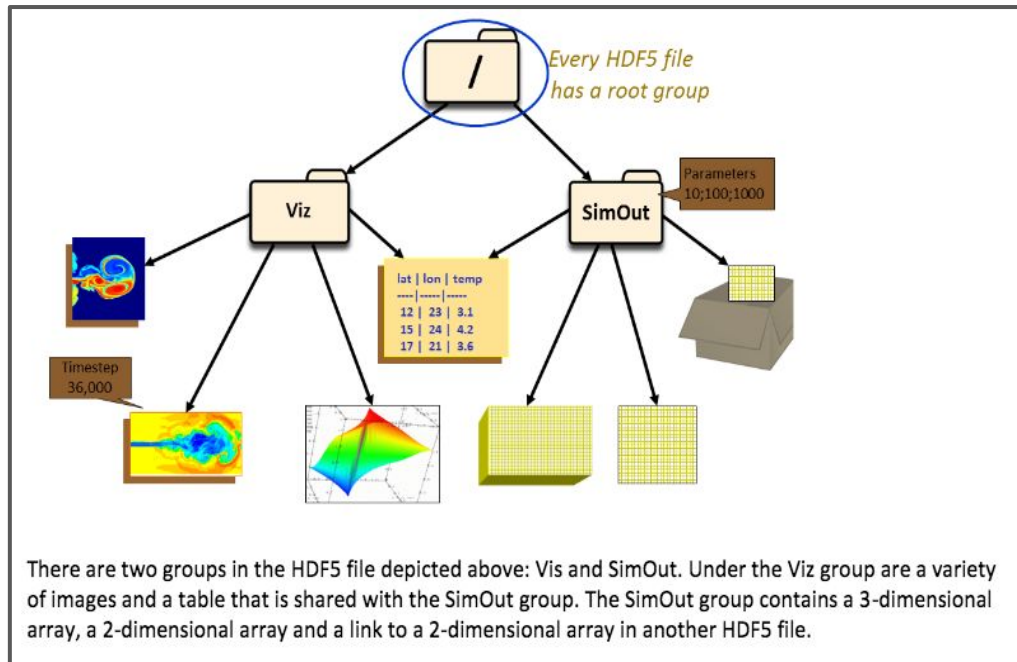
# HDF5: *Data Model*

Los dos objetos primarios son:

- Grupos (*groups*)
- Conjuntos de datos (*datasets*)

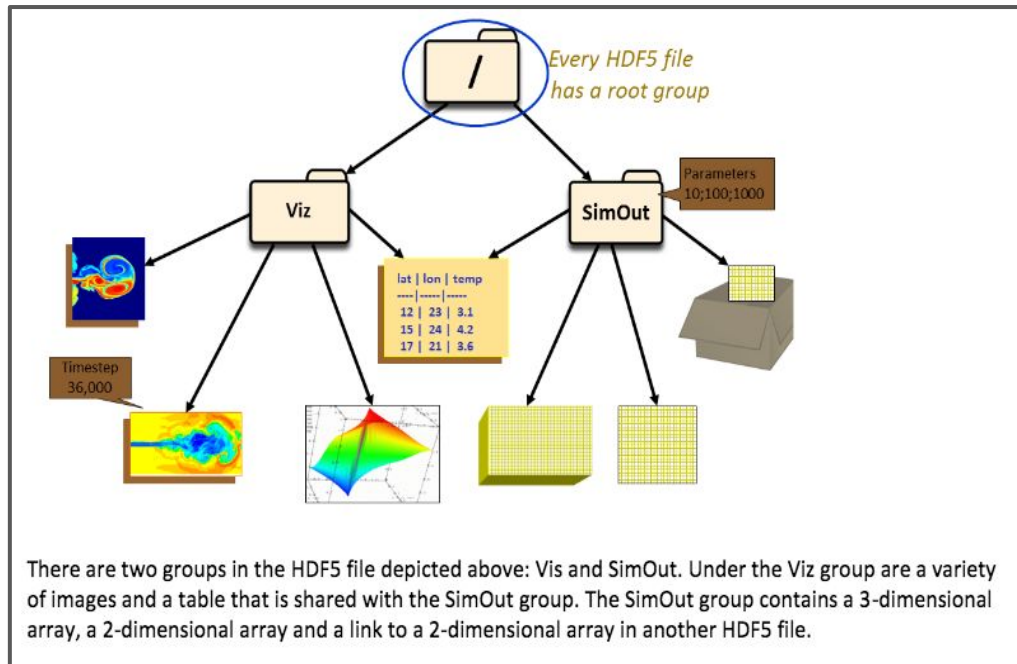
Existen otro tipo de objetos:

- *datatypes*
- *dataspaces*
- *properties*
- *attributes*



# HDF5: *Groups*

- Organizan los objetos de datos
- Cada archivo HDF5 contiene un grupo raíz que puede contener otros grupos o estar ligado a objetos en otros archivos.





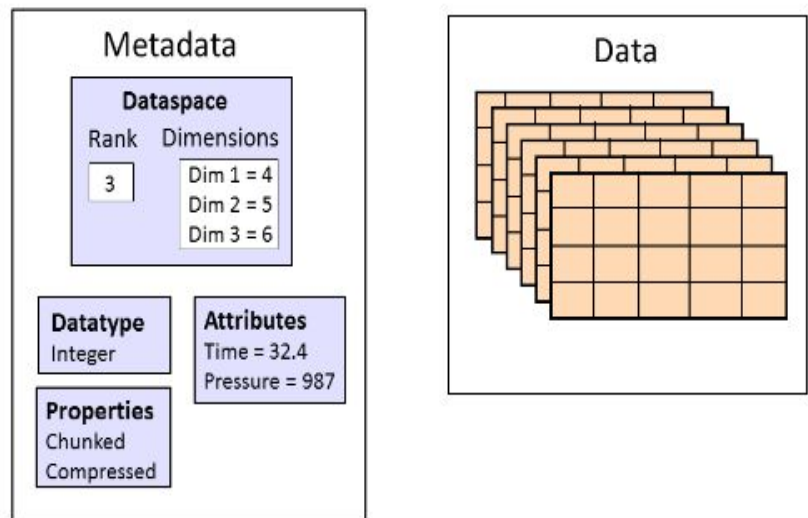
# HDF5: *Groups*

- Trabajar con grupos y miembros del grupo es similar a trabajar con directorios y archivos en UNIX.
- Los objetos en un archivo HDF5 se pueden describir mediante sus nombres completos (rutas absolutas):
  - / significa la raíz del grupo.
  - /grupo\_01 significa un miembro del grupo raíz de nombre grupo\_01.
  - /grupo\_01/datos significa un miembro del grupo grupo\_01 de nombre datos. A su vez es grupo\_01 es miembro del grupo raíz.



# HDF5: *Datasets*

- Organizan y contienen los valores de los datos "crudos".
- Consisten de:
  - Metadatos para describir los datos.
  - Los datos en sí mismos.
- Objetos importantes para describir los datos:
  - *datatypes*
  - *dataspaces*
  - *properties*
  - *attributes* (opcional)

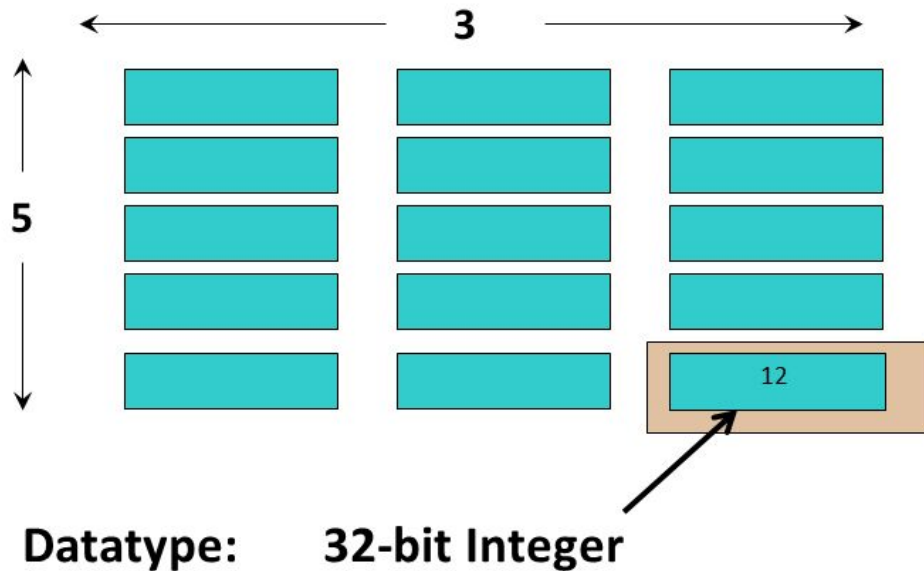


- Arreglo de 4 x 5 x 6 (*dataset*).
- Los datos son de tipo entero (*datatype*).
- Atributos: Time y Pressure (*attributes*).
- Los datos están fragmentados (*chunked*) y comprimidos (*compressed*).



# HDF5: *Datatype*

- Describe los elementos individuales en el conjunto de datos.
- Provee información completa para cualquier tipo de conversión.
- Se pueden agrupar en dos tipos:
  - *Pre-defined datatypes.*
  - *Derived datatypes.*



# HDF5: *Datatype, pre-defined*

Son creados por HDF5 y existen dos tipos:

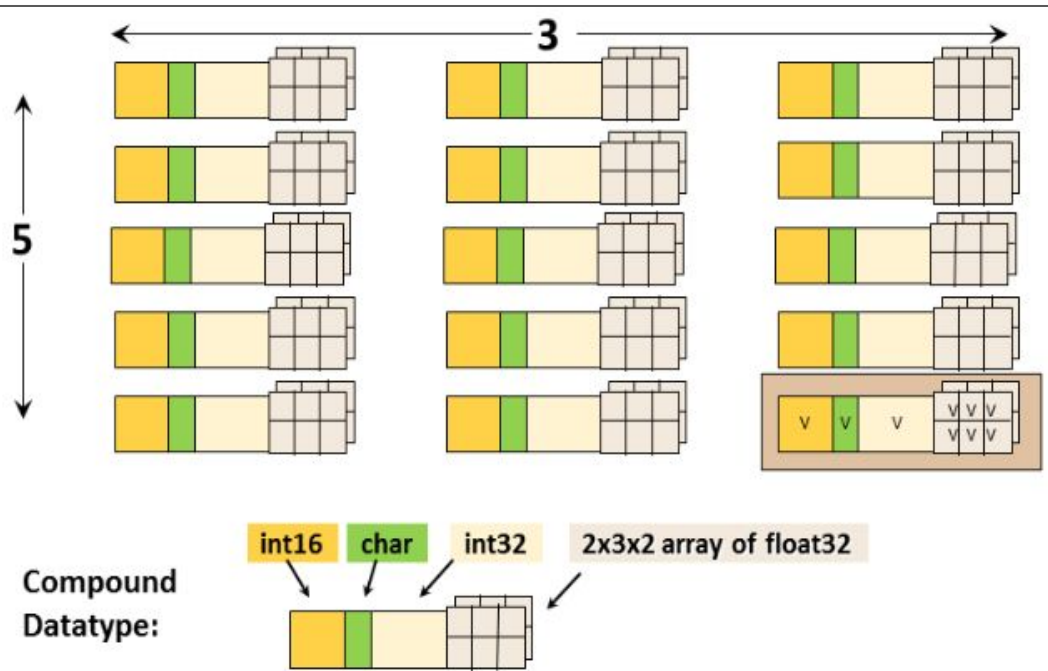
- *Standard datatypes*: son los mismos en todas las plataformas y son justo lo que se ve en el archivo HDF5. Sus nombres son del tipo H5T\_ARCH\_BASE, donde ARCH es el nombre de la arquitectura y BASE es el nombre del tipo de implementación. Por ejemplo: H5T\_IEEE\_F32BE\_IEEE indica tipo flotante estándar donde F32BE significa 32-bit Big Endian floating point.
- *Native datatypes*: se usan para simplificar las operaciones en memoria (lectura y escritura) y no son los mismos en plataformas diferentes.



# HDF5: *Datatype, derived*

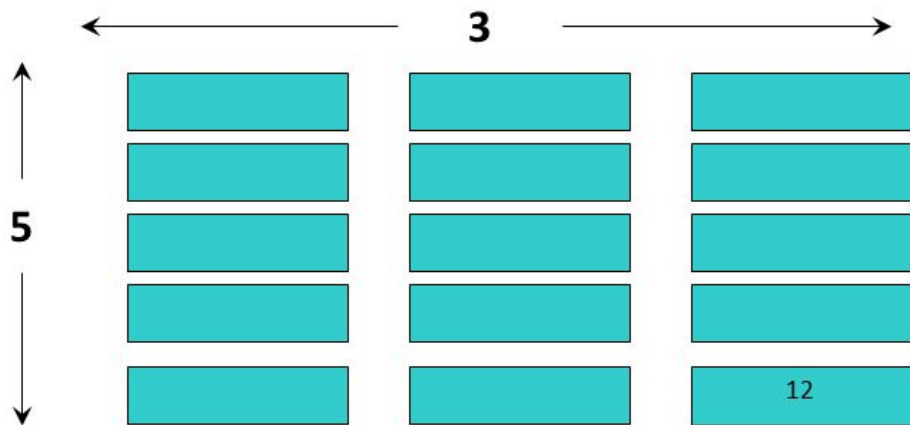
Son creados o derivados a partir de datos predefinidos. Por ejemplo:

- Consiste de:
  - un entero de 16 bits,
  - un caracter,
  - un entero de 32 bits,
  - un arreglo de flotantes de 32 bits de  $2 \times 3 \times 2$ .
- Es un arreglo bidimensional de  $5 \times 3$  (el *dataspace*).
- No confundir el *datatype* con el *dataspace*.



# HDF5: *Dataspace*

Describe la disposición en la que se encuentran los elementos del conjunto de datos (*dataset*) y puede consistir de: cero elementos (NULL), de un escalar o de un arreglo.



**Dataspace:**    **Rank = 2**  
**Dimensions = 5 x 3**



# HDF5: *Dataspace*

Tiene dos funciones:

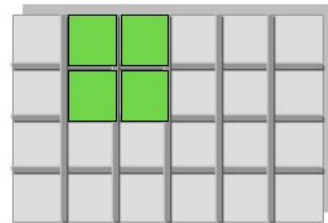
- Contiene la información espacial (*logical layout*) del conjunto de datos (*dataset*) almacenado en un archivo; incluye el rango (*rank*) y las dimensiones del *dataset*, esta información es permanente.
- Puede usarse para seleccionar una porción (subconjunto) del *dataset* que será usada en operaciones de entrada y salida (I/O).

Logical Layout



Rank = 2  
Dimensions = 4 x 6

Subset



Rank = 2  
Dimensions = 2 x 2



# HDF5: *Properties*

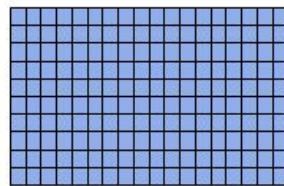
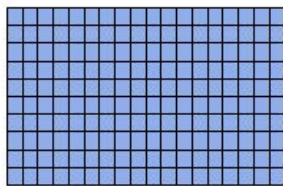
- Es una característica de un objeto HDF5. Existen propiedades por omisión que manejan las necesidades más comunes, que pueden ser modificadas usando el [API HDF5 Property List](#).
- Por ejemplo:
  - La propiedad de la disposición de almacenamiento de los datos (el [dataspace](#)) es contigua por omisión. Para un mejor rendimiento, esta disposición puede modificarse para que sea fragmentada ([chunked](#)) o fragmentada - comprimida ([chunked - compressed](#)).





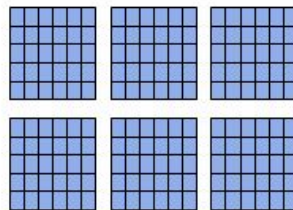
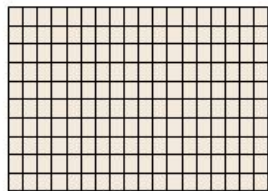
# HDF5: *Properties*

Contiguous  
(default)



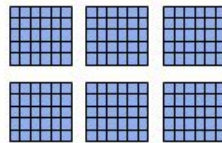
**Data elements  
stored physically  
adjacent to each  
other**

Chunked



**Better access time  
for subsets;  
extendible**

Chunked &  
Compressed



**Improves storage  
efficiency,  
transmission speed**



# HDF5: *Properties*

- Se pueden asociar, de manera opcional, atributos a los objetos HDF5.
- Tienen dos partes: un nombre y un valor.
- Los atributos no son objetos independientes, están ligados a otros objetos HDF5.
- Se accede a los atributos abriendo el objeto al que están ligados.
- Típicamente su tamaño es pequeño, contiene metadatos definidos por el usuario los cuales definen características del objeto al que están ligados.
- Al igual que los conjuntos de datos (*datasets*), los atributos tienen un *datatype* y un *dataspace*.
- Los atributos no soportan operaciones de I/O y no pueden ser comprimidos ni extendidos.



# HDF5 Programming Model and API





# Introducción

- HDF5 es un software escrito en C.
  - Se puede usar desde C++, FORTRAN (90 y 2003), Java y Python.
- Consiste de
  - Bibliotecas, Archivos de cabecera, Utilerías en línea de comandos, Scripts para compilación de aplicaciones y programas de ejemplo.
- El API de HDF5 es extenso, pero con solo unas pocas operaciones se puede hacer la mayor parte del trabajo.





# Introducción

- En el lenguaje C todas las funciones comienzan con H5\*:
  - H5A Attribute Interface.
  - H5D Dataset Interface.
  - H5F File Interface.
  - H5S Dataspace Interface.
- HDF5 High Level APIs.
  - HDF5 Lite ([H5LT](#)) – simplifies steps in creating datasets and attributes
  - HDF5 Image ([H5IM](#)) – defines a standard for storing images in HDF5
  - HDF5 Table ([H5TB](#)) – condenses the steps required to create tables
  - HDF5 Dimension Scales ([H5DS](#)) – standard for dimension scale storage
  - HDF5 Packet Table ([H5PT](#)) – provides a standard for storing packet data



# Introducción

- HDF5 provee de algunas herramientas:
  - *h5dump*:
    - A utility to dump or display the contents of an HDF5 File
  - *h5cc, h5c++, h5fc*:
    - Unix scripts for compiling applications
  - *HDFView*:
    - A java browser to view HDF (HDF4 and HDF5) files



# Introducción

- El paradigma general para trabajar con objetos en HDF5 es el siguiente:
  - Abrir el objeto → Acceder al objeto → Cerrar el objeto.
- El API impone un orden sobre las operaciones mediante dependencias en sus argumentos, por ejemplo:
  - Un archivo se debe abrir antes del *dataset* por que para abrir un *dataset* se requiere de un descriptor de archivo.
- Los objetos se pueden cerrar en cualquier orden. Una vez que el objeto se cierra ya no se puede acceder a él.

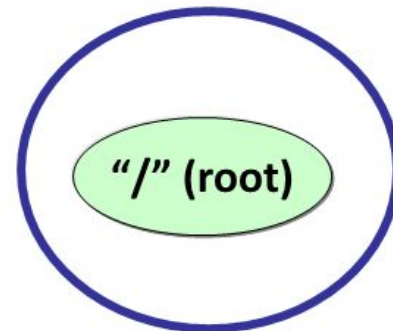


# Creación de un archivo

1. Especificar una lista de propiedades (o usar las de *default*).
2. Crear el archivo.
3. Cerrar el archivo (y la lista de propiedades si es necesario).

```
import h5py
import numpy as np
file = h5py.File('file.h5', 'w')
file.close()
```

file.h5

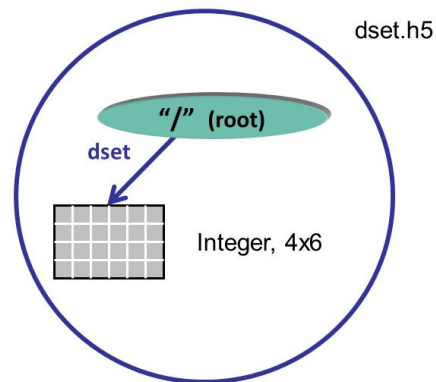




# Creación de un *dataset*

1. Definir las características del dataset (datatype, dataspace, properties).
2. Decidir a qué grupo adjuntar el dataset.
3. Crear el dataset.
4. Cerrar el dataset.

```
dataset=file.create_dataset("dset", (4, 6),  
                             h5py.h5t.STD_I32BE)
```



# Lectura y escritura de y desde un *dataset*

```
data = np.zeros((4,6))

for i in range(4):
    for j in range(6):
        data[i][j]= i*6+j+1

# Write data to dataset
dataset[...] = data

# Read data from dataset
data_read = dataset[...]
```

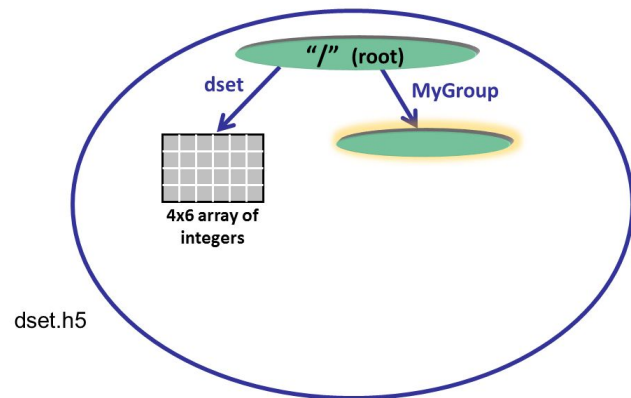




# Creación de un grupo

1. Decidir dónde ubicar el grupo. Abrir el grupo si no está abierto de antemano.
2. Definir las propiedades o usar las de *default*.
3. Crear el grupo.
4. Cerrar el grupo.

```
file = h5py.File('dset.h5', 'r+')
group = file.create_group('MyGroup')
file.close()
```



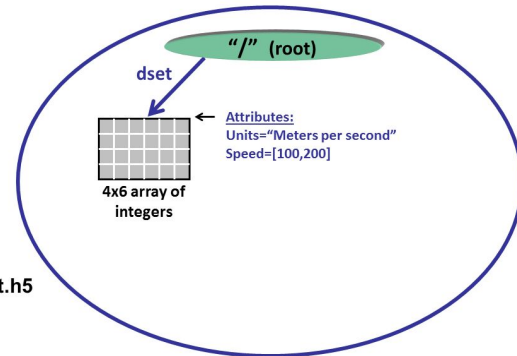
dset.h5



# Creación y escritura de un atributo

1. Abrir un objeto al que se desee agregar un atributo.
2. Crear el atributo.
3. Escribir el atributo.
4. Cerrar el atributo y el objeto al cual está ligado.

```
dataset.attrs["Units"] = "Meters per second"
attr_data = np.zeros((2,))
attr_data[0] = 100
attr_data[1] = 200
dataset.attrs.create("Speed", attr_data,
                    (2,), "i")
```



dset.h5

