

DATA SCIENCE AND MACHINE LEARNING APPLIED TO FINANCIAL MARKETS

Modulo III

CIENCIA DE DATOS

Profesor: Act. Fernando Ortega
Camargo

Motivación

Ciencia de datos

¿Qué es la ciencia?

Se puede denominar ciencia al conjunto de conocimientos obtenidos mediante la observación y el razonamiento sistemáticamente estructurados y de los que se deducen leyes y principios con carácter general con capacidad predictiva y que son comprobables experimentalmente.

La ciencia ofrece soluciones para los principales problemas de la vida y nos ayuda a responder a los grandes cuestionamientos de la humanidad. Es decir, es una de las vías más importantes de acceso al **conocimiento**.

¿Qué es la ciencia?

En la ciencia se determina un camino o modo de hacer las cosas en orden, es decir, un procedimiento de deducir la verdad y enseñarla. Es precisamente este “método”, el que asegura los alcances de la ciencia así como su proyección. En pocas palabras que los conocimientos no mueran.

“Metodizar” es asegurar la posibilidad de transmitir.

El método antes mencionado normalmente se denomina “científico” y consiste en la observación sistemática, la medición, la experimentación, la formulación, el análisis y la modificación de las hipótesis

Reproductibilidad



Refutabilidad



Pilares del método científico

¿Qué es la ciencia?

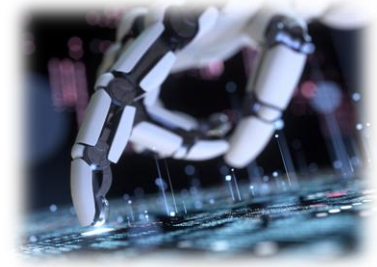
El método científico se basa en otros métodos como los clasificatorios, estadísticos, métodos hipotético-deductivos, etc. , por lo que referirse a el método científico es referirse a este conjunto de tácticas empleadas para constituir el conocimiento.

¿Cuál es el deber de un científico cuando se enfrenta a posibilidades imposibles?
Recopilar datos y estudios de acuerdo con el método científico.

*Frase de "**El astronauta de Bohemia**"
(2016), Jaroslav Kalfar*



Etapas del método científico



1. Observación

El primer paso será siempre la observación. Esta se podrá llevar a cabo de diferentes maneras como por ejemplo, con los sentidos o mediante herramientas que nos ayuden a mejorar la percepción de la realidad observada, por ejemplo los microscopios o telescopios.

2. Formulación de la hipótesis

La hipótesis es la explicación que se da a partir de las observaciones realizadas.

3. Experimentación

Una vez que se ha formulado la hipótesis, se llevará a cabo la fase de experimentación, **cuyo objetivo principal no será probar esta hipótesis, sino refutarla.**

4. Emisión de conclusiones y teoría

Una vez que se haya obtenido una hipótesis que sea imposible refutar*, se presentarán las conclusiones y se formulará la teoría correspondiente a las mismas, que constituirá un nuevo conocimiento científico hasta que se demuestre lo contrario.

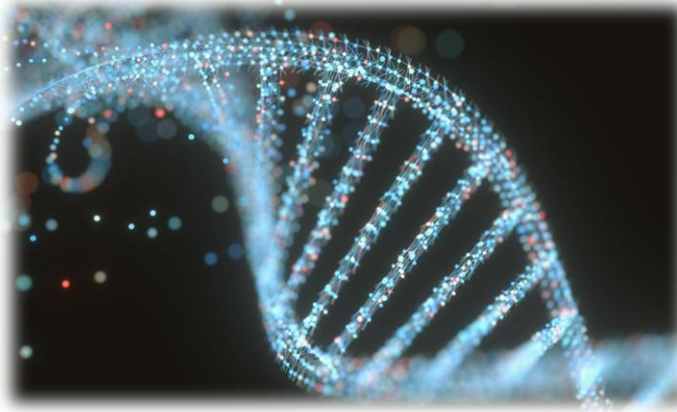
Etapas del método científico

5. Publicación y comparación

Consiste en publicar y compartir las conclusiones obtenidas, con el propósito de primeramente ampliar el conocimiento general de la sociedad y como segundo punto para que otros científicos puedan revisar y estudiar el hallazgo.

6. Ley

En el caso de que la teoría pueda ser demostrada mediante nuevas experimentaciones, confirmando su carácter irrefutable pasará a convertirse en ley. En este caso, se trata de una certeza basada en la experiencia tanto de las observaciones como de los experimentos y el estudio teórico.



¿y el conocimiento?

El conocimiento se podría definir como la conciencia o la comprensión de algo, como pueden ser los hechos (conocimiento descriptivo) y las habilidades (conocimiento procedimental). En general, el conocimiento puede adquirirse de muchas maneras y a partir de muchas fuentes, como la percepción, la razón, la memoria, la investigación científica y la práctica.

El conocimiento puede existir en 4 fases dentro de nosotros: Punto Ciego, Aprendizaje, Aplicación y Encarnación.

¿y el conocimiento?

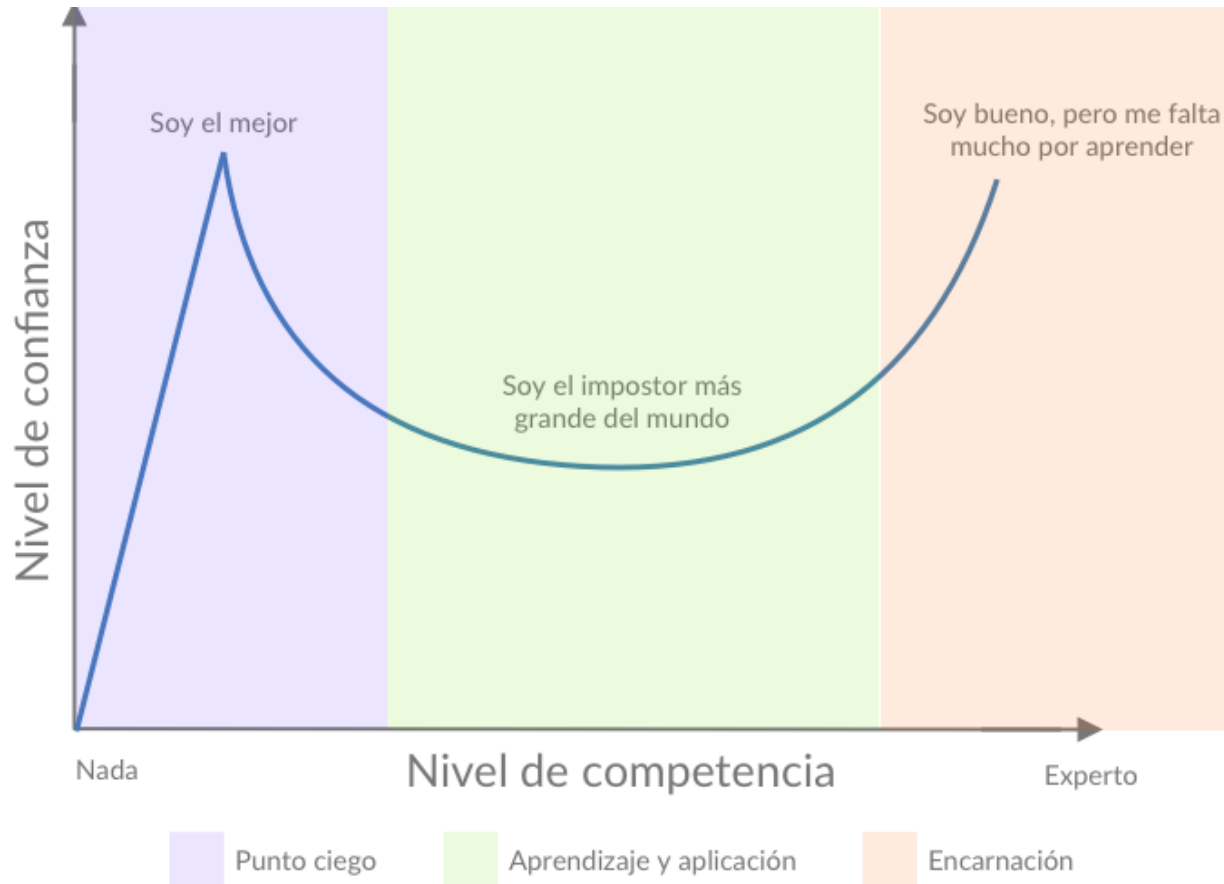
1.Punto Ciego: Inconsciente. La persona ignora lo que ignora, es decir, asume y supone pero no se cuestiona el por que de las cosas, simplemente acepta la realidad.

1.Aprendizaje: Consciente. La persona se da cuenta de su ignorancia y comienza a buscar de manera consciente expandir su conocimiento , empieza a estudiar, investigar, hacer preguntas y se vuelve mas receptiva a nuevas ideas.

1.Aplicación: Consciente. La persona comienza a interiorizar los aprendizajes de la etapa pasada, toma lo que ha estudiado y aprendido y lo aplica para terminar de asimilar el conocimiento, esta aplicación genera mas preguntas.

Encarnación: Inconsciente. La persona logra dominar su tarea y ahora la puede ejecutar sin pensar, es decir, logra aplicar su conocimiento de manera inconsciente, en esta fase es donde el conocimiento se vuelve sabiduría.

¿y el conocimiento?



Ciencia de datos

La ciencia de datos es el estudio de datos con el fin de extraer información significativa para empresas. Es un enfoque multidisciplinario que combina principios y prácticas del campo de las matemáticas, la estadística, la inteligencia artificial y la ingeniería de computación para analizar grandes cantidades de datos. Este análisis permite que los científicos de datos planteen y respondan a preguntas como “qué pasó”, “por qué pasó”, “qué pasará” y “qué se puede hacer con los resultados”.¹

Cuestionamiento Observación

Obtener Datos

Depurar datos

Explorar datos Hipótesis

Reproductibilidad

Refutabilidad

Modelar datos

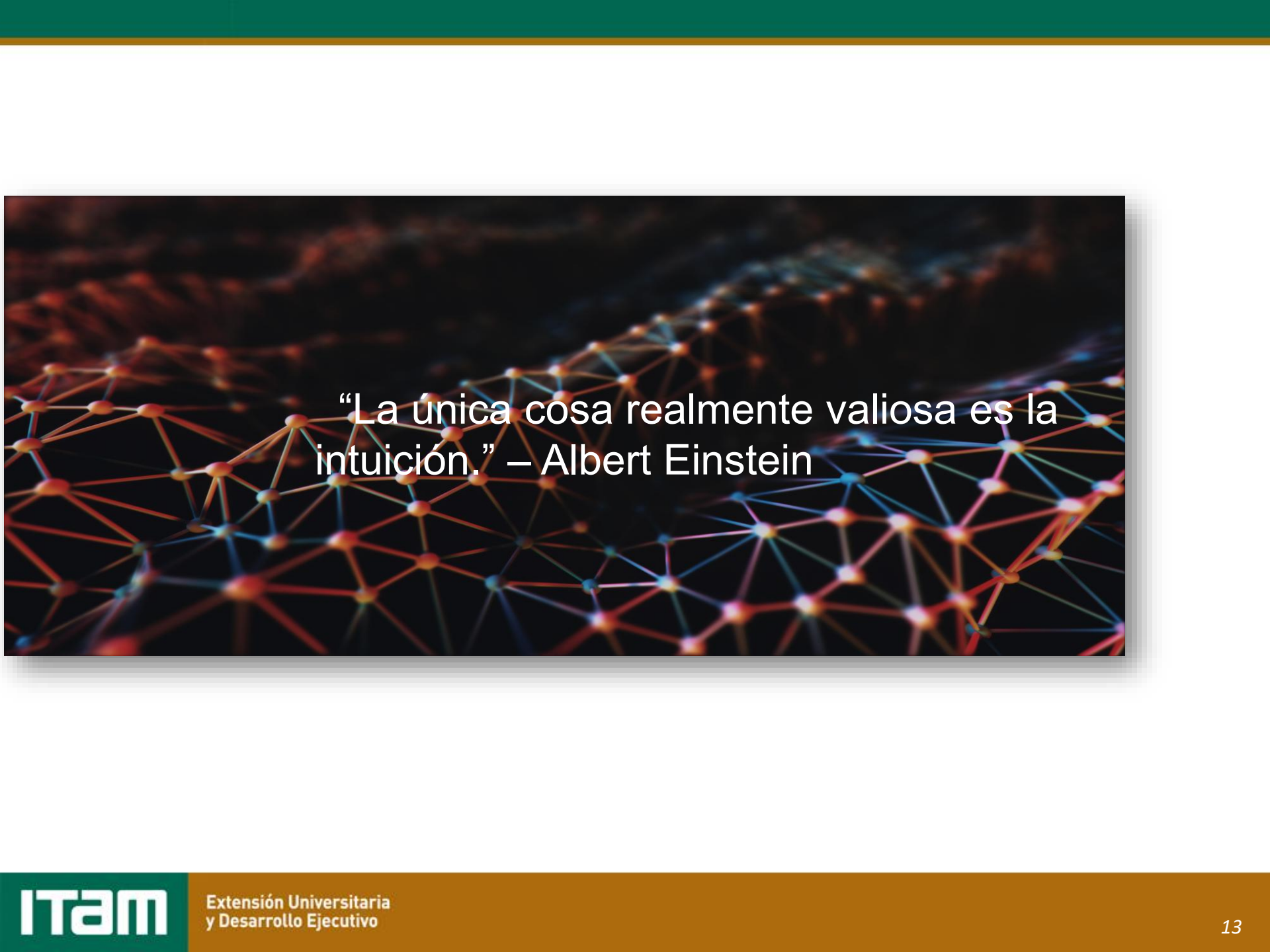
Experimentación

Interpretar datos

Conclusiones



1. Definición De AWS: <https://aws.amazon.com/es/what-is/data-science/>

An abstract background image featuring a complex network of glowing nodes and connecting lines, resembling a molecular structure or a data network. The nodes are small spheres in various colors (red, blue, yellow, green) and are interconnected by thin, translucent lines. The overall effect is a sense of depth and connectivity, with some nodes appearing brighter than others.

“La única cosa realmente valiosa es la intuición.” – Albert Einstein

Tema 1

BASES DE DATOS Y SQL

¿Para qué sirven las bases de datos?

Las bases de datos son arquitecturas que sirven para almacenar información y para que los usuarios de esta puedan interactuar con ella, ya sea para recuperarla y/o actualizarla. Esta información puede ser de cualquier tipo y de cualquier tema mientras sea significativo para el individuo u organización que la requiera.



Las bases de datos son de vital importancia en casi todas las áreas donde la informática sea requerida como por ejemplo, negocios de correo electrónico, medicina, genética, educación, ciencia, finanzas, administración etc.

Ventajas

Compacta: Su almacenamiento no requiere tantos recursos físicos como es el papel.

Rapidez: La máquina puede recuperar y actualizar los datos más rápido que un humano.

Menos trabajo: Eliminan el proceso de administrar archivos manualmente(automatización)

Frecuencia: La información está disponible en cualquier momento.

Protección: Los datos pueden estar mejor protegidos contra la pérdida no intencional y algún problema de seguridad .



Desventajas

Alta complejidad: Administrar una base de datos conlleva un grupo de programas complejos los cuales deben ser comprendidos en su totalidad para una correcta funcionalidad.

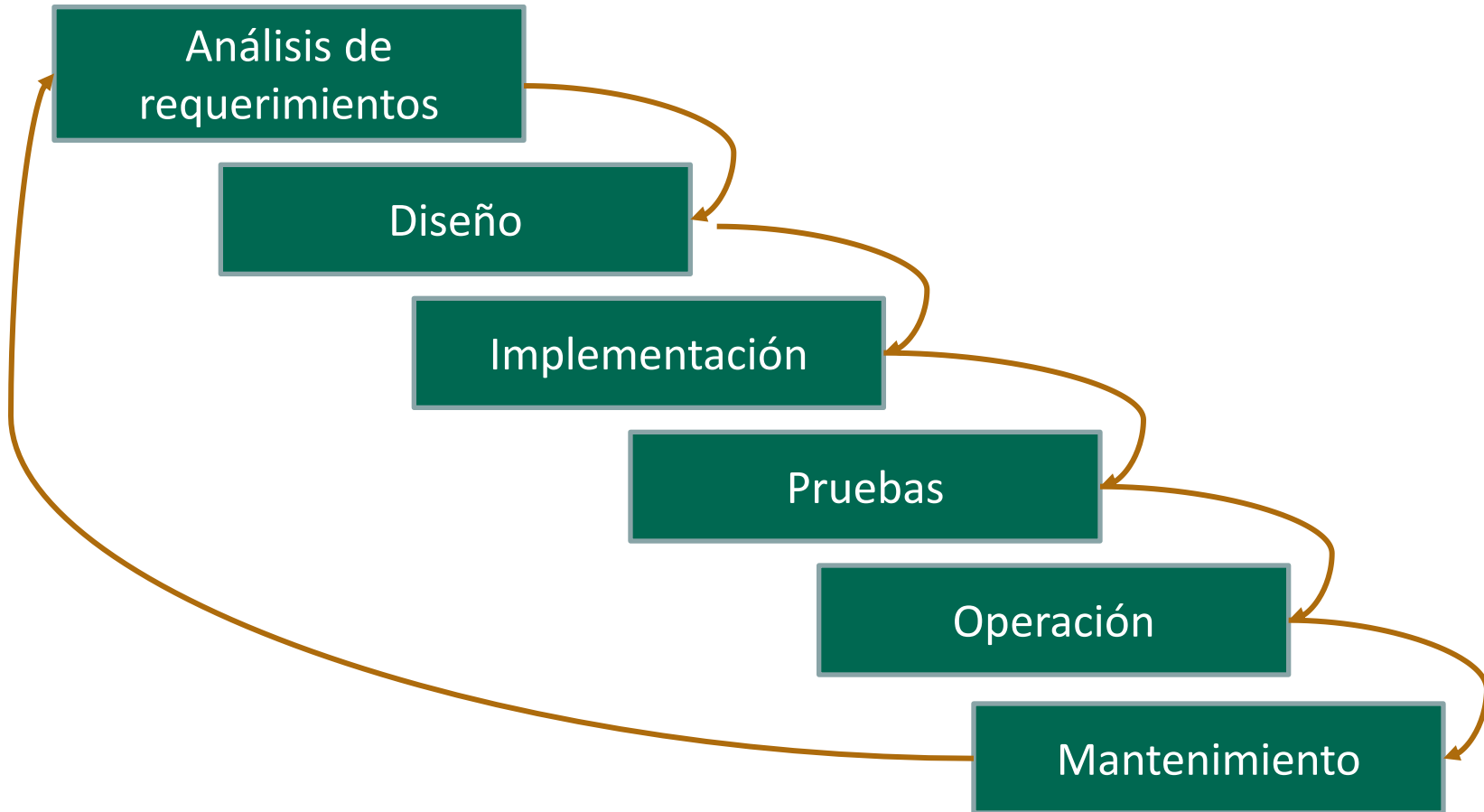
Gran tamaño: Se requiere infraestructura capaz de soportar una gran cantidad de espacio en disco y memoria para optimizar los procesos.

Costo: El costo de implementación normalmente depende del entorno y funcionalidad de la misma, incluyendo mantenimientos periódicos, con lo que los costos de implementación se pueden incrementar notoriamente.

Vulnerabilidad: el hecho de que todos los datos estén centralizados en una sola arquitectura hace que el sistema sea más vulnerable ante los fallos o actividades ilícitas.

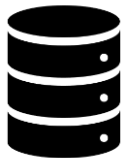


Ciclo de vida de una base



Conceptos básicos

- Dato
- Información
- Sistema de información
- Base de datos
- Sistemas manejadores de bases de datos



Conceptos básicos

-Dato

Un dato es una representación simbólica (numérica, alfabética, algorítmica, espacial, etc.) de un atributo o variable cuantitativa o cualitativa. Valor Atómico.

-Información

Se conoce como información al conjunto organizado de datos procesados que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje

Conceptos básicos

-Sistema de información

Se refiere a un conjunto ordenado de mecanismos que tienen como fin la administración de datos y de información, de manera que puedan ser recuperados y procesados fácil y rápidamente. (Grupo de información)

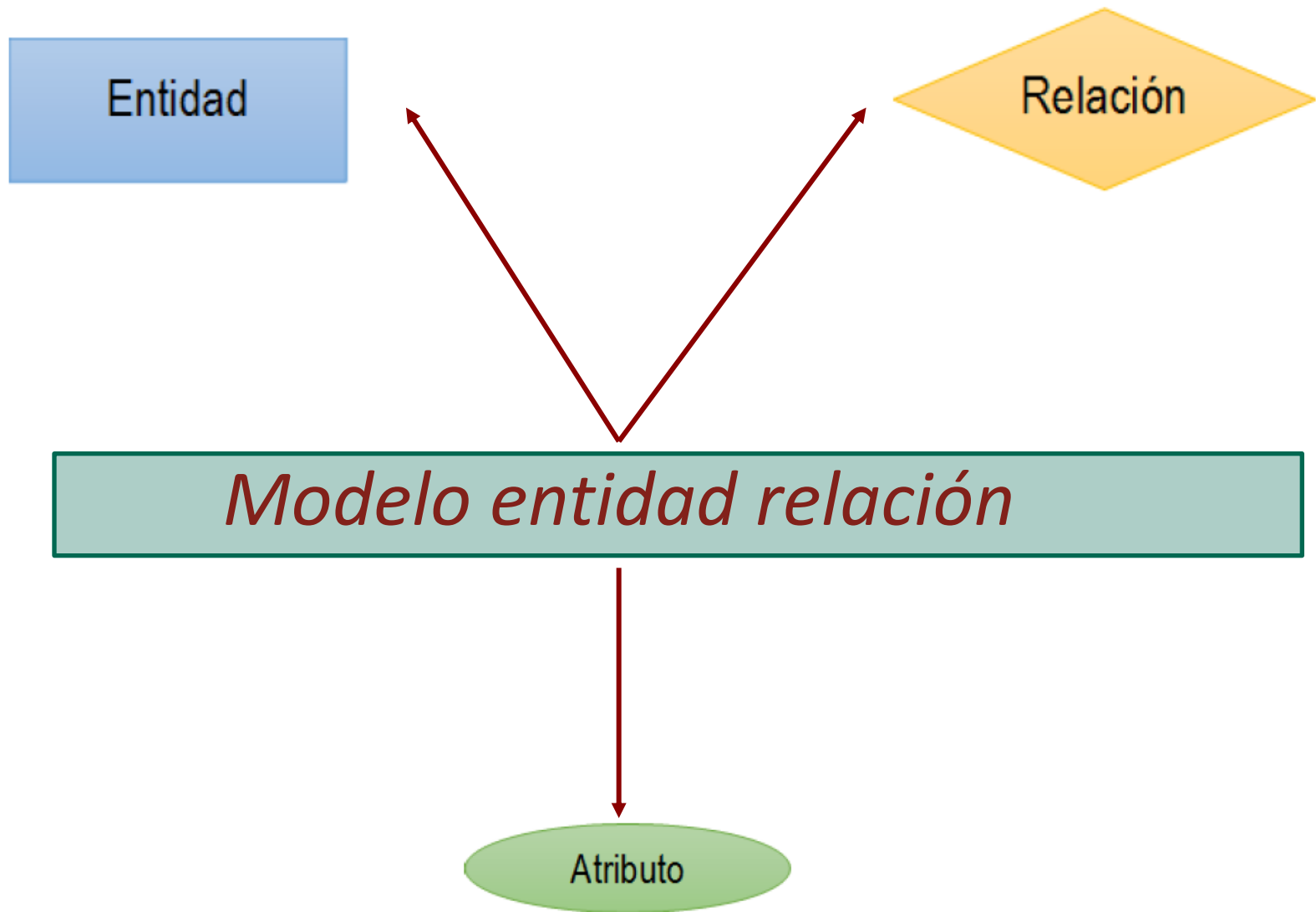
-Base de datos

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático.

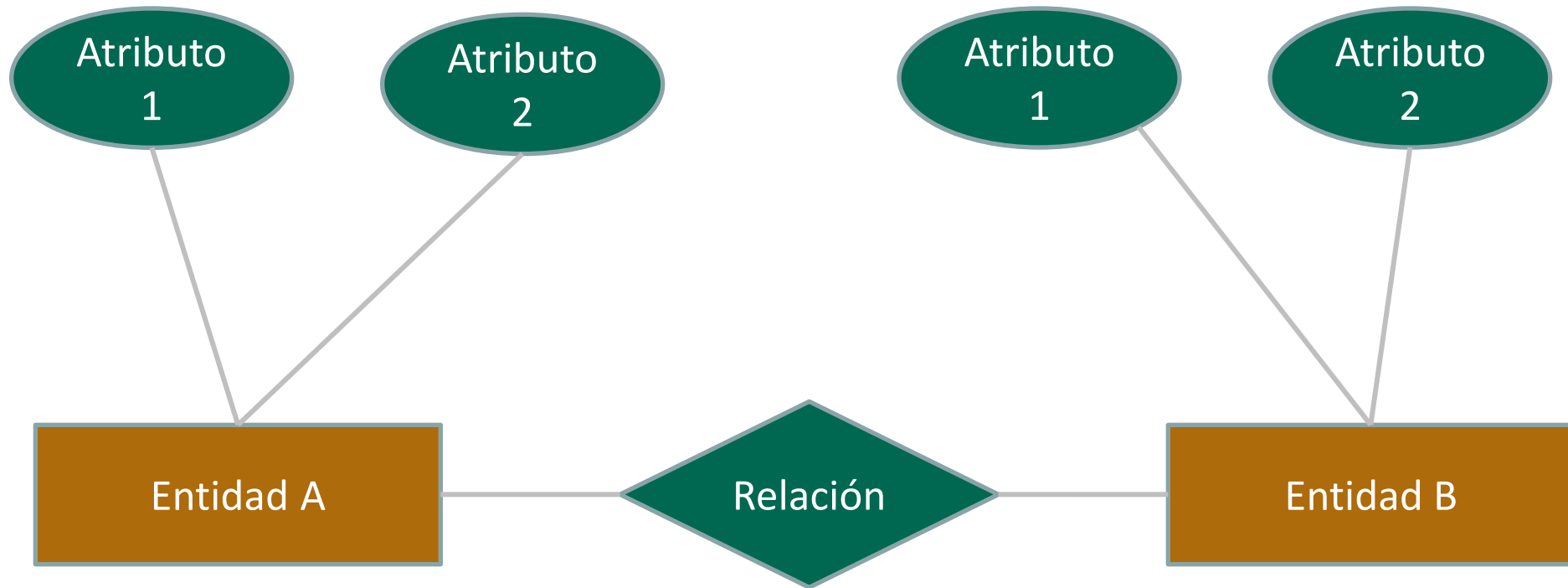
Conceptos básicos

-Sistemas manejadores de bases de datos

Un sistema manejador de bases de datos es una colección de software, orientado al manejo de base de datos, cuya función es servir de interfaz entre la base de datos, el usuario y las distintas aplicaciones utilizadas.



Modelo Entidad Relación



Mapeo y tuplas

Tabla Entidad

1

Entidad 1		
Atributo 1	Tipo de atributo1	Restricción A1
Atributo 2	Tipo de atributo2	Restricción A2
Atributo 3	Tipo de atributo3	Restricción A3
Atributo 4	Tipo de atributo4	Restricción A4
Atributo 5	Tipo de atributo5	Restricción A5

Tupla Entidad

1

Entidad 1			
Elemento	Atributo 1	Atributo 2	Atributo 3
1	Atributo 1 E1	Atributo 2 E1	Atributo 3 E1
2	Atributo 1 E2	Atributo 2 E2	Atributo 3 E2
3	Atributo 1 E3	Atributo 2 E3	Atributo 3 E3
4	Atributo 1 E4	Atributo 2 E4	Atributo 3 E4
5	Atributo 1 E5	Atributo 2 E5	Atributo 3 E5

GROUP BY

CREATE
DATABASE

UPDATE

Lenguaje de definición y manipulación

WHERE

CREATE TABLE

INSERT
INTO

SELECT
FROM

DELETE

Lenguaje de Definición de Datos (DDL) básico

**Creación de base
de datos**

`CREATE DATABASE ejemplo1;`

**Creación de
tablas**

`CREATE TABLE mi_primera_tabla(
primera_columna numeric,
segunda_columna text);`

Lenguaje de Definición de Datos (DDL) integridad

**Llaves
primarias**

`CREATE TABLE tabla1(
columna1 integer,
columna2 varchar(50),
PRIMARY KEY (columna1));`

Lenguaje de Definición de Datos (DDL)

integridad

Llaves foraneas

```
CREATE TABLE tabla2(  
  columna3 integer,  
  columna4 integer,  
  columna5 varchar(15),  
  FOREIGN KEY(columna4) REFERENCES tabla_1(columna1));
```

Dominio

```
CREATE TABLE Cliente(  
  ID_cliente integer,  
  Apellidos varchar(255),  
  Nombre varchar(255),  
  Genero char(1),  
  CHECK(Genero IN ('M', 'F')));
```

Existen otros casos de condición como: (A = B), (C <> D), (E NOT IN ('E', 'F', ..., 'X')), (X < 0), etc

Lenguaje de Definición de Datos (DDL)

integridad

No nulidad

```
CREATE TABLE Cliente1(  
  ID_Cliente INTEGER NOT NULL,  
  Apellidos VARCHAR(255) NOT NULL,  
  Nombre VARCHAR(255) NOT NULL);
```

Modificación de
restricciones

```
ALTER TABLE nombre_tabla  
DROP CONSTRAINT nombre_de_la_cláusula;  
ALTER TABLE nombre_tabla  
ADD CHECK (nombre_columna IN ('M','F'));
```

Lenguaje de manipulación de Datos (DML)

**Inserción de
datos**

`INSERT INTO nombre_tabla (columna1, columna2, ..., columnaN)
VALUES (valor1, valor2, ..., valorN);`

**Consulta
general**

`SELECT nombre_columna
FROM nombre_tabla
WHERE condicion;`

Actualización


`UPDATE nombre_tabla
SET columna_1 = nuevo_valor
WHERE condición;`

Actualización

`DELETE FROM nombre_tabla
WHERE condicion;`


Lenguaje de manipulación de Datos (DML) avanzado

Consultas agrupadas



```
SELECT nombre_columna1, nombre_columna2, ...  
FROM nombre_tabla  
WHERE condición GROUP BY atributo_agrupador  
HAVING condición_de_agrupación;
```

Funciones de agregación



```
SELECT Fecha, SUM(Total)  
FROM Orden  
GROUP BY Fecha;
```

Operadores y funciones útiles

BETWEEN Compara si un valor se encuentra dentro de los parámetros que el usuario puede definir

LIKE Compara si un valor cumple con un patrón que el usuario puede definir

SIMILAR TO Compara si un valor cumple con un patrón que el usuario puede definir de manera más general

IN Compara si un valor existe dentro de una lista de posibles valores definida por el usuario

NOT IN Compara si un valor no existe dentro de una lista de posibles valores definida por el usuario

And y Or Operadores lógicos

= Compara si dos valores son iguales

< Compara si el valor de la izquierda es menor que el de la derecha

> Compara si el valor de la izquierda es mayor que el de la derecha

< > Compara si dos valores son diferentes

Tipos de datos en SQL

Grupo	Tipo de dato	Intervalo	Almacenamiento
Numéricos exactos	bigint	De -2^{63} (-9.223.372.036.854.775.808) a $2^{63} - 1$ (9.223.372.036.854.775.807)	8 bytes
	int	De -2^{31} (-2.147.483.648) a $2^{31} - 1$ (2.147.483.647)	4 bytes
	smallint	De -2^{15} (-32.768) a $2^{15} - 1$ (32.767)	2 bytes
	tinyint	De 0 a 255	1 byte
	bit	Tipo de datos entero que puede aceptar los valores 1, 0 ó NULL	2 bytes
	decimal, numeric, decimal (p, s)	<ul style="list-style-type: none"> p (precisión): el número total máximo de dígitos decimales que se puede almacenar, tanto a la izquierda como a la derecha del separador decimal. La precisión debe ser un valor comprendido entre 1 y la precisión máxima de 38. La precisión predeterminada es 18. s (escala): el número máximo de dígitos decimales que se puede almacenar a la derecha del separador decimal. La escala debe ser un valor comprendido entre 0 y p. Sólo es posible especificar la escala si se ha especificado la precisión. La escala predeterminada es 0. <p>Con precisión máxima $10^{38} + 1$ y $10^{38} - 1$</p>	Precisión 1 - 9: 5 bytes
	money	Tipos de datos que representan valores monetarios o de moneda: de -922.337.203.685,4775808 a 922.337.203.685,4775807	8 bytes
	smallmoney	De - 214,7483648 a 214,7483647	4 bytes

Tipos de datos en SQL

Numéricos aproximados	float	De - 1,79E+308 a -2,23E-308, 0 y de 2,23E-308 a 1,79E+308	Depende del valor de n
	real	De - 3,40E + 38 a -1,18E - 38, 0 y de 1,18E - 38 a 3,40E + 38	4 Bytes
Fecha y hora	datetime	Del 1 de enero de 1753 hasta el 31 de diciembre de 9999	
	smalldatetime	Del 1 de enero de 1900 hasta el 6 de junio de 2079	
Cadenas de caracteres	char (n)	Caracteres no Unicode de longitud fija, con una longitud de n bytes. n debe ser un valor entre 1 y 8.000	n bytes
	varchar (n)	Caracteres no Unicode de longitud variable. n indica que el tamaño de almacenamiento máximo es de $2^{31} - 1$ bytes	n bytes (aprox.)
	text	En desuso, sustituido por <i>varchar</i> . Datos no Unicode de longitud variable con una longitud máxima de $2^{31} - 1$ (2.147.483.647) caracteres	max bytes (aprox.)
Cadenas de caracteres unicode	nchar (n)	Datos de carácter Unicode de longitud fija, con n caracteres. n debe estar comprendido entre 1 y 4.000	$2 * n$ bytes
	nvarchar (n)	Datos de carácter Unicode de longitud variable. n indica que el tamaño máximo de almacenamiento es $2^{31} - 1$ bytes	$2 * n$ bytes + 2 bytes
	ntext (n)	En desuso, sustituido por <i>nvarchar</i> . Datos Unicode de longitud variable con una longitud máxima de $2^{30} - 1$ (1.073.741.823) caracteres	$2 * n$ bytes

Tipos de datos en SQL

Cadenas binarias	binary (n)	Datos binarios de longitud fija con una longitud de n bytes, donde n es un valor que oscila entre 1 y 8.000	n bytes
	varbinary (n)	Datos binarios de longitud variable. n indica que el tamaño de almacenamiento máximo es de $2^{31} - 1$ bytes	n bytes
	image	En desuso, sustituido por <i>varbinary</i> . Datos binarios de longitud variable desde 0 hasta $2^{31} - 1$ (2.147.483.647) bytes	
Otros tipos de datos	cursor	Tipo de datos para las variables o para los parámetros de resultado de los procedimientos almacenados que contiene una referencia a un cursor. Las variables creadas con el tipo de datos <i>cursor</i> aceptan NULL	
	timestamp	Tipo de datos que expone números binarios únicos generados automáticamente en una base de datos. El tipo de datos <i>timestamp</i> es simplemente un número que se incrementa y no conserva una fecha o una hora	8 bytes
	sql_variant	Tipo de datos que almacena valores de varios tipos de datos aceptados en SQL Server, excepto <i>text</i> , <i>ntext</i> , <i>image</i> , <i>timestamp</i> y <i>sql_variant</i>	
	uniqueidentifier	Es un GUID (Globally Unique Identifier, Identificador Único Global)	16 bytes
	table	Es un tipo de datos especial que se puede utilizar para almacenar un conjunto de resultados para su procesamiento posterior. <i>table</i> se utiliza principalmente para el almacenamiento temporal de un conjunto de filas devuelto como el conjunto de resultados de una función con valores de tabla	
	xml	Almacena datos de XML. Puede almacenar instancias de xml en una columna o una variable de tipo xml	

Tipos de JOINS

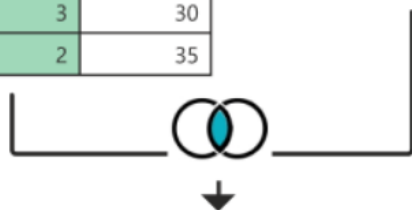
Inner

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	2	35

Right Table

ID	Country
3	Panama
4	Spain



Merged Table

Date	CountryID	Units	Country
1/3/2020	3	30	Panama

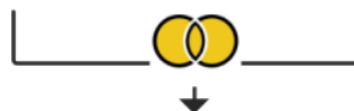
Full

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	2	35

Right Table

ID	Country
1	USA
2	Canada
3	Panama
4	Spain

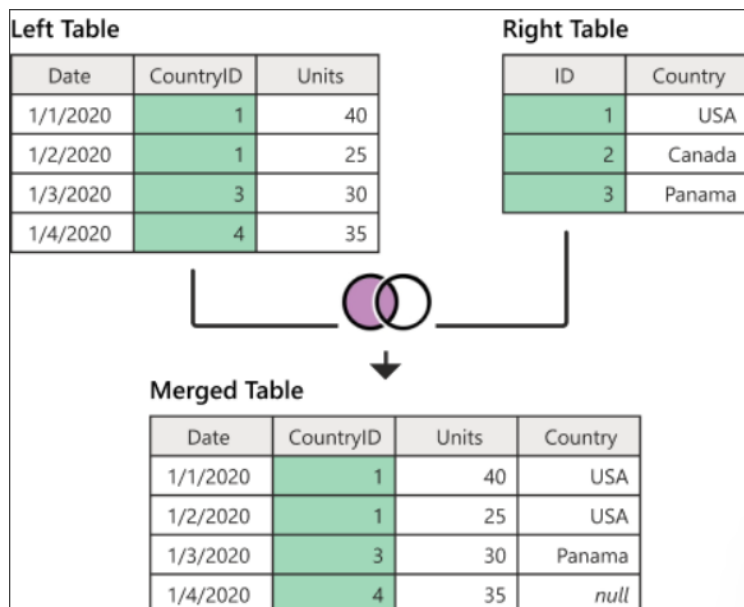


Merged Table

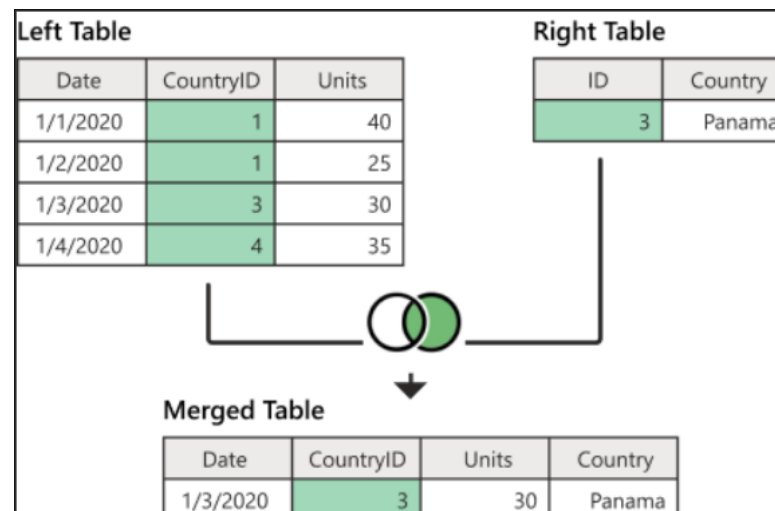
Date	CountryID	Units	Country
1/1/2020	1	40	USA
1/2/2020	1	25	USA
1/4/2020	2	35	Canada
1/3/2020	3	30	Panama
null	null	null	Spain

Tipos de JOINS

Left



Right

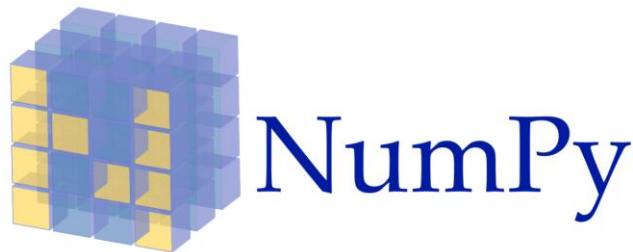


Tema 2

PYTHON CIENTÍFICO

NumPy

NumPy es el paquete fundamental para la computación científica en Python. Es una biblioteca de Python que proporciona un objeto de matriz multidimensional, varios objetos derivados (como matrices y matrices enmascaradas) y una variedad de rutinas para operaciones rápidas en matrices, incluida la manipulación matemática, lógica, de formas, clasificación, selección, transformadas discretas de Fourier, álgebra lineal básica, operaciones estadísticas básicas, simulación aleatoria y mucho más.



Documentación Completa:
<https://numpy.org/>

Matplotlib

Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python.

Entre sus posibilidades destacan:

- Creación de gráficos de calidad de publicación.
- figuras interactivas que puedan hacer zoom, desplazarse, actualizar.
- Personalización en diseño y el estilo visual.
- Posibilidad de exportación a muchos formatos de archivo.
- Compatibilidad con JupyterLab e interfaces gráficas de usuario.
- Amplia variedad de paquetes de terceros creados en Matplotlib.



Documentación Completa:
<https://matplotlib.org/>

SciPy

SciPy es una colección de algoritmos matemáticos y funciones de conveniencia creadas en la extensión NumPy de Python. Agrega un poder significativo a la sesión interactiva de Python al proporcionar al usuario comandos y clases de alto nivel para manipular y visualizar datos. Con SciPy, una sesión interactiva de Python se convierte en un entorno de procesamiento de datos y creación de prototipos de sistemas que rivaliza con los sistemas, como MATLAB, IDL, Octave, R-Lab y SciLab.



Documentación Completa:
<https://scipy.org/>

Tema 3

Introducción al análisis de datos

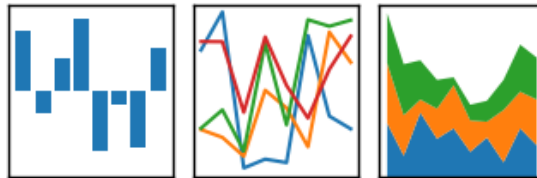
Pandas

Pandas es una librería de Python especializada en el manejo y análisis de estructuras de datos multidimensionales.

Las principales características de pandas son:

- Define nuevas estructuras de datos basadas en los arrays de la librería NumPy (pandas, series).
- Permite leer y escribir ficheros en formato CSV, Excel y bases de datos SQL así como conexiones a las mismas.
- Permite acceder a los datos mediante índices o nombres para filas y columnas, parecido a como se hace en las hojas de calculo.
- Ofrece métodos para reordenar, dividir y combinar conjuntos de datos.
- Permite trabajar con series temporales.
- Realiza todas estas operaciones de manera muy eficiente.

pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



Documentación Completa:
<https://pandas.pydata.org/>

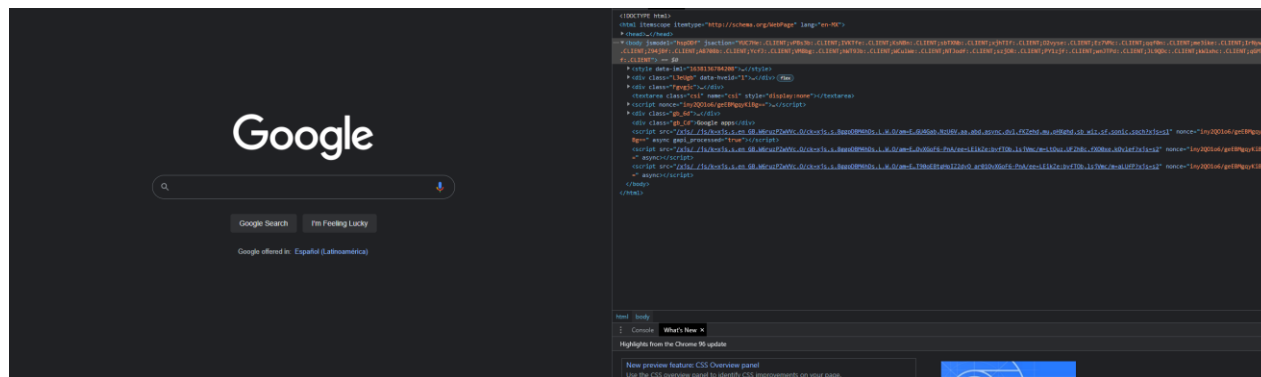
Tema 3.5

Webscrapping

Introducción a html

HTML (Lenguaje de Marcas de Hipertexto, HyperText Markup Language) es el componente más básico de la Web, es un lenguaje de marcado y es el más utilizado en todos los sitios web de la actualidad. Define el significado y la estructura del contenido web mediante etiquetas que servirán para la comunicación, formato y método de transferencia de información entre los distintos servidores.

Todas las paginas web que se conocen funcionan mediante un código html el cual puede ser accedido desde los navegadores predeterminados, por ejemplo , en el caso de Google Chrome, presionando click derecho y la opción de inspeccionar se puede analizar la estructura del html actual.



Introducción a html

Html funciona mediante etiquetas las cuales definirán la estructura de nuestra pag web, estas etiquetas harán referencia a títulos, párrafos, imágenes, hipervínculos dependiendo de la sintaxis que se utilice, de igual manera estas etiquetas definen el contenido.

El formato se dará mediante otra sintaxis que veremos mas adelante la cual tiene por nombre CSS (Cascading Style Sheets).

```
><style data-iml="1638136893753">...</style>
▼<div class="L3eUgb" data-hveid="1"> flex
  ><div class="o3j99 n1xJcf Ne6nSd">...</div> flex
... ><div class="o3j99 LLD4me yr19Zb LS80J">...</div> flex == $0
  ><div class="o3j99 ikrT4e om7nvf">...</div>
  ><div class="o3j99 qarstb">...</div>
  ><div class="o3j99 c93Gbe">...</div>
```

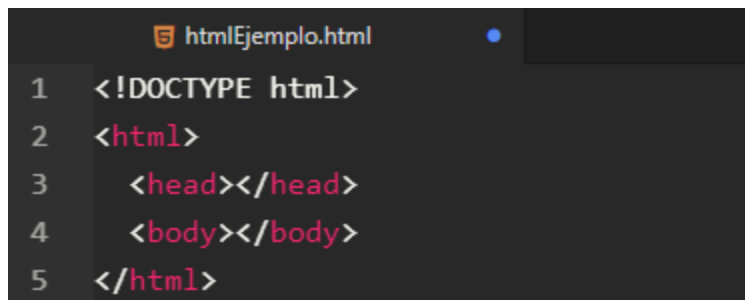
Por lo general la estructura básica de las etiquetas es la misma, mediante (< ó >) indicamos el inicio de una etiqueta, dentro de esto especificaremos distintos atributos, como pueden ser id's, clases, hipervínculos, etc), estos elementos estarán referenciados al css con lo que los elementos de una misma clase compartirán ciertos atributos de formato.

Introducción a html

Para empezar a escribir html, basta con escribir el código en cualquier editor de texto y simplemente guardarlo con la terminación .html , haciendo esto podemos empezar a crear una pagina web sencilla de manera local.

Es recomendable escribir el html en algún editor de código que tenga la sintaxis predefinida con lo que nos ahorrara trabajo de estar recordando todas las etiquetas posibles

Primeras etiquetas

A screenshot of a code editor window titled 'htmlEjemplo.html'. The editor shows five lines of HTML code with syntax highlighting: line 1 is '<!DOCTYPE html>', line 2 is '<html>', line 3 is ' <head></head>', line 4 is ' <body></body>', and line 5 is '</html>'. The code is written in a dark-themed editor with light-colored text.

```
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4   <body></body>
5 </html>
```

El inicio de nuestros códigos html siempre tendrá este formato, iniciamos nuestro código con html con la etiqueta de apertura (<>) y cierre (</>)

Dentro de las etiquetas especificaremos el contenido de la misma, hay que notar que el contenido de la pagina web siempre estará en las etiquetas de body, pero esto no implica que en otras etiquetas como head, no podamos especificar algo, simplemente en ellas no se verá reflejado el contenido en la pagina web pero si realizarán ciertas funciones, por ejemplo, el apartado de head dará el titulo a nuestra pagina, título que no estará plasmado internamente en el contenido de la pag web final.

```
<!DOCTYPE html>
<html>
  <head> Titulo que no aparece en la pag</head>
  <body> Contenido que si aparece en la web </body>
</html>
```

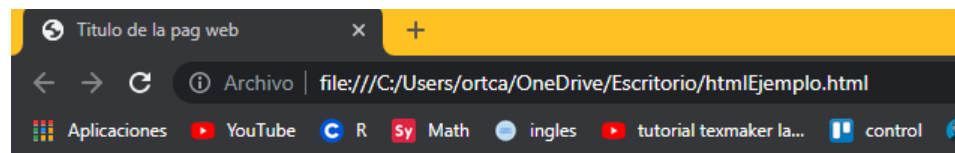
```
<!DOCTYPE html>
<html>
  <head> <title>Titulo que no aparece en la pag</title></head>
  <body> Contenido que si aparece en la web </body>
</html>
```


Títulos

Se especificarán mediante la etiqueta h y la numeración de orden

`<h1>Titulo 1</h1>`

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titulo de la pag web</title>
  </head>
  <body>
    <h1>Titulo 1</h1>
    <h2>Titulo 2</h2>
    <h3>Titulo 3</h3>
  </body>
</html>
```



Titulo 1

Titulo 2

Titulo 3

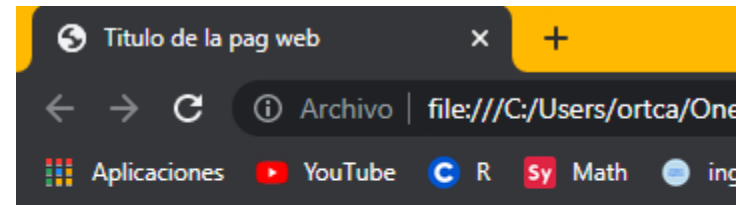
Párrafos

Se especificarán mediante la etiqueta p

`<p>Párrafos </p>`

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titulo de la pag web</title>
  </head>
  <body>
    <p> Hola, soy un parrafo</p>
    <p>Hola, soy otro parrafo</p>
  </body>
</html>
```

Cada que escriba un párrafo diferente este hará un salto de línea, si yo quisiera un salto de línea dentro de un mismo párrafo utilizare la etiqueta `
 </br>`



Hola, soy un parrafo

Hola, soy otro parrafo

Comentarios

Se utilizara dentro de una etiqueta <!-- comentario -->

```
<html>
  <head>
    <title>Titulo de la pag web</title>
  </head>
  <body>
    <p> Hola, soy un parrafo</p>
    <p>Hola, soy otro parrafo</p>
    <!-- <p>Esto es un comentario </p> -->
  </body>
</html>
```

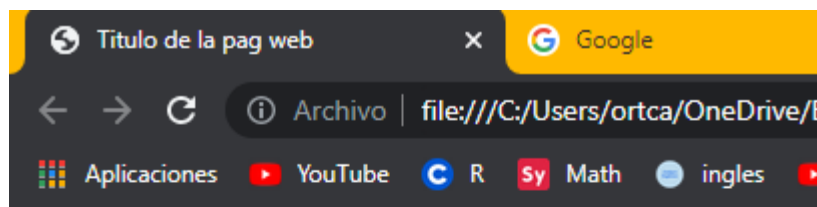
Teniendo este formato estándar en html, dentro de títulos o párrafos se pueden modificar algunos elementos de los textos, por ejemplo, tamaños, fuentes, etc.

Veremos a resumen estos elementos.

Formatos de fuentes

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titulo de la pag web</title>
  </head>
  <body>
    <p> <b>Hola, soy un parrafo en negrita</b></p>
    <p><code>Hola, soy otro parrafo con formato código</code></p>
    <p><em>Hola soy un parrafo en cursiva</em></p>
    <p><s>Hola soy un parrafo tachado</s></p>
    <p><small>Hola soy un parrafo pequeño</small></p>
    <p><strong>Hola soy un parrafo en negrita</strong></p>
    <p><u>Hola soy un parrafo subrayado</u></p>
    <!-- <p>Esto es un comentario </p> -->
  </body>
</html>
```

Formatos de fuentes



Hola, soy un parrafo en negrita

Hola, soy otro parrafo con formato código

Hola soy un parrafo en cursiva

~~Hola soy un parrafo tachado~~

Hola soy un parrafo pequeño

Hola soy un parrafo en negrita

Hola soy un parrafo subrayado

Divisiones o secciones

Estas divisiones son como “cajas” las cuales estarán en un bloque que no servirá para dividir nuestro código html, su sintaxis es la siguiente:

`<div>Párrafos </div>`

Estas divisiones sirven para particionar en bloques nuestro html con el fin de darle mas orden cuando ciertos elementos comparten características similares.

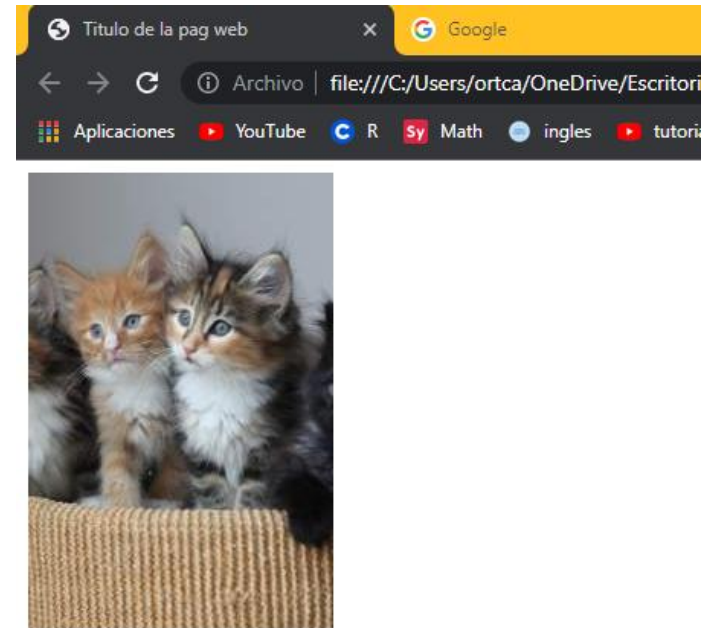
```
<!DOCTYPE html>
<html>
  <head>
    <title>Titulo de la pag web</title>
  </head>
  <body>
    <div>sección 1</div>
    <div>sección 2</div>
    <div>sección 3</div>

  </body>
</html>
```

Imágenes

Estas se insertarán mediante la etiqueta `img` y una url asociada a una imagen o recurso que se quiera insertar, para nuestro ejemplo visitaremos un sitio web que permite obtener imágenes aleatorias

```
<html>
  <head>
    <title>Titulo de la pag web</title>
  </head>
  <body>
    
  </body>
</html>
```



link

Dentro de nuestros párrafos o imágenes podemos usar la etiqueta a , la cual nos redireccionara a un link que nosotros queramos.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titulo de la pag web</title>
  </head>
  <body>
    
    <a href="https://www.google.com/"> Ir a google para mas info </a>
  </body>
</html>
```



[Ir a google para mas info](https://www.google.com/)

Introducción a css

CSS es un lenguaje que nos permitirá trabajar en conjunto con html para darle el formato deseado a nuestra pagina web, css es dependiente de html, contrario de este ultimo que puede existir por si solo.

Se denomina hojas de estilo en cascada por que puede tener varias hojas o “cajas” y cada una de ellas propiedades heredadas en forma de cascada de otras.

Con css se crean las reglas para decirle a nuestra pag como queremos mostrar nuestra información, y como vimos anteriormente mediante las clases o los ids podremos generalizar formatos a todos los elementos que compartan sus etiquetas.

Su sintaxis es muy parecida a los diccionarios en Python, donde dentro especificaremos, las etiquetas que queremos afectar y el atributo que queremos modificar. Se tiene que crear otro script y vincularlo en nuestro html de la siguiente forma.

Introducción a css

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo CSS</title>
    <link rel="stylesheet" href="main.css" type="text/css">
  </head>
  <body>
    <p>Parrafo</p>
    <spa>Esto es un span</spa>

  </body>
</html>
```

```
body {
  background-color: #fff000;
}
```

Parrafo

Esto es un span

Selectores Id y Class

Como vimos, parte importante del código html son las etiquetas , dentro de las cuales podemos especificar su id y clase, esto nos servirá para relacionar estos selectores a un apartado del css donde le daremos ciertas propiedades de formato a nuestras “cajas” dependiendo de si comparten su clase , o de manera individual por id.

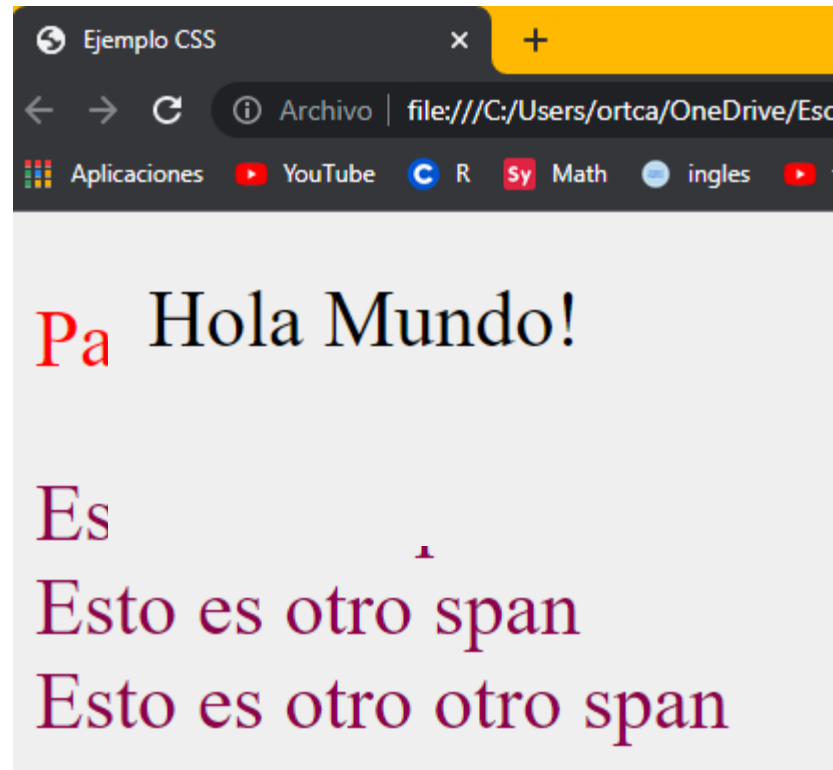
```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo CSS</title>
    <link rel="stylesheet" href="main.css" type="text/css">
  </head>
  <body>
    <p id="parrafo">Parrafo</p>
    <spa class="clase">Esto es un span</spa>
    <br/>
    <span class="clase">Esto es otro span</span>
    <br/>
    <span class="clase">Esto es otro otro span</span>
  </body>
</html>
```

```
body {
  background-color: #efefef;
}

#parrafo{
  color: #ff0000;
}

.clase{
  color: #8C004B;
}
```

Selectores Id y Class

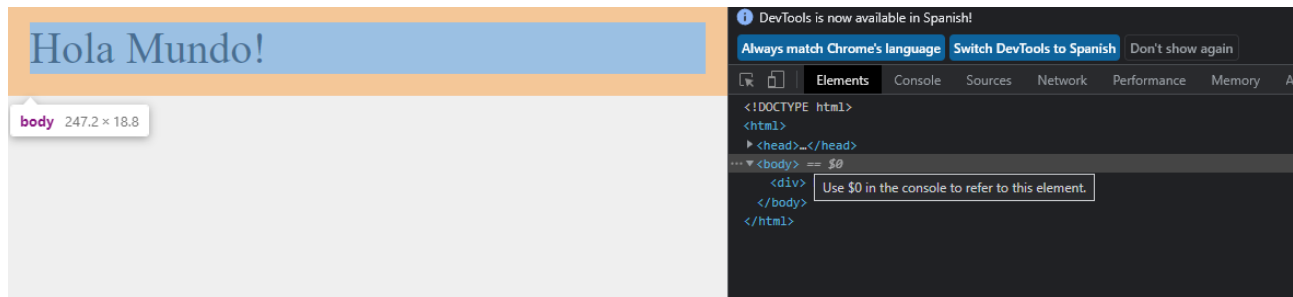


Modelo de caja

En html todas nuestras etiquetas son consideradas como “cajas” , bajo el concepto tradicional, hablar del modelo de caja, implica el cómo se modelarán los bloques de etiquetas en nuestra pagina.

Una manera fácil de ver este concepto es con el apartado inspeccionar como sigue

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo CSS</title>
    <link rel="stylesheet" href="main.css" type="text/css">
  </head>
  <body>
    <div>
      Hola Mundo!
    </div>
  </body>
</html>
```



Elementos

Márgenes

```
body {  
  background-color: #efefef;  
}  
  
.caja{  
  margin: 8px;  
}
```

Hola Mundo!

Hola Mundo!

Hola Mundo!

Elementos

Márgenes

```
.caja{  
  background-color: #00f;  
  margin: 8px;  
}
```



Hola Mundo!

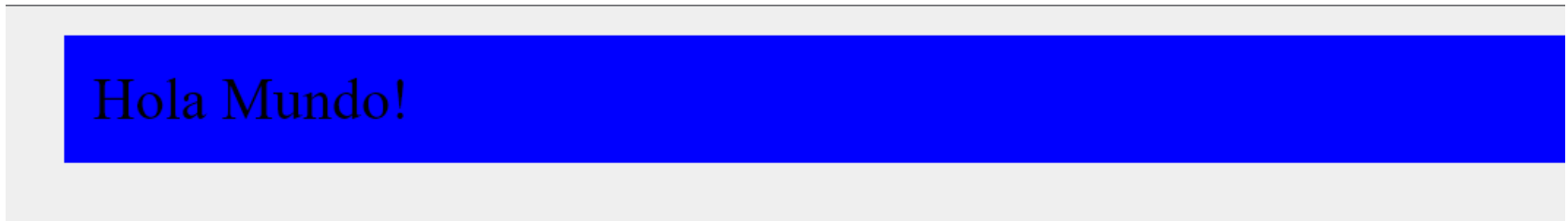
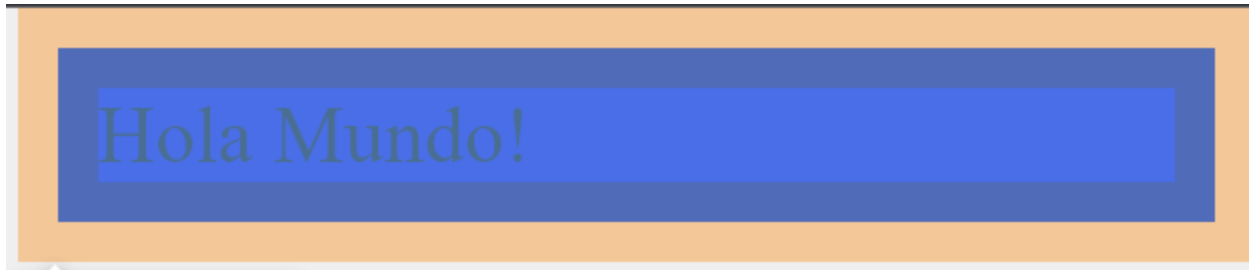


Hola Mundo!

Elementos

padding

```
.caja{  
  background-color: #00f;  
  margin: 8px;  
  padding: 8px;  
}
```



Elementos

Border(Entre margin y padding -- double,dashed,dotted,grooved,etc--)

```
.caja{  
  background-color: #00f;  
  margin: 8px;  
  padding: 8px;  
  border-style: solid;  
}
```



Hola Mundo!

Textos

Las propiedades de texto las podremos cambiar dependiendo si queremos placarlas a un párrafo, un link, una clase, etc.

De manera general tenemos las siguientes:

```
<html>
<head>
  <title>CSS</title>
  <link rel="stylesheet" href="main2.css" type="text/css">
</head>
<body>
  <p>Hola, soy un párrafo, y estoy siendo utilizado para practicar. Vamos a escribir un poco más de texto para ver como se comporta en nuestro sitio web</p>
  <a href="https://www.google.com">google</a>

</body>
</html>
```

Textos

```
body{
  background-color: #eee;
}

p {
  font-family:"Times New Roman";           /* Monaco, Times New Roman */
  font-style: italic; /* oblique */
  font-weight: 900; /* normal, bold unidades */
  font-size: 40px;
  color: #222fff;
  text-align:left; /* left ,right, justify */
  text-decoration: underline;
  text-indent: 40px; /*indentaciones */
  line-height: 1.7; /*espacios entre saltos de linea */
  word-spacing: 10px; /*separaciones entre palabras */
  text-shadow: 5px 2px 5px black; /*sombras a nuestro texto parametros: derecha , hacia abajo, que tan difuminado , color */
}

a {
  text-decoration: underline; /* decoracion del texto : none , overline,, line-through,underline */
  text-transform: uppercase; /* mayusculas o minusculas: uppercase , lowercase, capitalize */
  letter-spacing: 3px; /* espaciado entre letras */
}
```

Hola, soy un párrafo, y estoy siendo utilizado para practicar. Vamos a escribir un poco más de texto para ver como se comporta en nuestro sitio web

GOOGLE

Primeros pasos: Webscrapping

Introducción

Actualmente la web tiene más de 1.8 millones de paginas , lo cual implica aproximadamente 1.2 millones de terabytes de información, teniendo en cuenta lo anterior es evidente que uno de los principales esfuerzos en la actualidad es la recopilación de dicha información para fines muy específicos, pues podemos encontrar sitios con información desde los más simple como ventas de productos, hasta datos de simulaciones de partículas colisionando.

Con tal cantidad de información, una extracción manual es prácticamente imposible, motivo del cual nos centraremos en una técnica llamada webscraping.

Introducción

El webscraping no es la extracción de datos de la web, si no que implica la extracción de una manera automática mediante un software.
En sesiones posteriores haremos dicha automatización en nuestro lenguaje Python.

Extracciones web

Mecanismos formales

APIs
(Google,
twitter,facebook)

Mecanismos informales

Web Scraping
(basado en el html)

Desventajas del web Scraping

Se basa en el activo visual de la pagina o en su estructura html
Esto es una desventaja pues si cambia el contenido de dicha estructura, nuestro método de extracción también tendrá que cambiar para acoplarse a los cambios.

Algunas paginas web presentan candados de seguridad para la extracción de información, candados que podrían banear nuestra dirección IP por un intervalo de tiempo.

Es importante entender la estructura del cuerpo html , el cual se estructura de una forma jerárquica, o niveles, donde cada nivel tiene sus nodos, en este ejemplo los niveles son los tags , dentro de body, cada nivel se ira representando mediante una indexación e iniciara con su nuevo tag. Los nodos iniciales se denominarán nodos padres, y los subniveles que deriven de estos serán los nodos hijos. En el siguiente ejemplo el nodo raíz es body, y sus hijos serán p,a, div.

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS</title>
    <link rel="stylesheet" href="main2.css" type="text/css">
  </head>
  <body>
    <p>Hola, soy un párrafo, y estoy siendo utilizado para practicar</p>
    <a href="https://www.google.com">google</a>
    <div >
      <p>Soy otro parrafo</p>
    </div>

  </body>
</html>
```

A su vez, el tag div, que es hijo de body, tendrá dentro de el su hijo p.

XPath

Xpath es un lenguaje que permite construir expresiones para extraer información de un documento xml como son los dom o html de las paginas web.

Con este lenguaje se puede buscar y extraer solamente las partes del documento que nos interesan, por lo cual es muy útil para hacer el web Scraping

Para empezar a entender la sintaxis básica y hacer nuestras primeras búsquedas ingresaremos a la pagina :

<http://xpather.com/>



Las expresiones se escribirán en este apartado

```
.///*[self::abstract or self::subject or self::note][position() <= 2]
```

Aquí se resaltarán los resultados

```
//welcome-message
```

```
<app>  
  <welcome-message>Hi! This is xpather beta...  
  </welcome-message>
```

Las expresiones se escribirán en este apartado

```
.///*[self::abstract or self::subject or self::note][position() <= 2]
```

Aquí se resaltarán los resultados

```
//welcome-message
```

```
<app>  
  <welcome-message>Hi! This is xpather beta...  
  </welcome-message>
```

Pasos para búsquedas en XPath

Primeramente se tiene que definir el espectro de la búsqueda:

// nos realiza la búsqueda en cualquier parte del documento, en cualquier nivel

/ realizar búsqueda en la raíz del documento

//p

```
<body>
<p>Hola, soy un párrafo, y estoy siendo utilizado para practicar</p>
<a href="https://www.google.com">google</a>
<div >
  <p>Soy otro parrafo</p>
</div>
<div>
  <p>soy otro parrafo</p>
  <div>
    <a href="https://www.google.com">google</a>
  </div>
</div>
</body>
```

/body

```
<body>
  <p>Hola, soy un párrafo, y estoy siendo utilizado para practicar</p>
  <a href="https://www.google.com">google</a>
  <div >
    <p>Soy otro parrafo</p>
  </div>
  <div>
    <p>soy otro parrafo</p>
    <div>
      <a href="https://www.google.com">google</a>
    </div>
  </div>
</body>
```

Pasos para búsquedas en XPath

./ nos realiza la búsqueda en forma relativa o donde me encuentre en mi query, en el primer caso seria lo mismo que la raíz.

```
./body|
```

```
<body>
  <p>Hola, soy un párrafo, y estoy siendo utilizado para practicar</p>
  <a href="https://www.google.com">google</a>
  <div >
    <p>Soy otro parrafo</p>
  </div>
  <div>
    <p>soy otro parrafo</p>
  <div>
    <a href="https://www.google.com">google</a>
  </div>
</div>
</body>
```

Pasos para búsquedas en XPath

Después de definir el prefijo de búsqueda debemos especificar el nodo al que nos queremos dirigir, al cual accederemos mediante el tag.

```
// a
```

```
<body>
  <p>Hola, soy un párrafo, y estoy siendo utilizado para |
  <a href="https://www.google.com">google</a>
  <div >
    <p>Soy otro parrafo</p>
  </div>
  <div>
    <p>soy otro parrafo</p>
    <div>
      <a href="https://www.google.com">google</a>
    </div>
  </div>
</body>
```

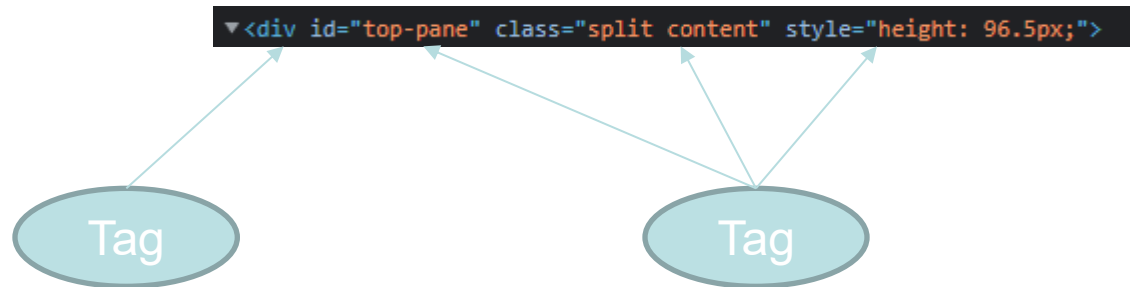
```
//p
```

```
<body>
  <p>Hola, soy un párrafo, y estoy siendo utilizado para practicar</p>
  <a href="https://www.google.com">google</a>
  <div >
    <p>Soy otro parrafo</p>
  </div>
  <div>
    <p>soy otro parrafo</p>
    <div>
      <a href="https://www.google.com">google</a>
    </div>
  </div>
</body>
```

Pasos para búsquedas en XPath

Seguido de nuestra selección de nodos, podemos focalizar mas la búsqueda mediante el uso de predicados los cuales definiré mediante corchetes después del nodo:

// tag [predicados]



LOS ATRIBUTOS DE NUESTRO TAG
ESTARAN DEFINIDOS POR ESPACIOS
DENTRO DE LA ETIQUETA Y UN
SIMBOLO DE IGUAL CON EL VALOR DE
CADA UNO

Predicados

```
▼<div id="top-pane" class="split content" style="height: 96.5px;">
```

Sintaxis igualdad

```
// tag [ @atributo="valor" ]
```

```
// div [ @id="top-pane" ]
```

Sintaxis no igualdad

```
// tag [ @atributo!="valor" ]
```

```
// div [ @class!="Split content" ]
```

Concatenar Predicados

```
▼<div id="top-pane" class="split content" style="height: 96.5px;">
```

Sintaxis

```
// tag [ @atributo="valor" and/or @atributo2 = "valor2" ]
```

Ejes anidados

Una vez ubicado nuestro elemento de búsqueda, podemos acceder a todos los niveles inferiores a este mediante la misma sintaxis, tomando en cuenta que // buscara todos los elementos inferiores a este nodo donde nos ubicamos y / únicamente a los hijos directos

```
// tag [ @atributo="valor" ] // tag2 [ @atributo2="valor2" ]
```

```
// tag [ @atributo="valor" ] / tag2 [ @atributo2="valor2" ]
```

```
// div [ @class="container" ] / li / span[ @atributo2="valor2" ]
```

Predicados por numeración

```
// div [ @class="container" ] / li[1] / span[ @atributo2="valor2" ]
```



[last()] Encontrar ultimo elemento

[posizio() = n] Encontrar por una posición

[contains(@atributo, "elemento")] Encontrar si un atributo coincide con cadena

[not(función)] Negar alguna condición lógica

[text()] Nos regresa el texto dentro de una etiqueta

Es importante saber que cada consulta nos regresara el nodo completo, por lo que una vez ubicado mi nodo, si quiero un elemento particular debo especificar el atributo.

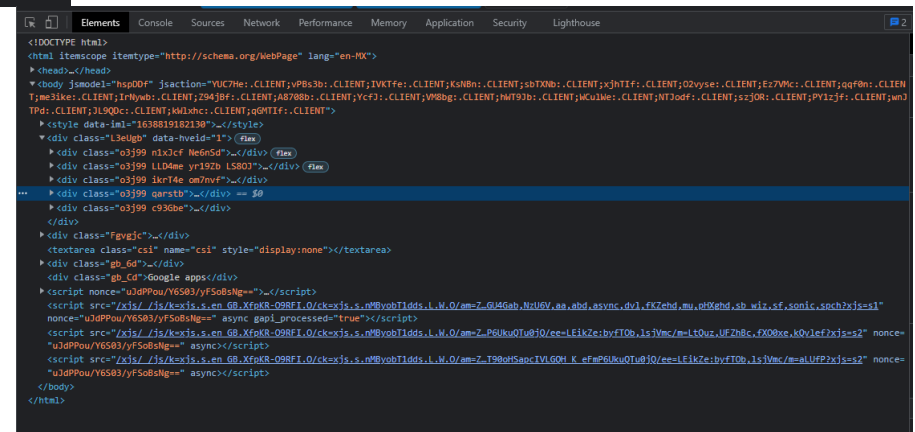
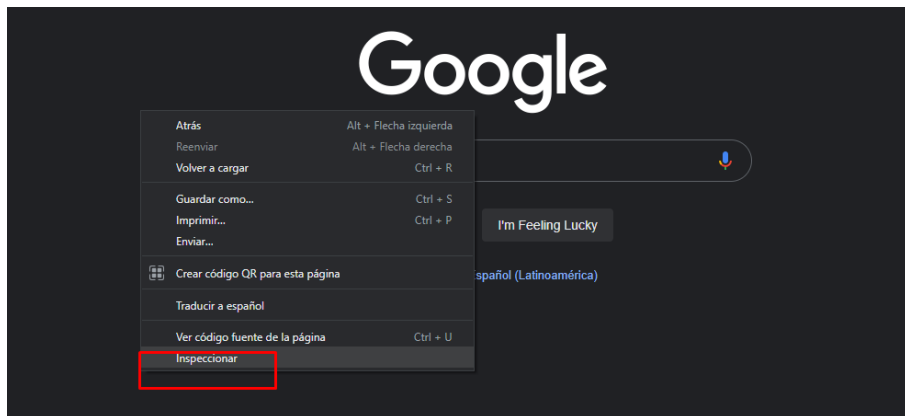
```
// h1[ contains( text() , "This is" ) ] / text()
```

```
// h1[ contains( text() , "This is" ) ] / @class
```

Documentación: <https://devhints.io/xpath>

Practicando en mi web

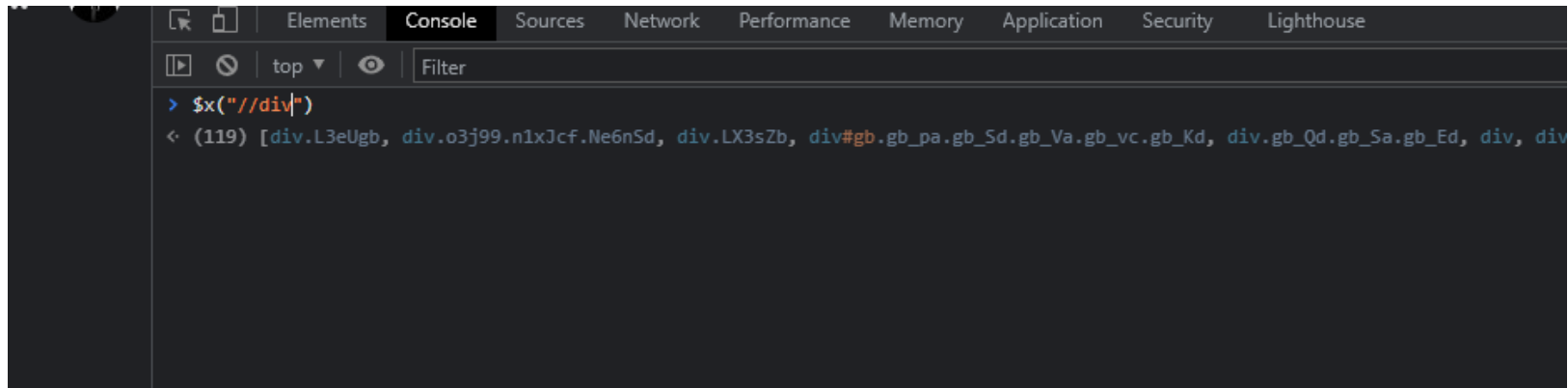
Para practicar nuestras expresiones xpath, podemos ingresar en cualquier navegador web de cualquier pagina, para ver el dom de las webs presionamos en cualquier lugar click derecho e inspeccionar, esto nos abrirá su dom.



Practicando en mi web

Nos vamos al apartado console, y en la consola colocamos la siguiente expresión:
`$x(" ")`

En ella, dentro de las comillas especificaremos nuestra búsqueda xpath, con lo que podemos ver en cualquier pagina web si una consulta esta correcta.



Búsqueda de todos los divs de la pestaña de Google.