

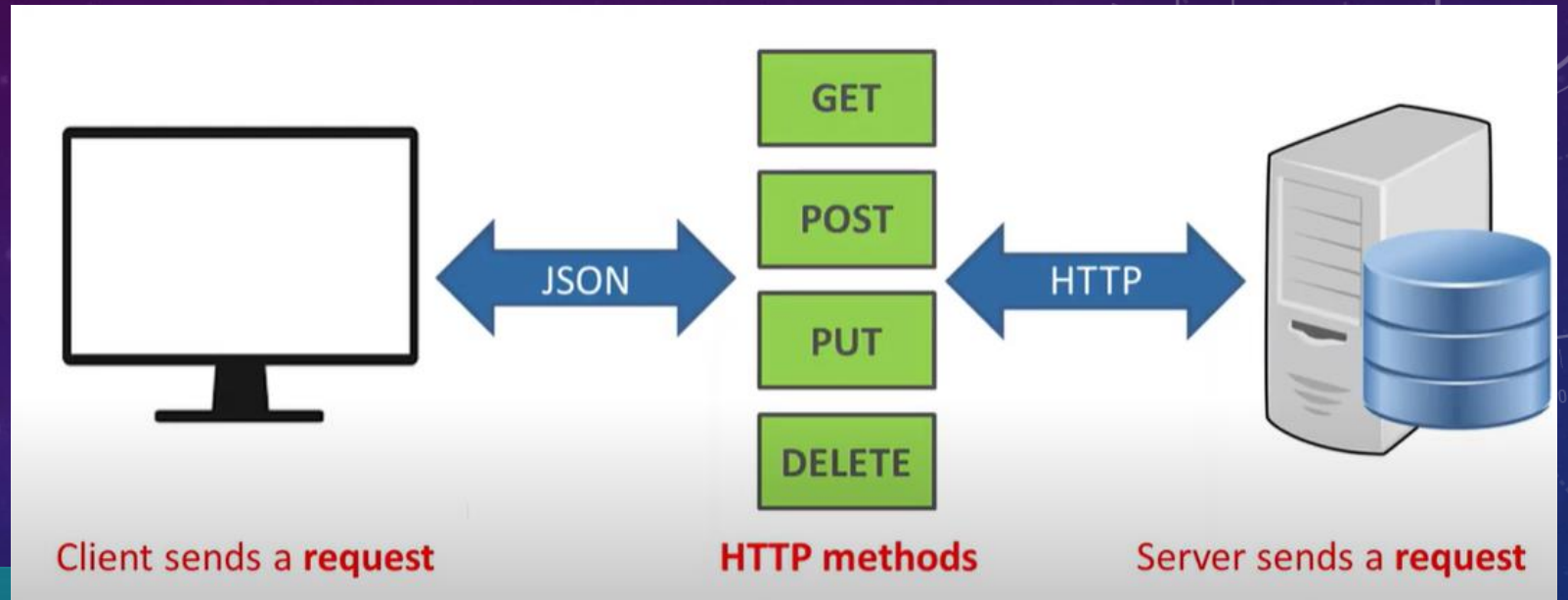
# LENGUAJES DE INTERCAMBIO DE INFORMACIÓN XML Y JSON

ESTUDIANTES:

ALCOCER VERGARA ALVARO

LOPEZ GARCIA JORGE ERNESTO

# INTRODUCCION



# XML (EXTENSIBLE MARKUP LANGUAGE):



```
ieza tipo="A">  
  <nombre>Tornillo</nombr  
  <descripcion>Cilindro m  
  </descripcion>  
  <caracateristica>  
    <tipo>metal</tipo>  
    <tamanyo>10</tamany  
  </caracateristica>  
  <vacio></vacio>  
ieza>
```

Hay muchos lenguajes basados en XML; Algunos ejemplos son XHTML, MathML, SVG, XUL, XBL, RSS, y RDF. También puedes crear uno propio.

## XML “CORRECTO”:

```
<message>
  <warning>
    Hola, mundo
  <!--missing </warning> -->
</message>
```




```
<message>
  <warning>
    Hola, mundo
  </warning>
</message>
```


## ENTIDADES:

Entidad	Caracter	Descripción
&lt;	<	Menor que
&gt;	>	Mayor que
&amp;	&	Ampersand
&quot;	"	Comilla doble
&apos;	'	Apóstrofe (o comilla sencilla)



# VENTAJAS Y DESVENTAJAS

- 
- Formato estructurado y mucho más fácil de comprender por la máquina.
  - Además, su uso te facilita la definición de estructuras complejas que pueden reutilizarse.

- 
- Se trata de un formato muy complicado de trabajar por las personas.
  - Es muy estricto, por lo que necesita un tiempo prolongado de procesamiento.
  - Tener un simple error con el namespace ya es suficiente para invalidar todo el documento.



# JSON (JAVASCRIPT OBJECT NOTATION):



JSON nació originalmente como una alternativa al uso de XML y poco a poco aumento su popularidad hasta llegar a ser uno de los estándares mas utilizados por los desarrolladores en todo el mundo. La idea principal era crear un lenguaje de estructuración que realizara tareas similares a las que realiza XML pero que lograra que el movimiento de información fuese mas ligero, específicamente para aquellos sistemas que manejaban enormes cantidades de información.

Este estándar esta mantenido por ECMAInternational y está basado en un subconjunto del lenguaje de programación JavaScript (ECMA-262). Como el anterior estándar es independiente del lenguaje de programación y del sistema operativo que lo ejecute.

JSON está construido por una colección de pares de nombre y valor o por una lista ordenada de valores. En muchos lenguajes de programación tenemos elementos parecidos como pueden ser los objetos, una estructura, un diccionario, un array....

cada elemento u objeto comienza por una llave y termina con otra. Cada nombre esta seguido de dos puntos y los elementos que están en la misma rama del árbol están separados por coma.

Este formato es el que se suele utilizar en los servicios web REST y en los últimos años le está comiendo el terreno que tenía ganado XML como fichero de intercambio, aunque muchos proveedores de APIs siguen encontrando valor a intercambiar datos en XML.

```
1 {  
2   "pieza": {  
3     "tipo": "A"  
4     "nombre": "Tornillo",  
5     "descripcion": "Cilindro mecánico con una cabeza utilizado en la fijación temporal de unas piezas con otras",  
6     "caracteristica": {  
7       "tipo": "metal"  
8       "tamanyo": 10  
9     },  
10    "vacio": ""  
11  }  
12 }
```

# VENTAJAS Y DESVENTAJAS

La estructura de JSON es más simple y ligera, sus archivos suelen pesar menos, la velocidad con la que procesa los datos es más rápida. funciona con todos los leguajes de programación, y como sus principales desventajas:

- Tiene una curva de aprendizaje mas amplia que XML
- La utilización de caracteres especiales como parte de su estructuración puede generar confusiones, especialmente si la personas a cargo de hacerlo no está familiarizada con su sintaxis.





# CONCLUSIÓN

- En definitiva, estos dos protocolos de intercambio de datos se usan habitualmente a la hora de establecer comunicación entre máquinas. Pese a ello, sus características propias hacen que tengan usos muy diferentes dependiendo de la información a transmitir y de la facilidad para el usuario de iniciar el proceso. Por lo tanto, es importante conocer bien las características de cada uno de ellos para poder decidir el más adecuado en cada proceso de datos.

