**FORMATO JSON**

**Definición**

**JSON** es un formato de texto ligero utilizado para el intercambio de datos. Basado en un subconjunto del lenguaje JavaScript, JSON permite representar información de manera estructurada mediante **pares clave-valor** y **listas ordenadas**. Aunque su sintaxis deriva de JavaScript, JSON es completamente independiente del lenguaje, lo que permite su uso en lenguajes como C, C++, Java, Python, PHP, entre otros. Gracias a estas propiedades, JSON se ha convertido en un estándar en aplicaciones web y desarrollo de software moderno.

**Características Principales**

* Es un formato **ligero y de fácil lectura** para los humanos.
* Permite que los datos sean fácilmente **interpretables por máquinas**.
* Utiliza estructuras universales: **objetos** y **arreglos**, lo que lo hace compatible con la mayoría de lenguajes de programación.
* Facilita el **intercambio de información entre clientes y servidores**.
* Su sintaxis es simple, reduciendo la complejidad de los datos comparado con otros formatos como XML.

**Estructura**

**Objetos**

Los **objetos** en JSON son colecciones de pares clave-valor delimitados por llaves { }. Cada clave debe ser una cadena de texto y los valores pueden ser de distintos tipos: números, cadenas de caracteres, booleanos, objetos, arreglos o valores nulos.  
**Ejemplo:**

{

"nombre": "Isabela",

"edad": 20,

"estudiante": true

}

**Arreglos**

Los **arreglos** son listas ordenadas de valores, delimitadas por corchetes [ ]. Pueden contener valores de cualquier tipo, incluidos objetos o incluso otros arreglos.  
**Ejemplo:**

["C++", "Java", "Python"]

**Tipos de Datos en JSON**

1. **String**: Cadenas de texto entre comillas dobles.
2. **Number**: Números enteros o decimales.
3. **Boolean**: Valores lógicos true o false.
4. **Object**: Colección de pares clave-valor.
5. **Array**: Lista ordenada de valores.
6. **Null**: Representa la ausencia de valor.

**Aplicaciones de JSON**

JSON se utiliza principalmente para el **intercambio de información entre aplicaciones**, especialmente en entornos web. Algunas de sus aplicaciones son:

* Comunicación entre **cliente y servidor** en aplicaciones web modernas.
* Almacenamiento de configuraciones en **archivos .json**.
* Serialización y deserialización de objetos en diferentes lenguajes de programación.
* Integración en **APIs y microservicios** para compartir datos de manera eficiente.
* Uso combinado con **AJAX** para actualizar contenido de páginas web sin recargar toda la página.

**Ejemplo de Datos JSON**

Un ejemplo de un objeto JSON que almacena información de un estudiante sería:

{

"nombre": "Isabela",

"edad": 20,

"universidad": "ESPE",

"materias": ["Programación", "Matemáticas", "Bases de Datos"],

"graduada": false,

"direccion": {

"ciudad": "Quito",

"pais": "Ecuador"

}

}

**Ventajas de JSON**

* **Ligero y rápido**: ocupa menos espacio que XML.
* **Fácil de usar y entender**: legible para humanos.
* **Compatible con casi todos los lenguajes de programación**.
* **Flexible**: permite estructuras jerárquicas complejas mediante objetos y arreglos.
* **Ideal para servicios web y APIs** modernas.

**Conclusión**

JSON se ha consolidado como un estándar para el intercambio de información debido a su simplicidad, compatibilidad y eficiencia. Su diseño ligero y su fácil comprensión permiten que los desarrolladores puedan integrar aplicaciones, servicios web y APIs de manera rápida y confiable. Gracias a su independencia de lenguaje y a su capacidad para representar datos de manera estructurada, JSON se mantiene como una herramienta esencial en el desarrollo de software moderno, especialmente en el ámbito web y móvil.