



**PONTIFICIA UNIVERSIDAD
CATOLICA DELECUADOR-
SEDE ESMERALDAS**

ASIGNATURA:

PROGRAMACION ESTRUCTURADA

DOCENTE:

KLEBER POSLIGUA

AUTOR:

RUBEN BONE

TEMA:

PARADIGMA FUNCIONAL

NIVEL:

SEGUNDO SEMESTRE

Deestructuración de objetos y arreglos

La deestructuración de objetos y arreglos es una característica en varios lenguajes de programación, como JavaScript, que permite extraer valores de manera concisa y eficiente. Esta técnica proporciona una forma conveniente de descomponer estructuras complejas, como objetos y arreglos, en variables individuales.

En el caso de la deestructuración de arreglos, puedes extraer valores directamente asignándolos a variables mediante la sintaxis de corchetes [].

La deestructuración simplifica la asignación de variables y mejora la legibilidad del código al evitar repeticiones innecesarias. Esta técnica es especialmente útil al trabajar con estructuras de datos complejas, permitiendo acceder fácilmente a los valores que necesitas.

Operador de propagación

El operador de propagación, también conocido como spread operator, es una característica presente en algunos lenguajes de programación, como JavaScript. Su función principal es descomponer estructuras de datos, como arreglos o objetos, para luego incorporar sus elementos de manera individual en otro contexto. Este operador se representa mediante tres puntos consecutivos (...).

¿Qué es JSON y su utilidad?

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero y fácilmente legible por humanos. Se utiliza para representar información estructurada mediante pares clave-valor, siendo versátil y compatible con múltiples lenguajes de programación. Su principal utilidad radica en la transferencia de datos entre sistemas, siendo ampliamente empleado en aplicaciones web y como estándar para el intercambio de información entre plataformas.

Filter - map - every - reduce - some – sort

Las funciones ``filter``, ``map``, ``every``, ``reduce``, ``some`` y ``sort`` en JavaScript son métodos de manipulación de arreglos.

- `filter`: Crea un nuevo arreglo con elementos que cumplen una condición.
- `map`: Transforma cada elemento del arreglo mediante una función.
- `every`: Verifica si todos los elementos cumplen una condición.
- `reduce`: Combina elementos para obtener un único valor acumulado.
- `some`: Comprueba si al menos un elemento cumple una condición.
- `sort`: Ordena los elementos del arreglo.

Estas funciones simplifican operaciones comunes en la manipulación de datos en arreglos, mejorando la claridad y eficiencia del código.