**INFORME**

**Aprendiz: Enyeer Manuel Granados Mardinis**

**Ficha: 2558104**

**Servicion Nacional de Aprendizaje – SENA.**

**Centro de Electricidad, Electrónica y Telecomunicaciones – CEET.**

**Analisis y Desarrollo de Software.**

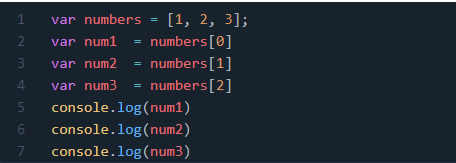
**Instructora: Isaura Maria Suarez Novoa.**

1. **¿Qué es una matriz?**

Una matriz en JavaScript es un tipo de objeto global que se utiliza para almacenar datos. Las matrices consisten en una colección o lista ordenada que contiene cero o más tipos de datos, y utilizan índices numerados que comienzan en 0 para acceder a elementos específicos.

Las matrices son muy útiles ya que almacenan valores múltiples en una sola variable, lo que puede condensar y organizar nuestro código, haciéndolo más legible y mantenible. Las matrices pueden contener cualquier tipo de datos, incluidos números, cadenas y objetos.

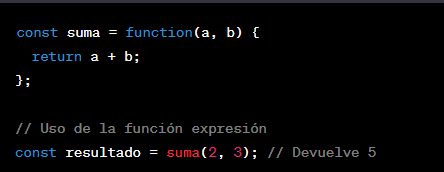
1. **Ejemplo de Matriz**



1. **¿Qué es una función expresión?**

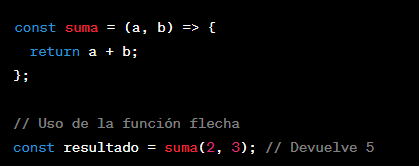
En JavaScript, una función expresión es una forma de definir una función de manera anónima y asignarla a una variable o utilizarla directamente en el contexto donde se necesita. A diferencia de una función con nombre, que se declara con la palabra clave “function” seguida de un identificador, una función expresión no tiene un nombre explícito y se define directamente como una expresión en una línea de código.

1. **Ejemplo de función expresión**

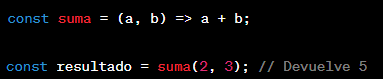


1. **¿Qué es una función flecha?**

En JavaScript, una función flecha (arrow function en inglés) es una sintaxis más corta y concisa para definir funciones. Se introdujo en ECMAScript 6 (ES6) para proporcionar una forma más elegante de crear funciones en comparación con las funciones tradicionales.



Si la función flecha tiene solo una expresión en su cuerpo, se puede simplificar aún más eliminando las llaves y la palabra `return`, ya que automáticamente se devolverá el resultado de la expresión.



Las funciones flecha son especialmente útiles cuando se utilizan en funciones de orden superior y en contextos donde no se necesita tener su propio valor de `this`. Sin embargo, debes tener en cuenta que no son adecuadas para todos los casos, ya que su comportamiento con respecto a `this` puede ser diferente al de las funciones tradicionales.

1. **¿Qué es una función recursiva?**

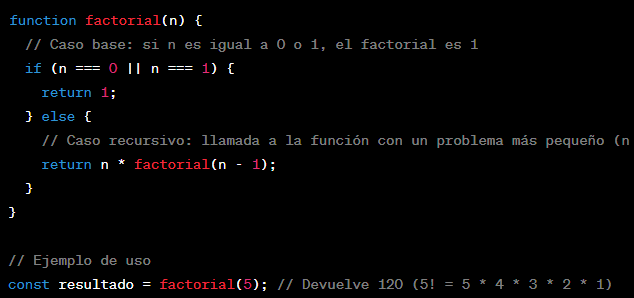
Una función recursiva en JavaScript es una función que se llama a sí misma durante su ejecución. Es una técnica de programación en la que una función se descompone en problemas más pequeños y se resuelve utilizando la misma función de manera repetida hasta que se alcanza un caso base que evita una llamada recursiva adicional.

Para que una función recursiva sea efectiva, debe cumplir dos condiciones principales:

**Caso base:** Es la condición que determina cuándo la función recursiva debe dejar de llamarse a sí misma y, en cambio, devolver un resultado directamente. Sin un caso base, la función podría entrar en un bucle infinito y causar un desbordamiento de pila.

**Caso recursivo:** Es la parte de la función donde se realiza la llamada recursiva a sí misma con un problema más pequeño. Este paso es esencial para que la función avance hacia el caso base.

1. **Ejemplo de función recursiva**

****