



Docs_

Código Junior vs Senior



Vamos a ver cinco escenarios comunes y cómo estos podrían ser implementados por ambos. Tenga en cuenta que estas son generalizaciones y que cada desarrollador tiene su propio estilo y nivel de experiencia.

1. Sumar dos números
2. Filtrar un array de números para obtener solo los números pares
3. Convertir una cadena de texto en mayúsculas y contar sus caracteres
4. Comprobar si un número es primo
5. Calcular el factorial de un número

Docs_ Generalidades Curiosidades

1. Sumar dos números

Junior:

```
function sumar(a, b) {  
  let suma = a + b;  
  return suma;  
}  
console.log(sumar(5, 3));
```

Senior (más conciso y directo):

```
const sumar = (a, b) => a + b;  
console.log(sumar(5, 3));
```

2. Filtrar un array de números para obtener solo los números pares

Junior:

```
function filtrarPares(arr) {  
  let pares = [];  
  for (let i = 0; i < arr.length; i++) {  
    if (arr[i] % 2 === 0) {  
      pares.push(arr[i]);  
    }  
  }  
  return pares;  
}  
console.log(filtrarPares([1, 2, 3, 4, 5, 6]));
```

Senior (usando métodos de array):

```
const filtrarPares = arr => arr.filter(num => num % 2 === 0);  
console.log(filtrarPares([1, 2, 3, 4, 5, 6]));
```

Docs_ Generalidades Curiosidades

3. Convertir una cadena de texto en mayúsculas y contar sus caracteres

Junior:

```
function mayusculasYContar(texto) {  
  let mayusculas = texto.toUpperCase();  
  let longitud = mayusculas.length;  
  return `Texto: ${mayusculas}, Longitud: ${longitud}`;  
}  
console.log(mayusculasYContar("Hola Mundo"));
```

Senior (más limpio y en una línea):

```
const mayusculasYContar = texto => `Texto: ${texto.toUpperCase()}, Longitud: ${texto.length}`;  
console.log(mayusculasYContar("Hola Mundo"));
```

4. Comprobar si un número es primo

Junior:

```
function esPrimo(num) {  
  for (let i = 2; i < num; i++) {  
    if (num % i === 0) {  
      return false;  
    }  
  }  
  return num > 1;  
}  
console.log(esPrimo(7));
```

Senior (optimizando la lógica y utilizando raíz cuadrada):

```
const esPrimo = num => {  
  for (let i = 2, s = Math.sqrt(num); i <= s; i++)  
    if (num % i === 0) return false;  
  return num > 1;  
};  
console.log(esPrimo(7));
```

Docs_ Generalidades Curiosidades

5. Calcular el factorial de un número

Junior:

```
function factorial(num) {  
  let resultado = 1;  
  for (let i = 2; i <= num; i++) {  
    resultado *= i;  
  }  
  return resultado;  
}  
console.log(factorial(5));
```

Senior (usando recursividad):

```
const factorial = num => num < 2 ? 1 : num * factorial(num - 1);  
console.log(factorial(5));
```

Estos ejemplos ilustran cómo un desarrollador senior podría escribir código más conciso y eficiente, a menudo utilizando características avanzadas del lenguaje y técnicas de programación más sofisticadas. Sin embargo, es importante destacar que la claridad y la mantenibilidad del código también son muy importantes, especialmente en equipos donde los desarrolladores con diferentes niveles de experiencia trabajan juntos.