

## 01. LEGIBILIDAD

Las corutinas permiten escribir código asíncrono de manera secuencial, lo que mejora la legibilidad y mantenibilidad del código

## 02. MANEJO DEL FLUJO

Utilizan estructuras de control que simplifican el manejo del flujo en comparación con los callbacks anidados o en pirámide.

## 03. MANEJO DE ERRORES

Permiten utilizar estructuras de manejo de errores como try-catch, lo que simplifica el manejo de errores en comparación con los callbacks, donde a menudo se deben manejar errores a través de interfaces y métodos.

## 04. EFICIENCIA

Están diseñadas para ser altamente eficientes en términos de uso de recursos, lo que puede llevar a una mejor utilización de CPU y memoria.

## 01. COMPATIBILIDAD CON PLATAFORMAS ANTIGUAS

fueron útiles antes de la creación de corutinas.

## 02. SIMPLICIDAD

Para operaciones asíncronas simples o casos de uso donde no hay anidamiento excesivo de callbacks, los callbacks pueden ser sencillos de implementar

## 03. COMPATIBILIDAD CON LIBRERÍAS Y APIS EXISTENTES

Muchas librerías y APIs de terceros utilizan callbacks para manejar respuestas asíncronas.

## 04. RETROALIMENTACIÓN

Pueden ser útiles cuando se necesita proporcionar retroalimentación o realizar acciones específicas en diferentes puntos del ciclo de vida de la operación asíncrona.

Corutinas

VS

Callbacks