Edwin Andrés Ortega Kou – 22305

Laboratorio 8 – Codelab

Corrutinas vs Callbacks

1. Código más legible:
   Las corrutinas permiten escribir código asíncrono que se parece más al código síncrono tradicional. Esto puede hacer que el código sea más fácil de leer y mantener.

2. Manejo de errores más sencillo:
   Con los callbacks, el manejo de errores puede volverse complicado, ya que es necesario pasar errores a través de múltiples niveles de callbacks. Las corrutinas, en cambio, permiten usar estructuras de manejo de errores más convencionales, como try/except.

3. Evitan el "Callback Hell":
   Una desventaja de los callbacks es que pueden llevar a un fenómeno conocido como "callback hell" o "pyramid of doom", donde tienes múltiples niveles de callbacks anidados, lo que complica la legibilidad y mantenimiento del código. Las corrutinas ayudan a aplanar esta estructura.

4. Composición más sencilla:
   Las corrutinas pueden ser más fáciles de componer y reutilizar que los callbacks. Puedes definir múltiples operaciones asíncronas y combinarlas de manera más intuitiva.

5. Control de flujo más flexible:
   Las corrutinas proporcionan un mayor control sobre el flujo de ejecución. Por ejemplo, puedes pausar y reanudar una corrutina en diferentes puntos, lo que es más complicado de lograr con callbacks.

6. Integración con sintaxis moderna:
   Lenguajes de programación modernos (como Python con async/await) han introducido sintaxis especializada para trabajar con corrutinas, lo que facilita su uso y ofrece ventajas de rendimiento y optimización.