**📌 Resumen – Sección 6: Programación asíncrona (C#)**

**1. Qué es programación asíncrona**

* **Síncrono**: Las tareas se ejecutan **una tras otra**. No puedes iniciar la siguiente hasta que termine la anterior. Ej: cocinar espagueti → luego pavo → luego postre (tiempo total = suma de todos los tiempos).
* **Asíncrono**: Permite **ejecutar tareas en paralelo**. Mientras una tarea espera (p.ej., conexión a BD, envío de correo), puedes iniciar otra y al final esperar que todas terminen.
* **Ventaja**: Aprovecha tiempos muertos, mejora el rendimiento y la respuesta del sistema.

**2. Método síncrono en C#**

* Ejemplo:
  + Conexión a BD (1 seg)
  + Envío de correo (1 seg)
  + Tiempo total ≈ 2 seg.
* Se usa Thread.Sleep() para simular el bloqueo y Stopwatch para medir el tiempo.
* **Desventaja**: No se avanza hasta que termine cada paso.

**3. Task**

* **Task** = Representa una operación asíncrona.
* **Sintaxis básica**:

public async Task<IActionResult> MetodoAsync() {

var tarea = new Task(() => {

// Código de la tarea

});

tarea.Start();

await tarea; // Esperar que termine

return Ok();

}

* **Puntos clave**:
  + async en la firma del método → habilita await.
  + await detiene el avance **solo** hasta que termine la tarea, sin bloquear el hilo principal.
  + Si no usas await, el código sigue y la tarea puede seguir ejecutándose en segundo plano.
  + Task<T> permite devolver resultados.

**4. Síncrono vs Asíncrono**

* **Síncrono**:
  + Tarea 1 (1 seg) → Tarea 2 (1 seg) → Total ≈ 2 seg.
* **Asíncrono** (ejecutar en paralelo con Task):
  + Tarea 1 (1 seg) + Tarea 2 (1 seg) → Total ≈ 1 seg.
* **C#** ya ofrece muchos métodos asíncronos listos (acceso a BD, lectura/escritura de archivos, llamadas HTTP).

**5. Conceptos clave a recordar**

* **async**: Marca el método como asíncrono.
* **await**: Espera el resultado de una tarea antes de continuar.
* **Task / Task<T>**: Representa operaciones asíncronas (con o sin retorno).
* **Ventaja**: Mejor rendimiento y aprovechamiento del tiempo de ejecución.