

Taller Grupal – Manejo de Tasks, Async/Await y Consumo de APIs en MVC

Instrucciones Generales

Cada célula (equipo) tendrá asignado un taller que representa un **caso real de negocio**.

El objetivo es:

1. **Entender el problema y proponer la solución** (con casos de uso y diagrama de clases).
2. **Implementar la solución en MVC con EF Core**.
3. **Aplicar buenas prácticas**: manejo de excepciones con try/catch, uso de async/await, Tasks simultáneos, consumo de API externa y deserialización de datos.
4. Preparar una **exposición de 10-15 minutos** donde:
 - Presenten el **caso de uso y el diagrama de clases**.
 - Expliquen la **arquitectura** del proyecto.
 - Muestren la **aplicación funcionando**.
 - “Vendan” la solución como si fuera un producto para el cliente (suplente TL).

Tiempo estimado: **4 horas en total** (3 horas desarrollo + 1 hora exposiciones).

♦ Talleres por Célula

Célula 1 – Gestión de Películas

Problema:

Una productora quiere un sistema donde los usuarios consulten películas por género, obtengan detalles y calificación desde una API externa de películas (por ejemplo OMDb API).

Requisitos:

- CRUD de películas locales con EF Core.
 - Llamar a la API externa para traer información de una película (sinopsis, actores, año).
 - Mostrar en la vista información combinada (local + externa).
 - Ejemplo de Task en paralelo: consultar 3 películas al mismo tiempo y mostrar resultados.
-

Célula 2 – Catálogo de Videojuegos

Problema:

Una tienda gamer quiere mostrar un catálogo con detalles de videojuegos consumiendo la API de

RAWG.

Requisitos:

- CRUD de videojuegos (nombre, género, precio).
 - API externa para traer trailers, imágenes o reseñas.
 - Tareas en simultáneo: buscar varios juegos por ID en paralelo.
 - Manejar errores si la API no responde con try/catch.
-

Célula 3 – Seguimiento de Criptomonedas

Problema:

Un usuario quiere consultar precios actualizados de criptomonedas. Usar CoinGecko API.

Requisitos:

- CRUD de criptomonedas favoritas en base local.
 - Llamado a la API externa para mostrar precio actual, variación en 24h y ranking.
 - Tarea paralela: obtener el precio de 5 monedas favoritas al mismo tiempo.
 - Si la API no responde, mostrar último valor guardado en BD.
-

Célula 4 – Noticias en Tiempo Real

Problema:

Un medio digital necesita mostrar titulares recientes consumiendo NewsAPI.

Requisitos:

- CRUD de categorías favoritas (ej. tecnología, deportes).
 - API externa para traer titulares más recientes.
 - Tarea paralela: consultar noticias de varias categorías en simultáneo.
 - Manejo de errores si la API devuelve error 429 (too many requests).
-

Célula 5 – Sistema de Recetas de Cocina

Problema:

Una app de cocina quiere recomendar recetas. Usar TheMealDB API.

Requisitos:

- CRUD de recetas favoritas en BD.
 - API externa para buscar recetas por ingrediente o nombre.
 - Tarea paralela: buscar varias recetas a la vez.
 - Mostrar receta con ingredientes, pasos e imagen.
-

Célula 6 – Catálogo de Libros

Problema:

Una librería online quiere mostrar su catálogo y conectarse con la API de OpenLibrary.

Requisitos:

- CRUD de libros locales.
 - API externa para traer portada, autor, descripción.
 - Tarea paralela: buscar varios ISBN a la vez.
 - Try/catch para manejar cuando el ISBN no existe.
-

Célula 7 – Seguimiento de Universidades

Problema:

Un ministerio de educación quiere mostrar universidades disponibles usando la API pública Universities API.

Requisitos:

- CRUD de universidades locales (nombre, país).
 - API externa para traer universidades por país.
 - Tarea paralela: consultar universidades de varios países en simultáneo.
 - Deserializar y mostrar información en tabla ordenada.
-

Célula 8 – Seguimiento de Países

Problema:

Una ONG necesita información sobre países usando REST Countries API.

Requisitos:

- CRUD de países de interés.
- API externa para traer población, idioma, capital, bandera.
- Tarea paralela: consultar varios países en simultáneo.
- Manejo de errores en caso de país inexistente.