**📌 Resumen – Flujos y Configuraciones en .NET**

**1. Modelos y DTOs**

* **Modelo**: Representa una tabla en la base de datos (ejemplo: Usuario con ID, correo, password, fecha de creación).
* **Problema**: No siempre queremos enviar todos los datos (ej. no mostrar password o fecha de creación en todas las pantallas).
* **DTO (Data Transfer Object)**: Clase ligera para transferir solo la información necesaria entre capas (ej. UserDto con solo ID y correo).
* **Ventaja**: Reduce transferencia de datos y optimiza la aplicación.

**2. Creación de un DTO**

* Se utiliza un **servicio externo de pruebas**: [JsonPlaceholder](https://jsonplaceholder.typicode.com/).
* Ejemplo de entidad: Post con userId, id, title, body.
* Se crea un PostDto que refleja esa estructura.
* Si una propiedad puede ser null, se agrega ? en su tipo (string?).

**3. HttpClient para solicitudes externas**

* **HttpClient**: Clase de .NET que permite consumir servicios REST (GET, POST, PUT, DELETE).
* Se implementa en un **servicio** (PostService) que devuelve un IEnumerable<PostDto>.
* Uso de **async/await** para llamadas asíncronas (esperar respuesta del servicio externo).
* **Deserialización**: Se convierte el JSON recibido en una lista de objetos (JsonSerializer.Deserialize).
* Problema común: Diferencia entre mayúsculas/minúsculas en nombres de propiedades.
  + Solución: Configurar JsonSerializerOptions { PropertyNameCaseInsensitive = true }.

**4. IHttpClientFactory**

* Evita **crear instancias manuales** de HttpClient en cada servicio.
* Se aplica el **patrón Factory**: una fábrica central crea y configura clientes HTTP.
* Configuración en Program.cs con:

builder.Services.AddHttpClient<IPostService, PostService>(c =>

{

c.BaseAddress = new Uri("https://jsonplaceholder.typicode.com/posts");

});

* Beneficios:
  + Reutilización del cliente.
  + Configuración centralizada de la URL.
  + Facilidad de mantenimiento (cambiar una sola vez la URL).

**5. Uso de appsettings.json**

* Archivo de configuración que permite **separar valores del código**.
* Ejemplo: Guardar la URL externa en appsettings.json:

"UrlPost": "https://jsonplaceholder.typicode.com/posts"

* Lectura en Program.cs:

builder.Configuration["UrlPost"]

* Ventaja: Diferentes configuraciones por entorno (Desarrollo, Producción, Pruebas).

✅ **Objetivo final de esta sección**:  
Organizar el código, optimizar llamadas a servicios externos y centralizar configuraciones, usando DTOs, HttpClientFactory y appsettings.json.