

Resumen de Conceptos Clave: Desarrollo de ABM y Filtros en C# / SQL Server

1. Baja Lógica vs. Baja Física

- **Baja Física (DELETE):** Borra permanentemente el registro de la base de datos. Es peligroso en entornos de producción porque destruye el historial.
- **Baja Lógica (UPDATE):** El registro se mantiene en la base de datos, pero se modifica una bandera o estado (ej. Activo = 0). Para el usuario desaparece porque modificamos el SELECT principal para que solo traiga los registros con Activo = 1. Es el estándar de la industria.

2. Filtro Rápido (En Memoria) vs. Filtro Avanzado (En Base de Datos)

- **En Memoria (Client-Side):** Traemos todos los datos a una lista (List<Pokemon>) y filtramos usando expresiones Lambda de C# (FindAll, Contains, ToLower). Es ideal para búsquedas en tiempo real (eventoTextChanged) cuando tenemos pocos datos, ya que no consume recursos de red, pero gasta memoria RAM local.
- **En Base de Datos (Server-Side):** Construimos una consulta SQL dinámica (concatenando variables en un string con sentencias WHERE y AND) y delegamos el trabajo pesado al motor de SQL Server. Solo viajan por la red los registros que coinciden. Es la única forma eficiente de manejar tablas con miles o millones de registros.

3. Programación Defensiva, Encapsulamiento y UX

- **Seguridad y Conexiones:** Siempre debemos asegurar el cierre de las conexiones a la base de datos utilizando el bloque finally, garantizando que la memoria se libere incluso si ocurre una excepción (CerrarConexion()).
- **Encapsulamiento:** Al usar clases de acceso a datos (Acceso_a_datos), las herramientas internas están protegidas. Para leer datos, debemos acceder a través de propiedades públicas (datos.Lector.Read()).
- **Experiencia de Usuario (UX) y Prevención de Nulos:** La interfaz gráfica no debe confiar en el usuario. Debemos implementar "Combos en Cascada" para guiar la búsqueda, validar que las cajas de texto no estén vacías antes de consultar, y utilizar casteos seguros (as string + IsNullOrEmpty) para prevenir excepciones catastróficas como NullReferenceException cuando se resetean los controles.