**06ArielDagner 🡪 8**

Capa Presentación

Alumnos de Ciclo (aprobados🡪nombre)

* Bien

Empresas para el ciclo (ordenadas alfabéticamente) con Solicitudes y Asignadas

* Bien

Alumnado con empresa asignada

* Muestra datos no pedidos

Asignar empresa

Retirar alumnado de empresa

Capa datos

* Búsquedas por clave siempre, que mandan a buscar entre todos sin usar las ventajas del modelo. Ejemplo: en AsignarEmpresa si tenemos el ciclo, solo debemos buscar entre **sus alumnos** no al alumno entre todos

AsignarEmpresa

* Ver este caso,

Alumnos alumno = bdFCTsEntities.Alumnos.Find(alum.NMatricula);

if (alumno == null) return $"El alumno/a {alum.Nombre} no existe o no es del ciclo {cicloBuscar.Nombre}"; // Si no existe es verdad, pero si no es el ciclo no sería esta la consulta

RetirarEmpresa y EliminarEmpresa

* Así no se sabría cual usar, lo lógico es que sea una sola
* Este mensaje no es lógico en la capa de datos: return $"El alumno/a {alumno.Nombre} tiene asignada la empresa {alumno.FCTs.Empresas.Nombre}, ¿Deseas eliminar la asignación?";

Comentarios generales

* Nombres de formularios sin seguir convención
* Nombres de métodos tampoco: retirarEmpresa debe ser RetirarEmpresa
* Mezcla de instrucciones en capas que no le corresponden
* Y seguís definiendo listas en Gestion que no tiene sentido (y si lo fueran debían ser propiedades). ¿qué ocurre si el usuario de la clase intenta acceder a Ciclos, sin llamar antes a GetCiclos? Lo lógico es que GetCiclos devuelva la lista directamente