03AritzXabier 🡪 7

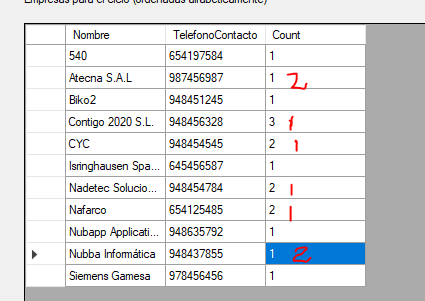
Capa presentación

Alumnos de Ciclo (aprobados🡪nombre)

* Ordenados por aprobado, pero no por nombre

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

* Ordenadas bien
* En lugar de Solicitudes y Asignadas 🡪 Count (que cuenta la suma de solicitudes para todos los ciclos)



Alumnado con empresa asignada

* Bien

Asignar empresa

* Muestra todo el alumnado del ciclo

Retirar empresa

* Los controles los hace en el formulario

Capa datos

Retirar alumnado de empresa

* No controla posibles errores public Empresa DevolverEmpresa(int nMatricula) // todo Debía hacer aquí los controles
* if (anadirEmpresa.FCTs.Count == Ofertafct.Cantidad) // No controla que las FCTs solicitadas sean justo de ese ciclo

Capa datos

* Si realizamos una consulta del tipo where **nunca** devolverá null, sino una lista vacía

List<OfertasFCT> ofertasCiclo = ciclo.OfertasFCTs.ToList();

List<Empresa> empresaPorciclo = (from emp in ofertasCiclo

where emp.IdCiclo == ciclo.Id // ¿Para qué preguntar si coinciden con el ciclo si eran las del ciclo?

select emp.Empresa).OrderBy(o => o.Nombre).ToList();

if (empresaPorciclo == null) // todo Nunca será null

Métodos como AñadirAlumnoEmpresa

public string AñadirAlumnoEmpresa(Ciclo anadirCiclo, Empresa anadirEmpresa, Alumno anadirAlumno, Profe anadirProfe, string tutorEmpresa) // No tiene sentido que traiga los objetos, solo los identificadores, ya que es en la colecciones existentes donde vamos a hacer las modificaciones

/ Con los id debía buscar los correspondientes, el ciclo, ver si tenía esa empresa, si tenía ese alumno o alumna, si el alumno estaba aprobado, si ....