

# 1.- Construir la estructura del proyecto

## Capa Interfaz de usuario → (Proyecto) WebAlumnosFP

- Añadamos el GridView con el formato que queramos y lo llamamos **dgv**
  - En editar columnas agregamos un CommandField con las siguiente configuración:
    - **ButtonType="Button"**
    - **DeleteText="Del"**
    - **EditText="Edit"**
    - **ShowCancelButton="False"**
    - **ShowDeleteButton="True"**
    - **ShowEditButton="True"**
  - En cuanto a sus eventos controlamos los dos que vamos a utilizar: **RowDeleting** y **RowEditing**.
- Incluirá la Carpeta de Datos: La base de datos del proyecto web debe estar en la carpeta App\_Data

## Capa (Proyecto) LogicaNegocioyADatos

- Añadimos el proyecto de Librería de Clases **LogicaNegocioyADatos**.
- A la interfaz de usuario añadimos referencia antes de que se nos olvide
- A la clase de este proyecto le llamamos **LNyAD** se podría hacer también una clase por cada entidad, ejemplo: LN\_Alumno; LN\_Grupo
- Añadimos una carpeta con las entidades Alumno y Grupo. Fijate que su espacio de nombres se llamará **LogicaNegocioyADatos.Entidades**. Esto habrá que tenerlo en cuenta al referirnos a ellas. Si queremos, le podemos quitar la referencia a la carpeta
- **Aquí instalamos el DataSet**
  - Cuando incorporamos la BD al proyecto, copiar la cadena de conexión de app.config a la Web.config
  - Pensar las consultas que vamos a necesitar:
    - **GetAlumnosConAliasGrupo** para mostrar todos con sus respectivos alias de grupo (para usarla hay que insertar la columna Grupo):  
SELECT idAlumno, Alumnos.idGrupo, Grupos.alias as Grupo,  
apellidosNombre, telefono, movil, dni, email  
FROM Alumnos, Grupos  
WHERE Alumnos.idGrupo=Grupos.idGrupo
    - **GetDataByIdGrupo (idGrup)** para mostrar los alumnos de un grupo:  
SELECT idAlumno, idGrupo, apellidosNombre, telefono, movil, dni, email  
FROM Alumnos  
WHERE idGrupo =?
    - **DeleteByIdAlumno(idAlumno)** (Puede que te interese tener también ésta para eliminar un alumno concreto):  
DELETE FROM Alumnos WHERE idAlumno = ?
  - En LNyAD construimos el adaptador y la tabla (todo estático) para cada una de las tablas de la BD. También construimos un registro del tipo de la tabla alumnos que nos servirá para hacer CRUD.

## 2.- Dar funcionalidad

### Funcionalidad de WebAlumnosFP

- **CargaCombo():** Comenzamos cargando el ddl enlazándolo a la lista de grupos que nos devolverá el método **ListaGrupos()** de la **LNyAD** a la cual le insertaremos en primer lugar el “grupo” Todos. No olvides el **.DataBind();**
- **CargaAlumnosGrupo():** A continuación cargamos el dgv con la tabla de todos los alumnos o los de un grupo, dependiendo del idGrupo seleccionado. Esta tabla nos la proporcionará el método **TablaAlumnos(idGrupo)** de la **LNyAD**
- Ocultamos las columnas de idAlumno e idGrupo
- **Eliminar Alumno:** Debe hacer:
  - Marcar el registro del alumno a borrar.
  - Lo mostrará en un panel de confirmación con un botón Si y otro No para responder a la pregunta: ¿Quieres eliminar a Pérez González, Antonio?
  - Actuará en consecuencia
- **Editar Alumno:** Debe hacer:
  - Obtener el alumno a editar a partir de su id.
  - Guardarlo en una variable de Sesión destinada a guardar el alumno
  - Saltar a la página de detalle
- **Añadir Alumno:** Debe hacer:
  - En la variable de Sesión anterior guardo null
  - Navegar a la página de detalle

### Funcionalidad de PaginaDetalleAlumno

- **CargaCombo():** Comenzamos cargando el ddl enlazándolo a la lista de grupos que nos devolverá el método **ListaGrupos()** de la **LNyAD**.
- **CargaControles():** Si la variable de sesión no es null cargamos los controles. En caso contrario hacemos que el combo indique: “Seleccione un Grupo”. Para cargar los controles tomará los datos de la variable de sesión.
- **El botón Cancelar** simplemente vuelve a la página Default sin validar.
- **El botón Aceptar:** Valida la página y, si es válida:
  - Construye un alumno a partir de los datos del formulario
  - Si la variable de Sesión es null se trata de un alumno nuevo y entonces llama al método **LNyAD.AddAlumno(alum)** que construirá un registro nuevo, lo rellenará con los datos del alumno, lo agregará a la tabla y hará Update.
  - Si la variable de Sesión no es null es que se trata de una modificación por lo que llamará al método **LNyAD.ModificaAlumno(alum)** que rellenará el registro con los datos del alumno y hará Update.
  - En cualquiera de los casos volverá a la página Default.

## Ten en cuenta

- 1) Los `if (!Page.IsPostBack)` de la página. Si no:
  - a. Puede que te encuentres con que los cambios que tú has hecho (ejemplo: seleccionar en un combo) se ignoran
  - b. Puede que te encuentres con que los controles no han cambiado
- 2) Configurar el Dataset: Consultas adecuadas con y sin campos foráneos, con parámetros, etc.
- 3) Cargar los dgv
- 4) Añadir, ocultar y mostrar columnas de los dgv
- 5) Seleccionar los eventos que vas a utilizar en los dgv: Cómo buscamos, de dónde obtenemos la fila y/o columna seleccionada, qué celda utilizaremos para buscar en el dataset; qué consulta del tableAdapter utilizaremos, etc.
- 6) Cargar los desplegables y seleccionar la parte visible y la del value. Aprender a obtener otros datos guardados en los desplegables.
- 7) Conseguir que en el desplegable del detalle no aparezca ninguno seleccionado, sino la frase "Seleccione un Grupo".
- 8) Que siempre que se borre un registro se pida confirmación.
- 9) Validar bien los controles es una parte muy importante de la gestión. Ten en cuenta especialmente:
  - a. Ojo que normalmente pongo campos obligatorios y opcionales: Ni los obligatorios se pueden dejar en blanco ni los opcionales hacerlos obligatorios. El fallo se considera el mismo
  - b. Algunos campos dependen de otros. Ejemplo: Si tengo puesto el móvil no es obligatorio el teléfono y viceversa.
  - c. Lo que no se pueda validar en el cliente hay que validarlo en el servidor.
  - d. No se nos olvide en los RangeValidator tener en cuenta el tipo de dato a validar. Ejemplo: Si los consideras string se comparan alfabéticamente, entonces el "65401" es anterior al "75"