1) Crear GestorPresupuesto.cs en carpeta servicios (tiene una instancia dao de IPresupuestoDao.cs, se ocupa de la logica como que me diga el proximo presupuesto, creo un private IPresupuestoDAO dao siendo dao un objeto y le doy directamente la interfaz que contiene los metodos, uso la interfaz pq me obliga a tener los metodos hechos, dao es la interfaz que llama a PresupuestoDao.cs junto con sus metodos)

Captura de pantalla de computadora

Descripción generada automáticamente

2) Crear IPresupuestoDAO.cs en carpeta accesoDatos (interfaz que contiene los metodos que debe implementar PResupuestoDAO al acceder a datos, me obliga a implementar los metodos en PresupuestoDAO.cs)

Texto

Descripción generada automáticamente

3) Crear PresupuestoDAO.cs en carpeta accesoDatos (va a contener los metodos encargados del presupuesto que utilizan la BD y contienen codigo SQL)

Texto

Descripción generada automáticamente

APLICACIÓN DE ESTO SACANDO EL PROXIMO\_PRESUOPUESTO DESDE FORM

1)

Texto

Descripción generada automáticamente

2)

Texto

Descripción generada automáticamente

3)

Texto

Descripción generada automáticamente

4) Patron Factory para quitar new GestorPresupuesto()

Texto

Descripción generada automáticamente

5) Creamos patron factory dentro de carpeta accesoDatos

Primero)

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Segundo)

Texto

Descripción generada automáticamente

Tercero) Tengo un gestor de presupuesto que tiene un constructor, ese constructor tiene un factory y ese factory lo que hace es crear el presupuesto.

Texto

Descripción generada automáticamente

Cuarto)

Texto

Descripción generada automáticamente

6) Ahora lo mismo con el método CargarProducto() del Form

Captura de pantalla de un celular

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

7) Ahora con el botón de Aceptar que el confirmar se encuentra en la clase Presupuesto

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

8) Aplicar patron Singleton

En el Form solo quedan los métodos referentes a la Interfaz

El Gestor se va a encargar de mandar a llamar a las funciones

El DAO repite lo de creación de conexión por lo tanto deberíamos crear una clase manipuladora que me permite tener siempre una instancia

Primero creamos una clase en accesoDatos a la cual la pueda llamar desde el DAO y no tener que siempre repetir lo de la conexión.

Primero) Crear helperDAO que seria un gestor de DAO o ayudante de DAO en carpeta accesoDatos

En HelperDAO.cs vamos a hacer toda la funcionalidad genérica que tenga que ver con los datos, quitar todo lo que se repite en PresupuestoDAO referido a la conexión a la BD.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Segundo) Creamos el Patron SIngleton para la instancia pero ahora tengo que generar a partir de ahí todos los métodos necesarios que me den la funcionalidad.

Texto

Descripción generada automáticamente

Tercero) Crear los métodos de las consultas repetitivas

Captura de pantalla de computadora

Descripción generada automáticamente

Cuarto) Ejecutar este método desde PresupuestoDAO

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Quinto) Optimizar método de ConsultaSQL en HelperDAO

Captura de pantalla de computadora

Descripción generada automáticamente

Sexto) Crear método en HelperDAO para obtener el próximo ID

Texto

Descripción generada automáticamente

Séptimo) Refactorizar método de obtención de próximo ID en PresupuestoDAO

Captura de pantalla de un celular

Descripción generada automáticamente

Octavo) Crear método en HelperDAO para Crear nuevo Presupuesto **(INCOMPLETO)**

Texto

Descripción generada automáticamente

Noveno) Refactorizar método de obtención de próximo ID en PresupuestoDAO **(INCOMPLETO)**

Texto

Descripción generada automáticamente

COSAS PARA HACER:

1. Terminar lo que no COMPLETO BOTTA (NO FUNCIONA)
2. Refactorizar presupuesto actualizar
3. Agregar los ENUM para la creación de formularios como POLLIOTTO
4. Mostrar detalle desde la misma pantalla
5. Armar Editar como Botta (2:02:11)
6. Btn\_editar
7. Update en frm\_alta\_presupuesto

Teniendo todo esto:

1. Luego de tener el Update y la Parte de Botta pasar todo a HelperDAO.
2. Dividir y organizar lo que falta

ORDEN:



PASOS

1. Frm\_Alta\_Presupuesto
2. GestorPresupuesto
3. IPresupuestoDAO
4. PresupuestoDAO
5. AbstractDAOFactory
6. DAOFactory
7. En GestorPresupuesto aplico el factory
8. Llamo el factory desde Frm\_Alta\_Presupuesto en gestor
9. HelperDao para el Patron SIngleton

Para armar carga datos UPDATE:

1. Frm\_alta\_presupuesto tener Presupuesto oPresupuesto
2. En Acción.UDPATE llamar función cargarPresupuesto con idPresupuesto
3. En cargarPresupuesto dentro de formPrincipal llamar a this.oPresupuesto para que dentro se almacene el resultado de Presupuesto obtenerPresupuestoPorId(int nro) que estará en IPresupuestoDAO
4. En PresupuestoDAO creo una función public Presupuesto ObtenerPresupuestoPorId(int nro) que me retorne dao.GetById(nro) desde el dao que es el GestorPresupuesto.
5. Creo una función GetById(int id) en HelperDAO que me devuelva el Presupuesto Completo Junto a sus Detalles con el Presupuesto Cargado.

Pasos en mi Proyecto

1. En frm\_alta\_presupuesto private Presupuesto oPresupuesto;
2. En Acción.UPDATE llamar función CargarPresupuesto(int nro\_presupuesto)
3. CargarPresupuesto(int nro\_presupuesto) llama al GestorPresupuesto que con la función ObtenerPresupuestoPorID(nro) llena el oPresupuesto mediante this.oPresupuesto = servicio.ObtenerPresupuestoPorID(nro); donde servicio es la instancia de GestorPresupuesto.
4. Creo la función ObtenerPresupuestoPorID(nro) en GestorPresupuesto retornando el Dao
5. En IPresupuestoDAO creo: public Presupuesto ObtenerPresupuestoPorID(int nro) para luego implementar el método en PresupuestoDAO
6. En PresupuestoDAO creo la función GetByID(int id) que me rellena tanto el oPresupuesto como el oDetalles y me los devuelve.
7. Defino el SQL en HelperDAO y luego me devuelve el objecto Presupuesto y lo voy pasando.

PASOS OBTENER PRESUPUESTOS LISTA

1. En Frm\_Alta\_Presupuesto
   1. 
   2. Texto

      Descripción generada automáticamente
   3. Texto

      Descripción generada automáticamente
   4. 
2. En GestorPresupuesto
   1. Texto

      Descripción generada automáticamente
3. En IPresupuestoDAO
   1. 
4. En PresupuestoDAO
   1. Texto

      Descripción generada automáticamente
5. En HelperDAO
   1. Texto

      Descripción generada automáticamente

PASOS CONSULTAR PRESUPUESTO CON SU DETALLE

1. En Frm\_Consultar\_Presupuesto
   1. Texto

      Descripción generada automáticamente
   2. Texto

      Descripción generada automáticamente
   3. Texto

      Descripción generada automáticamente
2. En Parametro
   1. Texto

      Descripción generada automáticamente
3. En GestorPresupuesto
   1. Texto

      Descripción generada automáticamente
4. En IPresupuestoDAO
   1. 
5. En PresupuestoDAO
   1. Interfaz de usuario gráfica, Texto, Sitio web

      Descripción generada automáticamente
6. En HelperDAO
   1. Texto

      Descripción generada automáticamente
   2. Forma

      Descripción generada automáticamente con confianza media

PASOS ELIMINAR PRESUPUESTO

1. En Frm\_Consular\_Presupuestos
   1. Texto

      Descripción generada automáticamente
   2. Texto

      Descripción generada automáticamente
   3. Texto

      Descripción generada automáticamente
2. En GestorPresupuesto
   1. Texto

      Descripción generada automáticamente
3. En IPresupuestoDAO
   1. 
4. En PresupuestoDAO
   1. Texto, Sitio web

      Descripción generada automáticamente
5. En HelperDAO
   1. Texto

      Descripción generada automáticamente

PARA INSERTAR NUEVO PRESUPUESTO

1. En Frm\_Alta\_Presupuesto
   1. Texto

      Descripción generada automáticamente
   2. Texto

      Descripción generada automáticamente
2. En GestorPresupuesto
   1. Texto

      Descripción generada automáticamente
3. En IPresupuestoDAO
   1. 
4. En PresupuestoDAO
   1. Captura de pantalla con la imagen de una pantalla

      Descripción generada automáticamente
5. En HelperDAO
   1. Texto

      Descripción generada automáticamente
   2. Texto

      Descripción generada automáticamente

PASOS PARA ACTUALIZAR PRESUPUESTO

1. En Frm\_Alta\_Presupuesto
   1. Texto

      Descripción generada automáticamente
   2. Texto

      Descripción generada automáticamente
   3. Texto

      Descripción generada automáticamente
2. En GestorPresupuesto
   1. Texto

      Descripción generada automáticamente
3. En IPresupuestoDAO
   1. 
4. En PresupuestoDAO
   1. Texto

      Descripción generada automáticamente
5. En HelperDAO
   1. Texto

      Descripción generada automáticamente
   2. Texto

      Descripción generada automáticamente

