El proyecto debe estar finalizado para esa mañana. Es la última fecha.

Si algún proyecto queda suspendido, los miembros del grupo tendrán que hacer, ***en 1 mes***, ***otro*** proyecto ***individual*** para poder aprobar el curso en marzo.

Comentarios generales para última revisión (3 de enero a partir de las 9 de la mañana)

* Presentación: Los formularios deben ser a pantalla completa, las cajas de texto, celdas de datagridview (si lo hay), etiquetas, deben ir acordes al texto que almacenan, si el campo es numérico que no deje teclear otros caracteres, …
* El programa debe permitir mostrar los datos que se están consultando en documento pdf, Excel o algo por el estilo.
* Se pedía que se pudiese ver el **inventario** de los productos que se están mostrando en pantalla
* Debe haber entidades para los diferentes conceptos, no solo Producto.
* Debe haber la posibilidad de realizar mantenimiento (por el administrador) de todas las tablas.
* Se deben poder hacer búsquedas por diferentes conceptos y en el caso de la descripción del producto no debe ser por valor exacto
* Todos los formularios deben tener botón o forma de volver
* Control de todas las posibles entradas de usuario en controles y de todos los posibles errores de programa (que deben llegar lo más claros posibles al o la usuaria final). 🡪 mensaje absurdo: La cadena de entrada no tiene el formato correcto
* Si hay datagridview debe tener columnas bien planteadas, incluso el tamaño de las celdas.
* Comprobar que no se van cargando continuamente formularios cargados (ver que ocurre al finalizar con )

Y además estaría bien que….

* No haya MessageBox ni InputBox sino algún tipo de formulario propio que los simule y sea más elegante
* Haya **ayudas** **claras** para el uso del proyecto, pulsando alguna tecla o botón, que ayuden a entender a un usuario o usuaria que ve por primera vez el programa.
* Pensar en el uso de teclas por si falla el ratón: <Alt>letra para controles, Return para botone de aceptación, Esc para salir, tabulador para moverse entre controles que vayan de forma lógica
* Posibilidad de hacer copias de seguridad y restaurar copia (la restauración solo administrador)

Comentarios que ya os hice en 1ª entrega

* **Copiar la BD en la solución**. Luego ir a propiedades de la bd y decirle que se copie en bin si es posterior
* Debéis **cuidar la presentación y la entrada de datos**. Pantallas completas, mensajes de error en formulario creado por vosotr@s no MessageBox que es muy feo, las cajas de texto que van a recibir números deben controlar (para eso está el evento orientado a KeyPress y luego TextChanged si copiar-pegar) que no se tecleen caracteres incorrectos, cuando pongamos el cursor en una caja de texto, estaría bien que esta se seleccionase automáticamente (para poder cambiar fácilmente el valor), que si faltan datos, el foco se ponga en el control erróneo o incluso mejor, que se cambie de color o usar el control de errores ErrorProvider.
* Cuidar los **borrados**, no se debe borrar sin pedir conformidad y si hay más tablas implicadas, avisar de las posibles consecuencias
* Seguir la convención de nombres de VS con nombres lógicos de cada capa.
* La capa de datos no debe estar pensando en la de presentación. Es al revés
* No usar siempre **while(dr.Read))** 🡪 Ver cuando es while y cuando if o sin if incluso (si ya se ha garantizado que hay filas)
* **Las clases no son las tablas**, deben tener toda la información lógica
* La idea de **acceso conectado**, es que se irá a buscar los datos cuando se necesiten, no desde el comienzo
* Comprobar que no se queden formularios cargados (ver que ocurre al finalizar con 