

1. **Leer y comprender** el documento adjunto pdf donde se explica el sistema con sus requisitos, análisis y diseño de jerarquía de clases, así como explicaciones y aclaraciones importantes. Debeis fijaros e identificar los conceptos de clases abstractas, atributos static @Override/sobreescritura de métodos, extends/herencia de clases, enum definidos, organizaciones mediante packages (2 capas)
2. **Revisad el código del proyecto**, las implementaciones de las clases, los packages creados que implementan un modelado estructurado en dos capas:
 - a. **Modelo**: organiza las clases básicas del programa (incluidos enumerados)
 - b. **Negocio**:
 - i. **clase NegocioService**: gestión de la lógica de negocio
 - ii. **clase MenuPrincipal**: interfaz/interacción con el usuario para recoger datos y mostrarlos. Aquí se sigue la siguiente estructura: como ejemplo vemos que existe un método datosProducto(), que se encarga de recoger los datos de un producto, distinguiendo los subtipos (llamando a los métodos pedirMueble(), pedirLavadora(), pedirTelevisor() según tipo de producto) y devuelve un objeto Producto. El producto devuelto será pasado como parámetro a introducirProducto(producto), para que sea almacenado en la lista de productos que tiene la clase NegocioService.
3. **Modifica el proyecto Almacen para cumplir los siguientes requerimientos**:
 - a. Crea un enum para los diferentes tipos de televisores: plasma, led, lcd, oled y utilízalo en la clase Televisor y en la recogida de datos
 - b. Implementa los metodos setPrecio() de las clases Mueble, Televisor, Lavadora. Sigue las especificaciones del documento explicativo.
 - c. Implementa los métodos en MenuPrincipal necesarios para las opciones de menú relativas a Clientes (Mayorista/Particular), siguiendo como base los métodos de Productos.
 - d. Al recoger datos de un cliente debéis calcular la letra del NIF y comprobar si es correcta. Si no lo fuera lanzar excepción y volver a solicitar el dato.
 - e. La gestión de las fechas está basado en jdk7, modifícalo para que utilice las clase de la jdk8 java.time.*. El formato de las **fechas** deberá ser como 04-agosto-17 (tanto al recogerlas como al mostrarlas). Y el almacenamiento LocalDate.
 - f. Si al recoger una fecha salta una ParseException deberás recogerla y lanzar una excepcion propia de la aplicación *FormatoFechaErroneo* mostrando el formato correcto y lo que se ha introducido mal. Pero el producto-mueble sí debe crearse.
 - g. Mejora la eficiencia de los bucles de los métodos buscarProducto(), buscarCliente() e buscarVenta()
 - h. Modifica buscarProducto() para que devuelva un objeto Producto y las llamadas a este método.
 - i. No se visualiza una venta. Solúcionalo.
 - j. El vendedor de una venta no se visualiza. Averigua porque y solúcionalo.
 - k. Para introducir una venta deben existir el producto y el cliente. Mejora el código utilizando los métodos buscaProducto() y buscaCliente().
 - l. Cuando aparezca un enum debéis mostrar todos los posibles valores pero usando algún método del enum