Una compañía de muebles española desea realizar un programa que gestione el catálogo de muebles y las tiendas que están distribuidas por el territorio español. Además, será necesario gestionar el stock de cada tienda. Para ello se necesita almacenar la siguiente información:

Mueble

EAN: Código de barras del producto (valor único).
nombre: Nombre del mueble.
color: Color/material del mueble. Debe ser uno de los siguientes (blanco, abedul, roble o negro).
estancia: Estancia de la casa para la que esta diseñado el mueble. Tomará uno de los siguientes valores (recibidor, salón, dormitorio, cocina o baño)
precio: Precio de venta del mueble.

Tienda

id: Código alfanumérico que identifica a la tienda.
ciudad: Nombre de la ciudad en la que se sitúa la tienda.
direccion: Dirección de la tienda.
telefono: Número de teléfono de la tienda.

APROBADO

La compañía de muebles necesita disponer, al menos, de las siguientes funcionalidades:

boolean addMueble(Mueble mueble)

A través de esta funcionalidad se creará y añadirá un nuevo mueble al catálogo. Se le pasará como argumento la instancia de un mueble. Se debe asegurar que el EAN del nuevo mueble no existe en el catálogo. La función devolverá verdadero si se ha insertado correctamente o falso en caso de error.

boolean addTienda(Tienda tienda)

A través de esta funcionalidad se creará y registrará una nueva tienda en la compañía. Se le pasará como argumento la instancia de una tienda. Se debe asegurar que el id de la nueva tienda no existe. La función devolverá verdadero si se ha insertado correctamente o falso en caso de error.

Mueble buscarMueble()

Esta función buscara un mueble por su EAN y lo retornará. En caso de no existir devolverá null.

Tienda buscarTienda()

Esta función buscara una tienda por su id y lo retornará. En caso de no existir devolverá null.

boolean modificarPrecio(int EAN, doble precio)

Este método permitirá modificar el precio de un mueble. Para ello la función debe recibir como parámetros el EAN del mueble y el nuevo precio. Se debe comprobar que el mueble existe y que el precio es mayor que 0. En caso de error la función devolverá false, en caso contrario true.

void mostrarMuebles()

Esta función mostrará por pantalla un listado de todos los muebles disponibles para todas las tiendas y se visualizarán todos los campos de la clase Mueble.

```
================================================
<EAN> : <nombre> <color> <estancia> <precio>
<EAN> : <nombre> <color> <estancia> <precio>
...
<EAN> : <nombre> <color> <estancia> <precio>
<EAN> : <nombre> <color> <estancia> <precio>
================================================
```

```
================================================
1845678901002 : Librería ABEDUL SALON 119,90€
1845675301008 : Silla ROBLE SALON 39,00€
1845679201001 : Aparador NEGRO RECIBIDOR 149,00€
1845678901001 : Librería BLANCO SALON 79,90€
================================================
```

void mostrarTiendas()

Esta función mostrará por pantalla un listado de todas las tiendas en el que se visualizarán todos los campos de la clase Tienda.

```
================================================  ================================================
<id> : <ciudad> <direccion> <telefono>            SA0001 : Salamanca 923 123 456  Av. Mirat
<id> : <ciudad> <direccion> <telefono>            VA0001 : Valladolid 983 123 123  Av.Salamanca
...                                               VA0002 : Valladolid 983 123 321  Paseo Zorrilla
<id> : <ciudad> <direccion> <telefono>            MA0001 : Madrid 914 456 789  Av. Santa Engracia
<id> : <ciudad> <direccion> <telefono>            ================================================
================================================
```

NOTABLE

Además la compañía de muebles necesita disponer de las siguientes funcionalidades:

boolean addStock(int EAN, String id, int stock)

Esta función añadirá un nuevo producto al stock de una tienda. Se recibirá como parámetro el EAN del mueble, el id de la tienda y el stock. Se creará un nuevo objeto de tipo Stock y se almacenará en una colección en la clase Tienda asegurándose de que no puede haber dos muebles con el mismo EAN en la misma tienda. Sí puede darse el caso de que un mismo EAN aparezca en dos tiendas diferentes. También se debe comprobar que el mueble y la tienda existan. La función devolverá verdadero si se ha insertado el stock correctamente o falso en caso de error.

void mostrarStock(String id)

Esta función mostrará por pantalla un listado de los muebles que tiene disponibles en la tienda indicada. La función recibe como parámetro el id de la tienda. El listado estará ordenado alfabéticamente por nombre del mueble. En caso de que el nombre sea el mismo se ordenará por EAN. La salida debe tener un formato como el siguiente:

```
<Id> : <ciudad> (<telefono>)                      SA0001 : Salamanca (923 123 456)
================================================  ================================================
<EAN> : <nombre> <color> <estancia> <precio>      1845679201001 : Aparador NEGRO RECIBIDOR 149,00€
<EAN> : <nombre> <color> <estancia> <precio>      1845678901001 : Librería BLANCO SALON 79,90€
...                                               1845678901002 : Librería ABEDUL SALON 119,90€
<EAN> : <nombre> <color> <estancia> <precio>      1845675301008 : Silla ROBLE SALON 39,00€
<EAN> : <nombre> <color> <estancia> <precio>      ================================================
================================================
```

boolean modificarStock(int EAN, String id, int stock)

Esta función modificará el stock de un producto en una tienda. Se recibirá como parámetro el EAN del mueble, el id de la tienda y el nuevo stock. Se debe comprobar que la tienda y el EAN existen. Además se debe comprobar que el stock es un valor positivo. En caso de error la función devolverá false, en caso contrario true.

SOBRESALIENTE

Además la compañía de muebles necesita disponer de las siguientes funcionalidades:

List<Registro> filtrarMueble(String id , Filtro filter)

A través de esta funcionalidad se podrán filtrar los muebles de una tienda con el fin de obtener una lista con aquellos muebles que cumplan una determinada característica impuesta por el filtro. La función recibirá como parámetros el id de la tienda junto con el filtro a aplicar (`Filtro`). En caso de que la tienda no exista, la función devolverá null.

Se podrán aplicar distintos tipos de filtros, y la obtención se realizará a través de las siguientes funcionalidades estáticas:

static Filtro filtroColor(Color color)

Se indicará el color y devolverá únicamente los muebles del color indicado.

static Filtro filtroEstancia(Estancia estancia)

Se indicará la estancia y devolverá únicamente los muebles de la estancia indicada.

static Filtro filtroPrecioMenor(Double precio)

Retornará únicamente los muebles con un precio inferior o igual al indicado.

Además también será necesario crear los siguientes filtros:

Filtro Not()

Devuelve una condición Not que invertirá el resultado de los filtros anteriores.

Filtro Or(Filtro f)

Devuelve una condición Or que aplicará una condición lógica OR sobre los filtros.

Filtro And(Filtro f)

Devuelve una condición And que aplicará una condición lógica AND sobre los filtros.

- **Se entiende que algo está correctamente implementado cuando la ejecución del programa lo demuestre.**
- **Se debe justificar la elección de los contenedores empleados mediante comentarios en el código.**
- **Para entregar la aplicación:**
  - Debes comprimir tu aplicación NetBeans que obligatoriamente tendrá como nombre tus apellidos y nombre (si te llamas Juan Carlos García Vaquero, a tu proyecto le deberás nombrar como GarciaVaqueroJuanCarlos). El fichero ZIP que subirás al Moodle también tendrá el mismo nombre que tu proyecto.