

Construir un proyecto para controlar las compras y ventas que se realizan en una tienda ficticia. El tipo **Tienda** se define por su **nombre** su **saldo** actual y su lista de movimientos. En el esquema puedes ver el tipo **Movimiento**

Para realizar el proyecto necesitamos también los siguientes elementos:

Un **Catalogo** con los dos siguientes tipos de artículos:

- **Producto**, que tiene los campos **nombre**, **costo**, **beneficio** y **cantidad**, cuyos tipos y significados se indican en el esquema que se te entrega. Debe tener un método PVP() que calcula el precio de venta al público añadiendo el porcentaje de beneficio al precio de costo.
- **Alimento**, que tiene los mismos campos que Producto, más un entero **fechaCad**, que será la fecha de caducidad en formato AAMMDD. Además su PVP() sólo aumenta un 50% al precio de costo. Importante: Al construir un artículo, nombre y costo son obligatorios, pero beneficio y cantidad se pueden omitir, y entonces se supone que sus valores serán respectivamente 50% y 10.

En el punto de entrada se construirá un catálogo con los artículos que tienes en el esquema.

Luego se construirá la «Tienda2014» con un saldo de 12000 euros.

A continuación se establecerá una variable llamada **fechaMovis** a 140102, y se entrará en el menú (lo tienes en Util.cs) con estas opciones:

- 1.- **Listar Catálogo.** [índice/ nombre / cantidad / costo / P.V.P/ Caducidad].
(Nota: En el listado, el índice de los artículos comienza en 1)
- 2.- **Fecha Movimientos:** Pide el año [14 a 20], mes y día (te conviene que sea en este orden) para establecer la nueva fecha **fechaMovis** de los movimientos que hagas. Esta fecha no puede ser anterior a la se hubiera establecido antes.
- 3.- **Comprar.** Pedirá el índice del producto a comprar. Si se elige índice 0, significa “nuevo artículo” (en este caso, presenta un formulario para introducir los datos del nuevo artículo) y lo añade al catálogo. Si se elige uno existente, sólo te preguntará la cantidad y —si el elemento lo necesita, es decir, si se introdujo— la fecha de caducidad. En este caso, se supone que todas las unidades del producto comprado se quedan con la nueva fecha de caducidad.
- 4.- **Vender:** Presenta los productos en existencia y un “formulario” para elegir el producto a vender y la cantidad. Cuando se agota el stock de un producto no se elimina del catálogo.
- 5.- **Lista de Movimientos.** [fecha / C o V. / Artículo / cantidad / Euros / Saldo.

Nota: Se valorará especialmente el control de errores. Ejemplos: control del tipo esperado en una pregunta de formulario; el stock actual del producto para poder vender; si tienes dinero para poder comprar; que no haya artículos repetidos; etc.