

Proyectos integradores, grupales.

Proyecto Ejemplo

En un supermercado se comercializan varios productos. De cada producto se conoce su código, su marca, precio, fecha vencimientos (DateTime), stock y código del proveedor. Además tiene una lista de proveedores que le entregan mercadería para reponer stock. De cada proveedor se conoce su nombre y apellido, dni, nro de proveedor, lista de productos que repone.

Se deberá desarrollar una aplicación, utilizando las clases que considere necesarias, que resuelva las funcionalidades que se muestra en el siguiente menú:

- a) Vender un producto. Debe hacerse control de stock. En caso de no haber suficiente stock se debe levantar una excepción indicando lo sucedido y se debe enviar un correo al proveedor solicitando la reposición del producto.
- b) Modificar el precio de un producto.
- c) Eliminar un proveedor por su código. Deben modificarse todos los productos que eran entregados por dicho proveedor.
- d) Listado de productos vencidos que deben ser devueltos a sus proveedores.
- e) Listado de proveedores indicando nombre de proveedor y cantidad total de productos que entrega.
- f) Agregar un producto al supermercado. Se debe detallar su proveedor. Si el proveedor no existe hay que agregarlo a la lista de proveedores. Si ya existe, se le debe agregar el producto a su lista.

Clases a definir:

Supermercado: tiene un arrayList de productos y un arrayList de proveedores

Producto: tienen codigo, marca, stock, precio, fecha vencimiento, codigo proveedor

Proveedor: tiene codigo, nom y ape, dni, y una lista de productos que reparte o provee

Clase Supermercado → métodos básicos

Constructor

agregarProd(Producto unP)

eliminarProd(Producto unP)

cantidadProd()

existeProd(Producto unP)

verProd(i) retorna el producto iesimo

todosProd () retorna el arrayList de productos

agregarProv(Proveedor unProv)

eliminarProv(Proveedor unProv)

cantProv()

verProve(k) retorna el kesimo proveedor

todosProv() retorna la lista de proveedores completa

Clase Producto → métodos básicos

Constructor

Propiedades para cada variable de instancia (con su get y set)

Clase Proveedor → métodos básicos

Constructor (inicializa nombre, dni, codigo, y el arrayList vacío)

Propiedades para los atributos nombre, dni ,codigo (con get y set)

agregarProd(Producto unP)

eliminarProd(Producto unP)

cantidad()

verProd(i) retorna el producto iesimo

todosP() retorna la lista de productos

En aplicación debe crear un supermercado y cargarle un par de proveedores.

Luego implementar las funciones necesarias para responder a los ítems del menú.

```
public static void Main()
```

```
{
```

```
    Super sup;
```

```
    string sigue;
```

```
    sup=new Super();
```

```
    /*crea el supermercado*/
```

```
    while (true)
```

```
    {
```

```
        Console.WriteLine("Elija una opción: ");
```

```
        Console.WriteLine("1- agrega producto ");
```

```
        Console.WriteLine("2- Modifica precio de producto ");
```

```
        Console.WriteLine("3 – Elimina proveedor ");
```

```
        Console.WriteLine("4- Listado de productos vencidos");
```

```
        Console.WriteLine("5 – Listado de proveedores ");
```

```
        Console.WriteLine("6- Vender producto ");
```

```
        Console.WriteLine("7- Agrega proveedor ");
```

```
        Console.WriteLine("8- Elimina producto ");
```

```
        Console.WriteLine(" 9 - Salir ");
```

```
        string op=Console.ReadLine();
```

```
        switch (op)
```

```
        {
```

```
            "1": { agregarProducto(sup) }
```

```
            "2":
```

```
                { int nroProd;
```

```
                  float nuevoP;
```

```
                Console.WriteLine("ingrese codigo de producto y precio nuevo");
```

```
                nroProd=int.Parse(Console.ReadLine());
```

```
                nuevoP=float.Parse(Console.ReadLine());
```

```
                modificaPrecio( sup, nroProd, nuevoP)
```

```
            }
```

```

"3":
{   int nroProve;

    Console.WriteLine("ingrese codigo del proveedor");
    nroProve=int.Parse(Console.ReadLine());

    eliminaProveedor(ref sup, nroProve)
}

"4":
{ listaVencidos(sup) }

"5": { listaProvedores(sup) }

"6": {   int cod,canti;

    Console.WriteLine("ingrese codigo del producto a comprar");
    cod=int.Parse(Console.ReadLine());

    Console.WriteLine("ingrese cantidad a comprar");
    canti=int.Parse(Console.ReadLine());

    venderProd(sup, cod, canti)
}

"7":
{
Proveedor pro;
int nroProv,dni;
string nombre;

Console.WriteLine("ingrese codigo, nombre, dni de un proveedor");
nroProv=int.Parse(Console.ReadLine());
dni=int.Parse(Console.ReadLine());
nombre=Console.ReadLine();

prov=new Proveedor(cod,dni, nombre); /*crea el proveedor y lo carga*/
sup.agregarProv(prov);               /*agrega el proveedor al super*/
}

"8":
{   int nroPro;

    Console.WriteLine("ingrese codigo del producto");
    nroPro=int.Parse(Console.ReadLine());

    eliminaProd(ref sup, nroPro)
}

"9": {
    Console.WriteLine("sale del sistema");

```

```

        Console.ReadKey();
        break;
    }
}
}

public static void agregarProducto(Super s)
{
    int cod, st, nroprov;
    string marca;
    float precio;
    int dia, mes, anio;

    Producto p;
    Console.WriteLine("ingrese codigo, stock, marca, precio y nro de proveedor de un producto");
    cod=int.Parse(Console.ReadLine());
    st=int.Parse(Console.ReadLine());
    marca=Console.ReadLine();
    precio=float.Parse(Console.ReadLine());
    nroprov=int.Parse(Console.ReadLine());
    Console.WriteLine("ingrese dia, mes y año de vencimiento");
    dia= int.Parse(Console.ReadLine());
    mes= int.Parse(Console.ReadLine());
    anio= int.Parse(Console.ReadLine());
    feven=new DateTime(anio,mes,dia);

    p=new Producto(cod,st,marca,precio,nroprov,feven);           /*crea producto*/
    s.agregarProd(p);                                           /*lo agrega al super*/

    ArrayList conjProv=new ArrayList();
    bool existe=false;
    conjProv=s.todosProv();                                     /*recupera lista de proveedores*/

    foreach(Proveedor ele in conjProv) /*busco si ya existe el proveedor*/
    {
        if ( ele.Codigo== nroProv)
        {
            existe=true;
            ele.agregarProd(p);                                  /*le agrego al proveedor el nuevo producto a su lista*/
            break;
        }
    }
    If ( !existe)                                               /*agrego el nuevo proveedor a supermercado*/
    {
        Proveedor pro;
        Console.WriteLine("ingrese nombre, dni de un proveedor");
        dni=int.Parse(Console.ReadLine());
        nombre=Console.ReadLine();
        prov=new Proveedor(nroProv,dni, nombre); /*crea el proveedor del nuevo producto*/
        prov.agregarProd(p);                                  /*le agrego al proveedor el producto*/
        s.agregarProv(prov);                                  /*agrega el proveedor al super*/
    }
}

```

```

Console.WriteLine("El producto se agrego con éxito");
Console.ReadKey();
}

public static void modificaPrecio( Super sup, int nroProd, float nuevoP)
{
    ArrayList conjProd;
    conjProd= sup.todosProd();           /*tomo la lista de productos*/
    foreach (Producto ele in conjProd)
    { if (ele.Codigo==nroProd)
        {
            ele.Precio=nuevoP;
            break;
        }
    }
    Console.WriteLine("El producto se modifiko con éxito");
    Console.ReadKey();
}

public void eliminarProveedor ( Super su, int numProv)
{
    ArrayList conjProv;
    bool esta=false;
    conjProv= su.todosProv();           /*tomo la lista de porveedores*/
    foreach (Proveedor ele in conjProv)
    { if (ele.Codigo==numProv)
        {
            esta=true;
            su.eliminarProv(ele);
            break;
        }
    }
    If (esta)
    {Console.WriteLine("El proveedor se elimino con éxito");
    Console.ReadKey();
    }
}

public void listaVencidos (Super su)
{
    ArrayList conjProd;
    int resul;
    conjProd=su.todosProd();           /*tomo la lista de productos*/
    foreach (Producto elem in conjProd)
    {
        resul=DateTime.compare(elem.FechaVen,DateTime.today);
        if ( resul >0)
    }
}

```

```

        Console.WriteLine("producto vencido " + elem.Codigo + " " + elem.CodProv);

        Console.ReadKey();
    }
}

```

```

public void listaProveedores (Super s)
{
    ArrayList conjProv;

    conjProv= su.todosProv;      /*tomo la lista de proveedores*/
    foreach (Proveedor ele in conjProv)
    {
        Console.WriteLine(ele.CodigoP + " " + ele.Nombre + " " + ele.cantidadP())
    }
    Console.ReadKey();
}

```

```

public void venderProd( Super s, int codP, int canU)
{
    try
    {
        ArrayList conjProd;
        conjProd= s.todosProd();
        foreach( Producto el in conjProd)
        {
            If ( el.Codigo= codP)
            { if (el.Stock < canU)
                throw new stockInsuficienteException();
            }
            else
            { el.Stock -=canU;
                break;
            }
        }
    }
    catch (stockInsuficienteException)
    { Console.WriteLine("No hay stock suficiente. ");
        enviarMail(el.CodProv);    /*no se implementa*/
    }
}

```

```

public class stockInsuficienteException: Exception { }

```