



Código de la solución

//Ejercicio 1

//En Sistema.cs

```
public double MontoTotalPorFechasYTipo(DateTime fechaDesde, DateTime
fechaHasta, TipoCliente tipo){
    double montoTotal = 0;
    foreach (var item in this.Clientes){
        if (item.Tipo == tipo){
            foreach (var compra in item.Compras){
                if (compra.Fecha >= fechaDesde && compra.Fecha <=
fechaHasta)
                    montoTotal += compra.CalcularCostoFinal();
            }
        }
    }
    return montoTotal;
}
```

//En Compra.cs

```
public double CalcularCostoFinal(){
    double costoFinal = 0;
    short cantidadTotalComprada = 0;
    foreach (var item in this.Muebles){
        costoFinal += item.Mueble * item.Cantidad;
        cantidadTotalComprada += item.Cantidad;
    }

    if (cantidadTotalComprada >= 3)
        return costoFinal * 0.95;
    else
        return costoFinal;
}
```

//En Mueble.cs

```
public abstract double CalcularCosto(bool Financiada);
```

//En Importado.cs

```
public override double CalcularCosto(bool Financiada){
    if (Financiada)
        return this.Costo * 1.20;
    else
        return this.Costo * 1.15;
}
```

//En Nacional.cs

```
public override double CalcularCosto(bool Financiada){
    if (Financiada)
        return this.Costo * 1.05;
    else
        return this.Costo * 0.95;
}
```

//Ejercicio 2

//En Sistema.cs

```
public List<Cliente> ClientesConCompra(){
    List<Cliente> ClientesQueCompraron = new List<Cliente>();
    foreach (var item in this.Clientes){
        if (item.Compras.Count > 0)
            ClientesQueCompraron.Add(item);
    }
    return ClientesQueCompraron.Sort();
}
```

//Ejercicio 3

//En controller

```
[HttpGet]
public IActionResult ComprasPorCliente(int id){
    if (id > 0){
        List<Compra> compras = instancia.GetComprasPorCliente(id);
        return View(compras);
    } else {
        ViewBag.Mensaje = "Id de cliente inválido.";
        return View("Index", "Home");
    }
}
```

//En Sistema.cs

```
public List<Compra> ComprasPorCliente(int Id){
    foreach (var item in this.Clientes){
        if(item.Id == Id){
            return item.Compras; //O hacer un foreach y agregar a
una lista de retorno
        }
    }
}
```

//En ComprasPorCliente.cshtml

```
@model IEnumerable<Examen.Models.Compra>
<table class="table">
    <thead>
```

```

        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Id)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Fecha)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.NroTicket)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Financiada)
            </th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.Id)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Fecha)
                </td>
                <td>
                    @Html.DisplayFor(modelItem =>
item.NroTicket)
                </td>
                <td>
                    @Html.DisplayFor(modelItem =>
item.Financiada)
                </td>
            </tr>
        }
    </tbody>
</table>

```