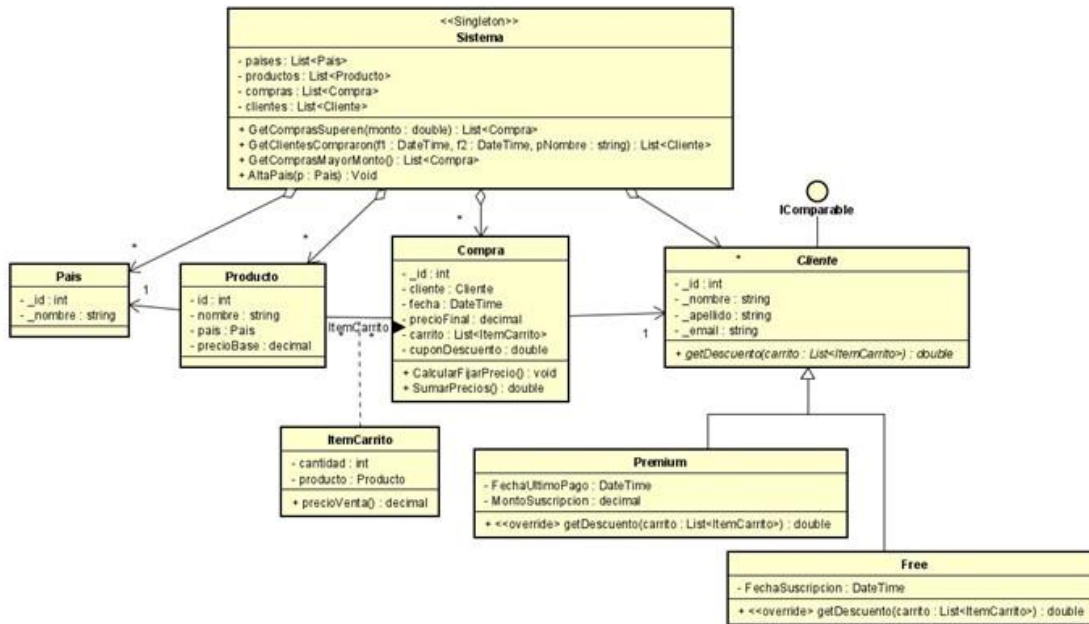


Solución examen Programación 2 | Febrero 2024



```

//EN Sistema
//Parte A
//a. Implementar el método que dado un monto permita obtener todas las compras que lo superen
public List<Compra> GetComprasSuperen(double monto)
{
    List<Compra> ret = new List<Compra>();
    foreach (Compra c in Compras)
    {
        if (c.PrecioFinal > monto)
        {
            ret.Add(c);
        }
    }
    return ret;
}

//En Compra
public void CalcularFijarPrecio()
{
    if (PrecioFinal.equals(0))
    {
        double descuentoCliente = Cliente.GetDescuento(carrito);
        double sumaPrecios = SumarPrecios();
        PrecioFinal = sumaPrecios * descuentoCliente;
        if (cuponDescuento < 1)
        {
            PrecioFinal = PrecioFinal * cuponDescuento;
        }
    }
}

public double SumarPrecios()
{
    double suma = 0;
    foreach (ItemCarrito ic in carrito)
    {

```

```

        suma += ic.Producto.Precio * ic.Cantidad;
    }
    return suma;
}
//En Cliente
public abstract double GetDescuento(List<ItemCarrito> carrito);
//En FREE
public override double GetDescuento(List<ItemCarrito> carrito)
{
    double descuento = 1;
    int CantidadProductos = 0;
    foreach (ItemCarrito ic in carrito)
    {
        CantidadProductos += ic.Cantidad;
    }
    if (CantidadProductos > 10 && FechaSuscripcion > new DateTime(2020, 01, 01))
    {
        descuento = 0.95;
    }
    return descuento;
}
//En PREMIUM
public override double GetDescuento(List<ItemCarrito> carrito)
{
    double descuento = 1;
    int CantidadProductos = 0;
    foreach (ItemCarrito ic in carrito)
    {
        CantidadProductos += ic.Cantidad;
    }
    if (CantidadProductos > 10)
    {
        descuento = 0.90;
    }
    return descuento;
}
/*b. Dado el nombre de un producto y dos fechas, obtener los
clientes que lo compraron entre esas fechas inclusive ,
ordenado por nombre ascendentemente. Si un cliente lo hubiera
comprado más de una vez, este no se debe repetir en la lista.*/
//EN cliente
public int CompareTo(Cliente other)
{
    if (Nombre.CompareTo(other.Nombre) > 1)
    {
        return 1;
    }
    else if (Nombre.CompareTo(other.Nombre) < 1){
        return -1;
    }else
    {
        return 0;
    }
}
}

```

```

//EN SISTEMA
public List<Cliente> GetClientesCompraron(DateTime f1, DateTime f2, string pNombre)
{
    Producto aux = new Producto();
    aux.Nombre = pNombre;

    List<Cliente> ret = new List<Cliente>();
    foreach (Compra c in Compras)
    {
        if (c.Carrito.Contains(aux) && c.Fecha > f1 && c.Fecha < f2 & !ret.Contains(c.Cliente))
        {
            ret.Add(c.Cliente);
        }
    }
    ret.Sort();
    return ret;
}

//En producto
public override bool Equals(object obj)
{
    return obj is Producto producto && Nombre == producto.Nombre;
}

/*c. Listar las compras de mayor monto.
Si hubiera más de una compra con el mismo monto,
y este sea el mayor, se deben mostrar todas*/
//EN SISTEMA
public List<Compra> GetComprasMayorMonto()
{
    List<Compra> ret = new List<Compra>
    double max = 0;
    foreach (Compra c in Compras)
    {
        if (c.PrecioFinal > max)
        {
            ret.Clear();
            max = c.PrecioFinal;
            ret.Add(c);
        }
        else if (c.PrecioFinal == max)
        {
            ret.Add(c);
        }
    }
    return ret;
}

/*
MVC: Implementar lo que considere necesario a nivel de vista, controlador y dominio que
permita resolver lo siguiente: */
//Agregar un Pais al sistema.
//En la vista
< form method = "POST" >
< input type = "text" name = "Nombre" placeholder = "Nombre" >
< input type = "text" name = "Codigo" placeholder = "Código" >

```

```

< input type = "submit" value = "Guardar" >
</ form >
< p > @ViewBag.resultado </ p >
//Controller
//Cuento con una instancia del Sistema singleton que llamo s
public IActionResult Create()
{
    return View();
}
[HttpPost]
public IActionResult Create(Pais p)
{
    try
    {
        s.AltaPais(p);
        ViewBag.resultado = "Alta correcta";
    }
    catch (Exception e)
    {
        ViewBag.resultado = e.message;
    }
}
//en sistema
public void AltaPais(p)
{
    Países.Add(p);
}
//Listado de productos con todos sus datos
//Vista Index.cshtml
@model List<Producto>
@foreach(Producto p in Model)
{
< p > @p.Id @p.Nombre @p.Pais.Nombre @p.PrecioBase </ p >
}
//Controller
public IActionResult Index()
{
    return View(s.GetProductos());
}

```