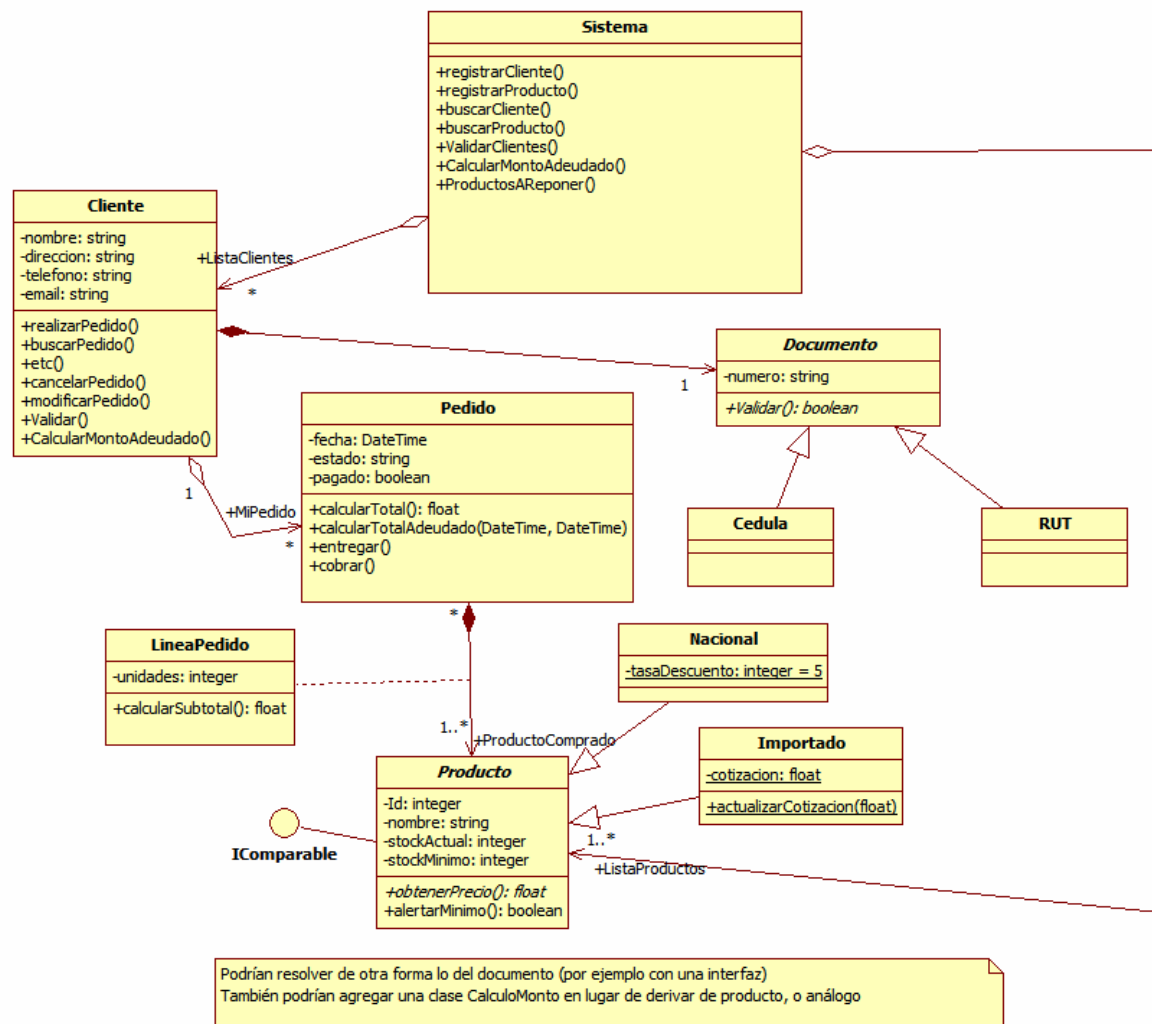


EVALUACION	SOLUCION EXAMEN	GRUPO	TODOS	FECHA	17/05/2013
MATERIA	PROGRAMACIÓN 2				
CARRERA	AP/ATI/APW				
CONDICIONES	<p>- Puntos: 100</p> <p>- Duración: 3 Horas</p> <p>- Sin material</p> <p>- Otros : No escriba la hoja de la letra</p> <p>Consultas solamente sobre interpretación de letra y sintaxis específica del lenguaje.</p> <p>Numerar las hojas entregadas.</p> <p>Indicar nombre del docente del curso en primera hoja del examen</p>				

1)



Punto 2a)

Clase cliente:

```
}  
    public bool Validar()  
    {  
        return  
            !string.IsNullOrEmpty(this.nombre) &&  
            !string.IsNullOrEmpty(this.telefono)  
            && !string.IsNullOrEmpty(this.direccion) &&  
            !string.IsNullOrEmpty(this.email)  
            && this.MiDocumento.validar();  
    }  
  
public abstract class Documento  
{  
    public long Numero { get; set; }  
  
    public abstract bool validar();  
}  
  
public class Cedula : Documento  
{  
    public override bool Validar()  
    {  
        return  
            this.Numero.ToString().Length >= 6 &&  
            this.Numero.ToString().Length <= 7;  
    }  
}  
public class RUT : Documento  
{  
    public override boolean Validar()  
    {  
        {  
            return this.Numero.ToString().Length == 12;  
        }  
    }  
}
```

Clase Sistema:

```
public List<Cliente> ValidarClientes(List<Cliente> lista)  
{  
    List<Cliente> listaAux= new List<Cliente> ();  
    foreach (Cliente cli in this.ListaClientes)  
    {  
        if !(cli.Validar())  
            listaAux.Add(cli);  
    }  
    return listaAux;  
}
```

Punto 2 b)

```
public class LineaPedido
{
    private int unidades ;
    private Producto producto;
    public float calcularSubtotal()
    {
        return this.unidades * this.producto.obtenerPrecio();
    }
}

public class Pedido
{
    public float calcularTotalAdeudado(System.DateTime f1, System.DateTime f2)
    {
        if (this.fecha >=f1 && this.fecha<=f2
            && this.estado=="entregado"
            && !this.pagado
        )

            return this.calcularTotal();
        else
            return 0;
    }
    public float calcularTotal()
    {
        float total = 0;
        foreach (LineaPedido l in this.ProductosComprados)
            total+=l.calcularSubtotal();
        return total;
    }
}

public abstract class Producto : IComparable
{
    protected float precio;
    public abstract float obtenerPrecio();
}

public class Nacional : Producto
{
    private static integer tasaDescuento = 5;
    public override float obtenerPrecio()
    {
        return this.precio * tasaDescuento /100;
    }
}
```

```
public class Importado : Producto
{
    private static float cotizacion;
    public override float obtenerPrecio()
    {
        return this.precio * Producto.cotizacion;
    }
}

public class Sistema
{
    .....
    public float CalcularMontoAdeudado(System.DateTime f1, System.DateTime f2)
    {
        float total = 0;
        foreach (Cliente cli in this.ListaClientes)
        {
            total += cli.CalcularMontoAdeudado(f1, f2);
        }
        return total;
    }
}

public class Cliente
{
    public float CalcularMontoAdeudado(DateTime f1, DateTime f2)
    {
        float total = 0;
        foreach (Pedido p in this.MisPedidos)
        {
            total += p.CalcularTotal(f1, f2);
        }
        return total;
    }
}
```

Punto 2d)

```
public abstract class Producto : System.IComparable
{
    public bool alertarMinimo()
    {
        return this.stockActual <= this.stockMinimo;
    }

    public int CompareTo(object obj)
    {
        Producto p = obj as Producto;
        if (p == null) return 1;
        return this.stockActual.CompareTo(p.stockActual);
    }
}
```

En la clase Sistema:

```
public List<Producto> ProductosAReponer()  
{  
    List<Producto> lista = new List<Producto>();  
    foreach (Producto p in this.ListaProductos)  
        if (p.alertarMinimo())  
            lista.Add(p);  
    lista.Sort();  
    return lista;  
}
```