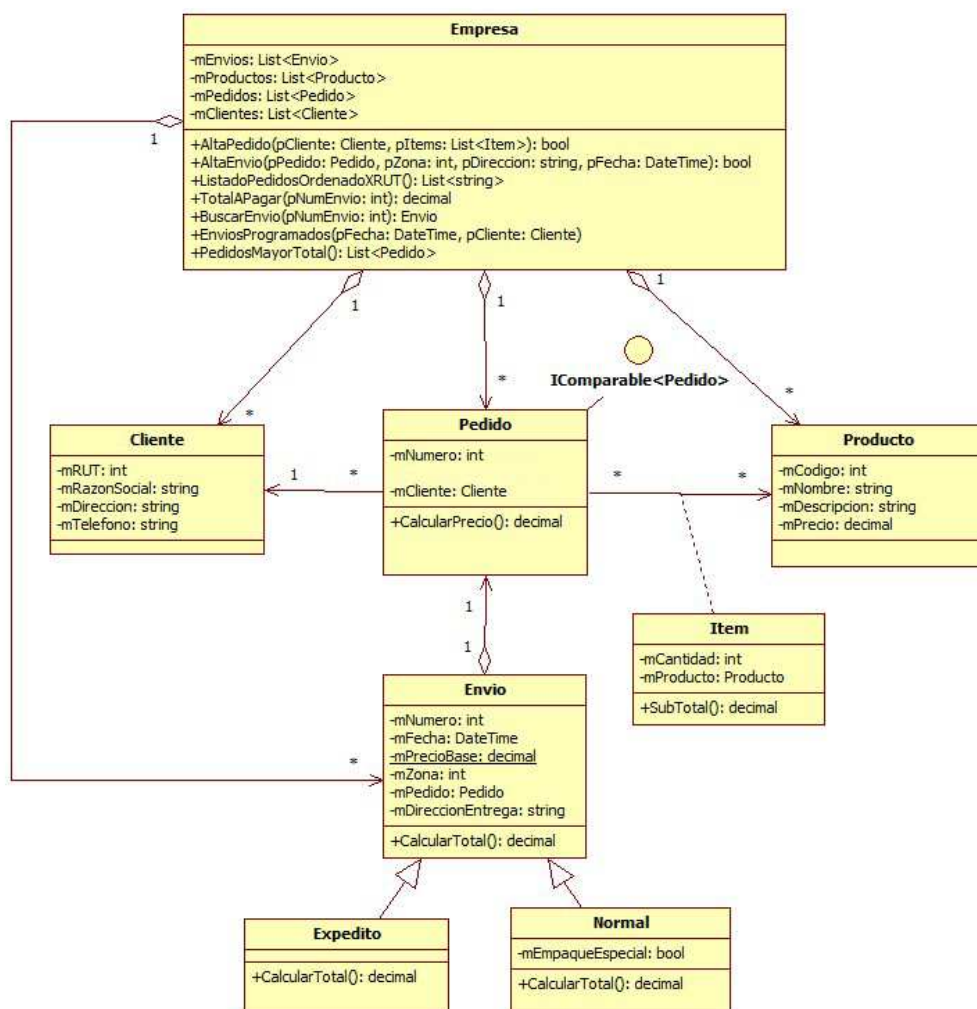


EVALUACIÓN	<b>SOLUCION EXAMEN</b>	GRUPO	TODOS	FECHA	26/02/13
MATERIA	Programación 2 – Plan 2011				
CARRERA	AP – ATI - APW				
CONDICIONES	<ul style="list-style-type: none"> <li>- Puntos: 100</li> <li>- Duración 3 hs – incluyendo lectura de la letra.</li> <li>- Sin material</li> <li>- Indicar el nombre del docente en la primera hoja.</li> <li>- Dudas exclusivamente de la letra o sintaxis no trivial de C#.NET.</li> <li>- No escribir en la hoja de la letra.</li> <li>- No entregar la hoja de la letra.</li> <li>- Numerar las hojas</li> </ul>				

**SOLUCIÓN:**

1.



2.b.

//EN CLASE EMPRESA

```
public List<string> ListadoPedidosOrdenadoXRUT() {  
  
    List<Pedido> aux = new List<Pedido>();  
    List<string> listado = new List<string>();  
  
    foreach (Pedido ped in this.Pedidos) {  
        if (ped.CalcularPrecio() > 500000) {  
            aux.Add(ped);  
        }  
    }  
  
    aux.Sort();  
  
    foreach (Pedido ped in aux) {  
        listado.Add("Número: " + ped.Numero +  
            ", RUT: " + ped.Cliente.RUT);  
    }  
  
    return listado;  
}
```

//EN CLASE PEDIDO

```
public class Pedido : IComparable<Pedido>  
{  
    ...  
  
    int IComparable<Pedido>.CompareTo(Pedido otro) {  
        if (otro != null) {  
            return otro.Cliente.RUT - this.Cliente.RUT;  
        }  
  
        return 0;  
    }  
  
    public decimal CalcularPrecio() {  
        decimal precio = 0;  
  
        foreach (Item item in this.Items) {  
            precio += item.SubTotal();  
        }  
  
        if (precio > 100000) {  
            precio = precio - (precio * (5 / 100));  
        }  
  
        return precio;  
    }  
}
```

2.c.

```
//EN CLASE EMPRESA

    public decimal TotalAPagar(int pNumEnvio) {
        Envio env = this.BuscarEnvio(pNumEnvio);

        if (env != null) {
            return env.CalcularTotal();
        }

        return 0;
    }

    public Envio BuscarEnvio(int pNumEnvio) {
        foreach (Envio env in this.Envios) {
            if (env.Numero == pNumEnvio) {
                return env;
            }
        }

        return null;
    }

//EN CLASE ENVIO

    public decimal CalcularTotal() {
        decimal total = Envio.PrecioBase;

        if (this.Zona == 2) {
            total = total * 1.1M;
        }

        return total + this.Pedido.CalcularPrecio();
    }

//EN CLASE EXPEDITO

public class Expedito : Envio { ...

    public override decimal CalcularTotal() {
        decimal total = base.CalcularTotal();

        return total * 1.03M;
    }

//EN CLASE NORMAL
public class Normal : Envio { ...

    public override decimal CalcularTotal()
    {
        decimal total = base.CalcularTotal();

        if (this.EmpaqueEspecial)
        {
            total = total * 1.01M;
        }

        return total;
    }
}
```

2.d.

```
//EN CLASE EMPRESA
```

```

public List<Envio> EnviosProgramados(DateTime pFecha,
                                     Cliente pCliente) {
    List<Envio> aux = new List<Envio>();

    foreach (Envio envio in this.Envios) {
        if (envio.Pedido.Cliente == pCliente &&
            envio.Fecha == pFecha) {
            aux.Add(envio);
        }
    }

    return aux;
}

```

**2.e.**

//EN CLASE EMPRESA

```

public List<Pedido> pedidosMayorTotal()
{
    List<Pedido> aux = new List<Pedido>();
    decimal mayorTotal = 0;

    foreach (Pedido pedido in this.Pedidos)
    {
        decimal totalPedido = pedido.CalcularPrecio();

        if (totalPedido >= mayorTotal)
        {
            if (totalPedido > mayorTotal)
            {
                mayorTotal = totalPedido;
                aux.Clear();
            }

            aux.Add(pedido);
        }
    }

    return aux;
}

```