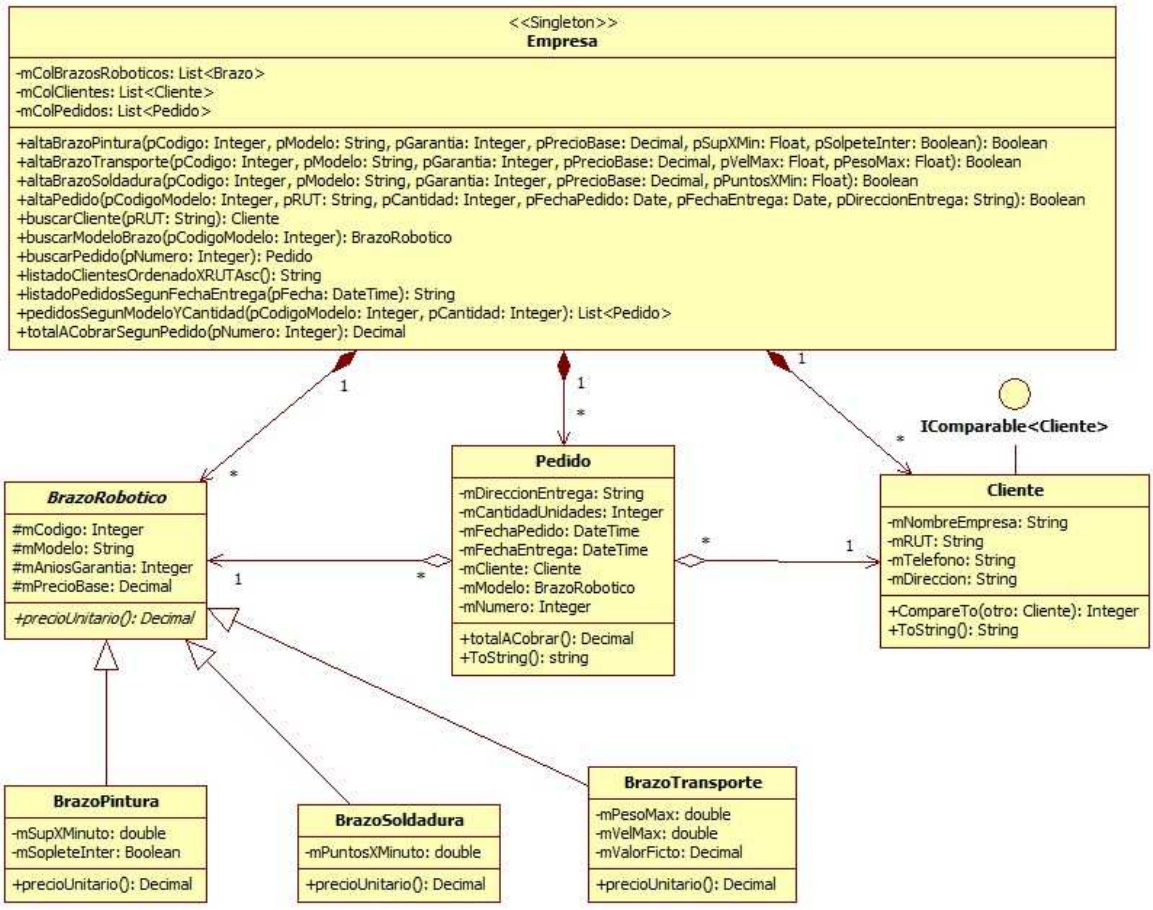
EVALUACIÓN	SOLUCION EXAMEN	GRUPO	TODOS	FECHA	26/10/2012
MATERIA	Programación 2 – Plan 2011				
CARRERA	AP – ATI - APW				
CONDICIONES	- Puntos: 100 - Duración 3 hs – incluyendo lectura de la letra. - Sin material - Indicar el nombre del docente en la primera hoja. - Dudas exclusivamente de la letra o sintaxis no trivial de C#.NET. - No escribir en la hoja de la letra. - No entregar la hoja de la letra. - Numerar las hojas				

NOTA: Se presenta una idea de solución no detallada.

Parte 1:



Parte 2.c:

En clase Cliente:

```
public class Cliente : IComparable<Cliente>
{
	...
	...

	public int CompareTo(Cliente otro)
	{
		if (otro != null)
		{
			return this.RUT - otro.RUT;
		}

		return 0;
	}

	public override string ToString()
	{
		return "RUT: " + this.RUT + " - Nombre: " + this.NombreEmpresa
	}
}
```

En clase Empresa:

```
public string listadoClientesOrdenadoXRUTAsc()
{
	string listado = "";
	List<Cliente> aux = this.ColClientes.Clone();

	aux.Sort();

	foreach (Cliente unCli in aux)
	{
		listado += unCli.ToString() + "\r\n";
	}

	return listado;
}
```

Parte 2.d:

En clase Pedido:

```
public override string ToString()
{
      return "Pedido Nro.: " + this.Numero.ToString() + " - Cliente: " +
      this.Cliente.Nombre + " - Tel. Contacto: " +
      this.Cliente.Telefono.ToString();
}
```

En clase Empresa:

```
public string listadoPedidosSegunFechaEntrega(DateTime pFecha)
{
      string listado = "";

      foreach (Pedido unPedido in this.ColPedidos)
      {
            if (unPedido.FechaEntrega < pFecha)
            {
                  listado += unPedido.ToString() + "\r\n";
            }
      }

      return listado;
}
```

Parte 2.e:

En clase Empresa:

```
public BrazoRobotico buscarModeloBrazo(int pCodigoModelo)
{
	foreach (BrazoRobotico unBrazo in this.ColBrazos)
	{
		if (unBrazo.Codigo == pCodigoModelo) return unBrazo;
	}

	return null;
}

public List<Pedido> pedidosSegunModeloYCantidad(int pCodigoModelo,
                                                int pCantidad)
{
	List<Pedido> aux = new List<Pedido>();
	BrazoRobotico unBrazo = this.buscarModeloBrazo(pCodigoModelo);

	if (unBrazo != null)
	{
		foreach (Pedido unPedido in this.ColPedidos)
		{
			if (unPedido.Modelo == unBrazo &&
				unPedido.Cantidad == pCantidad)
			{
				aux.Add(unPedido);
			}
		}
	}

	return aux;
}
```

Parte 2.f:

En clase Empresa:

```
public Pedido buscarPedido(int pNumero)
{
      foreach (Pedido unPedido in this.ColPedidos)
      {
            if (unPedido.Numero == pNumero) return unPedido;
      }

      return null;
}

public decimal totalACobrarSegunPedido(int pNumero)
{
      Pedido unPedido = this.buscarPedido(pNumero);

      if (unPedido != null)
      {
            return unPedido.totalACobrar();
      }

      return 0;
}
```

En clase Pedido:

```
public decimal totalACobrar()
{
      return this.Cantidad * this.Modelo.precioUnitario();
}
```

En clase BrazoRobotico:

```
public abstract decimal precioUnitario();
```

En clase BrazoPintura:

```
public class BrazoPintura : BrazoRobotico
{
      ...
      ...

      public override decimal precioUnitario()
      {
            decimal precio =
                  this.PrecioBase * (decimal)this.SupXMinuto;

            precio += (this.SopleteInter ? precio * 0.10M : 0);

            return precio;

      }
}
```

En clase BrazoSoldadura:

```
public class BrazoSoldadura : BrazoRobotico
{
	...
	...

	public override decimal precioUnitario()
	{
		return this.PrecioBase * (decimal)this.PuntosXMinuto;

	}
}
```

En clase BrazoTransporte:

```
public class BrazoTransporte : BrazoRobotico
{
	...
	...

	public override decimal precioUnitario()
	{
		return (this.PrecioBase * (decimal)this.PesoMax)
							+ this.ValorFicto;
	}
}
```