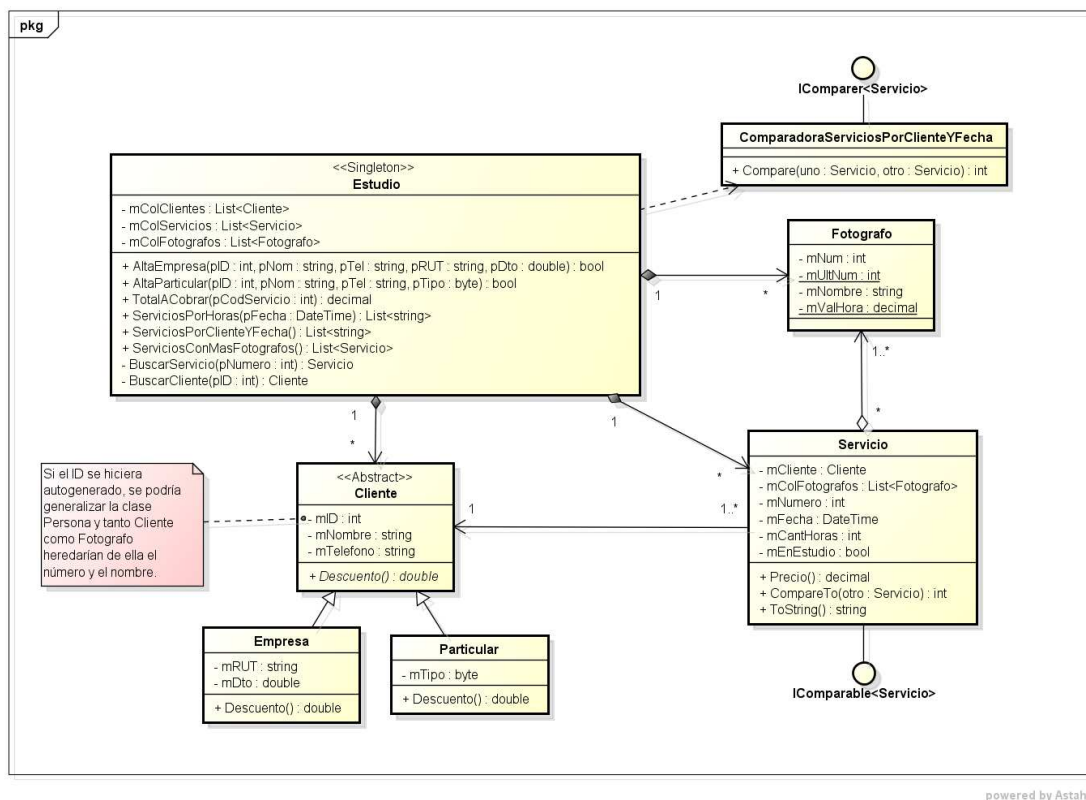


<b>EVALUACION</b>	<b>SOLUCION EXAMEN</b>	<b>GRUPO</b>	<b>TODOS</b>	<b>FECHA</b>	16/05/2014
<b>MATERIA</b>	PROGRAMACIÓN 2				
<b>CARRERA</b>	AP/ATI/APW				
<b>CONDICIONES</b>	<p>- Puntos: 100</p> <p>- Duración: 3 Horas</p> <p>- Sin material</p> <p>- Otros : No escriba la hoja de la letra</p> <p>Consultas solamente sobre interpretación de letra y sintaxis específica del lenguaje.</p> <p>Numerar las hojas entregadas.</p> <p><b>Indicar nombre del docente del curso en primera hoja del examen</b></p>				

## SOLUCIÓN:

### Parte 1



**Parte 2.b**

**EN CLASE ESTUDIO:**

```
public decimal TotalACobrar(int pCodServicio)
{
    Servicio elServicio = this.BuscarServicio(pCodServicio);

    if (elServicio != null)
    {
        return elServicio.Precio();
    }

    return 0;
}

public Servicio BuscarServicio(int pNumero)
{
    Servicio aux = null;
    int pos = 0;

    while (pos < this.mColServicios.Count && aux == null)
    {
        Servicio unS = this.mColServicios[pos];
        if (unS.Numero == pNumero)
        {
            aux = unS;
        }
        pos++;
    }

    return aux;
}
```

**EN CLASE SERVICIO:**

```
public decimal Precio()
{
    decimal precio =
        this.CantHoras * this.ColFotografos.Count * Fotografo.ValHora;
    double dto = this.Cliente.Descuento();

    precio -= precio * (decimal)dto / 100;

    return precio;
}
```

**EN CLASE CLIENTE:**

```
public abstract class Cliente
public abstract double Descuento();
```

**EN CLASE EMPRESA:**

```
public class Empresa : Cliente
public override double Descuento()
{
    return this.Dto;
}
```

---

**EN CLASE PARTICULAR:**

```
public class Particular : Cliente

public override double Descuento()
{
    if (this.Tipo == 1)
    {
        return 10;
    }

    return 0;
}
```

**Parte 2.c**

**EN CLASE ESTUDIO:**

```
public List<string> ServiciosPorHoras(DateTime pFecha)
{
    List<string> listado = new List<string>();

    this.mColServicios.Sort();

    foreach (Servicio unS in this.mColServicios)
    {
        if (unS.Fecha == pFecha)
        {
            listado.Add(unS.ToString());
        }
    }

    return listado;
}
```

**EN CLASE SERVICIO:**

```
public class Servicio : IComparable<Servicio>
public int CompareTo(Servicio otro)
{
    if (otro != null)
    {
        return this.CantHoras - otro.CantHoras;
    }

    return 0;
}
public override string ToString()
{
    return this.Numero + " " + this.Lugar + " " + this.CantHoras;
}
```

---

**Parte 2.d**

**EN CLASE ESTUDIO:**

```
public List<string> ServiciosPorClienteYFecha()
{
    List<string> listado = new List<string>();

    this.mColServicios.Sort(new ComparadoraServiciosPorClienteYFecha());

    foreach (Servicio unS in this.mColServicios)
    {
        listado.Add(unS.Cliente.Nombre + " " +
                    unS.Fecha.ToString("dd/MM/yyyy"));
    }

    return listado;
}
```

**EN CLASE COMPARADORASERVICIOSPORCLIENTEYFECHA:**

```
public class ComparadoraServiciosPorClienteYFecha : IComparer<Servicio> {

    public int Compare(Servicio uno, Servicio otro) {

        if (uno != null && otro != null)
        {
            int compNombres =
                uno.Cliente.Nombre.CompareTo(otro.Cliente.Nombre);
            if (compNombres == 0)
            {
                return uno.Fecha.CompareTo(otro.Fecha);
            }
            else
            {
                return compNombres;
            }
        }

        return 0;
    }
}
```

---

**Parte 2.e:**

**EN CLASE ESTUDIO:**

```
public List<Servicio> ServiciosConMasFotografos()
{
    int maxCantFotografos = 0;
    List<Servicio> aux = new List<Servicio>();

    foreach(Servicio unS in this.mColServicios) {

        int cantFotografos = unS.ColFotografos.Count;

        if (cantFotografos >= maxCantFotografos)
        {
            if (cantFotografos > maxCantFotografos)
            {
                maxCantFotografos = cantFotografos;
                aux.Clear();
            }

            aux.Add(unS);
        }
    }
    return aux;
}
```