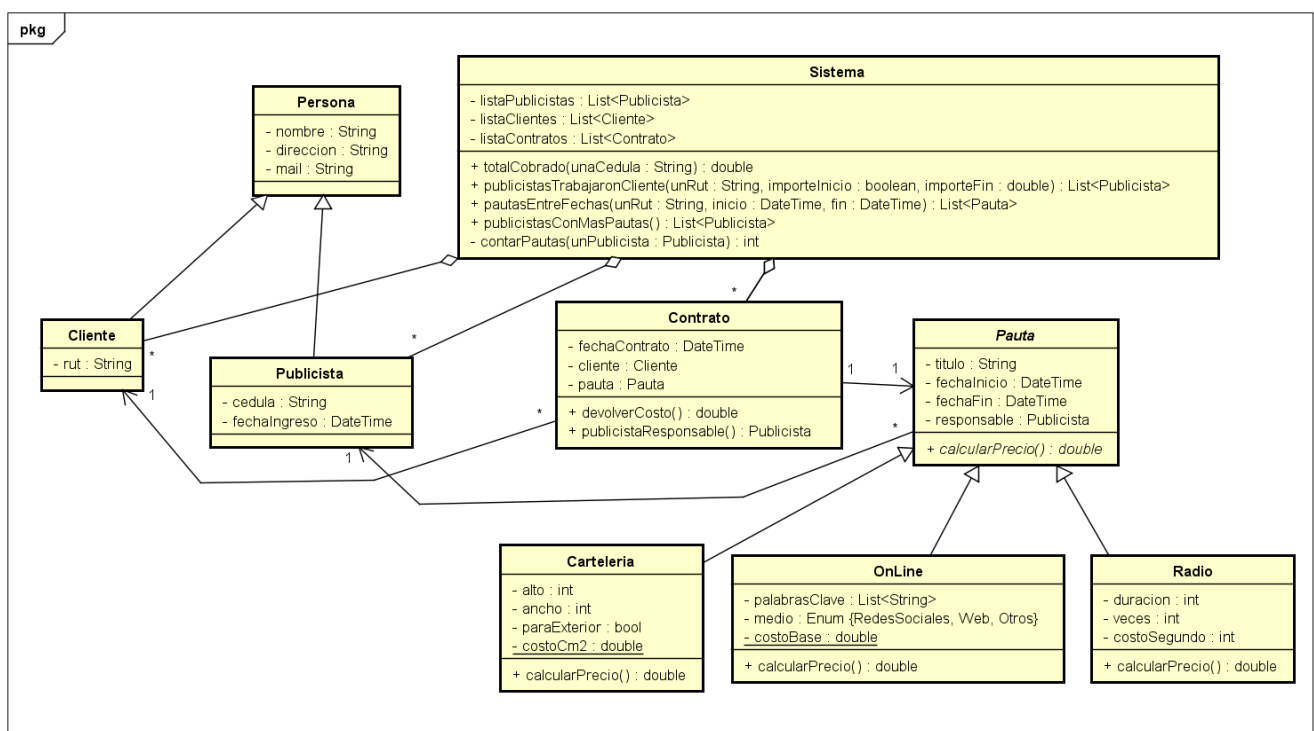


EVALUACION	SOLUCION EXAMEN	FECHA	9/08/16
MATERIA	PROGRAMACIÓN 2		
CARRERA	AP/ATI		
CONDICIONES	<p>- Puntos:100</p> <p>- Duración: 3 horas</p> <p>- Sin material</p> <p>- Consultas: Exclusivamente de interpretación y/o alcance de letra, o sintaxis de alguna función de C#.NET no utilizada regularmente en el curso</p> <p>-Indicar nombre del docente del curso en primera hoja del examen</p>		

Parte A



powered by Astah

Parte B1

```
// En Sistema
public double totalCobrado(string unaCedula) {
    double total = 0;
    foreach (Contrato unContrato in listaContratos) {
        if (unContrato.PulicistaResponsable().Cedula == unaCedula) {
            total += unContrato.devolverCosto();
        }
    }
    return total;
}

// En clase Contrato

public double devolverCosto() {
    return this.Pauta.calcularCosto();
}

public Publicita publicitaResponsable() {
    return this.Pauta.Responsable;
}

// En clase Pauta

    public abstract double calcularCosto();

// En clase Carteleria

public class Carteleria : Pauta
{
    public override double calcularCosto()
    {
        double costo = this.Largo * this.Ancho * Carteleria.CostoCm2;
        if (this.ParaExterior) {
            costo = costo * 1.10;
        }
        return costo;
    }
}

// En clase OnLine

public class Online : Pauta
{
    public override double calcularCosto()
    {
        double costo = Online.CostoBase * this.PalabrasClave.Count;
        if (this.Medio == Online.Medio.WebInstitucional) {
            costo = costo * 0.9;
        }
        return costo;
    }
}
```

// En clase Radio

```
public class Radio : Pauta
{
    public override double calcularCosto()
    {
        double costo = this.CostoSegundo * this.Duracion*this.Veces;

        return costo;
    }
}
```

Parte B2

// En Sistema

```
public List<Publicista> publicistasTrabajaronCliente(string unRut, double importeInicio,
                                                    double importeFin) {
    List<Publicista> resultado = new List<Publicista>();
    foreach (Contrato unContrato in listaContratos)
    {
        double costo = unContrato.devolverCosto();
        string rut = unContrato.Cliente.Rut();
        Publicista elPublicista = this.publicistaResponsable();
        if (rut.Equals(elRut) && costo >= importeInicio && costo <= importeFin)
        {
            if (!retorno.Contains(elPublicista)) {
                retorno.Add(elPublicista);
            }
        }

        return resultado;
    }
}
```

// En Publicista

```
public class Publicista {
    public override bool Equals(object obj)
    {
        Publicista otro = obj as Publicista;
        return this.Nombre.Equals(otro.Nombre);
    }
}
```

Parte B3

// En Sistema

```
public List<Pauta> pautasEntreFechas(string unRut, DateTime inicio, DateTime fin) {  
    List<Pauta> retorno = new List<Pauta>();  
  
    foreach (Contrato unContrato in listaContratos) {  
        string rut = unContrato.Cliente.Rut;  
        Pauta unaPauta = unContrato.Pauta;  
        if (rut.Equals(unRut) && unaPauta.FechaInicio >= inicio &&  
            unaPauta.FechaFin <= fin) {  
            retorno.Add(unaPauta);  
        }  
    }  
    retorno.Sort();  
    return retorno;  
}
```

// En clase Pauta

```
public abstract class Pauta: IComparable<Pauta>  
{  
    public override int CompareTo(Pauta otra) {  
        return otra.Responsable.Cedula.CompareTo(this.Responsable.Cedula)  
    }  
}
```

Parte B4

```
// En clase Sistema

public List<Publicista> publicistaConMasPautas() {
    List<Publicista> retorno = new List<Publicista>();

    int maximo = 0;

    foreach (Publicista unPublicista in listaPublicistas) {
        int cantidad = this.contarPautas(unPublicista);

        if (cantidad > maximo)
        {
            retorno.Clear();
            retorno.Add(unPublicista);
        }
        else {
            if (cantidad == maximo) {
                retorno.Add(unPublicista);
            }
        }
        return retorno;
    }
}

private int contarPautas(Publicista unPublicista)
{
    int veces = 0;
    foreach(Contrato unContrato in listaContratos)
    {
        if (unContrato.publicistaResponsable().Equals(unPublicista)) {
            veces++;
        }
    }
    return veces;
}
```