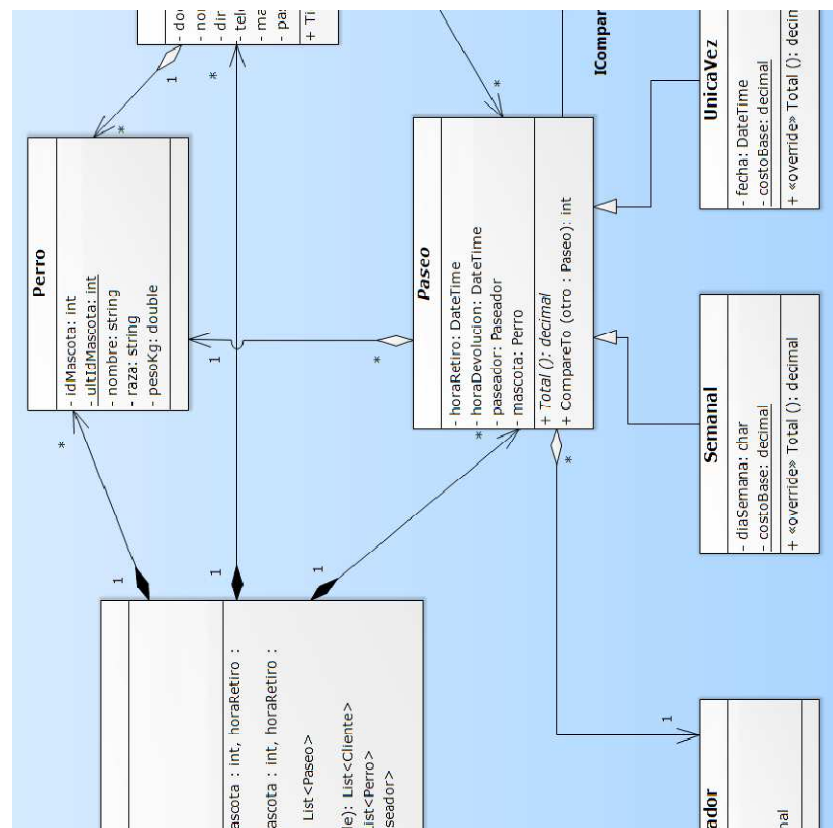


EVALUACION	EXAMEN	FECHA	19/02/2019
MATERIA	PROGRAMACIÓN II		
CARRERA	AP/ATI		
CONDICIONES	<p>- Puntos: 100</p> <p>- Duración: 3 Horas</p> <p>- Sin material</p> <p>- Otros : No escriba la hoja de la letra</p> <p>Consultas solamente sobre interpretación de letra y sintaxis específica no trivial del lenguaje.</p> <p>INDIQUE EL NOMBRE DEL DOCENTE EN LA CARÁTULA</p>		

1. Diagrama de clases UML



2. b.

En Veterinaria:

```

public List<Paseo> PaseosContratadosPorCliente(string docCliente)
{
    Cliente cli = this.BuscarCliente(docCliente);

    List<Paseo> aux = new List<Paseo>();

    if (cli != null)
    {
        aux = new List<Paseo>(cli.PaseosContratados);
    }
}

```

```
        aux.Sort();
    }

    return aux;
}

public Cliente BuscarCliente(string docCliente)
{
    Cliente aux = null;

    for(int n = 0; n < this.Clientes.Count && aux == null; n++)
    {
        if (this.Clientes[n].Documento == docCliente)
        {
            aux = this.Clientes[n];
        }
    }

    return aux;
}
```

En Paseo:

```
public abstract decimal Total();

public int CompareTo(Paseo otro)
{
    int ret = 0;
    if (otro != null)
    {
        ret = this.Total().CompareTo(otro.Total());
    }
    return ret;
}
```

En UnicaVez:

```
public override decimal Total()
{
    decimal total = UnicaVez.CostoBase + this.Paseador.ValorPaseo;

    if (this.Mascota.PesoKg > 20)
    {
        total += total * 0.05M;
    }
}
```

```
        return total;
    }
```

En Semanal:

```
public override decimal Total()
{
    decimal total = Semanal.CostoBase + this.Paseador.ValorPaseo;

    if (this.DiaSemana == 'S' || this.DiaSemana == 'D')
    {
        total += total * 0.1M;
    }

    return total;
}
```

2. c.

En Veterinaria:

```
public List<Cliente> ClientesConPerrosQuePesanMasDe(double pesoKg)
{
    List<Cliente> aux = new List<Cliente>();

    foreach (Cliente cli in this.Clientes)
    {
        if (cli.TienePerrosConPesoMayorA(pesoKg)) aux.Add(cli);
    }

    return aux;
}
```

En Cliente:

```
public bool TienePerrosConPesoMayorA(double pesoKg)
{
    bool ret = false;

    int i = 0;

    while (i < this.Mascotas.Count && !ret)
    {
        if (this.Mascotas[i].PesoKg > pesoKg) ret = true;
    }
}
```

```
        i++;  
    }  
  
    return ret;  
}
```

2. d.

En Veterinaria:

```
public List<Perro> PerrosPaseadosPorPaseador(int nroEmpleado)  
{  
    List<Perro> aux = new List<Perro>();  
  
    foreach (Paseo paseo in this.Paseos)  
    {  
        if (paseo.Paseador.NroEmp == nroEmpleado)  
        {  
            if (!aux.Contains(paseo.Mascota)) aux.Add(paseo.Mascota);  
        }  
    }  
  
    return aux;  
}
```

2. e.

En Veterinaria:

```
public List<Paseador> PaseadoresQueCobranMayorValorPaseo()
{
    List<Paseador> aux = new List<Paseador>();
    decimal mayorValor = 0;

    foreach (Paseador paseador in this.Paseadores)
    {
        if (paseador.ValorPaseo >= mayorValor)
        {
            if (paseador.ValorPaseo > mayorValor)
            {
                aux.Clear();
                mayorValor = paseador.ValorPaseo;
            }

            aux.Add(paseador);
        }
    }

    return aux;
}
```