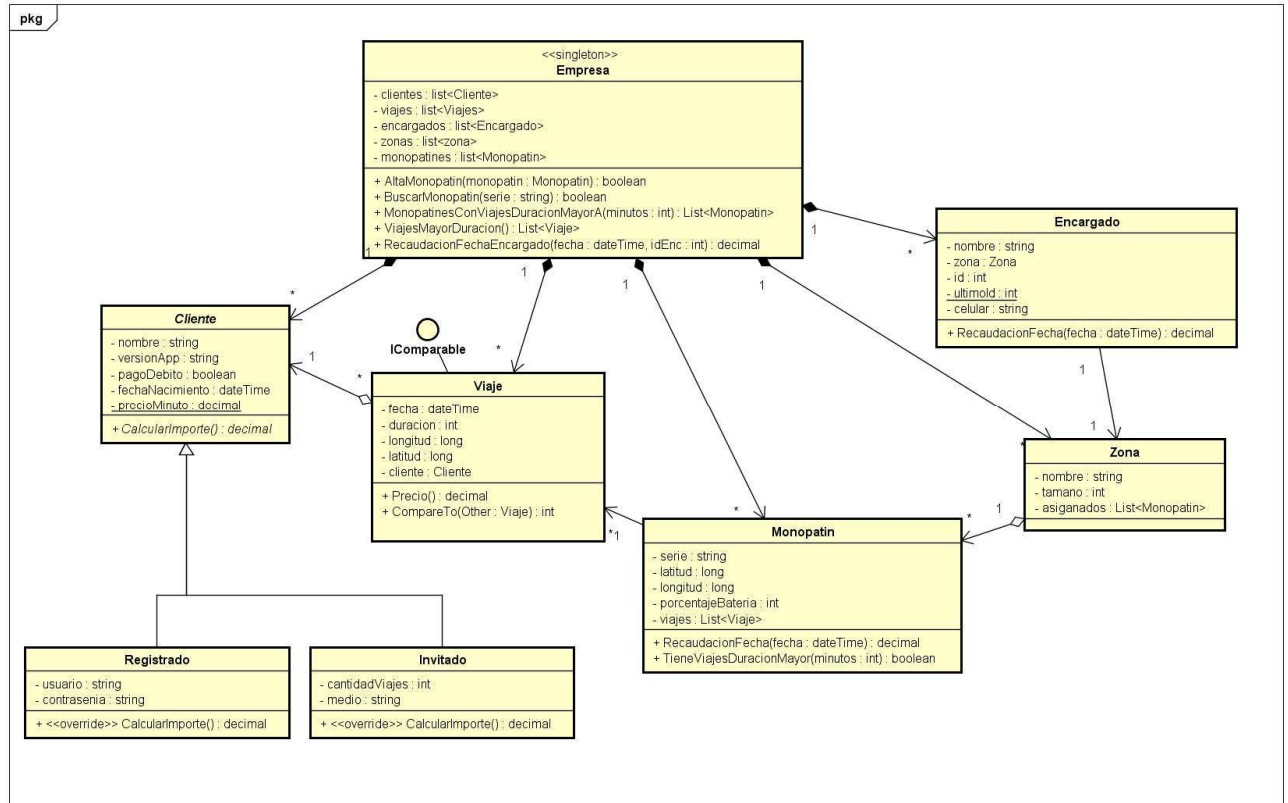


EVALUACION	Solución	GRUPO	TODOS	FECHA	17/05/2019
MATERIA	PROGRAMACIÓN 2				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES					



// Punto b

En sistema

```
// Dada una duración listar todos los monopatinés
//que tuvieron viajes cuya duración sea mayor la indicada.

public List<Monopatin> MonopatinesConViajesDuracionMayorA(int minutos)
{
    List<Monopatin> aux = new List<Monopatin>();
    foreach (Monopatin m in Monopatines)
    {
        if (m.TieneViajesDuracionMayor(minutos))
            aux.Add(m);
    }
    return aux;
}
```

Monopatin

```
public bool TieneViajesDuracionMayor(int minutos)
{
    int i = 0;
    bool tiene = false;
    while (!tiene && i < Viajes.Count)
    {
        if (Viajes[i].Duracion > minutos)
            tiene = true;
        i++;
    }
    return tiene;
}
```

// punto c

En Sistema

//Obtener el/los viajes (sin importar la zona) que duraron más en minutos, ordenados por fecha.

```
public List<Viaje> ViajesMayorDuracion ()
{
    int mayor = 0;
    List<Viaje> aux = new List<Viaje>();
    foreach (Viaje v in Viajes)
    {
        if (v.Duracion > mayor)
        {
            mayor = v.Duracion;
            aux.Clear();
            aux.Add(v);
        }
        else
        {
            if (v.Duracion == mayor)
                aux.Add(v);
        }
    }
    aux.Sort();
    return aux;
}
```

En viaje

```
public int CompareTo(Viaje other)
{
    return Fecha.CompareTo(other.Fecha);
}
```

// punto d

En Sistema

// Dado un encargado y una fecha, indicar el importe recaudado en esa fecha,
// por los viajes realizados en los monopatines que tiene asignados a su zona.

```
public decimal RecaudacionFechaEncargado(int idEnc, DateTime fecha)
{
    decimal total = 0;
    Encargado e = BuscarEncargado(idEnc);
    if (e != null)
    {
        total = e.RecaudacionFecha(fecha);
    }
    return total;
}
```

En Encargado

```
public decimal RecaudacionFecha (DateTime fecha)
{
    decimal total = 0;
    foreach (Monopatin m in zona.Asignados)
    {
        total += m.RecaudacionFecha(fecha);
    }
    return total;
}
```

En Monopatin

```
public decimal RecaudacionFecha (DateTime fecha)
{
    decimal total = 0;
    foreach (Viajes v in viajes)
    {
        If (v.fecha == fecha)
            total += v.Precio();
    }
    return total;
}
```

En Viajes

```
public decimal Precio()
{
    decimal precio = cliente.CalcularImporte() * Duracion;
    return precio;
}
```

Registrado

```
public override decimal CalcularImporte()
{
    decimal importe = PrecioMinuto * (decimal)0.85;

    if (DateTime.Today.Month == FechaNacimiento.Month &&
        DateTime.Today.Day == FechaNacimiento.Day)

        importe = 0;
    return importe;
}
```

Invitado

```
public override decimal CalcularImporte()
{
    decimal importe = PrecioMinuto;
    if (pagoDebito)
    {
        importe *= (decimal)0.95;
    }
    return importe;
}
```