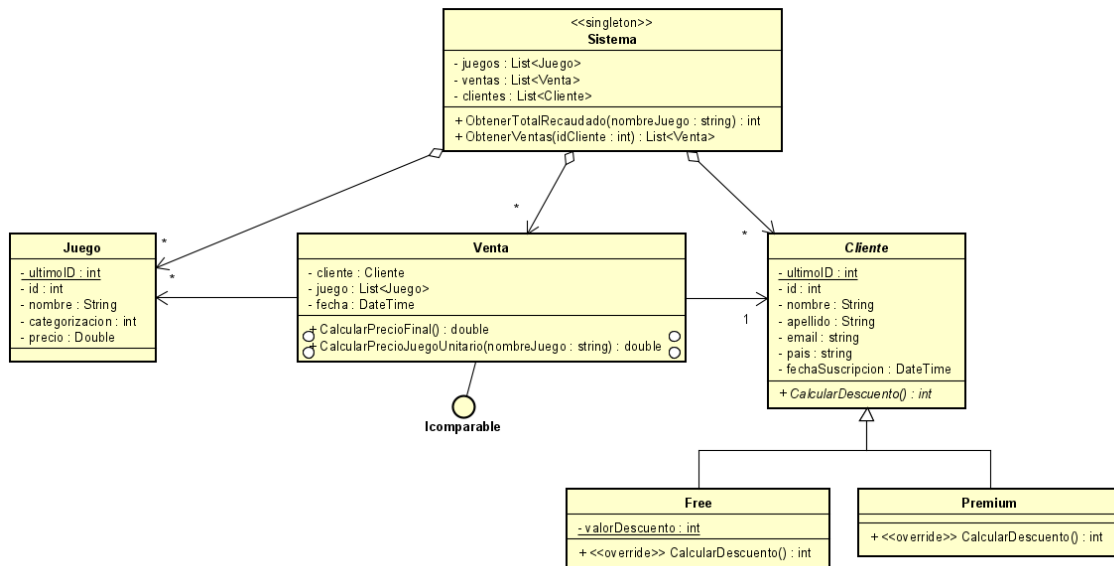


Solución:

UML:



2.

Cliente:

```
public abstract class Cliente
{
    public static int ultimoId { get; set; }
    public int Id { get; set; }
    public string Nombre { get; set; }
    public string Apellido { get; set; }
    public string Email { get; set; }
    public string Pais { get; set; }
    public DateTime FechaSuscripcion { get; set; }

    public abstract int calcularDescuento();
}
```

Free

```
public class Free : Cliente
{
    public static int ValorDescuento { get; set; }
    public override int calcularDescuento()
    {
        int dto = 0;

        if (this.Pais.Equals("Uruguay"))
        {
            dto = ValorDescuento;
        }
        return dto;
    }
}
```

Premium

```
public class Premium : Cliente
{
    public override int calcularDescuento()
    {
        int dto;

        if (this.Pais.Equals("Uruguay"))
        {
            if (this.FechaSuscripcion < DateTime.Parse("2018-01-01"))
            {
                dto = 40;
            }
            else
            {
                dto = 30;
            }
        }
        else
        {
            dto = 25;
        }
        return dto;
    }
}
```

```

public class Venta : IComparable<Venta>
{
    public Cliente Cliente { get; set; }
    public List<Juego> Juegos { get; set; }
    public DateTime Fecha { get; set; }

    public double calcularPrecioFinal() {
        double totalPrecioJuegos = 0;
        foreach (Juego j in Juegos) {
            totalPrecioJuegos += j.Precio;
        }
        int porcentajeDescuento = this.Cliente.calcularDescuento();
        double cantidadDescontar = (totalPrecioJuegos * porcentajeDescuento) / 100;
        double precioFinal = totalPrecioJuegos - cantidadDescontar;
        return precioFinal;
    }

    public double CalcularPrecioJuegoUnitario(string nombreJuego) {
        double ret = 0;

        foreach (Juego j in Juegos) {
            if (j.Nombre.Equals(nombreJuego)) {
                int porcentajeDescuento = this.Cliente.calcularDescuento();
                double cantidadDescontar = (j.Precio * porcentajeDescuento) / 100;
                double precioFinalJuego = j.Precio - cantidadDescontar;
                ret = precioFinalJuego;
            }
        }
        return ret;
    }
}

public int CompareTo(Venta other)
{
    int res;

    if (this.calcularPrecioFinal().CompareTo(other.calcularPrecioFinal()) > 0)
    {
        res = 1;
    }
    else if (this.calcularPrecioFinal().CompareTo(other.calcularPrecioFinal())
    < 0)
    {
        res = -1;
    }
    else
    {
        res = 0;
    }
    return res;
}
}

```

```
public class Sistema
{
    private Sistema() { }

    private static Sistema instancia = null;

    public static Sistema GetInstancia() {
        if (instancia == null) {
            instancia = new Sistema();
        }
        return instancia;
    }

    private List<Juego> juegos = new List<Juego>();
    private List<Venta> ventas = new List<Venta>();
    private List<Cliente> clientes = new List<Cliente>();

    public double ObtenerTotalRecaudado(string nombreJuego) {
        double totalRecaudado = 0;
        Recorremos la lista de ventas y a cada venta consultamos el valor final del juego
        específico que se quiere buscar. Si la venta lo incluye suma su valor y lo
        retorna.
        foreach (Venta v in ventas)
        {
            totalRecaudado += v.CalcularPrecioJuegoUnitario(nombreJuego);
        }
        return totalRecaudado;
    }

    public List<Venta> ObtenerVentas (int idCliente)
    {
        List<Venta> retorno = new List<Venta>();
        foreach (Venta v in ventas)
        {
            if (v.Cliente.Id.Equals(idCliente))
            {
                retorno.Add(v);
            }
        }
        retorno.Sort();
        return retorno;
    }
}
```

3 . MVC

Vista: ObtenerTotal.

```
@using (Html.BeginForm())
{
    <label>Nombre del juego:</label>
    <input type="text" name= "nombreJuego">
    <input type="submit" value= "Buscar">
    @ViewBag.total = "";
}
```

```
Sistema s = Sistema.getInstancia();
[HttpPost]

public ActionResult ObtenerTotal (string nombreJuego)
{
    int totalRecaudado = s.ObtenerTotalRecaudado(nombreJuego);

    @ViewBag.total = totalRecaudado;

    return View();
}
```