SEGUNDO EJERCICIO EVALUABLE - ISW - GRUPO D - CURSO 2017-18

Dado el diagrama de clases de la figura correspondiente al modelado del sistema de renovación de asignaturas optativas se pide:

a. *(4 puntos)* Escriba el código C# para definir cada una de las clases del modelo. Únicamente los atributos necesarios para implementar las relaciones y el constructor de cada clase.

b. *(6 puntos)* Obtener el diagrama de secuencia del siguiente escenario.

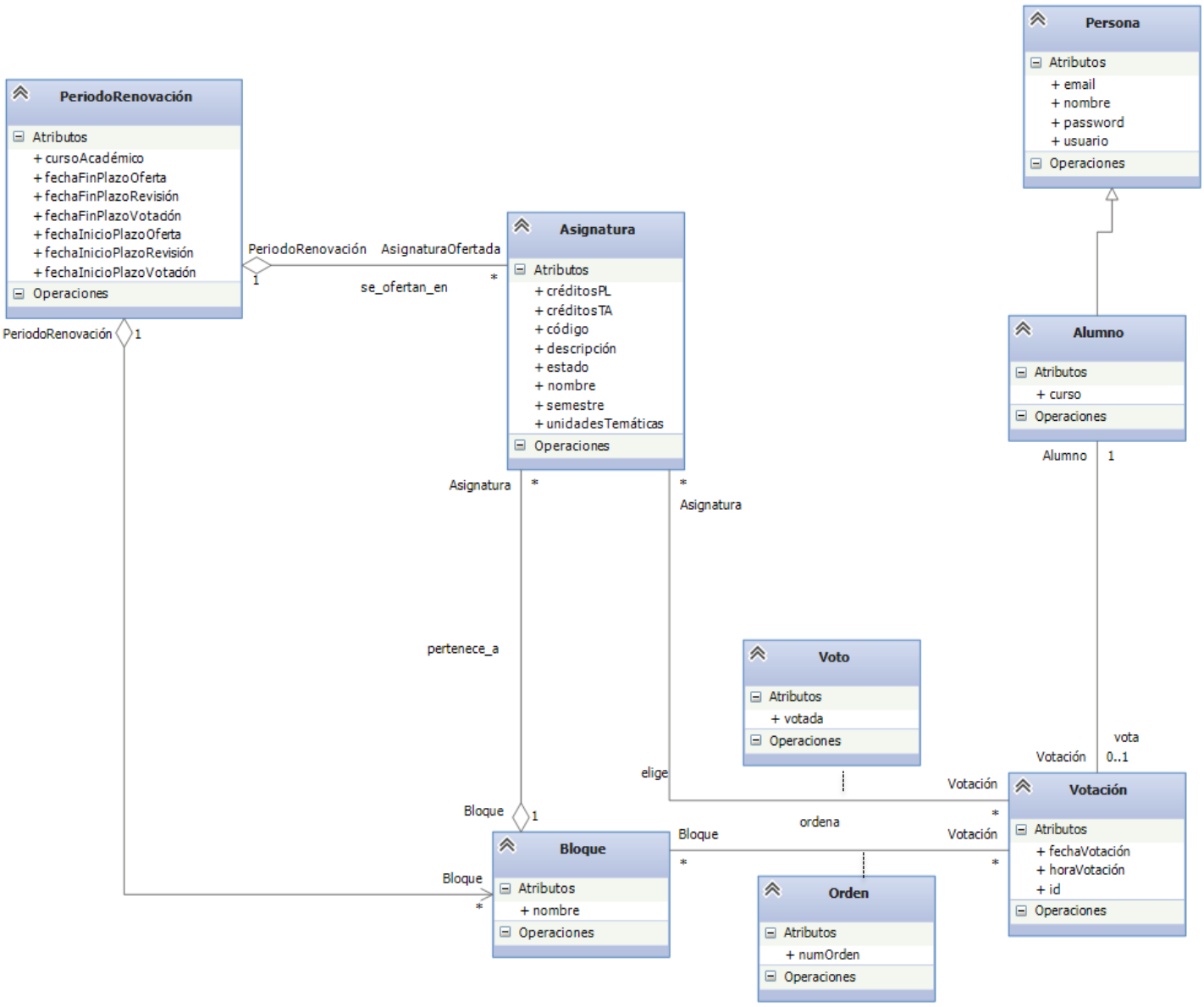
El sistema permitirá al presidente de la Comisión Académica generar el resultado de las votaciones realizadas por los alumnos para el periodo de renovación actual. El resultado será un documento que contendrá la siguiente información:

- Para cada bloque de especialización, el número de veces que ha elegido por un alumno como preferencia primera (numOrden).
- Para cada asignatura que ha participado en la votación (estado="aceptada") el número de votos recibidos.
- El listado de asignaturas se agruparán por los bloques de especialización.

Ejemplo:

Periodo de renovación de asignaturas optativas: 2017-18		
BLOQUE	ASIGNATURAS	VOTOS
Bloque Ciberseguridad (295 en primera opción)	Hacking Ético	487 votos
	Seguridad Web	473 votos
Bloque Videojuegos (196 en primera opción)	Entornos de desarrollo de videojuegos	452
	Animation and design of videogames	425
Bloque Multimedia (97 en primera opción)	Sistemas multimedia interactivos	422
	Redes multimedia	384
Bloque Inf. Industrial (49 en primera opción)	Diseño y Fabricación 3D	380
	Impresión 3D	312
Bloque Salud (42 en primera opción)	Gestión de la innovación y tec. en salud	470
	Diseño y gestión de SI genómicos	434

c. *(OPCIONAL)* Escriba el código que inicialice el sistema en un estado correcto y consistente, creando al menos una instancia de cada clase. Puede utilizar los valores de atributos que desee.


Persona
Atributos
+ email
+ nombre
+ password
+ usuario
Operaciones
PeriodoRenovación
Atributos
+ cursoAcadémico
+ fechaFinPlazoOferta
+ fechaFinPlazoRevisión
+ fechaFinPlazoVotación
+ fechaInicioPlazoOferta
+ fechaInicioPlazoRevisión
+ fechaInicioPlazoVotación
Operaciones
PeriodoRenovación
1
PeriodoRenovación
AsignaturaOfertada
1
*
se_ofertan_en
Asignatura
Atributos
+ créditosPL
+ créditosTA
+ código
+ descripción
+ estado
+ nombre
+ semestre
+ unidadesTemáticas
Operaciones
Alumno
Atributos
+ curso
Operaciones
Alumno
1
Asignatura
*
*
Asignatura
pertenece_a
Voto
Atributos
+ votada
Operaciones
vota
Votación
0..1
elige
Votación
*
ordena
Votación
Bloque
1
Bloque
Atributos
+ nombre
Operaciones
Bloque
*
*
Orden
Atributos
+ numOrden
Operaciones
Votación
Atributos
+ fechaVotación
+ horaVotación
+ id
Operaciones

a. *(4 puntos)* Escriba el código C# para definir cada una de las clases del modelo. Únicamente los atributos necesarios para implementar las relaciones y el constructor de cada clase.

A continuación se muestra el código completo: definición de atributos e implementación de constructores. En rojo aparece resaltado lo que se pedía en el ejercicio.

```
public class Persona
{
   public Persona(string nombre, string email, string password, string usuario)
   {
      this.nombre = nombre;
      this.email = email;
      this.password = password;
      this.usuario = usuario;
   }
   public virtual string nombre
  {
        get;
        set;
  }

  public virtual string usuario
  {
        get;
        set;
  }

  public virtual string password
  {
        get;
        set;
  }

  public virtual string email
  {
        get;
        set;
  }

}
```

```
public class Alumno : Persona
{
   public Alumno(string nombre, string email, string password, string usuario, string curso): base(nombre, email, password, usuario)
   {
      this.curso = curso;
   }
   public virtual string curso
  {
        get;
        set;
  }

  public virtual Votación Votación
  {
        get;
        set;
  }

}
```

```
public class Orden
{
   public Orden(int numOrden, Bloque b, Votación v)
   {
      this.numOrden = numOrden;
      this.bloque = b;
      this.votación = v;
   }
  public virtual int numOrden
  {
        get;
        set;
  }
   public virtual Bloque bloque
   {
      get;
      set;
   }

   public virtual Votación votación
   {
      get;
      set;
   }
}
```

```
public class Votación
{
  public Votación(int id, DateTime fecha, DateTime hora, Alumno a)
   {
      LOrdenes = new List<Orden>();
      LVotos = new List<Voto>();
      this.id = id;
      this.fechaVotación = fecha;
      this.horaVotación = hora;
      this.Alumno = a;
   }
   public virtual int id
  {
        get;
        set;
  }

  public virtual DateTime horaVotación
  {
        get;
        set;
  }

  public virtual DateTime fechaVotación
  {
        get;
        set;
  }

  public virtual ICollection<Orden> LOrdenes
  {
        get;
        set;
  }

  public virtual ICollection<Voto> LVotos
  {
        get;
        set;
  }

  public virtual Alumno Alumno
  {
        get;
        set;
  }

}
```

```
public class Voto
{
   public Voto(bool votada, Asignatura a, Votación v)
   {
      this.votada = votada;
      this.asignatura = a;
      this.votación = v;
   }
   public virtual bool votada
  {
        get;
        set;
  }
   public virtual Asignatura asignatura
   {
      get;
      set;
   }

   public virtual Votación votación
   {
      get;
      set;
   }
}
```

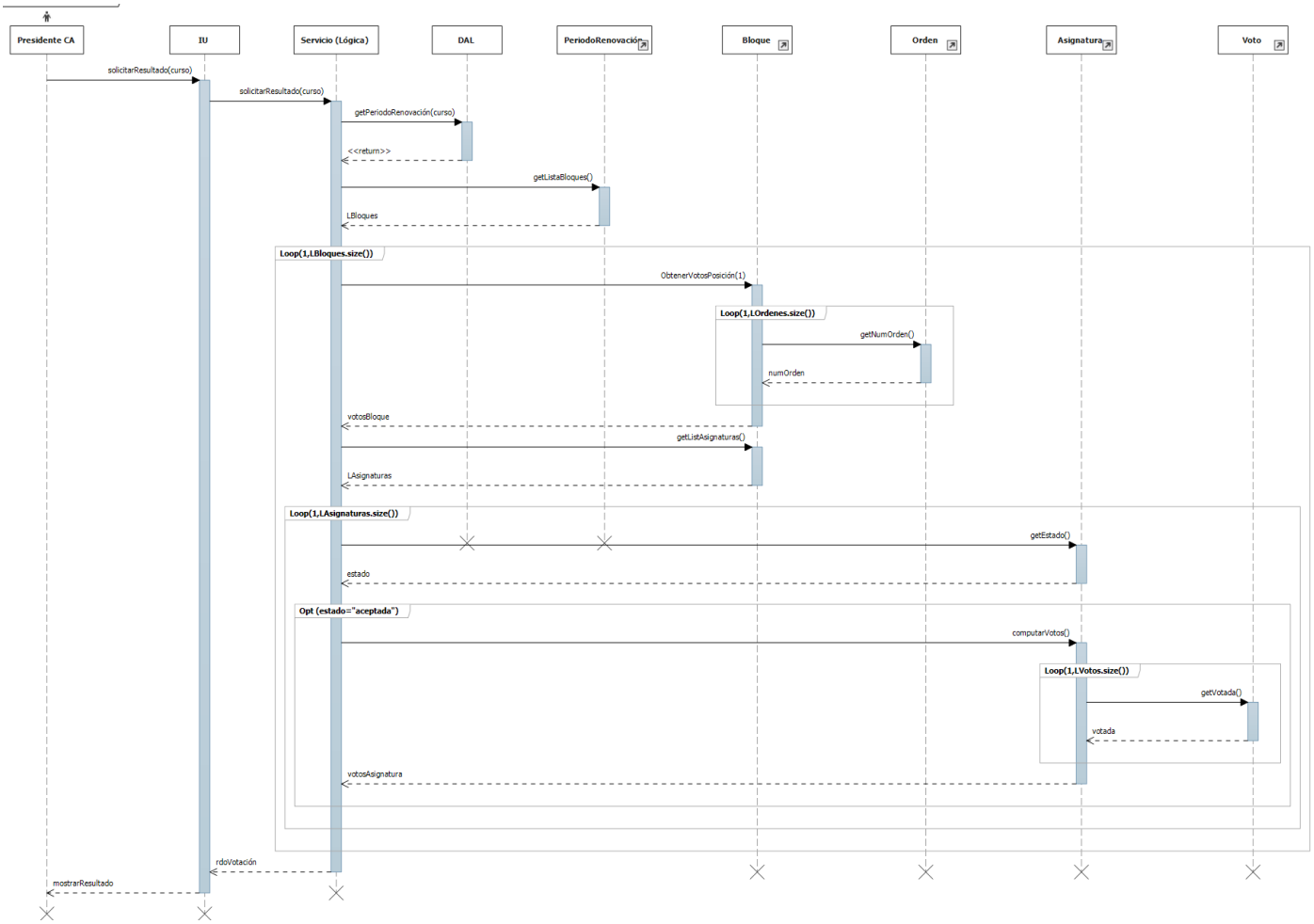
```
public class Bloque
{

// La relación con PeriodoRenovación es unidireccional, por ello Bloque no necesita declarar un atributo de tipo PeriodoRenovación

   public Bloque(string nombre)
   {
      this.LOrdenes = new List<Orden>();
      this.LAsignaturas = new List<Asignatura>();
      this.nombre = nombre;
   }
  public virtual string nombre
  {
        get;
        set;
  }

  public virtual ICollection<Asignatura> LAsignaturas
  {
        get;
        set;
  }

   public virtual ICollection<Orden> LOrdenes
  {
        get;
        set;
  }
```

```
}
----------------------------------------


public class Asignatura
{
   public Asignatura(float crPL, float crTA, int cód, string descripción, string estado, string nombre, string semestre, string unidades, PeriodoRenovación p, Bloque b)
   {
      this.LVotos = new List<Voto>();

      this.créditosPL = crPL;
      this.créditosTA = crTA;
      this.código = cód;
      this.descripción=descripción;
      this.estado = estado;
      this.nombre = nombre;
      this.semestre = semestre;
      this.unidadesTemáticas = unidades;
      this.PeriodoRenovación = p;
      this.Bloque = b;

   }
  public virtual int código
  {
        get;
        set;
  }

  public virtual string nombre
  {
        get;
        set;
  }

  public virtual int créditosTA
  {
        get;
        set;
  }

  public virtual int créditosPL
  {
        get;
        set;
  }

  public virtual string descripción
  {
        get;
        set;
  }

  public virtual string unidadesTemáticas
  {
        get;
```

```
        set;
    }

    public virtual string semestre
    {
        get;
        set;
    }

    public virtual string estado
    {
        get;
        set;
    }

    public virtual PeriodoRenovación PeriodoRenovación
    {
        get;
        set;
    }

    public virtual Bloque Bloque
    {
        get;
        set;
    }

    public virtual ICollection<Voto> LVotos
    {
        get;
        set;
    }
}
```

```
public class PeriodoRenovación
{
    public PeriodoRenovación(string curso, DateTime fechaFinPlazoOferta, DateTime fechaFinPlazoRevisión, DateTime fechaFinPlazoVotación, DateTime fechaInicioPlazoOferta, DateTime fechaInicioPlazoRevisión, DateTime fechaInicioPlazoVotación)
    {
        LBloques = new List<Bloque> ();
        this.cursoAcadémico = curso;
        this.fechaFinPlazoOferta = fechaFinPlazoOferta;
        this.fechaFinPlazoRevisión = fechaFinPlazoRevisión;
        this.fechaFinPlazoVotación = fechaFinPlazoVotación;
        this.fechaInicioPlazoOferta = fechaInicioPlazoOferta;
        this.fechaInicioPlazoRevisión = fechaInicioPlazoRevisión;
        this.fechaInicioPlazoVotación = fechaInicioPlazoVotación;

    }
```

```
    public virtual string cursoAcadémico
    {
        get;
        set;
    }

    public virtual DateTime fechaInicioPlazoOferta
    {
        get;
        set;
    }

    public virtual DateTime fechaFinPlazoOferta
    {
        get;
        set;
    }

    public virtual DateTime fechaInicioPlazoRevisión
    {
        get;
        set;
    }

    public virtual DateTime fechaFinPlazoRevisión
    {
        get;
        set;
    }

    public virtual DateTime fechaInicioPlazoVotación
    {
        get;
        set;
    }

    public virtual DateTime fechaFinPlazoVotación
    {
        get;
        set;
    }

    public virtual ICollection<Asignatura> AsignaturaOfertada
    {
        get;
        set;
    }

    public virtual ICollection<Bloque> LBloques
    {
        get;
        set;
    }

}
----------------------------------------
```


Presidente CA
IU
Servicio (Lógica)
DAL
PeriodoRenovación
Bloque
Orden
Asignatura
Voto
solicitarResultado(curso)
solicitarResultado(curso)
getPeriodoRenovación(curso)
<<return>>
getListaBloques()
LBloques
Loop(1,LBloques.size())
ObtenerVotosPosición(1)
Loop(1,LOrdenes.size())
getNumOrden()
numOrden
votosBloque
getListAsignaturas()
LAsignaturas
Loop(1,LAsignaturas.size())
getEstado()
estado
Opt (estado="aceptada")
computarVotos()
Loop(1,LVotos.size())
getVotada()
votada
votosAsignatura
rdoVotación
mostrarResultado

c. (OPCIONAL) Escriba el código que inicialice el sistema en un estado correcto y consistente, creando al menos una instancia de cada clase. Puede utilizar los valores de atributos que desee.

Suponemos que las clases tienen implementados los métodos para gestionar las colecciones, en concreto los métodos *AddObjeto (Objeto obj)* que añaden un objeto obj a una colección.

Deben inicializarse los objetos en el orden correcto y deben mantenerse las relaciones entre clases mediante la asignación de objetos en el constructor y/o mediante la inserción de objetos en las colecciones.

```
class Program
  {
    public void Main()
    {
      PeriodoRenovación p = new PeriodoRenovación("2017-18", new DateTime(2017, 10, 10), new DateTime(2017, 10, 20), new DateTime(2017, 11, 15), new DateTime(2017, 10, 1), new DateTime(2017, 10, 11), new DateTime(2017, 11, 10));

      Alumno a = new Alumno("Luis Irles", "lirles@gmail.com", "luisito", "lirles", "1");

      Votación v = new Votación(1, new DateTime(2017, 11, 15), new DateTime(2017, 11, 15, 12, 00, 00), a);

      Bloque b = new Bloque("Ciberseguridad");
      p.AddBloque(b); // relación Bloque - PeriodoRenovación, se añade el Bloque b a la colección de bloques de un PeriodoRenovación p

      Asignatura asig = new Asignatura(1.5f, 3.0f, 3561, "Esta asignatura mola mucho", "aceptada", "Ciberguridad ++", "A", "Tema 1, Tema 2, Tema 3", p, b);
      b.AddAsignatura(asig); // relación Bloque - Asignatura, se añade la Asignatura asig a la colección de asignaturas de un Bloque b

      Orden o = new Orden(1, b, v);
      b.AddOrden(o); // relación Bloque-Orden-Votación. Orden es clase asociación. Se añade el Orden o a la colección de Ordenes de un Bloque b
      v.AddOrden(o); // relación Bloque-Orden-Votación. Orden es clase asociación. Se añade el Orden o a la colección de Ordenes de una Votación v

      Voto voto = new Voto(true, asig, v);
      asig.AddVoto(voto); // relación Asignatura-Voto-Votación. Voto es clase asociación. Se añade el Voto o a la colección de Votos de una Asignatura asig
      v.AddVoto(voto); // relación  Asignatura-Voto-Votación. Voto es clase asociación. Se añade el Voto o a la colección de Votos de una Votación v

    }
  }
```