

## Ejercicio 1

Se desea modelar la gestión de unas competiciones automovilísticas. Una competición tiene un nombre y un periodo de celebración. Además, consta de unas carreras que tienen un nombre, una fecha de celebración y la ciudad donde se realiza. Cada carrera se compone de tramos (al menos uno) de los que se tiene un identificador, el número de kilómetros que consta y cuál ha sido el record de tiempo en recorrerlo.

Se dispone de una aplicación Web donde el propietario de un automóvil puede inscribirlo en una carrera, pudiendo inscribir tantos vehículos como quiera. Para inscribirlo, el propietario debe indicar la configuración del vehículo (posición y tamaño del alerón trasero, tipo y tamaño de los neumáticos) y su potencia máxima. Si no existe en el sistema, también puede dar de alta el vehículo, indicando su matrícula, cilindrada, potencia y antigüedad. El propietario tiene un DNI, nombre, tarjeta de crédito y teléfono de contacto.

La organización cuenta con jueces que se encargan de verificar los tramos y de validar las inscripciones de los vehículos. Un juez tiene un DNI, nombre, fecha de alta y salario. Además, debemos conocer las especialidades de cada juez, ya que sólo los que disponen de la especialidad “Topográfica” pueden verificar los tramos. La organización necesita saber qué juez ha verificado un tramo o ha validado una inscripción.

La organización maneja un registro de pilotos siendo un empleado de la organización el encargado de asignar un piloto a un automóvil inscrito. Para ello, se busca al piloto y se realiza la asignación. Si el piloto no existe, se le da de alta proporcionando su DNI, nombre, email y años de experiencia.

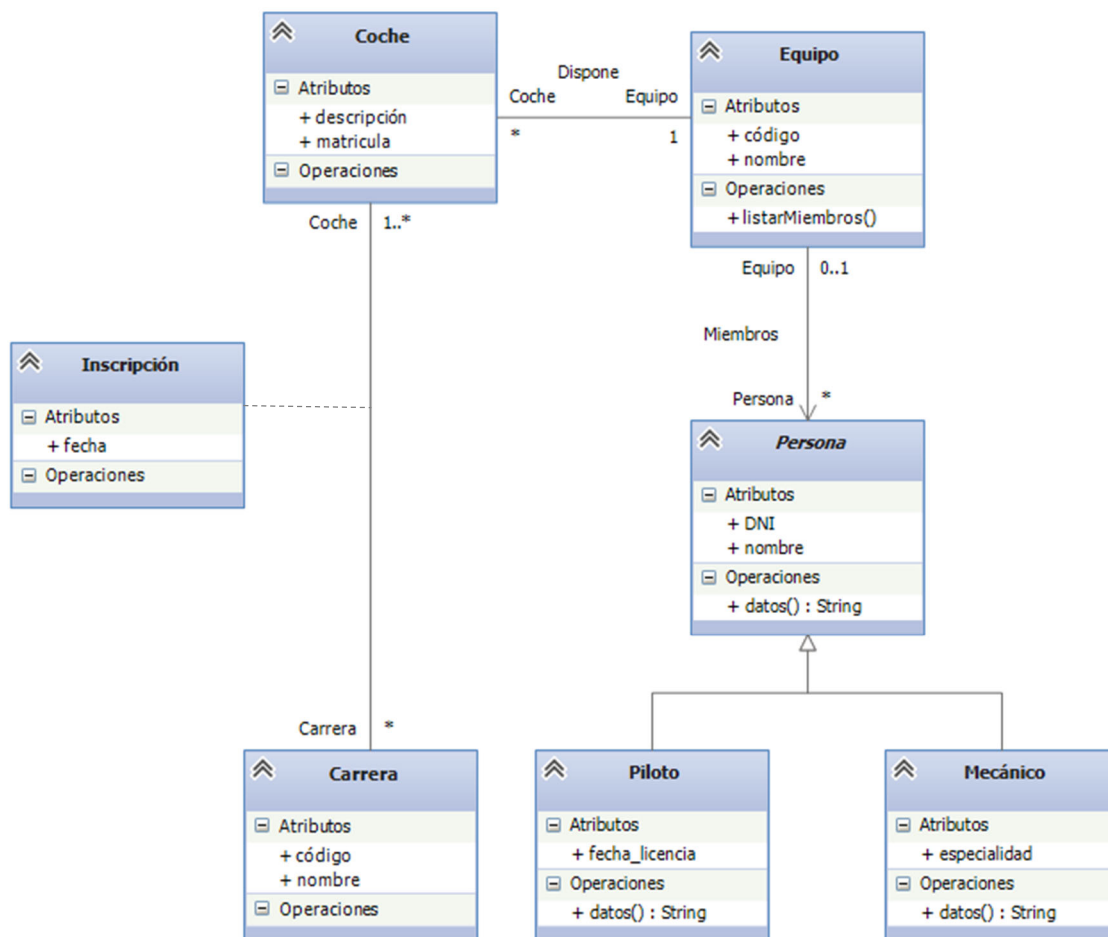
Tras cada carrera, el empleado registra el resultado de cada vehículo inscrito en la carrera anotando la posición, tiempo total de cada uno y enviando un mensaje al piloto indicándole la posición conseguida, siendo necesario que el piloto confirme su recepción. Al final del día en el que se ha celebrado una carrera, se enviará de forma automática un informe a cada propietario indicando el resultado de sus vehículos.

A partir de la descripción anterior, se pide:

- a) (5 puntos) Construir el diagrama de clases en UML, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de las relaciones que detectéis (no es necesario incluir los métodos ni los tipos de los atributos ni su visibilidad).

## Ejercicio 2

Dado el siguiente diagrama de clases en UML, realizad el diseño de clases en C#.

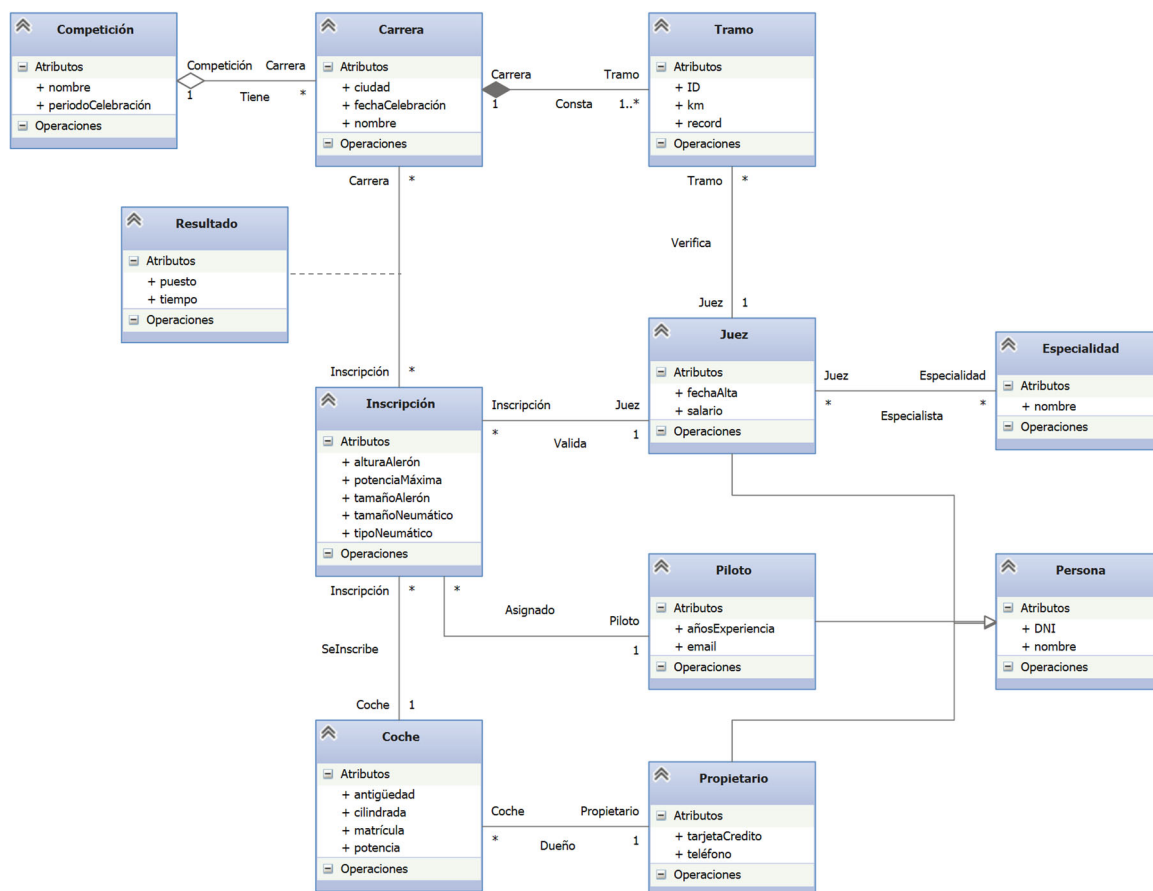


Se pide:

- Diseño de todas las clases, con sus atributos.
- Diseño de los constructores.

NOTA: La clase persona es abstracta.

## Solución ejercicio 1



## Solución ejercicio 2

```
public class Carrera
{
    private int código;
    private String nombre;
    private ICollection<Inscripción> inscripciones;

    public Carrera(int código, String nombre, Inscripción inscripción)
    {
        this.código = código;
        this.nombre = nombre;
        this.inscripciones = new List<Inscripción>();
        this.inscripciones.Add(inscripción);
    }
}

public class Coche
{
    private String descripción;
    private String matrícula;

    private ICollection<Inscripción> inscripciones;
    private Equipo dispone;

    public Coche(String descripción, String matrícula, Equipo equipo)
    {
        this.descripcion = descripción;
        this.matrícula = matrícula;
        this.dispone = equipo;
        this.inscripciones = new List<Inscripción>();
    }
}

public class Inscripción
{
    private DateTime fecha;

    private Coche coche;
    private Carrera carrera;

    public Inscripción(DateTime fecha, Coche coche, Carrera carrera)
    {
        this.fecha = fecha;
        this.coche = coche;
        this.carrera = carrera;
    }
}
```



```
public class Equipo
{
    private int código;
    private String nombre;

    private ICollection<Coche> disponibles;
    private ICollection<Persona> miembros;

    public Equipo(int código, String nombre)
    {
        this.código = código;
        this.nombre = nombre;
        this.disponibles = new List<Coche>();
        this.miembros = new List<Persona>();
    }
}

public abstract class Persona
{
    private String dni;
    private String nombre;

private Equipo equipo; //No contiene el equipo por la restricción de
                        //navegación Equipo → Persona

    public Persona(String dni, String nombre)
    {
        this.dni = dni;
        this.nombre = nombre;
    }
}

public class Piloto:Persona
{
    DateTime fecha_licencia;

    public Piloto(string dni, string nombre, DateTime fecha_licencia) :
        base(dni, nombre)
    {
        this.fecha_licencia = fecha_licencia;
    }
}

public class Mecanico:Persona
{
    private String especialidad;

    public Mecanico(string dni, string nombre, String especialidad) : base(dni,
        nombre)
    {
        this.especialidad = especialidad;
    }
}
```