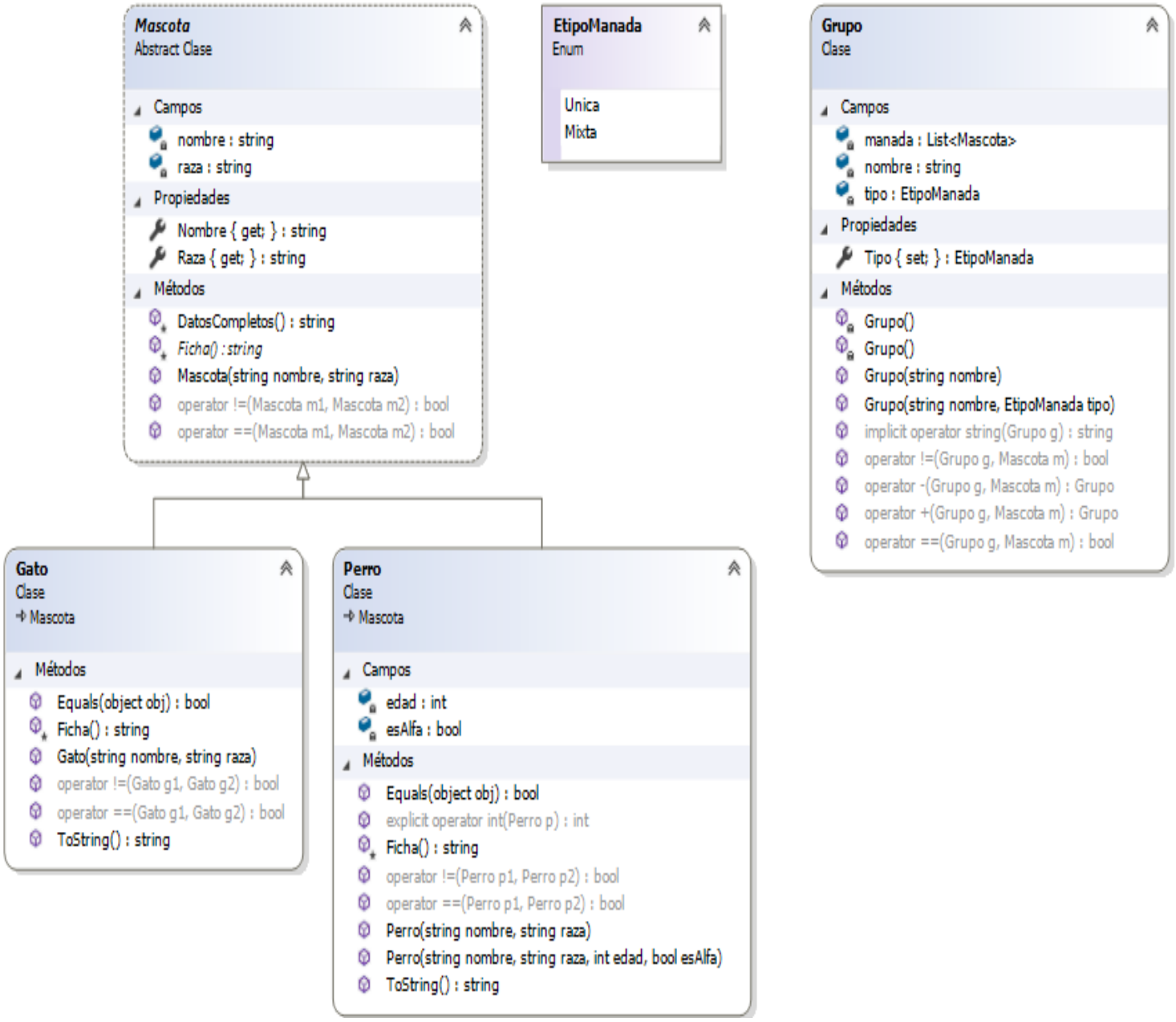
En todos los casos que sean posibles, reutilizar código.
Se tendrá en cuenta las convenciones aplicadas al proyecto. Las entidades estarán en un proyecto de tipo biblioteca de clases.

DIAGRAMA DE CLASES:



Mascota:
Clase pública con dos atributos privados (nombre y raza). El único constructor recibirá dos parámetros.
Las propiedades Nombre y Raza serán de sólo lectura.
Método abstracto y protegido Ficha() .
Método virtual y protegido DatosCompletos() . Retornará el nombre y el raza con el formato "Nombre - Raza".
Sobrecarga del operador == (igualdad), dos mascotas serán iguales si comparten nombre y raza.

Perro:
Clase pública que hereda de Mascota con dos atributos propios privados (edad y esAlfa). Contará con un constructor con dos sobrecargas, en caso de no recibir edad ni esAlfa, deberán inicializarlos como 0 (cero) y false respectivamente. Reutilizar código.
Implementar el método Ficha(). Retornará toda la información del Perro con el siguiente formato:

- Si es alfa: 'perro - Ramón - Salchicha, alfa de la manada, edad 2'
- Si no lo es: 'perro - Julio - Cruza, edad 13'

Sobrecarga de Operadores:

- Dos **Perros** serán iguales si comparten nombre, raza y edad. Reutilizar código.
- Conversión explicita de Perro a entero, retornando su edad.

Sobreescribir:

- Método ToString() para que publique la información del Perro. Reutilizar código.
- Método Equals() para que pueda ser comparado con cualquier objeto. Reutilizar código.

Gato:
Clase pública que hereda de Mascota.
Implementar el método Ficha(). Retornará toda la información del Gato con un formato similar al de Perro.
Sobrecarga de Operadores:

- Dos **Gatos** serán iguales si comparten nombre y raza. Reutilizar código.

Sobreescribir:

- Método ToString() para que publique la información del Gato. Reutilizar código.
- Método Equals para que pueda ser comparado con cualquier objeto. Reutilizar código.

Grupo:
Clase pública que contendrá una lista de Mascotas, un nombre y atributo de **clase** tipo (Enumerado TipoManada contendrá Única, Mixta).
Constructores:

- *De clase*: inicializará el tipo como Única.
- Por defecto privado, será el único lugar donde se inicialice la lista.
- El que recibe un parámetro inicializará el nombre.
- La última sobrecarga recibirá nombre y tipo.
- Reutilizar código.

Una sola propiedad de sólo **escritura** para el tipo.
Sobrecarga de Operadores:

- Un Grupo será igual a una Mascota si esta última forma parte de la lista.
- Si una mascota **no** forma parte de la lista, se podrá agregar con el **+**. Informar caso contrario.
- Si una mascota forma parte de la lista, se podrá quitar con el **-**. Informar caso contrario.
- Conversión implícita a String, debiendo quedar la información con el siguiente formato:

Grupo: Río – tipo: Mixta
Integrantes (6):
perro - Moro - Pitbull, edad 0
perro - Julio - Cruza, edad 13
perro - Ramón - Salchicha, alfa de la manada, edad 2
gato - José - Angora
gato - Mauri - Cruza
gato - Fer – Siamés

Teniendo el siguiente código en el método Main, se debería tener la salida de pantalla cómo lo muestra la siguiente figura.

```
Perro perroUno = new Perro("Moro", "Pitbull");
Perro perroDos = new Perro("Julio", "Cruza", 13, false);
Perro perroTres = new Perro("Ramón", "Salchicha", 2, true);
Perro perroCuatro = new Perro("José", "Angora", 2, false);
Perro perroCinco = new Perro("Ramón", "Salchicha", 2, false);

Gato gatoUno = new Gato("José", "Angora");
Gato gatoDos = new Gato("Mauri", "Cruza");
Gato gatoTres = new Gato("Fer", "Siamés");
Gato gatoCuatro = new Gato("Fer", "Siamés");

Console.WriteLine();

Grupo grupoUno = new Grupo("Río", EtipoManada.Mixta);

grupoUno += perroUno;
grupoUno += perroDos;
grupoUno += perroTres;
grupoUno += perroCuatro;
//repetidos
grupoUno += perroCinco;
grupoUno += perroUno;

grupoUno += gatoUno;
grupoUno += gatoDos;
grupoUno += gatoTres;
//repetido
grupoUno += gatoCuatro;

Console.WriteLine();

//Cantidad 7 (4 perros - 3 gatos)
Console.WriteLine(grupoUno);

grupoUno -= perroUno;
grupoUno -= perroTres;
grupoUno -= gatoTres;

//no están
grupoUno -= perroCinco;
grupoUno -= gatoTres;
grupoUno -= gatoCuatro;

Console.WriteLine();

//Cantidad 4 (2 perros - 2 gatos)
Console.WriteLine(grupoUno);

//son distintos
if (perroUno.Equals("perroUno"))
    Console.WriteLine("Son la misma mascota");
else
    Console.WriteLine("No son la misma mascota");
//son iguales
if (perroTres.Equals(perroCinco))
    Console.WriteLine("Son la misma mascota");
else
    Console.WriteLine("No son la misma mascota");

Console.ReadLine();
```

```
Ya está 'perro - Ramón - Salchicha, edad 2' en el grupo.
Ya está 'perro - Moro - Pitbull, edad 0' en el grupo.
Ya está 'gato - Fer - Siamés' en el grupo.

Grupo: Río - tipo: Mixta
Integrantes (7):
perro - Moro - Pitbull, edad 0
perro - Julio - Cruza, edad 13
perro - Ramón - Salchicha, alfa de la mananada, edad 2
perro - José - Angora, edad 2
gato - José - Angora
gato - Mauri - Cruza
gato - Fer - Siamés

No está el perro - Ramón - Salchicha, edad 2 en el grupo.
No está el gato - Fer - Siamés en el grupo.
No está el gato - Fer - Siamés en el grupo.

Grupo: Río - tipo: Mixta
Integrantes (4):
perro - Julio - Cruza, edad 13
perro - José - Angora, edad 2
gato - José - Angora
gato - Mauri - Cruza

No son la misma mascota
Son la misma mascota
```