


IMPORTANTE:

Generar un proyecto con el nombre: **apellido.nombre.división**. Por ejemplo Pérez.Juan.2A.

Los proyectos que no sean identificables, no serán corregidos.

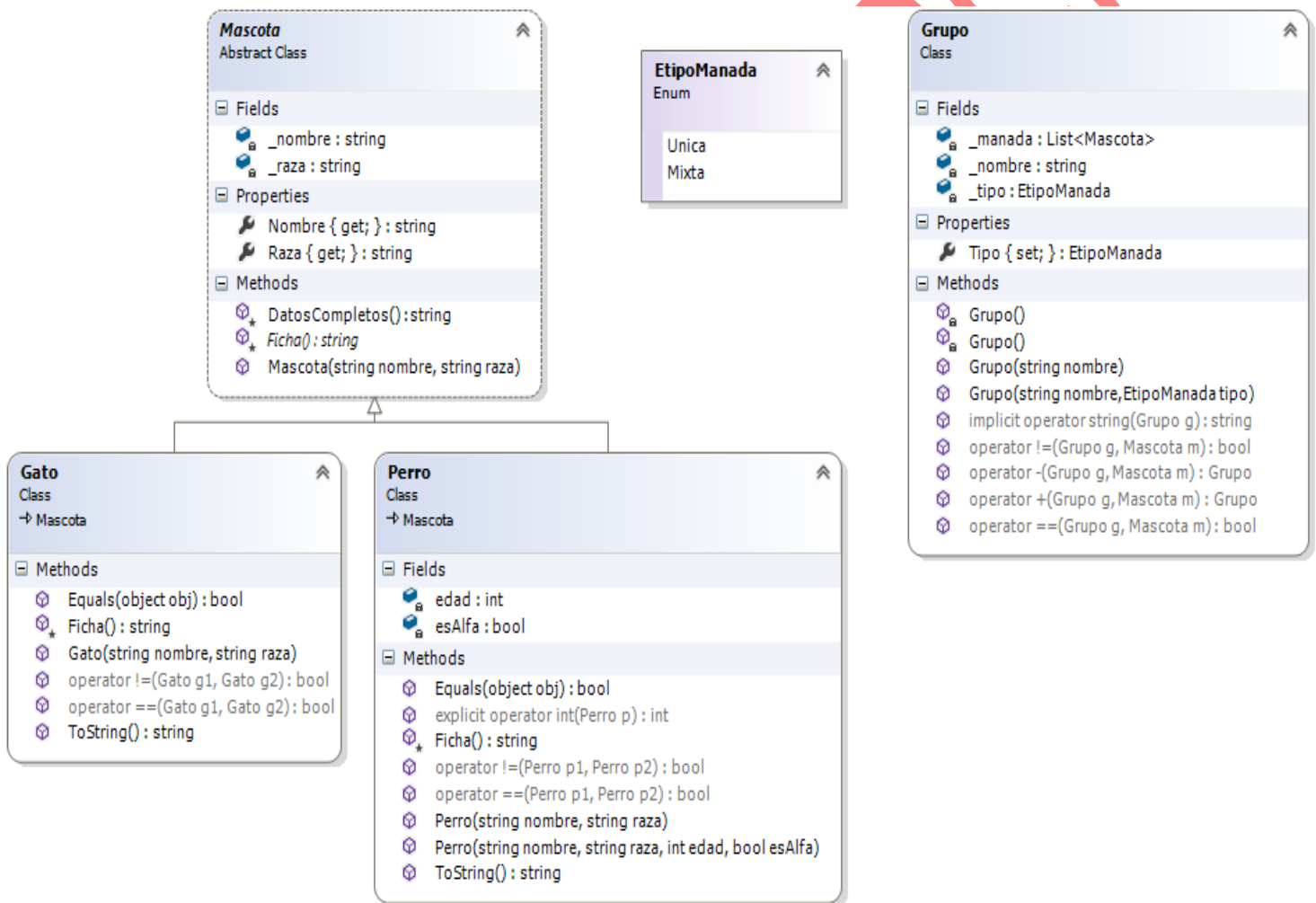
Sólo se corregirá lo que el alumno entregue de la siguiente forma:

Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP (nombrado como **apellido.nombre.división**) y dejar este último en el Escritorio de la máquina. Luego presionar el botón de la barra superior , cargar su apellido y presionar Aceptar. La barra superior deberá cambiar de color.

En todos los casos que sean posibles, reutilizar código.

Se tendrá en cuenta las convenciones aplicadas al proyecto.

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 `DatosCompleto()`. Retornará el nombre y el raza con el formato "Nombre 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.

Implementar el método Ficha(). Retornará toda la información del Perro con el siguiente formato:

- Si es alfa: Ramón Salchicha, alfa de la manada, edad 2 (siendo: Ramón el nombre, Salchicha la raza y 2 la edad)
- Si no lo es: Julio Cruza, edad 13 (siendo: Julio el nombre, Cruza la raza y 13 la edad)

Sobrecarga de Operadores:

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

Sobreescibir:

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

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.

Sobreescibir:

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

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.

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:

Con el siguiente Main, se obtendrá la siguiente salida:

```
static void Main(string[] args)
{
    Perro perroUno = new Perro("Moro", "Pitbull");
    Perro perroDos = new Perro("Julio", "Cruza", 13, false);
    Perro perroTres = new Perro("Ramón", "Salchicha", 2, true);
    Gato gatoUno = new Gato("José", "Angora");
    Gato gatoDos = new Gato("Hernán", "Cruza");
```

```
Ya está el Moro Pitbull, edad 0 en el grupo.
Grupo: Río - tipo: Mixta
Integrantes <6>:
Moro Pitbull, edad 0
Julio Cruza, edad 13
Ramón Salchicha, alfa de la mananada, edad 2
José Angora
Hernán Cruza
Fer Siamés

No está el Fer Siamés en el grupo.
Grupo: Río - tipo: Mixta
Integrantes <4>:
Moro Pitbull, edad 0
Julio Cruza, edad 13
José Angora
Hernán Cruza

No son la misma mascota
```

```
Gato gatoTres = new Gato("Fer", "Siamés");

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

grupoUno += perroUno;
grupoUno += perroDos;
grupoUno += perroUno;
grupoUno += perroTres;
grupoUno += gatoUno;
grupoUno += gatoDos;
grupoUno += gatoTres;

Console.WriteLine(grupoUno);

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

Console.WriteLine(grupoUno);

if (perroUno.Equals("perroUno"))
    Console.WriteLine("Son la misma mascota");
else
    Console.WriteLine("No son la misma mascota");

Console.ReadLine();
}
```

Tiempo máximo para resolver el examen: 90 minutos