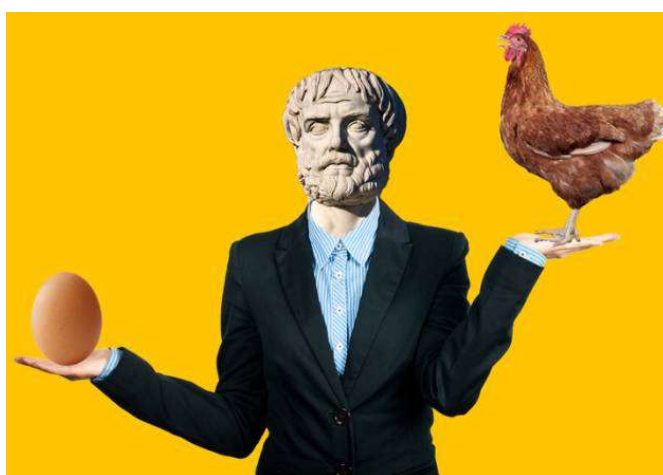


## El Torneo de la Verdad

**NOTA:** Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

Ante el auge y explosión de distintas escuelas filosóficas en lo que hoy conocemos como Antigua Grecia, Sócrates ha encargado a su discípulo Platón (usted) que realice un torneo entre todas las escuelas filosóficas existentes con un propósito: La búsqueda de la Verdad. Sócrates le ha explicado a Platón que las contradicciones y el enfrentamiento racional entre antagonistas filosóficos generan desarrollo en los conceptos e ideas.

Para organizar dicho torneo, su maestro le ha entregado el concepto de Filósofo y un boceto de las acciones a implementar para este torneo en forma de una clase y una interfaz:



```
class Filosofo
{
    ...
    string Nombre { get; set; }
    int Conocimiento { get; set; }
}

interface ITorneoFilosofico
{
    void RegistraEscuela(string escuela, IEnumerable<Filosofo> filosofos);
    IEnumerable<Filosofo> Miembros(string escuela);
    string PerteneceAEscuela(Filosofo filosofo);
    IEnumerable<Filosofo> FilofofosMasDestacados();
    IEnumerable<string> DameEscuelas();
    string Compite(string escuela1, string escuela2);
}
```

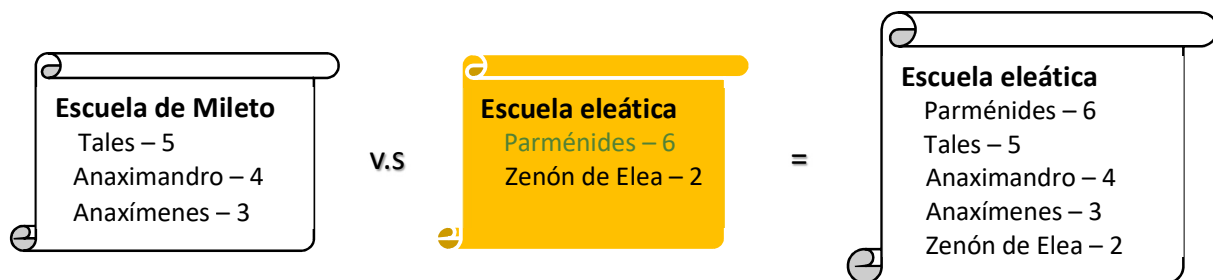
Para la clase `Filosofo` se tiene el nombre del filósofo y un número entero que indica su nivel de conocimiento. Además esta clase implementa el `Equals` y el `GetHashCode` basándose solamente en el nombre del filósofo.

Mientras que para la interfaz `ITorneoFilosofico` se tiene que:

- **RegistraEscuela** debe inscribir en el torneo una escuela conformada por un grupo de filósofos. El listado de filósofos estará ordenado por el valor del conocimiento de mayor a menor.
- **Miembros** devuelve los miembros de una escuela filosófica que está registrada en el torneo en orden descendente con respecto al valor de conocimiento. Si el nombre de la escuela no se encuentra en el torneo se devolverá `null`.
- **PerteneceAEscuela** devuelve la escuela a la que pertenece un filósofo determinado, en caso de que no pertenezca a ninguna se devuelve `null`.
- **FilosofosMasDestacados** devuelve el filósofo principal de cada escuela (el filósofo con más conocimiento por cada escuela, en caso de que existan dos o más con el mismo valor en el tope de una escuela se podrá devolver cualquiera de esos). No importa el orden en que se devuelva.
- **DameEscuelas** devuelve las escuelas inscritas en el torneo hasta el momento.
- **Compite** enfrenta a dos escuelas filosóficas siguiendo las reglas a continuación descritas:
  1. La competencia consiste en enfrentar a los representantes principales de cada escuela en un debate público, el ganador de ese debate se decide por el que mayor índice de conocimientos tenga (propiedad `Conocimiento` de `Filosofo`). Si hay un empate se continua con los segundos representantes con más conocimientos en otro debate, y así sucesivamente con los demás miembros de las escuelas hasta que se encuentre un ganador o hasta que una escuela no tenga contricante que presentar y en tal caso esa escuela resultaría perdedora. En caso de que el resultado total sea empate la regla 2 no se aplica y las dos escuelas permanecen sin cambios.
  2. Luego que se decida el ganador, la escuela que perdió los debates deberá unirse a la otra en aras de la búsqueda de la verdad. Es decir, los miembros de la escuela que perdió pasarán a ser miembros de la escuela ganadora y la escuela perdedora ya no existirá como tal en el torneo.
  3. El retorno de esta función es un `string` que contiene el nombre de la escuela ganadora o el string **"EMPATE"** en caso de que todos los debates terminarán sin vencedor.
  4. Siempre se garantiza que los dos nombres pasados como `string` sean escuelas previamente registradas

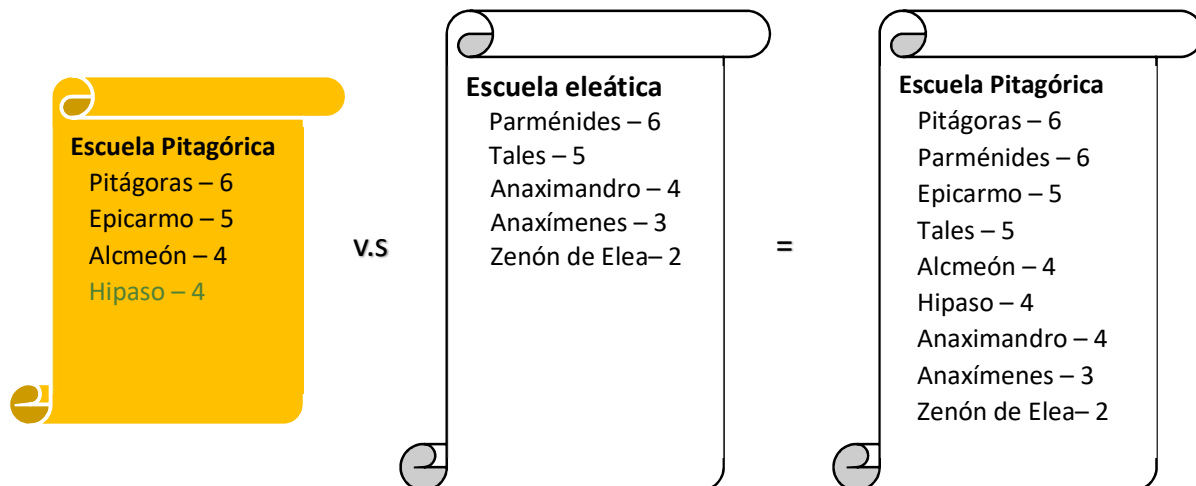
A continuación se muestran 4 ejemplos de competencias entre dos escuelas, marcado en **verde** el filósofo que decidió la competición, en **naranja** la escuela ganadora y el resultado de la unión del perdedor al ganador (esto es solo para que usted entienda mejor las figuras, por supuesto que su solución no tiene que tratar para nada con colores):

#### Ejemplo 1:



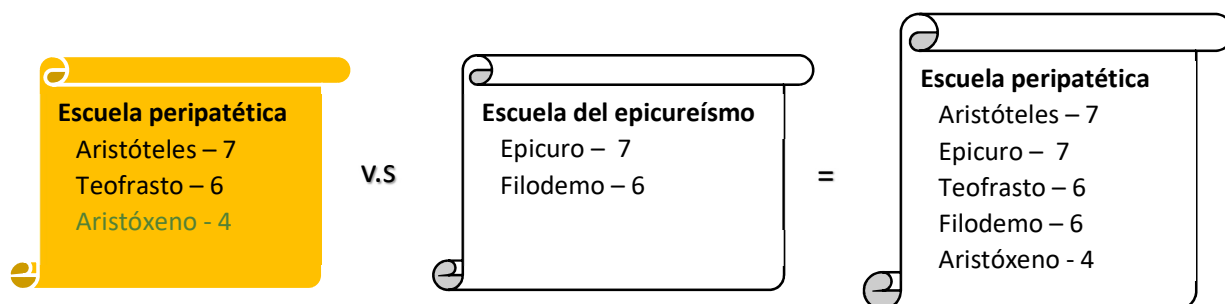
En este primer ejemplo, al enfrentarse Tales (con 5 puntos en el índice de conocimiento) de la Escuela de Mileto a Parménides de la Escuela de Elea (con 6 puntos en el índice de conocimiento) vence el segundo asegurándose la incorporación de la Escuela de Mileto a la de Elea.

#### Ejemplo 2:



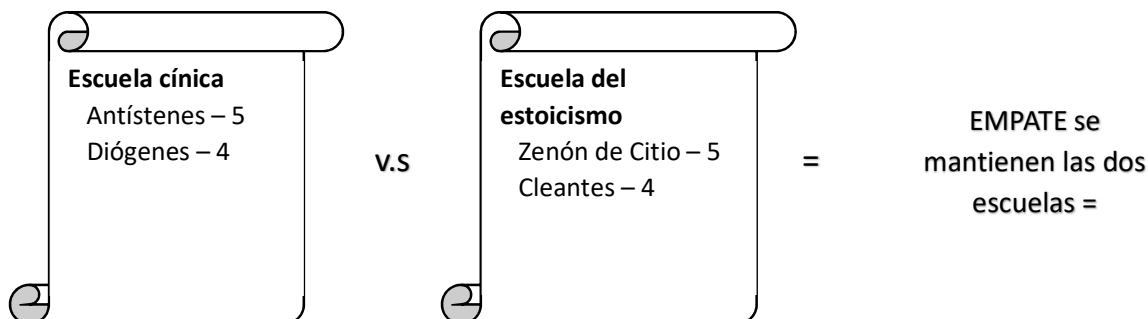
En el segundo ejemplo luego de que los 3 primeros debates entre las escuelas Pitagórica y Eleática terminaran en empate (Pitágoras v.s Parménides, Epicarmo v.s Tales y Alcmeón v.s Anaximandro), Hipaso vence a Anaxímenes, proclamándose ganadora la escuela de los Pitagóricos. Decidido el vencedor, la Escuela Eléatica desaparece y sus miembros pasan a formar parte de la Pitagórica vencedora.

#### Ejemplo 3:



Para este tercer ejemplo la escuela peripatética vence a la escuela de Epicuro luego de que la última no tuviera filósofo para presentar en el tercer debate luego de dos empates previos.

#### Ejemplo 4:



Para este último ejemplo el resultado es un empate, pues ni la Escuela Cínica, ni la Estoica tienen filósofos que presentar luego de 2 empates consecutivos. Por lo cual el torneo no registra cambios.

Finalmente, Sócrates le agrega una restricción adicional a usted alumno programador para que demuestre la sabiduría adquirida, así como su capacidad creativa e innovadora que consiste en:

Para programar la solución no puede usar arrays, ni ninguna de las estructuras de datos existente en System.Collections (ArrayList, List<T>, Dictionary<T>, etc.). El uso de cualquiera de estas estructuras implicará la anulación del examen. Debe resolverlo todo entonces usando solo su propia implementación de enlazables.

Usted debe haber recibido junto a este documento una solución de Visual Studio con dos proyectos: una biblioteca de clases (*Class Library*) y una aplicación de consola (*Console Application*). Usted debe completar la implementación de la clase `TorneoFilosofico` en el *namespace* `Weboo`. Examen que ya implementa la interface `ITorneoFilosofico`. En la biblioteca de clases encontrará la siguiente definición:

```
namespace Weboo.Examen
{
    public class TorneoFilosofico : ITorneoFilosofico
    {
        public string Compite(string escuela1, string escuela2)
        {
            throw new NotImplementedException();
        }

        public IEnumerable<string> DameEscuelas()
        {
            throw new NotImplementedException();
        }

        public IEnumerable<Filosofo> FilósofosMasDestacados()
        {
            throw new NotImplementedException();
        }

        public IEnumerable<Filosofo> Miembros(string escuela)
        {
            throw new NotImplementedException();
        }

        public string PerteneceAEscuela(Filosofo filosofo)
        {
            throw new NotImplementedException();
        }

        public void RegistraEscuela(string escuela, IEnumerable<Filosofo> filosofos)
        {
            throw new NotImplementedException();
        }
    }
}
```

NOTA: Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará.

## Ejemplo

```
ITorneoFilosofico torneo = new TorneoFilosofico();
Console.WriteLine($"Número de escuelas: {torneo.DameEscuelas().Count()}");
// ~> Número de escuelas: 0

Console.WriteLine($"Tales pertenece a: {torneo.PerteneceAEscuela(new Filosofo("Tales"))}");
// ~> Tales pertenece a:

torneo.RegistraEscuela("Escuela de Mileto", new Filosofo[3] {
    new Filosofo("Tales", 5),
    new Filosofo("Anaximandro", 4),
    new Filosofo("Anaximenes", 3)
});

Console.WriteLine($"Número de escuelas: {torneo.DameEscuelas().Count()}");
// ~> Número de escuelas: 1

Console.WriteLine($"Tales pertenece a: {torneo.PerteneceAEscuela(new Filosofo("Tales"))}");
// ~> Tales pertenece a: Escuela de Mileto

torneo.RegistraEscuela("Escuela eleatica", new Filosofo[2] {
    new Filosofo("Parmenides", 6),
    new Filosofo("Zenon de Elea", 2),
});

Console.WriteLine($"Número de escuelas: {torneo.DameEscuelas().Count()}");
// ~> Número de escuelas: 2

Console.WriteLine("Parmenides pertenece a: {0}", torneo.PerteneceAEscuela(new
Filosofo("Parmenides")));
// ~> Parmenides pertenece a: Escuela eleatica

Console.WriteLine("Miembros de la escuela eleatica: ");
foreach (var filosofo in torneo.Miembros("Escuela eleatica"))
{
    Console.WriteLine($"{filosofo.Nombre} ({filosofo.Conocimiento}), ");
}
Console.WriteLine();
// ~> Miembros de la escuela eleatica: Parmenides (6), Zenon de Elea (2),

Console.WriteLine("Ganador entre Escuela de Mileto y Escuela eleatica es: {0}",
    torneo.Compite("Escuela de Mileto", "Escuela eleatica"));
// ~> Ganador entre Escuela de Mileto y Escuela eleatica es: Escuela eleatica

Console.WriteLine("Número de escuelas: {0}", torneo.DameEscuelas().Count());
// ~> Número de escuelas: 1

Console.WriteLine("Miembros de la escuela eleatica: ");
foreach (var filosofo in torneo.Miembros("Escuela eleatica"))
{
    Console.WriteLine($"{filosofo.Nombre} ({filosofo.Conocimiento}), ");
}
Console.WriteLine();
```

```
// ~> Miembros de la escuela eleatica: Parmenides (6), Tales (5), Anaximandro (4),
Anaximenes (3), Zenon de Elea (2),

torneo.RegistraEscuela("Escuela Pitagorica", new Filosofo[4] {
    new Filosofo("Pitagoras", 6),
    new Filosofo("Epicarmo", 5),
    new Filosofo("Alcmeon", 4),
    new Filosofo("Hipaso", 4)
});

Console.WriteLine("Numero de escuelas: {0}", torneo.DameEscuelas().Count());
// ~> Número de escuelas: 2

Console.Write("Filosofos mas destacados: ");
foreach (var filosofo in torneo.FilosofosMasDestacados())
{
    Console.Write($"{filosofo.Nombre} ({filosofo.Conocimiento}), ");
}
Console.WriteLine();
// ~> Filosofos mas destacados: Parmenides (6), Pitagoras (6),

Console.Write("Escuelas: ");
foreach (var escuela in torneo.DameEscuelas())
{
    Console.Write($"{escuela}, ");
}
Console.WriteLine();
// ~> Escuelas: Escuela eleatica, Escuela Pitagorica

Console.WriteLine("Ganador entre Escuela eleatica y Escuela Pitagorica es: {0}",
    torneo.Compite("Escuela eleatica", "Escuela Pitagorica"));
// ~> Ganador entre Escuela eleatica y Escuela Pitagorica es: Escuela Pitagorica

Console.WriteLine("Numero de escuelas: {0}", torneo.DameEscuelas().Count());
// ~> Número de escuelas: 1

Console.Write("Miembros de la Escuela Pitagorica: ");
foreach (var filosofo in torneo.Miembros("Escuela Pitagorica"))
{
    Console.Write($"{filosofo.Nombre} ({filosofo.Conocimiento}), ");
}
Console.WriteLine();
// ~> Miembros de la escuela Pitagorica: Pitagoras (6), Parmenides (6), Epicarmo (5), Tales
(5), Anaximandro (4), Alcmeon (4), Hipaso (4), Anaximenes (3), Zenon de Elea (2),

torneo.RegistraEscuela("Escuela peripatetica", new Filosofo[] {
    new Filosofo("Aristoteles", 7),
    new Filosofo("Teofrasto", 6),
    new Filosofo("Aristoxeno", 4),
});

torneo.RegistraEscuela("Escuela del epicureismo", new Filosofo[] {
    new Filosofo("Epicuro", 7),
    new Filosofo("Filodemo", 6),
});

Console.WriteLine("Numero de escuelas: {0}", torneo.DameEscuelas().Count());
// ~> Número de escuelas: 3

Console.Write("Filosofos mas destacados: ");
foreach (var filosofo in torneo.FilosofosMasDestacados())
{
```

```

        Console.WriteLine($"{filosofo.Nombre} ({filosofo.Conocimiento}), ");
    }
    Console.WriteLine("\n");
    // ~> Filósofos mas destacados: Pitagoras (6), Epicuro (7), Aristoteles (7),

    Console.WriteLine("Epicuro pertenece a: {0}", torneo.PerteneceAEscuela(new
    Filosofo("Epicuro")));
    // ~> Epicuro pertenece a: Escuela del epicureismo

    Console.WriteLine("Ganador entre Escuela peripatetica y Escuela del epicureismo es: {0}",
    torneo.Compite("Escuela peripatetica", "Escuela del epicureismo"));
    // ~> Ganador entre Escuela peripatetica y Escuela del epicureismo es: Escuela peripatetica

    Console.WriteLine("Epicuro pertenece a: {0}", torneo.PerteneceAEscuela(new
    Filosofo("Epicuro")));
    // ~> Epicuro pertenece a: Escuela peripatetica

    Console.WriteLine("Escuelas: ");
    foreach (var escuela in torneo.DameEscuelas())
    {
        Console.WriteLine($"{escuela}, ");
    }
    Console.WriteLine("\n");
    // ~> Escuelas: Escuela peripatetica, Escuela Pitagorica

    torneo.RegistraEscuela("Escuela cinica", new Filosofo[] {
        new Filosofo("Antístenes ", 5),
        new Filosofo("Diogenes", 4),
    });

    torneo.RegistraEscuela("Escuela del estoicismo", new Filosofo[] {
        new Filosofo("Zenon de Citio ", 5),
        new Filosofo("Cleantes", 4),
    });

    Console.WriteLine("Numero de escuelas: {0}", torneo.DameEscuelas().Count());
    // ~> Número de escuelas: 4

    Console.WriteLine("Ganador entre Escuela cinica y Escuela del estoicismo es: {0}",
    torneo.Compite("Escuela cinica", "Escuela del estoicismo"));
    // ~> Ganador entre Escuela cinica y Escuela del estoicismo es: EMPATE

    Console.WriteLine("Escuelas: ");
    foreach (var escuela in torneo.DameEscuelas())
    {
        Console.WriteLine($"{escuela}, ");
    }
    Console.WriteLine("\n");
    // ~> Escuelas: Escuela peripatetica, Escuela Pitagorica, Escuela cinica, Escuela del
    estoicismo

```

**NOTA:** Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y de buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.