

Enunciado:

Tema: Miembros de instancia y miembros estáticos.

Desarrolle la clase cuyo prototipo se muestra en la **Figura 1**. Funcionalmente podemos definir que una conexión tiene un estado que puede ser “**Ocupada, Libre**”, ver **Figura 3**, la propiedad estática “**Instance**” es responsable de retornar una conexión libre o en otras palabras una conexión cuyo estado sea “**Libre**” tal como muestra la **Figura 2**, luego de encontrarla la establece en ocupada (65), en caso de no encontrar una conexión libre, la propiedad “**Instance**” lanza una excepción (69) con el mensaje “Todas las conexiones están ocupadas.”.

EL constructor estático de la **Figura 5**, muestra la existencia de 3 (tres) conexiones las cuales se inicializan como libres y se van ocupando a medida que son solicitadas mediante invocaciones al miembro “**Conexion.Instance**”, de este modo una conexión puede ser liberada si ya no se utiliza, cambiando su estado a “Libre”, con lo cual queda disponible para que la propiedad “**Conexion.Instance**” la retorne al ser invocada.

| |
|----------|
| Figura 1 |
|----------|

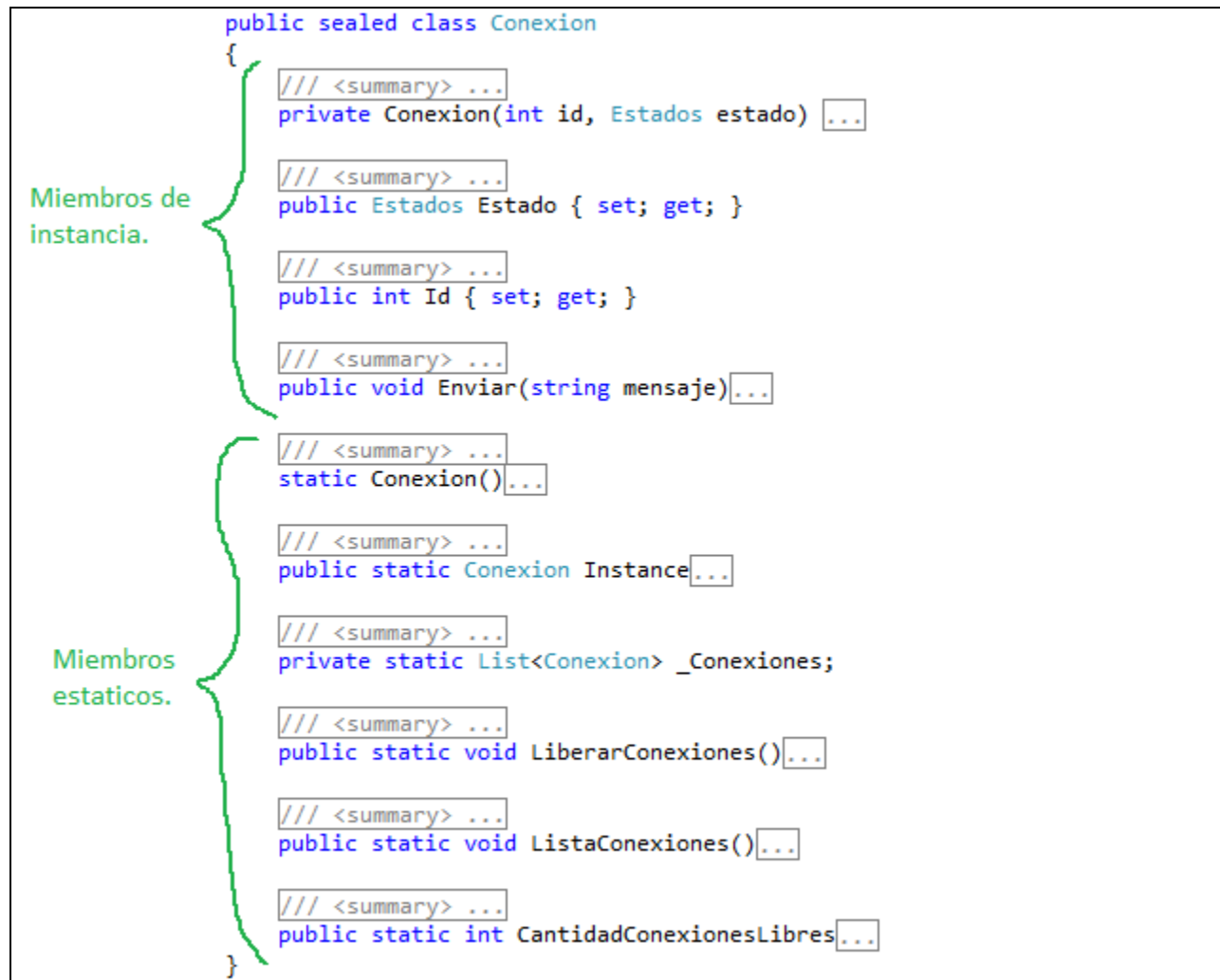


Figura 2

```

53 public static Conexion Instance
54 {
55     get
56     {
57         Conexion conexionLibre = null;
58         foreach (Conexion conexion in _Conexiones)
59         {
60             if (conexionLibre == null)
61             {
62                 if (conexion.Estado == Estados.Libre)
63                 {
64                     conexionLibre = conexion;
65                     conexion.Estado = Estados.Ocupada;
66                 }
67             }
68         }
69         if (conexionLibre == null) throw new Exception("Todas las conexiones están ocupadas.");
70         return conexionLibre;
71     }
72 }

```

Figura 3

```
namespace MaximoConexiones
{
    public enum Estados {Ocupada, Libre};
}
```

La **Figura 4**, muestra el código fuente que es responsable de listar las conexiones y sus estados correspondientes.

Figura 4

```
public static void ListaConexiones()
{
    Console.WriteLine("LISTADO DE CONEXIONES.");
    foreach (Conexion conexion in _Conexiones)
    {
        Console.WriteLine("Id: {0} estado: {1}", conexion.Id, conexion.Estado);
    }
}
```

Figura 5

```
static Conexion()
{
    _Conexiones = new List<Conexion>();
    _Conexiones.Add(new Conexion(0, Estados.Libre));
    _Conexiones.Add(new Conexion(1, Estados.Libre));
    _Conexiones.Add(new Conexion(2, Estados.Libre));
}
```

EL código de la **Figura 6**, arroja como resultado de su ejecución lo que se muestra en la **Figura 7**.

Figura 6

```
10 static void Main(string[] args)
11 {
12     Conexion conexion = Conexion.Instance;
13     conexion.Envia("hola");
14     Conexion.ListaConexiones();
15     conexion.Estado = Estados.Libre;
16
17     Conexion.ListaConexiones();
18
19     //Se ocupan las tres conexiones disponibles.
20     conexion = Conexion.Instance;
21     conexion = Conexion.Instance;
22     conexion = Conexion.Instance;
23     Conexion.ListaConexiones();
24
25     Console.ReadLine();
26 }
```

Figura 7

