

Programación 3

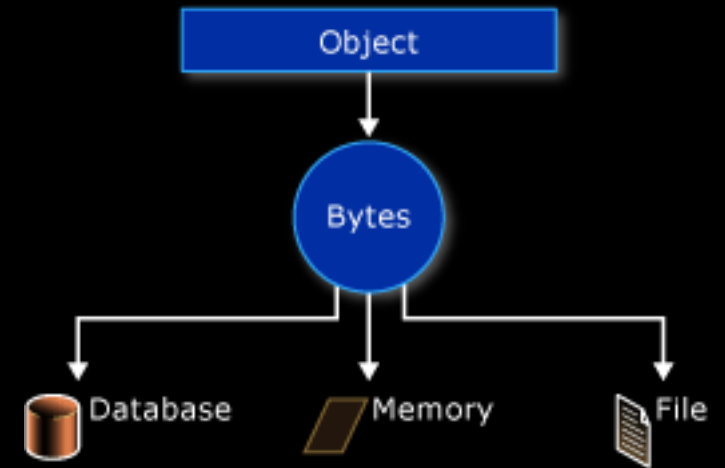
Orientación a objetos con C#

6. Serialización, Archivos

maurogullino@gmail.com

Serialization

- es la operación de convertir la imagen de memoria de un objeto en una serie de bytes
- se pueden almacenar, transmitir, imprimir....
- la operación contraria es *Deserialization*



```
namespace Empresa
{
    [Serializable]
    class Departamento
    {
        private List<Empleado> empleados;
        public Empleado MejorEmpleado { get; set; }

        public Departamento()
        {
            empleados = new List<Empleado>();
        }

        public void agregar(Empleado e)
        {
            empleados.Add(e);
        }
    }
}
```

attribute ←

```
namespace Empresa
{
    [Serializable]
    class Empleado
    {
        public string Nombre { get; set; }
        public decimal Sueldo { get; set; }

        public Empleado(string n, decimal s)
        {
            Nombre = n;
            Sueldo = s;
        }
    }
}
```

escritura

```
static void Main()
{
    Empleado e1 = new Empleado("Pepe", 1000);
    Empleado e2 = new Empleado("María", 5000);

    var d = new Departamento();
    d.agregar(e1);
    d.agregar(e2);
    d.MejorEmpleado = e2;

    Stream flujo = File.Create("empresa.bin");
    BinaryFormatter serializer = new BinaryFormatter();
    serializer.Serialize(flujo, d);
    flujo.Close();
}
```

lectura

```
static void Main()
{
    Departamento d2;

    if (File.Exists("prueba.bin"))
    {
        Stream flujo2 = File.OpenRead("prueba.bin");

        BinaryFormatter deserializer = new BinaryFormatter();
        d2 = (Departamento) deserializer.Deserialize(flujo2);

        flujo2.Close();

        //notar el casting
    }
}
```

Ejercicio

Empleados

Bienvenido al sistema de Gestión de Empleados

Buscar empleado:

Datos del empleado

Nombre:

Sueldo:

