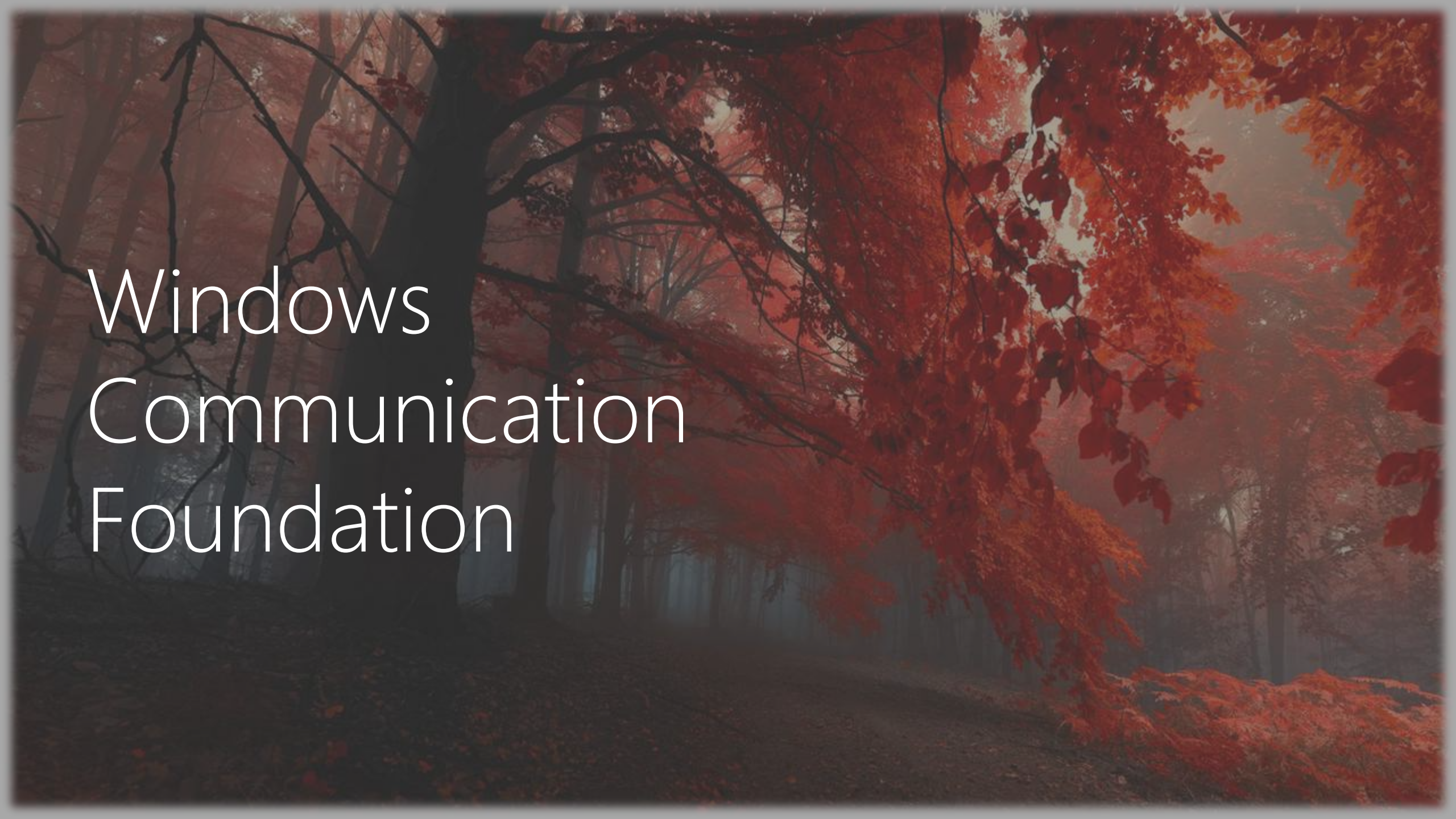
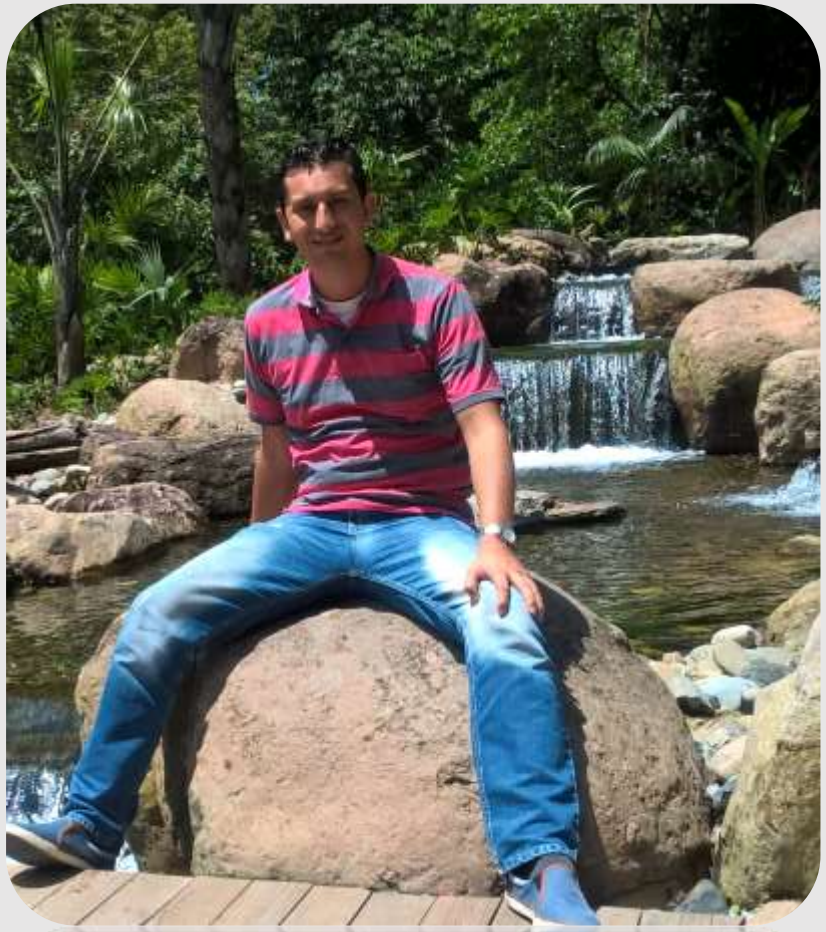

Windows
Communication
Foundation

Microsoft

@serandvaraco

Sergio Andrés Vargas Acosta

Microsoft Certified Solution Architec



https://github.com/serandvaraco

http://unespacioparanet.com

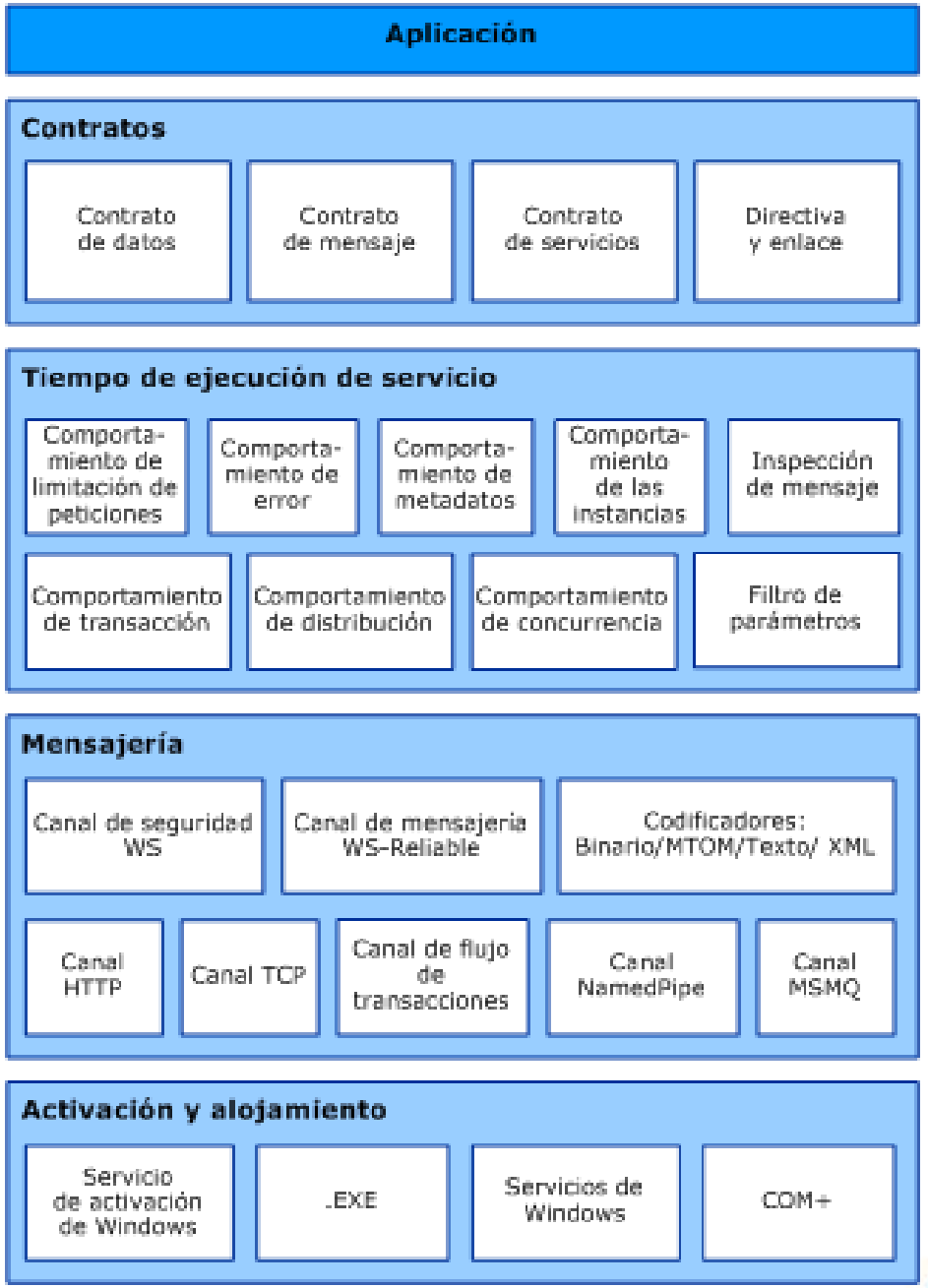
http://fb.com/unespacioparanet

Temas

- Arquitectura
- Implementar un contrato de servicio
- Configuración de servicios
- Mensajes confiables y en cola
- Transacciones
- Hospedar y ejecutar un servicio básico
- Creación de un cliente
- Configuración de un cliente básico
- Uso cliente WCF
- Interoperabilidad e integración
- WCF y ASP.NET Web API
- Compatibilidad con AJAX y REST
- Seguridad
- ASP.NET Webhook
- ASP.NET Signal R

WCF (Arquitectura)

- Contratos
- Ejecución de servicio
- Mensajería
- Activación



Microsoft

Microsoft .NET

Contratos

- Lenguaje de definición de esquemas XML (XSD) .
- protocolos SOAP.
- Las directivas y enlaces estipulan las condiciones exigidas para comunicarse con un servicio. (HTTP - TCP)

Ejecución de servicio

- Se producen durante la operación actual del servicio
- Limitación de peticiones
- Comportamiento de error
- Personalización de procesos en tiempo de ejecución

Mensajería

- Compuesta por canales
- Canales de transporte (TCP , MSMQ)
- Canales de protocolo (WS-Security, WS-Reliability)

Alojamiento y activación

- Un servicio con host propio
- hospedar o ejecutar en un ejecutable administrado por un agente externo (IIS, WAS, COM+)

Ciclo de vida de programación básica

- Definir contrato de servicio
- Implementar el contrato
- Configurar el servicio
- Hospedar el servicio
- Compilar una aplicación

Define el Servicio WCF

```
namespace WCFWebServices
{
    using System.Collections.Generic;
    using System.ServiceModel;

    [ServiceContract]
    0 referencias
    public interface IMarket
    {
        [OperationContract]
        0 referencias
        IList<Product> GetProducts();
    }
}
```

Indica que una interfaz o una clase define un contrato de servicio en una aplicación (WCF)

Indica que un método define una operación que es parte de un contrato de servicio en una aplicación (WCF)

Microsoft

Microsoft .NET

Define el Servicio WCF

```
namespace WCFWebServices
{
    using System.Collections.Generic;
    using System.Net.Security;
    using System.ServiceModel;

    [ServiceContract(
            Namespace = "http://unespacioparanet.com",
            Name = "MarketService",
            ProtectionLevel = ProtectionLevel.EncryptAndSign)]
    0 referencias
    public interface IMarket
    {
        [OperationContract()]
        0 referencias
        IList<Product> GetProducts();
    }
}
```

El servicio puede enriquecerse con niveles de protección

Indica que un método define una operación que es parte de un contrato de servicio en una aplicación (WCF)

Configuración del Servicio WCF

```
<system.serviceModel>
  <behaviors>
    <serviceBehaviors>
      <behavior>
        <serviceMetadata httpGetEnabled="true" httpsGetEnabled="true"/>
        <serviceDebug includeExceptionDetailInFaults="false"/>
      </behavior>
    </serviceBehaviors>
  </behaviors>
  <protocolMapping>
      <add binding="basicHttpsBinding" scheme="https" />
  </protocolMapping>
  <serviceHostingEnvironment aspNetCompatibilityEnabled="true"
                             multipleSiteBindingsEnabled="true" />
</system.serviceModel>
```

Comportamientos del Servicio

Indica los Metadatos de acceso al servicio

Esquema de conexión

Compatibilidad de Host

Microsoft

Microsoft .NET

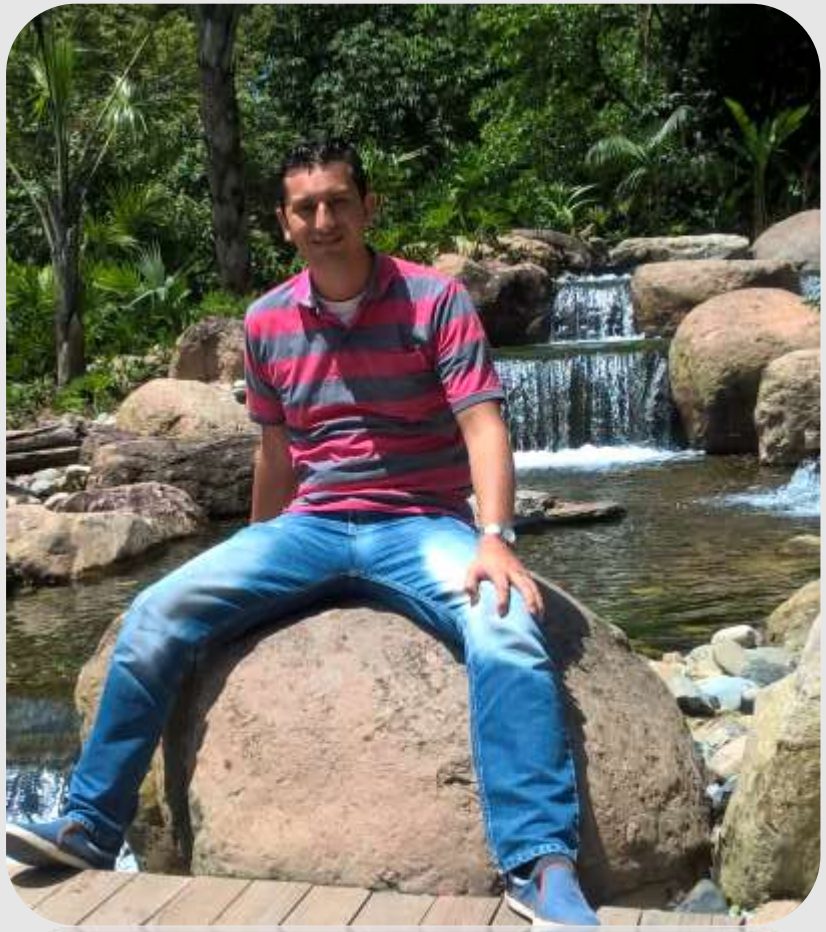

Demo:
programación básica WCF

Microsoft

@serandvaraco

Sergio Andrés Vargas Acosta

Microsoft Certified Solution Architec



https://github.com/serandvaraco

http://unespacioparanet.com

http://fb.com/unespacioparanet

Contrato de datos

Un contrato de datos es un acuerdo formal entre un servicio y un cliente que abstractamente describe los datos que se van a intercambiar.

```
[DataContract]
1 referencia
public class NumeroComplejo
{
    [DataMember]
    1 referencia
    public double Real { get; set; }
    [DataMember]
    1 referencia
    public double Imaginario { get; set; }

    0 referencias
    public NumeroComplejo(double real, double imaginario)
    {
        this.Real = real; this.Imaginario = imaginario;
    }

}
```

Definiciones

Nombres de miembros de datos

```
[DataContract(Name ="ComplexNumbers")]
1 referencia
public class NumeroComplejo
{
    [DataMember]
    1 referencia
    public double Real { get; set; }
    [DataMember]
    1 referencia
    public double Imaginario { get; set; }

    0 referencias
    public NumeroComplejo(double real, double imaginario)
    {
        this.Real = real; this.Imaginario = imaginario;
    }

}
```

```
[DataContract]
public class BaseType
{

    [DataMember]
    public string zebra;
}
[DataContract]
public class DerivedType : BaseType
{
    [DataMember(Order = 0)]
    public string bird;
    [DataMember(Order = 1)]
    public string parrot;
    [DataMember]
    public string dog;
    [DataMember(Order = 3)]
    public string antelope;
    [DataMember]
    public string cat;
    [DataMember(Order = 1)]
    public string albatross;
}
```

Orden de los miembros de datos

```
[DataContract]
2 referencias
public class Shape { }

[DataContract(Name = "Circle")]
0 referencias
public class CircleType : Shape { }

[DataContract(Name = "Triangle")]
0 referencias
public class TriangleType : Shape { }
```

Tipos conocidos de contratos

Microsoft

```
[DataContract]
1 referencia
public class Book { }

[DataContract]
1 referencia
public class Magazine { }

[DataContract]
[KnownType(typeof(Book))]
[KnownType(typeof(Magazine))]
0 referencias
public class LibraryCatalog
{
    [DataMember]
    System.Collections.Hashtable theCatalog;
}
```

Tipos No conocidos de contratos

Microsoft

```
[DataContract]
[KnownType(typeof(int[]))]
0 referencias
public class MathOperationData
{
    private object numberValue;
    [DataMember]
    0 referencias
    public object Numbers
    {
        get { return numberValue; }
        set { numberValue = value; }
    }
    //[DataMember]
    //public Operation Operation;
}
```

Tipos de contratos No Conocidos

Microsoft

```
[DataContract]
[KnownType("GetKnownType")]
0 referencias
public class DrawingRecord2<T>
{
    [DataMember]
    private T TheData;
    [DataMember]
    private GenericDrawing<T> TheDrawing;

    0 referencias
    private static Type[] GetKnownType()
    {
        Type[] t = new Type[2];
        t[0] = typeof(ColorDrawing<T>);
        t[1] = typeof(BlackAndWhiteDrawing<T>);
        return t;
    }
}
```

Tipos conocidos de contratos

Genericos

Microsoft

```
[DataContract]
0 referencias
public class Employee
{
    [DataMember]
    public string employeeName = null;
    [DataMember]
    public int employeeID = 0;
    [DataMember(EmitDefaultValue = false)]
    public string position = null;
    [DataMember(EmitDefaultValue = false)]
    public int salary = 0;
    [DataMember(EmitDefaultValue = false)]
    public int? bonus = null;
    [DataMember(EmitDefaultValue = false)]
    public int targetSalary = 57800;
}
```

Valores por defecto

Microsoft

```
[DataContract]
0 referencias
public class Car
{
    [DataMember]
    public string model;
    [DataMember]
    public CarConditionEnum condition;
}

[DataContract(Name = "CarCondition")]
1 referencia
public enum CarConditionEnum
{
    [EnumMember]
    New,
    [EnumMember]
    Used,
    [EnumMember]
    Rental,
    Broken,
    Stolen
}
```

Enumeraciones

Microsoft

```
[DataContract(Name = "CarCondition")]
0 referencias
public enum CarConditionWithDifferentNames
{
    [EnumMember(Value = "New")]
    BrandNew,
    [EnumMember(Value = "Used")]
    PreviouslyOwned,
    [EnumMember]
    Rental
}
```

Enumeraciones Personalizacion de valores

Microsoft