# CAPÍTULO IV

**PRUEBAS**

## INTRODUCCIÓN

En este capítulo primeramente se realizarán pruebas unitarias para la comprobación de distintos componentes del programa tales como clases y métodos, las cuáles estarán documentadas mediante una estructura de tablas donde se incluye la llamada, la descripción, la entrada, la referencia, el código y la salida.

A continuación se describirá el ambiente de pruebas, el cual tendrá como base a 4 computadores comunicándose entre si por medio del protocolo RTPS y en contraste se realizará la misma prueba con el protocolo RT-CORBA o Ada-DSA o DRTSJ. Además se adjunta capturas de pantalla tanto de la aplicación utilizando los protocolos y de capturas del flujo de datos con la herramienta wireshark, y se presenta un manual de usuario de las aplicaciones y del protocolo RTPS.

Finalmente, se realizará una comparación midiendo tiempos de respuesta y eficiencia del protocolo dentro de nuestro ambiente de pruebas.

## Pruebas unitarias del api rtps

### Codificadores

#### Prueba de los Elementos de los Mensajes.

|  |  |
| --- | --- |
| **Llamada:**  public static DataEncapsulation Serialize<T>(T obj, Encapsulation scheme = Encapsulation.CDR\_BE)  public static DataEncapsulation Deserialize(IoBuffer buffer, int length) | |
| ***Descripción*** | En esta prueba se verifica que el fichero de configuración no sea nulo |
| ***Entrada*** | Inicialmente se tiene el fichero de configuración |
| ***Referencia*** |  |
| ***Código*** | [TestMethod]  public void TestLocatorIpV4CDR\_BE()  {  Encapsulation Scheme = Encapsulation.CDR\_BE;  int bufferSize = 16 + 4 + 4 + CDRHeaderSize;  Locator v1 = new Locator(IPAddress.Parse("10.20.30.40"), 2700);  SerializedPayload payload = new SerializedPayload();  payload.DataEncapsulation = EncapsulationManager.Serialize<Locator>(v1, Scheme);  IoBuffer buffer = IoBuffer.Wrap(payload.DataEncapsulation.SerializedPayload);  Assert.AreEqual(bufferSize, buffer.Remaining);  Assert.AreEqual("00 00 00 00 00 00 00 01 00 00 0A 8C 00 00 00 00 00 00 00 00 00 00 00 00 0A 14 1E 28", buffer.GetHexDump());  Locator v2 = EncapsulationManager.Deserialize<Locator>(buffer);  Assert.AreEqual(v1, v2);  } |
| ***Salida*** | Nombre de la prueba: TestExistConfiguration  Resultado de la prueba: https://i-msdn.sec.s-msft.com/dynimg/IC689871.png  Duración de la prueba: 0:00:00,0286387 |

#### Prueba de Mensajes

### Transporte

#### Prueba de Detección de paquetes RTPS.

#### Pruebas de paquetes RTPS.

#### Pruebas del Simulador UDP.

### Utils

#### Pruebas de configuración.

#### Pruebas del generador de identidad.

#### Pruebas del PeriodicWorker.

#### Pruebas de tiempo

### Serializador

#### Pruebas del BuiltinTopic

#### Pruebas de Encapsulación CDR.

#### Pruebas de exploración de tipo.

#### Pruebas de paquetes.

#### Pruebas de primitivas.

|  |  |
| --- | --- |
| **Llamada:**  App.config | |
| ***Descripción*** | En esta prueba se verifica que el fichero de configuración no sea nulo |
| ***Entrada*** | Inicialmente se tiene el fichero de configuración |
| ***Referencia*** |  |
| ***Código*** | public void TestExistConfiguration()  {  Assert.IsNotNull(ddsConfig);  } |
| ***Salida*** | Nombre de la prueba: TestExistConfiguration  Resultado de la prueba: https://i-msdn.sec.s-msft.com/dynimg/IC689871.png  Duración de la prueba: 0:00:00,0286387 |