```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace TesisChat
{
    class ChatMessage
    {
        private readonly string values;

        public ChatMessage(string values)
        {
            this.values = values;
        }

        public string Values
        {
            get { return this.values; }
        }

        public override string ToString()
        {
            return this.values;
        }
    }
}
```

```
1  namespace TesisChat
2  {
3      partial class frmChat
4      {
5          /// <summary>
6          /// Variable del diseñador requerida.
7          /// </summary>
8          private System.ComponentModel.IContainer components = null;
9
10         /// <summary>
11         /// Limpiar los recursos que se estén utilizando.
12         /// </summary>
13         /// <param name="disposing">true si los recursos administrados se deben desechar; false en caso      ↵
           contrario.</param>
14         protected override void Dispose(bool disposing)
15         {
16             if (disposing && (components != null))
17             {
18                 components.Dispose();
19             }
20             base.Dispose(disposing);
21         }
22
23         #region Código generado por el Diseñador de Windows Forms
24
25         /// <summary>
26         /// Método necesario para admitir el Diseñador. No se puede modificar
27         /// el contenido del método con el editor de código.
28         /// </summary>
29         private void InitializeComponent()
30         {
31             this.txtMessage = new System.Windows.Forms.TextBox();
32             this.mnsMenu = new System.Windows.Forms.MenuStrip();
33             this.inicioToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
34             this.salirToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
35             this.conexiónToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
36             this.btnSend = new System.Windows.Forms.Button();
37             this.lsvHistory = new System.Windows.Forms.ListBox();
38             this.desconexiónToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
39             this.mnsMenu.SuspendLayout();
40             this.SuspendLayout();
41             //
42             // txtMessage
43             //
44             this.txtMessage.Location = new System.Drawing.Point(12, 237);
45             this.txtMessage.Name = "txtMessage";
46             this.txtMessage.Size = new System.Drawing.Size(369, 20);
47             this.txtMessage.TabIndex = 0;
48             this.txtMessage.TextChanged += new System.EventHandler(this.txtMessage_TextChanged);
49             this.txtMessage.Enter += new System.EventHandler(this.txtMessage_Enter);
50             this.txtMessage.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.              ↵
           txtMessage_KeyPress);
51             //
52             // mnsMenu
53             //
54             this.mnsMenu.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
55             this.inicioToolStripMenuItem,
56             this.conexiónToolStripMenuItem,
57             this.desconexiónToolStripMenuItem});
58             this.mnsMenu.Location = new System.Drawing.Point(0, 0);
59             this.mnsMenu.Name = "mnsMenu";
60             this.mnsMenu.Size = new System.Drawing.Size(534, 24);
61             this.mnsMenu.TabIndex = 1;
62             this.mnsMenu.Text = "menuStrip1";
63             //
64             // inicioToolStripMenuItem
65             //
66             this.inicioToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
67             this.salirToolStripMenuItem});
```

```
68              this.inicioToolStripMenuItem.Name = "inicioToolStripMenuItem";
69              this.inicioToolStripMenuItem.Size = new System.Drawing.Size(48, 20);
70              this.inicioToolStripMenuItem.Text = "Inicio";
71              //
72              // salirToolStripMenuItem
73              //
74              this.salirToolStripMenuItem.Name = "salirToolStripMenuItem";
75              this.salirToolStripMenuItem.Size = new System.Drawing.Size(96, 22);
76              this.salirToolStripMenuItem.Text = "Salir";
77              this.salirToolStripMenuItem.Click += new System.EventHandler(this.salirToolStripMenuItem_Click);
78              //
79              // conexiónToolStripMenuItem
80              //
81              this.conexiónToolStripMenuItem.Name = "conexiónToolStripMenuItem";
82              this.conexiónToolStripMenuItem.Size = new System.Drawing.Size(69, 20);
83              this.conexiónToolStripMenuItem.Text = "Conexión";
84              this.conexiónToolStripMenuItem.Click += new System.EventHandler(this.          ↙
       conexiónToolStripMenuItem_Click);
85              //
86              // btnSend
87              //
88              this.btnSend.Location = new System.Drawing.Point(388, 237);
89              this.btnSend.Name = "btnSend";
90              this.btnSend.Size = new System.Drawing.Size(120, 20);
91              this.btnSend.TabIndex = 2;
92              this.btnSend.Text = "Send";
93              this.btnSend.UseVisualStyleBackColor = true;
94              this.btnSend.Click += new System.EventHandler(this.btnSend_Click);
95              //
96              // lsvHistory
97              //
98              this.lsvHistory.FormattingEnabled = true;
99              this.lsvHistory.Location = new System.Drawing.Point(12, 41);
100             this.lsvHistory.Name = "lsvHistory";
101             this.lsvHistory.Size = new System.Drawing.Size(493, 173);
102             this.lsvHistory.TabIndex = 5;
103             //
104             // desconexiónToolStripMenuItem
105             //
106             this.desconexiónToolStripMenuItem.Name = "desconexiónToolStripMenuItem";
107             this.desconexiónToolStripMenuItem.Size = new System.Drawing.Size(86, 20);
108             this.desconexiónToolStripMenuItem.Text = "Desconexión";
109             this.desconexiónToolStripMenuItem.Click += new System.EventHandler(this.          ↙
       desconexiónToolStripMenuItem_Click);
110             //
111             // frmChat
112             //
113             this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
114             this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
115             this.ClientSize = new System.Drawing.Size(534, 291);
116             this.Controls.Add(this.lsvHistory);
117             this.Controls.Add(this.btnSend);
118             this.Controls.Add(this.txtMessage);
119             this.Controls.Add(this.mnsMenu);
120             this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.SizableToolWindow;
121             this.MainMenuStrip = this.mnsMenu;
122             this.Name = "frmChat";
123             this.Text = "DDS-RTPS (Demo) Chat";
124             this.mnsMenu.ResumeLayout(false);
125             this.mnsMenu.PerformLayout();
126             this.ResumeLayout(false);
127             this.PerformLayout();
128
129         }
130
131         #endregion
132
133         private System.Windows.Forms.TextBox txtMessage;
134         private System.Windows.Forms.MenuStrip mnsMenu;
```

```
135        private System.Windows.Forms.ToolStripMenuItem inicioToolStripMenuItem;
136        private System.Windows.Forms.ToolStripMenuItem salirToolStripMenuItem;
137        private System.Windows.Forms.ToolStripMenuItem conexiónToolStripMenuItem;
138        private System.Windows.Forms.Button btnSend;
139        private System.Windows.Forms.ListBox lsvHistory;
140        private System.Windows.Forms.ToolStripMenuItem desconexiónToolStripMenuItem;
141    }
142 }
143
144
```

```csharp
1 namespace TesisChat
2 {
3     partial class frmId
4     {
5         /// <summary>
6         /// Required designer variable.
7         /// </summary>
8         private System.ComponentModel.IContainer components = null;
9
10         /// <summary>
11         /// Clean up any resources being used.
12         /// </summary>
13         /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
14         protected override void Dispose(bool disposing)
15         {
16             if (disposing && (components != null))
17             {
18                 components.Dispose();
19             }
20             base.Dispose(disposing);
21         }
22
23         #region Windows Form Designer generated code
24
25         /// <summary>
26         /// Required method for Designer support - do not modify
27         /// the contents of this method with the code editor.
28         /// </summary>
29         private void InitializeComponent()
30         {
31             this.txtId = new System.Windows.Forms.TextBox();
32             this.btnEnter = new System.Windows.Forms.Button();
33             this.lblId = new System.Windows.Forms.Label();
34             this.SuspendLayout();
35             //
36             // txtId
37             //
38             this.txtId.Location = new System.Drawing.Point(12, 54);
39             this.txtId.Name = "txtId";
40             this.txtId.Size = new System.Drawing.Size(169, 20);
41             this.txtId.TabIndex = 0;
42             this.txtId.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.txtId_KeyPress);
43             //
44             // btnEnter
45             //
46             this.btnEnter.DialogResult = System.Windows.Forms.DialogResult.OK;
47             this.btnEnter.Location = new System.Drawing.Point(45, 86);
48             this.btnEnter.Name = "btnEnter";
49             this.btnEnter.Size = new System.Drawing.Size(118, 36);
50             this.btnEnter.TabIndex = 1;
51             this.btnEnter.Text = "Enter";
52             this.btnEnter.UseVisualStyleBackColor = true;
53             this.btnEnter.Click += new System.EventHandler(this.btnEnter_Click);
54             //
55             // lblId
56             //
57             this.lblId.AutoSize = true;
58             this.lblId.Location = new System.Drawing.Point(42, 28);
59             this.lblId.Name = "lblId";
60             this.lblId.Size = new System.Drawing.Size(70, 13);
61             this.lblId.TabIndex = 2;
62             this.lblId.Text = "Enter your Id:";
63             this.lblId.Click += new System.EventHandler(this.label1_Click);
64             //
65             // frmId
66             //
67             this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
68             this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
69             this.ClientSize = new System.Drawing.Size(199, 131);
```

```
70              this.ControlBox = false;
71              this.Controls.Add(this.lblId);
72              this.Controls.Add(this.btnEnter);
73              this.Controls.Add(this.txtId);
74              this.MaximizeBox = false;
75              this.MinimizeBox = false;
76              this.Name = "frmId";
77              this.ShowIcon = false;
78              this.ShowInTaskbar = false;
79              this.Text = "Login";
80              this.Load += new System.EventHandler(this.frmId_Load);
81              this.ResumeLayout(false);
82              this.PerformLayout();
83
84          }
85
86          #endregion
87
88          private System.Windows.Forms.TextBox txtId;
89          private System.Windows.Forms.Button btnEnter;
90          private System.Windows.Forms.Label lblId;
91      }
92 }
```

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace TesisChat
{
    static class Program
    {
        /// <summary>
        /// Punto de entrada principal para la aplicación.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(frmChat.GetInstance());
        }
    }
}
```

```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace TesisChat
8
9  {
10     public class User
11     {
12         private string userName;
13         private string id;
14
15          public User(string userName,string id)
16         {
17             this.userName = userName;
18             this.id = id;
19
20
21         }
22
23          public string UserName
24          {
25
26             get
27             {
28                 return this.userName;
29             }
30             set
31             {
32                 userName = value;
33             }
34
35
36         }
37          public string Id
38          {
39             get
40             {
41                 return this.id;
42             }
43             set
44             {
45                 id = value;
46             }
47         }
48         public override string ToString()
49         {
50             return this.userName;
51         }
52     }
53 }
54
```

```xml
1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <configuration>
4
5   <configSections>
6     <section name="DDS" type="Doopec.Configuration.DDSConfigurationSection, Doopec, Version=0.1.*, Culture= ↙
   neutral, PublicKeyToken=null"/>
7     <section name="RTPS" type="Doopec.Configuration.RTPSConfigurationSection, Doopec, Version=0.1.*, Culture ↙
   =neutral, PublicKeyToken=null"/>
8   </configSections>
9   <DDS xmlns="urn:Configuration" vendor="Doopec" version="2.1">
10    <!--
11    This is just a minimal sample configuration file that shows how to declare
12    the configuration sections.
13
14    Because an XML Schema Definition (XSD) is generated for each configuration
15    section, it should be trivial to edit these files because you have
16    IntelliSense on the XML definition.
17    -->
18    <boostrapType name="default" type="Doopec.Dds.Core.BootstrapImpl, Doopec"/>
19    <domains>
20      <!--
21    Defining the domains and the basic characteristics for every one,
22    like the profile that are used, and the configuration of QoS policies
23    that are going to use in the domain, finally a identifier is defined
24    -->
25      <domain name="Chat" id="0">
26        <transportProfile name="defaultRtps"/>
27        <qoSProfile name="defaultQoS"/>
28        <guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA061C"/>
29      </domain>
30      <domain name="Main" id="1">
31        <transportProfile name="defaultRtps"/>
32        <qoSProfile name="defaultQoS"/>
33        <guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA061D"/>
34      </domain>
35     <domain name="Main" id="2">
36        <transportProfile name="defaultRtps"/>
37        <qoSProfile name="defaultQoS"/>
38        <guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA061E"/>
39      </domain>
40       <domain name="Main" id="3">
41        <transportProfile name="defaultRtps"/>
42        <qoSProfile name="defaultQoS"/>
43        <guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA061F"/>
44      </domain>
45      <domain name="Main" id="4">
46        <transportProfile name="defaultRtps"/>
47        <qoSProfile name="defaultQoS"/>
48        <guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA062A"/>
49      </domain>
50    </domains>
51    <logLevel levelMin="DEBUG" levelMax="FATAL"/>
52    <!--
53    Defining the default profile of QoS Policy
54    entityFactory: Specifies whether an entity will have enable() called when it is created
55    -->
56    <qoSProfiles>
57      <qoSProfileDef name="defaultQoS">
58        <domainParticipantFactoryQos name="defaultDomainParticipantFactoryQoS">
59          <entityFactory autoenableCreatedEntities="true"/>
60        </domainParticipantFactoryQos>
61        <!--
62      Specifying the default values for QoS policy on DomainParticipant
63      entityFactory: Specifies whether an entity will have enable() called when it is created
64      userData: User Data not known by the middleware, but distributed by means of built-in topics
65      -->
66        <domainParticipantQos name="defaultDomainParticipantQoS">
67          <entityFactory autoenableCreatedEntities="true"/>
```

```
 68          <userData  value=""/>
 69        </domainParticipantQoS>
 70        <!--
 71    Specifying the default values for QoS policy on Topic
 72    topicData: User Data not known by the middleware, but distributed by means of built-in topics
 73    deadline: Set a maximum time in which an update must be published
 74    durability: This policy expresses if the data should 'outlive' their writing time,
 75         it can be VOLATILE, TRANSIENT_LOCAL_TRANSCIENT or PERSISTENT
 76    -->
 77       <topicQoS name="defaultTopicQoS">
 78         <topicData value=""/>
 79         <deadline  period="100"/>
 80         <durability  kind="VOLATILE"/>
 81       </topicQoS>
 82       <!--
 83    Specifying the default values for QoS policy on Publisher
 84    entityFactory: Specifies whether an entity will have enable() called when it is created
 85    groupData: User data not known by the middleware, but distributed by means of built-in topics
 86    partition: List of partitions that the enitity will connect with
 87    presentation: Specifies how the samples representing changes to data instances are presented to the    ↙
    subscribing application
 88         accessScope determines the largest scope spanning the entities for which the order and coherency↙
     of changes
 89         can be preserved, the two booleans control whether coherentAccess and orderedAccess are         ↙
    supported within
 90         the scope accessScope
 91    -->
 92       <publisherQoS name="defaultPublisherQoS">
 93         <entityFactory autoenableCreatedEntities="true"/>
 94         <groupData value=""/>
 95         <partition value=""/>
 96         <presentation accessScope="INSTANCE" coherentAccess="true" orderedAccess="true"/>
 97       </publisherQoS>
 98       <!--
 99    Specifying the default values for QoS policy on Subscriber
100    entityFactory: Specifies whether an entity will have enable() called when it is created
101    groupData: User data not known by the middleware, but distributed by means of built-in topics
102    partition: List of partitions that the enitity will connect with
103    presentation: Specifies how the samples representing changes to data instances are presented to the    ↙
    subscribing application
104         accessScope determines the largest scope spanning the entities for which the order and coherency↙
     of changes
105         can be preserved, the two booleans control whether coherentAccess and orderedAccess are         ↙
    supported within
106         the scope accessScope
107    List of partitions that the enitity will connect with. Allows restricting connections to only those   ↙
    entities with a matching partition
108    -->
109       <subscriberQoS name="defaultSubscriberQoS">
110         <entityFactory autoenableCreatedEntities="true"/>
111         <groupData value=""/>
112         <partition value=""/>
113         <presentation accessScope="INSTANCE" coherentAccess="true" orderedAccess="true"/>
114       </subscriberQoS>
115       <!--
116    Specifying the default values for QoS policy on DataWriter
117    deadline: Set a maximum time in which an update must be published
118    destinationOrder: How data is ordered on the receiving side
119    userData: User Data not known by the middleware, but distributed by means of built-in topics
120    durability: This policy expresses if the data should 'outlive' their writing time,
121         it can be VOLATILE, TRANSIENT_LOCAL_TRANSCIENT or PERSISTENT
122    durabilityService: Specifies the configuration of the durability service. That is, the service that  ↙
    implements
123            the DURABILITY kind of TRANSIENT and PERSISTENT
124    history: Buffering of data values before they are sent
125        KEEP_ALL tries to not lose any values
126        KEEP_LAST keeps up to "depth" values
127    latencyBudget: This is a HINT.Maximum duration between the data being written and data recevied by   ↙
    subscribing application
```

```
128        lifespan: Sets the expiration time of a published value. The expiration time is the time published +  ↙
     duration
129        ownershipStrength: Used to determine who owns the Instance if OWNERSHIP Qos is set to EXCLUSIVE
130        liveliness: Method used to maintain liveliness status. Length of time with no liveliness updates means ↙
      the entity is dead
131              Decreassing the leaseDuration shortens the time to detect the entity has gone away
132        reliability: Reliablity of the connection. BEST_EFFORT means values may be dropped or lost
133        resourceLimits: Limits amount of resources to be used to meet other QoS
134        transportPriority: This is a HINT. Suggestion for setting the priority of the transport
135        writerDataLifecycle: Indicates whether the system will dispose of an Instance when unregister() is     ↙
     called
136        -->
137          <dataWriterQoS name="defaultDataWriterQoS">
138            <deadline period="0"/>
139            <destinationOrder  kind="BY_SOURCE_TIMESTAMP"/>
140            <durability  kind="TRANSIENT"/>
141            <durabilityService historyDepth="0" historyKind="KEEP_LAST" maxInstances="1" maxSamples="1"        ↙
     maxSamplesPerInstance="1" serviceCleanupDelay="1"/>
142            <history kind="KEEP_LAST" depth="1"/>
143            <latencyBudget duration="1000000000"/>
144            <lifespan duration="100000000"/>
145            <liveliness kind="AUTOMATIC" leaseDuration="1"/>
146            <ownership kind="SHARED"/>
147            <ownershipStrength value="100"/>
148            <reliability kind="BEST_EFFORT" maxBlockingTime="1000"/>
149            <resourceLimits maxInstances="1" maxSamples="1" maxSamplesPerInstance="1"/>
150            <transportPriority value="1"/>
151            <userData value=""/>
152            <writerDataLifecycle autodisposeUnregisteredInstances="true"/>
153          </dataWriterQoS>
154          <!--
155      Specifying the default values for QoS policy over DataReader
156      deadline: Set a maximum time in which an update must be published
157      destinationOrder: How data is ordered on the receiving side
158      userData: User Data not known by the middleware, but distributed by means of built-in topics
159      durability: This policy expresses if the data should 'outlive' their writing time,
160           it can be VOLATILE, TRANSIENT_LOCAL_TRANSCIENT or PERSISTENT
161      history: Buffering of data values before they are sent
162           KEEP_ALL tries to not lose any values
163           KEEP_LAST keeps up to "depth" values
164      latencyBudget: This is a HINT.Maximum duration between the data being written and data recevied by        ↙
     subscribing application
165      liveliness: Method used to maintain liveliness status. Length of time with no liveliness updates means ↙
      the entity is dead
166              Decreassing the leaseDuration shortens the time to detect the entity has gone away.
167      reliability: Reliablity of the connection. BEST_EFFORT means values may be dropped or lost
168      resourceLimits: Limits amount of resources to be used to meet other QoS
169      readerDataLifecycle: Delay until automatically removing all samples (even ones not read/taken) for       ↙
     Instances
170                  whose state is no writer or disposed
171      timeBasedFilter: DataReader is only interested in published data that is minimum_separation apart
172      -->
173          <dataReaderQoS name="defaultDataReaderQoS">
174            <deadline period="0"/>
175            <destinationOrder  kind="BY_SOURCE_TIMESTAMP"/>
176            <durability  kind="TRANSIENT"/>
177            <history kind="KEEP_LAST" depth="1"/>
178            <latencyBudget duration="100"/>
179            <liveliness kind="AUTOMATIC" leaseDuration="1"/>
180            <ownership kind="SHARED"/>
181            <reliability kind="BEST_EFFORT" maxBlockingTime="1000"/>
182            <resourceLimits maxInstances="1" maxSamples="1" maxSamplesPerInstance="1"/>
183            <readerDataLifecycle autopurgeDisposedSamplesDelay="1000" autopurgeNowriterSamplesDelay="1000"/>
184            <timeBasedFilter minimumSeparation="1000"/>
185            <userData value=""/>
186          </dataReaderQoS>
187        </qoSProfileDef>
188      </qoSProfiles>
189  </DDS>
```

```
190   <RTPS xmlns="urn:Configuration">
191     <!--
192     This is just a minimal sample configuration file that shows how to declare
193     the configuration sections.
194
195     Because an XML Schema Definition (XSD) is generated for each configuration
196     section, it should be trivial to edit these files because you have
197     IntelliSense on the XML definition.
198     -->
199     <transports>
200       <transport name="defaultRtps" type="Doopec.Rtps.RtpsTransport.RtpsEngine, Doopec">
201         <ttl val="1"/>
202         <discovery name="defaultDiscovery">
203           <resendPeriod val="30000"/>
204           <useSedpMulticast val="true"/>
205           <portBase val="9000"/>
206           <domainGain val="250"/>
207           <participantGain val="2"/>
208           <offsetMetatrafficMulticast val="0"/>
209           <offsetMetatrafficUnicast val="10"/>
210           <metatrafficUnicastLocatorList val="localhost"/>
211           <metatrafficMulticastLocatorList val="239.255.0.1"/>
212
213         </discovery>
214         <rtpsWriter>
215           <heartbeatPeriod val="1000"/>
216           <nackResponseDelay val="200"/>
217           <nackSuppressionDuration val="0"/>
218           <pushMode val="true"/>
219
220         </rtpsWriter>
221         <rtpsReader>
222           <heartbeatResponseDelay val="500"/>
223           <heartbeatSuppressionDuration val="0"/>
224         </rtpsReader>
225       </transport>
226     </transports>
227   </RTPS>
228   <appSettings>
229     <add key="org.omg.dds.serviceClassName" value="Doopec.Dds.Core.BootstrapImpl, Doopec" />
230   </appSettings>
231   <startup>
232     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.1" />
233   </startup>
234 </configuration>
```