**DESCRIPCION ARCHIVO DE CONFIGURACIÓN**

**ESCUELA POLITÉCNICA NACIONAL**

**PROYECTO: MIDDLEWARE DE TIEMPO REAL BASADO EN EL MODELO PUBLICADOR/SUSCRIPTOR**

Índice

[1. Introducción 4](#_Toc420259561)

[2. Sección DDS 4](#_Toc420259562)

[2.1. Etiqueta DDS 4](#_Toc420259563)

[2.1.1. Etiqueta Boostrap 4](#_Toc420259564)

[2.1.2. Etiqueta Domains 5](#_Toc420259565)

[2.1.2.1. Etiqueta Transport Profile 5](#_Toc420259566)

[2.1.2.2. Etiqueta QoS Profile 5](#_Toc420259567)

[2.1.2.3. Etiqueta Guid 5](#_Toc420259568)

[2.1.3. Etiqueta logLevel 6](#_Toc420259569)

[2.1.4. Etiqueta QoS Profiles 6](#_Toc420259570)

[2.1.4.1. Etiqueta domainParticipantFactoryQos 7](#_Toc420259571)

[2.1.4.2. Etiqueta domainParticipantQos 7](#_Toc420259572)

[2.1.4.3. Etiqueta topicQos 7](#_Toc420259573)

[2.1.4.4. Etiqueta publisherQos 8](#_Toc420259574)

[2.1.4.5. Etiqueta suscriberQos 8](#_Toc420259575)

[2.1.4.6. Etiqueta dataWriterQos 8](#_Toc420259576)

[2.1.4.7. Etiqueta dataReaderQos 9](#_Toc420259577)

[3. Sección RTPS 9](#_Toc420259578)

[3.1. Etiqueta RTPS 9](#_Toc420259579)

[3.1.1. Etiqueta Transports 10](#_Toc420259580)

[3.1.1.1. Etiqueta Discovery 10](#_Toc420259581)

[3.1.1.2. Etiqueta rtpsWriters 10](#_Toc420259582)

[1.1.1.1. Etiqueta rtpsReader 11](#_Toc420259583)

# Introducción

El siguiente documento expondrá el archivo de configuración que se encontrara presente en las implementación DDS/RTPS, la cual está siendo desarrollada.

Primeramente se presentara la configuración de la sección DDS, y seguidamente la sección RTPS; dentro de cada uno de estas secciones, se describirá en detalle cada uno de los elementos que deberán ser configurados con cada una de sus posibles valores.

<configuration>

<configSections>

<section name="DDS" type="Doopec.Configuration.DDSConfigurationSection, Doopec, Version=0.1.\*, Culture=neutral, PublicKeyToken=null"/>

<section name="RTPS" type="Doopec.Configuration.RTPSConfigurationSection, Doopec, Version=0.1.\*, Culture=neutral, PublicKeyToken=null"/>

</configSections>

# Sección DDS

# Etiqueta DDS

<DDS xmlns="urn:Configuration" vendor="Doopec" version="2.1">

Como se observa la etiqueta DDS, necesitara tener configurado tanto el xmlns seteado tal cual se puede observar con el valor urn:Configuration, seguidamente se debe especificar el vendor, que en nuestro caso es Doopec y finalmente la versión.

# Etiqueta Boostrap

Como se observa la etiqueta BoostrapType, donde se configura el name con default, y de type, con lo que se muestra en el código

<boostrapType name="default" type="Doopec.Dds.Core.BootstrapImpl, Doopec"/>

# Etiqueta Domains

Como se observa la etiqueta Domains, anuncia a los participantes presentes y con los cuales se podrá interactuar por medio de RTPS, en este caso podemos observar a 3 domain configurados, cada uno de estos tiene dentro de su etiqueta **domain**, un name, es decir un nombre indistinto, y un **id**, para la identificación única del participante

# Etiqueta Transport Profile

Esta etiqueta definirá el perfil de transporte de información, con el cual trabajará DDS, para nuestro caso será defaultRTPS. Además al decir defaultRTPS se hace referencia a la configuración que más adelante será explicada

# Etiqueta QoS Profile

Esta etiqueta definirá el perfil de calidad de servicio, con el cual trabajará DDS, para nuestro caso será defaultQoS. Además al decir defaultQoS se hace referencia a la configuración de calidad de servicio por defecto que será explicada mas adelante

# Etiqueta Guid

Para la etiqueta Guid, se tendrán 3 posibles valores a tomar.

* Fixed, en el cual se deberá definir de forma manual el valor
* Random
* AutoId
* AutoIdFromIp
* AutoIdFromMac

<domains>

<domain name="Servidor" id="0">

<transportProfile name="defaultRtps"/>

<qoSProfile name="defaultQoS"/>

<guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA061C"/>

</domain>

<domain name="Servidor" id="3">

<transportProfile name="defaultRtps"/>

<qoSProfile name="defaultQoS"/>

<guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA061C"/>

</domain>

<domain name="Cliente1" id="1">

<transportProfile name="defaultRtps"/>

<qoSProfile name="defaultQoS"/>

<guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA061C"/>

</domain>

<domain name="Cliente2" id="2">

<transportProfile name="defaultRtps"/>

<qoSProfile name="defaultQoS"/>

<guid kind="Fixed" val="7F294ABE-33F2-40B9-BFF5-7D9376EA061C"/>

</domain>

</domains>

# Etiqueta logLevel

Como se observa la etiqueta logLevel, definirá un nivel mínimo y un máximo de logs con los siguientes posibles valores

* DEBUG
* ALL
* WARN
* INFO
* ERROR
* FATAL
* OFF

<logLevel levelMin="DEBUG" levelMax="FATAL"/>

# Etiqueta QoS Profiles

QoS Profiles, está compuesto, de un QosProfileDef, el cual tiene un domain Participant Factory, un domainParticipanQoS, un topicQoS, un publisherQoS, un suscriberQoS, un dataWriterQos, y un dataReaderQoS.

# Etiqueta domainParticipantFactoryQos

Además de asignar un nombre a esta etiqueta, tiene una etiqueta interna autoenableCreatedEntities, la cual define si se auto habilitan a los participantes creados o no.

<qoSProfiles>

<qoSProfileDef name="defaultQoS">

<domainParticipantFactoryQos name="defaultDomainParticipantFactoryQoS">

<entityFactory autoenableCreatedEntities="true"/>

</domainParticipantFactoryQos>

# Etiqueta domainParticipantQos

Ademas de asignar un nombre a esta etiqueta, tiene una etiqueta interna autoenableCreatedEntity, la cual define si se auto habilitan a los participantes creados o no, además de una etiqueta userData, en la cual se puede poner valores de acuerdo a la política userDataQoS policy.

<domainParticipantQos name="defaultDomainParticipantQoS">

<entityFactory autoenableCreatedEntities="true"/>

<userData value=""/>

</domainParticipantQos>

# Etiqueta topicQos

Además de asignar un nombre a esta etiqueta, tiene una etiqueta interna topicData, a la cual se le puede asignar un valor, aparte se puede agregar varias políticas de QoS, como en este caso deadline y durability.

<topicQoS name="defaultTopicQoS">

<topicData value=""/>

<deadline period="100"/>

<durability kind="VOLATILE"/>

</topicQoS>

# Etiqueta publisherQos

Además de asignar un nombre a esta etiqueta, tiene una etiqueta interna entityFactory, a la cual se le puede asignar un valor, aparte se puede agregar varias políticas de QoS, como en este caso groupData, partition, y presentation.

<publisherQoS name="defaultPublisherQoS">

<entityFactory autoenableCreatedEntities="true"/>

<groupData value=""/>

<partition value=""/>

<presentation accessScope="INSTANCE" coherentAccess="true" orderedAccess="true"/>

</publisherQoS>

# Etiqueta suscriberQos

Además de asignar un nombre a esta etiqueta, tiene una etiqueta interna entityFactory, a la cual se le puede asignar un valor, aparte se puede agregar varias políticas de QoS, como en este caso groupData, partition, y presentation.

<subscriberQoS name="defaultSubscriberQoS">

<entityFactory autoenableCreatedEntities="true"/>

<groupData value=""/>

<partition value=""/>

<presentation accessScope="INSTANCE" coherentAccess="true" orderedAccess="true"/>

</subscriberQoS>

# Etiqueta dataWriterQos

Además de asignar un nombre a esta etiqueta, se puede agregar varias políticas de QoS, como en este caso todas las políticas que pueden ser utilizadas en un escritor.

<dataWriterQoS name="defaultDataWriterQoS">

<deadline period="1"/>

<destinationOrder kind="BY\_SOURCE\_TIMESTAMP"/>

<durability kind="VOLATILE"/>

<durabilityService historyDepth="0" historyKind="KEEP\_LAST" maxInstances="1" maxSamples="1" maxSamplesPerInstance="1" serviceCleanupDelay="100"/>

<history kind="KEEP\_LAST" depth="1"/>

<latencyBudget duration="100"/>

<lifespan duration="100"/>

<liveliness kind="AUTOMATIC" leaseDuration="100"/>

<ownership kind="SHARED"/>

<ownershipStrength value="100"/>

<reliability kind="BEST\_EFFORT" maxBlockingTime="1000"/>

<resourceLimits maxInstances="1" maxSamples="1" maxSamplesPerInstance="1"/>

<transportPriority value="1"/>

<userData value=""/>

<writerDataLifecycle autodisposeUnregisteredInstances="true"/>

</dataWriterQoS>

# Etiqueta dataReaderQos

Además de asignar un nombre a esta etiqueta, se puede agregar varias políticas de QoS, como en este caso todas las políticas que pueden ser utilizadas en un lector.

<dataReaderQoS name="defaultDataReaderQoS">

<deadline period="1"/>

<destinationOrder kind="BY\_SOURCE\_TIMESTAMP"/>

<durability kind="VOLATILE"/>

<history kind="KEEP\_LAST" depth="1"/>

<latencyBudget duration="100"/>

<liveliness kind="AUTOMATIC" leaseDuration="100"/>

<ownership kind="SHARED"/>

<reliability kind="BEST\_EFFORT" maxBlockingTime="1000"/>

<resourceLimits maxInstances="1" maxSamples="1" maxSamplesPerInstance="1"/>

<readerDataLifecycle autopurgeDisposedSamplesDelay="1000" autopurgeNowriterSamplesDelay="1000"/>

<timeBasedFilter minimumSeparation="1000"/>

<userData value=""/>

</dataReaderQoS>

</qoSProfileDef>

</qoSProfiles>

</DDS>

# Sección RTPS

# Etiqueta RTPS

<RTPS xmlns="urn:Configuration">

Como se observa la etiqueta DDS, necesitara tener configurado tanto el xmlns seteado tal cual se puede observar con el valor urn:Configuration.

# Etiqueta Transports

La etiqueta transport trae consigo un nombre y un type, el cual especifica el Motor RTPS, además hay una etiqueta ttl, la cual cumple las funciones de time to live

<transports>

<transport name="defaultRtps" type="Doopec.Rtps.RtpsTransport.RtpsEngine, Doopec">

<ttl val="1"/>

# Etiqueta Discovery

La etiqueta Discovery introduce la configuración de los paquetes de descubrimiento, aquí podremos configurar el **periodo de reenvío**, definiremos si se usan **paquetes multicast** del tipo Sedp, el **puerto base** con el que se trabajará, el **domainGain y el participantGain**; también se configurara offsets tanto para **trafico unicast y multicast**, y se definirán lista de IP con las que se trabajara tanto en **modo multicast y modo unicast**.

<discovery name="defaultDiscovery">

<resendPeriod val="30000"/>

<useSedpMulticast val="true"/>

<portBase val="7400"/>

<domainGain val="250"/>

<participantGain val="2"/>

<offsetMetatrafficMulticast val="0"/>

<offsetMetatrafficUnicast val="10"/>

<metatrafficUnicastLocatorList val="localhost"/>

<metatrafficMulticastLocatorList val="239.255.0.1"/>

</discovery>

# Etiqueta rtpsWriters

La etiqueta rtpsWriter introduce la configuración de los writer rtps y su comportamiento, aquí podremos configurar el **periodo de envío de submensajes Heartbeat**, definiremos los **retardos de las respuestas por medio de NACK**, también el tiempo para que se suprima una respuesta NACK, y si estamos trabajando con modo **push**.

<rtpsWriter>

<heartbeatPeriod val="1000"/>

<nackResponseDelay val="200"/>

<nackSuppressionDuration val="0"/>

<pushMode val="true"/>

</rtpsWriter>

# Etiqueta rtpsReader

La etiqueta rtpsReader introduce la configuración de los reader rtps y su comportamiento, definiremos los **retardos de las respuestas por medio de Heartbeats**, también el tiempo para que se suprima una respuesta Heartbeat.

<rtpsReader>

<heartbeatResponseDelay val="500"/>

<heartbeatSuppressionDuration val="0"/>

</rtpsReader>

</transport>

</transports>

</RTPS>

<appSettings>

<add key="org.omg.dds.serviceClassName" value="Doopec.Dds.Core.BootstrapImpl, Doopec" />

</appSettings>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.1" />

</startup>

</configuration>