# Diagramas de interacción del protocolo RTPS

# Diagramas de interaccion con Estado

## Diagramas basados con la QoS Best Effort

### Best Effort Reader – Best Effort Writer con Key

1. El usuario DDS escribe datos por medio de la llamada a la operación *write* en el *DataWriter* DDS.
2. El *DataWriter* DDS llama a la operación *new\_change* en el *Writer* RTPS para crear un nuevo *CacheChange*. Cada uno de estos cambios es identificado únicamente por un *SequenceNumber*.
3. La operación *new\_change* retorna.
4. El *DataWriter* DDS utiliza la operación *add\_change* para almacenar el *CacheChange* dentro de *HistoryCache* del *Writer* RTPS.
5. El *HistoryCache* del *Writer* RTPS notifica el cambio por medio de la operación *notify\_change*  al *Publisher* DDS.
6. La operación *notify\_change*  retorna.
7. La operación *add\_change­* retorna.
8. La operación *write* retorna. El usuario ha completado la acción de escritura de datos.
9. El *HistoryCache*  del *Writer* DDS utiliza la operación *unsent\_changes* para informar al *ReaderProxy* que hay cambios o información no enviada.
10. La operación *unsent\_changes* retorna.
11. El *HistoryCache*  del *Writer* DDS utiliza la operación *can\_send* para informar al *ReaderProxy* que puede enviar los cambios.
12. La operación *can\_send* retorna.
13. El *DataWriter* DDS utiliza la operación *remove\_change*  en el *HistoryCache*  del *Writer* DDS para limpiar la cache. Esta operación puede ser realizada posteriormente.
14. La operación *remove\_change* retorna.
15. El *ReaderProxy* serializa la información mediante la operación *serialize* en el *Serializer*.
16. La operación *serialize*  retorna.
17. El *ReaderProxy*  envía el submensaje DATA al *MessageEncoder* para que sea encapsulado.
18. El *MessageEncoder* encapsula el mensaje y lo envía mediante la operación *encoded\_message­* al *UDPTransmitter*.
19. La operación *encoded­\_message retorna.*
20. El *UDPTransmitter* envía el mensaje *UDP\_Message* hacia la red de Datos.
21. El *UDPReceiver* recibe el mensaje *UDP\_Message* desde la red de Datos.
22. El *UDPReceiver* desencapsula el mensaje mediante la operación *doDecode*  en el *MessageDecoder*
23. La operación *doDecode* retorna.
24. El *MessageDecoder* envía el submensaje DATA al *WriterProxy*.
25. El *WriterProxy* llama a la operación *deserialize\_data* al *Deserializer*
26. La operación *deserialize\_data* retorna.
27. El *WriterProxy* llama a la operación *available\_change* dentro del *HistoryCache* del *Reader* RTPS, para la verificación de números de secuencia recibidos.
28. La operación *available\_change* retorna.
29. El *WriterProxy* genera un nuevo *CacheChange* con la operación *new\_change*  en el *Reader* RTPS.
30. La operación *new\_change* retorna.

### Best Effort Reader – BestEffort Writer sin Key

# Diagramas de interacción sin estado