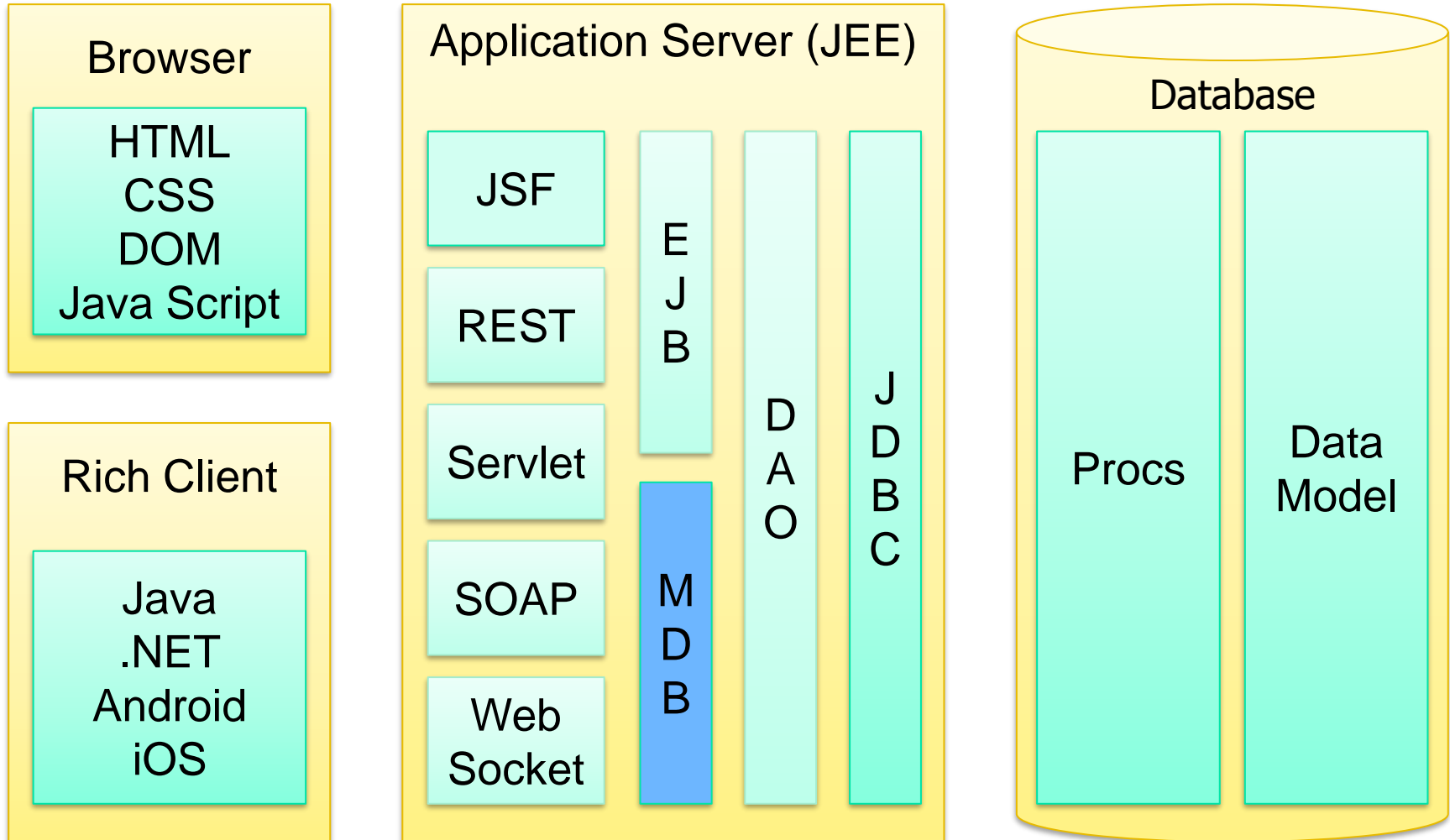




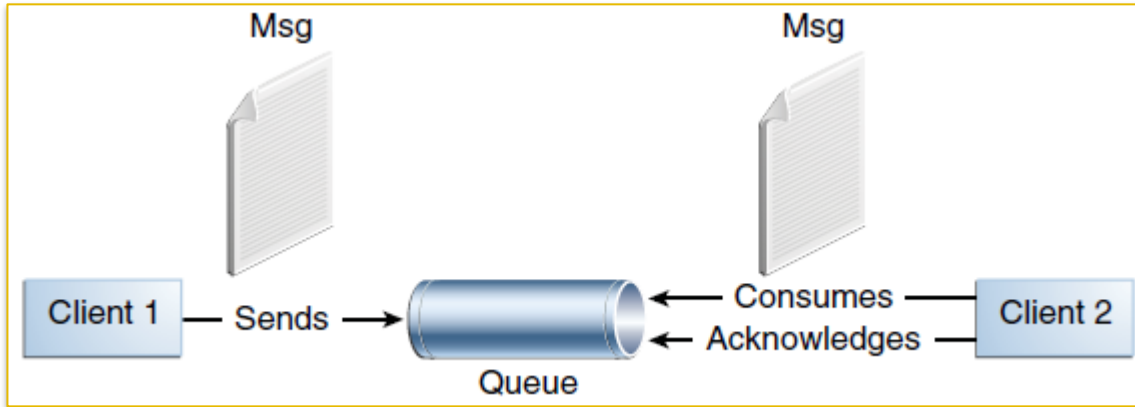
Aplicaciones Web (parte 6)

Eduardo Ostertag Jenkins, Ph.D.
OBCOM INGENIERIA S.A. (Chile)
Eduardo.Ostertag@obcom.cl

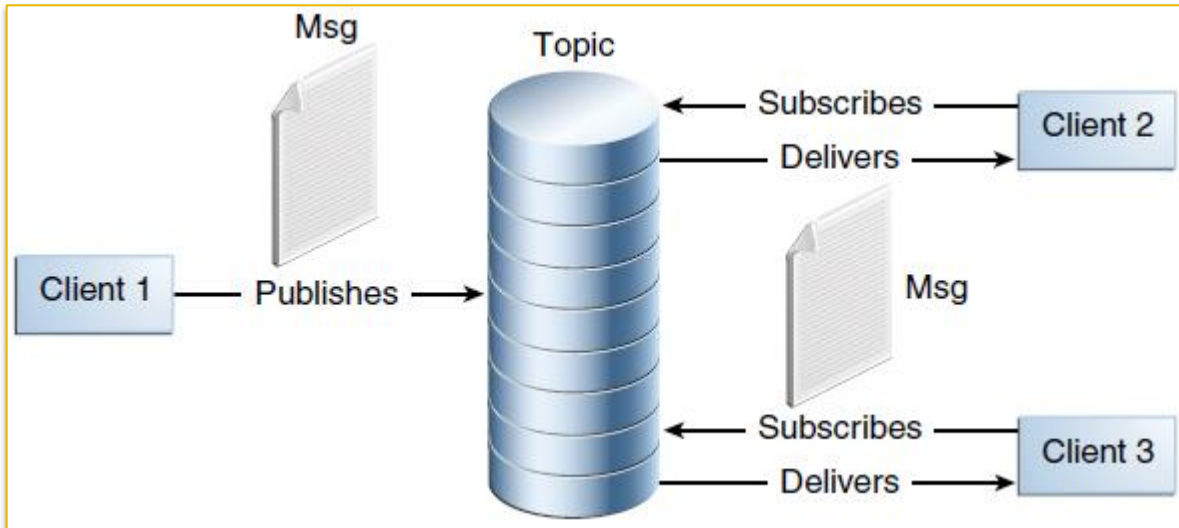
Message-Driven Beans: MDB



Message-Driven Beans (MDB)



Point-To-Point Queue

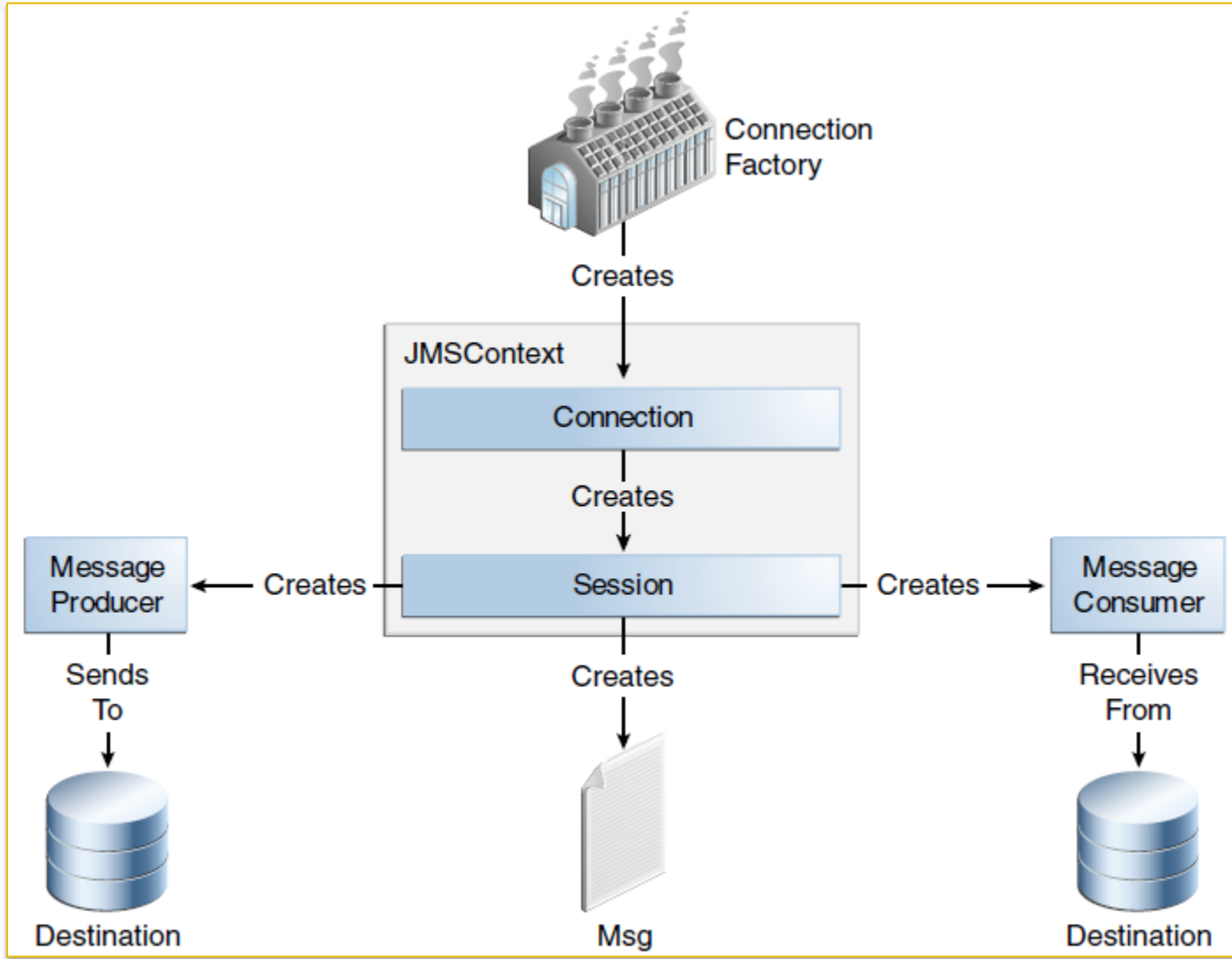


Publish/Subscribe Topic

Objetos JMS principales (1)

- Connection Factory
- Destination (Queue, Topic)
- Connection
- Session
- JMSContext (Connection + Session)
- Message Producer
- Message Consumer
- Message

Objetos JMS principales (2)



Clase Message-Driven Bean

- Asynchronous **Message Consumer**
- Decorada con `@javax.ejb.MessageDriven`
- Implementa `javax.jms.MessageListener`

```
@MessageDriven
public class SampleMdb implements MessageListener
{
    @Override
    public void onMessage(Message message)
    {
        ...
    }
}
```

Tipos de mensajes JMS

- **TextMessage**: texto (String)
- **MapMessage**: tabla nombre→valor (Map)
- **BytesMessage**: array de bytes (byte[])
- **StreamMessage**: stream objetos (Stream)
- **ObjectMessage**: objetos serializables 😊

Encabezados de mensajes

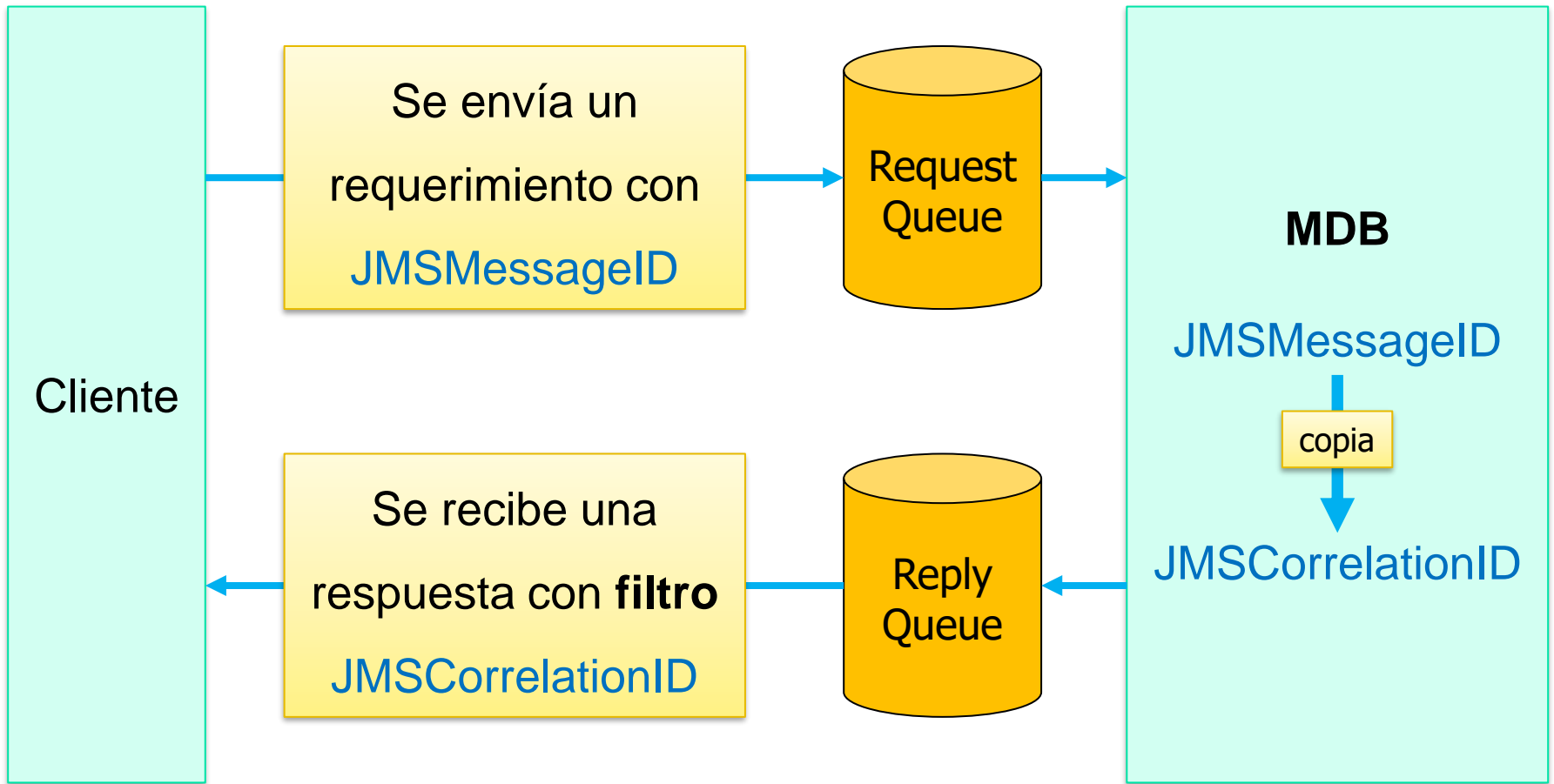
- **JMSDestination**: destino
- **JMSDeliveryMode**: persistente, no persistente
- **JMSDeliveryTime**: tiempo de entrega
- **JMSExpiration**: tiempo de expiración
- **JMSPriority**: prioridad
- **JMSMessageID**: identificación única
- **JMSTimestamp**: tiempo de envío
- **JMSCorrelationID**: asociación envío ↔ respuesta
- **JMSReplyTo**: destino de la respuesta
- **JMSType**: tipo suministrado por usuario
- **JMSRedelivered**: indica si reenviado

Selectores de mensajes

- Por defecto, los mensajes se consumen desde una cola en el orden en que fueron agregados (FIFO)
- Se puede consumir sólo los mensajes que cumplen una condición, ignorando los otros mensajes, pero respetando el orden de llegada
- Para esto, se usan **selectores** JMS:

```
String selector = "JMSScOlletationID='...' AND JMSType='...'";  
JMSConsumer consumer = context.createConsumer(queue, selector);  
Message message = consumer.receive();
```

Point-To-Point, Request-Reply



Enviamos mensaje al MDB

```
public String sendRequest(Serializable request)
    throws Exception
{
    try (JMSContext context = connectionFactory.createContext()) {
        ObjectMessage reqmsg = context.createObjectMessage(request);
        context.createProducer()
            .setJMSReplyTo(replyQueue)
            .send(requestQueue, reqmsg);
        return reqmsg.getJMSMessageID();
    }
}
```

Procesamos mensaje en MDB

```
@Override
public void onMessage(Message message)
{
    try {
        ObjectMessage reqmsg = (ObjectMessage) message;
        Serializable result = executeRequest(reqmsg.getObject());
        try (JMSContext context = connectionFactory.createContext()) {
            Destination replyTo = reqmsg.getJMSReplyTo();
            context.createProducer()
                .setJMSCorrelationID(reqmsg.getJMSMessageID())
                .send(replyTo, context.createObjectMessage(result));
        }
    } catch (Throwable thrown) {
        logger.log(Level.SEVERE, "Error processing message", thrown);
    }
}
```

Recibimos respuesta del MDB

```
public Serializable receiveReply(String correlID, long timeout)
    throws Exception
{
    try (JMSContext context = connectionFactory.createContext()) {
        String selector = "JMSCorrelationID='" + correlID + "'";
        try (JMSConsumer consumer = context.createConsumer(replyQueue, selector)) {
            return consumer.receive(timeout);    // returns null if timeout
        }
    }
}
```

Crear colas en Payara

The screenshot shows the Payara Server Management Console interface. The browser address bar indicates the URL is `ostertag7520:4848/common/index.jsf`. The user is logged in as 'admin' on the 'production' domain, connected to the server 'ostertag7520'. The main navigation menu on the left includes 'Applications', 'Lifecycle Modules', 'Monitoring Data', 'Resources', 'JMS Resources', and 'Configurations'. The 'Resources' section is expanded to show 'Destination Resources', which is highlighted in orange. Below the navigation, the 'JMS Destination Resources' page is displayed. It features a title 'JMS Destination Resources' and a brief description: 'JMS destinations serve as the repositories for messages. Click New to create a new destination resource. Click the name of a destination resource to modify its properties.' Below this, there is a table titled 'Destination Resources (2)'. The table has columns for 'Select', 'JNDI Name', 'Enabled', 'Resource Type', and 'Description'. Two resources are listed: 'jms/SampleRequestQueue' and 'jms/SampleReplyQueue', both of which are enabled and of type 'javax.jms.Queue'. A red box highlights these two rows in the table.

User: admin Domain: production Server: ostertag7520

Home About... Logout Help Online Help Enable Asadmin Recorder

payara[®]
SERVER

Applications
Lifecycle Modules
Monitoring Data
Resources
Concurrent Resources
Connectors
JDBC
JMS Resources
Connection Factories
Destination Resources
jms/SampleRequestQueue
jms/SampleReplyQueue
JNDI
JavaMail Sessions
Resource Adapter Configs
Configurations
default-config

JMS Destination Resources

JMS destinations serve as the repositories for messages. Click New to create a new destination resource. Click the name of a destination resource to modify its properties.

Destination Resources (2)

New... Delete Enable Disable

Select	JNDI Name	Enabled	Resource Type	Description
<input type="checkbox"/>	jms/SampleRequestQueue	✓	javax.jms.Queue	
<input type="checkbox"/>	jms/SampleReplyQueue	✓	javax.jms.Queue	

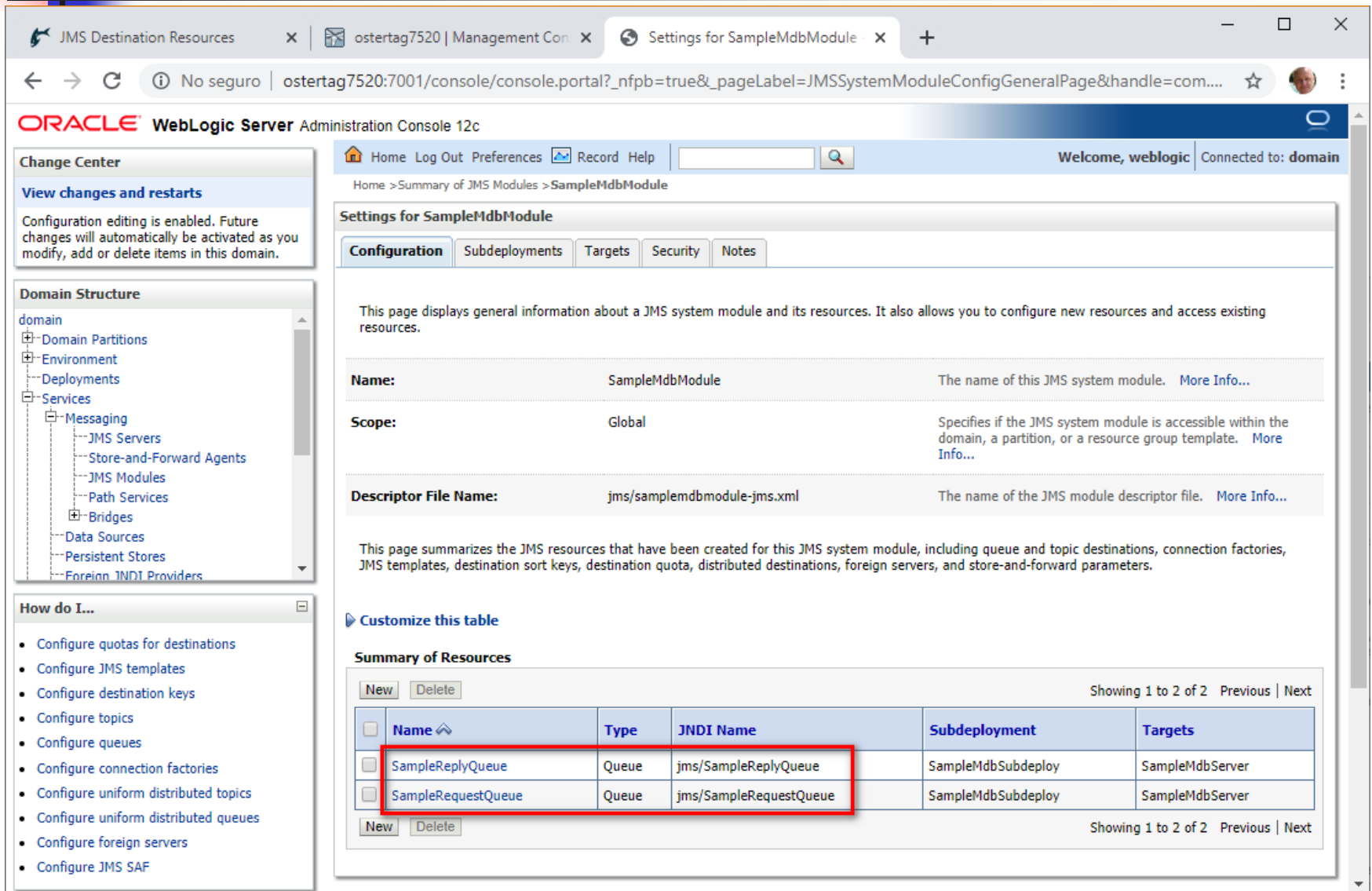
Crear colas en Wildfly

The screenshot shows the Wildfly Management Console interface. The browser address bar indicates the URL: `ostertag7520:10090/console/index.html#messaging-server-destination;server=default`. The page title is "JMS Queue" and it describes the function: "Defines a JMS queue." A search bar is present at the top of the table, and it shows "Showing 1 to 4 of 4 Items". The table lists the following queues:

Name ^	Selector
DLQ	
ExpiryQueue	
SampleReplyQueue	
SampleRequestQueue	

The "SampleReplyQueue" and "SampleRequestQueue" rows are highlighted with a red border. Below the table, there is a "Help" section with a "Selector" label, an "Entries" section, and a "Durable" property set to `true`. The sidebar on the left contains navigation options: Core Queue, JMS Queue (selected), JMS Topic, Security Setting, Address Setting, and Divert. The bottom of the page shows version information (3.2.1.Final), Tools, and Settings.

Crear colas en WebLogic



The screenshot shows the Oracle WebLogic Server Administration Console 12c interface. The browser address bar indicates the URL: `ostertag7520:7001/console/console.portal?_nfpb=true&_pageLabel=JMSSystemModuleConfigGeneralPage&handle=com...`. The page title is "Settings for SampleMdbModule".

The main content area is titled "Settings for SampleMdbModule" and includes tabs for "Configuration", "Subdeployments", "Targets", "Security", and "Notes". The "Configuration" tab is active, showing general information about the JMS system module:

- Name:** SampleMdbModule
- Scope:** Global
- Descriptor File Name:** jms/samplemdbmodule-jms.xml

Below this information is a "Summary of Resources" section, which contains a table of JMS resources. The table has columns for Name, Type, JNDI Name, Subdeployment, and Targets. Two queues are listed:

Name	Type	JNDI Name	Subdeployment	Targets
SampleReplyQueue	Queue	.jms/SampleReplyQueue	SampleMdbSubdeploy	SampleMdbServer
SampleRequestQueue	Queue	.jms/SampleRequestQueue	SampleMdbSubdeploy	SampleMdbServer

The "SampleReplyQueue" and "SampleRequestQueue" rows are highlighted with a red box. The console also includes a "Change Center" on the left, a "Domain Structure" tree, and a "How do I..." section with various configuration links.



OBCOM

Muchas gracias

Muchas

gracias