**Project: solicitudServicio**

Responsible: A good fella

Create and implement: IPTVServicio.xsd

    request: servicioRequest

    response: subscripcionResponse

Convert IPTVServicio.xsd to Java code with Xmlbeansgenerator

Generate IPTVServicio.jar

Export to IPTVServicio project

**Project: IPTVServicio**

Responsible: A good fella

Create ConsumerSpringListener

Create JmsSender

Implement ConsumerSpringListener to read from queue B

Queue B will recive iptvServicioRequest request

Parse message received to iptvServicioRequestDocument

Validate if message received is actually an iptvServicioRequestDocument object

Get iptvServicioRequest from iptvServicioRequestDocument

Create a altaUsuarioRequest and set it with the required elements.

Encapsulate it into altaUsuarioRequestDocument

Create a altaUsuarioRequestStub adding the URL of the AltaUsuarioProveedor web service

Get and store altaUsuarioResponseDocument

Validate if message received is actually a altaUsuarioResponseDocument

Get altaUsuarioResponse

If altaUsuarioResponse:registro is true, continue.

Else if altaUsuarioResponse:registro is false… The truth is that this can happen, but we won’t do anything, because it means even more work. We’re open to suggestions. Perhaps, we can send a message to queue C. Just a message that goes like: “Sorry, man, we’re having problems with our Proveedor right now, we will send you an email as soon it’s fixed”

Create a configUsuarioRequest and set it with the required elements.

Encapsulate it into configUsuarioRequestDocument

Create a configUsuarioRequestStub adding the URL of the configUsuarioProveedor service

Get and store configUsuarioResponseDocument

Validate if message received is actually a configUsuarioResponseDocument

Get configUsuarioResponseDocument

if configUsuarioRequest:config is true, continue.

Else if configUsuarioRequest:config is false… read again the red text.

Set subscripcionResponse:subscripcion to true and info to: “Subscripció aprobada, disfrute el servicio”

Send subscripcionResponseDocument to queue D.

END.

**Project: AltaUsuarioProveedor**

Responsible: A good fella

1) Create and implement AltaUsuarioProveedor.xml

request: altaUsuarioRequest

response: altaUsuarioResponse

Generate AltaUsuarioProveedor.jar to Java code with XMLBeans generator

Export to IPTV service project.

2) Create AltaUsuarioProveedor.wsdl

Generate a server with XmlBeansBinding

Implement AltaUsuarioImpl Class as needed:

    Consult user in Proveedor DB.

    if usuario is in Proveedor.usuarios

        set altaUsuarioResponse:registro to true

    else if usuario is not in Proveedor.usuarios

        insert into table usuarios a new usuario

        set altaUsuariosResponse:registro to true

Deploy server as AltaUsuarioProveedor.aar on Tomcat

3) Generate client code with xmlbeans binding

Compile client as AltaUsuarioProveedorClient.jar with XMLbeans generator

Export to IPTVServicio project

**Project: ConfigUsuarioProveedor**

responsible: A good fella

1) Create and implement ConfigUsuarioProveedor.xml

request: configUsuarioRequest

response: configUsuarioResponse

Generate ConfigUsuarioProveedor.jar to Java code with XMLBeans generator

Export to IPTV service project.

2) Create ConfigUsuarioProveedor.wsdl

Generate a server with XmlBeansBinding

Implement ConfigUsuarioImpl Class as needed:

    Insert into definicion\_usuarios correo and definicion

    Insert into canales\_usuarios correo and canales

        (insert as many values you need)

    set congigUsuarioResponse:config to true

    Deploy server as ConfigUsuarioProveedor.aar on Tomcat

3) Generate client code with xmlbeans binding

Compile client as CofigUsuarioProveedorClient.jar with XMLbeans generator

Export to IPTVServicio project