



Practica conocimientos SOA

Services & processes solutions de CV

Contenido

1.1	Caso de uso	1
1.2	Elastic Stack	1
1.2.1	Elastic Stack	1
1.2.2	Creación de un índice	6
1.3	Oracle SOA Suite 12.2.1.0.0	6
1.3.1	Create	18
1.3.2	Read	20
1.3.3	Update	22
1.3.4	Delete	24
1.4	Busqueda sobre los índices	26
1.4.1	Busca los últimos 20 documentos	26
1.4.2	Busca los documentos de un usuario en específico	27

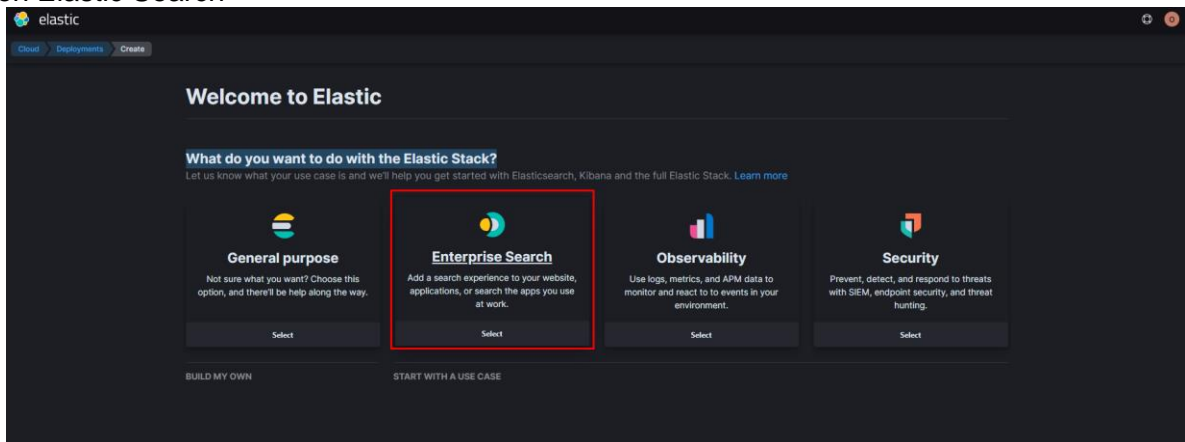
1.1 Caso de uso

La empresa “Automotriz SPS” no tiene el control de los cambios que realizan los usuarios de negocio a los parámetros establecidos para las reglas de descuento, por lo cual, requiere tener el rastreo de dichas actividades. El sistema de administración de reglas requiere un servicio de bitácora en el cual se pueda enviar dicha información, dicha solución se encuentra dentro de la red corporativa

1.2 Elastic Stack

1.2.1 Elastic Stack

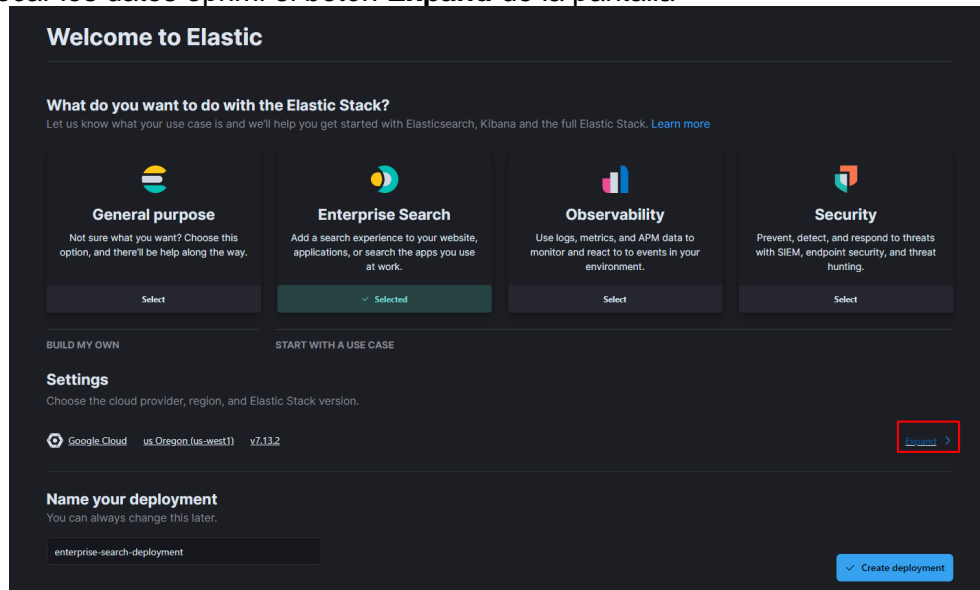
Lo primero hice es crear una cuenta en Elastic como versión de prueba, después elegí que es lo que quiero hacer con Elastic Search



Después coloque los datos propuestos en el documento **Práctica de Elastic 2021.pdf**, los cuales son:

- Nombre del deployment: sps_practica
- Plataforma: Amazon Web Service
- Region: US East (N. Virginia)
- Elastic Stack version: Mas reciente
- Optimize your deployment: I/O Optimized

Para poder colocar los datos oprimi el botón **Expand** de la pantalla



Una vez colocados los datos queda de la siguiente manera, en mi caso no encontré esta opción **Optimize your deployment: I/O Optimized**, seleccione el boton **Create deployment**.

The screenshot shows the 'Settings' page in the Elastic Cloud console. Under 'Cloud provider', the 'AWS' button is highlighted with a red box. Below it, the 'Region' dropdown is set to 'us N. Virginia (us-east-1)' and the 'Version' dropdown is set to '7.13.2', both also highlighted with red boxes. In the 'Name your deployment' section, the text 'sps_practica' is entered into the input field, which is also highlighted with a red box. A blue 'Create deployment' button is visible at the bottom right.

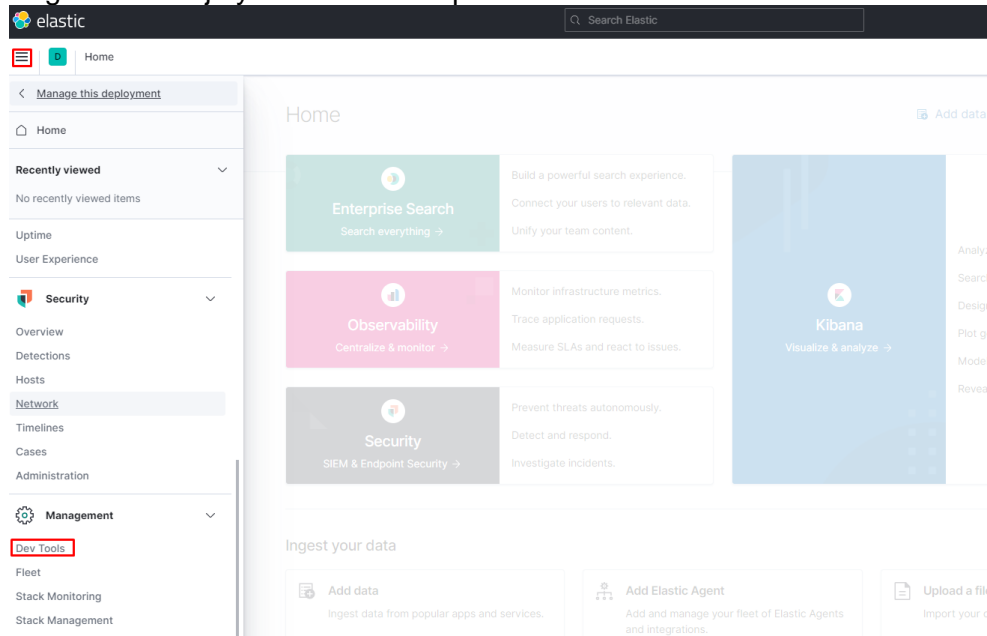
Esto tardara un poco para crear el despliegue. Descargue las credenciales para su almacenamiento.

The screenshot shows a modal dialog box titled 'Save your elastic deployment credentials. They are shown only once.' It contains a form with 'Username' set to 'elastic' and 'Password' set to a long alphanumeric string. A red box highlights the 'Download' button. Below the form, there is a link that says 'Continue without downloading'.

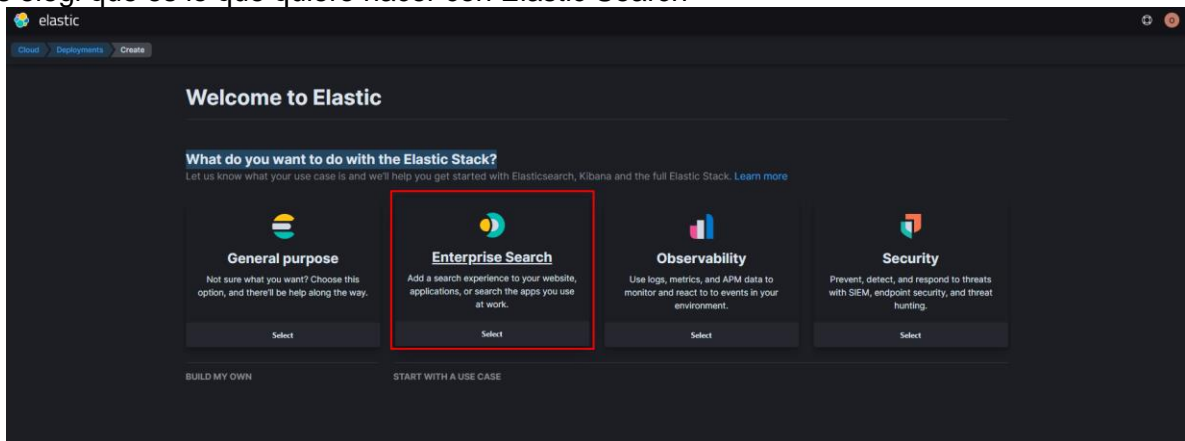
Una vez que termina esto seleccione la opción **Open Kibana**.

The screenshot shows the Elastic Cloud console after the deployment 'sps_practica' has been created. A green checkmark and the message 'Your deployment has been created.' are displayed. Below this, there is a section 'Get started with your deployment' with a button 'Open Enterprise Search'. At the bottom, the deployment details are shown, including the name 'sps_practica', ID '0f2a940', and version 'v7.13.2'. A red box highlights the 'Open Kibana' button in the bottom right corner.

Me abrió la siguiente pantalla en la cual seleccione el menú icono de las 3 líneas en la parte superior izquierda y me dirigi hasta abajo y seleccione la opción de **Dev Tools**.



En la pantalla visualice lo siguiente Lo primero hice es crear una cuenta en Elastic como versión de prueba, después elegi que es lo que quiero hacer con Elastic Search



Después coloque los datos propuestos en el documento **Práctica de Elastic 2021.pdf**, los cuales son:

- Nombre del deployment: sps_practica
- Plataforma: Amazon Web Service
- Region: US East (N. Virginia)
- Elastic Stack version: Mas reciente
- Optimize your deployment: I/O Optimized

Para poder colocar los datos oprimi el botón **Expand** de la pantalla

Welcome to Elastic

What do you want to do with the Elastic Stack?
Let us know what your use case is and we'll help you get started with Elasticsearch, Kibana and the full Elastic Stack. [Learn more](#)

General purpose
Not sure what you want? Choose this option, and there'll be help along the way.

Select

Enterprise Search
Add a search experience to your website, applications, or search the apps you use at work.

✓ Selected

Observability
Use logs, metrics, and APM data to monitor and react to events in your environment.

Select

Security
Prevent, detect, and respond to threats with SIEM, endpoint security, and threat hunting.

Select

BUILD MY OWN START WITH A USE CASE

Settings

Choose the cloud provider, region, and Elastic Stack version.

Google Cloud
 us Oregon (us-west1)
 v7.13.2
 Expand >

Name your deployment

You can always change this later.

enterprise-search-deployment

✓ Create deployment

Una vez colocados los datos queda de la siguiente manera, en mi caso no encontré esta opción **Optimize your deployment: I/O Optimized**, seleccione el boton **Create deployment**.

Settings

Choose the cloud provider, region, and Elastic Stack version.

Cloud provider
Pick a cloud and let us handle the rest. No additional accounts required.

Google Cloud
 Azure
 Amazon Web Services

Region
us N. Virginia (us-east-1)

Version
7.13.2

Name your deployment

You can always change this later.

sps_practica

✓ Create deployment

Esto tardara un poco para crear el despliegue. Descargue las credenciales para su almacenamiento.

sps_practica

Creating your deployment
Your deployment will be ready in a few minutes.

Get started with your deployment
The next step is to customize your deployment.

Forgot to save your credentials? You can download them now.

Save your **elastic** deployment credentials.
They are shown only once.

Username
elastic

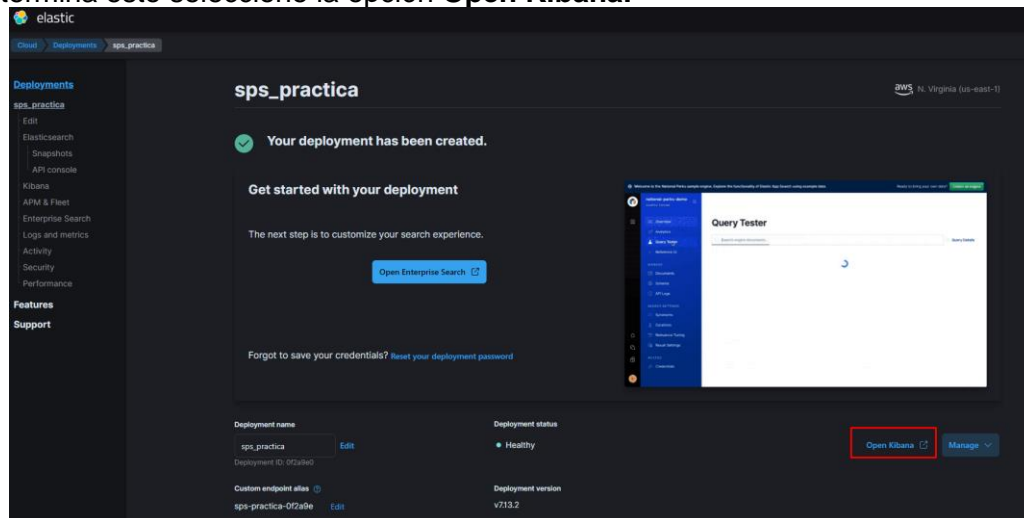
Password
Elasticsearch

Download

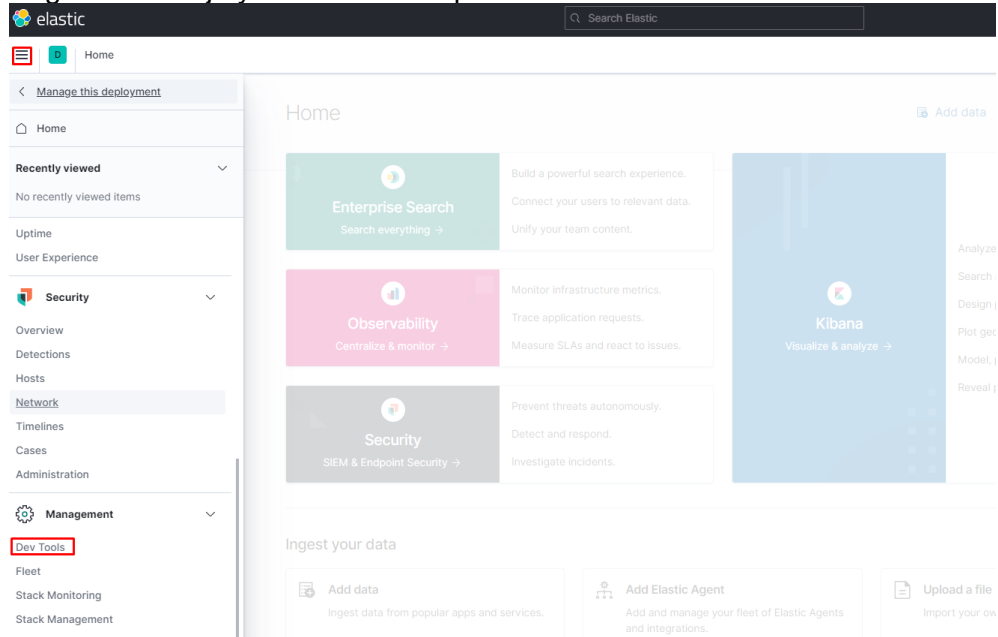
Continue without downloading

Query Tester

Una vez que termina esto seleccione la opción **Open Kibana**.



Me abrió la siguiente pantalla en la cual seleccione el menú icono de las 3 líneas en la parte superior izquierda y me dirigí hasta abajo y seleccione la opción de **Dev Tools**.



1.2.2 Creación de un índice

1.2.2.1 En tu menú de lado izquierdo tendrás las opciones con las que trabajaremos, por ahora revisa la opción Dev Tools, Management, Visualize y Dashboard.

1.2.2.2 Haciendo uso del API REST define la estructura del índice (Parte de la evidencia de entregables).

Lo que a mi se me ocurrió es:

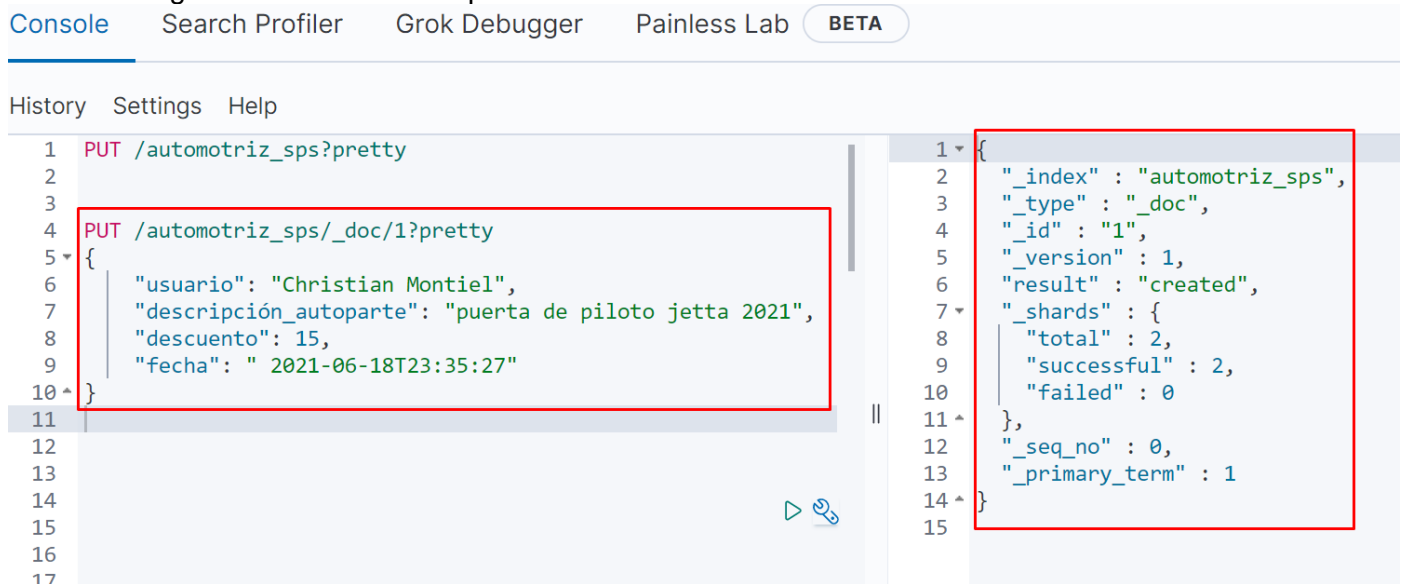
1. Nombre de usuario, persona que dará el descuento,
2. Descripción de la autoparte, breve descripción de la autoparte del carro.
3. Descuento, será el descuento que se le hará a la autoparte.
4. Fecha, fecha en la que se hizo el descuento.

Como nombre de el indice le coloque "automotriz_sps", el JSON quedo de la siguiente manera

```
{
  "usuario": "Christian Montiel",
  "descripción_autoparte": "puerta de piloto jetta 2021",
  "descuento": 15,
  "fecha": " 2021-06-18T23:35:27"
}
```

1.2.2.3 Haciendo uso del API REST crea un documento de prueba

Se creo el siguiente documento de prueba



The screenshot shows the DevTools interface with the REST client tab active. On the left, a PUT request is shown to the endpoint `/automotriz_sps/_doc/1?pretty` with a JSON body. On the right, the response is displayed as a pretty-printed JSON object. Both the request body and the response are highlighted with red rectangles.

```
PUT /automotriz_sps?pretty
PUT /automotriz_sps/_doc/1?pretty
{
  "usuario": "Christian Montiel",
  "descripción_autoparte": "puerta de piloto jetta 2021",
  "descuento": 15,
  "fecha": " 2021-06-18T23:35:27"
}
```

```
{
  "_index" : "automotriz_sps",
  "_type" : "_doc",
  "_id" : "1",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

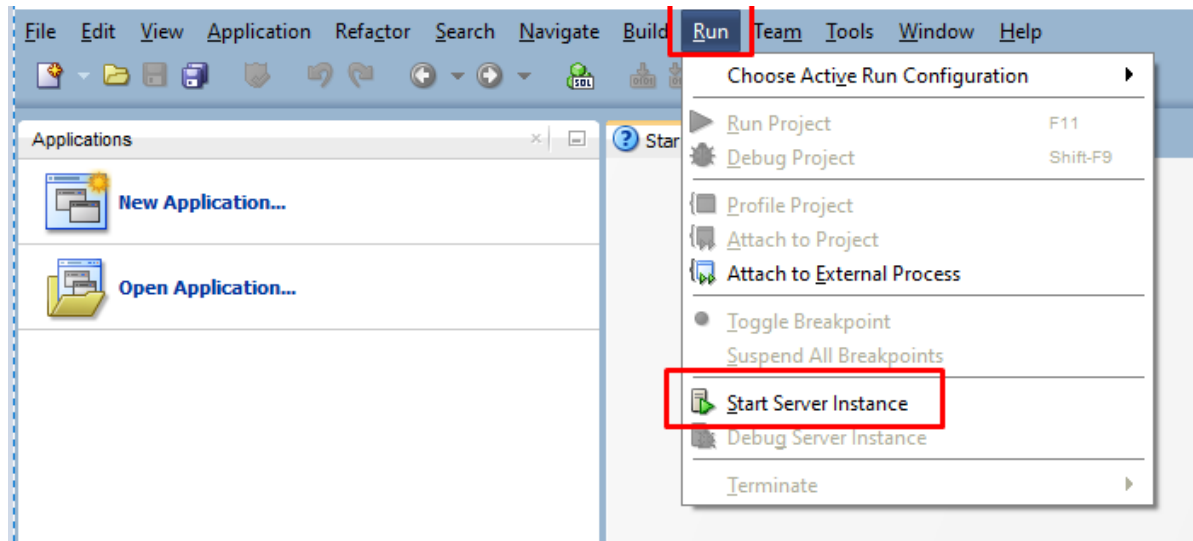
1.3 Oracle SOA Suite 12.2.1.0.0

Nota: En mi caso no pude hacer el consumo del API Rest, intente por varios medios pero no pude, lo que viene debajo es la idea que yo tenía para la resolución de este caso uso y el uso de todos los componentes que utilice

En mi caso no tenia un servidor para poder realizar los despliegues, para poder realizar el despliegue de los componentes se puede hacer con el servidor integrado que maneja el Jdeveloper o se puede hacer una instalación de un servidor.

Para poder realizar el inicio del servidor integrado en el Jdeveloper se realiza de la siguiente manera.

Abrimos el Jdeveloper y en los menús que aparecen en la parte de arriba seleccionamos **run** y después en **Start Server Instance**.



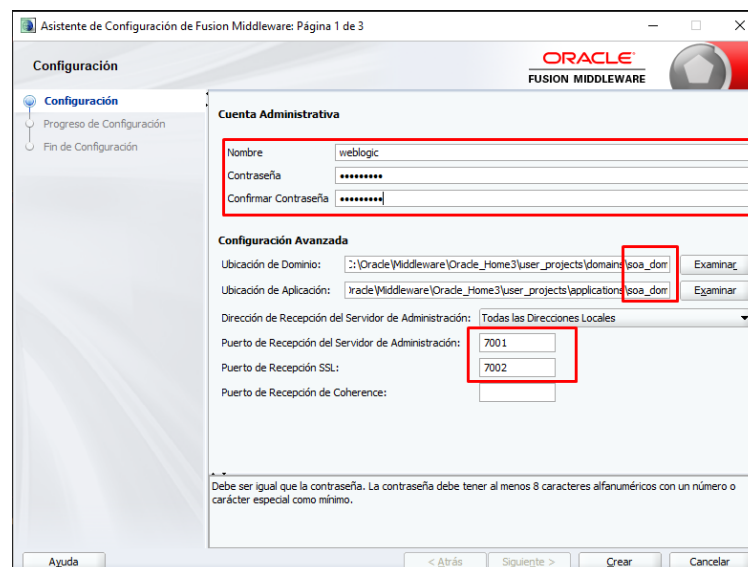
En mi caso no lo hice así, lo que yo hice fue una instalación de los servidores SOA y OSB desde terminal. Esto se realiza de la siguiente manera.

Desde el sistema operativo Windows se tiene que abrir una terminal con permisos de administrador y se ejecutan las siguientes instrucciones, se tendrán que insertar una por una.

Instalación de servidor SOA

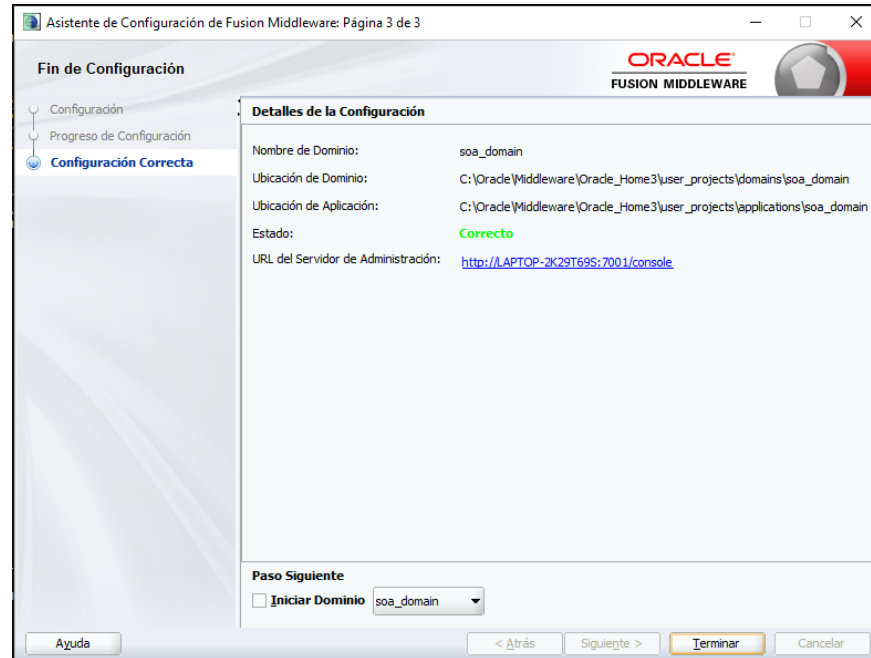
1. `set ORACLE_HOME=C:\Oracle\Middleware\Oracle_Home`
2. `cd %ORACLE_HOME%\oracle_common\common\bin`
3. `set QS_TEMPLATES=%ORACLE_HOME%\osb\common\templates\wls\oracle.soa_template.jar`
4. `qs_config.cmd`

Al ejecutar la última instrucción se despliega la siguiente pantalla en la cual tendremos que otorgarle un usuario y contraseña para poder acceder al servidor, en la parte de **Ubicación de Dominio** y **Ubicación de Aplicación** en la parte del cuadro de texto al final aparece **base_domain** se tendrá que cambiar por el tipo de servidor que queremos instalar en mi caso será **soa_domain** las demás opciones de configuración las dejamos por defecto.



Comenzará el proceso de instalación del servidor y cuando termine aparecerá una ventana como la siguiente en la cual se presionará el botón **Terminar**.

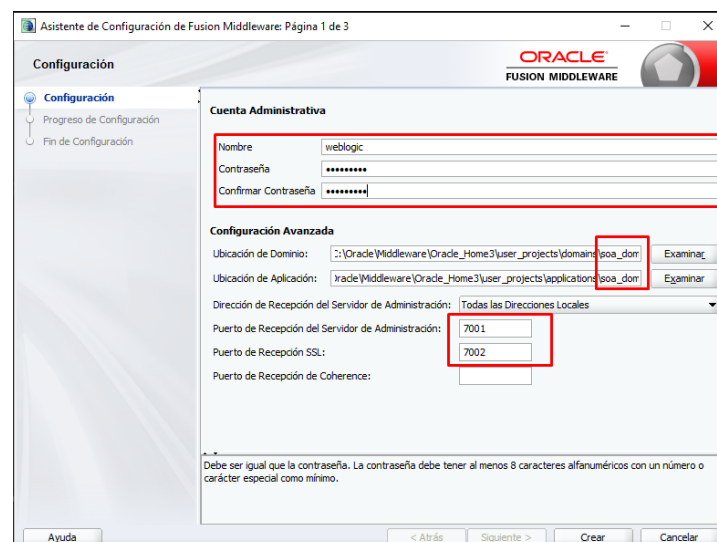
Nota: se recomienda guardar en algun blog de notas la url de **Url del Servidor de Administración**.



Instalación de servidor OSB

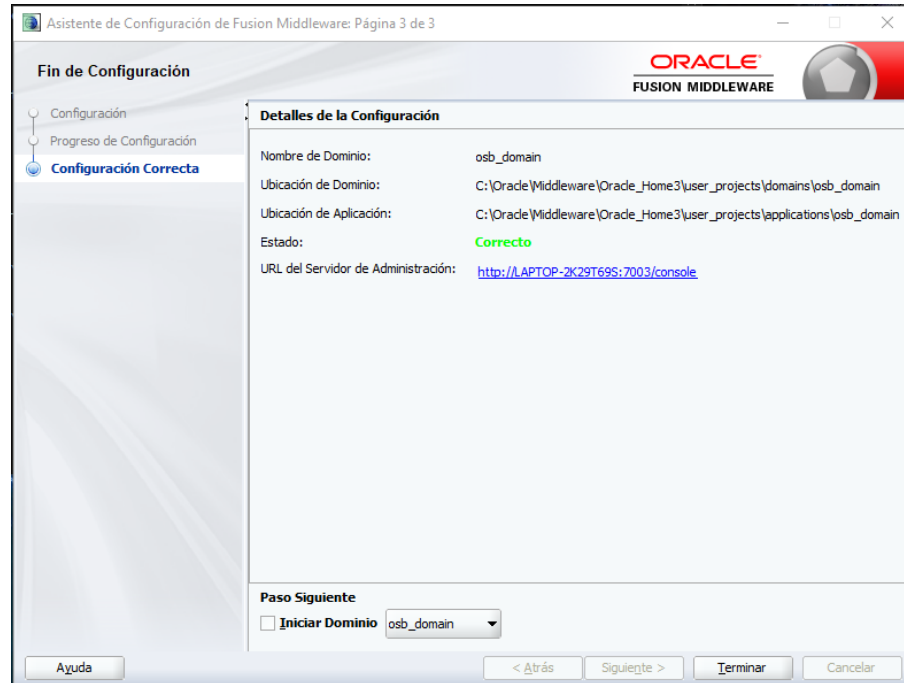
1. set ORACLE_HOME=C:\Oracle\Middleware\Oracle_Home
2. cd %ORACLE_HOME%\oracle_common\common\bin
3. set QS_TEMPLATES=%ORACLE_HOME%\osb\common\templates\wls\oracle.osb_template.jar
4. qs_config.cmd

Al ejecutar la última instrucción se despliega la siguiente pantalla en la cual tendremos que otorgarle un usuario y contraseña para poder acceder al servidor, en la parte de **Ubicación de Dominio** y **Ubicación de Aplicación** en la parte del cuadro de texto al final aparece **base_domain** se tendrá que cambiar por el tipo de servidor que queremos instalar en mi caso será **osb_domain**, en Puerto de Recepción del Servidor de Administración y Puerto de Recepcion SSL los cambiamos a 7003 y 7004 respectivamente.



Comenzará el proceso de instalación del servidor y cuando termine aparecerá una ventana como la siguiente en la cual se presionará el botón **Terminar**.

Nota: se recomienda guardar en algun blog de notas la url de **Url del Servidor de Administración**.

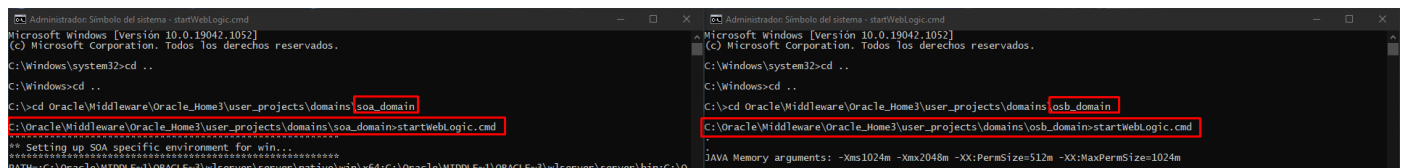


Una vez instalados los servidores procederemos a instalarlos para esto abrimos dos terminales con permisos de administrador y ejecutamos las siguientes instrucciones, instrucción en diferente terminal.

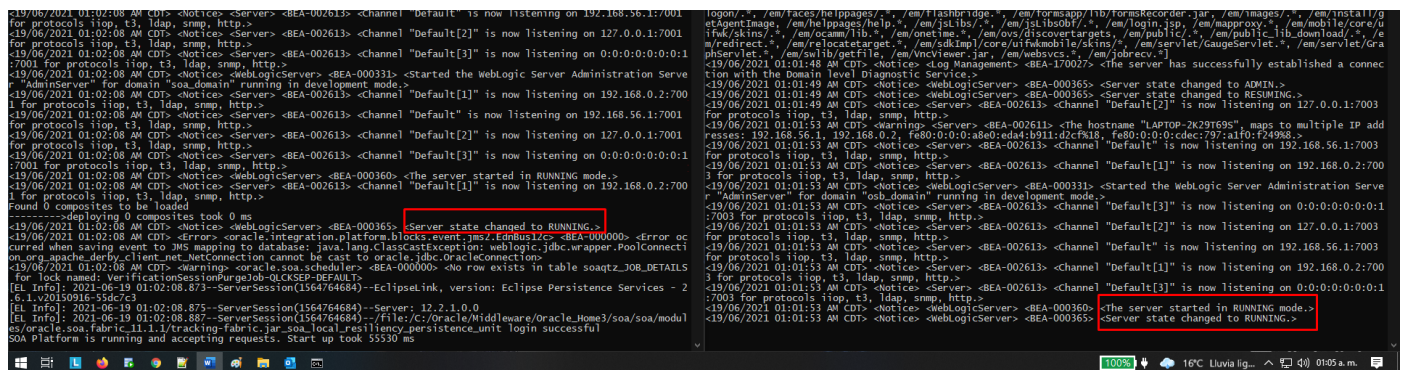
1. `cd Oracle\Middleware\Oracle_Home3\user_projects\domains\osb_domain`
2. `cd Oracle\Middleware\Oracle_Home3\user_projects\domains\soa_domain`

Después en ambas terminales ejecutamos la siguiente instrucción

1. `startWebLogic.cmd`



Después de ejecutar esta última instrucción en ambas terminales se empezarán a ejecutar y al final tiene que aparecer esta instrucción en cada terminal.



Ya terminado este proceso procedemos a ir a un navegador a ver los servidores instalados.

Tendremos el servidor SOA instalado.

The screenshot displays the Oracle Enterprise Manager Fusion Middleware Control 12c interface. The browser address bar shows the URL: `localhost:7001/em/faces/ai/soa/infra?type=oracle_soainfra&target=%2FDomain_soa_domain%2Fsoa_domain%2FAdminServer%2Fsoa-infra`. The page title is "ORACLE Enterprise Manager Fusion Middleware Control 12c". The left sidebar shows the navigation tree with "soa-infra" selected under "SOA Infrastructure". The main content area has tabs for "Dashboard", "Deployed Composites", "Flow Instances", and "Error Hospital". The "Dashboard" tab is active, showing a "Key Configuration" section with "Profile" set to "SOA FOUNDATION", "Instance Tracking" set to "Production", and "Default Query Duration" set to "Last 24 Hours". Below this is the "SOA Runtime Health" section, which shows a green status icon and the text "soa-infra (AdminServer)". To the right, the "Business Transaction Faults" section shows a "Last" filter set to "24 Hours" and a message: "Refresh region to show the latest data. Click graph to drill down." Below this is the "Composites and Adapters Availability" section, which shows a filter set to "AdminServer" and two green checkmarks indicating "No Composite Start-Up Errors" and "No EIS Connectivity Errors".

Tenemos el servidor OSB instalado

The screenshot displays the Oracle Enterprise Manager Fusion Middleware Control 12c interface for the Service Bus. The browser address bar shows the URL: `localhost:7003/em/faces/ai/osb/infra?type=bea_albsb&target=%2FDomain_osb_domain%2Fosb_domain%2FAdminServer%2Fservice-bus`. The page title is "ORACLE Enterprise Manager Fusion Middleware Control 12c". The left sidebar shows the navigation tree with "service-bus" selected under "Service Bus". The main content area has tabs for "Dashboard", "Alert History", "Service Health", "Resequence Messages", "Operations", and "Global Settings". The "Alerts" tab is active, showing a "SLA Alerts" filter and an "Alerts History Duration" of "30 minutes". Below this is a large circular area with the text "No Alerts in the current Alert History duration". To the right, there is a "Send" button and a "Service" button. Below the screenshot, there is another screenshot of the Oracle Service Bus Console 12c. The browser address bar shows the URL: `localhost:7003/servicebus/faces/resources`. The page title is "ORACLE Service Bus Console 12c". The interface shows a "Create", "Discard", and "Exit" buttons at the top. Below these buttons is a tree view showing the project structure: "All Projects" (expanded) containing "default" and "System".

Una vez que instalé los servidores OSB y SOA me puse a hacer el código para el requerimiento. Lo que hice primero fue generar una aplicación de OSB la cual tuviera dos carpetas, la carpeta **AutomotrizSPSApp** tiene el código para la resolución del caso de uso, tiene los ProxyService, Pipelines, WSLD, NXSD, BusinessService, la carpeta **MetaData** contiene todos los esquemas comunes y dvm que se ultizaron. Lo dividi de esta manera ya que me permite la reultizacion de esquemas y del dvm.

Genere el dvm **ERROR_CODE**, este dvm me permite setear los datos de los códigos de error que se utilizaran en el flujo. Los códigos de error 00XX representan el servicio que se expone del lado de OSB, los códigos 01XX representan los servicios que se usaron para el sistema destino **Elastic**.

Start Page AutomotrizSPSEBO.xsd AutomotrizSPSEBM.xsd ERROR_CODE.dvm

Domain Value Map(DVM)

Name: ERROR_CODE

Description:

Map Table:

CODE	USER_MESSAGE	ERROR_CODE
GEN-0000	Execution success	0000
GEN-0001	Invalid input data. Please verify that the data matches the schema	0001
GEN-0100	Elastic Response Service Error	0100
GEN-0101	Invalid input data. Please verify that the data matches the schema	0101

Genere el xsd **AutomotrizSPSEBM.xsd** para el mensaje canónico que se expondrá mediante un servicio SOAP generado por el ProxyService. Se generaron diferentes ComplexType, los que al principio tienen **Process** es porque viajan desde la llamada del servicio SOAP, es decir empezaran el procesamiento de la información, los que al principio tienen **Sync** es el mensaje casi final para los sistemas destino, decidí nombrarlo así ya que se esta realizando una sincronización de los descuentos ya sea que se creen, de lean, de actualicen o se borren. Dentro de estos mensajes están definidos la cabecera del request que se hará y se definió la respuesta ya sea que la respuesta sea satisfactoria o errónea.

Request de la transacción.

Para el Header quedo de la siguiente manera

Campo	Descripcion	Tipo de dato	Obligatorio
TrackingID	Identificador único de la instancia, este se asigna desde el request que manda el usuario	String	SI
InstanceID	Identificador único de la instancia, este se asigna en el flujo	String	NO
Operation	Operación que se realizara CRETE, READ, UPDATE, DELETE	String	SI

Para el Cuerpo del mensaje quedo de la siguiente manera

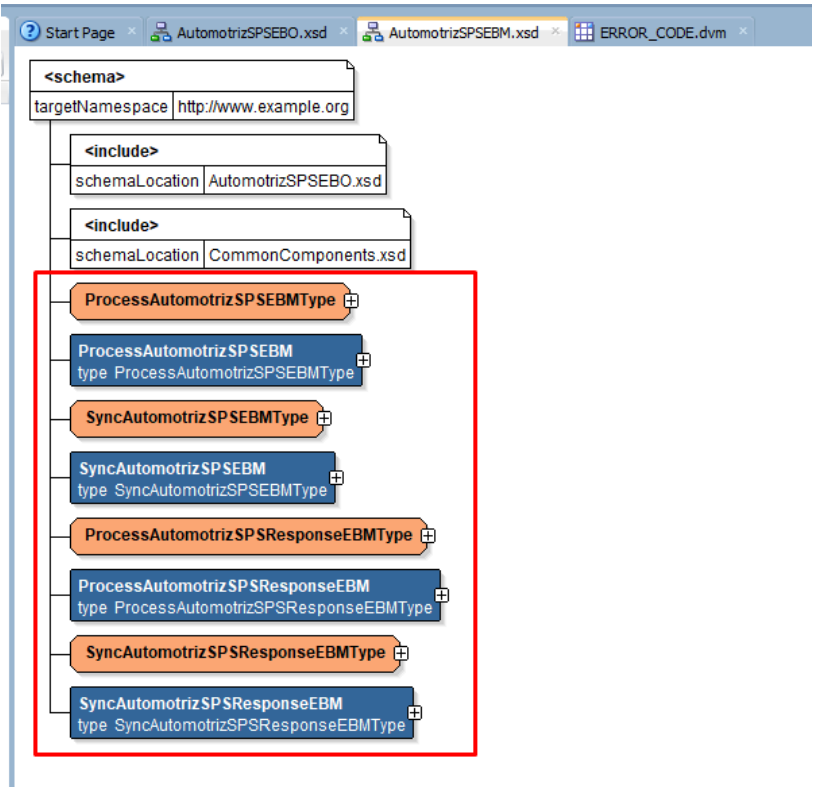
Campo	Descripcion	Tipo de dato	Obligatorio
Usuario	Nombre del usuario que hizo el descuento	String	SI
DescripcionAutoparte	Descripción de la autoparte	String	SI

Descuento	Descuento que aplicara	Long	SI
Fecha	Fecha en la que se realizo el descuento	String	SI

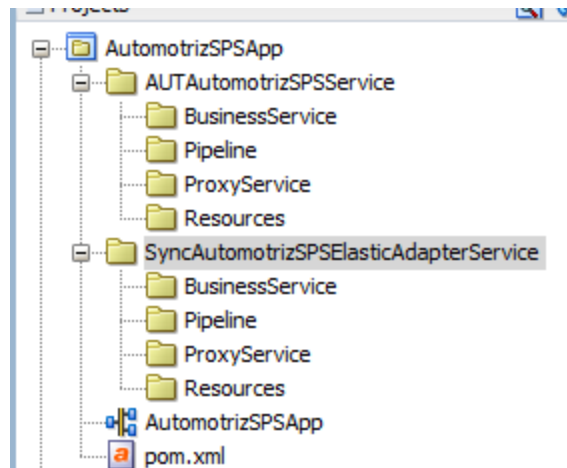
Response de la transacción.

Para el Response quedo de la siguiente manera

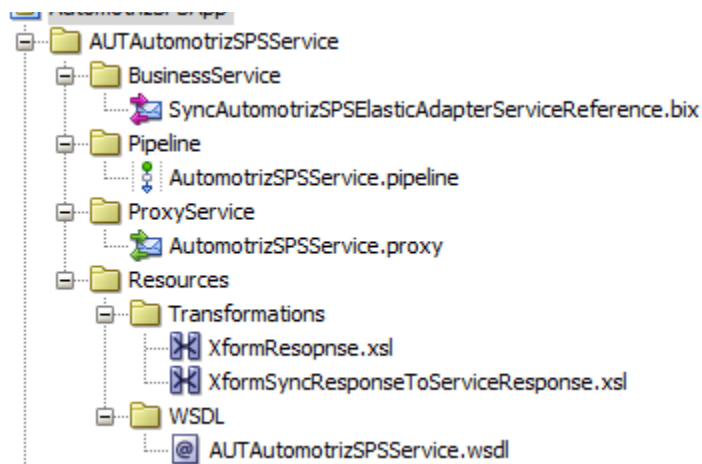
Campo	Descripcion	Tipo de dato
Code	Codigo de error generado	String
Message	Mensaje de error generado	String
Detail	Detalle del error	String
TrackingID	Identificador único de la instancia, este se asigna desde el request que manda el usuario	String
InstanceID	Identificador único de la instancia, este se asigna en el flujo	String
Operation	Operación que se intentó realizar CRETE, READ, UPDATE, DELETE	String



La carpeta **AutomotrizSPSApp** tiene dos subcarpetas las cuales son **AUTAutomotrizSPSService** y **SyncAutomotrizSPSElasticAdapterService**, la primera es la que tendrá el código para exponer el servicio SOAP, la segunda contendrá el código para dejar el mensaje al sistema destino, en este caso **Elastic**.



La carpeta **AUTAutomotrizSPSService** se muestra a continuación



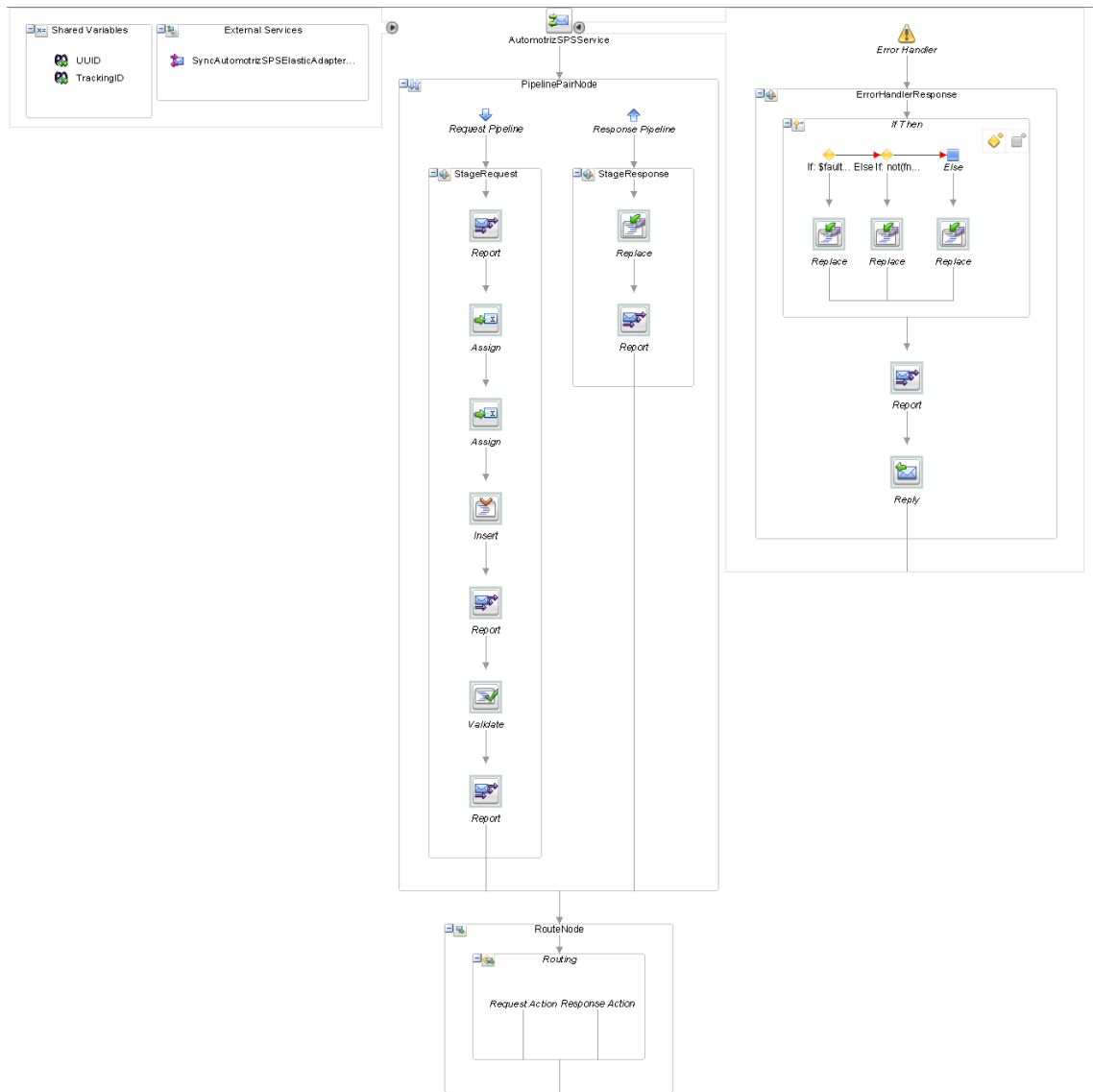
La carpeta BusinessService tiene el .bix que hace la llamada a nuestro servicio de Sync que es el servicio destino construido

La carpeta Pipeline tiene el .pipeline el cual es la lógica que tendrá este servicio.

La carpeta ProxyService tiene el .proxy que es el que se usa para exponer nuestro servicio como un servicio SOAP

La carpeta Resources tiene todos los recursos básicos para el funcionamiento de este proyecto.

Hablando del pipeline se explica cómo está constituido el código.

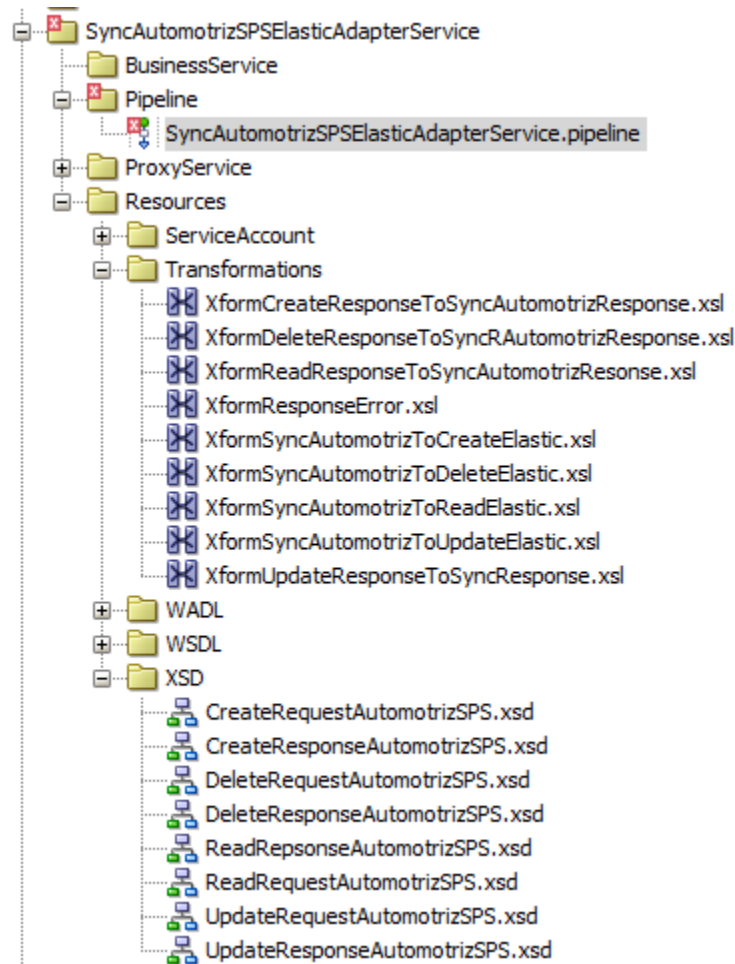


Primero se crearon las variables **UUID** y **TrackingID** esto para almacenar los valores y que no se pierda el dato en el flujo de la integración, en el **StageRequest** se tiene un **Report** este sirve como un tipo log el cual me indica como está llegando la información después es dos **Assing** y un **Insert** estos me sirven para asignar la información a variables temporales e insertar uno de estos datos dentro del XSD de request, después es un **Report**, un **Validate** y otro **Report**, el funcionamiento de los reports que funcionan como un log el cual me va indicando como van pasando los datos por la integración, el validate me ayuda a validar el esquema es decir que los tipos de datos en el request y en la definición del XSD sean los correctos. Después se tiene el RouteNode el cual hace la llamada al servicio de Sync.

En el StateResponse se tiene un report y un replace, nuevamente el report me funciona como un log para ver como viaja la información, el replace solo es un tratamiento de los datos a través de de una transformación (XSLT).

El ErrorHandlerResponse contiene el manejo de los errores que se vayan suscitando en el flujo.

La carpeta **SyncAutomotrizSPSElasticAdapterService** se muestra a continuación.



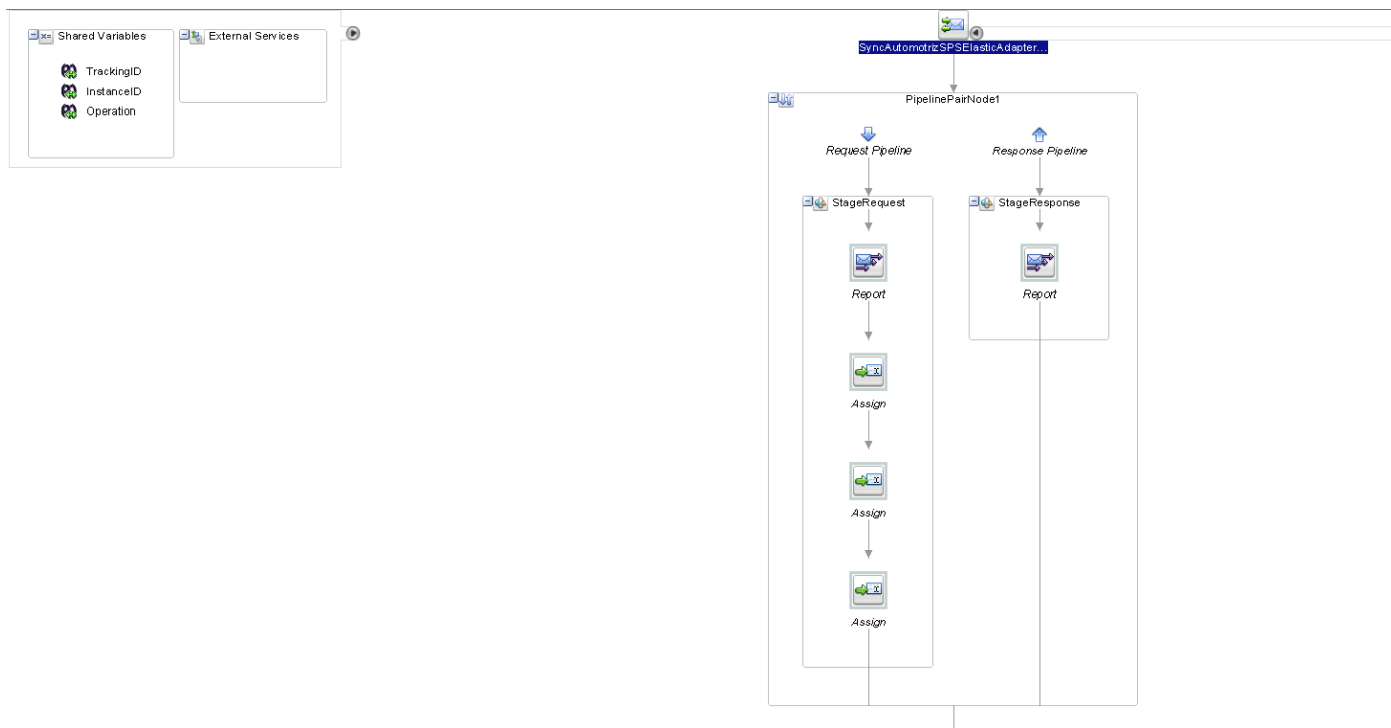
La carpeta **BusinessService** tiene el .biz que debería hacer la llamada al API Rest de Elastic

La carpeta **Pipeline** tiene el .pipeline el cual es la lógica que tendrá este servicio.

La carpeta **ProxyService** tiene el .proxy que es el que se usa para exponer nuestro servicio y será consumido por **AUTAutomotrizSPSService**

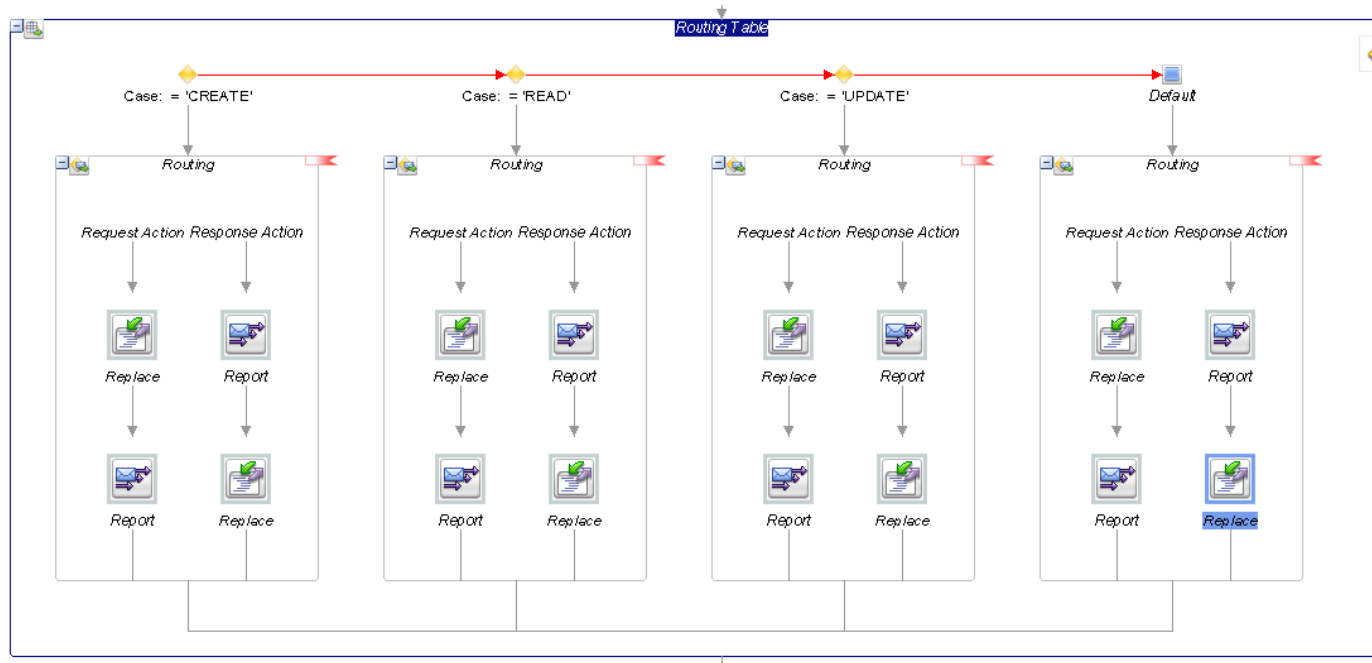
La carpeta **Resources** tiene todos los recursos básicos para el funcionamiento de este proyecto. De esta carpeta se hicieron unos esquemas nativos a partir de los JSON que expone como request o response Elastic para las diferentes operaciones que tiene este caso de uso

Hablando del pipeline se explica cómo está constituido el código.

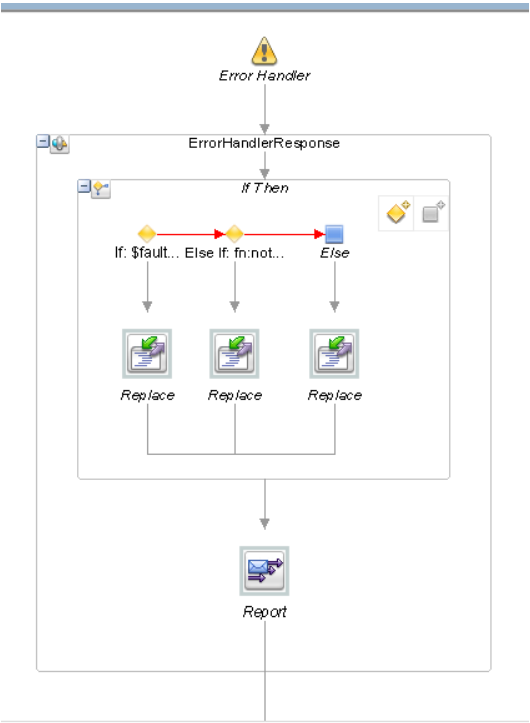


Primero se crearon las variables **InstanceID**, **TrackingID** y **Operation** esto para almacenar los valores y que no se pierda el dato en el flujo de la integración, en el **StageRequest** se tiene un **Report** este sirve como un tipo log el cual me indica como está llegando la información después es tres **Assing**, estos me sirven para asignar la información a variables temporales e insertar uno de estos datos dentro del XSD de request. Se tiene el Stage Response el cual solo contiene un report que me ayuda a ver como viajo la información hacia **AUTAutomotrizSPSService**

Después se creó un RouteTable, aquí mi idea era que con 4 .bix se colocaran las diferentes operaciones que se realizarían para el caso de uso y con los NXSD que se crearon eran pasar el EBM de la integración al ABM del sistema destino, se crearon las transformaciones para “simular” los casos de usos tanto en la petición así como en la respuesta.



Se tiene los reports para ver como viaja la información de las diferentes operaciones para el caso de uso.
Por último, se tiene el manejador de errores



En el EM de OSB se ven los reports de esta manera, como dije anteriormente esto me ayuda a visualizar como va el flujo y la información que va pasando por los diferentes puntos de la integración.

ORACLE Enterprise Manager Fusion Middleware Control 12c

WebLogic Domain | weblogic

Jun 20, 2021 3:32:38 PM CDT

service-bus

Search

Inbound Service Name

Error Code

Report Index

Date Range

From To

For the last 0 days 0 0 (h:min)

Search Reset

Message Reports

View Purge Detach

Report Index	DB Timestamp	Inbound Service	Path	Error Code
Error.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	OSB-382505
Validate.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
In.Tracking= string string string ...	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Error.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	OSB-382505
Validate.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
In.Tracking= string string string ...	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Validate.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
In.Tracking= string string string ...	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Response.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Request.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Validate.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
In.Tracking= string string string ...	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Request.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Validate.Tracking=string	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
In.Tracking= string string string ...	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Request.Tracking=1234567	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Validate.Tracking=1234567	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
In.Tracking= 1234567 CREATE ...	06/20/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	
Response.Tracking=1234567	06/19/2021 12:00:00 AM	AutomotrizSPSService	AutomotrizSPSApp/AUTAuto...	

1.3.1 Create

Mi idea para la creación de un nuevo documento es la siguiente.

En el servicio **SOAP** que se expuso de **OSB** en el request es mandarlo de esta manera

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:exam="http://www.example.org">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<exam:ProcessAutomotrizSPSEBM>
```

```
<exam:Header>
```

```
<exam:TrackingID>1</exam:TrackingID>
```

```
<exam:Operation>CREATE</exam:Operation>
```

```
</exam:Header>
```

```
<exam:DataArea>
```

```
<exam:Usuario>Christian</exam:Usuario>
```

```
<exam:DescripcionAutoparte>piston</exam:DescripcionAutoparte>
```

```
<exam:Descuento>15</exam:Descuento>
```

```
<exam:Fecha>2015</exam:Fecha>
```

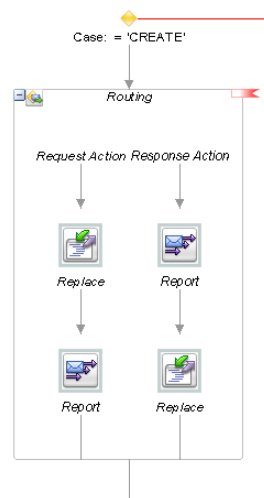
```
</exam:DataArea>
```

```
</exam:ProcessAutomotrizSPSEBM>
```

```
</soapenv:Body>
```

```
</soapenv:Envelope>
```

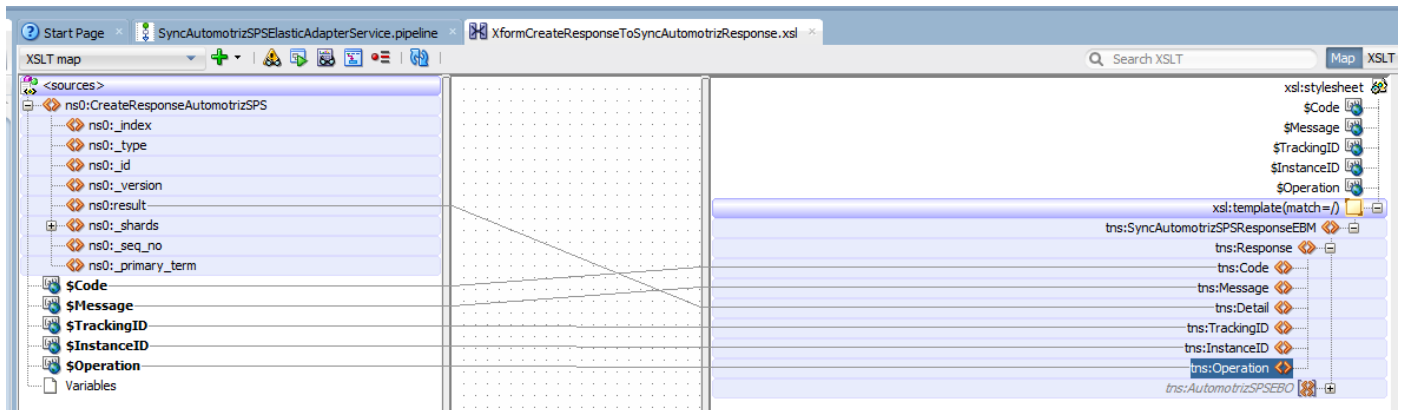
De esta manera se estaría enviando en el **TrackingID** este uno que seria un consecutivo el cual representaría el índice del documento a crear, se enviaría también la operación a realizar en este caso un **CREATE** en mayúsculas esto pasaría del servicio **AutomotrizSPSService** al **SyncAutomotrizSPSElasticAdapterService** y aquí pasaría por el **RouteTable** el cual detectaría la operación que se realizara llegando aquí



Y se en la respuesta del servicio de Elastic nos respondería con lo siguiente

```
1 {  
2   "_index" : "automotriz_sps",  
3   "_type" : "_doc",  
4   "_id" : "26",  
5   "_version" : 1,  
5   "result" : "created",  
7   "_shards" : {  
3     "total" : 2,  
3     "successful" : 2,  
3     "failed" : 0  
1   },  
2   "_seq_no" : 36,  
3   "_primary_term" : 1  
4 }  
5
```

Esto se transformaría al lenguaje que habla SOA en este caso XML con la siguiente transformación y haciendo un tratado de la información enviando como respuesta parte de la cabecera de la petición, y los códigos de error que están en el DVM.



Esta respuesta regresaría hasta el servicio **AutomotrizSPSService**.

1.3.2 Read

Mi idea para la lectura de un documento es la siguiente.

En el servicio SOAP que se expuso de OSB en el request es mandarlo de esta manera

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:exam="http://www.example.org">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<exam:ProcessAutomotrizSPSEBM>
```

```
<exam:Header>
```

```
<exam:TrackingID>1</exam:TrackingID>
```

```
<exam:Operation>READ</exam:Operation>
```

```
</exam:Header>
```

```
<exam:DataArea>
```

```
<exam:Usuario>Christian</exam:Usuario>
```

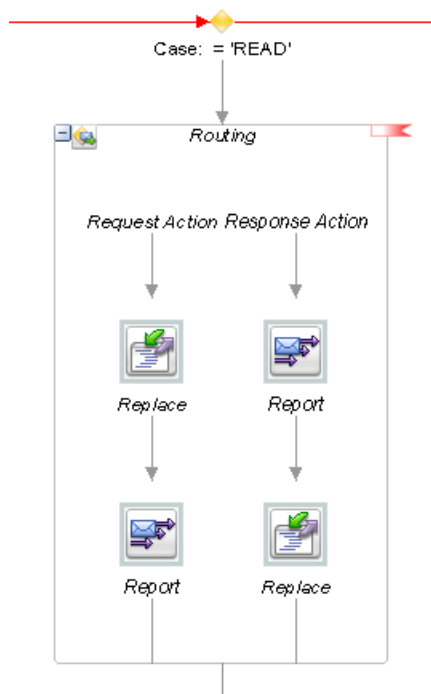
```
</exam:DataArea>
```

```
</exam:ProcessAutomotrizSPSEBM>
```

```
</soapenv:Body>
```

```
</soapenv:Envelope>
```

De esta manera se estaría enviando en el **TrackingID** este uno que sería un consecutivo, se enviaría también la operación a realizar en este caso un **READ** en mayúsculas esto pasaría del servicio **AutomotrizSPSService** al **SyncAutomotrizSPSElasticAdapterService** y aquí pasaría por el RouteTable el cual detectaría la operación que se realizara llegando aquí



En la URL de petición a Elastic se completaría de la siguiente manera

GET /automotriz_sps/_search?pretty=true&q=[parámetro del request]: [parámetro del request]

En nuestro caso queda de la siguiente manera

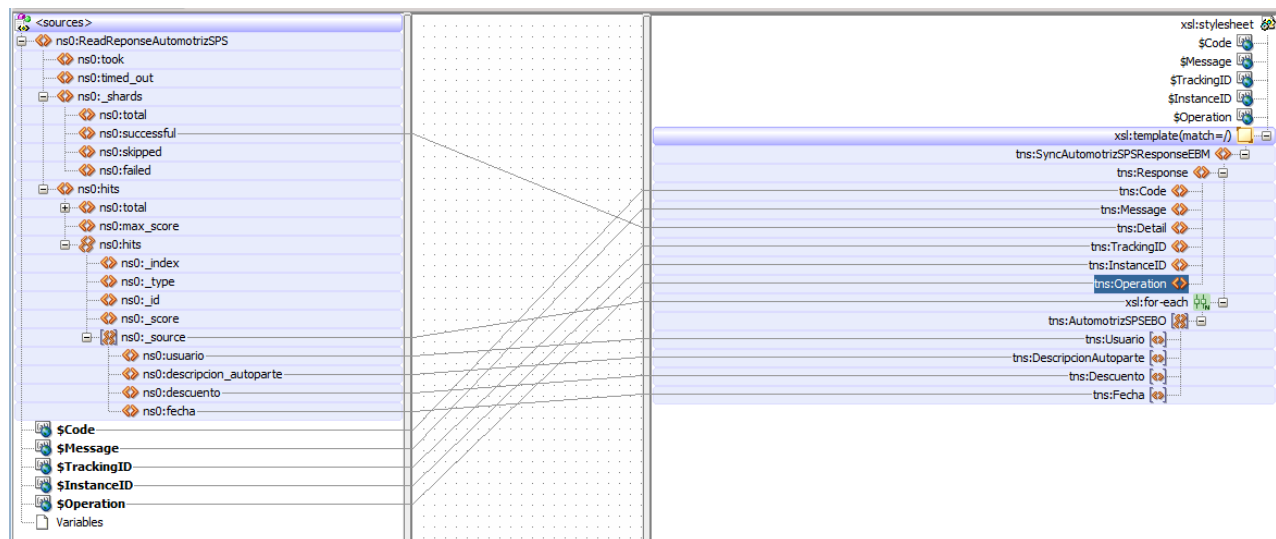
GET /automotriz_sps/_search?pretty=true&q=usuario: Andrea Ramirez

Y se en la respuesta del servicio de Elastic nos respondería con lo siguiente

```
story Settings Help
1 POST /automotriz_sps/_search
2 {
3   "size":20,
4   "sort":{"_id":"desc"},
5   "query":{"match_all":{}}
6 }
7
8 GET /automotriz_sps/_search?pretty=true&q=usuario:Andrea Ramirez
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
```

```
1 {
2   "took" : 3,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 6,
13      "relation" : "eq"
14    },
15    "max_score" : 2.990987,
16    "hits" : [
17      {
18        "_index" : "automotriz_sps",
19        "_type" : "_doc",
20        "_id" : "21",
21        "_score" : 2.990987,
22        "_source" : {
23          "usuario" : "Andrea Ramirez",
24          "descripción_autoparte" : "freno de mano",
25          "descuento" : 15,
26          "fecha" : " 2021-06-18T23:35:27"
27        }
28      },
29      {
30        "_index" : "automotriz_sps",
31        "_type" : "_doc",
32        "_id" : "22",
33        "_score" : 2.990987,
34        "_source" : {
35          "usuario" : "Andrea Ramirez",
36          "descripción_autoparte" : "radiador",
37          "descuento" : 15,
38          "fecha" : " 2021-06-18T23:35:27"
39        }
40      },
41      {
42        "_index" : "automotriz_sps",
43        "_type" : "_doc",
44        "_id" : "23"
```

Esto se transformaría al lenguaje que habla SOA en este caso XML con la siguiente transformación y haciendo un tratado de la información enviando como respuesta parte de la cabecera de la petición, y los códigos de error que están en el DVM.



Esta respuesta regresaría hasta el servicio **AutomotrizSPSService**.

1.3.3 Update

Mi idea para la actualización de un documento es la siguiente.

En el servicio **SOAP** que se expuso de **OSB** en el request es mandarlo de esta manera

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:exam="http://www.example.org">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<exam:ProcessAutomotrizSPSEBM>
```

```
<exam:Header>
```

```
<exam:TrackingID>1</exam:TrackingID>
```

```
<exam:Operation>UPDATE</exam:Operation>
```

```
</exam:Header>
```

```
<exam:DataArea>
```

```
<exam:Usuario>Christian</exam:Usuario>
```

```
<exam:DescripcionAutoparte>llanta</exam:DescripcionAutoparte>
```

```
<exam:Descuento>15</exam:Descuento>
```

```
<exam:Fecha>2015</exam:Fecha>
```

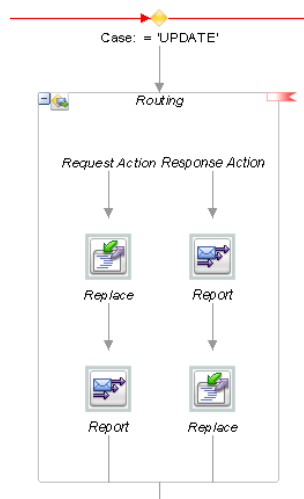
```
</exam:DataArea>
```

```
</exam:ProcessAutomotrizSPSEBM>
```

```
</soapenv:Body>
```

```
</soapenv:Envelope>
```

De esta manera se estaría enviando en el **TrackingID** este uno que sería un consecutivo el cual representaría el índice del documento a actualizar, se enviaría también la operación a realizar en este caso un **UPDATE** en mayúsculas esto pasaría del servicio **AutomotrizSPSService** al **SyncAutomotrizSPSElasticAdapterService** y aquí pasaría por el **RouteTable** el cual detectaría la operación que se realizara llegando aquí



En la URL de petición a Elastic se completaría de la siguiente manera

DELETE /automotriz_sps/_doc/[Indice que viene en el TrackingID]

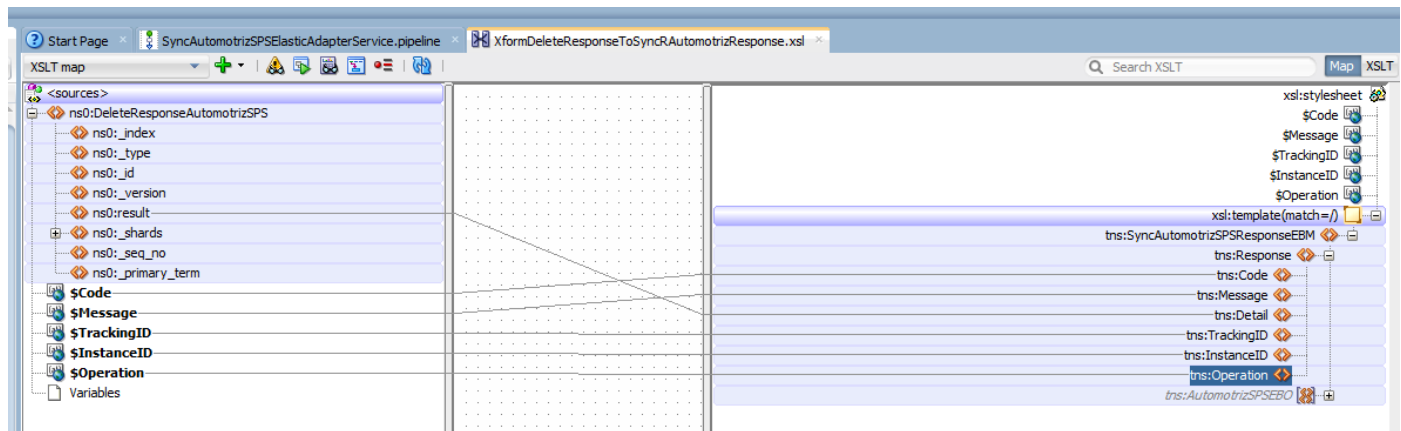
En nuestro caso queda de la siguiente manera

DELETE /automotriz_sps/_doc/

Y se en la respuesta del servicio de Elastic nos respondería con lo siguiente

```
1 {  
2   "_index" : "automotriz_sps",  
3   "_type" : "_doc",  
4   "_id" : "26",  
5   "_version" : 2,  
6   "result" : "deleted",  
7   "_shards" : {  
8     "total" : 2,  
9     "successful" : 2,  
10    "failed" : 0  
11  },  
12  "seq_no" : 37,  
13  "_primary_term" : 1  
14 }  
15
```

Esto se transformaría al lenguaje que habla SOA en este caso XML con la siguiente transformación y haciendo un tratado de la información enviando como respuesta parte de la cabecera de la petición, y los códigos de error que están en el DVM.



Esta respuesta regresaría hasta el servicio **AutomotrizSPSService**.

1.3.4 Delete

Mi idea para el borrado de un documento es la siguiente.

En el servicio **SOAP** que se expuso de **OSB** en el request es mandarlo de esta manera

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:exam="http://www.example.org">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<exam:ProcessAutomotrizSPSEBM>
```

```
<exam:Header>
```

```
<exam:TrackingID>1</exam:TrackingID>
```

```
<exam:Operation>UPDATE</exam:Operation>
```

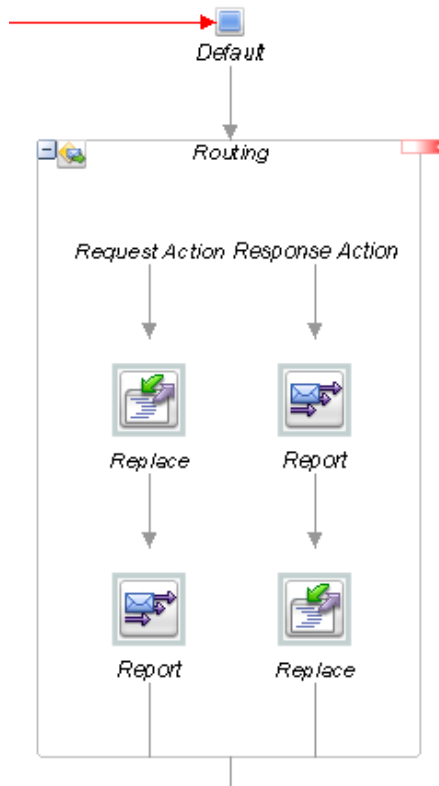
```
</exam:Header>
```

```
</exam:ProcessAutomotrizSPSEBM>
```

```
</soapenv:Body>
```

```
</soapenv:Envelope>
```

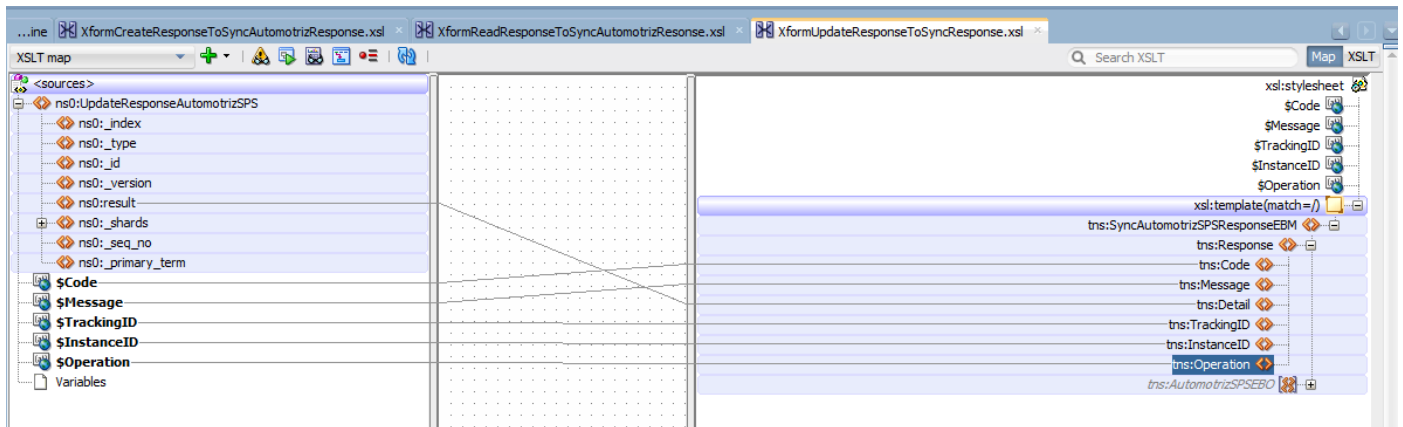
De esta manera se estaría enviando en el **TrackingID** este uno que sería un consecutivo el cual representaría el índice del documento a borrar, se enviaría también la operación a realizar en este caso un **DELETE** en mayúsculas esto pasaría del servicio **AutomotrizSPSService** al **SyncAutomotrizSPSElasticAdapterService** y aquí pasaría por el **RouteTable** el cual detectaría la operación que se realizara llegando aquí



Y se en la respuesta del servicio de Elastic nos respondería con lo siguiente

```
1 {  
2   "_index" : "automotriz_sps",  
3   "_type" : "_doc",  
4   "_id" : "26",  
5   "_version" : 1,  
5   "result" : "created",  
7   "_shards" : {  
3     "total" : 2,  
3     "successful" : 2,  
3     "failed" : 0  
1   },  
2   "_seq_no" : 36,  
3   "_primary_term" : 1  
4 }  
5
```

Esto se transformaría al lenguaje que habla SOA en este caso XML con la siguiente transformación y haciendo un tratado de la información enviando como respuesta parte de la cabecera de la petición, y los códigos de error que están en el DVM.



Esta respuesta regresaría hasta el servicio **AutomotrizSPSService**.

1.4 Búsqueda sobre los índices

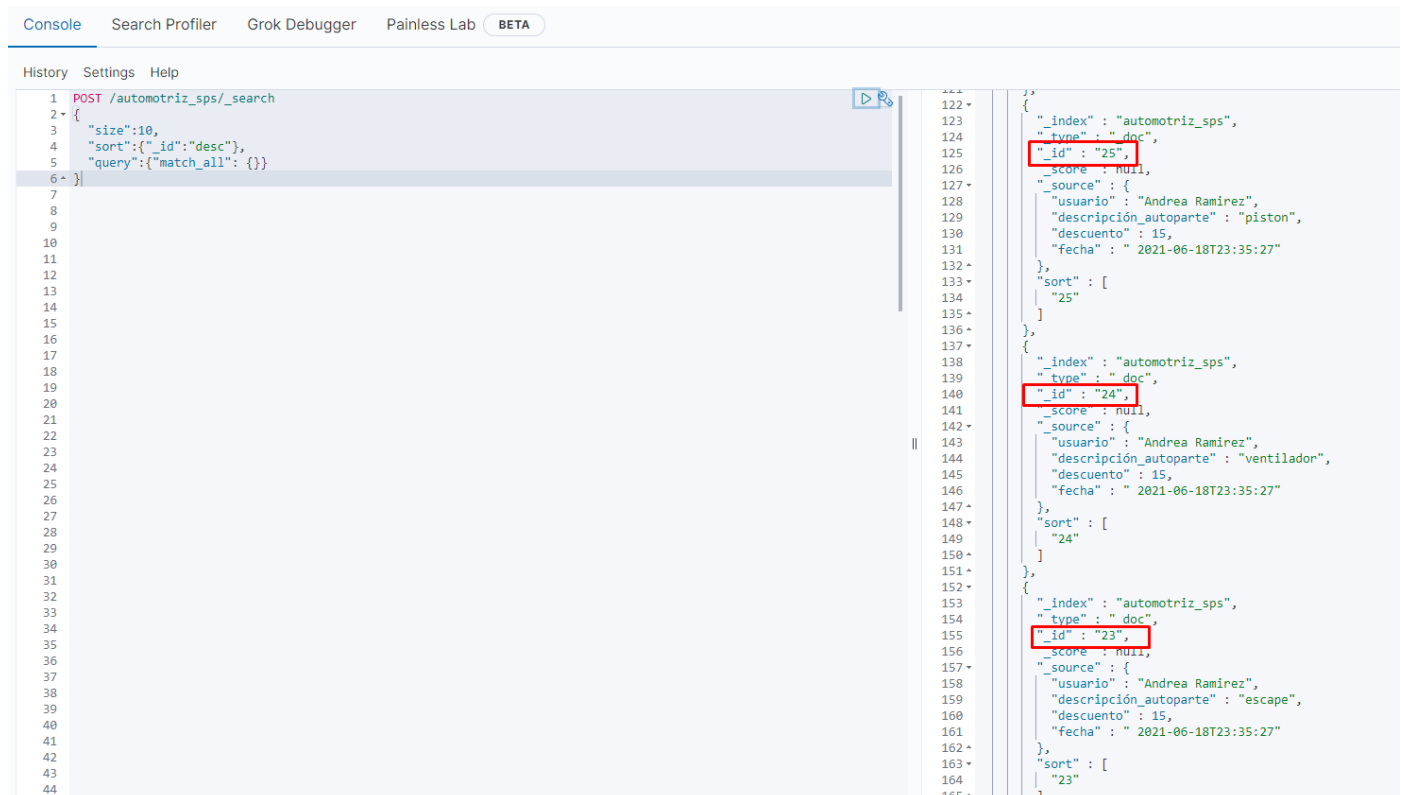
1.4.1 Busca los últimos 20 documentos

Con la siguiente instrucción me trae los últimos 20 documentos que se han insertado en Elastic

POST /automotriz_sps/_search

```
{
  "size":20,
  "sort":{"_id":"desc"},
  "query":{"match_all":{}}
}
```

Por temas de vista la imagen solo ve 3 pero si son los 20 solicitados



The screenshot displays the Elasticsearch DevTools interface. The top navigation bar includes 'Console', 'Search Profiler', 'Grok Debugger', 'Painless Lab', and a 'BETA' badge. Below the navigation bar, the 'History' tab is active, showing a list of queries. The first query is selected, and its details are shown in the main panel. The query is a POST request to /automotriz_sps/_search with the following body:

```
{
  "size":20,
  "sort":{"_id":"desc"},
  "query":{"match_all":{}}
}
```

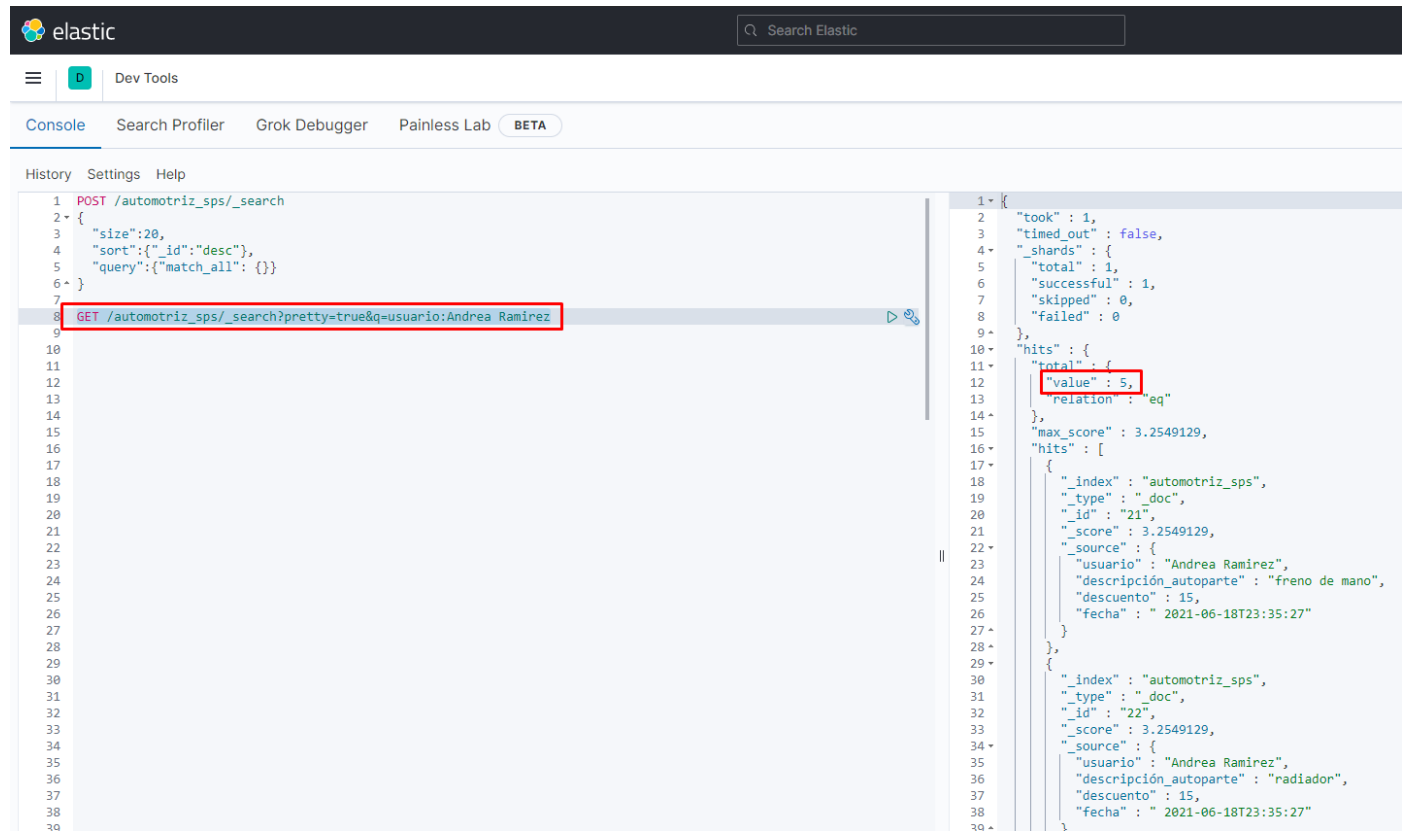
The results panel shows the search results. The first three results are visible, each with a red box highlighting the '_id' field. The results are sorted by '_id' in descending order. The first result has '_id': '25', the second has '_id': '24', and the third has '_id': '23'. Each result also includes '_source' information, such as 'usuario', 'descripción_autoparte', 'descuento', and 'fecha'.

1.4.2 Busca los documentos de un usuario en específico

Con la siguiente consulta se hace obtienen los documentos del un usuario en específico

GET /automotriz_sps/_search?pretty=true&q=usuario:Andrea Ramirez

Para este usuario solo hay 5 registros



The screenshot shows the Elasticsearch Dev Tools interface. The left pane displays the executed query: `GET /automotriz_sps/_search?pretty=true&q=usuario:Andrea Ramirez`. The right pane shows the resulting JSON response, which includes search statistics and a list of hits. The `"value": 5` in the `"total": { "value": 5 }` section is highlighted with a red box, indicating the total number of documents found. The hits section contains two document snippets, each with fields like `_index`, `_type`, `_id`, `_score`, and `_source`.

```
1 POST /automotriz_sps/_search
2 {
3   "size":20,
4   "sort":{"_id":"desc"},
5   "query":{"match_all": {}}
6 }
7
8 GET /automotriz_sps/_search?pretty=true&q=usuario:Andrea Ramirez
9
10 {
11   "took" : 1,
12   "timed_out" : false,
13   "_shards" : {
14     "total" : 1,
15     "successful" : 1,
16     "skipped" : 0,
17     "failed" : 0
18   },
19   "hits" : {
20     "total" : {
21       "value" : 5,
22       "relation" : "eq"
23     },
24     "max_score" : 3.2549129,
25     "hits" : [
26       {
27         "_index" : "automotriz_sps",
28         "_type" : "_doc",
29         "_id" : "21",
30         "_score" : 3.2549129,
31         "_source" : {
32           "usuario" : "Andrea Ramirez",
33           "descripción_autoparte" : "freno de mano",
34           "descuento" : 15,
35           "fecha" : " 2021-06-18T23:35:27"
36         }
37       },
38       {
39         "_index" : "automotriz_sps",
40         "_type" : "_doc",
41         "_id" : "22",
42         "_score" : 3.2549129,
43         "_source" : {
44           "usuario" : "Andrea Ramirez",
45           "descripción_autoparte" : "radiador",
46           "descuento" : 15,
47           "fecha" : " 2021-06-18T23:35:27"
48         }
49       }
50     ]
51   }
52 }
```