|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Leer desde MCA la información enviada por el BCK | | | | |  |
| Registro de modificaciones | | | | | | | |
| **Versión** | **Descripción** | | **Autor** | **Fecha creación** | **Aprobado por** | **Fecha de aprobación** | |
| 1.0 | UAT Inicial | | everis | 18/10/2019 |  | 18/10/2019 | |
|  |  | |  |  |  |  | |

/

**Contenido**

[1. Descripción 2](#_Toc25596380)

[2. Recursos 2](#_Toc25596381)

[3. Definición 2](#_Toc25596382)

[4. Criterios de aceptación 2](#_Toc25596383)

[5. Tabla de resultados 2](#_Toc25596384)

[6. Tests Automáticos con JUnit 4](#_Toc25596385)

[6.1 Test en MCA 4](#_Toc25596386)

[6.1.1 Guardado en el profile de cabeceras cuando no había nada previamente guardado y la propiedad está activada 4](#_Toc25596387)

[6.1.2 Guardado en el profile de cabeceras cuando ya existe información previa, está relacioanda con ARQUEO y la propiedad está activada 6](#_Toc25596388)

[6.1.3 Guardado en el profile de cabeceras cuando ya existe información previa, no está relacionada con ARQUEO y la propiedad está activada 7](#_Toc25596389)

[6.2 Test en BCK 9](#_Toc25596390)

[6.2.1 Clase PrepareArqueoInfoHandler 9](#_Toc25596391)

[6.2.1.1 Si el profile viene nulo no se ejecuta el seteo de cabeceras 9](#_Toc25596392)

[6.2.1.2 Si el canal no tiene permitido ejecutar el seteo de cabeceras 9](#_Toc25596393)

[6.2.1.3 Si el SN no es arqueable no se ejecuta el seteo de cabeceras 10](#_Toc25596394)

[6.2.1.4 Si el SN es arqueable y tiene un dato de arqueo, se ejecuta el seteo de cabeceras 11](#_Toc25596395)

[6.2.1.5 Si el SN es arqueable pero no tiene datos de arqueo, no se ejecuta el seteo de cabeceras 11](#_Toc25596396)

[6.2.1.6 Si el SN es arqueable y tiene varios datos de arqueo, se ejecuta el seteo de cabeceras 12](#_Toc25596397)

[6.2.2 Clase ArqueoDataImpl 13](#_Toc25596398)

[6.2.2.1 Si el profile es nulo se debe devolver un NullPointerException 13](#_Toc25596399)

[6.2.2.2 Si el profile contiene un mapa pero este está vacío, no se pasará ningún dato a las cabeceras 13](#_Toc25596400)

[6.2.2.3 Si el profile contiene un mapa con un único elemento, la cabecera contendrá ese único valor 14](#_Toc25596401)

[6.2.2.4 Si el profile contiene un mapa con varios elementos, la cabecera contendrá todos los valores 14](#_Toc25596402)

[7. Tests Manuales 15](#_Toc25596403)

[7.1 Autorización sin motivo automático 16](#_Toc25596404)

[7.2 Autorización con un motivo OK 16](#_Toc25596405)

[7.3 Autorización con un motivo NOK 17](#_Toc25596406)

[7.4 Autorización con un motivo OK y dos no automático 18](#_Toc25596407)

[7.5 Autorización con dos motivos NOK 19](#_Toc25596408)

[7.6 Autorización con un motivo OK y uno NOK 20](#_Toc25596409)

[7.7 Autorización con un motivo no automático, pero por defecto es Automático 20](#_Toc25596410)

# Descripción

Tras el desarrollo de la nueva implementación para poder traspasar la información de aquellos SN “arqueable” desde BCK a MCA, se adquiere la necesidad de comprobar que la respuesta dada a la información obtenida en las cabeceras provenientes desde BCK sea la correcta.

Esta nueva versión va a ser probada en el entorno de DSV, y utilizaremos una demo generada para comprobar el correcto funcionamiento del desarrollo realizado y de los test unitarios automáticos creados con JUnit.

El objetivo principal es demostrar que el tratamiento a las cabeceras provenientes desde BCK y hayan sido identificado como “arqueables”, se tratan de la forma adecuada, es decir, guardando una lista de String en un mapa cuya key sea “ARQUEO”, que a su vez será seteado en el “requestData” del profile en la key “outpuheaders”.

# Recursos

* Test automáticos con JUnit
* Test manuales

# Definición

|  |  |
| --- | --- |
| **RTC / Proyecto** | **Leer desde MCA la información enviada por el BCK** |
| **Fecha de ejecución** | 18/09/2019 |
| **Entorno** | LOCAL, DSV |

# Criterios de aceptación

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Historias de usuario** | | | | **Criterios de Aceptación** | | |  |
| **Ámbito** | **Id.** | Título | **Descripción** | **Id.** | | **Escenario** | **Descripción** | |
|  |  |  |  |  | |  |  | |
| **{#histories}{scope}** | {#historiesUser}{name} | - | - | {index} | | {scope} | {/historiesUser}-{/histories} | |
|  |  |  |  |  | |  |  | |

# Tabla de resultados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
|  |  |  |  |  |  |
| **{#test}{tIndex}** | {historia} | {criterio} | {caso\_de\_prueba} | {descripcion} | **OK**{/test} |

# Tests Automáticos con Junit

{#scopes}

## Test en **{name}**

<DESCRIPCION>.

{#test}

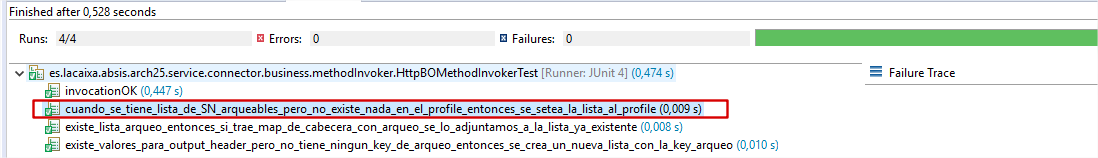
## **{caso\_de\_prueba}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
| {index} | {historia} | 1 | {caso\_de\_prueba} | {descripcion} | **OK** |

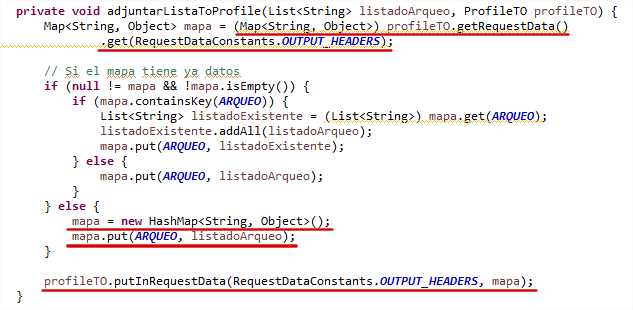
<DESCRIPCION>.

**TEST**

**{%image}**



**CLASE TRAZA:**

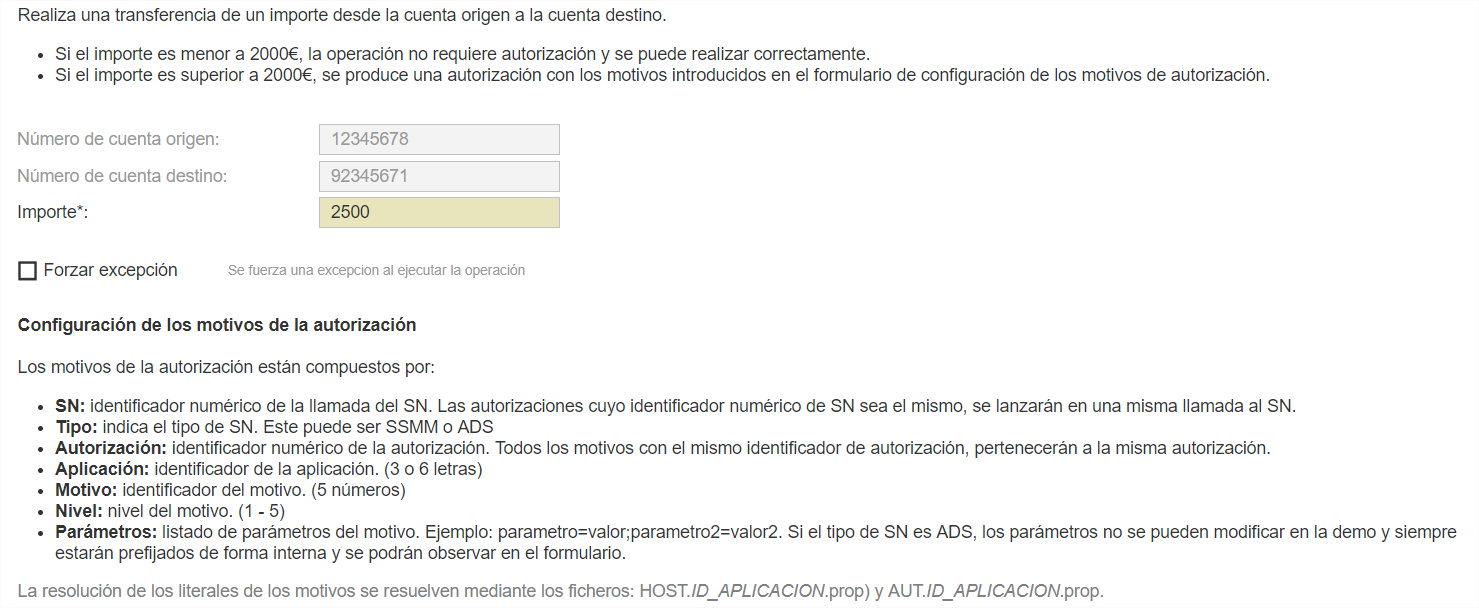


<OTRA\_DESCRIPCION>.

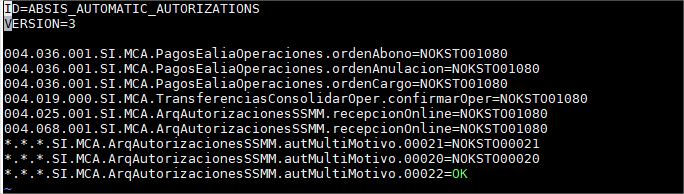
{/test}

{/scopes}

# Tests Manuales

Como se indicaba en la descripción, vamos a probar las respuestas dadas para los motivos de una autorización. Estas respuestas poder ser automáticas (OK o NOK) y no automáticas. Para ello simularemos la transferencia de un importe superior a 2000€, que requiere autorización.

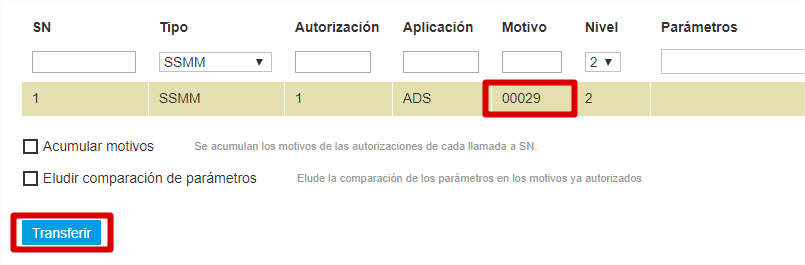
Previamente, el archivo de autorizaciones habrá sido cargado y podremos tener acceso desde BCK a las autorizaciones.



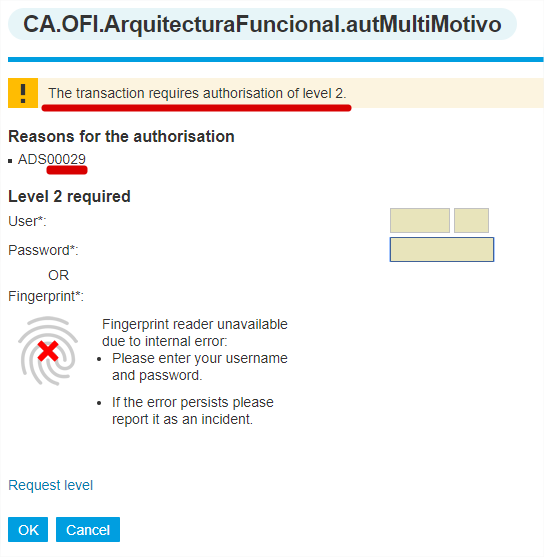
## Autorización sin motivo automático

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
| 14 | F-003 | 1 | Verificación del filtrado por un motivo que no es automático. | Se verifica que las autorizaciones con un motivo no automático funcionan correctamente. | **OK** |

Probamos con una autorización cuyo motivo es NO automático.



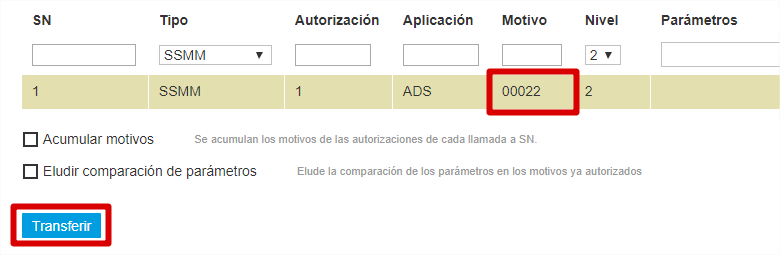
Nos pide una segunda autorización de forma manual por parte de un empleado con nivel 2.



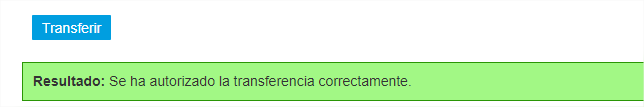
## Autorización con un motivo OK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
| 15 | F-003 | 1 | Verificación del filtrado por un motivo que es automático. | Se verifica que las autorizaciones con un motivo automático (OK) funcionan correctamente. | **OK** |

Probamos con una autorización cuyo motivo es AUTOMÁTICO



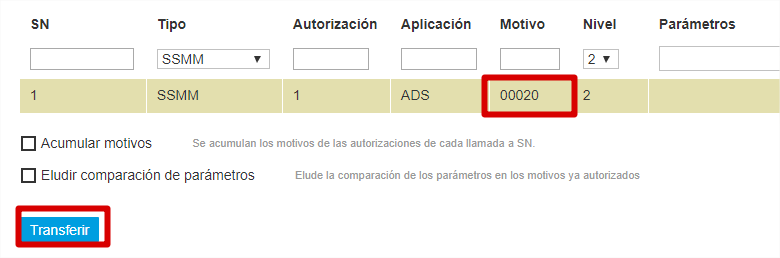
Se nos devuelve la siguiente respuesta:



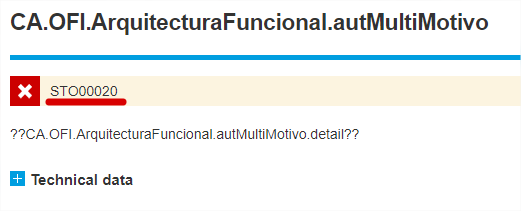
## Autorización con un motivo NOK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
| 16 | F-003 | 1 | Verificación del filtrado por un motivo que es automático pero NOK. | Se verifica que las autorizaciones con un motivo automático pero NOK funcionan correctamente. | **OK** |

Probamos con una autorización cuyo motivo es AUTOMÁTICO pero es un NOK.



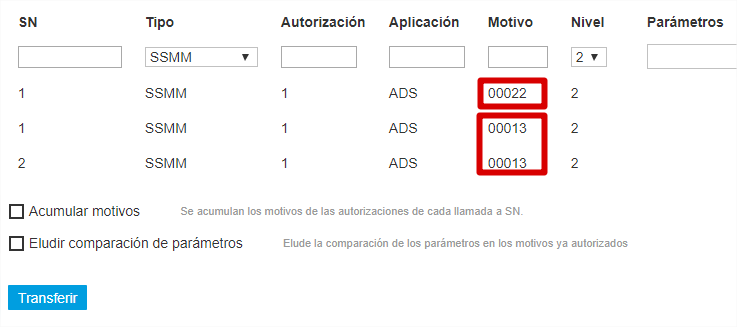
Devolviendo un error porque la transferencia no es posible para esa autorización



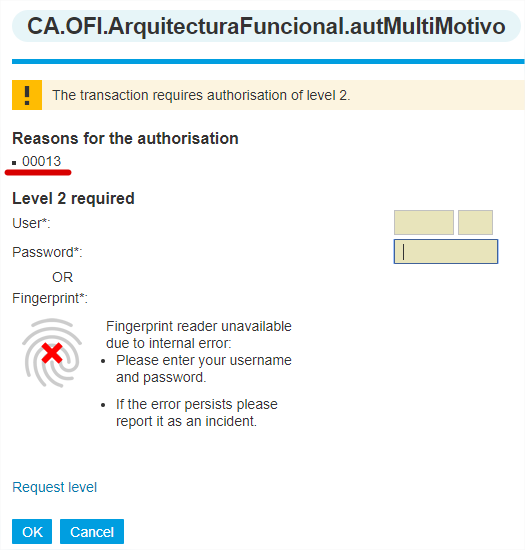
## Autorización con un motivo OK y dos no automático

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
| 17 | F-003 | 1 | Verificación del filtrado por un motivo que es automático OK y dos no automáticos. | Se verifica que las autorizaciones con un motivo automático OK y dos no automáticos funcionan correctamente. | **OK** |

Uno de los motivos es AUTOMÁTICO y con respuesta OK (00022) y otros dos, son NO automáticos (00013).



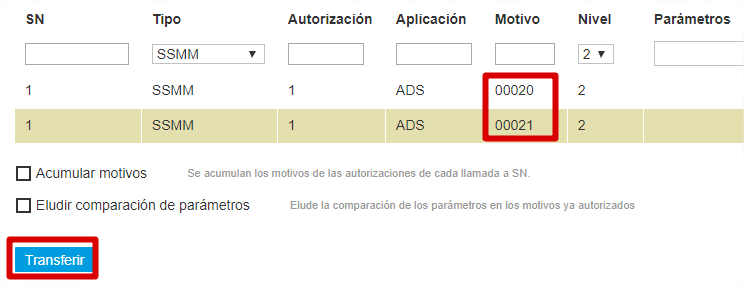
Como al menos existe uno que es NO automático, se va a solicitar que la autorización sea autorizada por un empleado con un nivel 2.



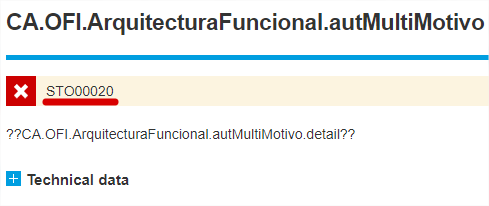
## Autorización con dos motivos NOK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
| 18 | F-003 | 1 | Verificación del filtrado de dos motivos NOK. | Se verifica que las autorizaciones con dos motivos automáticos NOK funcionan correctamente. | **OK** |

Probamos con una autorización que tiene dos motivos AUTOMÁTICOS y ambos son NOK.



La respuesta nos indica que la transferencia no se puede realizar para la primera de la autorización NOK que comprueba.



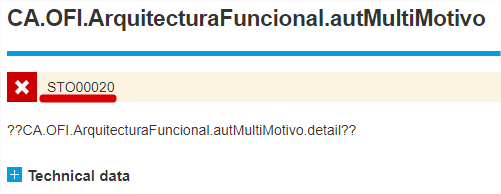
## Autorización con un motivo OK y uno NOK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
| 19 | F-003 | 1 | Verificación del filtrado de un motivo OK y otro NOK | Se verifica que las autorizaciones con un motivo automático OK y otro NOK funcionan correctamente. | **OK** |

Se indica dos motivos, ambos automáticos pero un motivo OK (00021) y otro NOK (00022)



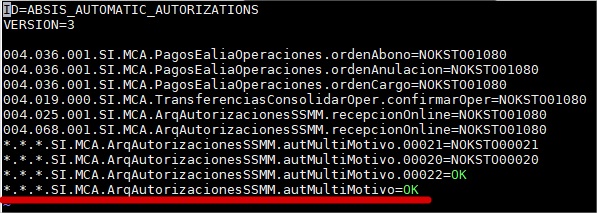
La respuesta indica que no se puede realizar la transferencia ya que uno de los motivos es NOK.



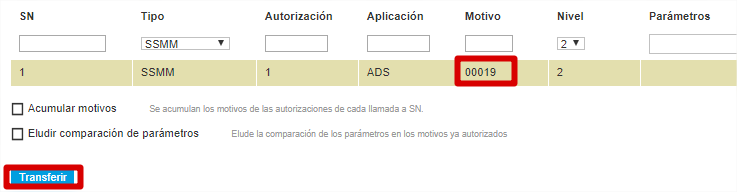
## Autorización con un motivo no automático, pero por defecto es Automático

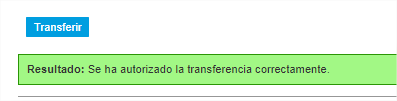
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Historia** | **Criterio** | **Caso de prueba** | **Descripción** | **Estado** |
| 20 | F-003 | 1 | Verificación del filtrado de un motivo inexistente pero por defecto es OK | Se verifica que una autorización que no es automática pero su servicio si lo es, devolverá un OK. | **OK** |

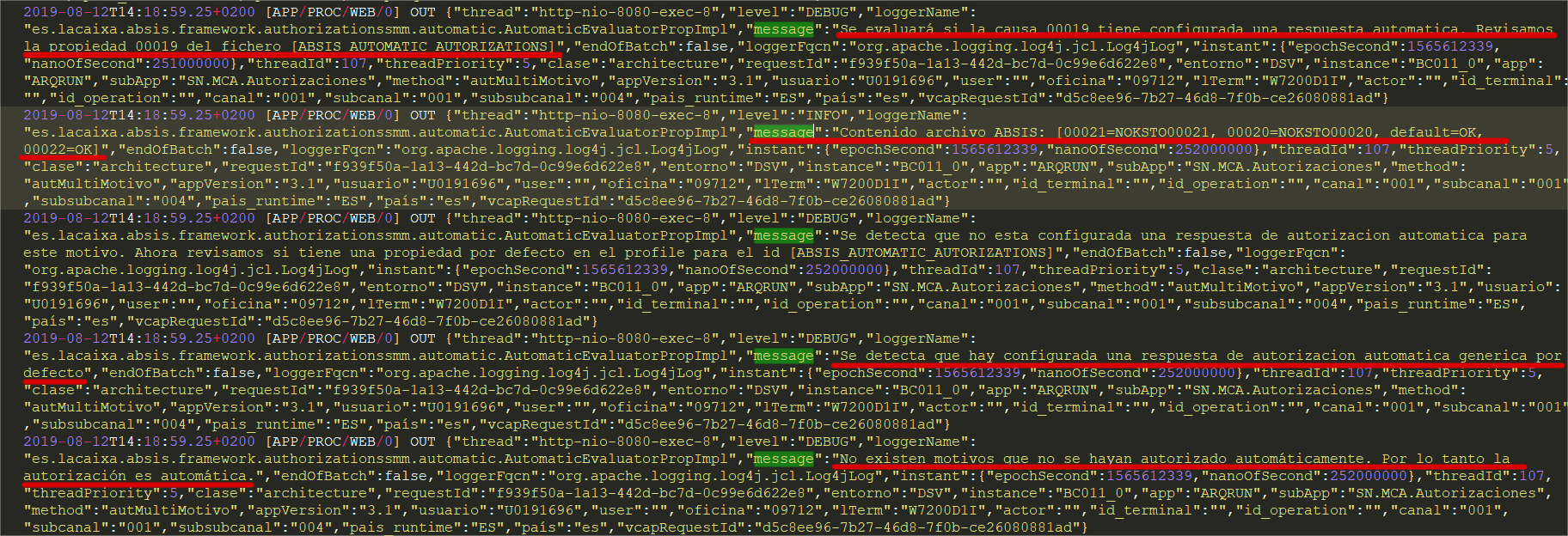
Se indica un motivo que no es automático pero por defecto, el servicio si es automático, por lo que no requiere de autorización de segundo nivel y la autorización es automática y OK.



Como se ve en la imagen, hemos añadido una nueva autorización para en caso de que no se encuentre la autorización entre los motivos dados, se compruebe la de por defecto y según su valor, devolver una respuesta. En este caso, indicado un OK para la autorización automática.





El log en consola nos muestra lo siguiente: