
This is the **published version** of the bachelor thesis:

Jara Vélez, Karel Gabriel; Benítez Fernández, Yolanda, dir. Desarrollo de un 'cockpit' empresarial basado en SAP Integration Suite. 2023. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/290104>

under the terms of the  license

Desarrollo de un 'cockpit' empresarial basado en SAP Integration Suite

Karel Gabriel Jara Velez

Resumen— Actualmente, muchos sectores empresariales que utilizan SAP como su ERP están usando el software PI/PO. Este quedará próximamente obsoleto, debido a que SAP dejará de mantenerlo a partir del año 2027, cosa que hará que múltiples empresas tengan que realizar una migración de este programa a otro. En el caso de que quieran mantenerse con SAP, la opción más viable es migrar a SAP Integration Suite con el fin de mantener todos sus sistemas e integraciones de forma fácil, debido a que este sistema ofrece muchas facilidades para realizar la migración. Esta es una herramienta muy potente, pero el monitoreo de sus mensajes puede llegar a ser confuso y complicado de rastrear, siendo esta la principal razón por la cual, desde nuestra empresa, se ha decidido apostar por la elaboración de una aplicación de monitoreo de mensajes, para facilitar el seguimiento de los datos. Este 'cockpit' empresarial permitirá una mejor visualización de los flujos de mensajes del sistema y su clasificación.

Palabras clave— React, Node.js, cockpit, monitoreo, aplicación, SAP, Integration Suite, BTP, Back-End, Front-End, Access, Fiori, flujos, personalización.

Abstract— Currently, multiple business sectors that use SAP as their ERP are using the software PI/PO. This software will be deprecated soon, because SAP will stop the maintenance of it from the year 2027, which consequently will cause that multiple companies will have to migrate from this software to another one. In case that they want to continue with SAP, the most viable option is migrating to SAP Integration Suite as they will keep all their systems and integrations, as this system offers a lot of facilities to execute the migration. This one is a very powerful tool, but the monitoring of its messages can be confuse and hard to track, being the main reason to, from our company, bet on the development of an application to monitor the messages, in order to ease the tracking of the data. This business cockpit will allow a better visualization of the message flows of the system and its classification.

Index Terms— React, Node.js, cockpit, monitoring, application, SAP, Integration Suite, BTP, Back-End, Front-End, Access, Fiori, flows, personalization.



1 INTRODUCCIÓN

Actualmente SAP Integration Suite consta de un monitor de mensajes poco intuitivo y que no facilita la lectura de los mensajes, debido a que simplemente muestra el nombre de los mensajes por Integration Flow y su correspondiente estado. Esta opción por defecto puede llevar a error, debido a que no se puede saber de primeras cuáles son los mensajes ligados al que estamos visualizando inicialmente, a no ser que se utilice la búsqueda por el campo 'CorrelationId'. Esto puede llevar a muchos clicks y es tedioso si hay un número de flujos masivo de forma constante, cosa muy común en los sistemas empresariales.

Estos flujos son clave en el caso de que haya errores o se quiera comprobar el correcto funcionamiento del sistema, y una mejor visualización de estos datos ayuda al usuario a entender mejor cómo se encuentra el sistema. Es por esta razón que se quiere diseñar una aplicación que

resuelva estos problemas de monitoreo de SAP Integration Suite, dando una herramienta de visualización de mensajes más intuitiva, personalizable y visual para facilitar el entendimiento de los datos y mejorar la experiencia del usuario.

2 ESTADO DEL ARTE

Para este proyecto se han contemplado diversas opciones:

1. Desarrollar la aplicación con las herramientas de SAP Build y SAP Business Application Studio. Esta opción permite desarrollar aplicaciones tanto en SAP Build Apps como en SAP Business Application Studio para luego ser desplegadas en SAP Build Workzone, un entorno compartido entre los miembros de la compañía en el cuál se pueden añadir aplicaciones personalizadas, mediante la exposición de una página web en el entorno o la generación de *Fiori Cards*, fácilmente desplegables.
2. Crear una aplicación con React y Node.js. Mediante el desarrollo en estos lenguajes la aplicación es mucho más personalizable y adaptable a las nece-

- E-mail de contacto: karel.jara.velez@gmail.com
- Mención realizada: Ingeniería del Software
- Trabajo tutorizado por: Yolanda Benítez Fernández (Área de Ciencias de la Computación e Inteligencia Artificial)
- Curso 2023/24

sidades empresariales, además de tener una gran escalabilidad para futuras actualizaciones. Se puede desplegar de una forma simple en nuestro sistema SAP y en cualquier otro que permita el despliegue de estas aplicaciones.

3. Utilizar el módulo de SAP Analytics Cloud para mostrar una pantalla de monitoreo correspondiente al SAP Integration Suite. Esta es una buena elección para las empresas que dispongan de esta herramienta, debido a que solo haría falta enlazar la API de SAP Integration Suite con esta. Pese a ello, se debe de tener la licencia y, además, la personalización es limitada.

Inicialmente se decidió optar por la primera opción, debido a la facilidad que conlleva el uso de estas herramientas para el desarrollo de aplicaciones. Además, la empresa tiene una licencia de estas, por lo que es factible el uso de estas.

Pese a ello, como se explicará más adelante en el desarrollo, se realizó un cambio de tecnología, escogiendo esta vez la segunda opción, usando React y Node.js.

3 OBJETIVOS

SAP Integration Suite es una plataforma que permite la integración de sistemas, comunicando unos con otros de forma que se puedan transferir mensajes entre ellos correctamente. Los mensajes se transmiten a través de *Integration Flows*, y cada uno que pasa a través de estos se ve reflejado en el monitoreo, controlando que se envían correctamente y reflejando si estos fallan. En grandes empresas esto puede suponer un reto, debido a que pueden tener una cantidad de mensajes muy elevada, dificultando el control de fallos y, además, presenta la dificultad de tener un control más visual sobre los distintos flujos de mensajes que pueda haber debido a que en la pantalla de monitoreo se pueden mezclar entre ellos al estar ejecutándose a la vez.

El objetivo principal del proyecto es desarrollar una aplicación de monitoreo que muestre los diferentes flujos de mensaje que se encuentran presentes en el sistema, en este caso SAP Integration Suite, permitiendo al usuario personalizarlos para que pueda identificarlos de una forma más sencilla, añadiéndoles un nombre identificativo. Con esto, se conseguirá un mayor entendimiento de los distintos mensajes que se están ejecutando en el sistema y el estado de estos, haciendo más fácil la labor de controlar los flujos.

A continuación, se hará un desglose de los distintos objetivos que se plantearon al iniciar el proyecto:

1. Mejorar la visualización de los datos con una interfaz simple y clara: actualmente SAP Integration Suite dispone de una interfaz de monitoreo no demasiado intuitiva y si se está ejecutando una cantidad masiva de mensajes es muy complicado tener un control de estos. Para mejorar esto, la

aplicación ha de ser intuitiva y ha de conseguir mejorar la visualización de los flujos de mensajes.

2. Personalización de la aplicación para adaptarla a las necesidades de la empresa y sus requisitos: la aplicación ha de permitir a los usuarios poder personalizar sus propios flujos de mensajes, debido a que estos varían entre las empresas.
3. Optimización de la gestión de los datos: como previamente se ha mencionado, el monitor de SAP Integration Suite no permite diferenciar los distintos flujos de una forma sencilla, por lo que nuestra aplicación debe diferenciarlos para poder optimizar la gestión de los distintos procesos.
4. Escalabilidad y rendimiento: la aplicación ha de ser capaz de gestionar una cantidad masiva de datos, manteniendo una fluidez correcta y una visualización de esta gran cantidad de datos adecuada. Se realizarán distintos filtros para que sea más sencillo realizar una búsqueda sobre los distintos flujos de mensajes.
5. Identificación de flujos de mensajes, diferenciándolos entre ellos según los artefactos que estén presentes en el flujo según como los designe el usuario. Es importante también que la arquitectura de los *Integration Flows* se adecue a este requisito para poder diferenciar los flujos.

4 METODOLOGÍA

En este proyecto la metodología que se ha usado es una metodología Agile, que se basa en un desarrollo iterativo.

A lo largo del proyecto se han realizado tres reuniones de seguimiento con la empresa para comprobar el correcto desarrollo de la aplicación. Estas reuniones no se realizaron de una forma periódica debido a que no ha habido una fecha establecida para estas desde el inicio del desarrollo. Pese a ello, cada una de estas se realizaron con un margen aceptable entre ellas para comprobar los avances en el proyecto y los posibles cambios de este. A continuación, se detallará el contenido de cada una:

1ª reunión: en esta se revisó la viabilidad de desarrollar la aplicación en los entornos de SAP Build y SAP Business Application Studio. Se decidió que eran las tecnologías adecuadas para el desarrollo del proyecto.

2ª reunión: a partir de unos contactos con otras empresas, se discutió sobre el desarrollo del proyecto con SAP Build y SAP Business Application Studio, debido a que inicialmente no se tuvieron en cuenta otras tecnologías. Las herramientas anteriormente mencionadas eran algo limitadas y se necesitaba disponer de una licencia de SAP Build Apps debido a que con la versión gratuita solo se permite hacer dos despliegues, razón por la cual la aplicación no podría mantenerse ni recibir actualizaciones

sin invertir en la licencia, dificultando su comercialización. Además, esta tecnología no es tan flexible y limita al desarrollador. Es por ello que se decidió cambiar la tecnología para desarrollar la aplicación en React y Node.js, siendo esta más versátil y escalable.

3ª reunión: en esta reunión se comprobó el estado del proyecto, mostrando el desarrollo realizado.

Dado que se ha usado esta metodología, a partir de estas reuniones se ha podido actuar delante de algún imprevisto y añadir funcionalidades que inicialmente no se habían planteado, siendo así el desarrollo bastante versátil frente a los cambios.

5 PLANIFICACIÓN

La planificación inicial del proyecto ha contado con cuatro apartados distintos:

1. Entregas: se especifican las diferentes entregas del proyecto de fin de grado.
2. Teoría: en esta se complementan las distintas tecnologías de las cuáles se ha tenido que adquirir conocimientos.
3. Práctica: se mencionan las distintas labores técnicas a realizar durante el desarrollo de la aplicación, especificando las metas a alcanzar y el tiempo estimado que se necesita para completarlas.
4. Presentaciones: se especifica el tiempo estimado que llevará la preparación de la presentación.

Esta planificación ha variado durante el proyecto, debido a varios imprevistos durante el desarrollo. En los apéndices A1 y A2 se pueden observar el diagrama de Gantt inicial y el diagrama de Gantt final, respectivamente.

Este cambio de planificación ha sido provocado debido al cambio de tecnología en la cual se ha desarrollado la aplicación. Como se puede observar, en la semana 11 del proyecto se tiene que ha tenido que detener el desarrollo de la aplicación en SAP Build y SAP Business Application Studio para empezar el desarrollo en React y Node.js. Este cambio es debido a que las diferentes empresas pueden no tener una licencia de SAP Build, y pese a que hay una versión gratuita de estas herramientas, solo se permitirían hacer dos despliegues de una aplicación, limitando así el mantenimiento de esta.

En cambio, con React y Node.js se puede desplegar la aplicación todas las veces que sea necesario, solventando el problema del mantenimiento.

6 REQUISITOS

Con el fin de alcanzar los objetivos anteriormente descritos se han definido los siguientes requisitos, divididos entre requisitos funcionales y no funcionales.

6.1 Requisitos funcionales

Se han expuesto los siguientes requisitos funcionales para que la aplicación cubra las principales necesidades y se puedan alcanzar los objetivos:

1. Monitorización de los datos del sistema: la aplicación ha de presentar por pantalla los distintos flujos de mensajes de forma clara.
2. Búsquedas de mensajes por intervalos de fechas: el usuario ha de poder realizar una búsqueda entre fechas concretas, para poder visualizar los flujos de mensajes que se encuentren en ese rango.
3. Opciones de personalización: se ha de permitir que el usuario pueda personalizar los distintos flujos de datos.
4. Interfaz de usuario intuitiva: la aplicación ha de destacar por su usabilidad y fácil manejo, permitiendo que nuevos usuarios la puedan usar de forma sencilla.
5. Capacidad de ordenar los datos: la aplicación ha de permitir ordenar los datos según varios parámetros, tales como la ordenación por fechas, de forma alfabética y por sistemas remitentes o receptores.
6. Mostrar información de los mensajes: se ha de mostrar información sobre los distintos mensajes que conforman un flujo.
7. Análisis de los datos en tiempo real: la aplicación ha de permitir actualizar los datos y también realizar una actualización según un tiempo marcado.
8. Búsquedas según escritura sobre los diversos campos que se muestren: el usuario podrá realizar una búsqueda por texto que filtrará los flujos según el texto introducido.

6.2 Requisitos no funcionales

Se han determinado los siguientes requerimientos no funcionales:

1. Procesamiento de datos masivos sin que el rendimiento de la aplicación se vea afectado, permitiendo una correcta visualización de los datos y que se mantenga la fluidez de la aplicación.
2. Accesible para usuarios de la organización, permitiendo que accedan a la aplicación de forma independiente cada uno de ellos.
3. Tolerancia a fallos, la aplicación ha de controlar los distintos fallos que se pueden producir.
4. Debe ser fácilmente configurable para su uso en

otros entornos empresariales, debido a que el consumo de datos se basa en la API de SAP Integration Suite, esta es distinta por cada entorno empresarial.

7 ANÁLISIS Y DESARROLLO

7.1 Casos de uso

Para empezar con el desarrollo se definen los siguientes casos de uso:

- Búsqueda de mensajes según un intervalo de fechas, el usuario podrá introducir un rango de fechas y se podrán buscar los mensajes ejecutados dentro de este, facilitando la búsqueda para fechas concretas.
- Actualizar la tabla de datos, ha de permitir al usuario actualizar los datos cuando este quiera, disponiendo de un botón de actualización.
- Ordenar de mayor a menor o viceversa la cantidad de mensajes según su estado.
- Ordenar alfabéticamente los nombres de los flujos que realizan los mensajes.
- Mostrar los mensajes según el estado seleccionado, teniendo la posibilidad de hacer una búsqueda más concreta sobre el estado de estos.
- Mostrar todos los mensajes de un flujo, al seleccionar un flujo se deben mostrar todos los mensajes relativos a este, con sus correspondientes estados. En el caso que se produzca un error, se podrá identificar de forma más rápida qué mensaje falla.
- Mostrar la información de un mensaje, viendo su contenido, estado y fecha de ejecución.
- Configurar la aplicación, el usuario puede configurar la aplicación según sus necesidades, dando versatilidad.
- Filtrar los flujos mediante una búsqueda, el usuario puede introducir texto para realizar una búsqueda que coincida con el texto escrito según distintos parámetros.

7.2 Inicio del desarrollo en SAP Build

Como se ha comentado previamente, el comienzo del proyecto empezó con un desarrollo en las plataformas de SAP Build y SAP Business Application Studio, resaltando sobretudo la primera. Para empezar, se ha tenido que crear una *Destination*, la cual permitiría a nuestra aplicación acceder a los datos de SAP Integration Suite y a todos sus mensajes. La creación de esta la podemos observar en el apéndice A3.

Una vez configurada, se puede proceder a crear la aplicación en SAP Build Apps. En este programa, se accede al apartado *Data*, que será donde se asocie la *Destination* previamente creada. En el apéndice A4 se puede ver la *Destination* añadida.

Una vez añadida se tendrá acceso a todos los

endpoints de la API de SAP Integration Suite. El endpoint que se usará en este proyecto será el de *MessageProcessingLogs*, dado que ofrece toda la información de los mensajes del sistema y permitirá moldearlos para extraer los datos deseados. Se ha de activar para que pueda ser usado.

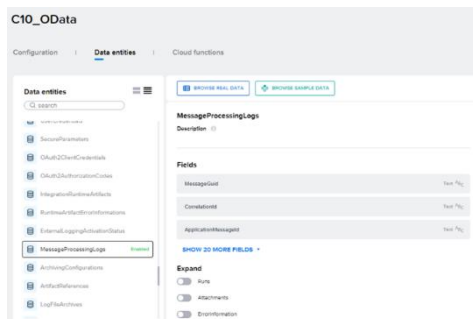


Ilustración 1. Se muestra el endpoint de *MessageProcessingLogs* activado

Una vez hecho, se asocia a una Data Variable, que hará una llamada a la API para obtener todos los datos de los mensajes. Uno de los problemas que se han encontrado es la estructura de los datos, debido a que esta era muy difícil de compaginar con el monitor que se quería desarrollar, ya que es un *array* que contiene toda la información de cada mensaje individualmente y en nuestro caso se quieren juntar todos los mensajes por *Integration Flow*, para clasificar el número de mensajes por estado, dando una visualización rápida de estos.

Hay varias herramientas en SAP Build Apps para poder tratar los datos, pero se ha preferido realizar un script en JavaScript para transformar la estructura de datos de origen a otra que facilita el monitoreo que queremos debido a que las opciones predeterminadas no eran suficientes como para poder transformar efectivamente los datos a la estructura deseada.



Ilustración 2. Definición de la Data Variable

En la ilustración previa se observa la definición de *GetMessageProcessingLogs*, variable que contendrá toda la información de los mensajes, obteniendo estos datos haciendo una llamada a la API.

Además, se incluye el script mencionado anteriormente, que es el siguiente:

```
1 const originData = inputs.dataSrc;
2
3 const groupedData = {};
4
5 originData.forEach(item => {
6   const integrationFlowName = item.IntegrationFlowName;
7
8   if (!groupedData[integrationFlowName]) {
9     groupedData[integrationFlowName] = {
10       IFlow: integrationFlowName,
11       Details: []
12     };
13   }
14   groupedData[integrationFlowName].Details.push(item);
15 });
16
17 const resultArray = Object.values(groupedData);
18
19 return ({GroupedMessages: resultArray});
```

Ilustración 3. Conversión de la estructura de datos de origen a la estructura deseada

Con esta variable se puede empezar a desarrollar el monitor. Para ello, se utiliza una fila con columnas y con la opción *Repeat with* se podrán recorrer todas las posiciones, mostrando en cada fila un *Integration Flow* junto a el número de mensajes por estado. Esto ha sido posible debido a la funcionalidad de Formulas dentro de SAP Build, permitiendo manejar la variable *GetMessageProcessingLogs* para que muestre los datos de la forma que queremos.

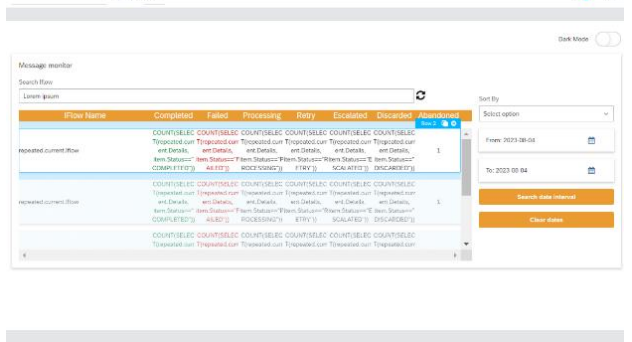


Ilustración 4. Monitor sin ejecutar en SAP Build, se muestran las distintas fórmulas por cada campo

Se han implementado distintos filtros, uno por introducción de texto, que mostrará por pantalla los *Integration Flows* que contengan el texto introducido por el usuario y otros filtros por orden alfabético para la columna IFlow Name o numérico para el resto, clicando en las cabeceras de la tabla, ordenando de forma descendente en primera instancia y, si se vuelve a hacer clic, de forma ascendente.

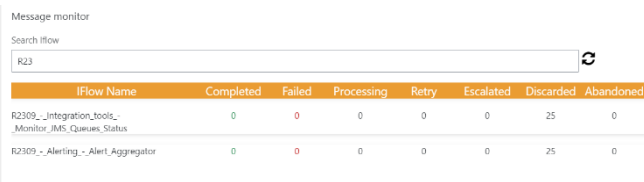


Ilustración 5. Búsqueda por texto en SAP Build

Message monitor

Search IFlow

IFlow Name	Completed	Failed	Processing	Retry	Escalated	Discarded	Abandoned
Notify_Certificate_Expiration	45	41	0	0	0	111	0
KSelf_Dispatcher	29	0	0	0	0	0	0
KSelf_Inbound	22	0	0	0	0	0	0
Email_Alert_Sender	15	0	0	0	0	0	0
Integration_tools_-_Check_erroneous_artifacts	9	5	0	0	0	14	0
KSelf_Outbound	7	0	0	0	0	0	0
Sign_Invoice_TBAl	3	0	0	0	0	0	0
Send_Invoice_TBAl	2	0	0	0	0	0	0
Z17PFIN1_Kubewar_Moudfiar	2	0	0	0	0	0	0

Ilustración 6. Ordenación por cantidad de mensajes en estado *Completed*

También se ha añadido la posibilidad de hacer una búsqueda entre fechas. Se pueden seleccionar dos fechas y, una vez seleccionadas, se selecciona el botón *Search date interval* para hacer una llamada a la API y que devuelva los mensajes entre ese rango de fechas.

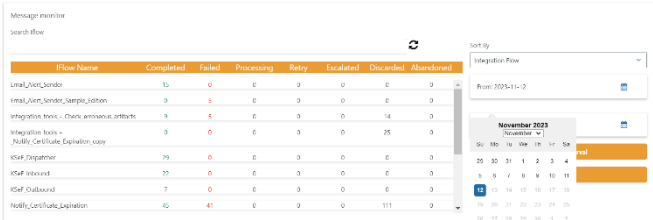


Ilustración 7. Desplegables fechas

Si se selecciona alguna de las numeraciones se pasará a otra pantalla que mostrará los mensajes que contengan ese estado del respectivo *Integration Flow*, y si se selecciona este último se mostrarán todos los mensajes del correspondiente. En el caso de que se seleccionen los mensajes fallados, además del detalle del mensaje también se mostrará el mensaje de error.

Message History

ALL messages

IFlow Name	Status	LogStartDate	LogEndDate	URL	Information
Email_Alert_Sender	COMPLETED	2023-10-23T16:37:37	2023-10-23T16:37:39	Go to SAP message log	
Email_Alert_Sender	COMPLETED	2023-10-23T16:33:23	2023-10-23T16:33:24	Go to SAP message log	
Email_Alert_Sender	COMPLETED	2023-10-23T16:31:09	2023-10-23T16:31:10	Go to SAP message log	
Email_Alert_Sender	COMPLETED	2023-10-23T16:26:15	2023-10-23T16:26:16	Go to SAP message log	
Email_Alert_Sender	COMPLETED	2023-10-23T16:17:14	2023-10-23T16:17:16	Go to SAP message log	
Email_Alert_Sender	COMPLETED	2023-10-23T15:24:12	2023-10-23T15:24:14	Go to SAP message log	
Email_Alert_Sender	COMPLETED	2023-10-23T15:22:18	2023-10-23T15:22:19	Go to SAP message log	
Email_Alert_Sender	COMPLETED	2023-10-23T15:11:08	2023-10-23T15:11:09	Go to SAP message log	
Email_Alert_Sender	COMPLETED	2023-10-23T15:10:03	2023-10-23T15:10:03	Go to SAP message log	

Ilustración 8. Listado de todos los mensajes de un *Integration Flow*

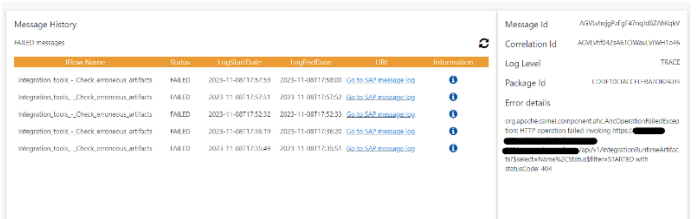


Ilustración 9. Detalles del mensaje y mensaje de error

Todas las funcionalidades descritas anteriormente son las que se llegaron a desarrollar antes del cambio de tecnología. De ahora en adelante, se detallará el desarrollo de la aplicación en React y Node.js.

7.3 Desarrollo en React y Node.js

El cambio de tecnología se realizó en una fase intermedia/avanzada del proyecto. Se tenía un desconocimiento sobre las nuevas herramientas que se iban a usar, pero poco a poco se fue adaptando. El planteamiento era el mismo que el de la anterior aplicación, pero se añadió un objetivo más, que sería el de mostrar los flujos de mensajes. Además, es importante mencionar que, para que la aplicación sea más potente, es necesario que en los *Integration Flows* que se realicen se añadan los llamados *Custom Log Headers*. Estos, si se añaden, aparecen en los monitoreos de los mensajes y ayudan a identificar de forma más fácil los mensajes. Los *Custom Log Headers* se pueden añadir incluyendo una propiedad en el flujo y un Groovy Script, como se indica a continuación:

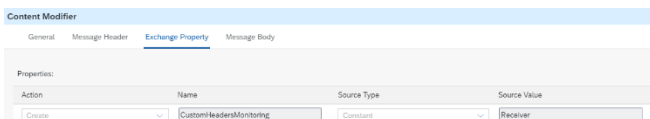


Ilustración 10. Se añade la propiedad Custom-Log-HeadersMonitoring

```
def Message addCustomMessageLogsFromHeadersAndProperties(Message message) {
    def messageLog = messageLogFactory.getMessageLog(message);
    def properties = message.getProperties();
    def headers = message.getHeaders();

    def search_headers = properties.get("CustomLogHeadersMonitoring");
    def search_properties = properties.get("CustomLogPropertiesMonitoring");

    if (search_headers != null) {
        String[] headers_moni = search_headers.split("#");
        for (int i=0; i<headers_moni.length; i++) {
            if (headers.get(headers_moni[i]) != null) {
                messageLog.addCustomHeaderProperty(headers_moni[i], headers.get(headers_moni[i]));
            }
        }
    }

    if (search_properties != null) {
        String[] properties_moni = search_properties.split("#");
        for (int i=0; i<properties_moni.length; i++) {
            if (properties.get(properties_moni[i]) != null) {
                messageLog.addCustomHeaderProperty(properties_moni[i], properties.get(properties_moni[i]));
            }
        }
    }

    return message;
}
```

Ilustración 11. Se incluye este Groovy Script en el flujo

Se pueden añadir la cantidad deseada de propiedades, siempre y cuando se separen mediante un '#' y exista un header con ese nombre. Para nuestra arquitectura de datos, añadiremos los siguientes *Custom Headers*:

- Sender
- Receiver
- IDOCNumber

Con estos podremos obtener los datos del remitente del mensaje y a que sistema lo envía, proporcionando información útil al usuario. Para mantener estos headers, se ha de permitir la transmisión de estos a lo largo de las distintas capas que tenga el flujo, usando el campo

Allowed Headers. También es importante que los *Integration Flows* estén relacionados entre sí de una manera fácil de entender para que a la hora de definir el flujo de mensajes en la aplicación sea más simple.

El editor sobre el cual se ha desarrollado la aplicación es Visual Studio Code, dada su versatilidad en cuanto a los diferentes lenguajes que soporta y el poder usar las terminales de forma más sencilla. Se comenzó generando una aplicación de React, generando una estructura Cliente-Servidor, definiendo el directorio del Cliente como *frontend* y el directorio del servidor como *backend*. Esto ha facilitado el desarrollo debido a que se han podido realizar muchas pruebas de la aplicación el local, comprobando en todo momento que el estado de esta era el correcto y que cada funcionalidad añadida coexistía con el resto.

En el apéndice A5 se puede observar la estructura de archivos final.

En el backend se encuentra todo lo relativo al servidor, las llamadas a las APIs y la lectura de archivos de bases de datos, que se comentará más adelante. En el archivo *server.js* se definen todas las rutas y el puerto que usará la aplicación. En este caso, definiremos los *MessageLogs*, que devolverán la información de los mensajes, y el *getDatabase*, que recogerá los distintos flujos de una base de datos de Access. El archivo queda así:

```
backend > JS server.js >
1  const express = require('express');
2  const path = require('path');
3  const app = express();
4  const port = process.env.PORT || 3001;
5  const messageRoute = require('./routes/messageLogs');
6  const runtimeRoute = require('./routes/runtimeArtifacts');
7  const databaseRoute = require('./routes/getDatabase');
8
9  // Use the routes in the application
10 app.use('/api', messageRoute);
11 app.use('/api', runtimeRoute);
12 app.use('/api', databaseRoute);
13
14 app.listen(port, () => {
15   console.log(`Server listening on port ${port}`);
16 });
17
18 if (process.env.NODE_ENV === 'production') {
19   app.use(express.static('client/build'));
20
21   app.get('*', (req, res) => {
22     res.sendFile(path.resolve(__dirname, 'client', 'build', 'index.html'));
23   });
24 }
```

Ilustración 12. Archivo server.js

En la carpeta de frontend definiremos todo lo relativo a este. El archivo más importante es el *App.js*, que contendrá los distintos elementos que se mostrarán por pantalla. Este archivo es el principal esqueleto de la aplicación, y dentro de este podremos incluir distintas funcionalidades a medida que se desarrolle el proyecto.

La aplicación consta de una tabla que mostrará los distintos flujos de mensajes. Esto se puede realizar debido a que los mensajes de SAP Integration Suite contienen un campo llamado *CorrelationId*, que es clave a la hora de mostrar los mensajes relacionados entre sí. Por esta razón

9 CONCLUSIONES

El cambio de tecnología en mitad del proyecto resultó inicialmente duro, debido al desconocimiento sobre React y Node.js, ya que nunca había desarrollado en estos lenguajes.

Esta transición implicó, además de el aprendizaje en un corto periodo de tiempo sobre estas tecnologías, un cambio en la perspectiva del proyecto, debido a que, como se puede observar en el desarrollo de SAP Build Apps, el enfoque de relacionar los mensajes según los distintos flujos no estaba presente, incluyendo solo una mejora visual del actual monitor de SAP Integration Suite.

Pese a ello, se ha logrado conseguir una aplicación que actualmente es funcional, desde la cual se puede seguir desarrollando y que cumple con el objetivo principal de este proyecto, el mostrar una clasificación de mensajes según su flujo y poder personalizar estos, haciendo que el monitoreo de los mensajes sea más visual.

Este proyecto me ha ayudado a manejarme en las distintas tecnologías planteadas, adquiriendo una gran variedad de conocimientos y, sobretodo, las distintas perspectivas que se pueden dar según la tecnología que se use. SAP Build Apps es una plataforma algo más limitada, debido a que se basa en Low Code/No Code, enfocada en personas que no tienen muchos conocimientos en programación, pero creo que sigue siendo una buena herramienta para programadores más experimentados. Por otro lado, el desarrollo en React y Node.js ha sido más complejo pero estoy contento con los resultados obtenidos.

AGRADECIMIENTOS

Quiero agradecer a mi tutora del TFG, Yolanda Benítez, por su guía y paciencia a la hora de las entregas de los informes, a mi familia por darme la posibilidad de haber podido estar en la universidad y a todas las personas que he conocido a lo largo de estos años, que han hecho mucho más llevadera la vida universitaria.

BIBLIOGRAFÍA

- [1] 'SAP Business Accelerator Hub', SAP, <https://api.sap.com/>
- [2] 'SAPUI5 SDK', SAP, <https://ui5.sap.com/>
- [3] 'SAP Build', SAP, https://help.sap.com/docs/SAP_BUILD?locale=en-US
- [4] Kapoor, S. (2023, 17 de agosto). Interactive Reporting Dashboard for SAP Cloud Integration using SAP Analytics Cloud. SAP Blogs. <https://blogs.sap.com/2020/08/26/sap-analytics-cloud-content-for-sap-cloudplatform-integration/>
- [5] 'UI5 Web Components for React', SAP SE and UI5 Web Components for React Contributors, <https://ui5-webcomponents-react.netlify.app/?path=/docs/gettingstarted--docs>
- [6] 'Node.js tutorial in Visual Studio Code', Visual Studio Code, <https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>
- [7] Van der Neut, S. (2020, 1 de enero). Using SAP UI5 Web Components with ReactJS. Acorel.

- <https://www.acorel.nl/2020/01/using-sap-ui5-web-components-with-reactjs/>
- [8] 'Build a Single Page Application Using UI5 Web Components for React', SAP, <https://developers.sap.com/mission-react-spa.html>
- [9] Precious, L. (2021, 15 de diciembre). Building and structuring a Node.js MVC application. LogRocket blog. <https://blog.logrocket.com/building-structuringnode-js-mvc-application/>
- [10] Bechtol, E. (2020, 21 de enero). Node Service-oriented Architecture. Codementor. <https://www.codementor.io/@evanbechtol/node-service-orientedarchitecture-12vjt9zs9i>
- [11] 'NPM', npm, Inc., <https://www.npmjs.com/>
- [12] 'React Flow', xyflow, <https://reactflow.dev/>
- [13] Mu, S. (2021, 9 de septiembre). How I structure full stack web application with react and nodejs. Medium. <https://steve-mu.medium.com/how-i-structure-full-stack-web-application-with-react-and-nodejs-b8ec9d2bc9b4>

A1. Diagrama de Gantt inicial

[illegible][illegible]

The screenshot displays the SAP BTP Cockpit interface, specifically the 'Destinations' configuration page. The left sidebar contains navigation links for Overview, Services, Service Marketplace, Cloud Foundry, HTML5 Applications, Connectivity, Destinations (highlighted), Cloud Connectors, Security, Help and Support, Useful Links, and Legal Information. The main content area shows a table of destinations with columns for Name, Type, Description, URL, Proxy Type, Authentication, User, and Password. Three destinations are listed: 'sap-build-apps-runtime' (Type: HTTP, URL: https://*), 'swg_contentrepo' (Type: HTTP, URL: https://*), and 'worklon_dest' (Type: HTTP, URL: https://*). Below the table is the 'Destination Configuration' section for the selected 'sap-build-apps-runtime' destination. It shows details like Name (CSD_OData), Type (HTTP), Description (CSD OData Service), URL (https://*), Proxy Type (Internet), Authentication (Basic Authentication), User (kurt.jaraj@cod3308.com), and Password (*****). To the right of this section is the 'Additional Properties' section, which lists properties like HTML5 Dynamic, sap-process-id, webIDEEnabled, webIDESystem, webIDEUsage, and editio_hsp_sap_odata_h_*. The bottom of the page has a footer with links: Edit, Clone, Export, Delete, and Check Connection.

The screenshot shows the SAP Message Monitor interface. The top navigation bar includes the SAP logo and the title 'Message Monitor'. Below the navigation bar, there are filters for 'View type', 'Start Date', 'End Date', 'Search date', 'Update Message', 'Order by Sender', and 'Filter by Receiver'. A 'Go to configuration' button is located on the right. The main table displays a list of messages with columns: Processed, CorrelationID, IDOCNumber, Sender, Receiver, Status, and Logfile. The 'Status' column shows 'RETRY' for the first three rows and 'RECEIVED' for the rest. A 'Database Data' dialog box is open in the center, showing a table with columns: Processed, Layer, MessageType, PkType, and Out. The dialog box has a 'Search On...' input field, a 'Reset Filter' button, and a 'Close' button.

Processed	CorrelationID	IDOCNumber	Sender	Receiver	Status	Logfile
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	CDBreaker Labapze	RETRY	20-10-2022T09:08:58.000
	ADH094W72e773uagQ0CMV00	00000000000000000000	SAP	Labapze CDBreaker	RETRY	20-10-2022T09:08:58.000
	ADH094W7					

A8. Archivo Access cargado

The screenshot shows the SAP Message Monitor interface. A modal window titled 'Database Data' is open, displaying a table with columns: ProcessId, Layer, Message Type, PkType, and QoS. The table contains three rows of data. Below the table are buttons for 'Upload File', 'Close', and 'Download'.

ProcessId	Layer	Message Type	PkType	QoS
DEBMS	EDBMS_Sender_JOC_Chester	EDBMS_Sender_JOC_Chester	EDBMS_Sender_JOC_Chester	High
DEBMS	EDBMS_Sender_JOC_Chester	EDBMS_Sender_JOC_Chester	EDBMS_Sender_JOC_Chester	Medium
DEBMS	EDBMS_Sender_JOC_Chester	EDBMS_Sender_JOC_Chester	EDBMS_Sender_JOC_Chester	Low

A9. Filtrado por texto

The screenshot shows the SAP Message Monitor interface with a list of messages. The 'Filter by' field is set to 'de'. The table displays columns: ProcessId, CorrelationId, IDOCNumber, Sender, Receiver, Status, and LogEnd. The messages are filtered by the 'de' keyword.

ProcessId	CorrelationId	IDOCNumber	Sender	Receiver	Status	LogEnd
DEBMS	AGV4RPF2m75uqGQCMVGV	00000000353806	SAP	EDBMS_Sender_JOC_Chester	RETRY	22-12-2023T10:09:18.809
DEBMS	AGV5HfPQKzXvVSTPhGBvZ	00000000353808	SAP	EDBMS_Sender_JOC_Chester	RETRY	22-12-2023T09:44:11.057
DEBMS	AGW4HbBV DkXaORAlucSVUWRJC	00000000353809	SAP	EDBMS_Sender_JOC_Chester	RETRY	22-12-2023T09:41:37.271
DEBMS	AGV4TKZqAqGzSpQzGIBVpG	00000000353807	SAP	EDBMS_Sender_JOC_Chester	RETRY	22-12-2023T09:33:36.782
DEBMS	AGW4HbBV DkXaORAlucSVUWRJC	00000000353806	SAP	EDBMS_Sender_JOC_Chester	RETRY	22-12-2023T09:23:36.046
DEBMS	AGW4HbBV DkXaORAlucSVUWRJC	00000000353806	SAP	EDBMS_Sender_JOC_Chester	RETRY	22-12-2023T09:23:36.054
DEBMS	AGW4HbBV DkXaORAlucSVUWRJC	00000000353806	SAP	EDBMS_Sender_JOC_Chester	RETRY	22-12-2023T09:23:36.056
DEBMS	AGW4HbBV DkXaORAlucSVUWRJC	00000000353806	SAP	EDBMS_Sender_JOC_Chester	RETRY	22-12-2023T09:23:36.058
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				CANCELLED	20-12-2023T11:10:56.052
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	20-12-2023T11:10:56.051
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.153
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.154
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.155
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.156
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.157
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.158
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.159
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.160
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.161
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.162
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.163
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.164
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.165
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.166
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.167
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.168
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.169
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.170
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.171
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.172
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.173
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.174
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.175
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.176
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.177
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.178
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.179
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.180
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.181
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.182
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.183
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.184
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.185
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.186
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.187
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.188
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.189
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.190
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.191
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.192
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.193
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.194
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.195
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.196
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.197
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.198
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.199
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.200
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.201
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.202
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.203
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.204
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.205
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.206
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.207
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.208
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.209
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.210
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.211
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.212
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.213
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.214
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.215
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.216
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.217
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.218
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.219
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.220
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.221
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.222
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.223
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.224
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.225
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.226
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.227
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.228
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.229
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.230
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.231
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.232
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.233
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.234
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.235
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.236
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.237
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.238
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.239
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.240
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.241
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.242
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.243
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.244
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.245
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.246
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.247
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.248
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.249
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.250
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.251
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.252
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.253
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.254
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.255
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.256
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.257
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.258
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.259
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.260
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.261
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.262
DEMO	AGWCC1qgX8QaRkLdQpVn7Nv				COMPLETED	19-12-2023T22:25:07.263
DEMO	AGWCC1qgX8QaRkLd					

A11. Filtro según *Sender*

Monitoring App

localhost:3000

SAP

Message Monitor

Filter by: Start Date: YYYY-MM-dd End Date: YYYY-MM-dd Search dates Update Messages Filter by Sender: SAP Filter by Receiver: Go to configurations

ProcessId	CorrelationId	IDOCNumber	Sender	Receiver	Status	LogEnd
DEBMA5	AGV4RyPY2mZL5wspQ24CMY6V	0000000003539856	SAP	EDIBroker Hubspot	RETRY	22-12-2023T10:09:18.809
DEBMA5	AGV5HfImQkbaXhVABTbfCIRvZ	0000000003539858	SAP	Hubspot EDIBroker	RETRY	22-12-2023T09:44:12.057
DEBMA5	AGWAH6B6V-DkXaORuMoz5XUWRJC	0000000003539859	SAP	EDIBroker Hubspot	RETRY	22-12-2023T09:41:37.271
DEBMA5	AGV4TX2XykgLxSpDpuZGhGBYP6	0000000003539857	SAP	EDIBroker Hubspot	RETRY	22-12-2023T09:33:38.782
DEBMA5	AGWAN8vbmC5ykyvYH14t_Yo7	0000000003539861	SAP	Hubspot EDIBroker	RETRY	22-12-2023T09:23:36.046
DEBMA5	AGWAKcZ4FvWk45XaXzCuhLJocl	0000000003539860	SAP	Hubspot EDIBroker	RETRY	22-12-2023T09:23:35.954
DEBMA5	AGWAN-D9Ld0GwA9PwPjFjowYIR	0000000003539863	SAP	EDIBroker Hubspot	RETRY	22-12-2023T09:23:35.950
	AGWAN5Yhc6YvdtbP_UQeZiFM4X	0000000003539862	SAP	Hubspot EDIBroker	COMPLETED	18-12-2023T13:16:56.062
DEBMA5	AGVvIbgrUOI9XB_s4dyK4zyt5Y	0000000003539853	SAP	Hubspot EDIBroker	CANCELLED	12-12-2023T11:07:00.961
DEBMA5	AGVZ_OgM4557ooGZBub92mqvI	0000000003539855	SAP	Hubspot EDIBroker	CANCELLED	12-12-2023T11:07:00.948
DEBMA5	AGV298TYR3zZ5yC3_Fdcptw5v2j	0000000003539854	SAP	Hubspot EDIBroker	CANCELLED	12-12-2023T11:07:00.940
DEBMA5	AGVvWxG5d8c3nZgT72JubKa_2u	0000000003539848	SAP		CANCELLED	07-12-2023T10:51:05.492
DEBMA5	AGVvjuTqXCGqQqFhw_2rx1qgHv	0000000003539849	SAP	[EDIBroker, Hubspot]	CANCELLED	07-12-2023T10:51:05.418
DEBMA5	AGVvW2iOCbz1u8K2tEV3OYRMv	0000000003539852	SAP	EDIBroker Hubspot	CANCELLED	07-12-2023T10:51:05.341
DEBMA5	AGVvKqhecpdHNWZQw29GUUcy_m7	0000000003539851	SAP	Hubspot EDIBroker	CANCELLED	07-12-2023T10:51:05.175
DEBMA5	AGVvIOcc-EiLJvxdokm_c8_AJP	0000000003539844	SAP		CANCELLED	07-12-2023T10:50:51.513
DEBMA5	AGVvDeHulBCF3Cf7vuoelccTMa0O	0000000003539839	SAP		CANCELLED	07-12-2023T10:50:51.498

A12. Filtro según *Receiver*

Monitoring App

localhost:3000

SAP

Message Monitor

Filter by: Start Date: YYYY-MM-dd End Date: YYYY-MM-dd Search dates Update Messages Filter by Sender: Filter by Receiver: Hubspot Go to configurations

ProcessId	CorrelationId	IDOCNumber	Sender	Receiver	Status	LogEnd
DEBMA5	AGV4RyPY2mZL5wspQ24CMY6V	0000000003539856	SAP	EDIBroker Hubspot	RETRY	22-12-
DEBMA5	AGV5HfImQkbaXhVABTbfCIRvZ	0000000003539858	SAP	Hubspot EDIBroker	RETRY	22-12-
DEBMA5	AGWAH6B6V-DkXaORuMoz5XUWRJC	0000000003539859	SAP	EDIBroker Hubspot	RETRY	22-12-2023T09:41:37.271
DEBMA5	AGV4TX2XykgLxSpDpuZGhGBYP6	0000000003539857	SAP	EDIBroker Hubspot	RETRY	22-12-2023T09:33:38.782
DEBMA5	AGWAN8vbmC5ykyvYH14t_Yo7	0000000003539861	SAP	Hubspot EDIBroker	RETRY	22-12-2023T09:23:36.046
DEBMA5	AGWAKcZ4FvWk45XaXzCuhLJocl	0000000003539860	SAP	Hubspot EDIBroker	RETRY	22-12-2023T09:23:35.954
DEBMA5	AGWAN-D9Ld0GwA9PwPjFjowYIR	0000000003539863	SAP	EDIBroker Hubspot	RETRY	22-12-2023T09:23:35.950
	AGWAN5Yhc6YvdtbP_UQeZiFM4X	0000000003539862	SAP	Hubspot EDIBroker	COMPLETED	18-12-2023T13:16:56.062
DEBMA5	AGVvIbgrUOI9XB_s4dyK4zyt5Y	0000000003539853	SAP	Hubspot EDIBroker	CANCELLED	12-12-2023T11:07:00.961
DEBMA5	AGVZ_OgM4557ooGZBub92mqvI	0000000003539855	SAP	Hubspot EDIBroker	CANCELLED	12-12-2023T11:07:00.948
DEBMA5	AGV298TYR3zZ5yC3_Fdcptw5v2j	0000000003539854	SAP	Hubspot EDIBroker	CANCELLED	12-12-2023T11:07:00.940
DEBMA5	AGVvjuTqXCGqQqFhw_2rx1qgHv	0000000003539849	SAP	[EDIBroker, Hubspot]	CANCELLED	07-12-2023T10:51:05.418
DEBMA5	AGVvW2iOCbz1u8K2tEV3OYRMv	0000000003539852	SAP	EDIBroker Hubspot	CANCELLED	07-12-2023T10:51:05.341
DEBMA5	AGVvKqhecpdHNWZQw29GUUcy_m7	0000000003539851	SAP	Hubspot EDIBroker	CANCELLED	07-12-2023T10:51:05.175
	AGVvLH1J80PX08fM3A8Bdg_dhpV	0000000003539850	SAP	Hubspot EDIBroker	COMPLETED	07-12-2023T10:33:56.823