Desarrollo de iFlows para Gestión de Alertas en SAP Cloud Platform Integration: Un Enfoque Innovador en la Monitorización de Sistemas Empresariales

Daniel Lopez Sanchez, 1496866

Resumen—En entornos empresariales, la detección y gestión eficientes de anomalías y errores en procesos operativos representan un desafío significativo. Este trabajo presenta un enfoque innovador para la gestión de alertas dentro de dichos entornos, mediante el desarrollo de iFlows en SAP Cloud Platform Integration (CPI). Se implementa un sistema de alertas automatizado que facilita la monitorización en tiempo real y la notificación rápida de anomalías y errores en procesos empresariales. Utilizando la metodología Kanban y tecnologías como Node.js y React, se desarrolla una solución integrada que mejora significativamente la eficiencia operativa. El proyecto demuestra la viabilidad y eficacia de los iFlows en SAP CPI, ofreciendo una perspectiva novedosa en la gestión de alertas y su aplicación en diferentes sectores empresariales.

Palabras clave —SAP Cloud Platform Integration (CPI), iFlows, Gestión de Alertas, Monitorización de Sistemas, Integración Empresarial, Automatización de Procesos, Metodología Kanban, Node.js, React, Eficiencia Operativa.

Abstract —In business environments, efficiently detecting and managing anomalies and errors in operational processes poses a significant challenge. This work introduces an innovative approach to alert management within these environments, through the development of iFlows in SAP Cloud Platform Integration (CPI). An automated alert system is implemented, facilitating real-time monitoring and rapid notification of anomalies and errors in business processes. Utilizing Kanban methodology and technologies like Node.js and React, an integrated solution is developed, significantly enhancing operational efficiency. The project demonstrates the feasibility and effectiveness of iFlows in SAP CPI, offering a novel perspective on alert management and its application across various business sectors.

Index Terms—SAP Cloud Platform Integration (CPI), iFlows, Alert Management, System Monitoring, Business Integration, Process Automation, Kanban Methodology, Node.js, React, Operational Efficiency.

1 Introducción

En el actual panorama empresarial, la gestión eficaz de alertas es crucial para garantizar la continuidad y la eficiencia de los procesos en producción. Este Trabajo de Fin de Grado se centra en la creación y optimización de iFlows en SAP Cloud Platform Integration (CPI), con el objetivo de proporcionar un mecanismo eficiente para el envío de alertas a los usuarios finales y/o equipo de soporte, idealmente a través de correo electrónico o plataformas colaborativas como JIRA y Microsoft Teams. Dentro del marco de SAP, una empresa líder en software de gestión empresarial, CPI destaca como una herramienta clave para la integración eficiente y segura de sistemas y aplicaciones. Los iFlows en CPI, que permiten definir pasos para flujos de integración como la extracción y transformación de datos, son esenciales para conectar y comunicar sistemas, automatizando y optimizando procesos empresariales complejos. En muchas empresas, la detección de errores se realiza de manera tardía o manual, un desafío que este proyecto aborda desarrollando iFlows para la emisión de alertas automáticas, contribuyendo a una respuesta más rápida y eficiente en entornos empresariales.

Para comprender mejor este enfoque, es necesario familiarizarse con algunos conceptos clave. SAP, es una empresa multinacional alemana que se especializa en software de gestión empresarial. Su plataforma proporciona soluciones integrales para una amplia gama de procesos empresariales, como finanzas, recursos humanos, logística y más.

Dentro del ecosistema SAP, encontramos SAP Cloud Platform Integration (CPI), una herramienta esencial que facilita la integración de diferentes sistemas y aplicaciones dentro de una organización. Básicamente, CPI actúa como un intermediario que permite que distintas partes de un sistema empresarial se comuniquen y compartan datos de manera eficiente y segura.

En este contexto, los iFlows, o flujos de integración, emergen como elementos fundamentales dentro de la plataforma CPI. Una pieza clave en la arquitectura de nuestro sistema de alertas es el uso de 'Process Direct' en SAP Cloud Platform Integration (CPI), un adaptador que facilita la invocación directa de un iFlow desde otro dentro de la misma instancia de CPI. Estas herramientas

visuales permiten diseñar secuencias de operaciones, tales como la extracción y transformación de datos, así como mapeos, entre otras acciones y son complementadas con "groovy scripts" que es el lenguaje de programación utilizado en SAP CPI basado en java. Al facilitar la conexión y comunicación entre sistemas diversos, los iFlows posibilitan la automatización de procesos empresariales complejos y la mejora en la gestión del flujo de datos. Sin embargo, a pesar de estas capacidades avanzadas, se observa una carencia notable en el ámbito empresarial: la ausencia de sistemas de alerta implementados que permitan una detección temprana de errores. Tradicionalmente, la identificación de fallos se realiza solo después de su ocurrencia o mediante inspecciones manuales, lo cual introduce demoras significativas en la resolución de problemas, subrayando la importancia de incorporar mecanismos de alerta proactivos para una gestión más eficaz.

2 OBJETIVOS

El propósito de este Trabajo de Fin de Grado es abordar la problemática de la detección tardía de errores en entornos empresariales mediante la implementación de un sistema automatizado de alertas utilizando iFlows en SAP Cloud Platform Integration (CPI). Los objetivos específicos son:

- Emisión Automática de Alertas: Implementar un sistema de iFlows que permita la generación y envío inmediato de alertas a personal relevante en el momento exacto de la detección de un error, optimizando así el tiempo de respuesta.

- Personalización y Flexibilidad: Diseñar estos iFlows para que sean altamente personalizables y fácilmente integrables en diferentes entornos empresariales, especialmente con sistemas ya existentes.

- Facilitación de Actualizaciones y Escalamiento: Garantizar que los cambios necesarios en los iFlows existentes para incorporar este sistema de alertas sean mínimos y que el mantenimiento del sistema sea sencillo. Además, permite que la adición de más interfaces para aplicar este sistema sea un proceso fácil y sin complicaciones.

- Implementación de una Aplicación de Monitoreo: Complementar los iFlows con una herramienta de monitoreo que permita una gestión más eficiente y flexible de las alertas, proporcionando una visión integral de los errores y su estado en tiempo real. Destinada a empresas que utilizan SAP y buscan herramientas que faciliten su operatividad. Se busca diseñar una solución accesible incluso para aquellos que no poseen conocimientos avanzados de desarrollo, democratizando así el uso de la aplicación y extendiendo su utilidad a un espectro más amplio de usuarios dentro de la organización.

- Mejora de la Eficiencia Operativa: Buscar una notable mejora en la eficiencia operativa y una reducción del tiempo de inactividad en los procesos empresariales, gracias a una comunicación de alertas más ágil y precisa.

- Evaluación del Impacto: Medir el éxito del proyecto a través de indicadores clave, como la reducción en el tiempo de respuesta ante incidencias y el aumento de la satisfacción del usuario.

3 CONTEXTO Y ESTADO DEL ARTE

Tradicionalmente, en SAP CPI y en entornos empresariales similares, la detección de errores y la gestión de alertas se han basado en sistemas de monitoreo que requieren revisión manual o alertas básicas automatizadas. Estas alertas suelen ser genéricas y no proporcionan información detallada sobre el error, lo que lleva a una identificación y resolución más lenta de los problemas.
Los iFlows creados en SAP CPI introducen una mejora significativa en este proceso. Permiten no solo la detección automatizada de errores sino también la personalización avanzada de las alertas. Esto significa que las alertas pueden ser más específicas, incluyendo detalles cruciales sobre el error, lo que facilita una respuesta más rápida y eficaz. Además, la integración con plataformas modernas de comunicación como JIRA, Microsoft Teams, Slack y correo electrónico permite una distribución más eficiente de las alertas, asegurando que lleguen a las personas adecuadas de manera inmediata.

Además, la facilidad de integración y mantenimiento de estos iFlows hace que sean una solución atractiva para las empresas. Pueden ser implementados y adaptados con relativa facilidad en diferentes entornos empresariales, mejorando así la flexibilidad y la escalabilidad del sistema de alertas.

El correo electrónico (Email) es una herramienta de comunicación estándar en las empresas, utilizada para la distribución de alertas debido a su universalidad y su capacidad para transmitir información detallada.

Microsoft Teams es una plataforma de colaboración que integra chat, reuniones, llamadas y colaboración en documentos en un solo lugar, facilitando la comunicación en tiempo real dentro de las organizaciones.

JIRA, por otro lado, es un sistema de seguimiento de proyectos y problemas utilizado principalmente por equipos de desarrollo de software para registrar, organizar y priorizar errores y nuevas características de manera eficiente.
Slack es una plataforma de comunicación empresarial que ofrece un espacio de trabajo donde los equipos pueden colaborar. Se organiza en canales donde los miembros pueden enviar mensajes, compartir archivos y realizar llamadas. Slack integra muchas herramientas y servicios, permitiendo a los usuarios centralizar sus tareas y comunicaciones en una sola aplicación. Es ampliamente utilizado por organizaciones para la comunicación

interna, gestión de proyectos, y como un medio para integrar diversos servicios de automatización y notificaciones.

4 Metodologia

En la realización de este Trabajo de Fin de Grado, se implementó la metodología Kanban, una estrategia orientada a optimizar la gestión del proyecto y alinearla con los objetivos establecidos. La capacidad de Kanban para mejorar la eficiencia a través de la visualización del flujo de trabajo ha sido fundamental para organizar y priorizar las tareas. Se adoptó un tablero de Kanban para gestionar visualmente el proyecto, estructurado en columnas que representan los diferentes estados de las tareas: 'TO DO', 'DOING', y 'DONE'. Este sistema no solo refleja el progreso del proyecto, ofreciendo una clara distinción entre tareas pendientes, en curso y finalizadas, sino que también facilita una gestión ágil y promueve la colaboración con el equipo de la empresa, permitiendo un seguimiento claro y efectivo del avance del trabajo. A continuación se puede ver en la Figura 1 una representación visual de cómo es el tablero.

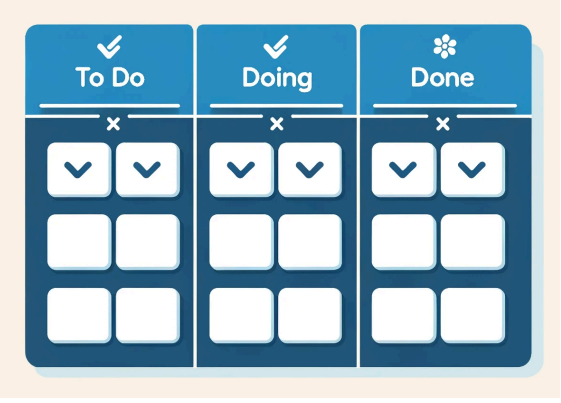


Figura 1: Ejemplo ilustrativo de un tablero de Kanban utilizado en la gestión del proyecto

El tablero de Kanban presentado sirve como ejemplo de la metodología implementada para la organización y priorización de tareas, destacando la importancia de la colaboración y la comunicación con la empresa en el desarrollo exitoso del proyecto. A continuación se explican unos puntos importantes, las fases que ha tenido el proyecto utilizando esta metodología :

- Enfoque en la Codificación: Dado que el corazón del proyecto reside en el desarrollo de iFlows para SAP CPI, se ha puesto especial énfasis en el ciclo de codificación. Este enfoque ha permitido concentrar los esfuerzos en la creación y refinamiento de los iFlows, asegurando que la solución desarrollada sea robusta y alinee con las necesidades identificadas.

- Planificación y Evaluación: La metodología Kanban se ha complementado con un diagrama de Gantt, que ha servido para establecer una línea de tiempo detallada y planificar las fases del proyecto. La planificación ha incluido la asignación de tiempos para el desarrollo, pruebas y evaluación, asegurando que las actividades se alineen con los objetivos y los tiempos del TFG.

- Alineación con Objetivos: La metodología elegida ha demostrado ser adecuada para los objetivos planteados, permitiendo una adaptación fluida ante imprevistos y proporcionando un marco de trabajo que ha facilitado el logro de los resultados esperados.

5 Arquitectura utilizada para el sistema de alertas

La estructura del sistema de alertas diseñado en este Trabajo de Fin de Grado se articula en cuatro niveles estratégicamente definidos, cada uno con funciones específicas y diseñados para asegurar una integración óptima con los procesos de negocio de los clientes.

- Nivel 1: Integración y Captura de Excepciones

En la base de la arquitectura se encuentran los procesos de negocio de los clientes. La modificación de estos iFlows del cliente es mínima, solo se añade un proceso de excepción. Su función principal es que si un mensaje resulta erróneo, este proceso de excepción se activa y registra el mensaje fallido en un DataStore, etiquetandolo con un identificador (Message Guid) único. Este registro es crucial ya que actúa como un enlace directo a la información del mensaje para su procesamiento posterior.

- Nivel 2: iFlow de Gestión de Alertas 'Manage Alerts'

Este nivel vital opera un iFlow denominado "Manage Alerts". Su rol es monitorear el DataStore y procesar los mensajes fallidos. Cada 10 segundos lee el contenido del DataStore donde se guardan los identificadores únicos de los mensajes erróneos; este iFlow consulta la API del CPI para obtener los detalles del mensaje que ha fallado, utilizando el Message Guid como referencia. Aquí, un Groovy script es fundamental para extraer la información del JSON de respuesta obtenida de la api, esta información es necesaria para la notificación. Durante este proceso, se implementó un enrutador que asegura la integridad de los datos: si los datos no están completos, el mensaje se reingresa en el DataStore; si están completos, se procede al Value Mapping.

- Nivel 3: Configuración de Notificaciones con Value Mapping

El sistema configurable de Value Mapping opera asignando propiedades específicas de los mensajes a rutas de proceso determinadas, lo que permite seleccionar de manera automática la herramienta de ticketing adecuada. Por ejemplo, basándose en ciertos criterios como la urgencia, el tipo de error, o el departamento afectado, el Value Mapping asigna el mensaje a la herramienta más conveniente—Email, Slack, Teams o JIRA—para su gestión. Esta metodología asegura que la respuesta al

incidente se canalice a través del medio más eficaz, optimizando así el tiempo de resolución y la asignación de recursos para el mantenimiento y expansión de los flujos de trabajo empresariales.

- Nivel 4: iFlows de Notificación Específicos

El cuarto y último nivel se compone de iFlows individuales responsables de construir y enviar las notificaciones finales a las herramientas seleccionadas. Cada iFlow está especializado en la formación del mensaje para una plataforma de notificación específica, garantizando que la alerta se comunique en el formato más adecuado y eficiente para cada canal.

Este enfoque por niveles no solo asegura una gestión de alertas precisa y eficiente sino que también proporciona una solución que es escalable y fácil de mantener. La arquitectura está diseñada para introducir una mínima perturbación en los iFlows de los clientes y para facilitar la incorporación de mejoras y nuevas funcionalidades en el futuro. A continuación en la figura 2 se mostrará un esquema de la arquitectura a seguir.

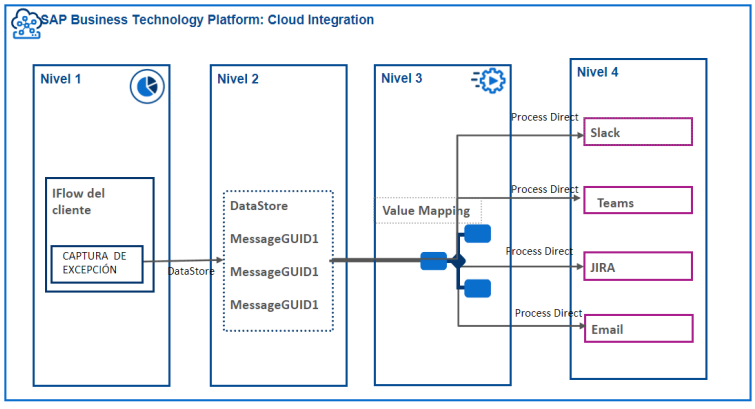


Figura 2: Esquema de cómo están relacionados los diferentes niveles

6 Implementación de iFlows para la Gestión de Alertas

La sección sobre la implementación de iFlows para la gestión de alertas es fundamental en el desarrollo de sistemas de notificación eficientes dentro de este Trabajo de Fin de Grado. El enfoque se centra en la adaptabilidad y personalización de las notificaciones a través de Email, JIRA, Slack, y Teams, destacando la flexibilidad del sistema para satisfacer diversas necesidades de comunicación empresarial. Este capítulo aborda detalladamente la configuración y el proceso detrás de cada método de notificación, ilustrando cómo se pueden optimizar los flujos de trabajo para una gestión de alertas efectiva y coherente con los estándares de la organización.

6.1 Email

Su función es recibir y procesar la información de alertas para generar notificaciones dirigidas a los interesados y enviarlas por correo electrónico. A continuación, se detallan los componentes y funcionalidades del iFlow desarrollado del envío a través de correo electrónico:

1. Origen de Datos: Este iFlow recibe datos directamente del iFlow "Manage Alerts" (explicado en el nivel 2 de la arquitectura) a través de un 'Process Direct' cuya ruta es determinada por un Value Mapping. Esto asegura que la información transferida es relevante y concisa, facilitando la precisión en la notificación final.

2. Configuración de Parámetros de Log: Aunque es opcional, este paso permite establecer parámetros para los registros (logs) del mensaje, proporcionando información detallada sobre el funcionamiento del iFlow en caso de errores. Estos registros son esenciales para la depuración y el mantenimiento continuo del sistema.

3. Almacenamiento de Datos: Se realiza una captura de los logs establecidos y del cuerpo del mensaje que se guarda como un archivo de texto adjunto en el monitor . Esto no solo sirve para el seguimiento, sino que también conserva la información necesaria para la composición de la notificación.

4. Groovy Script para Preparación de Notificación: Se ejecuta un script Groovy que establece las variables necesarias para la notificación, como la introducción del correo electrónico. Este script es clave para personalizar el contenido de la notificación de acuerdo con las necesidades específicas de la alerta. Siempre que llegue a este punto vendrá toda la información necesaria para poder componer la notificación de error, porque el “Manage Alerts” se encargará de ello.

5. Composición de la Notificación: El paso final es la composición del correo electrónico, que se realiza en formato HTML. Aquí se integran todas las variables definidas previamente junto con la información proveniente del iFlow "Manage Alerts". El diseño del correo electrónico se ajusta para reflejar el aspecto deseado, asegurando que la notificación sea no solo informativa sino también visualmente coherente con la imagen corporativa.

Cada uno de estos pasos está diseñado para asegurar que la notificación generada sea clara, informativa y entregada de manera efectiva al destinatario final, cumpliendo con los requisitos de comunicación de la organización.

Para una representación visual detallada del proceso de flujo de trabajo del iFlow de notificación, ver la siguiente figura 3 del apéndice A.

6.2 JIRA

El iFlow de integración con JIRA se ha desarrollado con el

propósito específico de gestionar la creación y el envío de incidentes dentro de la plataforma JIRA. A continuación, se desglosan los elementos clave y la funcionalidad de este iFlow que envía las alertas a través de JIRA:

1. Origen de Datos: Este iFlow inicia su proceso al recibir datos del 'Process Direct', procedentes de un iFlow previo, que siempre es el iFlow "Manage Alerts". Este diseño garantiza una transmisión coherente y precisa de la información hacia el proceso de JIRA.

2. Script Groovy para la Creación del Contenido del Incidente: El núcleo del iFlow es un script Groovy encargado de construir el mensaje de alerta en el formato JSON, que es el requerido por la API de JIRA. El script se ocupa de definir la estructura de los campos obligatorios y opcionales del incidente, adaptando los datos recibidos a las necesidades de JIRA.

3. Conexión HTTP para el Envío a la API de JIRA: El paso subsiguiente implica el uso de una conexión HTTP para transmitir el mensaje de alerta a la API de JIRA. Este es un paso crucial donde se concreta el envío del incidente a JIRA para su creación y manejo subsecuente.

Tras la transmisión exitosa del mensaje a la API de JIRA, el iFlow concluye su operación. Aunque el proceso de envío es automatizado, se confía en que la API de JIRA administrará el incidente de acuerdo con los protocolos y flujos de trabajo configurados en la plataforma.

Este iFlow es fundamental para garantizar que las alertas identificadas sean debidamente registradas en JIRA, promoviendo un manejo eficaz y la asignación oportuna de acciones para su resolución.

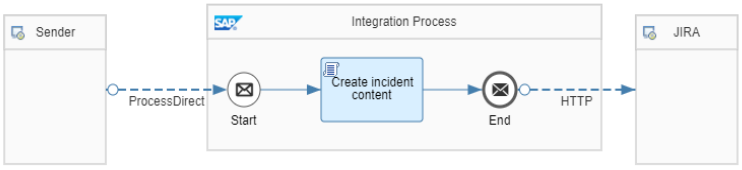


Figura 4: Flujo de trabajo del iFlow de Integración con JIRA

La imagen ilustra el flujo de trabajo descrito, desde la recepción de datos hasta el envío del incidente.

6.3 Microsoft Teams

Este iFlow representa una secuencia de procesos diseñados para enviar notificaciones automatizadas a Microsoft Teams, utilizando la API de Microsoft Graph como puente de comunicación. Microsoft Graph proporciona acceso a los datos almacenados en servicios de Microsoft 365. Las aplicaciones personalizadas pueden usar la API de Microsoft Graph para conectarse a los datos y usarla en aplicaciones personalizadas para mejorar la productividad de la organización.

A continuación, se detalla cada etapa del flujo:

1. Origen de Datos: Se repite el mismo caso que en la implementación del Email.

2. Parámetros del token: Se definen las cabeceras necesarias para conseguir el token para acceder a la api de Microsoft graph.

3. “GET token”: Se hace una llamada http para conseguir el token para poder utilizar la API de Microsoft Graph, y desde ahí poder enviar el

Name	Source Type	Source Value
grant_type	Constant	password
client_id	Constant	52724e0e-3d1f-488b-9d16-ef5d21856a
client_secret	Constant	TV38Q~XSOHwcn9BpMxEoHPD.AW2(
scope	Constant	https://graph.microsoft.com/ChatMessag...
username	Constant	felipe.martin@code10it.com
password	Constant	Hap70140
Content-type	Constant	application/x-www-form-urlencoded

mensaje a un chat de Teams.

4. Configuración del token: Un groovy script que crea una cabecera llamado “Authorization” donde le asigna el token conseguido anteriormente.

5. Composición del Mensaje : Aquí, el contenido de la notificación se estructura en formato JSON, configurado el mensaje para que se vea como una 'Card' que es un formato más atractivo texto en las conversaciones.

6. Envío mediante Microsoft Graph (Request Reply): El mensaje es enviado utilizando la API de Microsoft Graph, que es el medio disponible para comunicar con chats privados en Teams, a diferencia de otros métodos limitados a canales de equipo.

7. Finalización y Recepción: Tras el envío exitoso, el iFlow se completa y la notificación es recibida por el destinatario en Teams.

En la siguiente figura 5 se puede ver el iFlow desarrollado para enviar alertas a Teams.

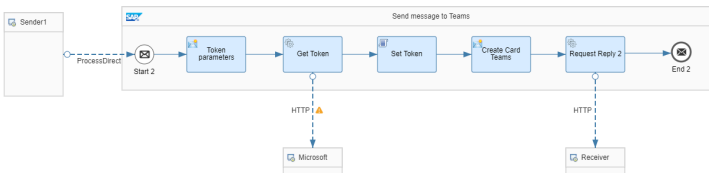


Figura 5: Flujo de trabajo del iFlow de Integración con Microsoft teams.

En la figura 6 siguiente se puede ver un ejemplo de una ‘Card’ con una notificación de Teams.

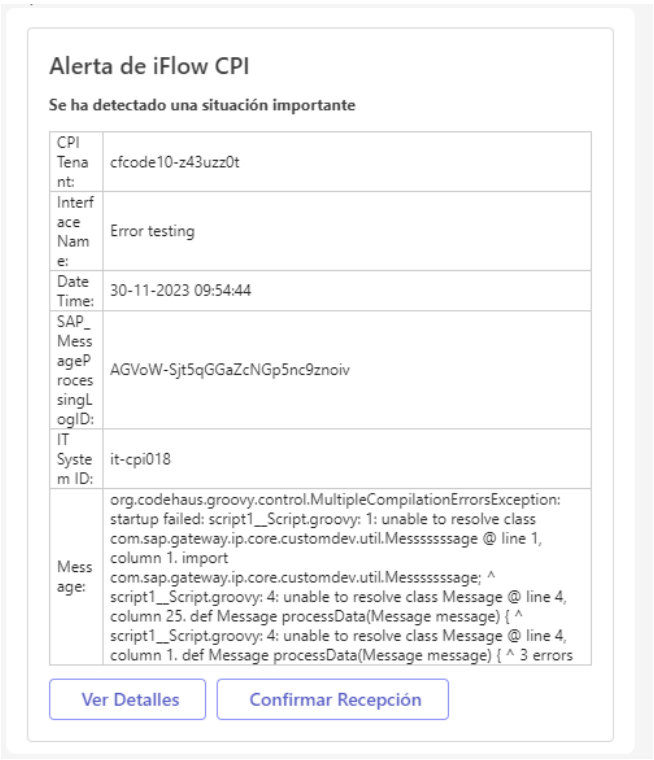
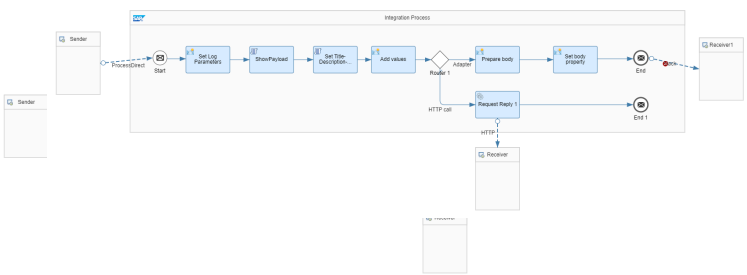


Figura 6: Ejemplo de alerta de notificación que te llega en el chat de Teams.

6.4 Slack

Este iFlow es el responsable de orquestar el envío de notificaciones automatizadas hacia Slack mediante SAP Cloud Platform Integration (CPI). Inicia con la recepción de mensajes mediante 'Process Direct', seguido de la adición y almacenamiento de registros de logs. Una característica distintiva de este iFlow es su capacidad de enrutamiento, lo que permite dos métodos de envío de notificaciones: directamente a través de una conexión HTTP a la API de Slack y mediante un adaptador pre integrado en SAP. Independientemente del método, el objetivo es garantizar la entrega exitosa de la alerta al destino final.

Para una comprensión más profunda del proceso y la estructura del iFlow de notificación para Slack, se ha incluido una representación visual detallada en la siguiente figura.



7 GESTIÓN DE CREDENCIALES EN SAP CLOUD PLATFORM INTEGRATION (CPI)

En este proyecto, se ha hecho un uso extensivo de las funcionalidades de seguridad de SAP CPI para la gestión eficiente de las credenciales necesarias en las conexiones a diversas APIs. Este apartado describe cómo se han administrado y protegido estas credenciales.

7.1 Uso del 'Manage Security' en CPI

Para asegurar una autenticación segura y eficiente en los iFlows, SAP CPI centraliza y gestiona las credenciales de conexión a través de su herramienta "Manage Security". Aquí, cada usuario y contraseña se asocian con un alias único, creando un repositorio de credenciales seguro y fácilmente referenciable. Es importante destacar que estos alias son exclusivos para su uso dentro de los iFlows que operan bajo nuestra instancia de SAP CPI, lo que garantiza que la autenticación se maneje de manera segura y que los datos sensibles permanezcan protegidos y aislados dentro de nuestro entorno controlado.

8 DESARROLLO DE UN FRONT-END PARA LA VISUALIZACIÓN DE DATOS ÚTILES PARA EL MONITOR UTILIZANDO NODEJS Y REACT

SAP Build Apps es una herramienta dentro de la suite de productos de SAP que permite a los usuarios desarrollar aplicaciones de negocio. Es conocida por ofrecer una experiencia de desarrollo simplificada, permitiendo a los usuarios con poca o ninguna experiencia en codificación construir aplicaciones funcionales a través de un enfoque de configuración en lugar de codificación. La decisión de migrar de SAP Build Apps a una solución personalizada utilizando Node.js y React se tomó en respuesta a las limitaciones de SAP Build Apps para el desarrollo de aplicaciones complejas y las dificultades en su comercialización. La necesidad de una aplicación de monitoreo más avanzada y personalizable llevó a esta transición estratégica, buscando mayor libertad en la creación de funcionalidades y una mejora en el aspecto visual.

Se han alcanzado varios objetivos en el desarrollo de nuestra aplicación. A continuación, se presentan los avances.

8.1 Avances en la Aplicación:

- Componentes Desarrollados: Hasta la fecha, se han completado tres componentes :
  - Artifacts : Muestra los iFlows que están desplegados con información adicional de cada una.
  - Mensajes : Muestra los mensajes que se han procesado
  - Material de Seguridad : Se puede ver los alias de las diferentes credenciales

Para ver un ejemplo visual de los componentes ver figura 7 del apéndice A.

- Comparación con el Monitor Integrado de CPI: Se ha conseguido una presentación de información más eficiente y accesible. Por ejemplo, en la aplicación desarrollada para este TFG, los iFlows se muestran en una fila en lugar de en dos columnas como en CPI, lo que mejora la legibilidad y la navegación.

Para ver un ejemplo visual de la comparación ver figura 8 del apéndice A.

8.3 Reflexiones sobre el desarrollo del front end

Se ha contemplado la incorporación de "Fiori Elements", son plantillas que dan el aspecto que utiliza sap en sus aplicaciones, para enriquecer la interfaz de usuario con un diseño que armonice con las aplicaciones SAP. Esta propuesta surge con la finalidad de ofrecer al usuario una experiencia visual coherente y familiar. Así mismo, se aspira a brindar una gama de opciones estéticas para la personalización de la interfaz, incluyendo la posibilidad de modificar la apariencia mediante el relleno de campos predefinidos. Esta flexibilidad permitiría que la aplicación se ajuste a las preferencias visuales de los usuarios finales.

Una característica distintiva y un considerable valor añadido de la aplicación desarrollada en este Trabajo de Fin de Grado es su portabilidad. Actualmente, la monitorización de procesos se encuentra confinada al cockpit de SAP, con acceso restringido exclusivamente a través de su portal web. Esta limitación restringe la flexibilidad y la accesibilidad para los usuarios que requieren supervisar los procesos desde diferentes entornos o dispositivos.

Estas implementaciones buscan superar las barreras de acceso y uso, ofreciendo una herramienta integral que responda a las necesidades de los usuarios en un contexto de trabajo dinámico y móvil. Al disponer de la aplicación tanto en versión web como descargable, se maximiza la disponibilidad y se mejora la experiencia del usuario, permitiendo una gestión más eficiente y proactiva de los procesos empresariales.

8.4 Trabajo futuro

Es el inicio de un proyecto más grande y se tiene pensado realizar más trabajo en un futuro.

Para la sección de visualización y gestión de credenciales de la aplicación, se está desarrollando una interfaz de usuario intuitiva que incluirá cajas expandibles, permitiendo a los usuarios ver más o menos detalles según lo necesiten. También se está considerando la implementación de una funcionalidad de arrastrar y soltar (drag and drop) que facilitará la organización personalizada de las credenciales por parte de los usuarios. Además, se están diseñando características que permitirán la creación y carga de nuevas credenciales de forma sencilla y directa, mejorando la gestión de las mismas dentro de la aplicación.

Se contempla la implementación de mejoras como el cambio de la columna 'ID' por 'Package' y la optimización visual. Se está trabajando en implementar funcionalidades adicionales, como enlaces directos a mensajes en CPI y la mejora de la visualización de errores.

También, se contempla la posibilidad de desarrollar una aplicación descargable, lo que facilitaría el acceso sin necesidad de una conexión directa al cockpit de SAP, proporcionando así una opción más flexible y adaptativa para el monitoreo en tiempo real.

Los próximos pasos incluyen la solución de desventajas identificadas, como la mejora visual y la implementación de funcionalidades pendientes, para asegurar una herramienta de monitoreo integral y eficaz.

8.4 Arquitectura utilizada para la aplicación

Esta tarea se llevó a cabo a través de la API de SAP Cloud Platform Integration, donde se utilizaron métodos específicos para acceder y recuperar datos cruciales. Este proceso fue esencial para alimentar la aplicación con información actualizada y relevante. Además, la aplicación se desarrolló utilizando el modelo vista-controlador (MVC), lo cual proporcionó una estructura organizada y eficiente. El modelo MVC permitió separar la interfaz de usuario (vista), la lógica de negocio (controlador) y los datos de la aplicación (modelo), facilitando así un desarrollo más ágil y mantenible.

Además, se ha propuesto que la aplicación se adapte al modelo de operaciones de la empresa CODE 10, la cual se estructura en una arquitectura de cuatro capas:

- Capa de Recepción (iFlows Genéricos): En esta capa inicial, iFlows genéricos capturan los mensajes entrantes. Basándose en el remitente y el destinatario, los mensajes se clasifican y se encolan en un sistema JMS para su procesamiento posterior.

- Capa de Enrutamiento: La lógica de enrutamiento reside en esta capa, determinando el flujo de mensajes a su destino en la tercera capa, guiada por parámetros como el "Sender", "Receiver", e "InterfaceName".

- Capa de Aplicación: En esta etapa, se aplican las transformaciones necesarias, como mapeos y codificaciones, adaptando el mensaje para su entrega final.

- Capa de Envío (iFlows Genéricos): La capa final es responsable de distribuir el mensaje procesado al destinatario adecuado.

La mejora propuesta permitiría a los usuarios filtrar y

visualizar el recorrido de los mensajes a través de las distintas capas, identificando por qué iFlows específicos ha transitado y visualizando mensajes relacionados.

En el apéndice A figura 9 podemos ver un esquema de la arquitectura utilizada en la empresa.

8 CONCLUSIONES

Al culminar este Trabajo de Fin de Grado, hemos logrado implementar con éxito un avanzado sistema de alertas mediante iFlows en SAP CPI, marcando una mejora significativa en la gestión de errores en entornos empresariales. La incorporación de una aplicación interactiva, desarrollada con Node.js y React, ha potenciado la funcionalidad del sistema, integrándose de manera efectiva con plataformas colaborativas como Teams, Email, Slack y JIRA. Este enfoque ha demostrado ser esencial para la adaptabilidad y eficiencia del sistema de alertas, probando su aplicabilidad en una variedad de contextos comunicativos empresariales.

La transición estratégica hacia la utilización de tecnologías emergentes y la decisión de migrar de SAP Build Apps a soluciones personalizadas basadas en JavaScript no solo han enriquecido el proyecto, sino que también han ampliado sus capacidades. Este cambio ha implicado un profundo proceso de aprendizaje y adaptación, subrayando la importancia de la innovación continua y la autodidacta en el desarrollo tecnológico.

Los objetivos específicos planteados al inicio de este proyecto se han cumplido satisfactoriamente. La implementación del sistema de iFlows ha permitido la emisión automática de alertas personalizadas y su distribución inmediata al personal relevante, optimizando el tiempo de respuesta ante errores. La personalización y flexibilidad de estos iFlows se han demostrado al integrarse sin problemas en diferentes entornos empresariales, resaltando la versatilidad del sistema. Además, la facilitación de actualizaciones y el escalado del sistema de alertas han sido implementados con éxito, asegurando que el mantenimiento y la expansión del sistema sean procesos sencillos y eficientes.

El complemento de una aplicación de monitoreo ha mejorado la visibilidad, ofreciendo una interfaz accesible y fácil de usar evitando las limitaciones del cockpit de SAP, promoviendo una monitorización más flexible y desde diversos dispositivos. Esta accesibilidad ampliada refleja el compromiso del proyecto con la mejora de la eficiencia operativa y la reducción del tiempo de inactividad en procesos empresariales, a la vez que se evalúa su impacto a través de indicadores clave como la reducción del tiempo de respuesta y el aumento de la satisfacción del usuario.

En resumen, este proyecto ha cumplido y superado los objetivos propuestos, estableciendo un marco sólido para futuras innovaciones. La experiencia adquirida, los desafíos superados y las soluciones implementadas representan un valioso aprendizaje profesional, destacando la importancia de la perseverancia, la paciencia y la adaptabilidad en el ámbito del desarrollo de software y la gestión de proyectos tecnológicos."

AGRADECIMIENTOS

Mi más profundo agradecimiento va dirigido a mi tutora del TFG, cuya orientación y apoyo constante han sido esenciales en cada paso de este proyecto. Su asesoramiento ha sido invaluable, y estoy sinceramente agradecido por haber tenido la oportunidad de trabajar bajo su dirección.

También deseo expresar mi gratitud a CODE 10, en especial a Antonio Valladares, Sergio Martínez y Luis Gómez, por su inestimable ayuda, ideas innovadoras y consejos prácticos en el desarrollo del sistema de alertas. Su apoyo y guía han sido fundamentales para el éxito de este trabajo.

Por último, pero no por ello menos importante, agradezco infinitamente a mi madre por su paciencia, comprensión y amor incondicional durante todo este proceso. Su apoyo ha sido un pilar fundamental en mi vida y en este proyecto.

BIBLIOGRAFIA

[1] Valladares, A. (2023, septiembre). Comunicación personal. [Problemas en el mantenimiento de sistemas].
[2] SAP Tutorial Navigator | Tutorials for SAP Developers. (s. f.). SAP. https://developers.sap.com/tutorial-navigator.html
[3] Canva. (s. f.). https://www.canva.com/es_es/
[4] Biblioguías: Citas y elaboración de bibliografía: El plagio y el uso ético de la información: estilo IEEE. (s. f.). https://biblioguias.uam.es/citar/estilo_ieee
[5] Create an application with SAP Build Apps | SAP Tutorials. (s. f.). SAP. https://developers.sap.com/mission.appgyver-low-code.html
[6] Create a sales order app using SAP Build Apps | SAP Tutorials. (s. f.). SAP. https://developers.sap.com/group.sap-build-apps-process-trigger.html
[7] SAP Business Accelerator Hub. (s. f.). https://api.sap.com/api/MessageProcessingLogs/resource/Error_Information
[8] REST APIs. (s. f.). https://developer.atlassian.com/server/jira/platform/rest-apis/
[9] Tutorial: Intro to React – react. (s. f.). React. https://legacy.reactjs.org/tutorial/tutorial.html
[10] Qué es MVC. (s. f.). DesarrolloWeb.com. https://desarrolloweb.com/articulos/que-es-mvc.html
[11] Build Node.js apps with Visual Studio code. (2021, 3 noviembre). https://code.visualstudio.com/docs/nodejs/nodejs-tutorial
[12] React JavaScript tutorial in Visual Studio code. (2021, 3 noviembre). https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
[13] Duggal, A. (2022, 16 junio). How to make a filter component in React. freeCodeCamp.org. https://www.freecodecamp.org/news/how-to-make-a-filter-component-in-react/
[14] Yüksel, M. (2023, 26 junio). Create responsive sidebars with React-Pro-Sidebar and MUI - LogRocket blog. LogRocket Blog.

https://blog.logrocket.com/creating-responsive-sidebar-react-pro-sidebar-mui/
[15] Pasando funciones a componentes – react. (s. f.). React. https://es.legacy.reactjs.org/docs/faq-functions.html
[16] Sharma, P., & Sharma, P. (2022, 13 octubre). How to call Web API with UseEffect Hook in React TypeScript? Cynoteck. https://cynoteck.com/es/blog-post/how-to-call-web-api-with-useeffect-hook-in-react-typescript/
[17] SAP Business Accelerator Hub. (s. f.). https://api.sap.com/package/CloudIntegrationAPI/odata

APENDICE A

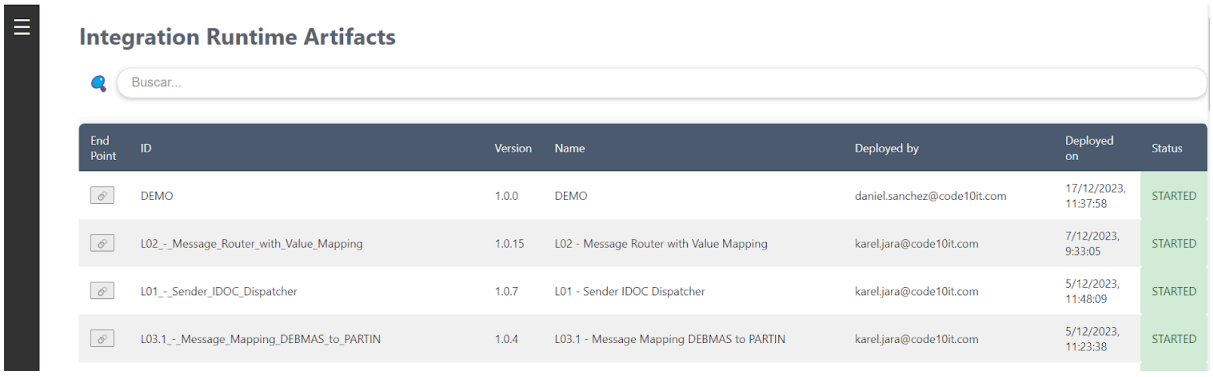


Figura 7.1: Componente de Artifacts, donde se verán los iFlows con información relevante de cada uno.

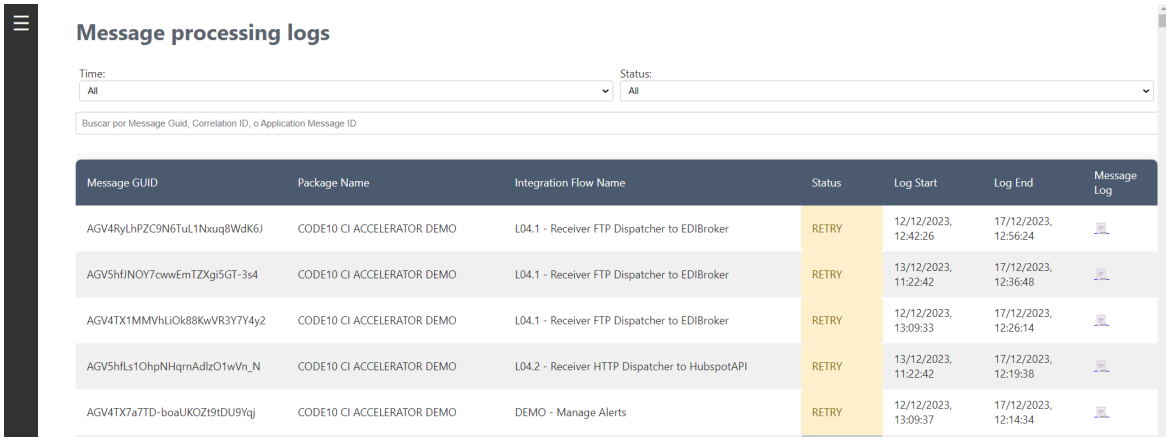


Figura 7.2: Componente de Mensajes

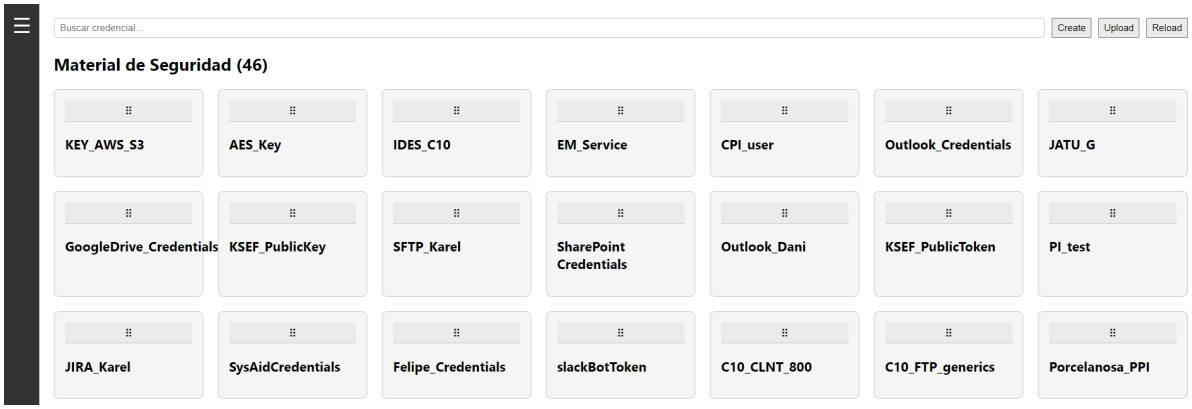
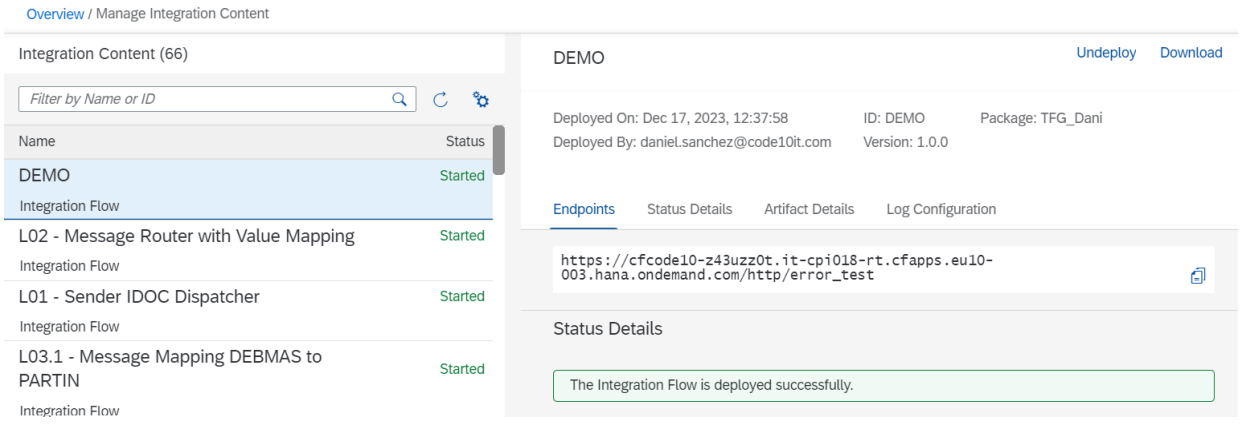
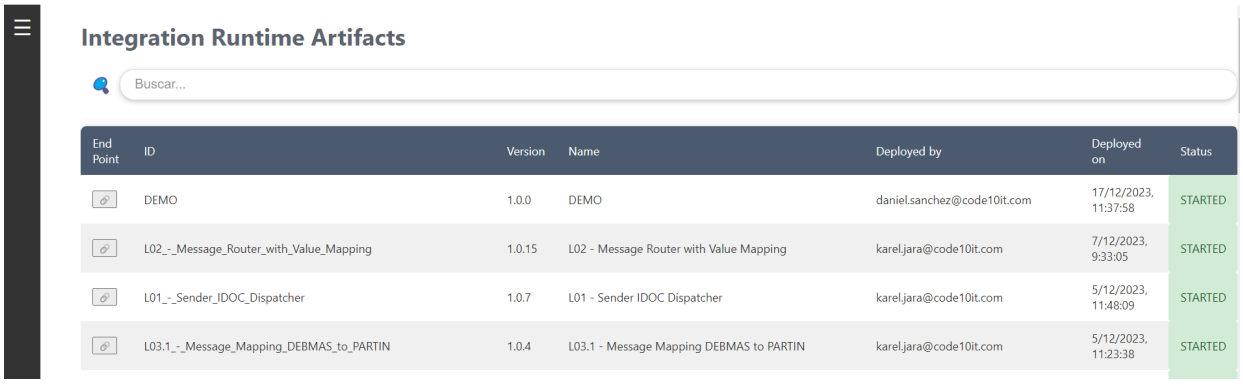


Figura 7.3: Componente de Material de Seguridad



Monitor CPI Integrado

VS



Monitor aplicación desarrollada

Figura 8: Comparación entre la aplicación y el monitor integrado de CPI

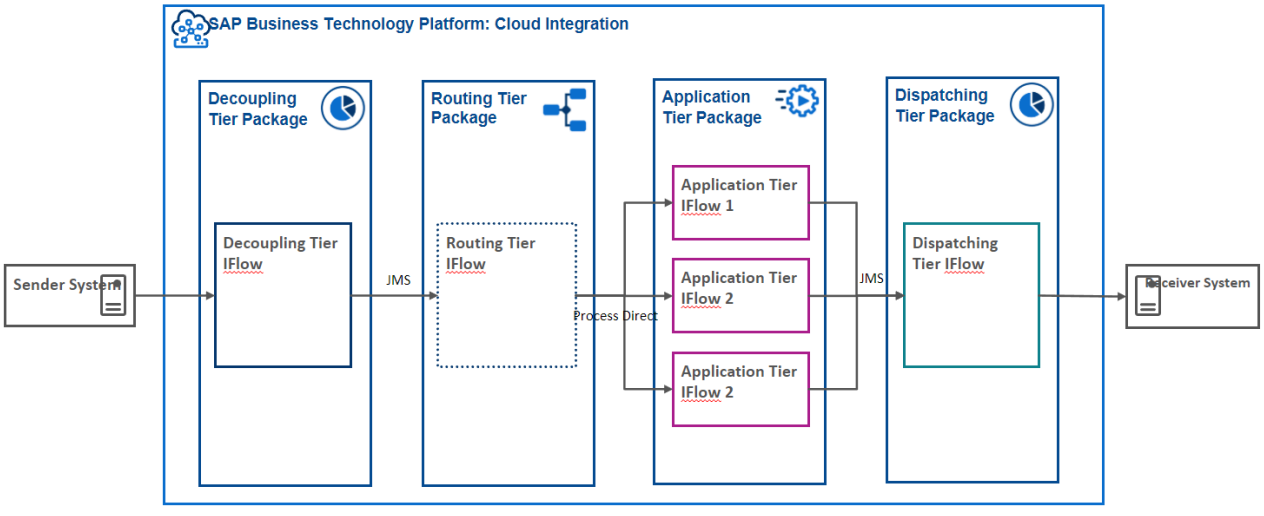


Figura 9: Arquitectura utilizada en CODE 10 de las 4 capas.

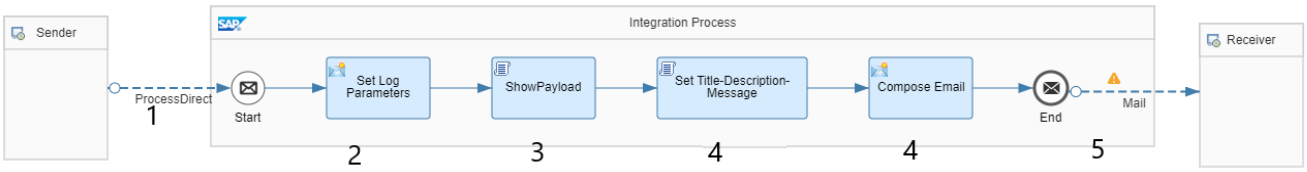


Figura 3: iFlow que envía alerta a través de un Email.