

Diseño Técnico de Integración entre Chatbot y Máquina SAP S/4 con SAP CPI

Diseño Técnico de Integración entre Chatbot y Máquina SAP S/4 con SAP CPI	1
1. Descripción General del Proyecto	1
2. Arquitectura de la Solución	2
2.1 Componentes Principales	2
2.2 Diagrama de Arquitectura	3
3. Flujo de Integración	3
3.1 Descripción del Flujo	3
4. Tecnologías y Protocolos	10
5. Seguridad	11
5.1 Autenticación	11
5.2 Cifrado de Datos	11
6. Manejo de Errores y Excepciones	11
6.1 Tipos de Errores Esperados	11
7. Pruebas	12
7.1 Estrategia de Pruebas	12
7.2 Pruebas de Seguridad	14

1. Descripción General del Proyecto

Objetivo:

El objetivo de esta integración es conectar un chatbot basado en inteligencia artificial con una máquina SAP a través de SAP CPI, permitiendo que los usuarios realicen consultas sobre información almacenada en el sistema SAP, como el estado de facturas o pedidos de manera automatizada y en tiempo real. Esta solución busca mejorar la experiencia del cliente, reducir tiempos de respuesta y optimizar la gestión de consultas

mediante la automatización del proceso de comunicación entre el chatbot y la máquina SAP.

Alcance:

Esta integración cubre el intercambio de información entre el chatbot y la máquina SAP mediante SAP CPI. El chatbot permitirá a los usuarios realizar consultas específicas relacionadas con:

- Ver un listado completo de pedidos o la información detallada de cada pedido.
- Descarga automática de facturas del pedido.
- Atención a cualquier consulta realizada por el cliente.

El proyecto incluye:

- Implementación y desarrollo de un chatbot para facilitar la comunicación.
- Configuración y desarrollo de flujos de integración en SAP CPI.
- Comunicación entre el chatbot (basado en Llama 3.2) y la máquina SAP (SAP S/4HANA).
- Transformación de datos entre los formatos JSON y XML.

Exclusiones:

- No se contempla la modificación del sistema SAP para agregar nuevas funcionalidades, solo la lectura de información existente.
- No se incluyen integraciones con otros sistemas externos más allá de la máquina SAP y el chatbot.

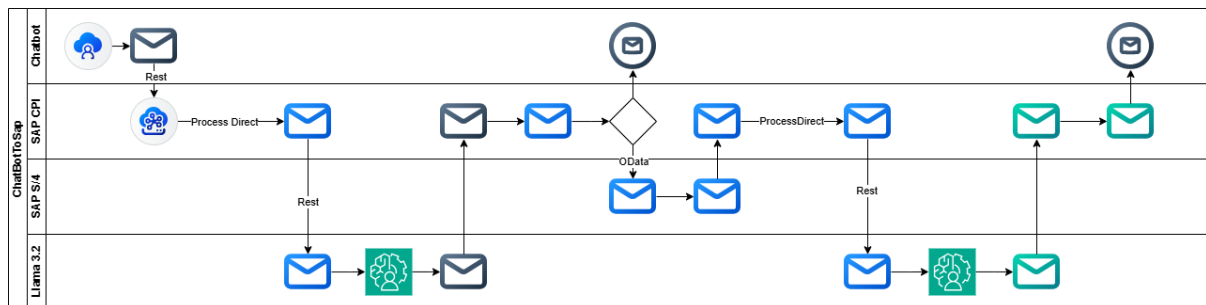
2. Arquitectura de la Solución

2.1 Componentes Principales

- **Chatbot:** Aplicación web que consta de un backend desarrollado en Node.js (JavaScript) y un frontend implementado en HTML con JavaScript. El backend gestiona la lógica y las interacciones con SAP CPI, mientras que el frontend ofrece una interfaz interactiva para que los usuarios realicen consultas.
- **Inteligencia artificial:** Llama 3.2.
- **SAP CPI (Cloud Platform Integration):** Middleware para la gestión de mensajes.

- **Máquina SAP:** SAP S/4.
- **Conectores:** HTTP, OData v2.

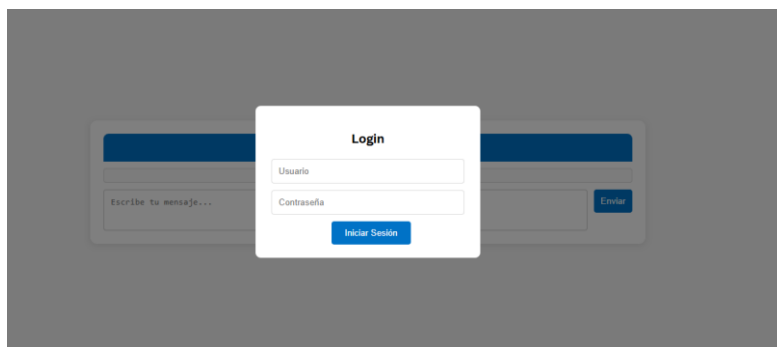
2.2 Diagrama de Arquitectura



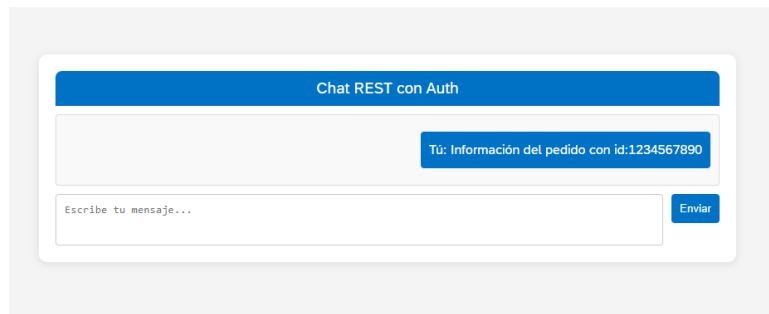
3. Flujo de Integración

3.1 Descripción del Flujo

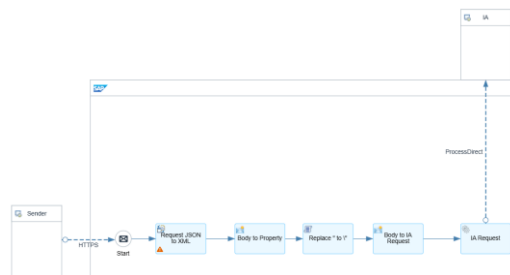
1. Se introducen las credenciales del integration suite (client_ID y client_secret) en el chatbot.



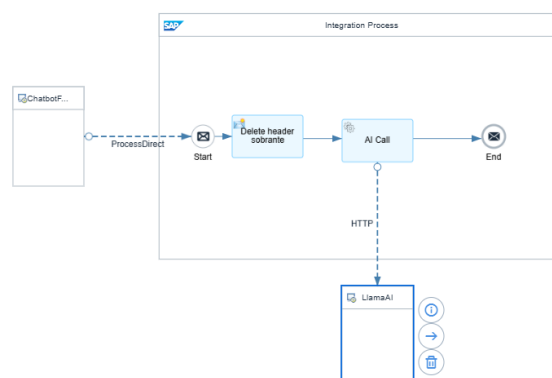
2. Enviamos un nuevo mensaje por el chatbot.



3. El chatbot envía la solicitud a SAP CPI a través de un canal HTTPS al flujo que se llama "ChatbotTOSAP".
4. SAP CPI transforma el mensaje:
 - a. formato JSON a formato XML.
 - b. extrae el cuerpo del mensaje a propiedades para poder utilizarlo más adelante.
 - c. Con una función groovy transforma el mensaje añadiendo el escape \ a las " del mensaje (transformamos " en \").
 - d. Generamos el JSON con la estructura correcta que necesita la IA.
 - e. Llamamos al flujo que contiene la IA ("ConexionIA") para que nos genere una respuesta.

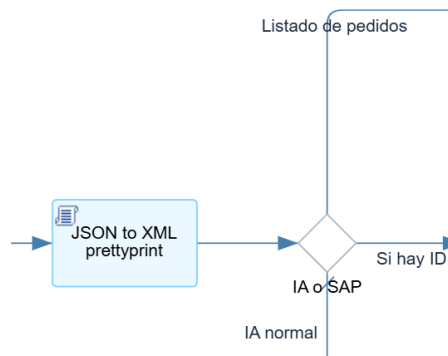


5. El flujo de la IA recibe el mensaje con el body de la consulta a hacer y elimina un header que pasa duplicado. Se realiza la consulta y la respuesta vuelve al flujo general.



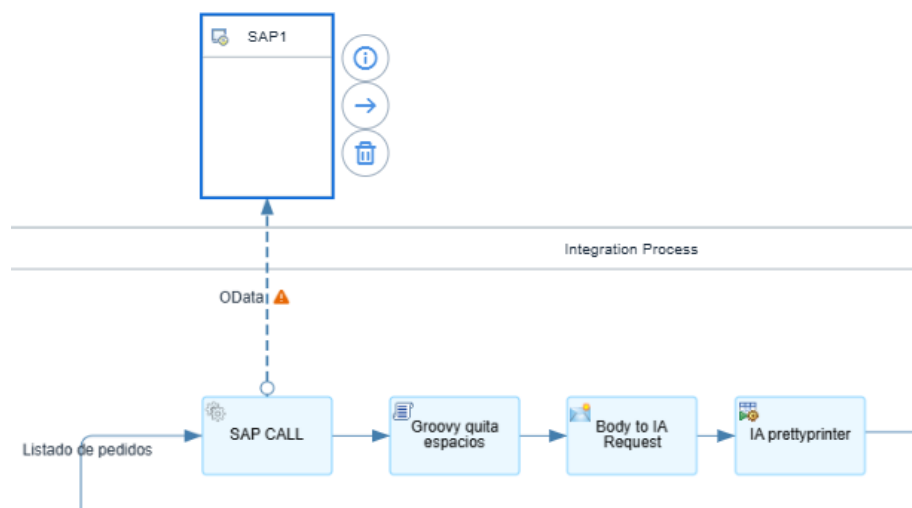
6. La respuesta de la IA llega en formato JSON donde dentro del campo que queremos tenemos una estructura XML con la información que queremos. Con

una función groovy extraemos el XML de la respuesta y la pasamos a un router con 3 opciones.



7. Ruta “Listado de pedidos”

- Si la IA a encontrado “listado” como parte del mensaje del usuario se rellena un campo del XML llamado “listado”. Esto inicia un camino que devolverá como resultado un listado de todos los pedidos en la maquina SAP S/4.



- Realizamos una llamada a SAP a través de un canal OData V2, Así lo hemos configurado.

OData	
General	Connection
CONNECTION DETAILS	
Address: * <input type="text" value="http://sapiot:8080/sap/opu/odata/sap/Z09_PEDIDOS_SRV/"/>	
Proxy Type: * <input type="text" value="On-Premise"/>	
Location ID: <input type="text" value="BECA"/>	
Authentication: * <input type="text" value="Basic"/>	
Credential Name: * <input type="text" value="SAPIOT_Lucas"/>	
CSRF Protected: <input checked="" type="checkbox"/>	
Reuse Connection: <input checked="" type="checkbox"/>	

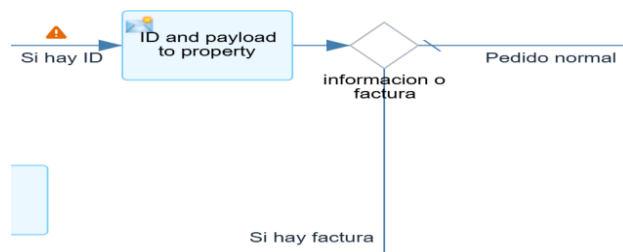
OData	
General	Processing
PROCESSING DETAILS	
Operation Details: * <input type="text" value="Query (GET)"/>	
Resource Path: * <input type="text" value="PedidoSet"/>	

- La respuesta de esta llamada es una estructura XML con la información solicitada.

- d. Utilizamos un Script groovy que transforma el XML en un texto plano para poder enviárselo como consulta a la IA, que preparamos en un content modifier y así poder devolver los datos al usuario de una forma más sencilla y comprensible.
- e. De aquí pasamos a “IA prettyprinter” que es la llamada a un local integration process que describiremos más adelante.
- f. Después de esto el flujo finaliza.

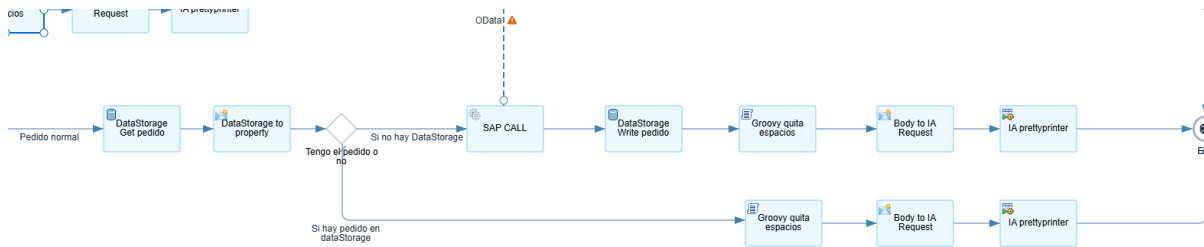
8. Ruta “Si hay ID”

- a. Si la IA a encontrado un “id” como parte del mensaje del usuario se rellena un campo del XML llamado “id”.



- b. Extraemos del cuerpo del mensaje el id encontrado y el cuerpo entero y los guardamos como propiedades para utilizar después.

c. Ruta “Pedido normal”

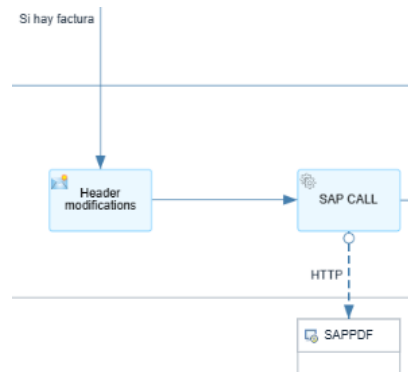


- i. Esta es la ruta por defecto, en caso de que el flujo reciba un id.
- ii. Lo primero revisa si hay un DataStorage con el id recibido que contenga la información del pedido, después, guarda el cuerpo del mensaje como una propiedad.
- iii. Ruta “Si no hay DataStorage”.
 1. En caso de que no haya DataStorage el mensaje seguirá siendo el mismo que al inicio (los 2 cuerpos de mensaje que hemos pasado a propiedades serán iguales) o puede estar vacío.

2. En cualquiera de estas opciones entramos por este camino y hacemos una llamada a SAP para obtener la información del pedido solicitada.
3. Hacemos una llamada a SAP a través de un canal OData V2.
Esta es la configuración que hemos utilizado:

The image displays two side-by-side screenshots of the OData configuration interface. The left screenshot shows the 'Connection' tab with the following fields: Address (http://sapiot-8080/sap/opu/odata/sap/Z09_PEDIDOS_SRV/), Proxy Type (On-Premise), Location ID (BECA), Authentication (Basic), Credential Name (SAPIOT_Lucas), CSRF Protected (checked), and Reuse Connection (checked). The right screenshot shows the 'Processing' tab with the following fields: Operation Details (Read (GET)), Resource Path (PedidoSet(IdPedido='{property.id}')}, Query Options (SelectIdPedido,IdCliente,TxtCliente,Importa,Moneda,FechaDeEntrega,Dirccion,FechaPedido,TipoPedido), Enable Batch Processing (unchecked), Custom Query Options (empty), Content Type (Atom), Page Size (empty), Attach Error Details on Failure (checked), and Timeout (in min) (1).

4. La respuesta de esta llamada es una estructura XML con la información solicitada.
 5. A continuación, guardamos el pedido en el DataStorage para futuras consultas.
 6. Utilizamos un Script groovy que transforma el XML en un texto plano para poder enviárselo como consulta a la IA, que preparamos en un content modifier y así poder devolver los datos al usuario de una forma más sencilla y comprensible.
 7. De aquí pasamos a “IA prettyprinter” que es la llamada a un local integration process que describiremos más adelante.
 8. Después de esto el flujo finaliza.
- iv. Ruta “Si hay pedido en DataStorage”.
1. Esta es la ruta por defecto, en caso de que encuentre el DataStorage.
 2. En este camino se coge el cuerpo del mensaje que será el pedido almacenado en DataStorage y lo pasamos por el Script groovy “quita espacios” explicado anteriormente y preparamos la consulta del usuario sobre el pedido pasándole a la IA el pedido como contexto.
 3. De aquí pasamos a “IA prettyprinter” que es la llamada a un local integration process que describiremos más adelante.
 4. Después de esto el flujo finaliza.
- d. Ruta “Si hay factura”



- i. Esta ruta nos va a devolver la factura.pdf que solicitemos por id.
- ii. Primero modificamos una serie de headers que son necesarios para la llamada.
- iii. Después realizamos una llamada a SAP a través de un canal HTTP porque tenemos que utilizar un parámetro (\$value) que el conector OData no acepta.

HTTP

General **Connection**

CONNECTION DETAILS

Address: *

Query:

Proxy Type:

Location ID:

Method:

Send Body: ☐

Authentication:

Credential Name: *

Timeout (in ms):

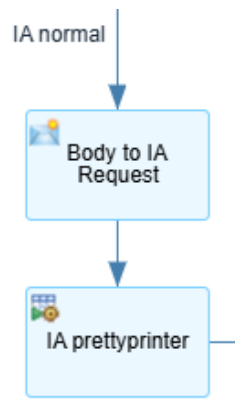
Throw Exception On Failure: ☒

Attach Error Details on Failure: ☒

- iv. Finalmente, SAP nos devuelve el pdf de esa factura y el flujo finaliza.

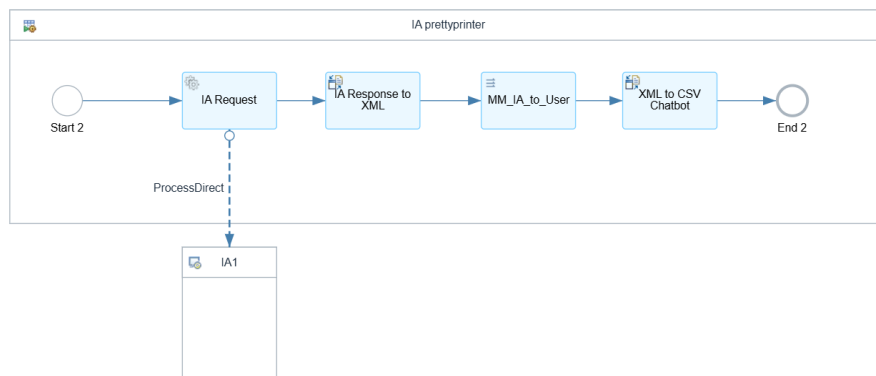
9. Ruta “IA normal”

- a. Esta es la ruta por defecto que en caso de que el XML que devuelve la IA tiene vacíos el campo “id” y el campo “listado” actuará como un chatbot normal y devolverá respuestas lógicas a las consultas del usuario.



- b. Consta de un content modifier que prepara la consulta con la estructura correcta para realizar la consulta a la IA pasando la consulta inicial del usuario como mensaje y la llamada a “IA prettyprinter” que se detalla más adelante.
- c. Después de este paso el flujo finaliza.

10. “IA prettyprinter” local integration process



- a. Este flujo sirve para que, utilizando a la IA, procese y reescriba a lenguaje natural la información que recibe.
- b. Este flujo recibe un mensaje con la estructura correcta para hacer una consulta a la IA.
- c. El mensaje se envía al flujo donde tenemos la IA implementada a través de un processDirect.
- d. Este flujo nos devuelve la respuesta en formato JSON, la transformamos a XML.
- e. Una vez en XML lo mapeamos contra una estructura XML que solo contiene el campo con la respuesta de la IA.
- f. El flujo finaliza convirtiendo el XML de la respuesta a CSV devolviendo así una cadena de texto plano.

4. Tecnologías y Protocolos

- **Chatbot (Frontend y Backend):**
 - **Backend:**
 - **Tecnología:** Node.js (JavaScript)
 - **Descripción:** El backend gestiona la lógica del chatbot, recibe las solicitudes de los usuarios, interactúa con SAP CPI y procesa las respuestas.
 - **Frontend:**
 - **Tecnología:** HTML, JavaScript
 - **Descripción:** El frontend del chatbot proporciona una interfaz interactiva para que los usuarios puedan hacer consultas de manera intuitiva.
- **SAP CPI (Cloud Platform Integration):**
 - **Adaptadores:**
 - **HTTP Adapter:** Para recibir las solicitudes del chatbot y comunicarse con APIs externas e interactuar con la máquina SAP.
 - **OData V2 Adapter:** Para interactuar con la máquina SAP utilizando servicios OData.
 - **ProcessDirect Adapter:** Para comunicarse internamente en SAP CPI entre distintos flujos.
 - **Transformación de Datos:**
 - **JSON a XML:** Transformación de los mensajes de formato JSON (chatbot) a XML (máquina SAP).
 - **XML a JSON:** Transformación de las respuestas de XML a JSON para el chatbot.
- **Máquina SAP (SAP S/4HANA):**
 - **OData Services:** Servicios OData que permiten consultar datos de la máquina SAP mediante peticiones RESTful.
- **Protocolo de Comunicación:**
 - **HTTP/HTTPS:** Utilizado para la comunicación entre el chatbot y SAP CPI, y entre SAP CPI y los servicios web externos.
 - **REST:** El chatbot envía y recibe mensajes a través de llamadas API RESTful en formato JSON.
 - **Odata:** Si la máquina SAP expone servicios OData, SAP CPI lo gestionará mediante el adaptador OData para intercambiar mensajes en formato XML.
- **Seguridad:**
 - **Autenticación:**

- **Basis Auth (SAP CPI):** Utilizado para la autenticación de usuarios o sistemas entre el chatbot y las APIs expuestas por SAP CPI.
- **Basic Auth (SAP S/4):** Utilizado para la autenticación de usuarios o sistemas con la máquina SAP.

5. Seguridad

5.1 Autenticación

- **OAuth (SAP CPI):** Utilizado para la autenticación de usuarios o sistemas entre el chatbot y las APIs expuestas por SAP CPI.
- **Basic Auth:** Utilizado para la autenticación de usuarios o sistemas con la máquina SAP.

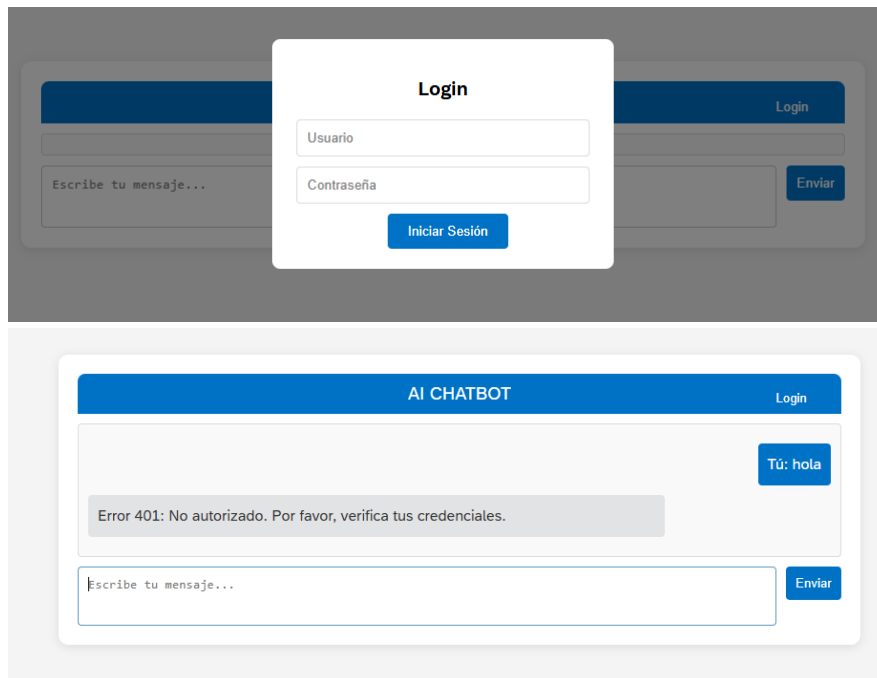
5.2 Cifrado de Datos

- **Cifrado en tránsito:**
 - Las comunicaciones entre el chatbot y SAP CPI se realizan a través de **HTTPS**. Este protocolo asegura que los datos estén protegidos mediante **cifrado TLS (Transport Layer Security)** durante la transmisión, evitando interceptaciones o accesos no autorizados.
- **Cifrado en reposo:**
 - No se contempla cifrado de datos en reposo como parte de esta integración, ya que los datos no se almacenan de forma persistente en el chatbot ni en SAP CPI.

6. Manejo de Errores y Excepciones

6.1 Tipos de Errores Esperados

- **Errores de autenticación:**
 - **Descripción:** Al iniciar el chatbot aparece un pop-up de login en el que debes de meter las credenciales de SAP CPI. En caso de error avisa.



- **Respuesta:** Esto nos dará un error 401 y tendremos que volver a meter las credenciales en el botón de arriba a la derecha que pone “Login”.
- **Errores de formato de datos:**
 - **Descripción:** Los mensajes enviados entre los sistemas pueden contener formatos incorrectos (por ejemplo, JSON mal formateado o XML no válido).
 - **Respuesta:** SAP CPI valida los mensajes antes de enviarlos a la máquina SAP. Si se detecta un formato incorrecto, se registra el error y se devuelve una respuesta de error en el chat.
- **Errores de procesamiento en la máquina SAP:**
 - **Descripción:** La máquina SAP puede devolver errores cuando no puede procesar una solicitud (por ejemplo, un ID de factura no válido).
 - **Respuesta:** SAP CPI captura los errores de SAP (como códigos de estado HTTP 500, 404, etc.) y envía un mensaje de error informativo al chatbot.
- **Errores en APIs externas (IA):**
 - **Descripción:** Fallos en la interacción con servicios externos, como tiempos de espera agotados, entradas no válidas o respuestas no válidas.
 - **Respuesta:** Se registran los errores y se informa en el chat del error.

7. Pruebas

7.1 Estrategia de Pruebas

1. Pruebas de Unidad:

- a. **Objetivo:** Validar el correcto funcionamiento de componentes individuales, tanto en el backend del chatbot como en las integraciones de SAP CPI.
 - b. **Componentes:**
 - i. **Backend del chatbot:** Se realizarán pruebas unitarias para asegurarse de que los módulos de procesamiento de mensajes, llamadas al flujo de integración, y gestión del historial de conversaciones funcionan correctamente.
 - ii. **SAP CPI:** Pruebas unitarias en las transformaciones de datos, adaptadores (HTTP, OData), y mapeo de datos entre los sistemas.
2. **Pruebas de Integración:**
- a. **Objetivo:** Verificar que los diferentes componentes del sistema (chatbot, SAP CPI y la máquina SAP) interactúan correctamente entre sí.
 - b. **Escenarios:**
 - i. **Prueba de interacción del chatbot con SAP CPI:** Asegurarse de que las solicitudes del chatbot se envíen correctamente a SAP CPI y que las respuestas se reciban en el formato adecuado.
 - ii. **Prueba de interacción de SAP CPI con la máquina SAP:** Validar que SAP CPI pueda comunicarse correctamente con la máquina SAP, enviar datos y recibir respuestas.
 - iii. **Prueba de la IA:** Verificar que la IA se integre correctamente y entregue los datos esperados.
 - c. **Herramientas:** Postman, SAP CPI Monitor, y logs detallados para identificar cualquier error durante las interacciones.
3. **Pruebas de Usuario (UAT - User Acceptance Testing):**
- a. **Objetivo:** Validar que el sistema cumple con los requisitos del negocio y las expectativas del usuario final.
 - b. **Escenarios:**
 - i. **Interacción del usuario con el chatbot:** Probar diferentes tipos de consultas y verificar que el chatbot responde correctamente según lo esperado (incluyendo las respuestas del sistema SAP).
 - ii. **Pruebas de comportamiento ante errores:** Validar que el chatbot maneje correctamente los errores de SAP CPI o la máquina SAP y que entregue un mensaje adecuado al usuario.
 - iii. **Flujos de usuario completos:** Realizar pruebas de extremo a extremo desde que el usuario ingresa una consulta hasta que recibe la respuesta, incluyendo la autenticación y validación de resultados.

7.2 Pruebas de Seguridad

1. Pruebas de Autenticación y Autorización:

- a. **Objetivo:** Validar que solo los usuarios y servicios autorizados pueden acceder a los recursos.
- b. **Escenarios:**
 - i. Prueba de autenticación fallida (por ejemplo, con credenciales incorrectas o tokens caducados).
- c. **Herramientas:** Postman, el chatbot.