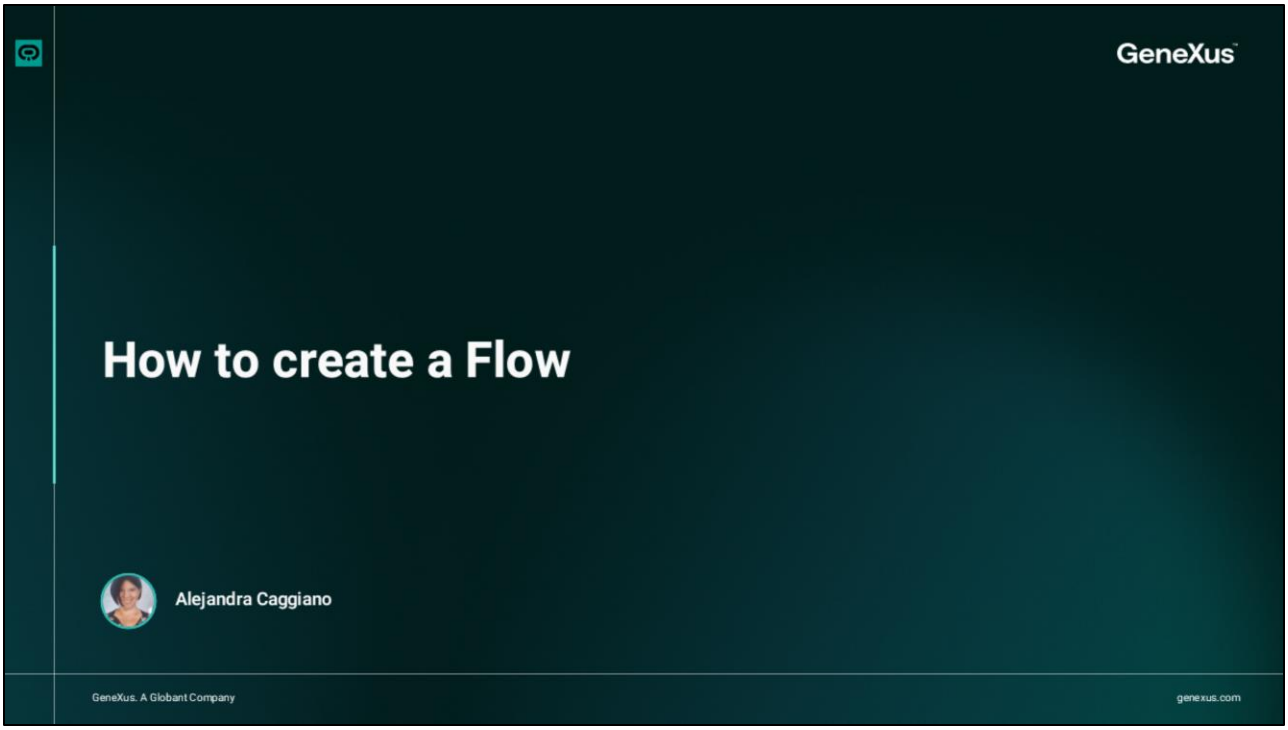
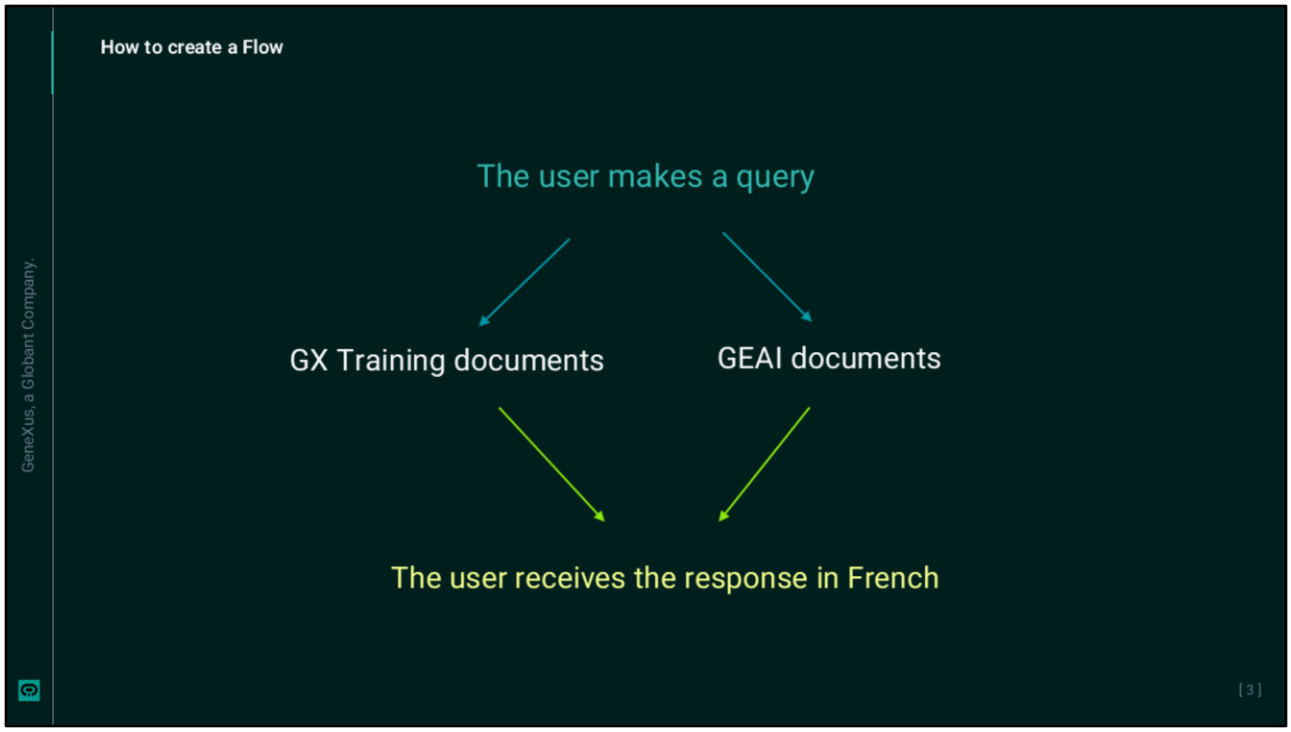
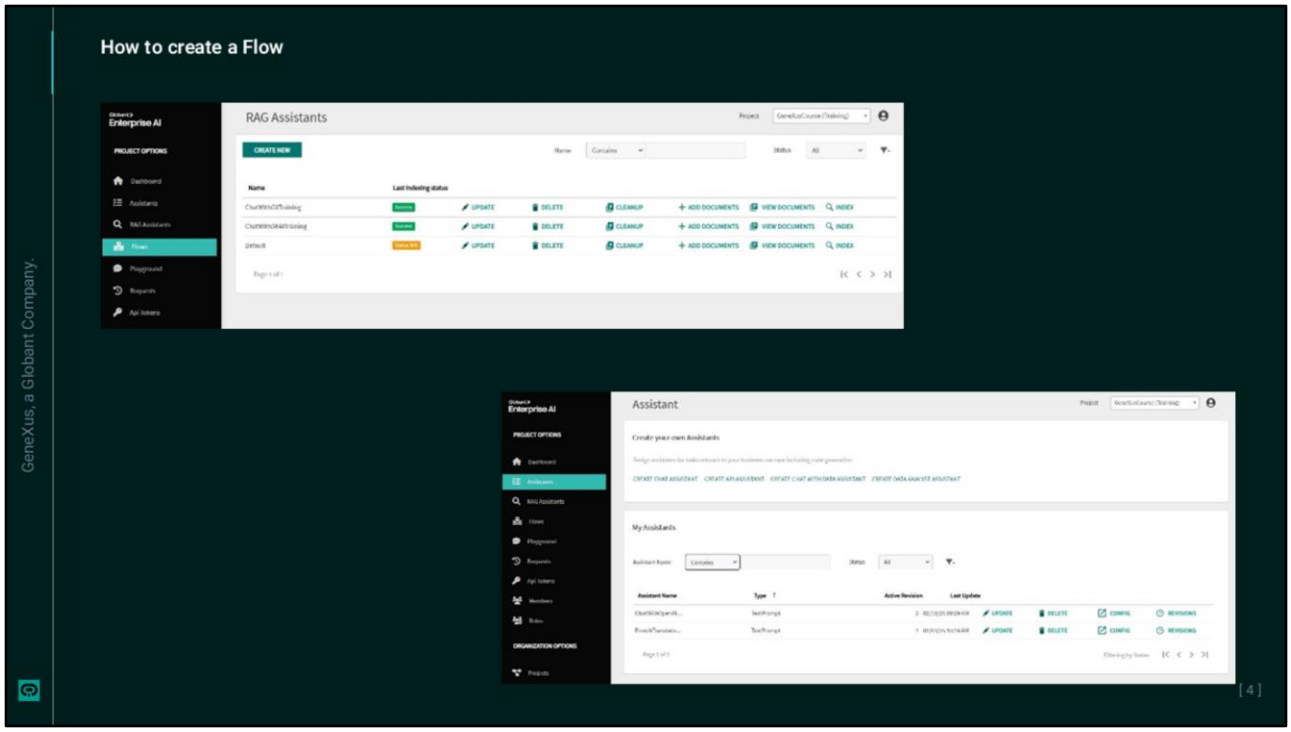

Globant
Enterprise AI



Vamos a crear nuestro primer Flujo.



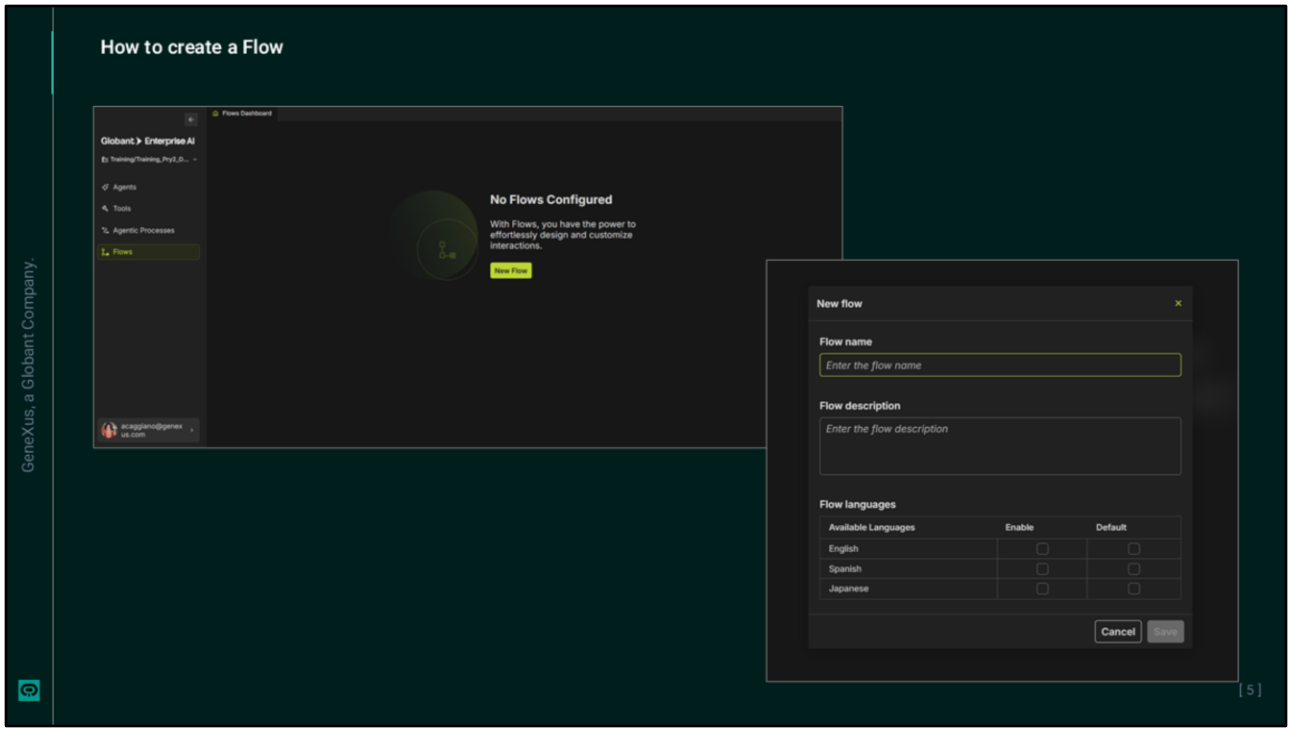
El objetivo es ofrecer al usuario final la posibilidad de realizar una consulta sobre algún tema relativo a capacitación de GeneXus o Globant Enterprise AI, y la respuesta obtenida sea traducida al Frances.



En primer lugar, observemos los asistentes definidos.

Accedemos al backoffica, seleccionamos el proyecto y vemos que tenemos definidos dos RAG Assistants: uno de ellos interactúa con documentos de GeneXus training, y otro que interactúa con documentos de Globant Enterprise AI.

Además, tenemos un Chat Assistant que traduce texto de ingles a francés.



Bien. Vamos a crear el Flujo. Desde el menú, seleccionamos **The Lab**, y accedemos a una nueva ventana (Flow Builder), donde podremos crear y administrar o editar los flujos asociados al proyecto seleccionado.

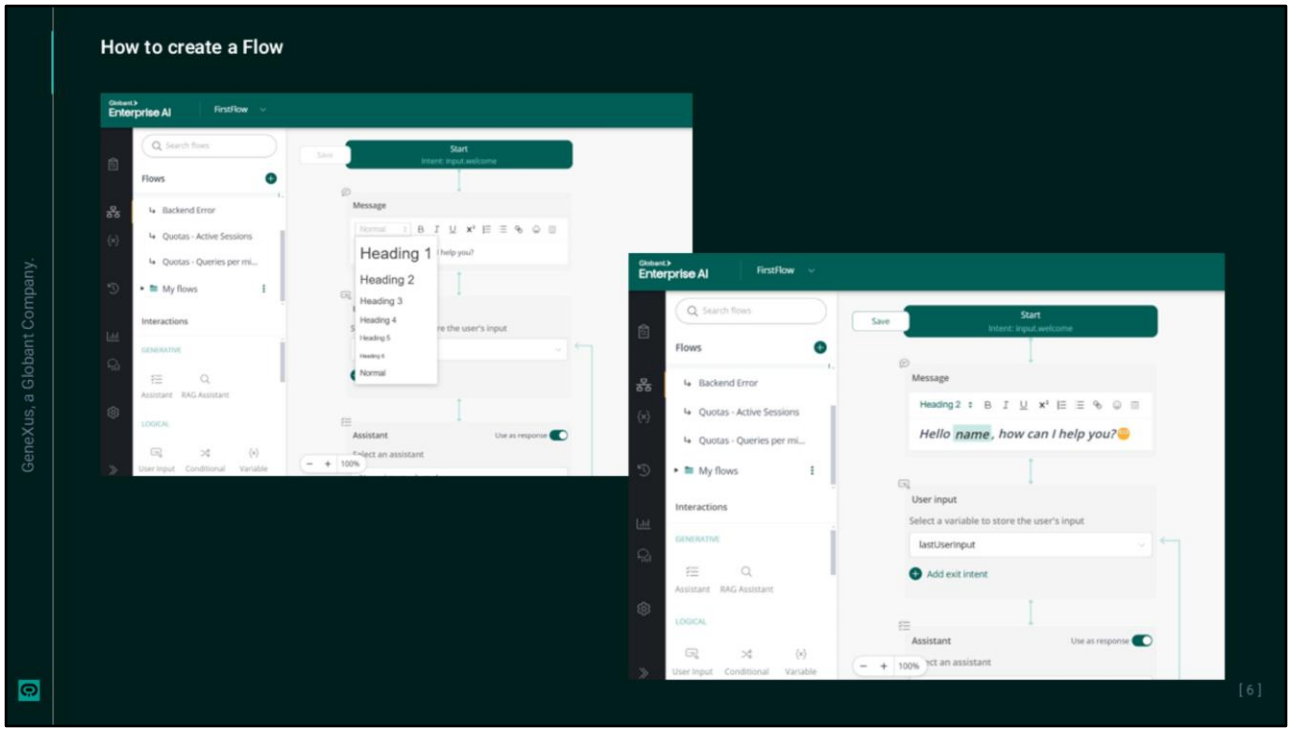
Para comenzar a crear un nuevo flujo, seleccionamos **New Flow**. Aquí debemos indicar:

- El nombre del flujo que permitirá identificarlo fácilmente. En nuestro ejemplo vamos a poner “FrenchTrainingDocumentation”
- Y una descripción, que, si bien es opcional, recomendamos agregar, para comprender su propósito.

Resolves GeneXus and GEAI queries, and translates their response into French

- Finalmente, la opción Languages, establece el idioma en el que se configurará el flujo y define el idioma de los mensajes codificados. Es posible seleccionar varios idiomas, por lo que el mismo mensaje puede estar disponible en diferentes idiomas.

Presionamos ‘Save”.

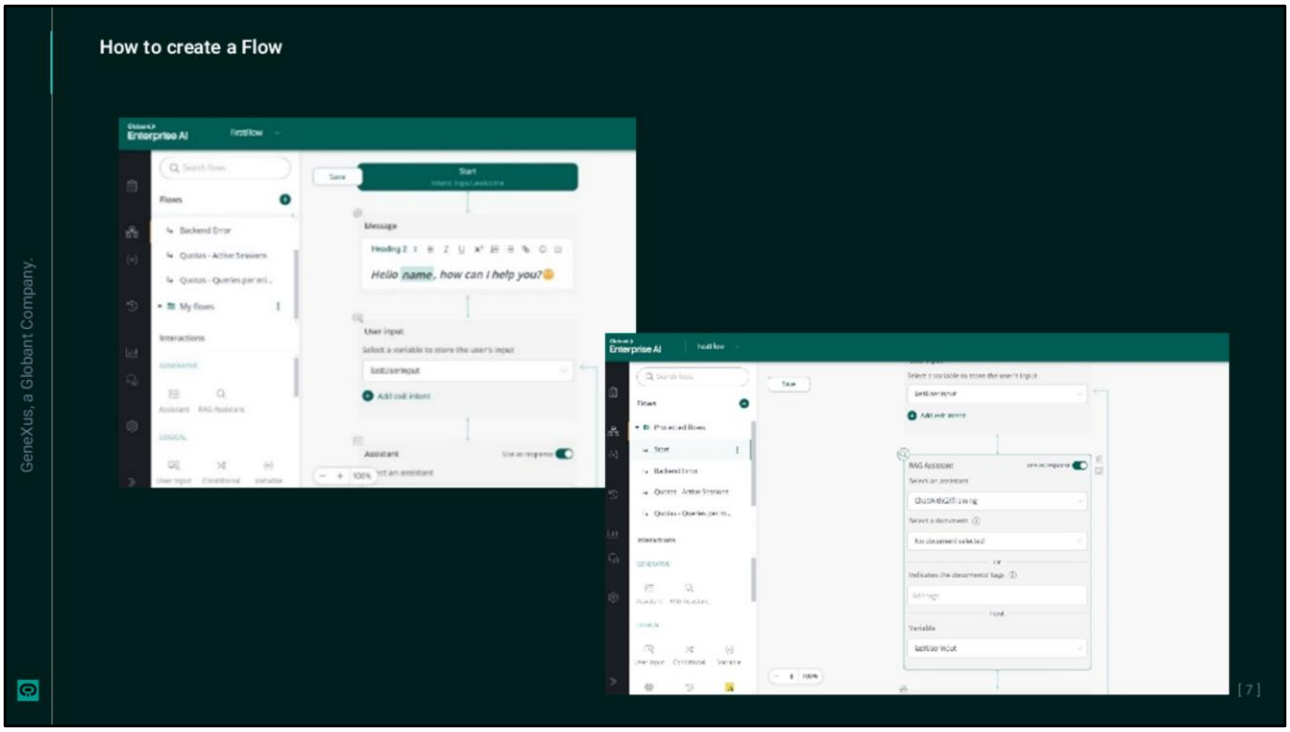


Debemos ahora agregar los asistentes y personalizar el comportamiento. Para eso, presionamos “Edit”.

Este nodo inicial, llamado Start, se genera automáticamente, y establece una configuración básica que se puede personalizar, añadiendo o modificando interacciones, mensajes y asistentes según sea necesario. Marca el comienzo del flujo de conversación. Este nodo se activa cada vez que se dispara el flujo, y tiene como objetivo principal iniciarlo y, a partir de ahí, se conectan los demás nodos que gestionan la interacción con el usuario.
Es posible cambiar el nombre del nodo, desde Flows, Rename Flow.

El siguiente nodo es el nodo “Message”, y es quien se encarga de enviar el mensaje de bienvenida o cualquier otro mensaje inicial configurado.
En este caso, el mensaje es: "Hola {nombre}, ¿en qué puedo ayudarte?". donde, {nombre} representa una variable dinámica que se completa con el nombre del usuario.
Este mensaje es personalizable, pudiéndose también formatear con distintos estilos como podemos ver desde aquí…..y agregar listas, enlaces y emojis. Personalicemos nuestro saludo inicial…
Este último botón permite insertar variables dinámicas en el mensaje.
Al seleccionarlo se muestra una lista de las variables disponibles en el sistema, y también es posible definir nuevas variables personalizadas.

En caso de querer eliminar este nodo Message, lo hacemos desde aquí.



Bien. A continuación, el flujo captura la entrada del usuario a través del nodo User input y la guarda en la variable lastUserInput para su uso posterior.

Luego vemos el nodo Assistant, donde se debe configurar el asistente para realizar acciones específicas. Este nodo utiliza la información almacenada en la variable lastUserInput, que contiene el texto ingresado por el usuario final en el nodo User input.
Y si presionamos aquí, vemos que se nos ofrecen los asistentes definidos en la opción Assistants del proyecto donde nos hemos posicionado en el backoffice.
Pero para nuestro ejemplo, necesitamos elegir uno de los RAG Assistants. Así que eliminamos este nodo, y desde este menú vemos que podeos definir interacciones con Assistants y con RAG Assistants.

Así que arrastramos la opción RAG Assistant hasta la ubicación deseada dentro del Flujo. En nuestro ejemplo lo ubicamos a continuación del nodo User input.
Elegimos el asistente ChatWithGXTraining, y luego este campo permite seleccionar un documento específico que actuará como fuente para responder a la solicitud del usuario final. De hacerlo así, el asistente solo utilizará dicho documento como referencia.
También es posible agregar etiquetas que permitirán al asistente buscar solamente en los documentos que coincidan con las etiquetas indicadas para responder a la solicitud del usuario final.

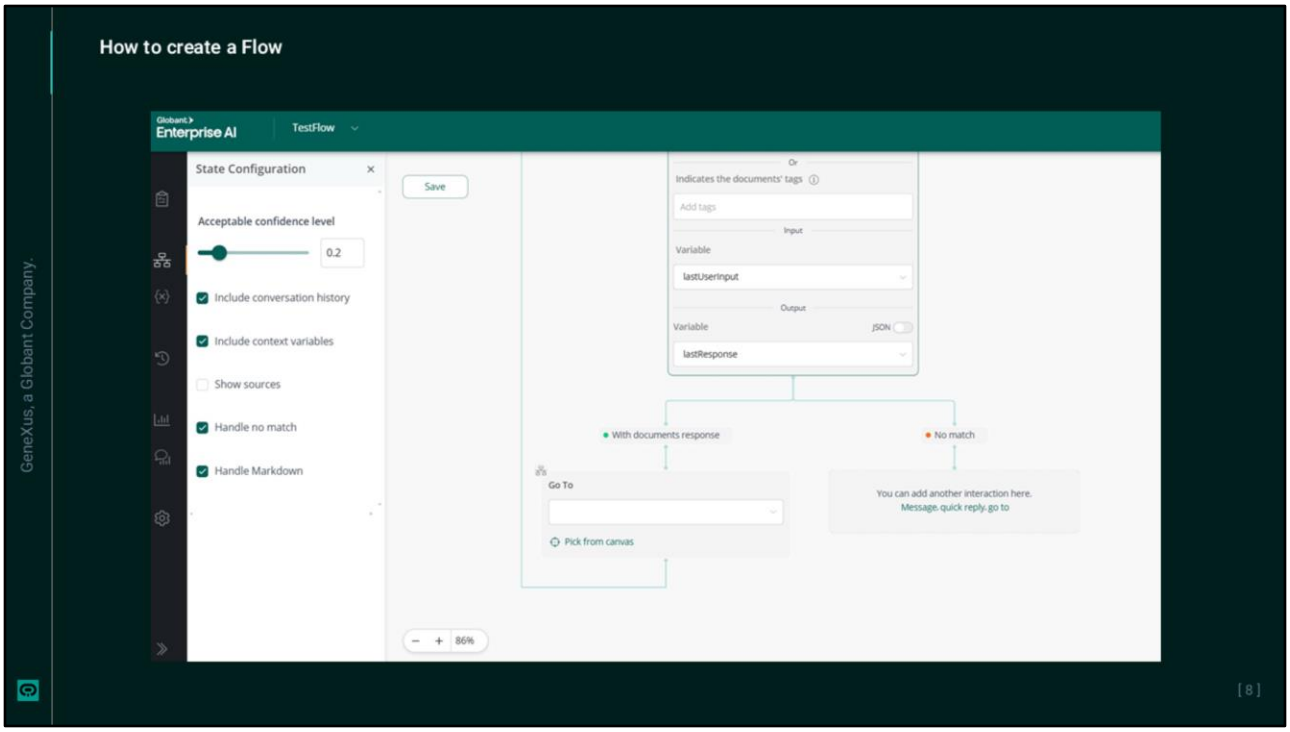
En la sección Input, se muestra un campo llamado Variable. Allí se define la variable que se utilizará como entrada para el nodo; por ejemplo, "lastUserInput".

Como mencionamos, esta variable contiene la entrada más reciente del usuario y se utilizará como base para la consulta de los documentos o etiquetas seleccionados.

La respuesta del asistente se puede almacenar en esta misma variable lastUserInput o en una nueva variable, según se considere. Si observamos aquí, por defecto la opción Use as response está habilitada, lo que hace que la respuesta del asistente se use directamente en la variable lastUserInput.

En nuestro ejemplo queremos guardarla en otra variable, así que desactivamos esa opción, y vemos que se habilita el campo Output donde podemos seleccionar o crear una nueva variable en la que se almacenará la salida generada por el nodo.

En este caso nos interesa guardar la respuesta del RAG Assistant en una nueva variable llamada lastResponse.



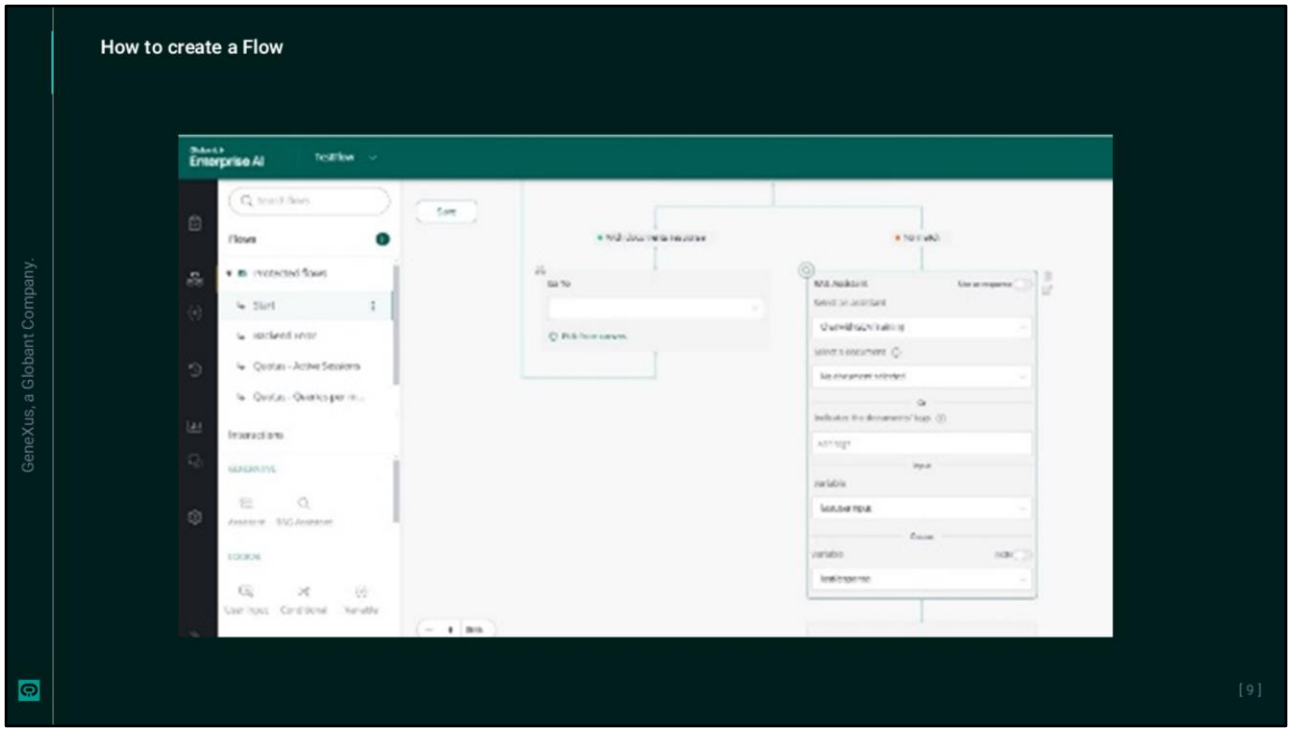
Bien. Observemos ahora este ícono del lápiz a la derecha del nodo. Al seleccionarlo se abre el menú "Configuración de estado" donde es posible modificar lo siguiente:

- Nivel de confianza aceptable: el valor predeterminado es 0,2, e indica el nivel de confianza en la coincidencia de los fragmentos que obtiene el RAG assistant después de buscar en la base de datos vectorial. Sólo se considerarán válidas las respuestas que cumplan o superen este nivel de confianza.
- La opción de Incluir el historial de conversaciones permite al asistente tener en cuenta los mensajes anteriores al generar respuestas.
- Incluir variables de contexto: también se encuentra habilitada de forma predeterminada y permite que las variables de contexto almacenadas durante la conversación se envíen a los asistentes para que puedan utilizarlas dentro de las indicaciones.
- La opción Mostrar fuentes, permite al asistente mostrar las fuentes de información utilizadas para generar la respuesta, lo que ayuda a brindar transparencia y validar la precisión de la respuesta.
- Y también la posibilidad de Manejar respuesta vacía: Esta opción está deshabilitada de forma predeterminada y maneja situaciones en las que el asistente no puede generar una respuesta válida. Cuando está habilitado, abre una nueva rama de flujo, lo que permite configurar acciones específicas a seguir en caso de que no exisa respuesta disponible. En esta nueva rama, es posible agregar cualquiera de los componentes disponibles en el menú para continuar la interacción según sea necesario.

En nuestro ejemplo, si el RAG Assistant encuentra la respuesta en los documentos de GeneXus training, se desea que el flujo llame al asistente correspondiente para traducir la respuesta al francés y almacenarla en la misma variable de entrada (lastResponse).

Pero si no se encuentra ninguna respuesta, el flujo deberá pasar a un segundo RAG Assistant configurado con los documentos utilizados en los cursos de Globant Enterprise AI.

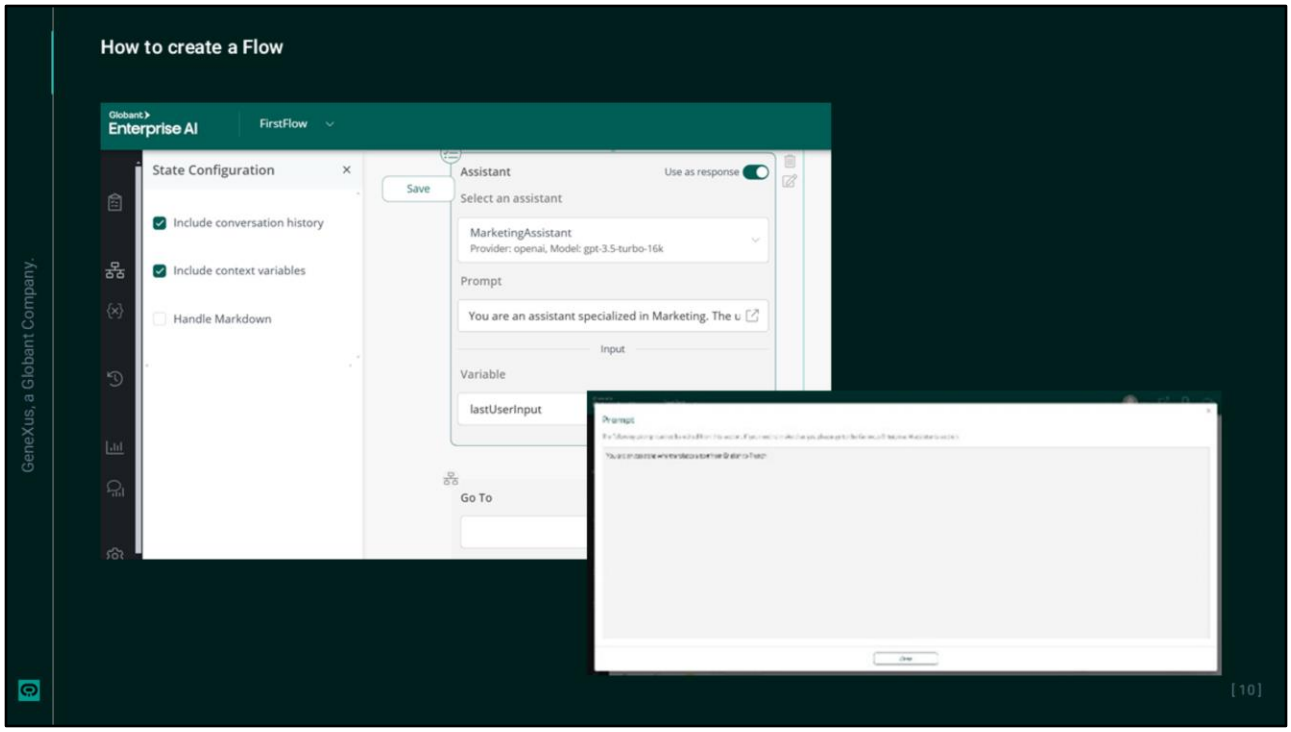
Así que marcamos la opción Handle no match, y vemos que se abre una nueva rama.



Entonces, cuando la consulta del usuario no encuentre respuesta en el primer RAG Assistant, deberá continuar por este lado.

Así que cerramos el cuadro de configuración, y arrastramos una nueva interacción con un RAG assistant. Seleccionamos el asistente correspondiente, que tendrá a lastUserInput como variable de ingreso, que recordemos, es la variable que contiene la última entrada del usuario.

La respuesta generada por este RAG Assistant se deberá almacenar en la variable lastResponse, así que desactivamos esta marca para que nos permita indicar la variable de salida.

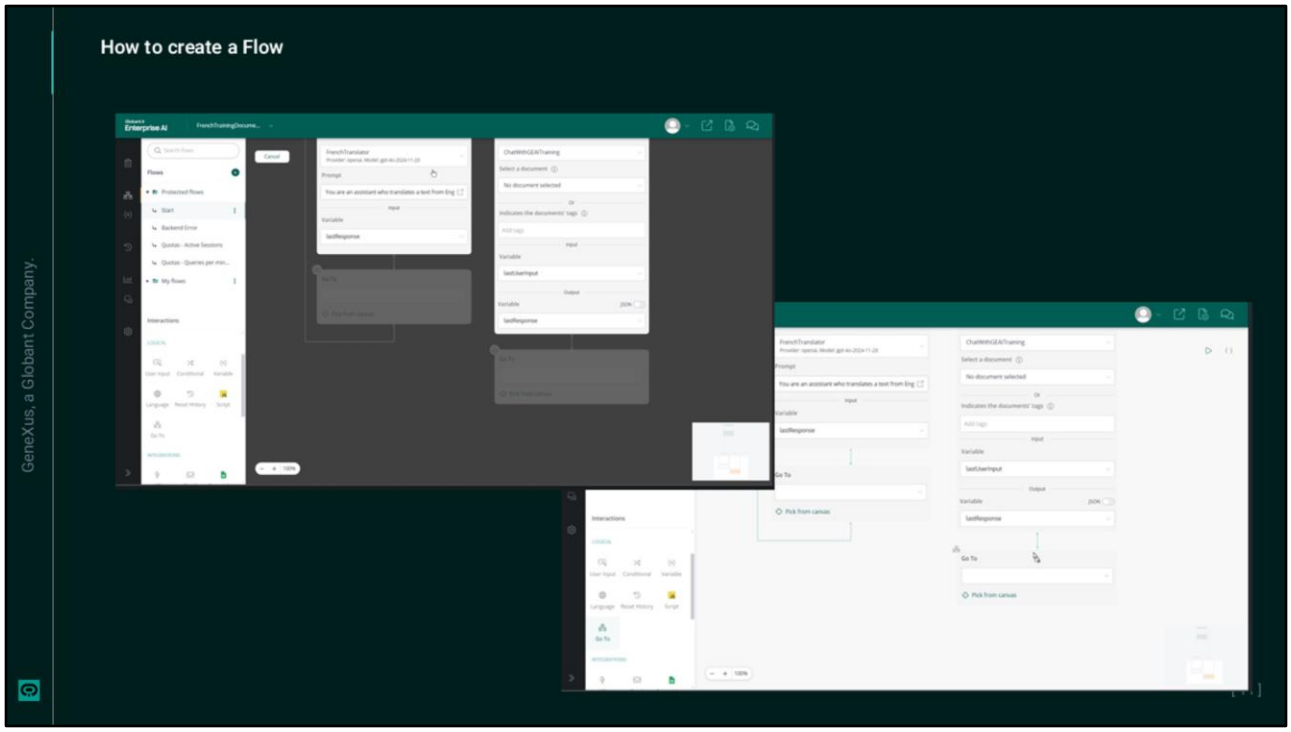


Bien. Pero si la respuesta a la consulta del usuario se encuentra con la acción del primer RAG Assistant entonces se deberá traducir a Frances.

Así que elegimos la interacción con un Assistant y lo arrastramos hasta su ubicación en el Flujo.

Elegimos entonces el asistente que traduce al Frances, y vemos que el campo Prompt muestra el mensaje que define la tarea del asistente. Este mensaje es de solo lectura y no se puede editar.

Almacenamos su respuesta en la misma variable lastResponse.



Finalmente, el nodo Go to se utiliza para conectar diferentes flujos dentro de la interacción. Esto permite seleccionar el flujo al que se desea redirigir al usuario final en función de su interacción actual.

Al hacer clic en la barra en blanco (que corresponde al menú desplegable) se muestra la lista de flujos disponibles:

Además, la opción "Pick from canvas" permite seleccionar un nodo directamente desde el lienzo, facilitando la navegación y la conexión entre diferentes partes del Flujo.
En este caso, hay dos nodos 'Go to':

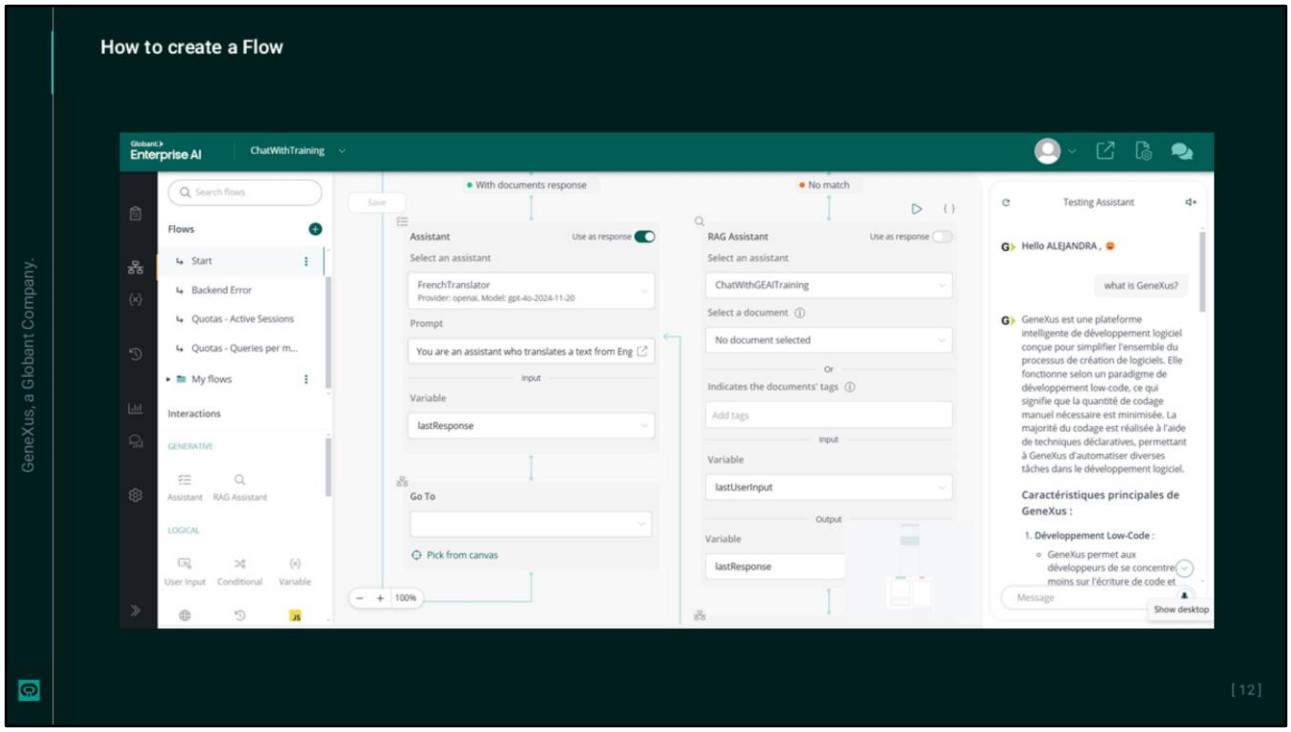
- Uno de ellos regresa al nodo Input del usuario'. Dado que el nodo anterior (que traduce a francés) no almacena la respuesta en una nueva variable y simplemente redirige a la 'Entrada del usuario', la respuesta recibida por el usuario final es la respuesta misma a su pregunta traducida al francés.
- El otro nodo 'Go to' debe enviar la respuesta de este segundo RAG Assistant al nodo que se encarga de traducir al francés utilizando la variable lastResponse.

Así que desde el menú elegimos la interacción Go to, y la arrastramos hasta su posición. Elegimos Pick from canvas e indicamos el nodo Assistant al cual debemos redirigir esta salida.

La variable lastResponse contiene la respuesta obtenida de cualquiera de los dos RAG Assistants ejecutados previamente en el Flujo.

Repasemos entonces la secuencia de las variables:
El usuario ingresa una consulta y se recibe en la variable lastUserInput.
El primer RAG Assistant recibe esa entrada y si tiene la respuesta entonces la guarda en la variable lastResponse que, a su vez, es recibida por el asistente que traduce al Frances.
En caso de no tener la respuesta, la consulta del usuario en lastUserInput ingresa al segundo RAG assistant que guarda su respuesta en la variable lastResponse y la envia al chat assistant para su traducción.

De esta forma, el flujo asegura que el usuario final reciba la respuesta en frances, independientemente del RAG Assistant que la generó.

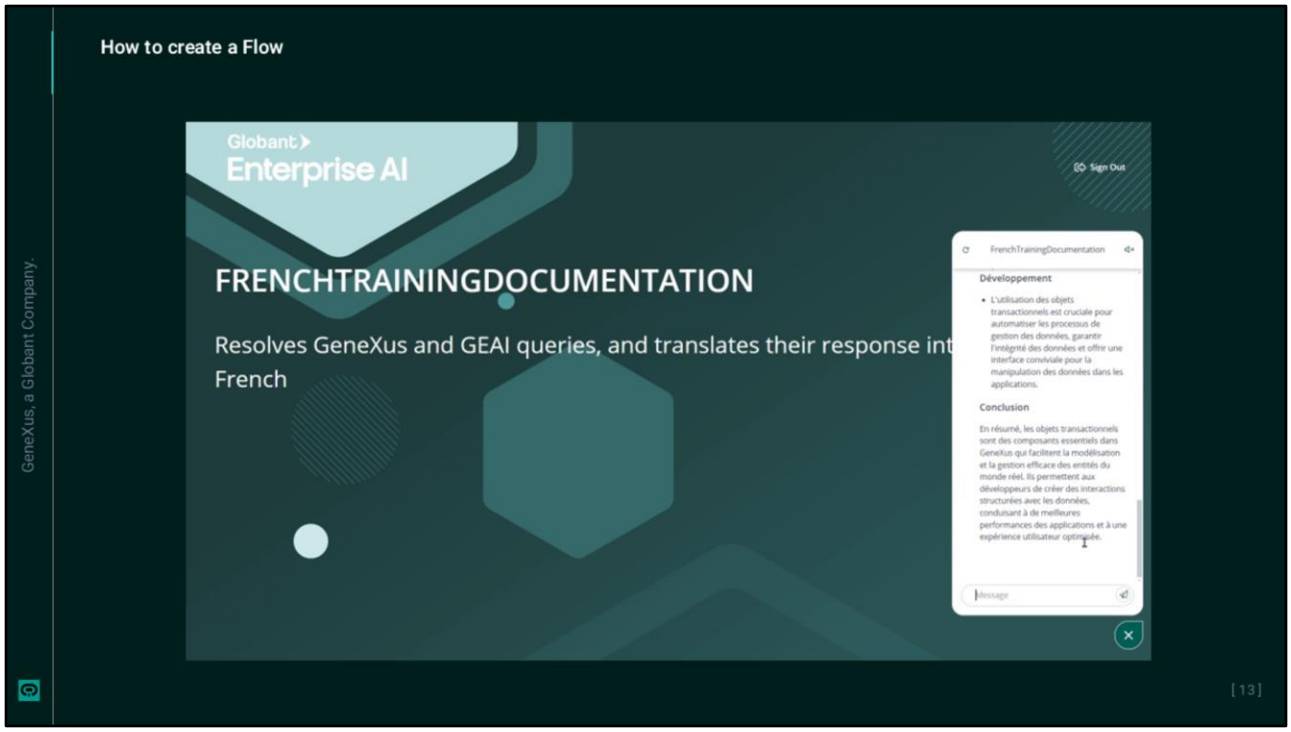


Bien. Llegados a este punto estamos en condiciones de testear el Flujo, así que primero lo salvamos presionando Save.

Tenemos tres opciones para testear el Flujo desde esta pantalla de diseño.
Estas dos opciones abren una ventana a la derecha de la pantalla, mientras que esta otra opción abrirá una nueva ventana en el navegador.
Probemos desde la propia pantalla.

Vemos el mensaje inicial. Preguntemos ¿Que es GeneXus?

Obtenemos la respuesta en francés



Ahora probemos desde esta otra opción…

De esta forma hemos creado y testeado nuestro primer Flujo. A continuación, crearemos un Flujo con componente condicional.


Globant
Enterprise AI