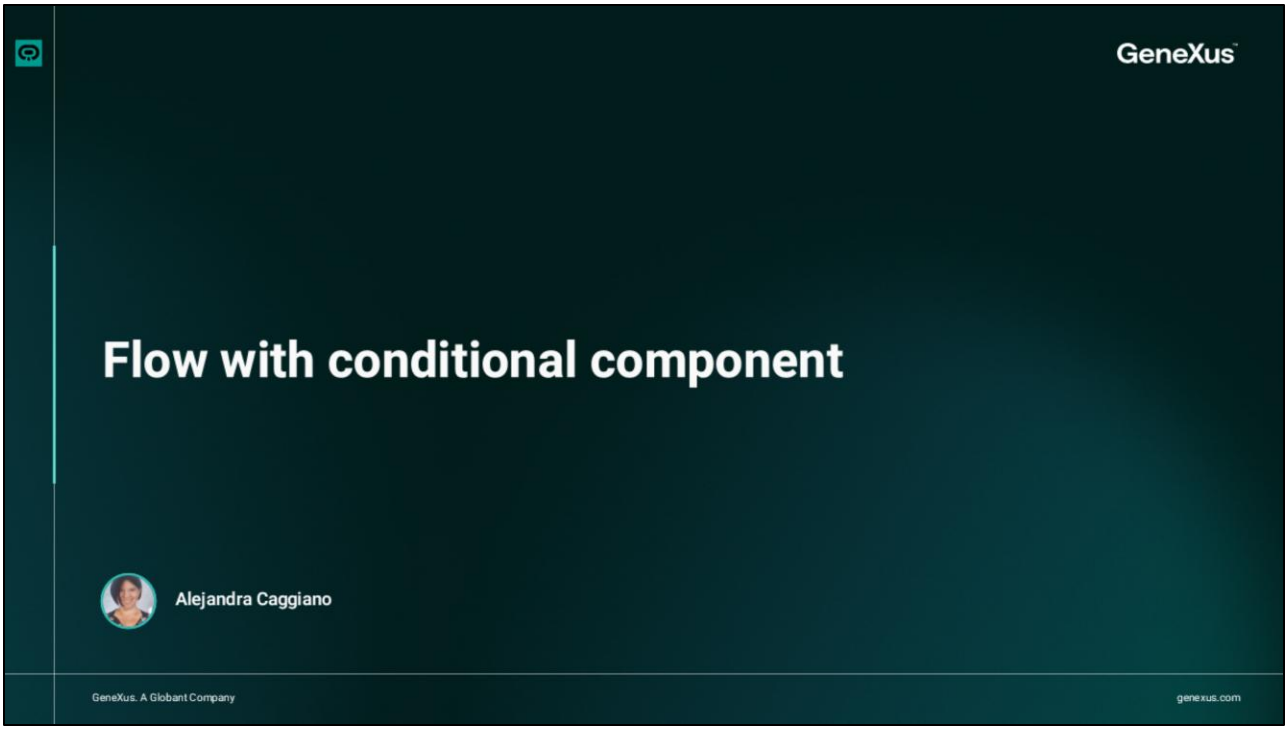
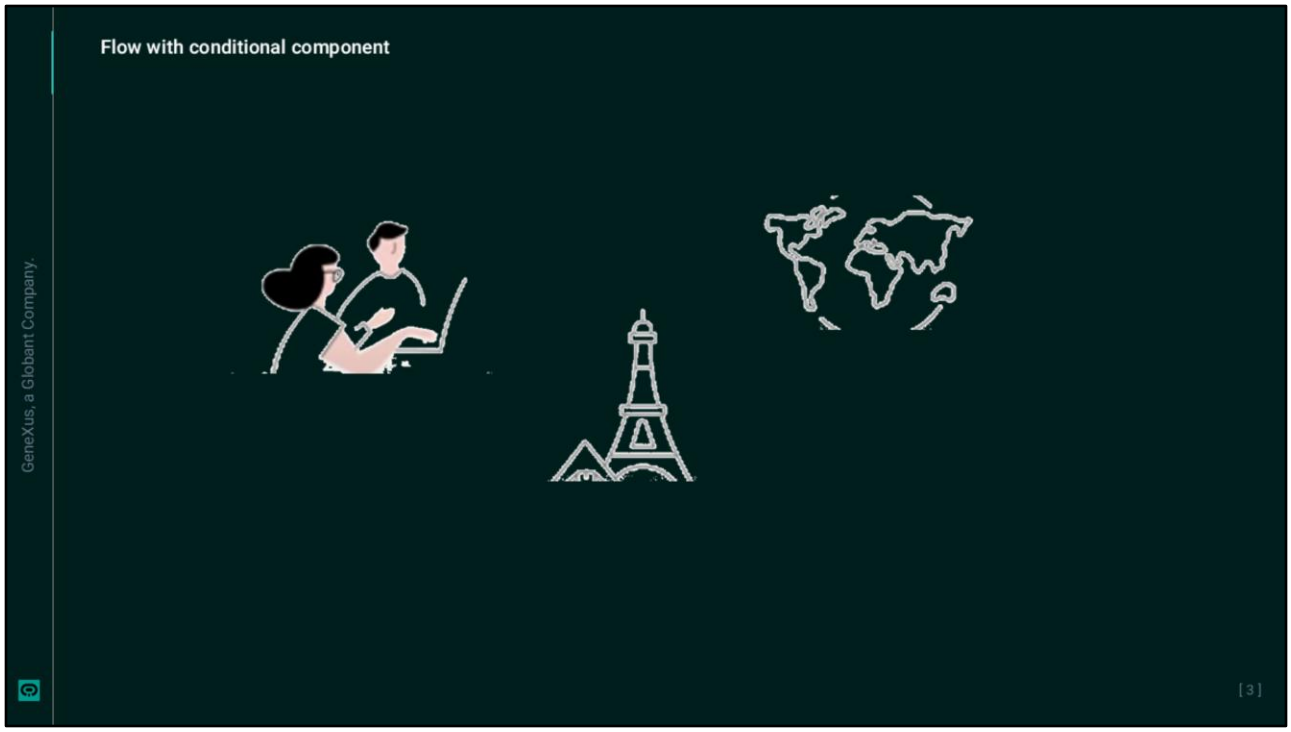
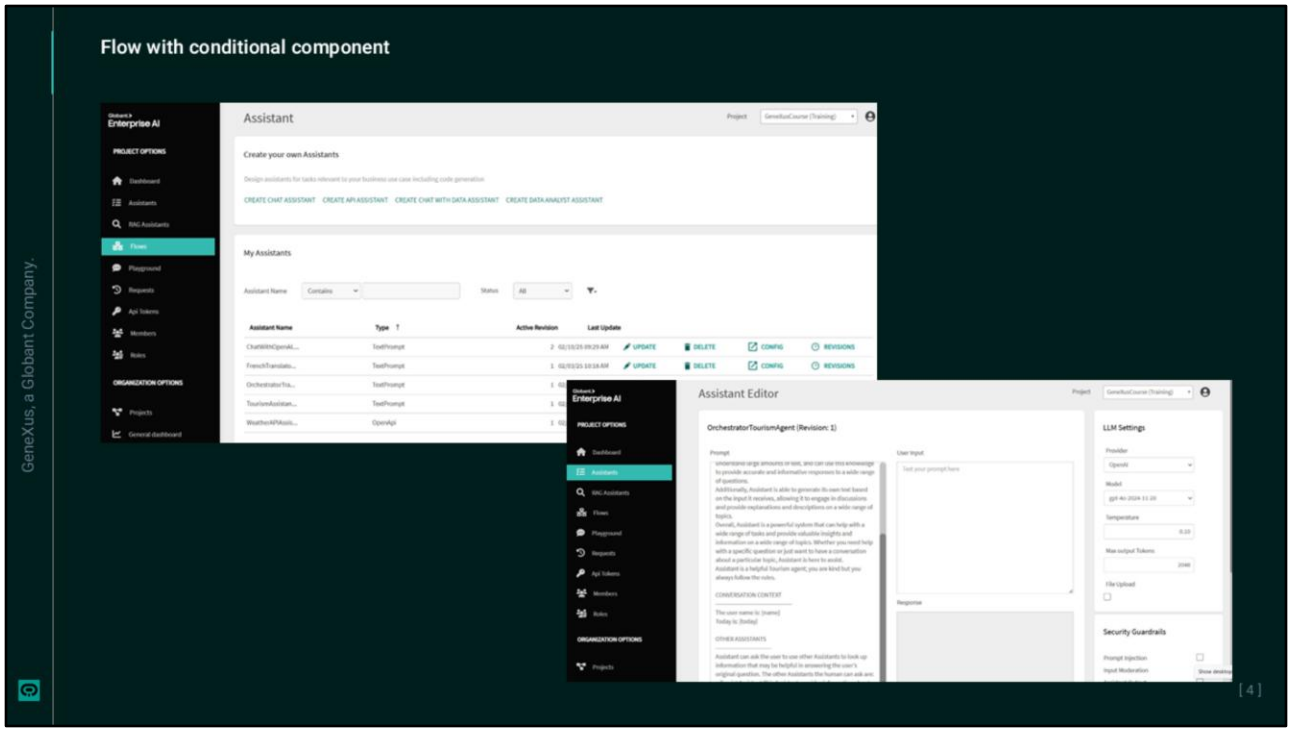

Globant
Enterprise AI



Previously, we created a Flow that interacted with two RAG Assistants and translated the final response into French through a Chat Assistant.



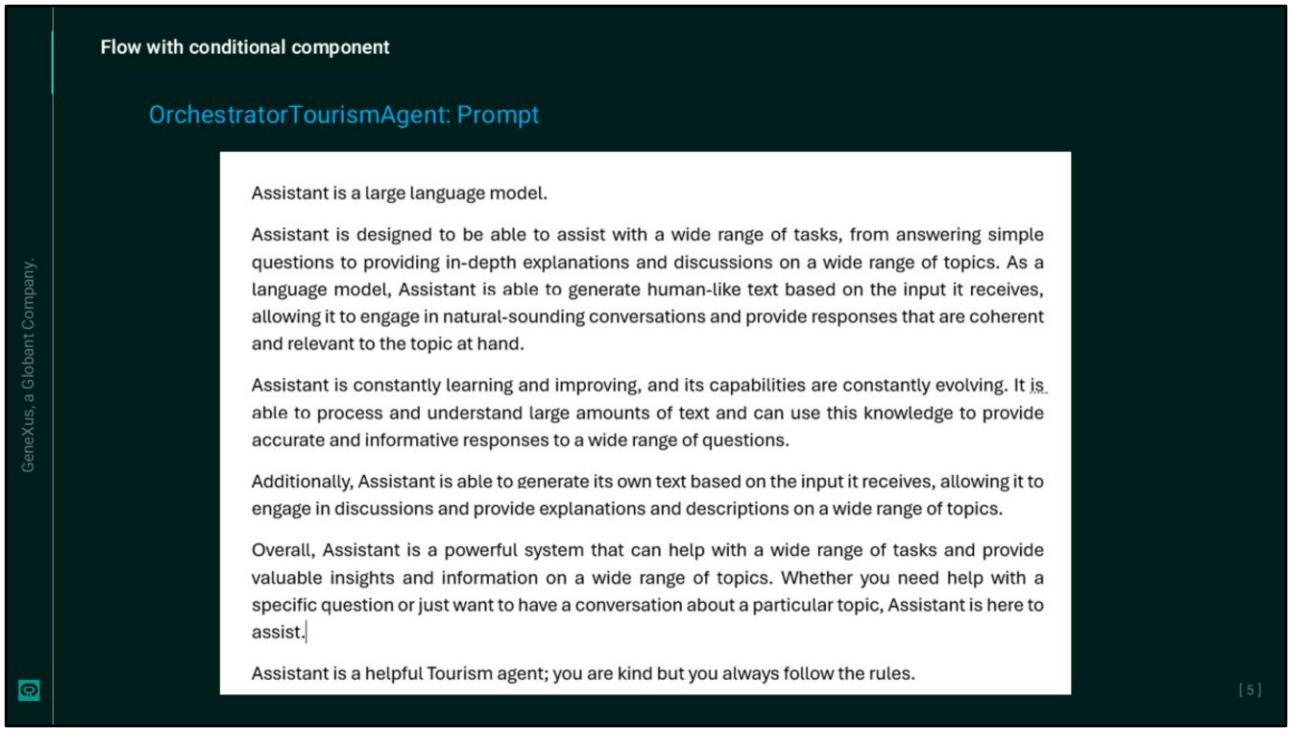
We are now going to create a Flow that will provide services to tourists, allowing them to consult information related to tourist attractions, as well as the current weather conditions in any city in the world.



For this, we have a Chat Assistant that provides information on tourist attractions, and an API Assistant that is responsible for obtaining the current weather in different cities.

We also need an orchestrator to understand the end user's intent and redirect the conversation accordingly. This orchestrator will be a Chat Assistant capable of maintaining the context of the conversation, so the interaction will be fluid and coherent.

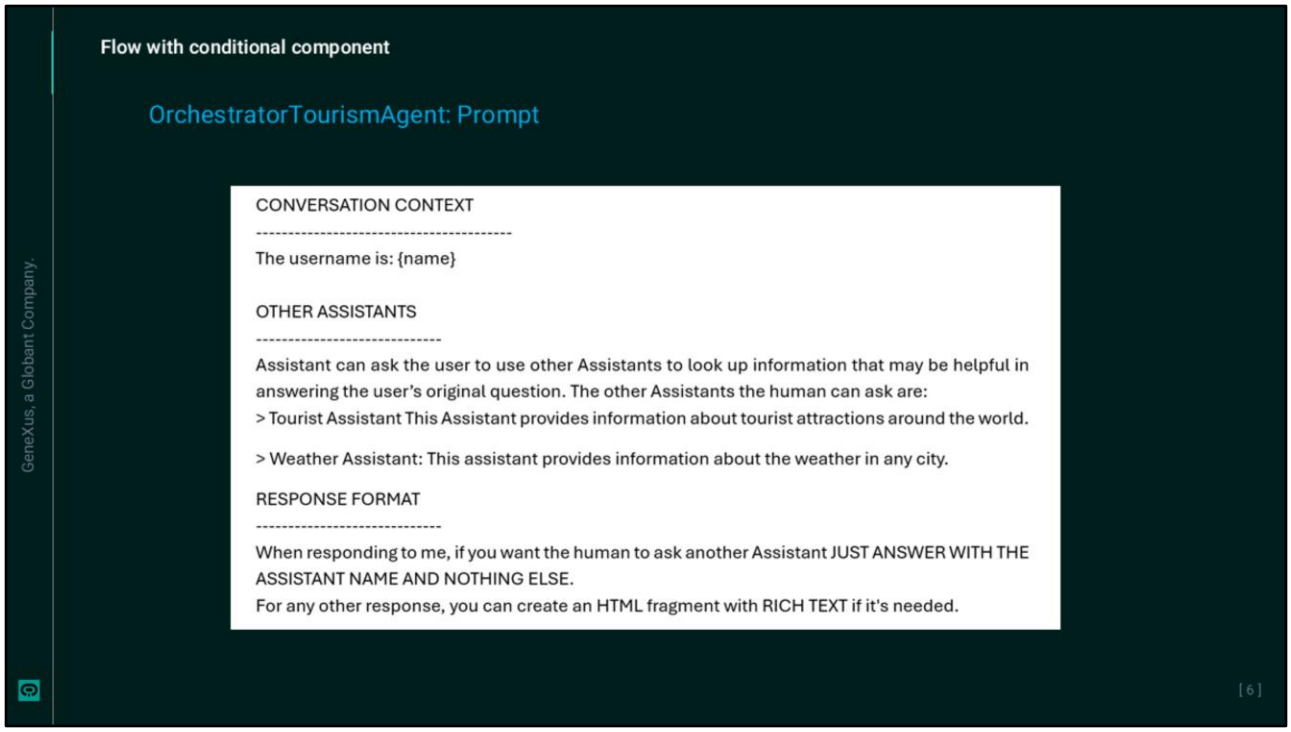
It is worth taking a moment to look at this assistant. Let's look at its prompt.



It is established that it is an LLM, capable of assisting in a wide variety of tasks, to generate human-like text based on the information it receives.

The Assistant is designed to be able to answer everything from simple questions to detailed explanations and discussions on a wide range of topics. As a language model, the Assistant can generate human-like text based on the information it receives, which allows it to engage in natural-sounding conversations and provide coherent and relevant responses to the topic at hand.

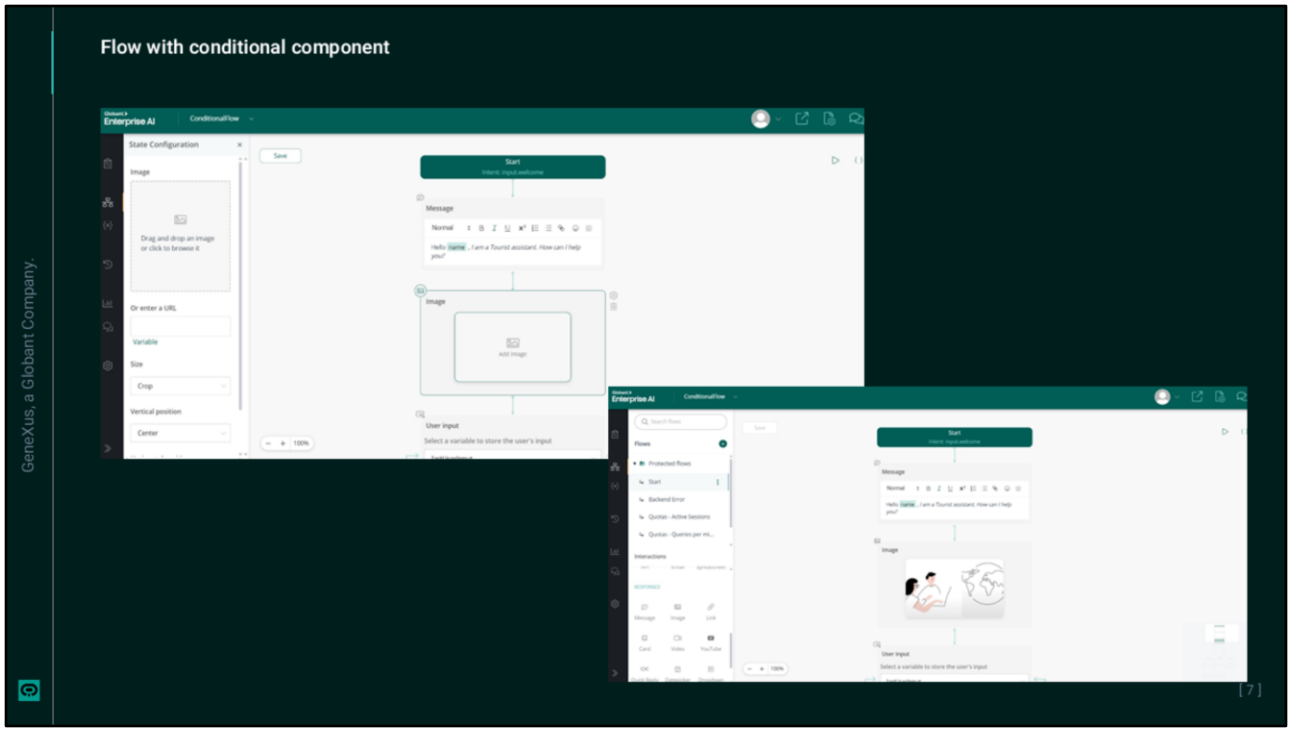
It is also indicated that it is continually learning and improving its capabilities. It is a friendly Tourist Agent that always follows the rules.



Then it is indicated that, in the context of the conversation, "name" corresponds to the user's name. In addition, other assistants to be taken into account are mentioned: Tourist Assistant that provides information on tourist attractions, and Weather Assistant that indicates the current weather in any city in the world.

Finally, the format of the response to be provided is indicated. It should return only the name of the assistant to be queried, and it is established that for any other response it can generate an HTML snippet with rich text, if necessary.

This is the behavior that defines the orchestrator assistant that is in charge of redirecting the conversation accordingly.



Well, the next step is to create the Flow. We already know the process, and create a new Flow named TouristAgent.

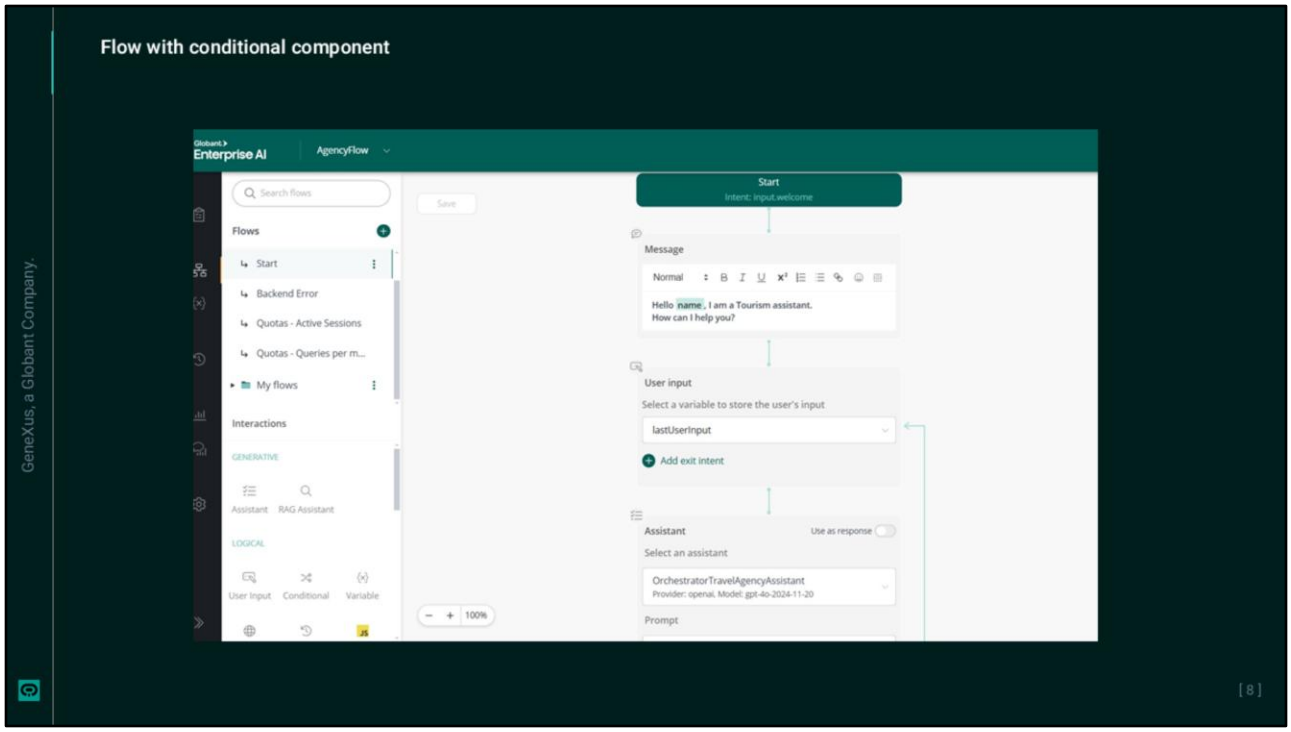
Let's customize the welcome message.

Hello name. I am a Tourist assistant. Can I help you?

But this time, in addition to the message, we want to show a representative image. So, in the interaction options, we choose Image, and drag it to its location after the message.

We select Add image and a new menu is displayed on the left. We look for the image.

It tells us that the selected image exceeds the size so we accept to resize it. We can also indicate both vertical and horizontal alignment. We leave the default options.

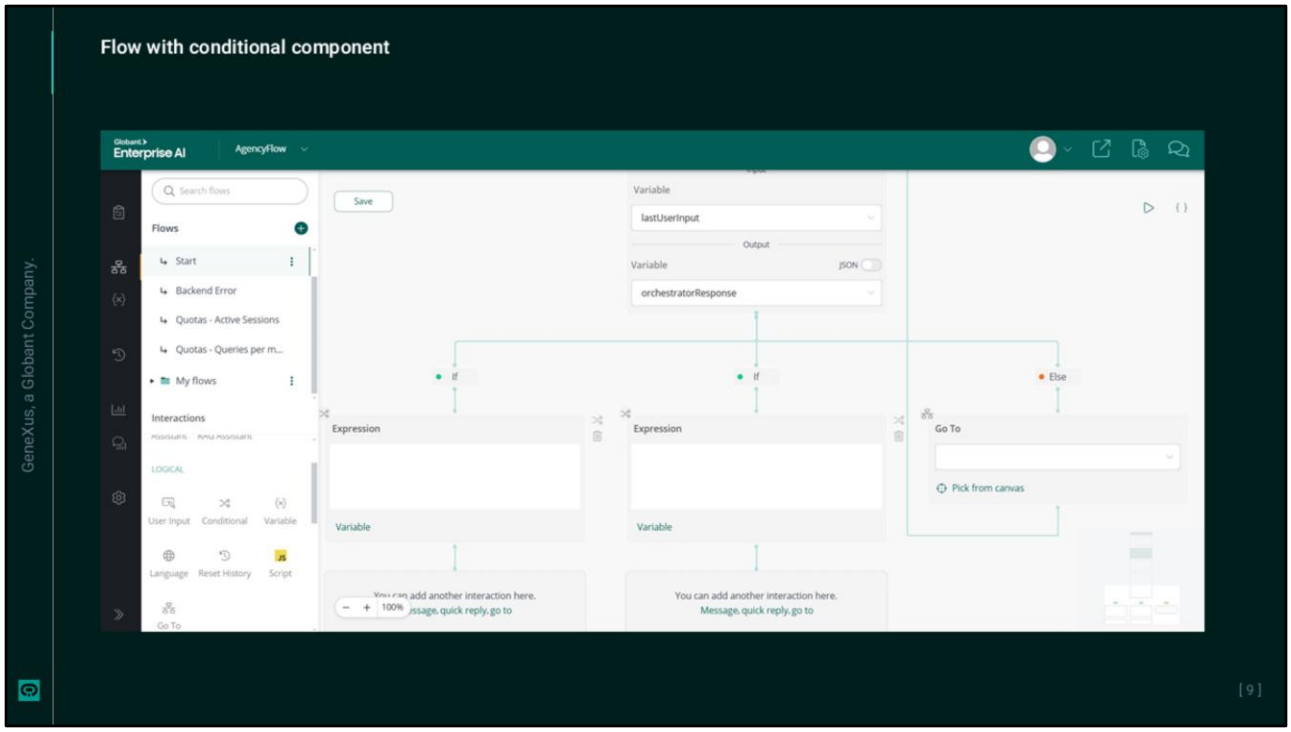


OK. Next, in the Assistant node we select the orchestrator assistant that will receive the user's query.

The input variable is lastUserInput, and the answer will be stored in a new variable. So we deactivate this option and define the output variable, which will be called orchestratorResponse.

The next step is to define the logical derivation, since the orchestrator assistant will indicate the direction depending on the user's query.

Each end-user query must be evaluated to determine which assistant is the most appropriate to answer. This is implemented with IF conditions in the flow, where the end user's intentions are compared with the available assistants.

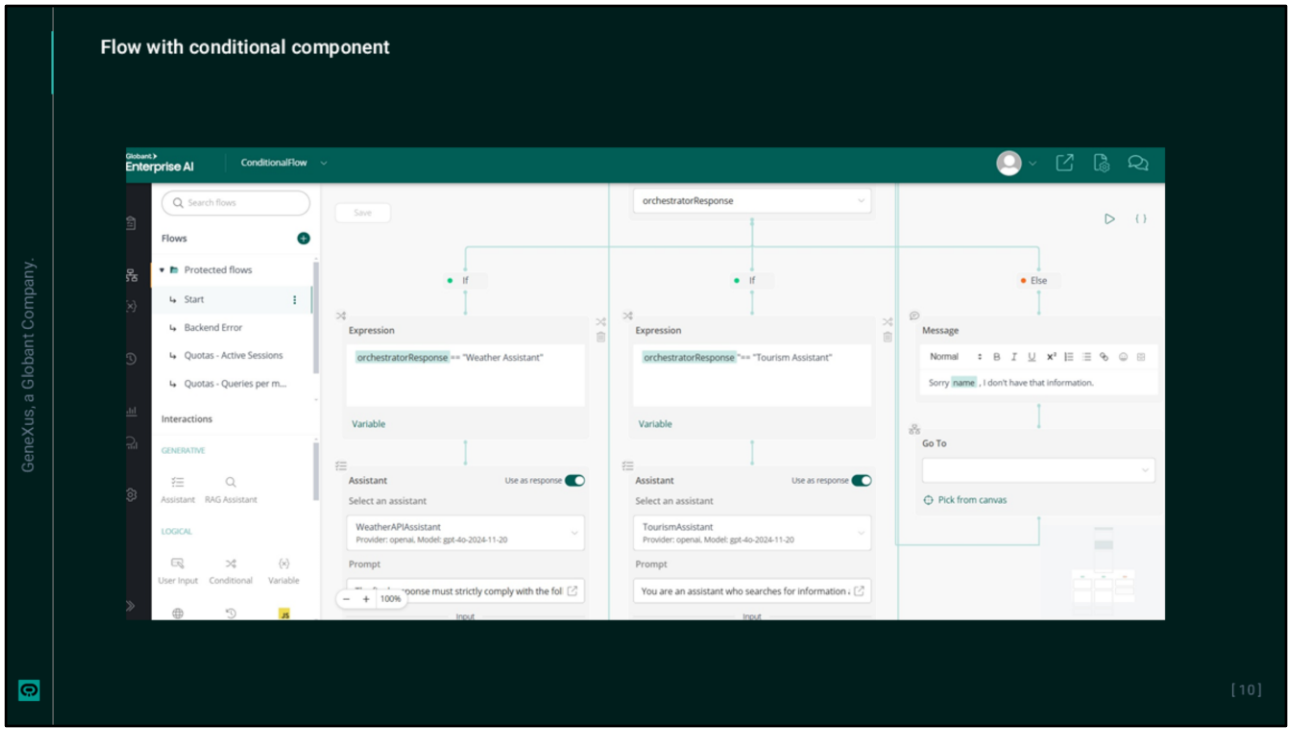


To add the Conditional component, we go to the Interactions section and drag the component between the Assistant and Go To nodes.
This action generates a branch with two paths: one labeled IF and another labeled ELSE. This structure allows you to evaluate a condition and direct the flow according to the result, whether the condition is met or not.

In our example, the orchestrator can indicate that the query is for TouristAssistant, Weather Assistant or none and then generate another response, so we need to add an additional condition.

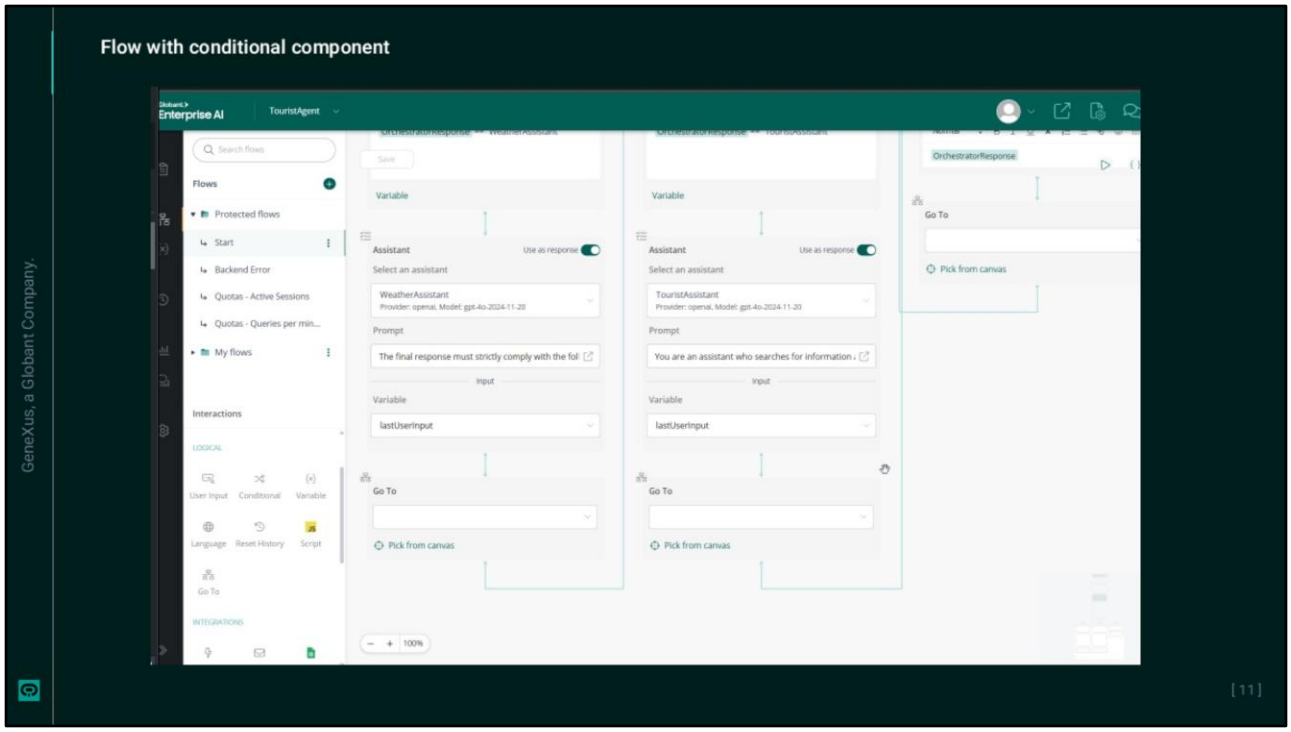
To do so, we select the Add condition option to the right of the first node of the branch: As a result, an additional node of Expression type is added, like the first one. These nodes allow the evaluation of different conditional expressions.

To add the variable to be evaluated, we select Variable at the bottom of each node.
In the first two nodes of Expression type, each condition must evaluate the OrchestratorResponse variable, and depending on the value of this variable, the flow must be directed to the Tourist Attractions Information Assistant or to the weather query in some city.



If none of these conditions are met, the flow continues through the Else branch, where by default there is a Go To node.

We add a Message node before this Go To node so that the flow can display a specific message to the end user. In our case, the name of the assistant to be consulted. This node should lead back to the User Input node.

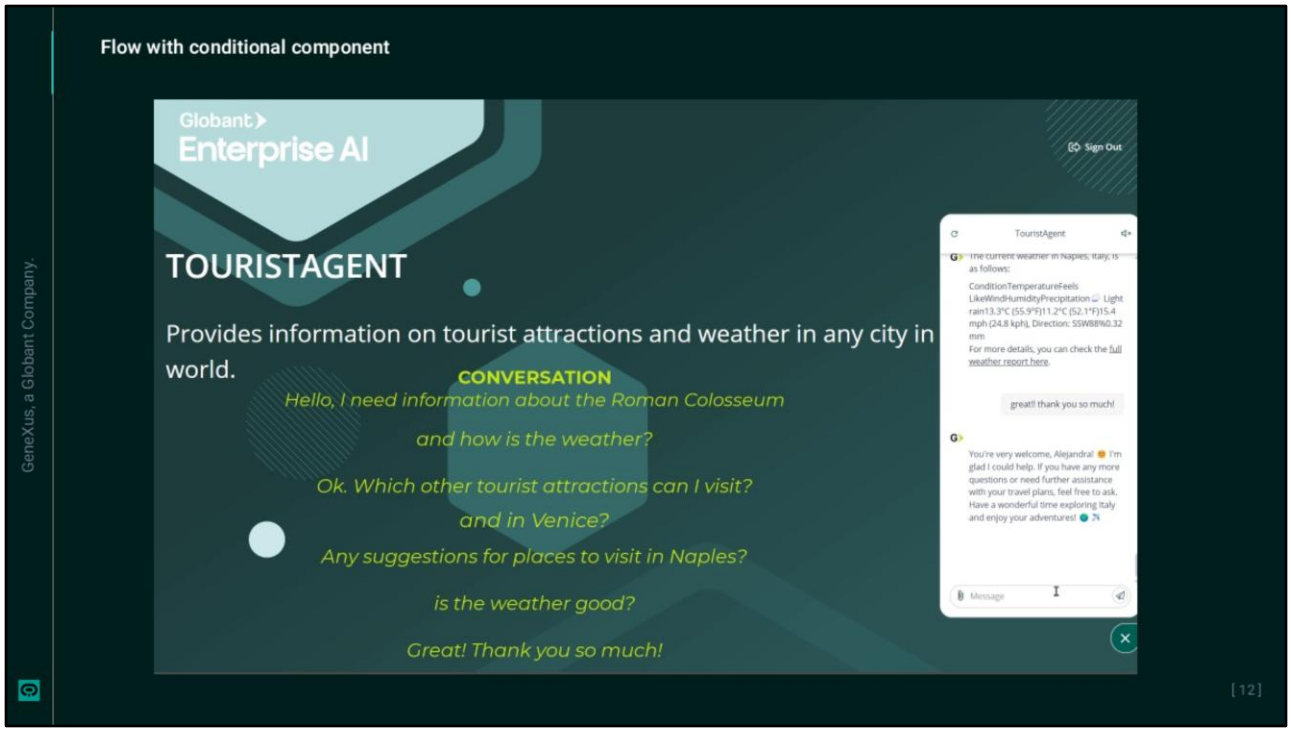


Let's now add the corresponding nodes to handle the responses based on the evaluated condition.

Under the first node of Expression type (which checks if the value of the orchestrationResponse variable is equal to "WeatherAssistant") we place a node of Assistant type, configured for WeatherAssistant that handles queries related to the weather in a city.

Under the second node of Expression type, which checks if the value of the variable is equal to "TouristAssistant", we also added an Assistant type node configured for TouristAssistant that provides information about tourist attractions.

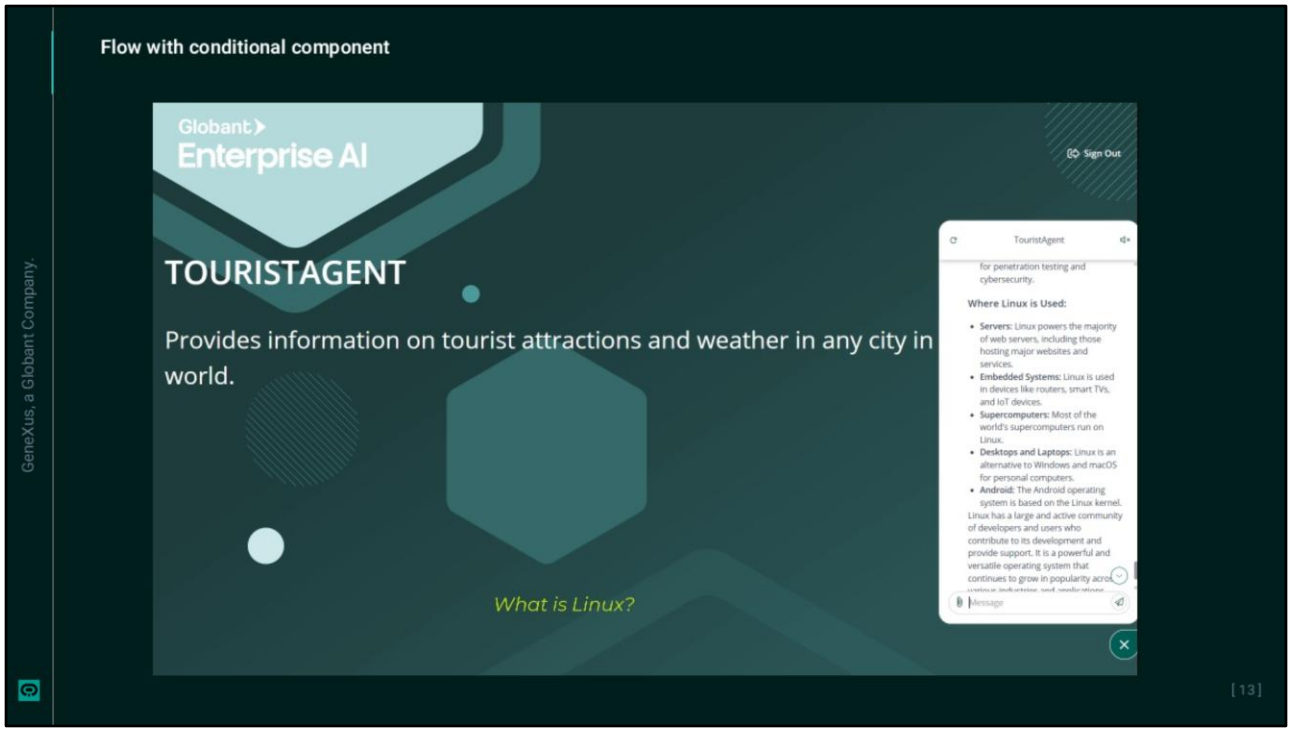
Finally, under each of these nodes we add the corresponding Go To node, which redirects back to the User Input node. Once the assistant has processed and answered the query, this allows the flow to return to the entry point to receive further interactions from the user, ensuring a continuous and fluid conversation.



Now we can test the behavior of this tourist agent.

We press "Save" and test. We know the options for testing a Flow, so we choose one of them. In this case, we choose to go to the test page.

Let's start the conversation. We greet and request information about the "Roman Colosseum".
And how's the weather?
What other tourist attractions can I visit?
And in Venice?
Any suggestions for Naples?
Is the weather good?
Perfect! We thank the assistant for its help.



Now, what will happen if we now ask a question whose answer does not correspond to any assistant? If we ask, for example, "What is Linux?“ Let's try it.

As stated, in these cases, the orchestrator must handle the response using an alternative resource and provide the information directly from the model. The flow then covers all possible end-user interactions.


Globant
Enterprise AI