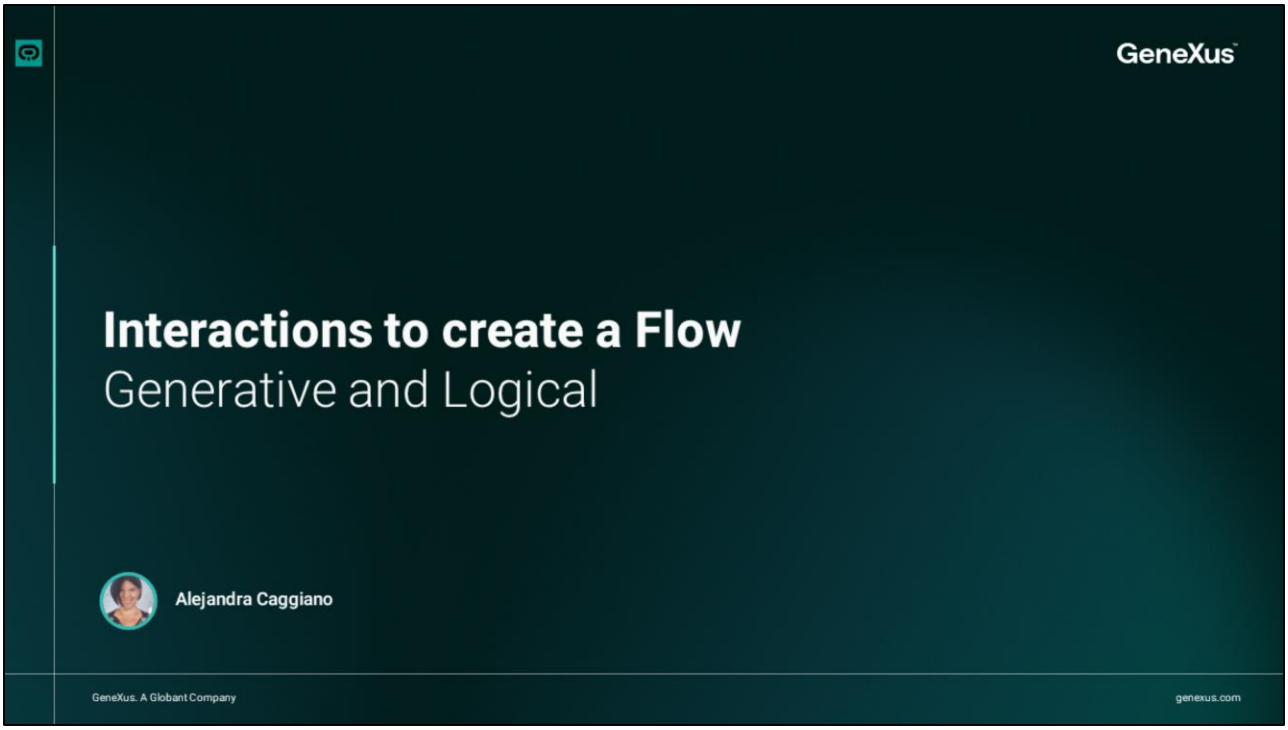
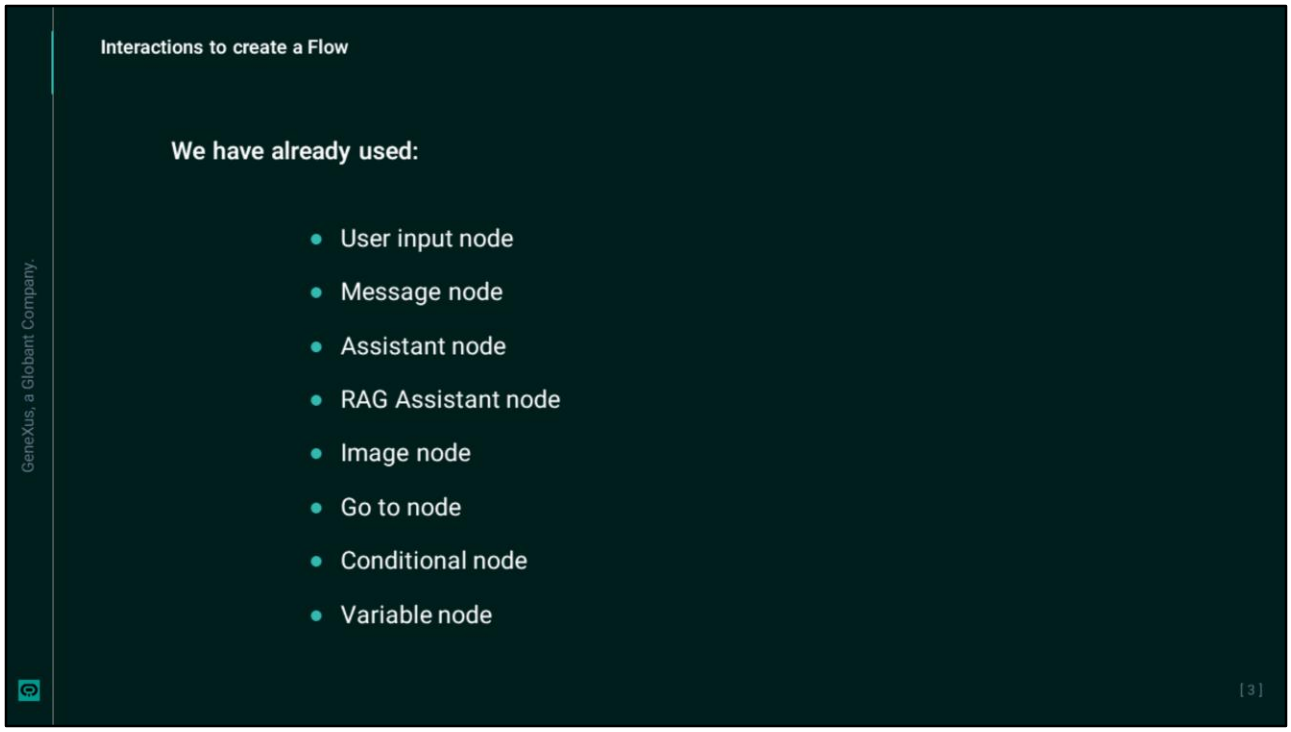
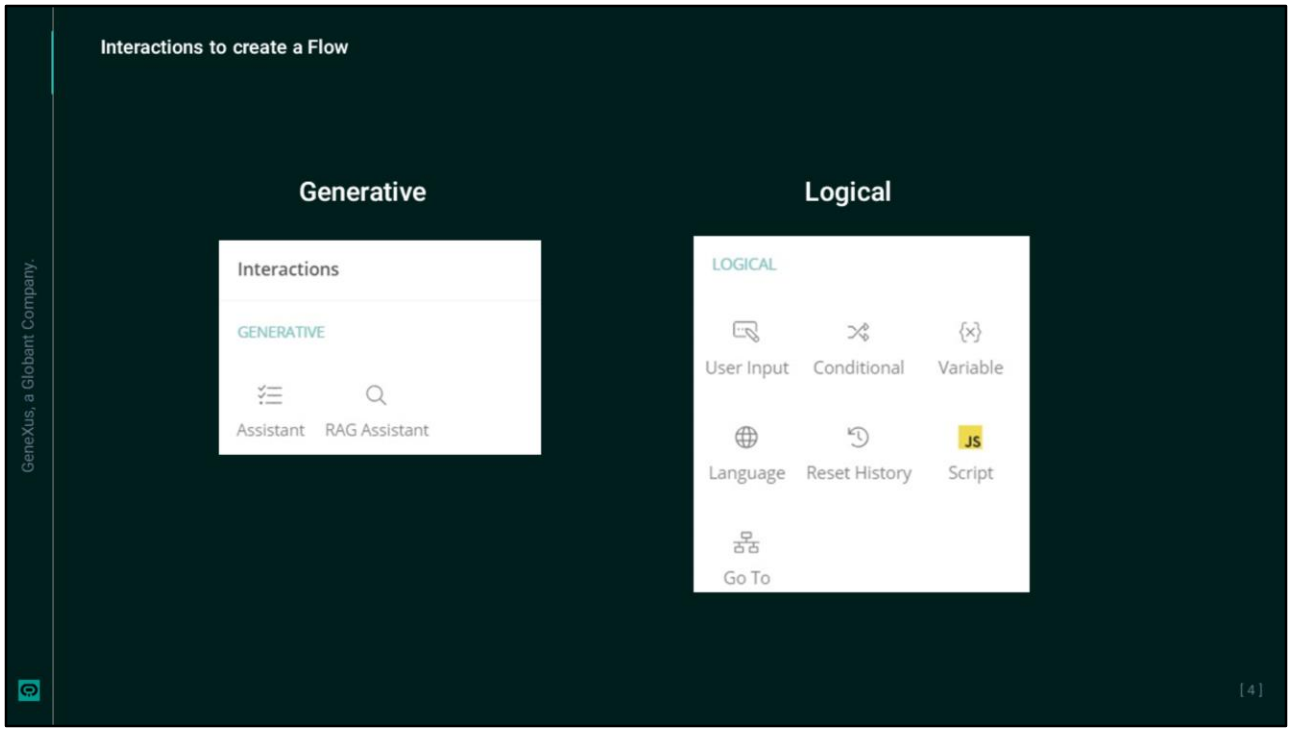

Globant
Enterprise AI



In previous videos, we used different interactions in the creation of our Flows.

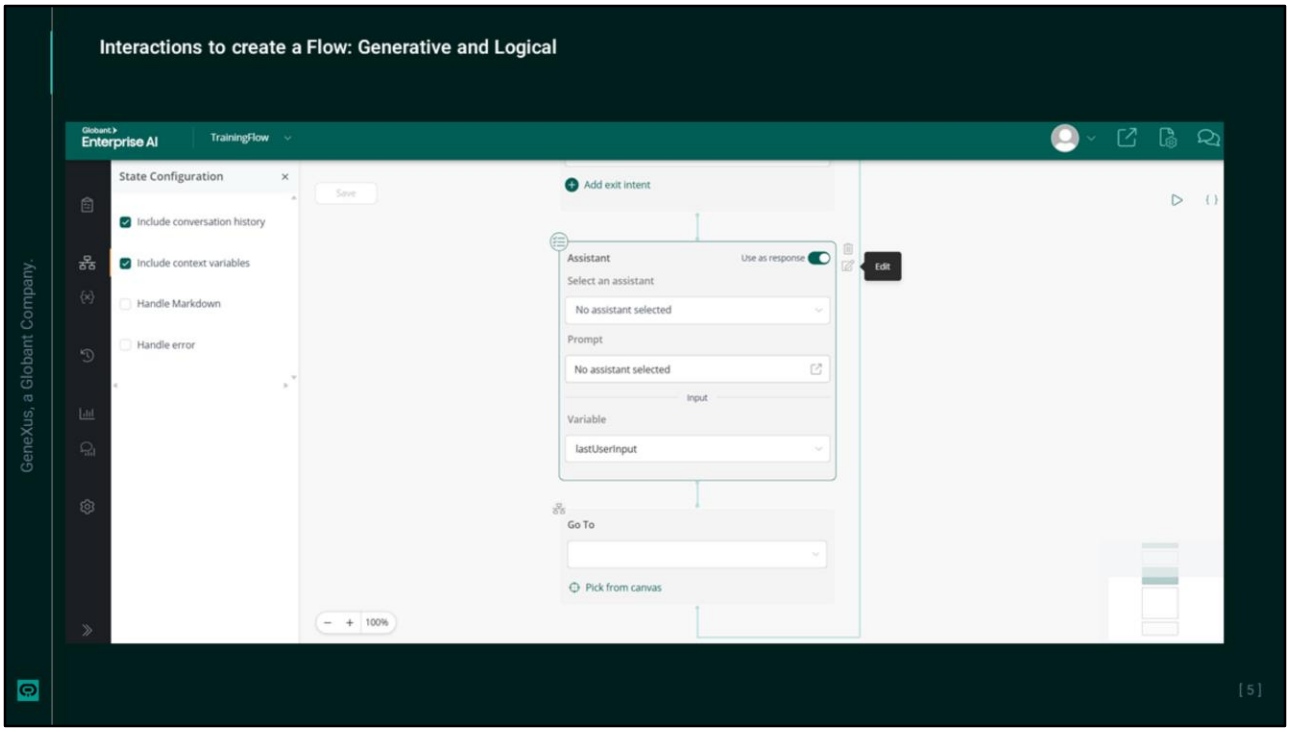


We have defined message nodes and assistants, added images, derived the flow using Go To nodes, evaluated conditions using a conditional component and used variables.



We are now going to focus on the interactions section, to learn about the components we can use to design flows that meet the needs of the end user.

Let's start with the **Generative and Logical** categories.



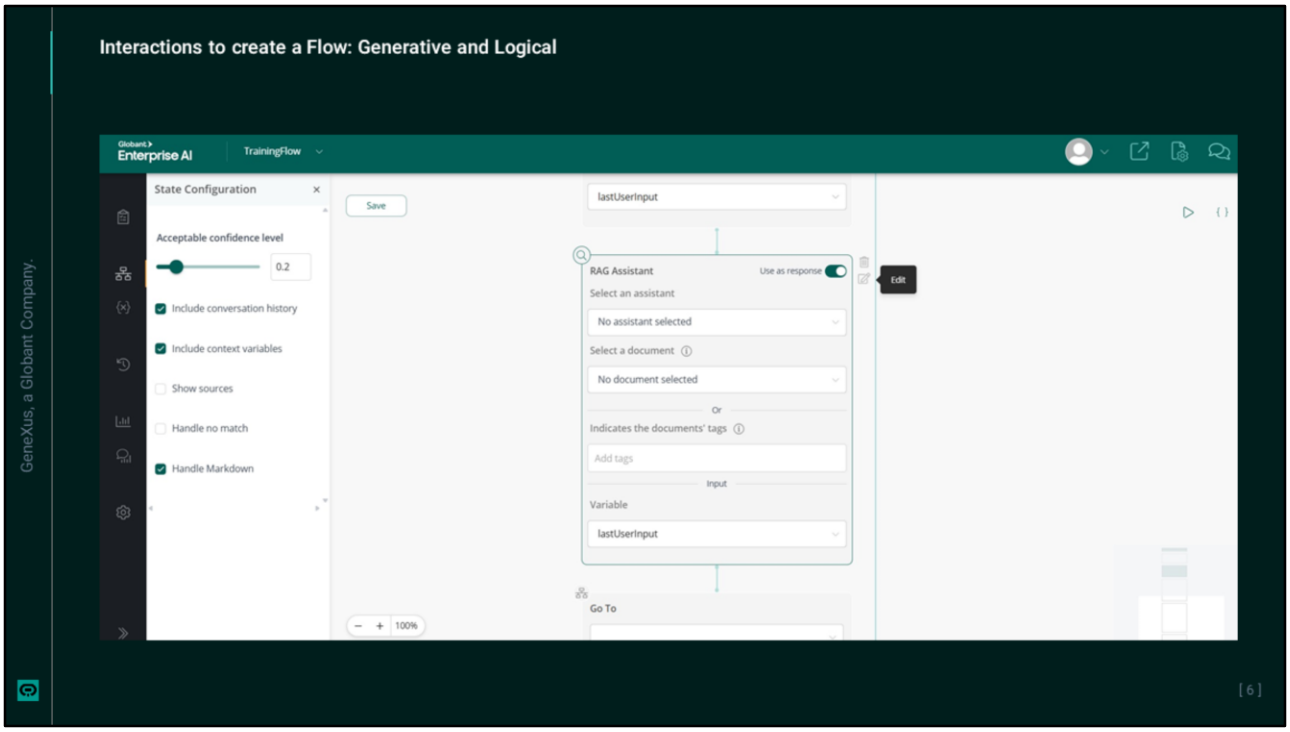
Under the **Generative** category we find the assistants that use generative artificial intelligence: Assistant and RAG Assistant.

Although we have already seen it, it is worth remembering that the Assistant node allows us to configure a Data Analyst Assistant, API Assistant, Chat Assistant, Chat with Data Assistant, which have been previously defined in the project selected in the Globant Enterprise AI backoffice.

This node always requires an input variable, such as lastUserInput, which receives the information provided by the user.

On the other hand, it is possible to configure the output variable. By default, the assistant's response is stored in lastUserInput, but if we deactivate the "Use as response" option we can set a different output variable, and optionally save the response in JSON format.

Through the Edit option at the top right edge of the node it is possible to access other configurations. We can include the conversation history and context variables to further customize the interaction, as well as define actions to be taken when a successful response is not generated.



As for the **RAG Assistant** node, it allows integrating a RAG assistant previously defined in the project selected in the Globant Enterprise AI Backoffice.

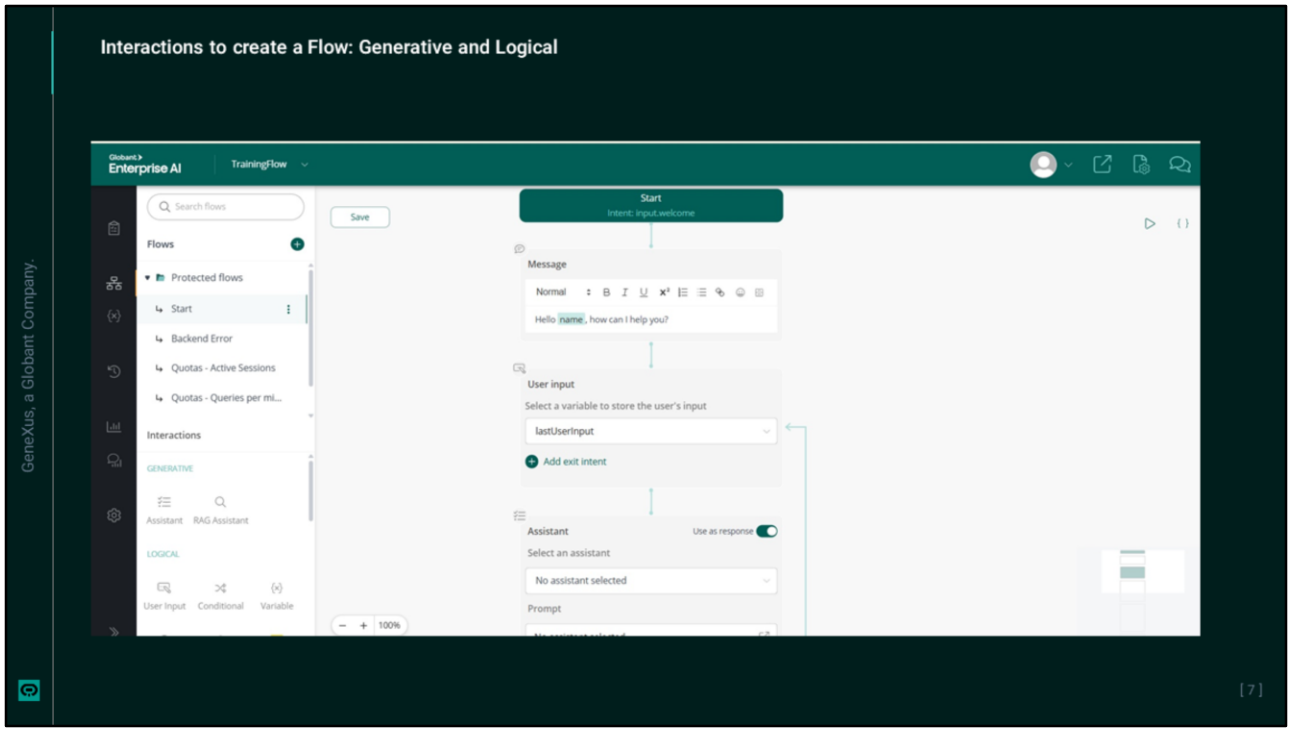
Once selected, it is possible to indicate a specific document or labels that will serve as the source of the answers; otherwise, the documents used in the assistant definition will be used.

Variables are handled in the same way as in the Assistant node. That is, an input variable is always required and it is possible to configure a different output variable, which can be in JSON format.

We can also set the minimum confidence level for responses, include the conversation history and context variables, and choose to display the sources of information used.

We can also configure the actions to be taken when an answer is not found in the provided documents.

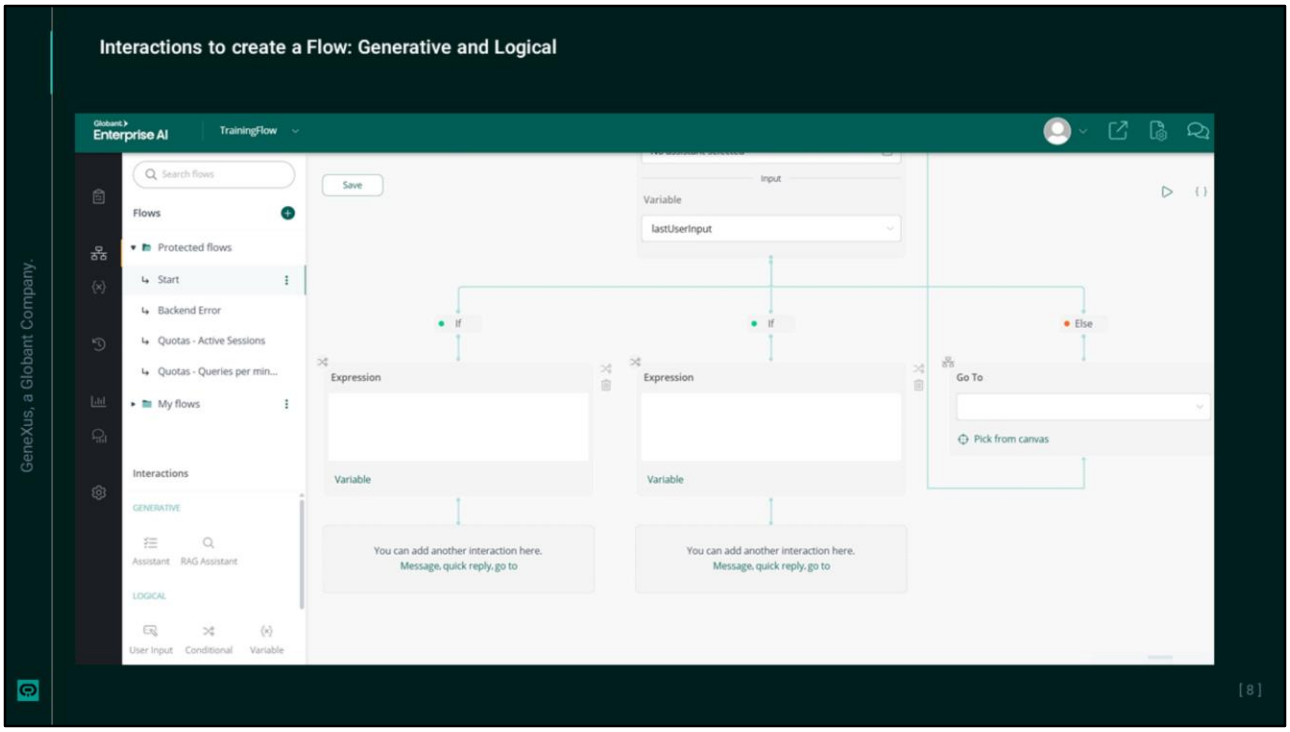
We have already used these components in previous videos.



Good. Let's now look at the options under the **Logica**l category.

This section includes logical components that allow you to control the flow of the conversation and manipulate variables or conditions within the flow.

The User input node, which we also already know, receives the user input and stores it in a variable for use in subsequent nodes.



To evaluate logical conditions and decide which path the flow should take, we use a Conditional component. When this component is added, the flow is divided into branches. By default, one is for "If" and the other for "Else".

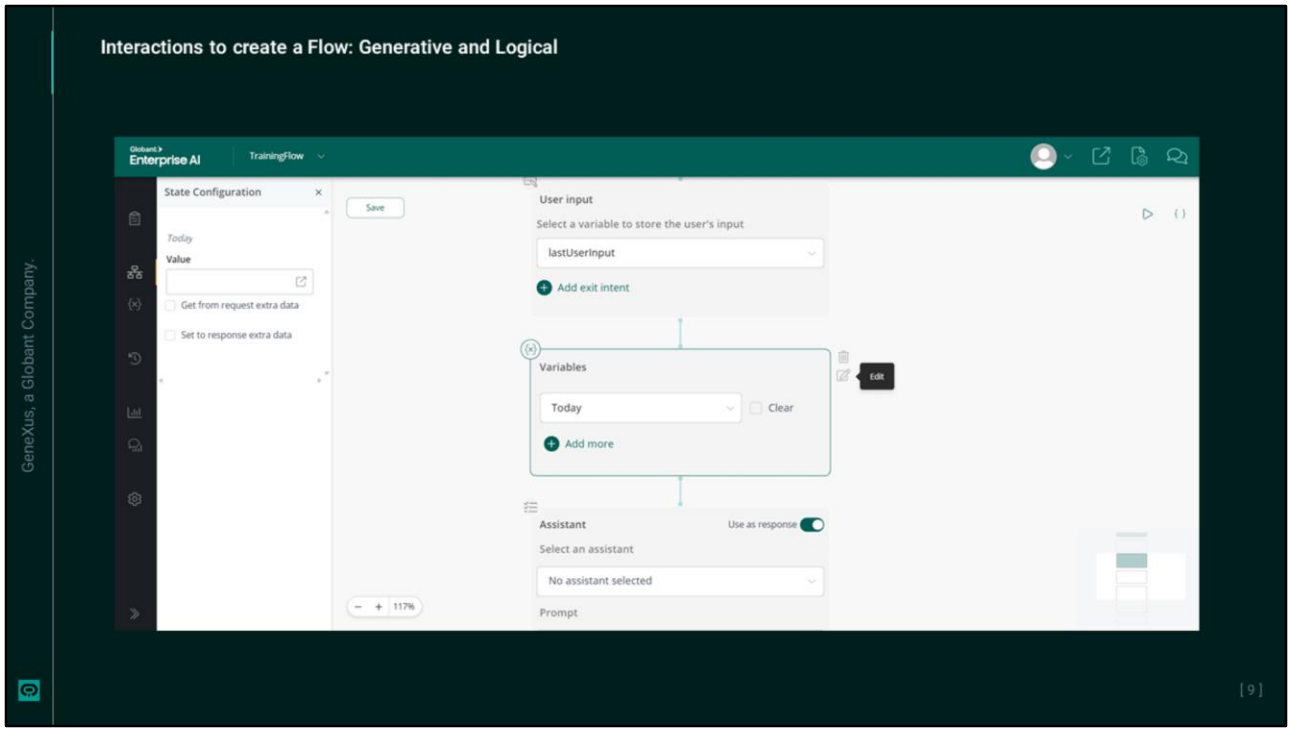
The node of the "If" branch is by default of Expression type, while the node of the "Else" branch is of Go To type. As we have already seen, it is possible to add more "If" conditions by clicking on "Add condition". Doing so adds a branch with another node of Expression type, by means of which it is possible to evaluate different conditional expressions.

Each branch is governed by the result of the evaluated condition. Depending on the value of a variable, the flow is directed to different actions or assistants. If the condition is not met, the flow continues through the "Else" branch.

To define a condition, we select a variable and apply a relational operator to perform the comparison. The relational operators supported are:

- Equal to (Example: lastResponse == Response)
- Not equal to (Example: lastResponse != Response)
- Less than (Example: lastResponse < Response)
- Less than or equal to (Example: lastResponse <= Response)
- Greater than (Example: lastResponse > Response)
- Greater than or equal to. (Example: lastResponse >= Response)

It is important to note that the variable must have a value assigned to it before the comparison can be made. This can be done in any component that allows assigning a value to a variable.

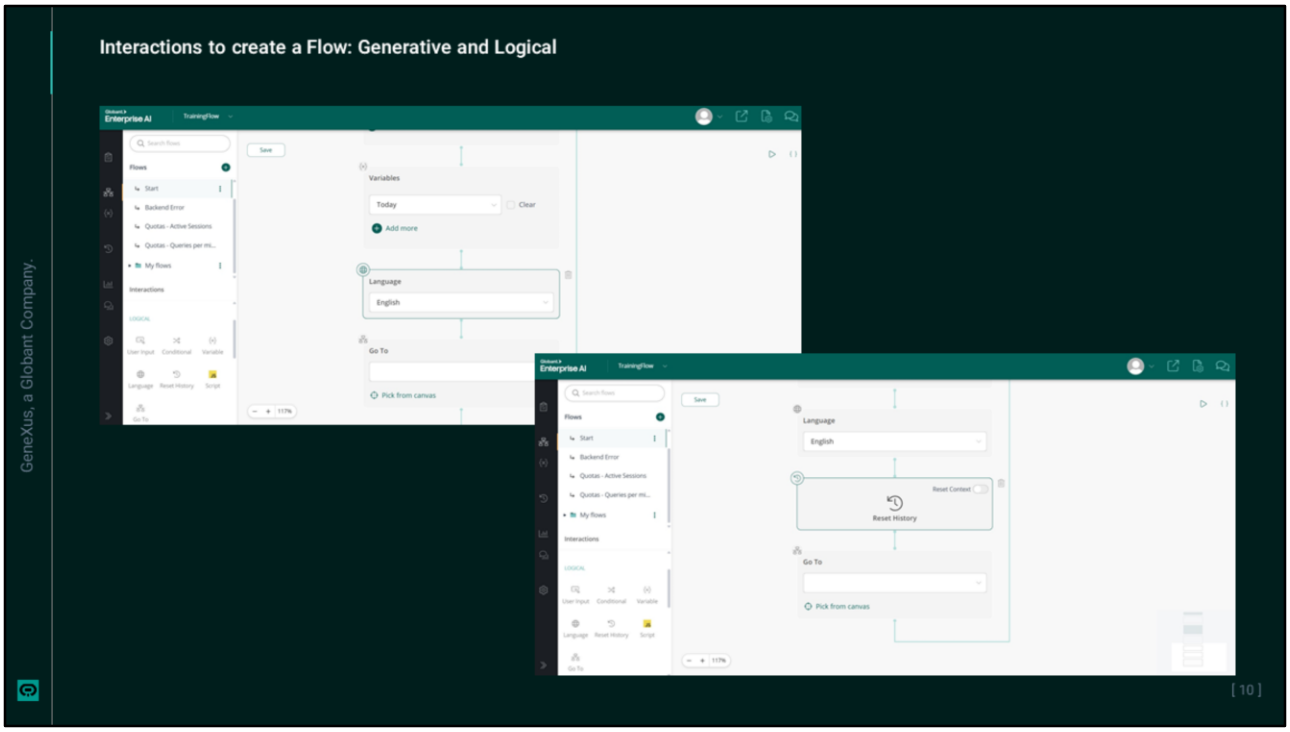


Okay, let's move on to the Variable node. This node is used to manipulate variables. This allows setting, modifying or querying values to be used in the flow.

To define a variable, we select here, and then "Add new", and indicate the name, for example, Today. To set the value of this variable, we select Edit to the right of the node, and see this options menu.

The "Get from request extra data" option assigns the value of an additional property from the request data to the variable.

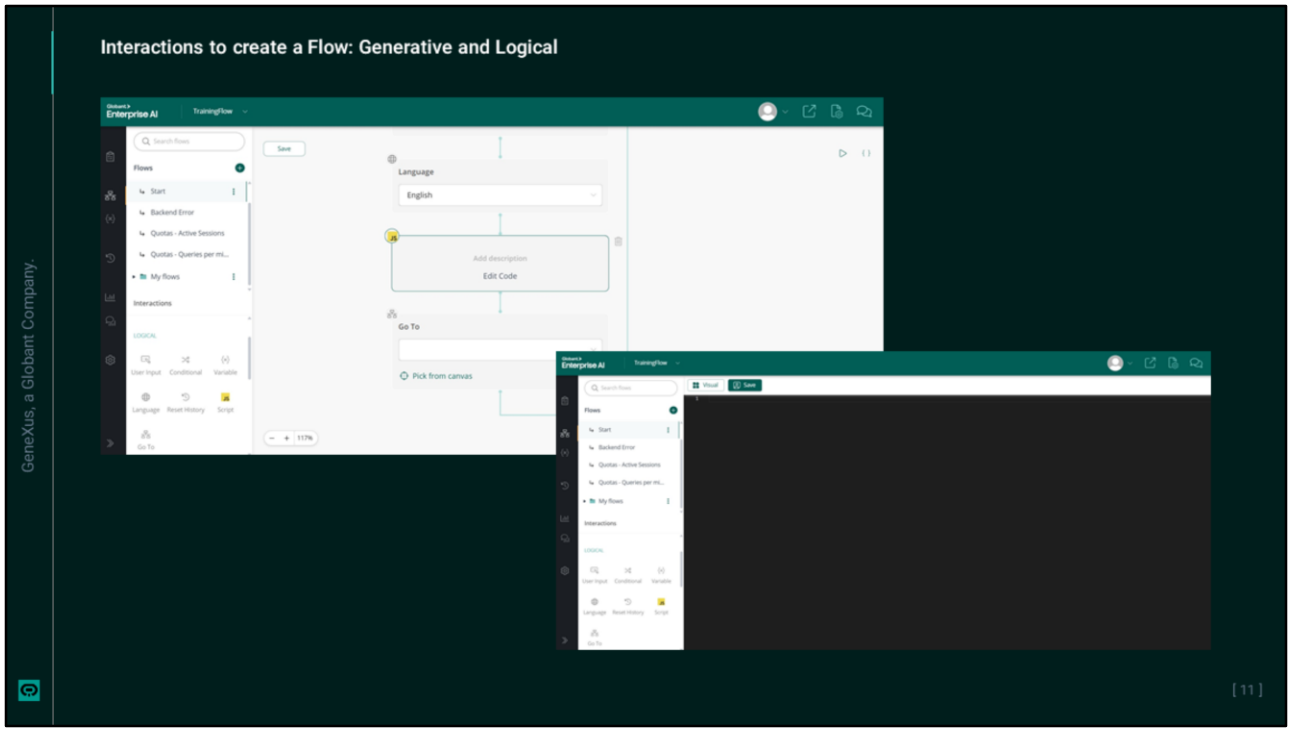
Meanwhile, the "Set to response extra data" option assigns the value of the variable to a property of the response data.
In the Value window, we can dynamically set the value of the variable. We can enter this type of expression to assign the current date to the variable. This allows data to be generated and manipulated dynamically within the flow.



Let's move on to the Language node. This option allows changing the language of the interaction according to the user's needs or the context of the flow.

Then, the Reset history option allows resetting the conversation history, which is useful for starting a new interaction without previous responses influencing the new flow.

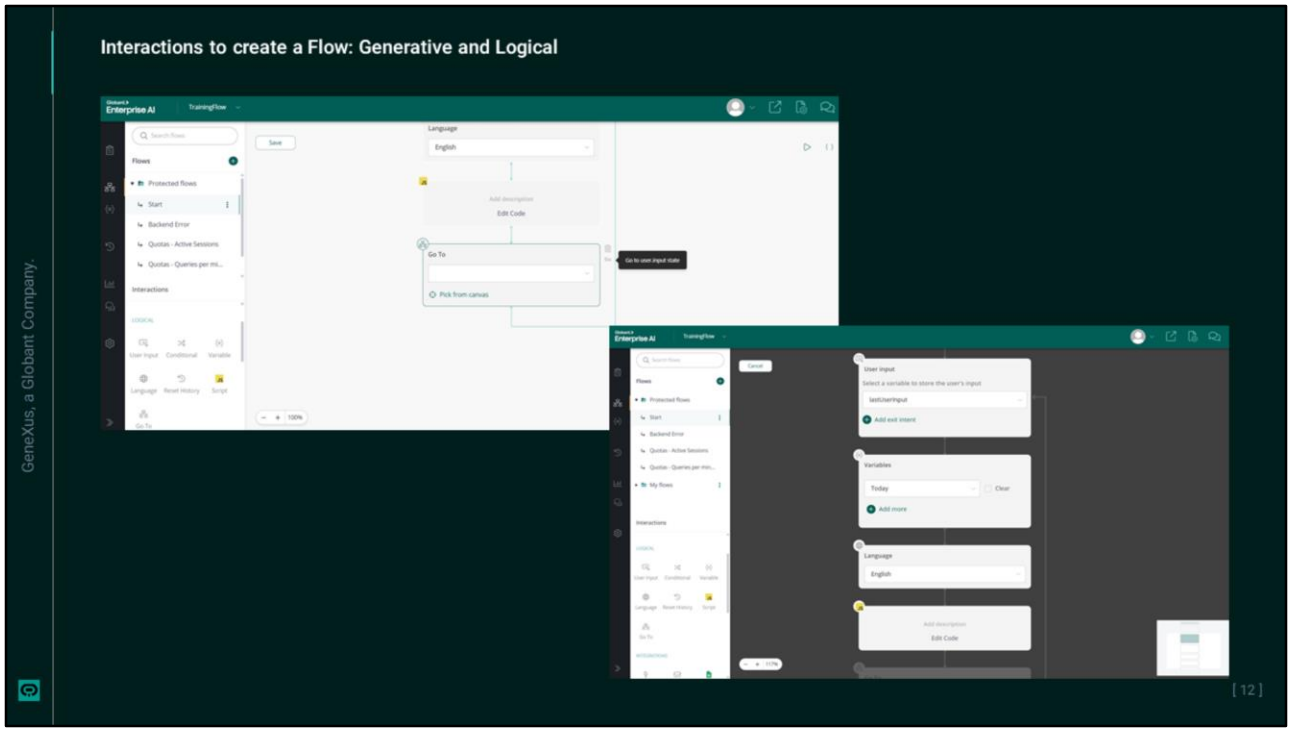
In addition, if the Reset context option, which is disabled by default, is enabled, both the conversation history and all the context variables associated with the flow will be restored. This means that the data or values previously stored in the flow variables will also be restored.



Okay, let's move on to the Script node. This option allows you to insert JavaScript code that runs on the server, providing flexibility to implement specific functionalities that are not covered by the standard system components.

When inserting this component, we see an area to add a description and an "Edit code" button, which opens a code editor with a dark workspace to write and save JavaScript code.

This component allows us to access context variables, and to define and use custom functions within the script.



Finally, the Go To node connects different flows or nodes, facilitating end-user navigation and redirection to different parts of the flow as needed.

When inserting it, we can select the flow to which we want to redirect the user by clicking on the drop-down bar, which displays all available flows.

In addition, the "Pick from canvas" option allows us to visually select a node directly from the canvas, creating a direct connection between nodes.

The "Go To" icon next to the "Go To" node allows you to navigate directly to the node to which you it is connected, making it easier to review and modify connections.

Next, we will look at the interaction options under the **Integrations** category.


Globant
Enterprise AI