

---

**ESTUDO PARA DESENVOLVIMENTO DE UM ASSISTENTE DIGITAL COM INTELIGÊNCIA ARTIFICIAL  
VISANDO POTENCIALIZAR O ATENDIMENTO AOS ALUNOS DE INSTITUIÇÕES DO ENSINO  
SUPERIOR****STUDY FOR THE DEVELOPMENT OF A DIGITAL ASSISTANT WITH ARTIFICIAL INTELLIGENCE  
INTENDING TO MAXIMISE THE ATTENDENCE TO STUDENTS OF HIGHER EDUCATION**

Diego Rodrigues Lima da Silva<sup>1</sup>, Gabriel Nascimento Lunardelli<sup>2</sup>, Guilherme Oliveira Shiroma<sup>3</sup>, Willian Levi Sharo de Oliveira<sup>4</sup>, Luciano Nascimento Corsino<sup>5</sup>

**RESUMO:** Estudar a possibilidade de utilização de um assistente digital inteligente para que haja uma melhora no atendimento aos alunos das instituições de ensino superior. Optou-se pelo método da pesquisa bibliográfica, com a finalidade de adquirir conhecimento e referências. O estudo incluiu a utilização de tecnologias como: inteligência artificial, *machine learning* e processamento de linguagem natural. Para o desenvolvimento do assistente digital, decidiu-se pelo o uso da plataforma IBM Cloud e da API IBM Watson Assistant. Espera-se que haja uma melhora significativa na qualidade do atendimento aos alunos da instituição, redução de chamados padrão, além de redução de custos para a instituição acadêmica. Para se chegar ao resultado esperado será necessário um levantamento de dados inicial através de uma persona, desenvolvimento de um fluxo de conversa estruturado, realização de testes durante e depois do desenvolvimento e a monitoração constante das conversas feitas pelo assistente digital. Estudos mostram que o assistente digital poderá ser capaz de realizar o primeiro nível de atendimento do *call center* da instituição de ensino superior e, principalmente, oferecer um atendimento de qualidade aos alunos, porém para se chegar a este nível será necessário um estudo estruturado para que o desenvolvimento do assistente seja de qualidade.

**PALAVRAS-CHAVE:** Atendimento ao Público. Instituição de Ensino Superior. Assistente Digital Inteligente. Inteligência Artificial. IBM Watson Assistant. Chatbot.

**ABSTRACT:** *It highlight the study for the development of an intelligent digital assistant so that there is an improvement in the attendance of higher education. It was decided for the method of bibliographic research, with the purpose of acquiring knowledge and references. The study included the use of technologies such as artificial intelligence, machine learning and natural language processing. For the development of the digital assistant, it was decided to use the IBM Cloud platform and the IBM Watson Assistant API. It is expected that there will be a significant improvement in the quality of attendance to the students of the institution, reduction of standard tickets, besides cost reduction for the academic institution. To achieve the expected result, an initial data collection through a persona, development of a structured conversation flow, testing during and after development, and constant monitoring of the conversations made by the digital assistant will be necessary. Studies show that the digital assistant may be able to perform the first level of call center of the higher education and, mainly, to offer a quality service to the students, but to reach this level will require a structured study so that the development of the assistant is of quality.*

**KEYWORDS:** *Public Attendance. Higher Education. Digital Assistant. Artificial Intelligence. IBM Watson Assistant. Chatbot.*

---

<sup>1</sup> Graduando em Ciência da Computação. Universidade UNIVERITAS/UNG. diego.rls@hotmail.com

<sup>2</sup> Graduando em Ciência da Computação. Universidade UNIVERITAS/UNG. gabriellunardelli@hotmail.com

<sup>3</sup> Graduando em Ciência da Computação. Universidade UNIVERITAS/UNG. guilhermeshiroma@hotmail.com

<sup>4</sup> Graduando em Ciência da Computação. Universidade UNIVERITAS/UNG. willian\_levi@hotmail.com

<sup>5</sup> Doutorando em Educação. Unicamp – Universidade Estadual de Campinas. luciano.corsino@hotmail.com

ESTUDO PARA DESENVOLVIMENTO DE UM ASSISTENTE DIGITAL COM INTELIGÊNCIA ARTIFICIAL VISANDO POTENCIALIZAR O ATENDIMENTO AOS ALUNOS DE INSTITUIÇÕES DO ENSINO SUPERIOR  
Diego Rodrigues Lima da Silva, Gabriel Nascimento Lunardelli, Guilherme Oliveira Shiroma, Willian Levi Sharo de Oliveira, Luciano Nascimento Corsino<sup>5</sup>

## INTRODUÇÃO

O atendimento ao aluno é um enorme desafio para muitas instituições acadêmicas, isso porque o volume de informações que um atendente precisa dominar para atender todas as solicitações do público é imensa e os gastos para ter trabalhadores com capacitação dimensionado à demanda de todos os alunos é alto. Já para os alunos, o impacto é elevado, pois ocasionam longas filas de atendimento e resultados insatisfatórios.

Com os consumidores cada dia mais exigentes, a utilização da inovação tecnológica tem sido a salvação para as instituições que buscam elevar seus resultados e ofertar uma experiência diferenciada aos seus alunos. Por outro lado, nenhum dos problemas citados é incomum aos assistentes digitais. Eles são capazes de responder qualquer questão de forma imediata, 24 horas por dia. Por essa razão, há uma ampliação da quantidade de assistentes digitais em diversos lugares. Também são denominados por outros nomes, como assistentes virtuais, agentes virtuais ou simplesmente *chatbots*. Até 2020, 85% das interações dos consumidores será conduzida por um procedimento automático (GARTNER SUMMITS INC, 2011) e será a primeira escolha para distinguir uma empresa de seus competidores (WALKER INFORMATION INC, 2016).

O objetivo desse estudo é indicar a possibilidade de melhorar o atendimento dos alunos nas instituições de ensino superior. Desenvolvendo um assistente digital com inteligência artificial para automatizar o primeiro nível de atendimento do *call center*, evidenciando os prós e contras. Espera-se ganhar em escala de atendimento e os atendentes humanos poderão ser direcionados para outras funções de maior complexidade, resultando na redução de custos.

## MATERIAIS E MÉTODOS

### IBM CLOUD

Segundo descrito no site oficial, a IBM Cloud é a plataforma na nuvem da empresa IBM, a plataforma traz ferramentas que facilitam o desenvolvimento, enquadrando fatores que possibilitam a criação de serviços do nível simples ao avançado. Disponibiliza serviços para criar, executar, implementar e gerenciar aplicativos na nuvem. A Figura 1 mostra a tela inicial da IBM Cloud.

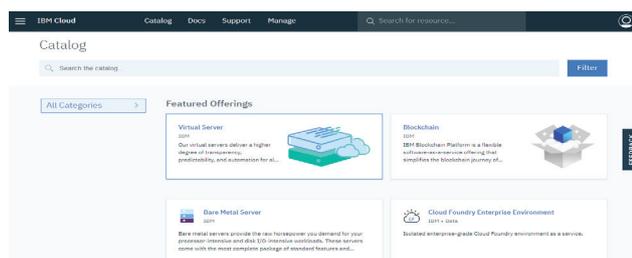


Figura 1 - IBM Cloud

### WATSON ASSISTANT

Conforme descrito pela IBM, o serviço Watson Assistant é uma API direcionada para o desenvolvimento de assistentes digitais, combinando aprendizado de máquina, compreensão de linguagem natural e ferramentas de diálogo integradas para criar fluxos de conversa. Possui uma interface simples e intuitiva, possibilitando ao usuário criar seu serviço independentemente de seu conhecimento técnico. A escolha de utilização deste serviço se deve pela facilidade e simplicidade de se criar e gerenciar o assistente digital. A Figura 2 exibe a tela inicial do Watson Assistant.

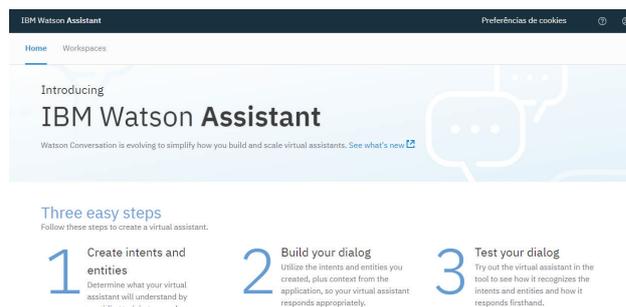


Figura 2 - Watson Assistant

### INTELIGÊNCIA ARTIFICIAL

Inteligência Artificial (IA) pode ser explicada como a área da computação que estuda formas de se desenvolver sistemas ou dispositivos que simulem a capacidade humana de raciocínio, tomada de decisões e solução de problemas (RICH, 1988), inclui em suas características fatores que ajudam a compreender melhor a proposta para alcançar um resultado satisfatório (ZUBEN, 2011). Para melhor entendimento da proposta, segue um exemplo do comportamento da IA baseadas em etapas:

1. Verificar uma consistência lógica ou se há contradição de ideias.
2. Analisar a existência de incoerência das ideias e o que há no histórico das informações apresentadas.
3. Formular de maneira ampla o conhecimento, analisando a proposta e se a mesma satisfaz, adequando-se a situação (HILL; FORD, 2015).

### MACHINE LEARNING

O *machine learning* pode ser definido como um modelo de análise de dados automatizado que contribui nos modelos analíticos, está ligado diretamente a inteligência artificial defendendo a tese que os sistemas são capazes de aprender com dados e as informações que a ele são passados (GOLDBERG; HOLLAND, 1988), consegue detectar tipos de padrões sendo capaz de tomar decisões com o mínimo possível de intervenção humana ou até mesmo sem nenhuma intervenção. (BATISTA, 2003). O aprendizado de máquina supervisionado, no qual se funda-

menta o método utilizado, se determina por três condições de termos na qual se deve seguir ao objetivo, sendo eles:

1. Um indutor, que classificará o termo como entrada separando em vetores, palavras e muitas vezes utilizando o banco de dados para isso.
2. Um exemplo (denominado como registro ou dado), são princípios de conjunto de atributos que armazena informações para uma futura execução da programação e seus códigos predefinidos.
3. E os atributos, que são separados por seus tipos, ordem, relevância e regras, que são essenciais para que a aplicação obedeça sua programação de maneira correta, quanto mais dados armazenados em atributos, mais complexa, estruturada e completa será a aplicação (MONARD; BARANAUSKAS, 2003).

### PROCESSAMENTO DE LINGUAGEM NATURAL

Processamento de linguagem natural (PLN) é uma área de pesquisa e aplicação que explora como computadores podem ser usados para entender e manipular linguagem natural no formato de texto ou som para que consigam produzir resultados úteis (LOPES; VIEIRA, 2010). São modelos computacionais desenvolvidos para realizarem funções onde é necessário o uso de informações expressas da linguagem natural. (PEREIRA, 2011). O assistente digital construído com PLN, deverá analisar e interpretar a mensagem inserida pelo usuário e a partir disso entregará a resposta mais coerente, assim como em uma conversa humana.

### JSON

Na computação, JSON é um acrônimo para *JavaScript Object Notation*, definido como uma linguagem de programação baseada em texto para transferência de dados, amplamente utilizada para transferir objetos em *JavaScript* entre computadores através do modelo de *string*. O JSON possuiu características a serem destacadas como: simplicidade para leitura, método ágil para análise sintática, facilidade para se gerar objetos e ainda é possível transmitir e manipular dados através da internet de forma eficiente (FONSECA; SIMÕES, 2007). No exemplo da Figura 3, é demonstrado como uma resposta simples do assistente digital deverá ser programada no JSON.

```
{
  "output": {
    "generic": [
      {
        "values": [
          {
            "text": "Está é a resposta em texto."
          }
        ],
        "response_type": "text",
        "selection_policy": "sequential"
      }
    ]
  }
}
```

Figura 3 - Exemplo de resposta em JSON

### APPLICATION PROGRAMMING INTERFACE

Na programação de computadores, uma interface de programação de aplicativos (API) em termos gerais, é um conjunto de métodos de comunicação claramente definidos entre vários componentes (BERGLER, 1996). Uma API facilita o desenvolvimento de um programa de computador, fornecendo todos os blocos de construção, que são então montados pelo programador.

### 2.1 TÉCNICAS DE IMPLEMENTAÇÃO

#### INTENÇÕES

São propósitos ou objetivos expressos na entrada do usuário, deverá ser responsável por identificar como o assistente irá responder a uma pergunta. Como é possível digitar a mesma informação de formas variadas, desenvolver um assistente que trate cada uma dessas entradas será excessivamente trabalhoso (SHEVAT, 2017), para isto existe as intenções, nas quais serão fornecidos exemplos de frases e posteriormente o sistema deverá identificar outros exemplos e relacionar com as existentes, usando o PLN. Na Figura 4 é representado o desenho de uma intenção.

Onde fica a biblioteca?  
 Cadê o prédio da secretária?  
 Onde eu acho a sala da coordenação?  
 Me mostre onde fica a lanchonete?  
 Você sabe me dizer em que lugar é o laboratório?

Intenção = buscar localização

Figura 4 - Representação de uma intenção

#### ENTIDADES

São complementos da informação, o assistente digital deverá identificar estes itens na entrada do usuário e extrai-los, podendo assim, realizar uma abordagem diferenciada para aquela pergunta. As entidades que possuírem valores parecidos deverão ser agrupadas e os sinônimos deverão ser relacionados a essas palavras, assim poderá ser possível reconhecer uma entidade que possuí valores iguais.

## FUZZY MATCHING

É um método que fornece uma capacidade aprimorada de processar consultas baseadas em palavras para localizar frases ou sentenças correspondentes (WEBER; KLEIN, 2003).

- **Stemming** – deverá reconhecer o formulário raiz de valores de entidades que possuem várias formas gramaticais.
- **Erro de ortografia** - mapeará a entrada do usuário para a entidade correspondente apropriada, apesar da presença de erros ortográficos ou pequenas diferenças sintáticas.
- **Correspondência parcial** – deverá sugerir sinônimos baseados em *substrings* presentes nas entidades definidas e atribuir uma pontuação de confiança mais baixa em comparação à correspondência exata da entidade.

## PADRÕES

São específicos para um valor de entidade, deverá ser inserido como uma expressão regular no campo. Como no exemplo abaixo, os padrões para valores de telefone, e-mail e matrícula poderão ser definidos da seguinte maneira:

- celular:  $(\d{2}) (\d{5})-(\d{4})$ . Exemplo: 11 96587-6551
- e-mail:  $(\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b)$ . Exemplo: nome@dominio.com
- matrícula:  $(\d{8})$ . Exemplo: 24285225

Para a sua utilização será necessário armazenar o texto que corresponde ao padrão em uma variável de contexto, dentro da árvore de diálogo.

## CONSTRUINDO DIÁLOGO

A caixa de diálogo usará as intenções identificadas na entrada do usuário, além do contexto, para interagir com o mesmo e, por fim, fornecer uma resposta útil.

A resposta pode ser para uma pergunta como: "Posso mudar de turno?" ou a execução de um comando, "Abra um chamado". A intenção e a entidade podem ser informações suficientes para identificar a resposta correta ou a caixa de diálogo poderá solicitar ao aluno informações necessárias para responder corretamente. Por exemplo, se um usuário perguntar: "Como corrigir o boleto?" o assistente poderá querer esclarecer se o boleto não está disponível ou se está com valor errado, se o aluno deseja abrir chamado ou falar com atendente, e assim por diante.

## NÓS DE DIALOGO

Cada nó de diálogo deverá conter, no mínimo, uma condição e uma resposta, conforme a Figura 5.



Figura 5 - Nó de diálogo

**Condição:** Especificará as informações que deverão estar presentes na entrada do usuário para que esse nó no diálogo seja acionado. As informações poderão ser uma intenção, um valor de entidade ou um valor da variável de contexto.

**Resposta:** A elocução que o serviço usará para responder ao usuário. A resposta também poderá ser configurada para acionar ações programáticas.

Pode-se pensar no nó como tendo uma construção se/então: se essa condição for verdadeira, retorne essa resposta. Por exemplo, o nó representado na Figura 6 a seguir será acionado se a função de processamento de linguagem natural do serviço determinar que a entrada do usuário contém a intenção de inclusão de disciplina. Como resultado do nó a ser acionado, o serviço responderá com uma resposta apropriada.

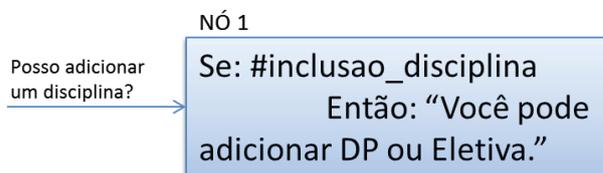


Figura 6 - Reconhecendo uma intenção

Um único nó com uma condição e resposta poderá manipular solicitações simples dos alunos. Mas, na maioria das vezes, os usuários terão perguntas mais sofisticadas ou irão querer ajuda com tarefas mais complexas. Neste caso poderá ser adicionado nós filhos que irão solicitar ao usuário informações adicionais, conforme Figura 7.

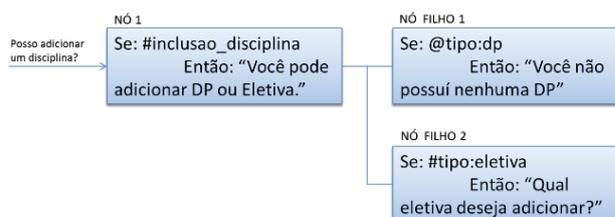


Figura 7 - Nó solicitando informações

## FLUXO DE DIÁLOGO

A caixa de diálogo criada deverá ser processada pelo serviço do primeiro nó da árvore até o último, esse fluxo está representado na Figura 8:

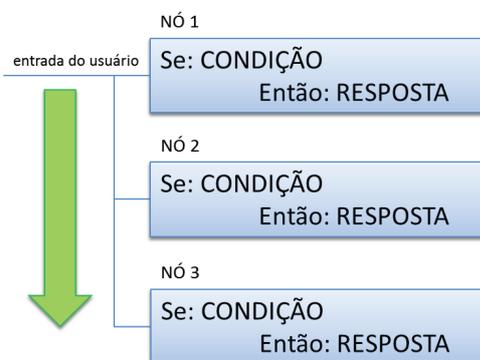


Figura 8 - Fluxo do diálogo

Conforme percorrerá a árvore, se o serviço encontrar uma condição que é atendida, ele disparará esse nó. Em seguida, ele se moverá ao longo do nó acionado para verificar a entrada do usuário em relação a qualquer condição do nó filho.

Conforme demonstrado na Figura 9, o aplicativo continuará a percorrer a árvore de diálogos do primeiro ao último nó, ao longo de cada nó acionado, deverá ser feita uma varredura do primeiro ao último nó filho e ao longo de cada nó filho acionado outra varredura deverá ser feita até alcançar o último nó na ramificação que está seguindo.

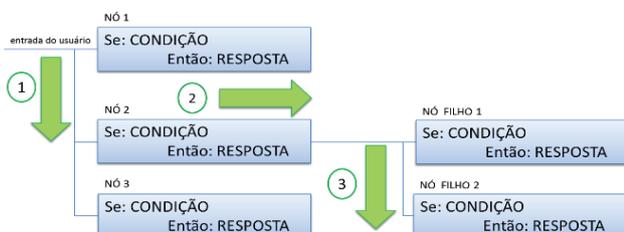


Figura 9 - Varredura no nós

Quando se começa a construir o diálogo, deve-se determinar os ramos a serem incluídos e onde colocá-los. A ordem dos ramos é importante porque os nós serão avaliados do primeiro ao último. O primeiro nó que deverá ser acionado é o qual, cuja condição corresponda à entrada do usuário; quaisquer nós que vierem depois na árvore não serão acionados.

Quando o serviço atingir o final de uma ramificação ou não conseguir encontrar uma condição que seja avaliada como verdadeira a partir do conjunto atual de nós filhos que está avaliado, a varredura deverá voltar para a base da árvore. E mais uma vez, o serviço irá processar os nós raiz do primeiro ao último. Se nenhuma das condições forem avaliadas como verdadeiras, a resposta do último nó da árvore, que deverá ter uma condição de *"anything\_else"*

sempre será avaliada como verdadeira e deverá ser retornada.

O fluxo do primeiro ao último poderá ser interrompido das seguintes maneiras:

Personalizando o que acontece depois que um nó for processado. Por exemplo, configurar um nó para pular diretamente para outro nó depois que ele for processado, mesmo que o outro nó seja posicionado anteriormente na árvore.

Configurando respostas condicionais para pular para outros nós.

Ativando as configurações de desvio para nós de diálogo. Os desvios também poderão afetar o modo como os usuários irão se movimentar pelos nós no tempo de execução. Programando os desvios longe da maioria dos nós e configurando retornos, os usuários poderão pular de um nó para outro e voltar novamente com mais facilidade. Os desvios serão melhor explicados adiante.

## VARIEDADE DE RESPOSTAS

Se os usuários retornarem ao serviço de conversação com frequência, eles poderão ficar entediados ao receber as mesmas saudações e respostas todas as vezes (SPANHOL, 2017). Será possível adicionar variações às respostas para que a conversa possa responder à mesma condição de maneiras diferentes.

No exemplo da Figura 10, o serviço fornece respostas diferentes para saudações ao usuário:

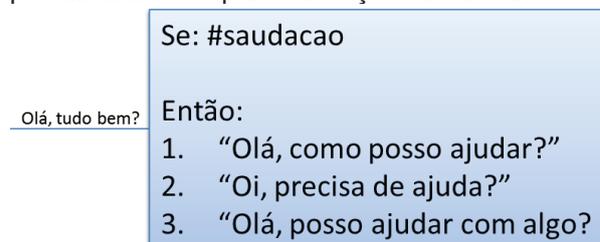


Figura 10 - Nó com variedade de respostas

Poderá optar por alternar as variações de resposta sequencialmente ou em ordem aleatória.

**Sequencial:** o sistema retornará à primeira variação de resposta na primeira vez que o nó de diálogo for acionado, a segunda variação de resposta na segunda vez em que o nó for acionado e assim por diante, na mesma ordem em que forem definidas as variações no nó.

**Randômico:** O sistema selecionará aleatoriamente uma cadeia de texto da lista de variações na primeira vez em que o nó do diálogo for acionado e selecionará aleatoriamente outra variação na próxima vez, mas sem repetir a mesma cadeia de texto consecutivamente.

### RESPOSTA COMPLEXA

Será retornado respostas com elementos interativos ou multimídia, como imagens ou botões clicáveis, para simplificar o modelo de interação do aplicativo e aprimorar a experiência do aluno.

**Imagem:** Poderá incorporar uma imagem na resposta. O arquivo de imagem de origem deverá estar hospedado em algum local na *web* e ter uma URL que possa ser usada como referência.

**Opção:** Adicionará uma lista com uma ou mais opções. Quando o usuário clicar em uma das opções, um valor de entrada será associado e enviado ao serviço. As opções poderão ser exibidas como botões clicáveis ou como uma lista suspensa.

**Pausa:** Forçará o aplicativo a aguardar por um número especificado de milissegundos antes de continuar com o processamento. Esse tipo de resposta será válido se for necessário executar uma ação que leve algum tempo. Por exemplo, um nó pai faz uma chamada ao banco de dados e exibe o resultado em um nó filho, enquanto o assistente faz a consulta no banco de dados o mesmo poderá exibir a seguinte mensagem ao usuário, "Aguarde enquanto verifico a informação".

### RESPOSTAS CONDICIONAIS

Um único nó de diálogo poderá fornecer respostas diferentes, cada resposta acionada por uma condição. Deverá usar esse método para abordar vários cenários em um único nó.

O nó ainda terá uma condição principal, que determinará a circunstância para usar o nó e processar os requisitos e respostas que ele irá conter.

No exemplo da Figura 11, o serviço usa informações coletadas do aluno para adaptar sua resposta e fornecer informações sobre a quantidade de faltas.

Quantas faltas eu tenho?	Condição: #faltas
	Resposta: If: \$disciplina == "Informática" "Você está com 3 faltas."
	If: \$disciplina == "Cálculo" "Você tem 5 faltas, o limite de faltas foi atingido."
	If: \$disciplina == "Programação" "Você não tem faltas em Programação."
	Else: "Por favor, informe a disciplina."

**Figura 11 - Nó de resposta condicionais**

Esse nó único agora fornecerá a função equivalente de quatro nós separados.

### A OPÇÃO JUMP TO

Será opcional configurar o assistente para pular para outro nó, para isso deverá ser especificado se a ação se destinará à uma resposta ou condição do nó de diálogo selecionado:

**Resposta:** Se a instrução estiver como alvo a seção de resposta do nó de diálogo selecionado, ela deverá ser executada imediatamente. Ou seja, o sistema não avaliará a condição do nó de diálogo, ele processará a resposta do nó de diálogo selecionado imediatamente. Designar a resposta será útil para encadear vários nós de diálogo juntos. A resposta será processada como se a condição desse nó de diálogo fosse verdadeira. Se o nó de diálogo selecionado tiver outra ação *Jump to*, essa ação também poderá ser executada imediatamente.

**Condição:** Se a instrução for designada à seção de condição do nó de diálogo selecionado, o serviço verificará primeiro se a condição do nó de destino é avaliada como verdadeira.

- Se a condição for avaliada como verdadeira, o sistema processará o nó de destino imediatamente.
- Se a condição não for avaliada como verdadeira, o sistema irá mover para o próximo nó irmão para avaliar sua condição e repetirá esse processo até encontrar um nó de diálogo com uma condição que seja avaliada como verdadeira.
- Se o sistema processar todos os nós irmãos e nenhuma das condições for avaliada como verdadeira, a estratégia básica de *fallback* deverá ser usada e o diálogo avaliará os nós no nível base da árvore de diálogo.

Designar a condição será útil para encadear as condições dos nós de diálogo. Por exemplo, primeiro verificar se a entrada contém uma intenção, como #declaracoes, e se isso acontecer, será possível verificar se a entrada contém entidades, como @vinculo, @carga\_horaria ou @forma\_de\_ingresso. As condições de encadeamento ajudarão a estruturar árvores de diálogo maiores.

**Aguardar a entrada do usuário:** aguarda nova entrada do usuário e, em seguida, começará a processá-la do nó para o qual é acessado. Essa opção será útil quando, o nó de origem fizer uma pergunta e se deseja pular para um nó separado para processar a resposta do usuário à pergunta.

### PROCESSAMENTO DE DIÁLOGO

Entenda como a caixa de diálogo é processada quando o aluno interage com o assistente digital, a Figura 12, mostra um exemplo de interação no ambiente de testes.

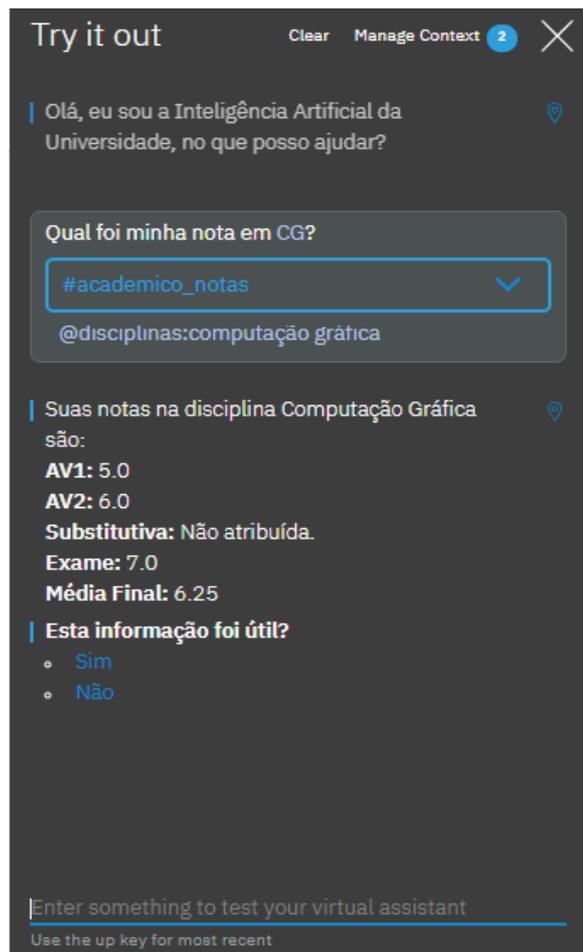


Figura 12 - Interface gráfica de testes

### VARIÁVEIS DE CONTEXTO

Uma variável de contexto deverá ser uma variável que se define em um nó. Especificando um valor padrão para isso. Poderá referenciar uma variável de contexto a partir das condições de resposta do nó de diálogo para mostrar diferentes respostas dependendo de um valor fornecido por um serviço externo ou pelo usuário.

### PASSANDO O CONTEXTO DO APLICATIVO

Informações do aplicativo deverão ser passadas para o diálogo definindo uma variável de contexto e passando a variável de contexto para o diálogo.

Por exemplo, na Figura 13 o aplicativo poderá definir uma variável de contexto \$hora\_do\_dia e passá-la para a caixa de diálogo que poderá usar as informações para adaptar a saudação exibida ao usuário.

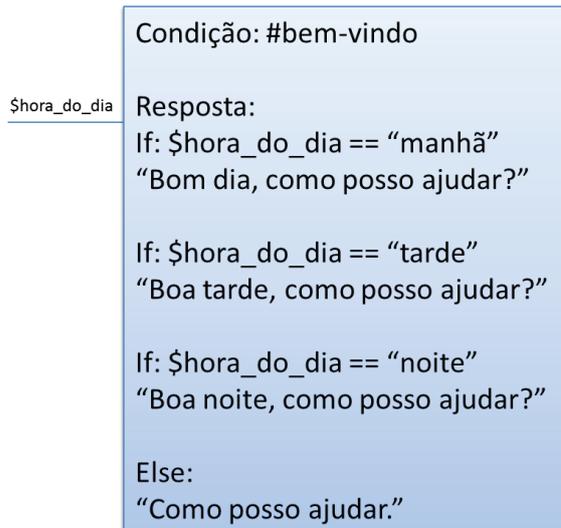


Figura 13 - Nó com variáveis de contexto

Neste exemplo, o diálogo sabe que o aplicativo define a variável com um desses valores: manhã, tarde ou noite. Ele irá verificar cada valor e, dependendo do valor presente, retornará à saudação apropriada. Se a variável não for passada ou tiver um valor que não corresponda a um dos valores esperados, será exibida uma saudação genérica ao usuário.

### PASSANDO CONTEXTO DE UM NÓ PARA OUTRO NÓ

O diálogo também poderá adicionar variáveis de contexto para passar informações de um nó para outro ou para atualizar os valores das variáveis de contexto. Como o diálogo pede e obtém informações do usuário, ele poderá acompanhar as informações e referenciá-las posteriormente na conversa. Por exemplo, em um nó, pode-se solicitar aos alunos o nome deles e, em um nó posterior, chama-lo pelo nome, este exemplo é demonstrado na Figura 14.

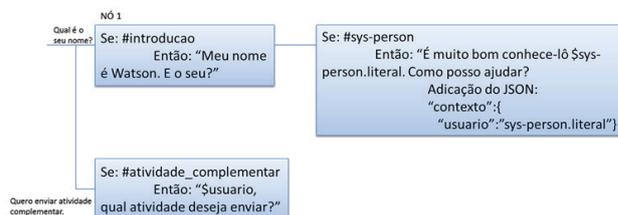


Figura 14 - Nó passando informações para outro nó

Neste exemplo, a entidade do sistema @sys-person será usada para extrair o nome do usuário da entrada, caso o usuário forneça. No editor JSON, a variável de contexto \$nome\_usuario deverá ser definida com o valor @sys-person. Em um nó subsequente, a variável de contexto \$nome\_usuario será incluída na resposta para chamar o usuário pelo nome.

## PROCESSAMENTO DAS VARIÁVEIS DE CONTEXTO

Para um nó com respostas condicionais, a variável de contexto deverá ser criada e configurada quando a condição de uma resposta específica for atendida e essa resposta for processada. Por exemplo, se for definido uma variável de contexto para a resposta condicional nº 1 e a resposta condicional do processo de serviço for a nº 2, a variável de contexto que for definida para a resposta condicional nº 1 não deverá ser criada e configurada.

## DESVIO DE CONVERSA

Um desvio ocorrerá quando o usuário estiver no meio de um fluxo de diálogo projetado para atender a um objetivo e alternar abruptamente os tópicos para se iniciar um fluxo de diálogo projetado para direcionar um objetivo diferente. Se nenhum dos nós na ramificação da caixa de diálogo que estará sendo processada corresponder à meta da entrada mais recente do usuário, a conversa retornará à árvore para verificar as condições do nó raiz para uma correspondência apropriada. As configurações de desvio disponíveis por nó oferecerão a capacidade de adaptar ainda mais esse comportamento.

Com as configurações de desvio, será possível permitir que a conversa retorne ao fluxo de diálogo que foi interrompido quando o desvio ocorreu. Por exemplo, o aluno pode estar solicitando mudança de turno, mas alterna o tópico para perguntar sobre a mensalidade. A caixa de diálogo deverá responder à pergunta sobre mensalidade e depois levar o usuário de volta ao ponto em que ele parou no processo de solicitação de mudança de turno. Permitir desvios de conversa dará aos usuários mais controle sobre o fluxo da conversa em tempo de execução. Eles poderão alterar os tópicos, seguir um fluxo de diálogo sobre o tópico não relacionado até o fim e, em seguida, retornar para onde estavam antes. O resultado será um fluxo de diálogo que simulará com mais precisão uma conversa humana.

## IMPORTÂNCIA DOS SLOTS

Deverá usar os slots para obter as informações necessárias para responder com precisão o usuário. Por exemplo, se os alunos perguntarem sobre o atendimento humano, mas o horário interfere nesse tipo de atendimento, o assistente digital irá fazer uma pergunta de acompanhamento sobre qual tipo de atendimento o aluno deseja antes de responder. Poderá então adicionar condições de respostas que levam em consideração as informações fornecidas, exemplo na Figura 15.

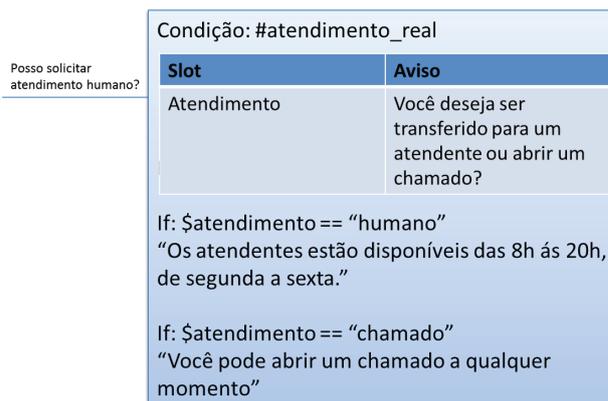


Figura 15 - Nó com slots

Os slots ajudarão a coletar informações necessárias para concluir uma tarefa complexa para o aluno, como coletar dados para abertura do chamado, exemplo demonstrado na Figura 16.

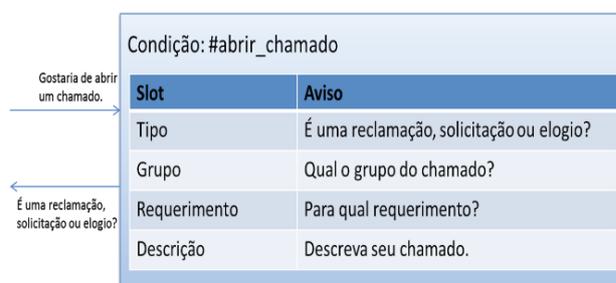


Figura 16 - Slots coletando informações

O usuário poderá fornecer valores para vários slots ao mesmo tempo. Por exemplo, a entrada contém dois dos valores necessários ausentes: o tipo e o grupo. O serviço deverá reconhecer e armazenar os dois, cada um em seu slot correspondente. Em seguida, ele exibirá o *prompt* associado ao próximo slot vazio, conforme exemplo da Figura 17.

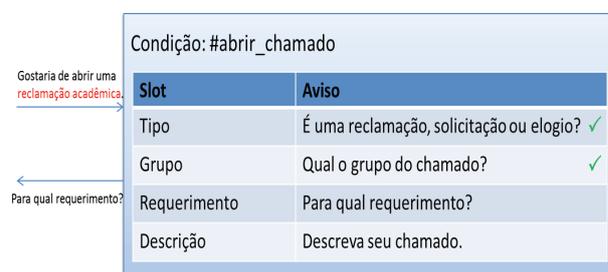


Figura 17 - Slot verificando entradas

Os slots possibilitarão que o serviço responda a perguntas de acompanhamento sem ter que restabelecer a meta do usuário. Por exemplo, um usuário poderá solicitar uma disponibilidade de bolsa de estudo e fazer uma pergunta sobre bolsa de estudo em outro campus ou de outro modelo. As variáveis necessárias serão salvas pelo assistente, como tipo

ESTUDO PARA DESENVOLVIMENTO DE UM ASSISTENTE DIGITAL COM INTELIGÊNCIA ARTIFICIAL VISANDO POTENCIALIZAR O ATENDIMENTO AOS ALUNOS DE INSTITUIÇÕES DO ENSINO SUPERIOR  
Diego Rodrigues Lima da Silva, Gabriel Nascimento Lunardelli, Guilherme Oliveira Shiroma, Willian Levi Sharo de Oliveira, Luciano Nascimento Corsino<sup>5</sup>

de bolsa e campus, em slots, se o usuário fizer uma pergunta de acompanhamento com novos valores de variáveis, o assistente poderá substituir os valores de slot pelos novos valores fornecidos e fornecer uma resposta que reflita a nova informação, exemplo na Figura 18.



Figura 18 - Slot complexo

O uso de slots deverá produzir um fluxo de diálogo mais natural entre o usuário e o serviço, sendo mais fácil para gerenciamento do que tentar coletar as informações usando muitos nós separados.

### 3. RESULTADO E DISCUSSÃO

#### RESULTADO ESPERADO

O principal resultado esperado será que haja uma melhora significativa na qualidade do atendimento oferecido aos alunos da instituição acadêmica, os estudos sobre assistentes digitais mostram que o mesmo poderá ter a capacidade de realizar o primeiro nível de atendimento do *call center* conforme sua eficácia for evoluindo. Para se alcançar o resultado esperado será necessário realizar um levantamento de dados eficaz na fase inicial, um desenvolvimento muito bem estruturado como foi explicado neste artigo, realização de testes de forma exaustiva e a monitoração das conversas após a implementação do assistente digital.

#### LEVANTAMENTO DE DADOS

O primeiro passo para criação do assistente digital deverá ser definir o público alvo e então realizar um levantamento de dados, para isso poderá ser criada uma persona. A persona deverá ser construída com dados reais e em cima do comportamento e características do aluno. Como também, representará a criação de situações reais vividas pelo aluno dentro da instituição acadêmica. A persona poderá ser definida através do contato direto com o público-alvo. A natureza representativa e precisa dos dados será importante para realizar uma análise das informações adquiridas e mapear os dados para se decidir quais assuntos serão prioritários no início do desenvolvimento do assistente digital e assim definir quais serão as soluções apresentadas pela aplicação.

#### REALIZAÇÃO DE TESTES

A realização de testes será outro ponto crucial para que o assistente digital tenha um bom desempenho.

O primeiro teste a ser realizado será durante o próprio desenvolvimento, para verificar se o assistente digital está se comportando corretamente conforme programado. O segundo teste será disponibilizar o assistente digital para uma parcela exclusiva de usuários, este modelo de teste terá como objetivo saber como será o comportamento do assistente em contato direto com seu público.

Os testes deverão resultar em informações relevantes para o aprimoramento do assistente digital.

#### MONITORAÇÃO E APRIMORAMENTO

Esta etapa deverá ser contínua durante o tempo que o assistente digital estiver disponível para o público. Será de extrema importância monitorar o comportamento do assistente digital e com os dados de *feedback* aprimorar seu desempenho.

Poderá usar a monitoração para responder a questões avaliativas como:

- No último mês, qual foi o dia em que houve o maior ou menor número de conversas?
- Qual foi a média de conversas por semana durante o último mês?
- Quais intenções apareceram com mais frequência na semana passada?
- Quais valores de entidade foram reconhecidos mais vezes em fevereiro?

#### GRÁFICOS E ESTATÍSTICAS

Criar *scorecards* estatísticos fornecerá dados de registros para o aplicativo, que podem trazer as seguintes informações:

- Total de conversas
- Média de mensagens por conversação
- Máximo de conversas
- Total de mensagens
- Compreensão fraca
- Usuários ativos
- Média de conversas por usuário
- Principais intenções e entidades

O próprio serviço do Watson Assistant já disponibiliza uma área para testes e monitoração.

#### DISCUSSÃO

O assistente digital será útil para aquelas instituições de ensino superior, que pretendem melhorar o atendimento aos alunos, portanto, quando o *call center* da instituição do ensino superior (IES) não consegue atender toda a demanda de atendimento, quando os alunos reclamam das soluções prestadas ou se a IES está buscando redução de custos, estes são os principais motivos para determinar a imple-

mentação de um assistente digital.

O aprimoramento do assistente digital deverá ser contínuo durante todo o tempo que o aplicativo estiver disponível ao público, para isso deverá ter uma equipe direcionada para realizar o monitoramento e análise de dados profunda e eficaz para melhorar o desempenho do assistente digital.

É importante ressaltar que para a análise de dados ter resultados satisfatórios será preciso que a mesma seja feita de forma adequada, só assim o desempenho do assistente digital será aprimorado continuamente.

### CONCLUSÃO

O desenvolvimento do presente estudo possibilitou uma análise de como um assistente digital inteligente poderá melhorar o atendimento realizado pelas instituições do ensino superior aos alunos.

Foram demonstrados o problema, justificativa, materiais e metodologia para o desenvolvimento, evidenciando as tecnologias necessárias para se atingir o objetivo, além de aspectos técnicos, demonstrando como funcionará o fluxo de conversa, o modelo de aprendizado de máquina e como o processamento de linguagem natural possibilitará ao assistente comparar o contexto das perguntas evitando fornecer respostas incorretas. No entanto, foi reforçado que durante e após o desenvolvimento, a fase de testes será essencial para a evolução do aplicativo, e será crucial realizar análises de conversas após a implementação para que seja possível melhorar o desempenho do assistente digital de forma contínua e atingir o resultado esperado.

### REFERÊNCIAS

BATISTA, Gustavo Enrique de Almeida Prado Alves. *Pré-processamento de dados em aprendizado de máquina supervisionado*. 2003. 232 f. Tese (Doutorado em Ciência) – Universidade de São Paulo, São Carlos, 2003.

BERGLER, Frank. *Application program interface*. Washington DC: U.S. Patent and Trademark Office, 1996.

FONSECA, Rúben; SIMOES, Alberto. *Alternativas ao XML: YAML e JSON*. Braga : Universidade do Minho, 2007.

GARTNER SUMMITS. CRM Strategies and Technologies to Understand, Grow and Manage Customer Experiences. Los Angeles, p. 1-9, abr. 2011.

GOLDBERG, David E.; HOLLAND, John. H. *Genetic algorithms and machine learning*. Ann Arbor: Universidade de Michigan, 1988.

HILL, Jennifer; FORD, W. Randolph; FARRERAS, Ingrid G. *Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations*. Nova Iorque: Science Direct, 2015.

LOPES, Lucene; VIEIRA, Renata. *Processamento de linguagem natural e o tratamento computacional de linguagens científicas*. Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul, 2010.

MONARD, Maria Carolina.; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquinas. In: MONARD, Maria Carolina.; BARANAUSKAS, José Augusto. *Sistemas Inteligentes Para Engenharia..* São Paulo: Universidade de São Paulo, 2003, p. 39-56.

PEREIRA, Silvio do Lago. *Processamento de Linguagem Natural*. São Paulo: Universidade de São Paulo, 2011.

RICH, Elaine. *Inteligência Artificial*. 2.ed. Altus Cove: McGraw Hill, 1988.

SHEVAT, Amir. *Designing bots: Creating conversational experiences*. O'Reilly Media: São Francisco, 2017.

SPANHOL, Tamira Silva. *Um estudo sobre a interação entre usuários e chatterbots*. 2017. 65 f. Trabalho de Conclusão de Curso (Bacharel em Tecnologia) – Universidade Federal de Santa Catarina, Araranguá, 2017.

WALKER INFORMATION. *Customers 2020: The Future of B-to-B Customer Experience*. Indianapolis, p. 1-42, 2011.

WEBER, Leo; KLEIN, Pedro Antonio Trierweiler. *Aplicação da lógica fuzzy em software e hardware*. Editora da ULBRA: Canoas, 2003.

ZUBEN, Fernando José Von. *Introdução a Inteligência Artificial*. Campinas: Universidade Estadual de Campinas, 2011.