

# AKIRA: UM CHATTERBOT PARA EMPRESAS DE TELEMÁTICA

Ismael Laurentino Pires, Wilson Souza de Medeiros Junior

Curso de Bacharelado em Ciência da Computação – Universidade do Sul de Santa Catarina (UNISUL) – Campus Tubarão

Caixa Postal 370 – 88704-900 – Tubarão – SC – Brasil

ismaelpires@gmail.com, wilsonmdrs@gmail.com

**Abstract:** *This article presents the development of a ChatterBot called AKIRA, an intelligent conversation system that seeks to simulate a human interlocutor through text messaging. In order to improve the treatment of intention between the user and the bot, an organization model of the AIML knowledge base was created, providing a more natural experience between both. In this way, it was possible to assist in customer service in a telematics company, performing the initial service, solving simple problems and answering doubts.*

**Resumo:** *Este artigo apresenta o desenvolvimento de um ChatterBot denominado AKIRA, um sistema inteligente de conversação que busca simular um interlocutor humano através de mensagens de texto. Com o objetivo de melhorar o tratamento de intenção entre usuário e bot, foi criado um modelo de organização da base de conhecimento AIML, proporcionando uma experiência mais natural entre ambos. Desse modo, foi possível auxiliar os atendimentos ao cliente em uma empresa de telemática, realizando o atendimento inicial, resolvendo 68,33% dos problemas simples e respondendo dúvidas.*

**Palavras-chave:** ChatterBot. AIML. iAIML.

## 1. Introdução

ChatterBot é um programa desenvolvido para interagir com usuários em ambientes de bate-papo<sup>1</sup>. Este programa surge como uma ferramenta importante no relacionamento entre empresa e cliente, devido ao fato de que possui uma ampla base de conhecimento e disponibilidade para atender quando for chamado, independente de horário. É capaz de interagir de forma natural, simulando um atendente humano.

---

<sup>1</sup> Ambientes virtuais dedicados a comunicação (por escrito) através da internet e entre duas ou mais pessoas.

O *chat*<sup>2</sup> é a ferramenta utilizada por clientes que buscam soluções rápidas, grande parte dos consumidores consideram uma boa forma de comunicação [THE ZENDESK BENCHMARK, 2015]. Atualmente um simples ChatterBot consegue resolver de 70% a 90% dos problemas. O restante da demanda é direcionado para um atendimento especializado [PRADO, 2016]

Os ChatterBots tem sido considerado o futuro da interação entre consumidor e empresa. Muitas empresas acreditam que estes robôs podem estar envolvidos na próxima grande revolução da interação entre pessoas e marcas. Facebook, Microsoft, Telegram, WeChat, Kik Messenger já investem nos *bot* como uma nova interface de interação [GOMES, 2016].

Nesse contexto, de forma a aplicar esta ferramenta no meio empresarial, surgiu a necessidade de inseri-la em uma empresa de telemática, onde as conversas via *chat* vão além de um simples bate-papo, mas o teor da conversa concentra-se em sua maioria, na linguagem técnica, direcionada aos produtos e serviços oferecidos pela empresa.

A Empresa de Telemática Del Grande desenvolve soluções de tecnologia CTI para gravação telefônica, sistemas de autoatendimento, URA, DAC, PABX IP com recursos para *call centers* e SAC's, correios de voz, discadores automatizados e servidores de fax [DEL GRANDE, 2017]. Para prestar atendimento aos seus clientes a empresa utiliza telefones, e-mails e *chats* online como meios de comunicação. Os atendimentos por *chat* são realizados pelos colaboradores da empresa e iniciados pelos clientes. A empresa não possui um fluxo grande de chamados diários, com uma média mensal de 140 atendimentos via chat, no entanto, os atendentes possuem outras funções, ocasionando uma demora nos poucos atendimentos efetuados.

Baseado nisso, o ChatterBot Akira (“brilhante” em japonês) tem por finalidade, proporcionar um atendimento inicial, com o objetivo de resolver problemas simples sem a intervenção humana, reduzir o tempo de espera para o atendimento, auxiliar o atendente como um especialista e proporcionar atendimento em tempo integral.

As seções deste artigo estão organizadas conforme a descrição a seguir. Na seção 2 é apresentada a contextualização de forma a esclarecer os conceitos de ChatterBot, AIML e iAIML, logo em seguida a seção 3 apresenta os trabalhos correlatos mostrando alguns projetos desenvolvidos sobre o mesmo assunto. Na seção 4 apresentamos os materiais e métodos, explicando as ferramentas utilizadas e os procedimentos adotados para elaboração e teste do ChatterBot. A seção 5 apresenta os resultados obtidos e as discussões sobre os mesmos. Por fim, na seção 6 é apresentada a conclusão do trabalho.

## **2. Contextualização**

### **2.1 ChatterBot e AIML**

Um ChatterBot é um programa de computador que tem por objetivo responder perguntas de tal forma que a pessoa que estiver interagindo com ele tenha a impressão de estar conversando com outra pessoa. Ou seja, ele emula uma conversa, uma comunicação humana [TEIXEIRA; MENEZES, 2003]. Segundo Laven (1996), “é um programa que tenta simular a conversa escrita, com o usuário, tentando, pelo menos

---

<sup>2</sup> Sala de bate-papo

temporariamente, se fazer passar por um ser humano”.

Diferentemente dos *chats* convencionais, os ChatterBots - ou robôs de conversação - são máquinas com as quais os humanos interagem por meio da linguagem escrita. A diferença dos *chats* convencionais, em que deve haver pessoas interagindo com pessoas por meio do computador, é que os chatterbots são programados para tentar imitar um interlocutor em conversação.

De acordo com Bush (2001), o termo AIML significa *Artificial Intelligence Markup Language*. AIML é uma linguagem de marcação baseada em XML (*extensible Markup Language*), destinada a criação de aplicações compostas por inteligência artificial, mantendo a implementação simples de programar, fácil de entender e altamente sustentável.

AIML possibilita a criação de um robô com Inteligência Artificial que interage com usuários, por meio de conversas, fazendo com que um humano pense que esteja “conversando” com outro humano. A linguagem AIML é capaz de representar e relacionar expressões em linguagem natural, permitindo manter um diálogo simples.

A linguagem AIML é documentada através de etiquetas, que permitem que seja manipulada e desenvolvida. As etiquetas básicas de AIML são apresentadas na Figura 1. Uma lista mais completa pode ser visualizada em [BUSH, 2001].

```
<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0.1" encoding="UTF-8"?>
  <category>
    <pattern> BOM DIA, TUDO BEM?</pattern>
    <template>
      Bom dia, tudo certo.
    </template>
  </category>
</aiml>
```

**Figura 1. Etiquetas básicas de AIML**

**Fonte: AUTORES, 2017.**

Na figura 1 é apresentado primeiro a etiqueta que define a versão e a codificação do XML utilizado. A principal etiqueta é `<aiml>` que marca o início e o fim de um documento AIML. Ela contém a versão e codificação das informações utilizadas no documento. A etiqueta `<aiml>` deve conter pelo menos uma etiqueta `<category>`. Pode-se criar vários arquivos AIML onde cada arquivo contém uma única etiqueta `<aiml>`. A etiqueta `<category>` é uma unidade do conhecimento. Para cada possível pergunta feita ao ChatterBot é necessário criar uma nova categoria. A solicitação que é feita ao ChatterBot fica entre as etiquetas `<pattern>` e a resposta entre as etiquetas `<template>`. Tudo que estiver entre as etiquetas `<template>` será analisado pelo programa como resposta a solicitação.

## 2.2 iAIML

Conforme Neves (2005), iAIML (AIML com intenção) é um mecanismo para tratamento de informações de intenção para ser acrescentado a sistemas baseados em AIML, a fim de minimizar os problemas referentes ao desempenho dos chatterbots em conversações. iAIML é uma extensão do AIML que permite que seja adicionada classes

de intenção. A solução desenvolvida passa essencialmente por uma redefinição da base AIML padrão, que incorpora informações de intencionalidade e regras que utilizam essas informações na interpretação e geração de sentenças.

Neves (2005) usou para tratamento de intenção o princípio da Teoria da Análise da Conversação [MARCUSCHI, 1986

Segundo Marcuschi (1986), a conversação é um fenômeno único, que pode ser analisada a partir de organizadores locais e globais, nesse sentido, Marcuschi divide o diálogo em três fases do ponto de vista global (abertura, desenvolvimento e fechamento) e aponta uma série de elementos organizadores locais (o turno, as sequências, a preferência e os tópicos). Dentre esses elementos locais, destacamos as sequências, mais especificamente, os pares adjacentes, ou pares conversacionais, por sua estrutura altamente padronizada (NEVES, 2005, p20).

### **3. Trabalhos Correlatos**

Os ChatterBots têm se tornado uma alternativa eficaz e amigável de interação com o usuário, consequentemente, sua aplicação no mercado de trabalho vai de entretenimento à suporte na resolução de problemas. Empresas de grande porte como Google, Facebook, Microsoft, IBM, Telegram, Apple investem nessa alternativa para melhorar a interface homem maquina [GOMES, 2016].

Leonhardt [2003] apresenta um *bot* para interação em ambientes educacionais, o qual trata assuntos relacionados as matérias das disciplinas de física e rede de computadores, possui disponibilidade via web e a cada interação com o usuário o ChatterBot Elektra responde e sugere um novo assunto, porém não possui tratamento para intenção, sentenças desconhecidas e mensagens repetitivas. Outra deficiência do projeto é a análise das conversas, feitas por intermédio de logs da aplicação, o que dificulta a análise e correção dos erros contidos na base

Wallace [2003] apresenta o ChatterBot A.L.I.C.E., trabalho que foi um marco no que diz respeito a gerações de *bots*, isso, devido à criação da linguagem AIML, a qual serve de base para a maioria dos *bots* criados atualmente. Sua inovação está na forma como é apresentado: além de muita documentação, apresenta uma saudação sonora ao visitante, tem um grande poder de comunicação, além de uma interface gráfica que estimula o diálogo [BUSH, 2001]. Atualmente existe uma Fundação que promove a disseminação do software gratuito. O chatterbot original, A.L.I.C.E. tem uma base de conhecimento constituída por centenas de fatos, citações e ideias de seu criador. Apresenta um vocabulário de mais de 5000 palavras. É programada para dar muitas informações a seu respeito e pode sugerir até que o usuário a veja cantar. No entanto, Wallace não oferece tratamento para intenções do usuário, assim como tratamento para sentenças desconhecidas e mensagens repetitivas.

### **4. Materiais e Métodos**

#### **4.1 Materiais**

Para o desenvolvimento do ChatterBot Akira foram utilizadas algumas ferramentas, as quais são: plataforma web de gerenciamento, banco de dados, importador e interpretador AIML.

As tecnologias utilizadas para o desenvolvimento da plataforma foram as linguagens de programação PHP, *Javascript*, HTML e CSS. Utilizou-se também um

banco de dados para armazenamento do conteúdo que veio a compor a base AIML. O tipo da entidade utilizada para guardar os dados foi *MariaDB* com a versão 10.0.13, foi definido um modelo relacional com 16 tabelas.

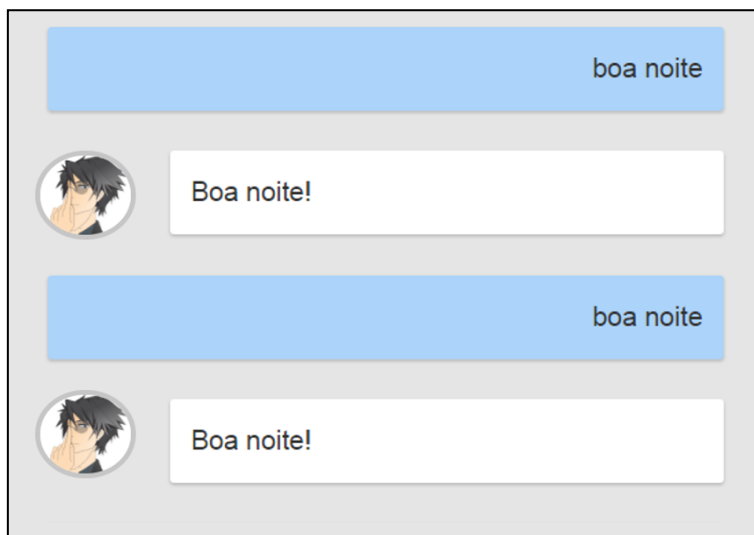
AIML é uma linguagem de marcação que precisa de um interpretador, o qual irá ler o roteiro escrito e executar as funções conforme programado. Há diversos interpretadores na internet disponíveis para download, optamos então por um que melhor se adaptasse ao cenário atual da empresa no que diz respeito às linguagens utilizadas. Sendo assim, utilizamos um interpretador escrito em Python, o interpretador PyAIML. A fim de testar a base AIML, utilizou-se a interface desenvolvida pela Del Grande, o Intelix Chat. Para integrar o sistema ChatterBot com o Intelix, foi utilizado o serviço “*DGChatServer*” que processa os dados e os eventos realizados pelo cliente, e as ações por parte do ChatterBot, possibilitando assim, a comunicação entre ambos.

## 4.2 Métodos

Os métodos utilizados dizem respeito ao planejamento da base de conhecimento AIML, coleta de dados, mapeamento dos dados cadastrados, cadastro e gerenciamento da base AIML, assim como, os procedimentos de testes e parâmetros para análise dos resultados

### 4.2.1 Planejamento da Base de Conhecimento

A maneira como o *bot* irá pensar é o fator mais importante na construção de um ChatterBot, é o que diferencia cada *bot* fabricado. Uma base de conhecimento AIML padrão, responde ao cliente sempre baseado no texto que foi inserido, independente do contexto em que essa mensagem se encontra (Figura 2).



**Figura 2 – Resposta padrão de um ChatterBot**

**Fonte: AUTORES, 2017.**

A partir dessa análise, percebeu-se a necessidade de algumas características, que ajudariam o *bot* a interagir de uma maneira mais natural, para isso foi utilizado o tratamento de intenção de Neves (2005).

- a) Reconhecer quando o cliente está fora de contexto;

Utilizou-se o iAIML (item 2.2), separando a conversa em dois tipos de contexto, global e local. O contexto global separa a conversa inteira em três partes (seções)

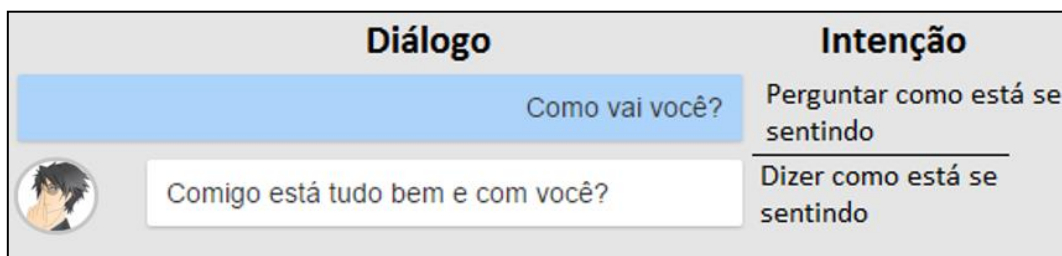
distintas, abertura, desenvolvimento e fechamento. O monitoramento da conversa mediante as seções criadas, deu-se através das seguintes regras:

- Abertura – mensagens dessa seção não podem ser inseridas nas demais seções. Um exemplo prático, é o cliente dizer olá quando está se despedindo.
- Desenvolvimento e fechamento: mensagens dessas seções podem ser inseridas a qualquer momento.

Como qualquer conversa o desenvolvimento é a parte principal, pois trata do real motivo do contato, dessa maneira possui livre acesso independente do contexto atual. No caso do fechamento, fica livre ao cliente encerrar a conversa quando bem entender.

No contexto local, cada mensagem inserida pelo cliente, possui uma classe, a qual reflete a intenção da mesma, cada classe possui uma ou mais classes correspondentes, denominados “pares adjacentes”.

Cada mensagem cadastrada é marcada com duas classes de intenção (Figura 3), a primeira diz respeito à intenção do usuário, e a segunda à intenção do *bot*. Com isso é possível monitorar a conversa a nível local, caso o cliente saia do contexto.



**Figura 3. Exemplo de classes de intenção**

**Fonte: AUTORES, 2017.**

As classes de intenção (Tabela 1) foram definidas de acordo com o sentido da mensagem inserida pelo usuário (*user\_intention*) ou pelo *bot* (*bot\_intention*). As classes de intenção surgiram à medida que a base de conhecimento AIML se expandia, conforme a necessidade, totalizando 62 classes. Para as mensagens que faziam referência a algum produto da empresa, sua classe de intenção era o próprio produto.

**Tabela 1. Exemplos de classes de intenção e suas respectivas seções**

Abertura	Desenvolvimento	Fechamento
Saudar	Reclamar	Despedir-se
Apresentar-se	Continuar conversa	Aceitar Despedidas
Cumprimentar	Duvida	Negar despedidas
Perguntar sobre o bot	Perguntar se o usuário entendeu a mensagem	Perguntar se precisa de mais alguma coisa
Responder sobre o bot	Perguntar se o usuário conseguiu acessar o link	Anunciar que vai encerrar o dialogo

b) Reconhecer quando o cliente está sendo repetitivo;

Para atender esse requisito foram criadas categorias de controle para algumas classes das seções de abertura, desenvolvimento e fechamento (Tabela 2).

**Tabela 2. Exemplos de categorias de controle.**

<b>Categorias de controle</b>	<b>Classe de Intenção</b>	<b>Seção</b>
CTRL PCS	Perguntar como está se sentindo	Abertura
CTRL PAJ	Pedir ajuda	Desenvolvimento
CTRL DEP	Desculpar-se	Desenvolvimento

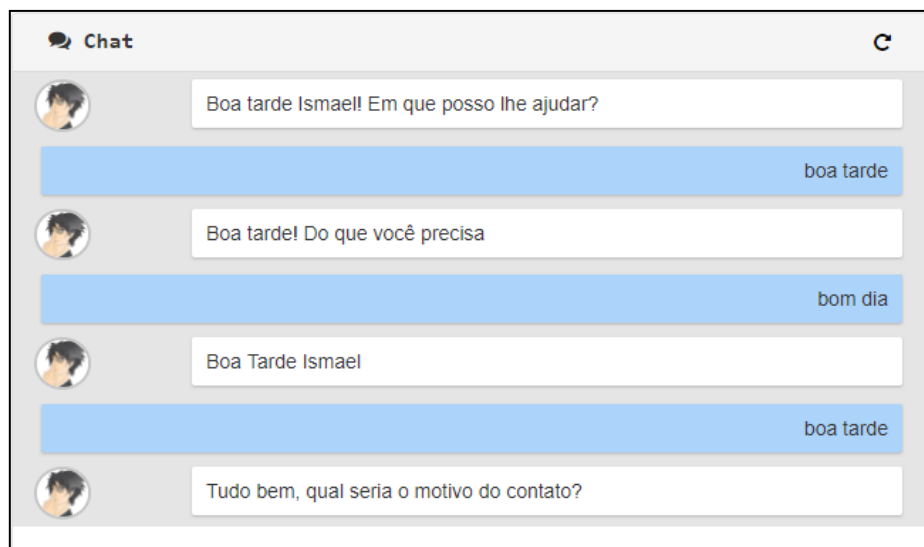
Basicamente, o monitoramento foi feito através de variáveis de controle que indicavam quando o cliente estava sendo repetitivo ao inserir mensagens com a mesma intenção (Figura 4), essas variáveis iniciavam com “False”. Caso a intenção fosse usada, o valor mudaria para “True” e cairia no campo de tratamento de mensagem repetitiva.

```
<category>
  <pattern>CTRL_CUM</pattern>
  <template>
    <condition name="CTRL_CUM" value=TRUE>
      <think><set name="BOT_INTENTION">CONTINUAR CONVERSA</set></think>
      <random>
        <li>Ok! Qual seria o motivo do teu contato?</li>
        <li>Tudo bem!Do que você precisa?</li>
      </random>
    </condition>
    <condition name="CTRL_CUM" value=FALSE>
      <think><set name="BOT_INTENTION">CUMPRIMENTAR</set></think>
      <think><set name="CTRL_CUM">TRUE</set></think>
      <get name="TEMPLATE"></get>
    </condition>
  </template>
</category>
```

**Figura 4. Exemplo de uma categoria de controle – classe de intenção cumprimentar**

**Fonte: AUTORES, 2017.**

Na Figura 5, a mensagem do cliente “boa tarde” pertence à classe saudar, o *bot* responde normalmente e aguarda, logo em seguida o cliente insere uma segunda saudação, além de reconhecer que o cliente está sendo repetitivo o *bot* reconhece que a saudação “bom dia” não se usa no período da tarde, sendo assim responde ao usuário de maneira correta e pergunta o motivo do contato afim de dar andamento na conversa.

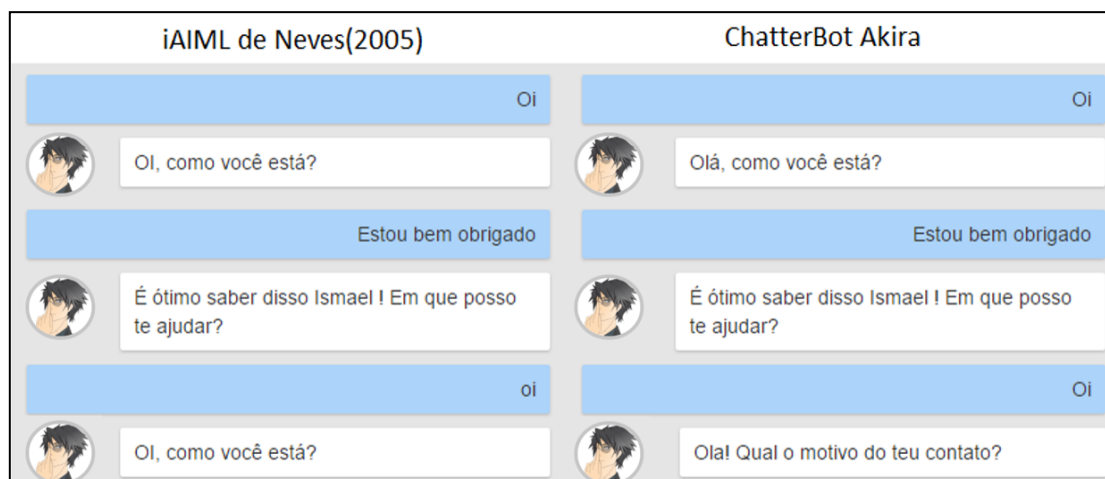


**Figura 5. Exemplo de conversa com tratamento de repetição**

**Fonte: AUTORES, 2017.**

O monitoramento do período correto para as saudações (Figura 5), foi feito mediante uma variável que utiliza o horário do sistema para definir o período atual, com isso, antes de responder ao cliente o bot verifica se o período está correto.

O modelo criado para tratamento de repetição se diferencia do modelo de Neves (2005), pelo fato de que, independente do momento em que cliente repete a intenção, o *bot* irá tratá-la da mesma maneira, sendo que o iAIML de Neves (2005) trata apenas repetições consecutivas (Figura 6).



**Figura 6. Diferenças no tratamento de repetição**

**Fonte: AUTORES, 2017.**

### c) Tratamento de sentenças desconhecidas

O cadastro do coringa asterisco (\*) (figura 7), proporciona uma válvula de escape caso não tenha nenhuma sentença equivalente à mensagem inserida.



```

<category>
  <pattern>*</pattern>
  <template>
    Não compreendi sua mensagem, você poderia ser mais claro?
  </template>
</category>

```

**Figura 7. Sentença desconhecida padrão**

**Fonte: AUTORES, 2017.**

Ao inserir condições (figura 8) para verificar as classes de intenção, foi possível melhorar o tratamento das sentenças desconhecidas de acordo com a última intenção do bot. Sendo assim o *bot* poderá retomar a conversa mesmo não encontrando uma sentença equivalente. Porém se o usuário continuar inserindo sentenças desconhecidas, o sistema contabiliza e transfere para um atendente humano caso o número de tentativas exceda o limite.

```

<category>
  <pattern>*</pattern>
  <template>
    * Verifica a ultima intenção do bot *
    <condition name="BOT_INTENTION" value=[Classe de intenção]>
    * altera o valor da variavel UNKNOWN *
      <think><set name="UNKNOWN">TRUE</set></think>
    * Mensagem de retomada de conversa de acordo com o ultimo assunto tratado *
      Desculpe, mas não entendi a sua mensagem, tem a ver com [Classe de intenção]?
    </condition>
    * Verifica o numero de vezes que o bot não reconheceu a mensagem *
    <condition name="NUM_TRY_UNKNOWN" value=[Número de tentativas]>
    * Encaminha o atendimento para um atendente humano
      <srai>EXHAUSTEDTRY</srai>
    </condition>
    * Verifica se a variavel UNKNOWN é False *
    <condition name="UNKNOWN" value=FALSE>
    * Mensagem padrão de sentenças desconhecidas *
      Não compreendi o que você disse. Por favor seja mais claro.
    </condition>
  </template>
</category>

```

**Figura 8. Sentença desconhecida modificada**

**Fonte: AUTORES, 2017.**

## 4.2.2 Coleta de Dados

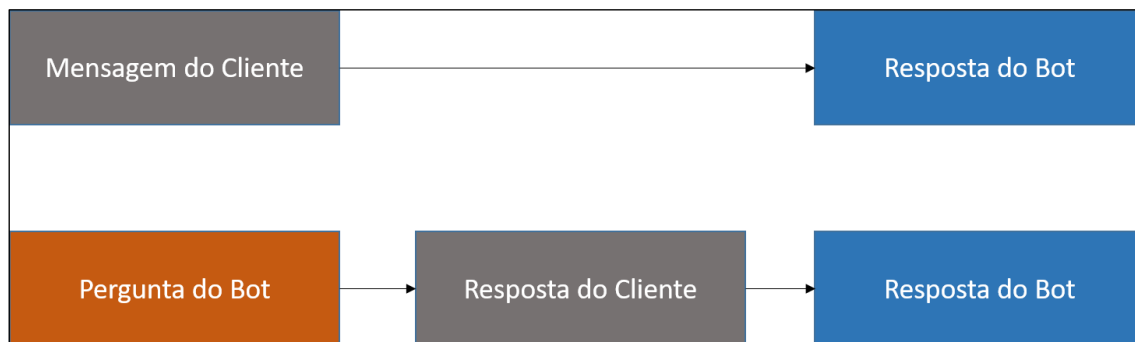
A coleta de dados pertinentes às seções de abertura e fechamento, no qual o conteúdo são as diversas saudações, cumprimentos, agradecimentos, etc., procedeu-se através do conhecimento dos autores e de buscas na internet.

A seção de desenvolvimento, inclui assuntos tratados na empresa em questão os manuais da empresa possuem uma variedade de informações dos diversos assuntos tratados pela mesma, servindo como principal fonte na coleta de dados desta seção.

Em seguida, foi analisado o histórico de conversas desenvolvidas ao longo do tempo, entre funcionários e clientes via chat, este histórico possibilitou sintetizar de maneira mais coloquial diversas respostas.

### 4.2.3 Mapeamento dos Dados Cadastrados

Antes do processo de cadastramento, um fator importante a se considerar foi o controle sobre as sentenças cadastradas, com o crescimento da base de conhecimento, a qual pode conter milhares de sentenças, o gerenciamento se torna uma tarefa árdua a ser elaborada, pensando nisso, foi elaborado de maneira simples, diversas planilhas de mapeamento de dados. Os dados foram organizados entre mensagens do cliente e respostas do bot. Também foi definido que para cada resposta do cliente de uma pergunta específica do bot, teria uma resposta do *bot* (Figura 9). Todo conteúdo inserido na base AIML, obrigatoriamente fica registrado de acordo com a sua classe e finalidade, a utilização de planilhas permitiu que em meio a uma diversidade de dados, uma busca fosse efetuada.

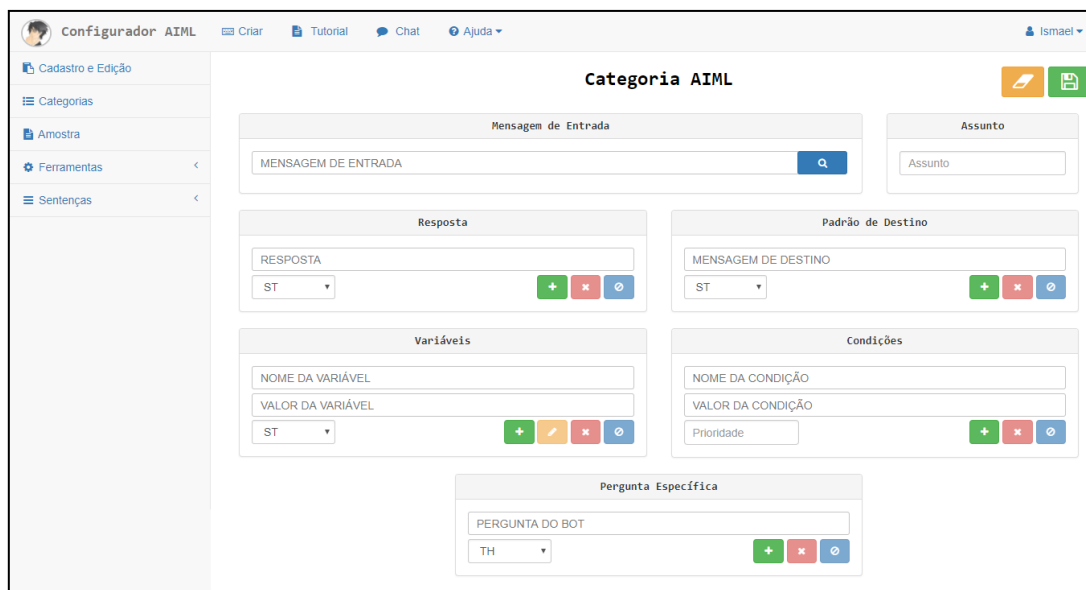


**Figura 9. Organização dos dados inseridos**

**Fonte: AUTORES, 2017.**

### 4.2.4 Cadastro e Gerenciamento da Base AIML

O cadastramento e Gerenciamento da base AIML deu-se através da plataforma Web de Gerenciamento (Figura 10), desenvolvida em PHP e hospedada em um servidor da empresa, possibilitando o acesso remoto de qualquer lugar.



**Figura 10. Plataforma Web**

**Fonte: AUTORES, 2017.**

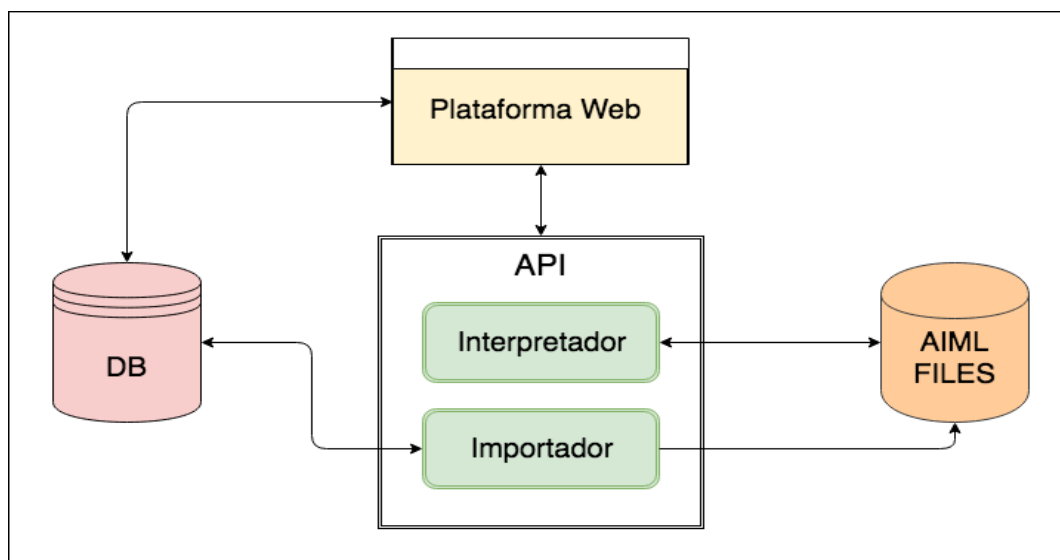
As funcionalidades da plataforma desenvolvida incluem:

- Cadastro/alteração/exclusão de categorias na base AIML;
- Cadastro/alteração/exclusão de ferramentas de controle (classes, assuntos, abreviaturas, palavras reservadas e variáveis);
- Listagem de categorias cadastradas;
- Visualização do código AIML;
- Tratamento de sentenças desconhecidas;
- Chat de interação para testes;

A estrutura da plataforma web (Figura 11) era composta por uma interface que se comunicava com o banco para gerenciar e armazenar os dados. Ela também contava com a ajuda de dois métodos: o importador e o interpretador. Ambos eram executados através de um RESTFUL API<sup>3</sup>, desenvolvida na linguagem Python, ou seja, a plataforma se comunicava com uma aplicação que era iniciada no servidor Linux e enviava a sua requisição. Desta forma o processo de armazenamento de sentenças cadastradas no Banco de Dados para os arquivos AIML acontecia via requisição HTTP, onde era chamado o método de importação, que consultava o Banco de Dados para a capturar, analisar e tratar dos dados, depois então criava os arquivos, conforme a sintaxe da linguagem AIML.

---

<sup>3</sup> Também conhecida como um serviço web RESTful - é baseada na tecnologia de transferência de estado de representação (REST), em um estilo arquitetônico e em uma abordagem às comunicações frequentemente utilizadas no desenvolvimento de serviços web



**Figura 11. Estrutura do Projeto**

**Fonte: AUTORES, 2017.**

A plataforma também disponibilizava ambientes de Chat para testes, que por sua vez se comunicava com o interpretador PyAIML, que traduzia o que o usuário havia mencionado ou escrito no Chat e consultava na sua base de conhecimento (arquivos AIML criados pelo importador) uma sentença equivalente, e retornava para plataforma o resultado da consulta em formato JSON (*JavaScript Object Notation*), uma formatação leve de troca de dados.

Com essa estrutura (Figura 11), foi possível obter mais segurança no que diz respeito ao armazenamento das informações e a possibilidade de remontar ou atualizar a base AIML de acordo com a necessidade.

#### **4.2.5 Procedimentos de Teste**

Afim de validar o ChatterBot Akira, foi necessário deixá-lo capacitado a resolver questionamentos referentes ao produto DGPhone, um softphone SIP<sup>4</sup> que permite realizar chamadas via sistema. Para efetuar os testes com os 12 colaboradores, foi disponibilizado uma lista de problemas e dúvidas (tabela 3). Para cada colaborador foi permitido realizar um total de 5 interações com o ChatterBot, totalizando 60 diálogos, escolhendo a cada interação um item diferente.

---

<sup>4</sup> A sigla SIP (*Session Initiation Protocol*) pode ser traduzida como um “Protocolo de iniciação de Sessão” com código aberto, cuja função é permitir a comunicação entre usuários através de uma rede VoIP (rede de conexão por voz com a internet).

**Tabela 3. Lista de problemas e dúvidas**

Problemas	Dúvidas
Não consigo realizar chamadas pelo DGPhone	O que significa
Não consegue registrar o ramal	Para que serve
Não consegue registrar o <i>login</i> do atendente	Como utilizar
Chamada apresenta eco	Como configurar
Chamada com muito barulho	Como realizar chamadas
Volume da chamada está alto	Como realizar transferência
Volume do <i>headset</i> está alto	Como realizar conferência
DGPhone parou de funcionar	Como colocar chamada em espera
DGPhone com erro após atualização	Como realizar mais de uma chamada simultânea
DGPhone com erro " <i>access violation at address</i> "	Como monitorar chamadas
DGPhone com erro	Como habilitar/desabilitar a opção siga-me
DGPhone não abre	Como habilitar/desabilitar a opção não perturbe
DGPhone com ruído	Como habilitar/desabilitar a opção auto <i>answer</i>
DGPhone está mudo	Como baixar o manual do DGPhone
	Como baixar o instalador do DGPhone
	Como abrir mais de uma instância do DGPhone
	Em qual sistema operacional roda o DGPhone
	Como configurar o DGPhone com <i>gateway</i>
	Como salvar um número para discagem rápida
	Como salvar os contatos

O ChatterBot Akira foi submetido a um segundo teste, onde foram extraídos do histórico de atendimento da empresa 20 diálogos, no qual o assunto abordado estava relacionado ao produto DGPhone. O teste foi realizado inserindo as mensagens do cliente na plataforma de *chat* do ChatterBot Akira, contabilizando o tempo de duração da interação.

#### 4.2.6 Análise dos Resultados

Os resultados dos testes com os colaboradores e dos testes efetuados com o histórico de atendimentos foram categorizados em “Resolveu” e “Resolveu parcialmente” (Tabela 4).

**Tabela 4. Categorias de resultados**

Categoria	Significado
(1). Resolveu	Atendimento solucionado
(2). Resolveu parcialmente	Atendimento transferido

Os diálogos categorizados como “resolveu parcialmente”, referem-se aos atendimentos que foram transferidos para um atendente humano, sendo que essa transferência pode ocorrer por dois motivos distintos, atendimentos transferidos para que houvesse um tratamento especializado e atendimentos transferidos por tentativas de comunicação esgotadas

Para o questionário pós teste, foram definidos três critérios de avaliação (Tabela 5).

**Tabela 5. Questionário Pós Testes - Critérios de Avaliação**

<b>Critérios</b>
(1) Quantidade de problemas/duvidas resolvidas
(2) A coerência das respostas
(3) Qualidade do dialogo

Para os testes no histórico de atendimentos, o tempo que o ChatterBot Akira levou para efetuar o atendimento foi comparado com o tempo que o atendente havia levado para finalizar o atendimento.

## **5. Resultados e Discussões**

No momento do teste o AIML possuía um total de 2.650 registros na base de conhecimento, dos quais, 12% são sentenças de abertura, 80% de desenvolvimento e 8% de fechamento. O questionário pós teste obteve os seguintes resultados (Tabela 6).

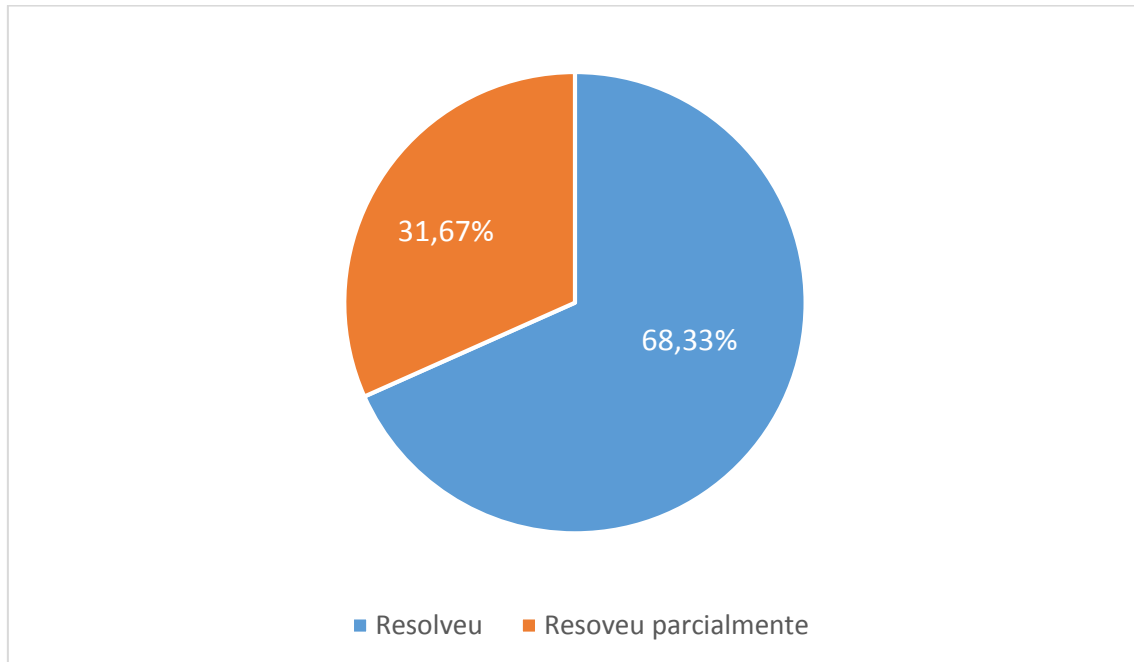
**Tabela 6. Resultados obtidos após o uso do ChatterBot Akira**

<b>Critérios de Avaliação</b>	<b>Todas</b>	<b>Grande parte</b>	<b>Metade</b>	<b>Poucas</b>	<b>Nenhuma</b>
(1) Quantidade de problemas/duvidas resolvidas	0%	42,9%	42,9%	14,3%	0%
	<b>Ótimo</b>	<b>Bom</b>	<b>Regular</b>	<b>Ruim</b>	
(2) A coerência das respostas	9,2%	38,2%	52,6%	0%	
(3) Qualidade do dialogo	0%	42,9%	52,1%	0%	

Analisando os resultados apresentados na Tabela 6, podemos identificar que o ChatterBot Akira, conseguiu resolver grande parte dos problemas e dúvidas. De acordo com o critério (1) não houve atendimento sem resolução dos problemas ou dúvidas, pois com o tratamento de sentenças desconhecidas, se o número de tentativas de compreensão se esgotassem, o *bot* redirecionava as conversas para um atendente. Podemos apontar de acordo com o critério (1) que grande parte das questões e metade das questões foram resolvidas com o índice de 42,9%.

Para os critérios (2) e (3) os resultados obtidos tiveram um índice de aprovação satisfatório, indicando que o tratamento de repetições e o controle da conversa a nível global e local foram bem-sucedidos, pois ao analisar o registro dos diálogos foi possível notar que o *bot* tentava retomar a conversa quando o usuário saia fora do contexto.

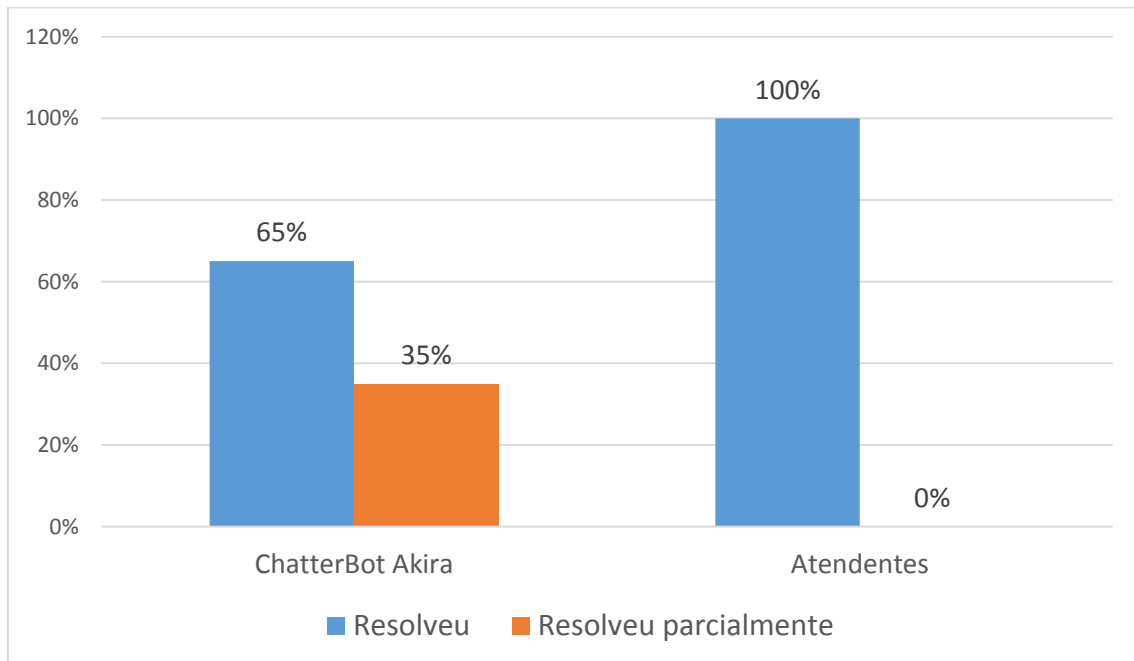
**Gráfico 1. Resultado dos atendimentos**



Analisando o Gráfico 1 nota-se que o ChatterBot Akira conseguiu alcançar um índice de 68,33% (Categoria 1), solucionando grande parte dos atendimentos realizados. Transferiu 31,67% (Categoria 2) dos chamados por tentativas esgotadas na compreensão das mensagens informadas pelo cliente.

O Gráfico 2 mostra um comparativo entre os atendimentos realizados pelo *bot* e os realizados pelos atendentes.

**Gráfico 2. Histórico - Resultados do Atendimento**

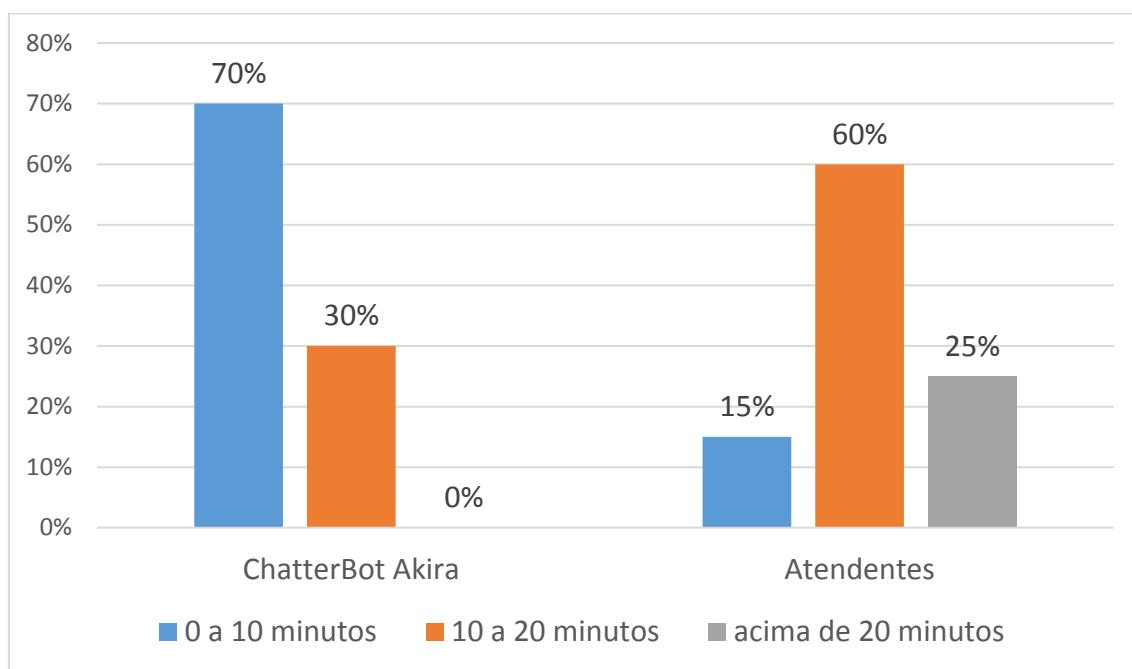


De acordo com gráfico 2, 65% dos atendimentos analisados, não precisariam de um atendente humano para solucioná-lo, apenas 35% necessitaram de um atendimento especializado. Consequentemente, os atendentes que realizam mais de uma

tarefa na empresa, não precisariam dispor de tempo para responder a uma simples dúvida ou resolver problemas triviais.

Outro resultado interessante, foi o tempo médio dos atendimentos submetidos a teste (Gráfico 3). Cerca de 30% dos chamados apurados pelo histórico, levou entre 10 e 20 minutos e os outros 70% menos de 10 minutos, sendo que na média dos atendentes 25% dos chamados levaram mais de 20 minutos

**Gráfico 3. Histórico – Tempo de Atendimento**



Outro ponto a se observar no histórico de atendimentos, foi o fato de que o cliente não possuía uma resposta imediata do atendente, gerando diversas mensagens do cliente antes de obter uma resposta, no entanto, quando o mesmo dialogo foi submetido ao ChatterBot Akira, cada mensagem era prontamente respondida, levando a uma resolução mais rapidamente.

## 6. Conclusão

Um dos diferenciais do projeto foi o desenvolvimento de uma plataforma Web de gerenciamento. Normalmente uma base de conhecimento AIML possui um número grande de sentenças, podendo atingir milhares de registros, quanto maior a quantidade de dados, mais árduo o processo de análise dos mesmos, porém, com o desenvolvimento da plataforma o processo manipulação dos dados tornou-se mais rápido e prático.

De acordo com os resultados obtidos, conclui-se que o ChatterBot Akira atendeu o seu objetivo, conseguindo realizar os atendimentos iniciais e transferindo os chamados nos casos em que a conversa exigiu um tratamento especializado. Os testes referentes à disponibilidade de tempo não foram efetuados, devido ao *bot* não obter uma base de conhecimento suficiente para testes com o cliente final, no entanto, com um êxito satisfatório nos testes realizados, supomos que sua desenvoltura não sofreria variação no caso de ser submetido a uma disponibilidade maior.

Como possíveis trabalhos futuros esperamos melhorar o mapeamento de dados feito em planilhas, migrando para a própria plataforma de gerenciamento, evitando



inconsistência de dados, e aperfeiçoar a plataforma de gerenciamento para inserir mais funcionalidades da linguagem AIML.

Por fim, observamos que o ChatterBot Akira se mostrou eficaz, não só para lidar com atendimentos em empresas de telemática, mas para os mais variados setores empresariais que buscam um atendimento inicial para seus clientes e uma interação natural de acordo com as qualidades do Akira.

## Referências

- BUSH, Noel. *Artificial Intelligence Markup Language (AIML) Version 1.0.1 A.L.I.C.E. AI Foundation Working Draft*. (rev 006), Outubro 2001. Disponível em: <<http://www.alicebot.org/TR/2001/WD-aiml/>>. Acessado em 25 de Maio de 2016.
- DEL GRANDE Informática e Telecomunicações LTDA. **Empresa**. <<http://www.delgrande.com.br/empresa.html>>. Acessado em 03 de Março de 2017
- GOMES, Helton Simões. **Especial Chatbots**. G1 Portal de Notícias da Globo. Acessado em Setembro de 2016. Disponível em <<http://especiais.g1.globo.com/tecnologia/2016/especial-chatbots/>> . Acessado em 29 de Abril de 2017.
- LAVEN, Simon. The Simon Laven Page. [S.l.: s.n.], 1996. Disponível em: <<http://www.simonlaven.com>>. Acessado em 21 março de 2016
- LEONHARDT, Michelle Denise. Et al. **Elektra, Um chatterbot para uso em ambiente educacional**. UFRGS. 2003
- MARCUSCHI, L. A. **Gêneros textuais emergentes no contexto da tecnologia digital**. In: MARCUSCHI, L. A.; XAVIER, A. C. Hipertexto e gêneros digitais. Rio de Janeiro: Editora Lucerna, 2004. p. 13-67.
- NEVES, André M. M. **iAIML: um mecanismo para o tratamento de intenção em Chatterbots**. 2005. 120 f. Tese (Doutorado em Ciência da Computação) – Pós-Graduação em Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife.
- PRADO, Ana Laura. **Chatbots vão dominar seu smartphone. Entenda por quê**. Revista Exame.com. Outubro de 2016. Disponível em: > Acessado em 29 de abril de 2017
- SCHEGLOFF 1972 Schegloff, E. A. **Sequencing in Conversational Openings, New York**. 1972. Disponível em:0<<http://www.sscnet.ucla.edu/soc/faculty/schegloff/>> Acessado em 28 de Outubro de 2016.
- TEIXEIRA, S.; MENEZES, C. S. de. **Facilitando o uso de ambientes virtuais através de agentes de conversação**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCACÃO, 14, 2003. Rio de Janeiro. Anais... Rio de Janeiro, 2003. p.483-492
- THE ZENDESK BENCHMARK, **In focus: Live Chat**, 2015. Disponível em <<https://d26a57ydsghvgx.cloudfront.net/content/resources/zendesk-benchmark-Q1-2015.pdf>>. Acessado em 22 de abril de 2016
- WALLACE, Richard S. **The Elements of AIML Style**. ALICE A. I. Foundation, Inc., 2003. **AIML: Artificial Intelligence Markup Language**. Disponível em: <http://www.alicebot.org/aiml.html>. Acesso em 09 de Maio de 2016.