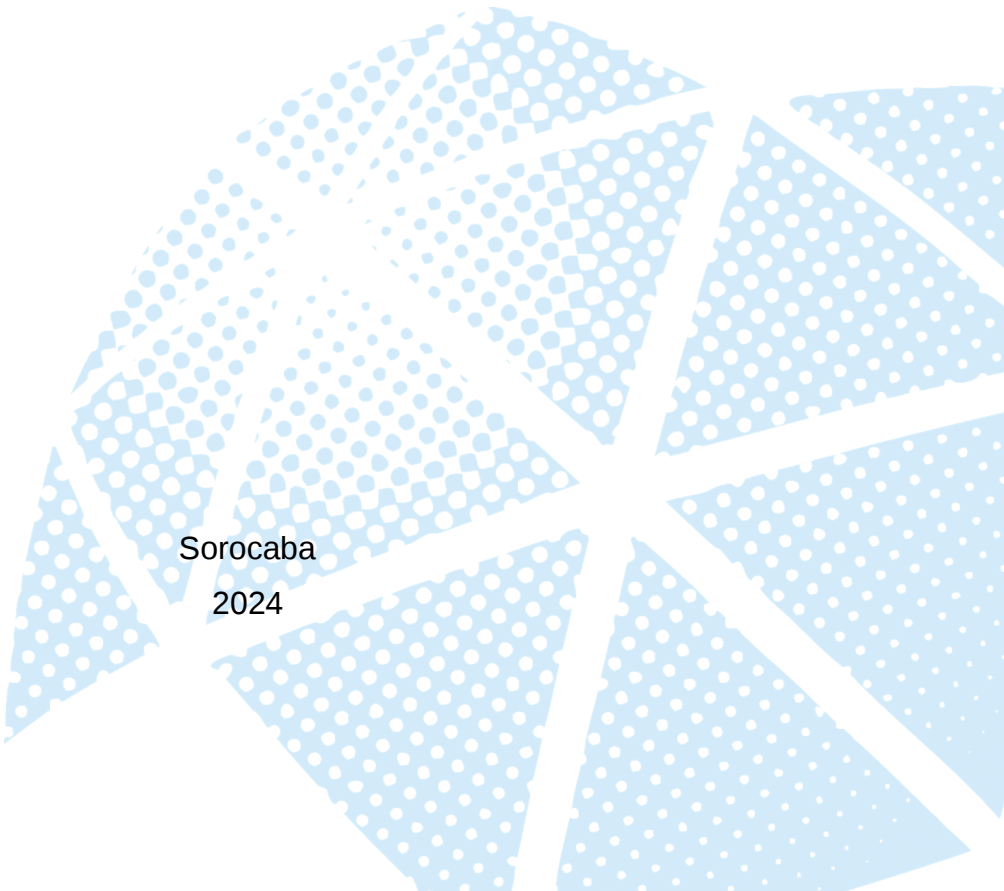


UNIVERSIDADE ESTADUAL PAULISTA – UNESP
Instituto de Ciência e Tecnologia de Sorocaba – UNESP Sorocaba

OTÁVIO DE ALMEIDA BIANCHINI

DESENVOLVIMENTO DE ASSISTENTE VIRTUAL PARA SUPORTE A
ASSISTÊNCIA TÉCNICA DE BOMBAS CENTRÍFUGAS
EM PROCESSOS INDUSTRIAIS

Sorocaba
2024



OTÁVIO DE ALMEIDA BIANCHINI

**DESENVOLVIMENTO DE ASSISTENTE VIRTUAL PARA SUPORTE A
ASSISTÊNCIA TÉCNICA DE BOMBAS CENTRÍFUGAS
EM PROCESSOS INDUSTRIAIS**

Projeto Final de Curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista (UNESP), como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientadora: Prof^a. Dr^a. Marilza Antunes de Lemos

Sorocaba

2024

B577d	Bianchini, Otavio de Almeida Desenvolvimento de assistente virtual para suporte a assistência técnica de bombas centrífugas em processos industriais / Otavio de Almeida Bianchini. -- Sorocaba, 2024 134 p. : il., tabs. Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba Orientadora: Marilza Antunes de Lemos 1. Agentes inteligentes (Software). 2. Manutenção. 3. Bombas centrífugas. 4. Assistência técnica. I. Título.
-------	--

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba.
 Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

OTÁVIO DE ALMEIDA BIANCHINI

**DESENVOLVIMENTO DE ASSISTENTE VIRTUAL PARA SUPORTE A
ASSISTÊNCIA TÉCNICA DE BOMBAS CENTRÍFUGAS
EM PROCESSOS INDUSTRIAIS**

Projeto Final de Curso apresentado à Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba, para obtenção do título de Bacharel em Engenharia de Controle e Automação

Data da defesa: 10 / 04 / 2024

Banca Examinadora:

Prof.^a Dr.^a Marilza Antunes de Lemos
UNESP - Instituto de Ciência e Tecnologia – Campus de Sorocaba

Prof. Dr. Márcio Alexandre Marques
UNESP - Instituto de Ciência e Tecnologia – Campus de Sorocaba

Prof. Dr. Eduardo Verri Liberado
UNESP - Instituto de Ciência e Tecnologia – Campus de Sorocaba

RESUMO

O campo da pesquisa em diagnóstico automático de falhas baseado em técnicas de Inteligência Artificial remonta há algumas décadas. Em anos mais recentes, os métodos da Inteligência Artificial, particularmente Machine Learning, tem contribuído para que sistemas preditivos alcancem bons resultados prevendo quando determinado elemento industrial poderá falhar. Apesar dos avanços em diagnóstico e predição de falhas automáticos, nenhum desses métodos eliminam a necessidade de um técnico complementar o trabalho de manutenção, seja pela substituição de componentes na planta industrial, manutenção preventiva ou outras ações humanas de manutenção. Um processo eficaz de manutenção pode requerer uma equipe técnica bem treinada, processos de manutenção bem estabelecidos e um conjunto de sistemas que auxiliem esta equipe. Atualmente as aplicações Chatbots ou assistentes virtuais vem obtendo sucesso em interações com humanos em diversas áreas. O presente trabalho explora esse recurso e apresenta o desenvolvimento de um assistente virtual do tipo Chatbot para dar suporte à equipe técnica de manutenção de bombas centrífugas em um contexto industrial. A aplicação visa auxiliar sobre especificidades desses dispositivos, bem como em operações de retirada e instalação dos equipamentos. O desenvolvimento foi realizado com a ferramenta Chatterbot na linguagem Python. Conhecimento técnico especializado foi modelado e embutido na aplicação. Recursos de Machine Learning em Chatterbot foram aplicados para treinamento e refinamento das respostas ao usuário. Testes preliminares com banco de dados reduzido indicaram que o chatbot E-MANUEL obteve uma taxa de sucesso de 89,2%, e apenas em 7,7% das interações não foi capaz de gerar resposta ao usuário dentro do contexto esperado.

Palavras-chave: assistente virtual; chatbot; sistema inteligente; machine learning; Inteligência Artificial; bombas centrífugas; manutenção.

ABSTRACT

The field of research into automatic fault diagnosis based on Artificial Intelligence techniques dates back a few decades. In more recent years, Artificial Intelligence methods, particularly Machine Learning, have contributed to predictive systems achieving good results by predicting when a certain industrial element may fail. Despite advances in automatic diagnosis and failure prediction, none of these methods eliminate the need for a technician to complement maintenance work, whether by replacing components in the industrial plant or other human maintenance actions. An effective maintenance process may require a well-trained technical team, well-established maintenance processes and a set of systems that assist this team. Currently, Chatbots or virtual assistant applications have been successful in interactions with humans in several areas. The present work explores this resource and presents the development of a Chatbot-type virtual assistant to support the technical team maintaining centrifugal pumps in an industrial context. The application aims to assist with the specificities of these devices, as well as in field removal and installation operations. The development was carried out with the Chatterbot tool in the Python language. Specialized technical knowledge was modeled and embedded in the application. Machine Learning resources in Chatterbot were applied to train and refine user responses. Preliminary tests with a reduced database indicated that the E-MANUEL chatbot achieved a success rate of 89.2%, and in only 7.7% of interactions it was unable to provide a response to the user within the expected context.

Keywords: virtual assistant; chatbot; intelligent system; machine learning; Artificial Intelligence; centrifugal pumps; maintenance.

SUMÁRIO

1	INTRODUÇÃO.....	8
2	REVISÃO TEÓRICA.....	10
2.1	ARQUITETURA DE <i>CHATBOTS</i>	11
2.2	CHATTERBOT	13
2.3	BOMBAS CENTRÍFUGAS	18
2.4	PRINCIPAIS MODOS DE FALHA DE BOMBAS CENTRÍFUGAS	19
2.4.1	Cavitação	19
2.4.2	Falha de Lubrificação	20
2.4.3	Falha de Rolamento	20
2.4.4	Falha no Selo	21
2.4.5	Desbalanceamento e Desalinhamento	21
2.5	EVOLUÇÃO DOS CONCEITOS DE MANUTENÇÃO	22
2.5.1	Manutenção Corretiva.....	22
2.5.2	Manutenção Preventiva	23
2.5.3	Manutenção Preditiva	24
3	MATERIAIS E MÉTODOS.....	25
3.1	ESPECIFICAÇÕES TÉCNICAS DA BOMBA KSB-ETA	25
3.2	PROCESSOS DE MANUTENÇÃO DA BOMBA CENTRÍFUGA KSB-ETA.....	28
3.3	BANCO DE DADOS DA EMPRESA	30
3.4	INSTALAÇÃO E ESTUDO DA FERRAMENTA CHATTERBOT	31
3.5	MODELAGEM DA INTERAÇÃO HUMANO-MÁQUINA	34
3.5.1	Modelagem do Procedimento de Busca de Dados Técnicos.....	36
3.5.2	Modelagem do Processo de <i>Checklist</i>	37
3.5.3	Modelagem do Guia de Ações de Manutenção	39
3.5.4	Modelagem do Guia de Falhas	39
3.6	IMPLEMENTAÇÃO DO <i>CHATBOT</i> E-MANUEL	40
3.6.1	Implementação da busca de dados técnicos para o usuário	41
3.6.2	Implementação do Adaptador Lógico <i>checklist</i>	47
3.7	TREINAMENTO DO <i>CHATBOT</i> EM AÇÕES DE MANUTENÇÃO	49
3.8	METODOLOGIA DE TESTES	51
4	RESULTADOS E DISCUSSÕES	57
5	CONCLUSÃO	65
	REFERÊNCIAS	67

APÊNDICE A – CÓDIGO FONTE DA INICIALIZAÇÃO DO CHATBOT	71
APÊNDICE B – CÓDIGO FONTE DA INTERAÇÃO COM O USUÁRIO ...	72
APÊNDICE C – CÓDIGO FONTE DO PRÉPROCESSADOR.....	72
APÊNDICE D – CÓDIGO FONTE DE FUNÇÕES UTILITÁRIAS.....	72
APÊNDICE E – CÓDIGO FONTE DO ADAPTADOR LÓGICO <i>CHECKLIST</i>	73
APÊNDICE F – CÓDIGO FONTE DE BUSCADADOSLUB.....	74
APÊNDICE G – CÓDIGO FONTE DE BUSCADADOSTEC	75
APÊNDICE H – CÓDIGO FONTE DO TREINO DE CHATBOT	76
APÊNDICE I – CÓDIGO FONTE DO ARQUIVO TESTES.PY	85
APÊNDICE J – ENTRADAS E SAÍDAS ESPERADAS DOS TESTES	96
APÊNDICE K – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA AB123456	99
APÊNDICE L – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA ET5678.....	101
APÊNDICE M – RELATÓRIO DO UNITTEST PARA A BOMBA GA2005	104
APÊNDICE N – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA JB1722.....	107
APÊNDICE O – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA MD1120.....	110
APÊNDICE P – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA NM8379.....	113
APÊNDICE Q – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA PT1322.....	116
APÊNDICE R – Relatório de resultado do unittest para a bomba tc2709	119
APÊNDICE S – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA TT2332.....	122
APÊNDICE T – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA XZ0987.....	125
APÊNDICE U – CODIGO FONTE DE TESTETEMPO.PY	128
APÊNDICE V – GUIA DE INSTALAÇÃO E CONFIGURAÇÃO DO E- MANUEL E DAS FERRAMENTAS DE DESENVOLVIMENTO	129

1 INTRODUÇÃO

Devido aos constantes avanços na automação dos processos industriais e o contínuo aumento no número de equipamentos e dispositivos nas linhas de produção industriais, a atenção tem se voltado cada vez mais para as questões que envolvem a manutenção das plantas industriais.

Nos primórdios da criação de equipes de manutenção, em torno de 1930, a ideia que se tinha de manutenção era uma atuação apenas no momento da falha e da quebra do equipamento, conhecida como manutenção corretiva. Porém, com o crescimento do processo industrial e o aumento da complexidade da produção esse processo se tornou cada vez mais ineficiente. Esses resultados conduziram o processo de manutenção para a realização de ações periódicas (preventiva) ou por meio da observação constante do equipamento (preditiva), evitando-se a ocorrência das falhas. Desta forma, a manutenção passou a ganhar espaço no processo produtivo, e a vida útil dos equipamentos se tornou uma preocupação crescente (SILVA *et al.*, 2017).

Atualmente, o planejamento e execução das ações de manutenção são vistos como parte do processo produtivo ao garantir a confiabilidade nos processos da empresa, e esta confiabilidade se traduz em competitividade no mercado (CARELLI; MACIEL, 2006; MENGUE; SELITO, 2013). Porém, não basta somente o investimento bruto para bons resultados em manutenção, também é necessária a aplicação de técnicas de manutenção modernizadas. De fato, as perdas por manutenção inadequada ou excessiva podem chegar a um terço do valor investido (CARELLI; MACIEL, 2006), sendo a maior parcela de perdas em manutenção relacionada a falhas de planejamento e problemas de execução (MENGUE; SELITO, 2013).

Um estudo realizado por Silva *et al.* (2017) mostra a aplicação e os impactos do processo de manutenção utilizando uma bomba centrífuga Agrimec, modelo BCI 300, aplicada ao contexto industrial de transporte de água. O trabalho evidencia como a devida manutenção reduz muitas perdas para a empresa, sendo a financeira causada pela parada repentina na produção, a pessoal causada por acidentes de fábrica e a ambiental resultante dos contaminantes gerados e perdidos no meio ambiente.

McKee *et al.* (2011) realizou estudo sobre os diferentes modos de falhas que podem ocorrer em uma bomba centrífuga, os principais sintomas e causas, bem como algumas formas de combatê-los. O autor destaca a presença de bombas modernas que já possuem sensores acoplados, permitindo a obtenção de informações em tempo real das condições de operação da bomba, porém, muitos operadores não são capazes de interpretar tais informações, sugerindo o uso de um sistema inteligente que utilize métodos de análise de dados e Inteligência Artificial para auxiliar no diagnóstico do estado das bombas.

O campo da pesquisa em diagnóstico automático de falhas com apoio de técnicas de Inteligência Artificial remonta há algumas décadas (ZHU *et al.*, 2023; HE; HE, 2017; HOANG; KANG, 2019). Em anos mais recentes a pesquisa avançou no desenvolvimento de sistemas preditivos, que analisam históricos de dados obtidos por sensoramento e conseguem prever falhas de elementos de automação, como motores rotativos, entre outros. Os métodos da Inteligência Artificial, particularmente *Machine Learning*, tem contribuído para que tais sistemas preditivos alcancem bons resultados apontando quando determinado elemento poderá falhar, antes que de fato isto ocorra (CARVALHO *et al.*, 2019; SUSTO *et al.*, 2014; THEISSLER *et al.*, 2021). Porém, apesar de todos esses avanços em diagnóstico e predição de falhas automáticos, nenhum desses métodos eliminam a necessidade de um técnico complementar ao trabalho de manutenção, seja pela substituição de um componente na planta industrial ou de outras ações humanas de manutenção. Um processo eficaz de manutenção pode requerer uma equipe técnica bem treinada, processos de manutenção bem estabelecidos e um conjunto de sistemas que auxiliam esta equipe. Atualmente as aplicações de Chatbots ou assistentes virtuais vem obtendo sucesso em interações com humanos em diversas áreas, incluindo a área de educação e treinamento (ALHASSAN *et al.*, 2022; CHANG; HWANG; GAU, 2022; NEUMANN *et al.*, 2021).

O presente trabalho tem por objetivo o desenvolvimento de um assistente virtual (Chatbot) para dar suporte à equipe técnica de manutenção de bombas centrífugas em um contexto industrial. O chatbot visa auxiliar sobre especificidades de bombas centrífugas, suporte em processos e operações de manutenção como retirada e instalação do equipamento em campo e checklist nos dispositivos.

2 REVISÃO TEÓRICA

Chatbot, ou assistente virtual, é o termo comumente utilizado para descrever um *software*, que tem por objetivo simular um diálogo com outro ser humano de forma natural, por meio de troca de mensagens ou por voz (HUSSAIN *et al.*, 2023). Atualmente, estamos presenciando uma sociedade cada vez mais digitalizada, o que, consequentemente, se faz necessária a utilização do *chatbot*. No dia a dia, presenciamos a sua manifestação em várias atividades, como assistentes pessoais em *smartphones*, *tablets* e *notebooks* e terapia de auto ajuda (SERBAN *et al.*, 2017). Esse cenário resulta da capacidade desses programas em reduzirem os custos de serviços, e serem capazes de lidar com vários clientes simultaneamente. Outra característica que contribui para a popularização desses assistentes virtuais é o seu design mais amigável ao usuário, simulando uma conversa com outro ser humano, disponibilizando a informação de forma mais intuitiva e direta. Dessa maneira, facilita a experiência do consumidor, para que não precise buscar manualmente uma dúvida em uma FAQ (*Frequently Asked Questions* ou “Perguntas Frequentes” em português) (ADAMOPOULOU; MOUSSIADES, 2020a), por exemplo.

Os *chatbots* apresentam diversas características construtivas que mudam de acordo com a aplicação e complexidade. Os principais parâmetros utilizados para classificar os diferentes tipos de *chatbots* são: domínio, serviço, objetivos, a geração de resposta, ajuda humana e o método de construção (ADAMOPOULOU; MOUSSIADES, 2020b).

Com relação a **geração de respostas** os *chatbots* são separados em *Rule-based*, baseados em recuperação e generativo (ADAMOPOULOU; MOUSSIADES, 2020a). Os primeiros programas considerados *chatbots* se baseavam na geração de resposta por meio do método **Rule-based**. Esse método consiste em relacionar padrões (*Patterns*) de entradas possíveis a serem fornecidas pelo usuário com modelos (*Templates*) armazenados em seu banco de dados (CAHN, 2017). Os *chatbots* **baseados em recuperação** são capazes de interpretar a entrada de usuário sem que esta precise se adequar a um padrão. Esse método utiliza a técnica *Deep Learning* da Inteligência Artificial para identificar palavras-chave no texto e fazer a correspondência com uma base de dados com respostas pré-definidas (ADAMOPOULOU; MOUSSIADES, 2020b). Os *chatbots* generativos além de interpretarem entradas sem que estas precisem se enquadrar em um padrão são

capazes de gerar as próprias respostas além de respostas pré-definidas em seu sistema (CAHN, 2017).

Desse modo, a sua interação pode ser caracterizada de acordo com o **domínio** da conversa, podendo ser um domínio aberto, que permita interação sobre qualquer assunto, ou domínio fechado, onde o assistente é voltado para interação sobre determinados assuntos e pelo tipo de resposta, podendo ser respostas vindas de base de dados ou respostas geradas pelo *software* (ALHASSAN *et al.*, 2022).

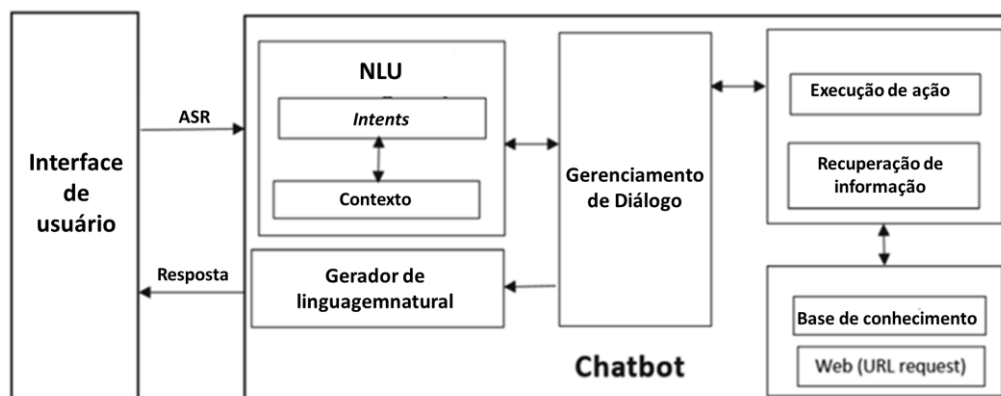
No caso da classificação por **ajuda humana**, grande parte dos *chatbots* atuam de forma completamente autônoma, porém algumas funções exigem uma flexibilidade maior, tornando uma ação humana necessária. Essa intervenção classifica o *chatbot* como auxiliado por humano (ADAMOPOULOU; MOUSSIADES, 2020b).

Por fim, a última forma de classificação se refere à **plataforma de desenvolvimento**, sendo classificado como *open-source* ou fechado/comercial (ADAMOPOULOU; MOUSSIADES, 2020a).

2.1 ARQUITETURA DE CHATBOTS

As etapas de desenvolvimento do *chatbot* variam de acordo com seu propósito, funcionalidades e técnicas usadas. Porém, de modo geral, existe uma arquitetura e um conjunto de conceitos comuns a essas aplicações (ALHASSAN *et al.*, 2022), como apresentado na Figura 1.

Figura 1 - Arquitetura padrão de chatbot



Fonte: Adamopoulou e Moussiades (2020b)

Alguns *chatbots* permitem a interação com o usuário por meio da fala, dispensando a necessidade de escrita e tornando mais forte a sensação de interação com outro ser humano. Para realizar a interação por meio de sons é utilizado o sistema de Reconhecimento Automático de Fala ou **ASR** (*Automatic Speech Recognition*). O ASR é responsável pela captação da fala humana e pela sua transição para palavras compreendidas pelo sistema (CAHN, 2017).

Captada a entrada do usuário, a próxima etapa envolve a compreensão da linguagem natural através do **NLU** (*Natural Language Understand*) (ALHASSAN *et al.*, 2022). O NLU é responsável pela compreensão do texto, captando o contexto (NEUMANN *et al.*, 2021); por armazenar o objeto e o assunto da conversa, *extraíndo* as *intents* (intenções); pelos elementos do texto que serão usados como base para a busca da resposta que deve ser fornecida pelo *chatbot* (ADAMOPOULOU; MOUSSIADES, 2020b). Dentro da entrada de usuário também podem ser extraídos as *entities* (entidades) (NEUMANN *et al.*, 2021), parâmetros como horário ou localidade que podem ser passados no texto (ADAMOPOULOU; MOUSSIADES, 2020b).

O componente de **Gerenciamento de Diálogo** é responsável por armazenar e atualizar os contextos e intenções da conversa (ADAMOPOULOU; MOUSSIADES, 2020b). O **gerador de linguagem natural** apresenta uma função paralela ao NLU, buscando uma compreensão da linguagem humana, realizando a melhor conversão possível de linguagem de máquina para a linguagem humana. Porém, neste caso, a intenção é gerar um texto compreensível para o usuário e que soe natural (ALHASSAN *et al.*, 2022). Por fim, a arquitetura inclui um programa voltado em transformar a resposta do assistente virtual em um áudio que soe natural para a percepção humana.

Atualmente, o aprendizado de máquina é a subárea de conhecimento mais pesquisada e aplicada na resolução de problemas dentro da grande área de conhecimento em Inteligência Artificial. Trata-se de um método iterativo de aprendizagem baseado na coleta, análise e interpretação de dados de forma automática, utilizando algoritmos. Para o assistente virtual isso pode se tornar uma ferramenta poderosa em diferentes etapas para interação com o usuário, de modo que a comunicação humana seja mais bem compreendida e as respostas pareçam mais naturais enquanto atendem ao seu objetivo.

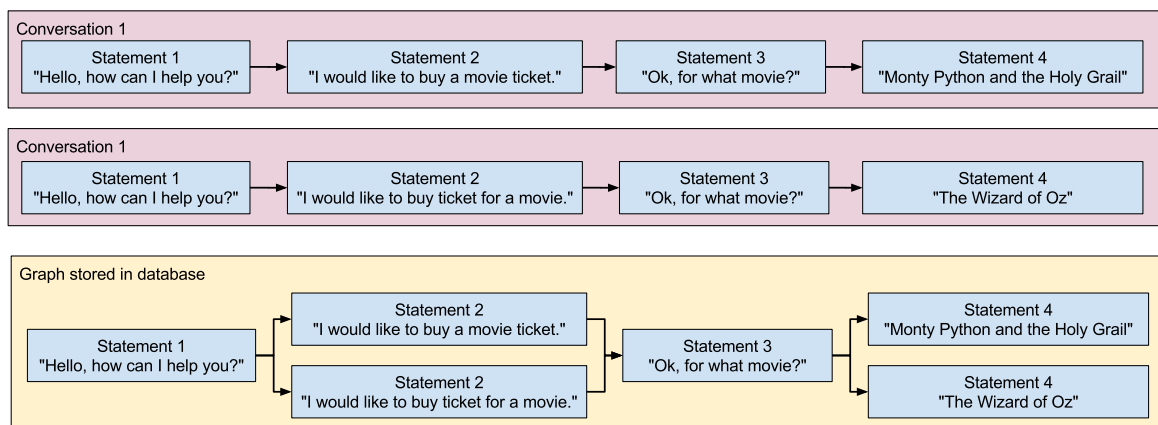
2.2 CHATTERBOT

Chatterbot (GUNTHER COX REVISION, c2021a) é uma ferramenta em Python que facilita a interpretação e geração de respostas automatizadas para um usuário. Para produzir diferentes tipos de respostas, ele utiliza uma seleção de algoritmos, facilitando a criação de bots de bate-papo e automação de conversas com usuários. O design do Chatterbot permite o treinamento do assistente virtual para dialogar com qualquer idioma. Além disso, a natureza de aprendizado de máquina do Chatterbot permite que uma aplicação *chatbot* melhore seu próprio conhecimento de possíveis respostas à medida que interage com humanos e outras fontes de dados informativos.

Uma instância não treinada do Chatterbot começa sem saber como se comunicar. Cada vez que um usuário insere uma instrução, a biblioteca salva o texto inserido e o texto ao qual a instrução foi respondida. À medida que o Chatterbot recebe mais entradas, o número de respostas, e a precisão de cada uma em relação à entrada, aumentam. O Chatterbot busca pela equivalência mais próxima a entrada do usuário, para que assim possa relacioná-lo com a respectiva resposta.

O Chatterbot inclui ferramentas que simplificam o processo de treinamento de bate-papo, o qual envolve o uso de uma caixa de diálogo previamente preparada para ser usada de exemplo pelo Chatterbot. A Figura 2, mostra dois diálogos exemplos (em rosa) e o grafo resultante (em amarelo) após o processo de treinamento. Esse processo de treinamento cria e complementa uma estrutura de dados interna de declarações e respostas, utilizadas para melhor responder o usuário.

Figura 2 - Estrutura de declarações e respostas



Fonte: Gunther Cox Revision (c2021b)

Várias possibilidades de treinamento vêm integradas ao Chatterbot. Esses utilitários variam desde permitir que se atualize o grafo de conhecimento do banco de dados do *Chatbot*, com base em uma lista de declarações que representam uma conversa, até ferramentas que permitem que se treine o *Chatbot*, com base em um corpo de dados de treinamento previamente carregados. É possível também a criação da própria aula de treinamento. Isso é recomendado quando se deseja treinar o Bot com dados armazenados em um formato que ainda não é compatível com uma das classes pré-criadas. A biblioteca Chatterbot trabalha com um conjunto de conceitos como se segue:

Adaptadores - é uma classe conectável que permite que uma instância do *Chatbot* execute algum tipo de funcionalidade.

Adaptador lógico – é uma classe de adaptador que permite que uma instância do *Chatbot* selecione uma resposta.

Adaptador de armazenamento – é uma classe que permite que um *chatbot* armazene informações em algum lugar, como por exemplo um banco de dados.

Corpus – é um conjunto grande e estruturado de textos. Eles são usados para fazer análises estatísticas e testes de hipóteses, verificando ocorrências ou validando regras linguísticas dentro de um escopo linguístico específico.

Pré-processadores – é um membro de uma lista de funções que podem ser usadas para modificar a entrada de texto que o *chatbot* recebe antes que o texto seja passado para o adaptador lógico para processamento.

Declaração – é uma única *string* de texto representando algo que pode ser dito.

Palavra de pesquisa – é uma palavra que não é uma palavra de parada e foi cortada de alguma forma.

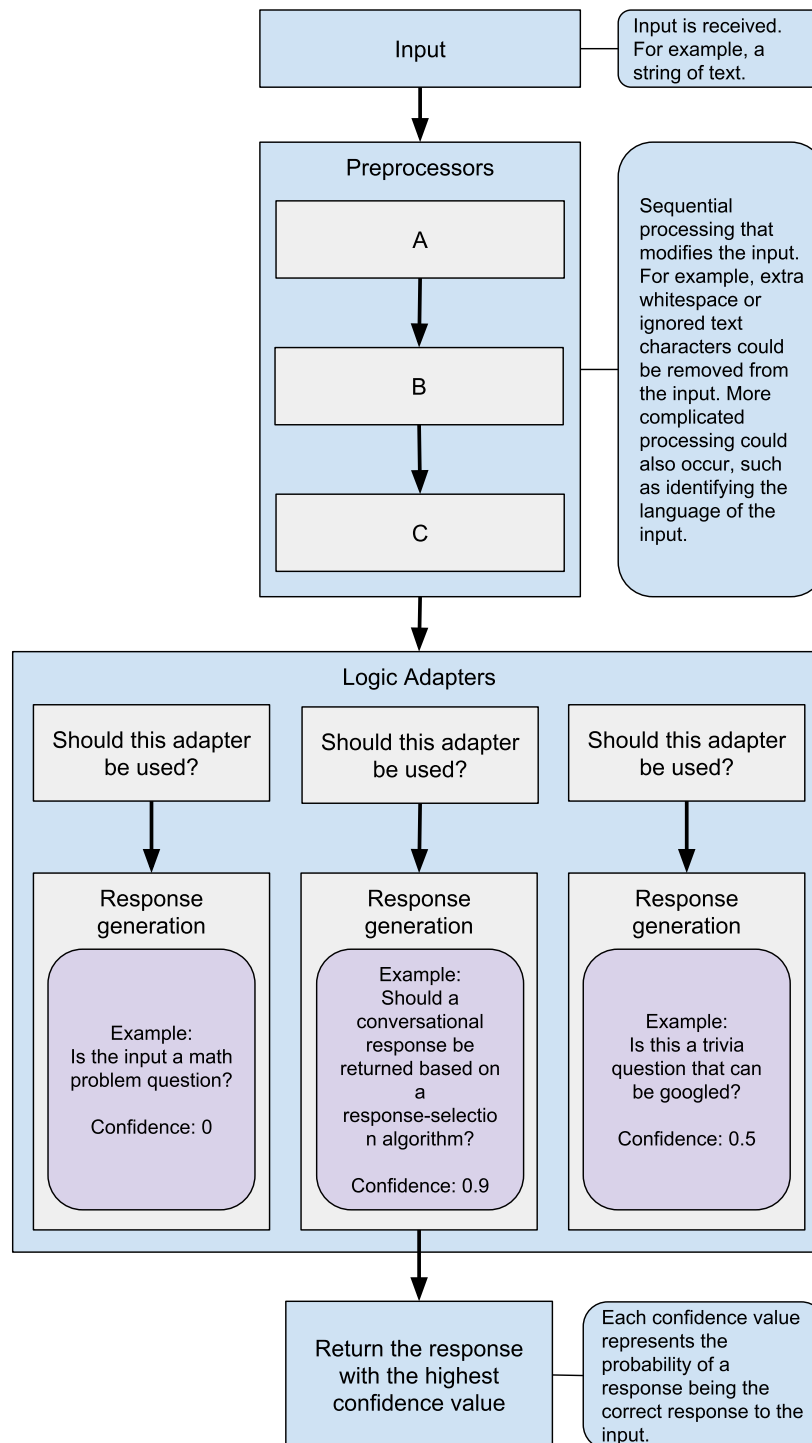
Decorrente – é um processo através do qual uma palavra é reduzida a uma forma derivada.

Palavra de parada – é uma palavra comum que geralmente é filtrada durante o processo de análise de texto.

Resposta – é uma única sequência de texto que é pronunciada como uma resposta ou um reconhecimento a uma declaração.

A biblioteca se apresenta como uma ferramenta para a criação de um *chatbot* funcional capaz de receber uma entrada, interpretá-la e buscar pela resposta que melhor se adapta para aquela entrada. A estrutura base do Chatterbot é ilustrada na Figura 3.

Figura 3 - Estrutura padrão do Chatterbot



Fonte: Gunther Cox Revision (c2021c)

De acordo com a Figura 3, Chatterbot recebe a declaração de entrada, transfere a informação para os pré-processadores a fim de limpar (remoção de espaços), extrair, organizar a informação. Na sequência, os adaptadores lógicos são utilizados, onde cada um retorna um número como resposta, que representa a probabilidade de o mesmo responder corretamente ao usuário.

Ao receber as sentenças com os seus respectivos valores de confiança, Chatterbot seleciona a resposta mais frequentemente retornada pelos adaptadores ou, caso todas sejam distintas, retorna aquela com maior grau de confiança.

A implementação de um *chatbot* parte da criação do arquivo denominado *chatbot.py*. Em seguida, o *Chatbot* precisa ser importado (Algoritmo 1).

Algoritmo 1 - Criação de chatbot com Chatterbot

```

From Chatterbot import Chatbot //Cria uma nova instância da classe Chatbot.

bot = Chatbot(
    'Norman',
    storage_adapter='Chatterbot.storage.SQLStorageAdapter',
    database_uri='sqlite:///database.sqlite3'
)                                     // cria o novo chatbot chamado Normam

```

O Chatterbot contém classes integradas denominadas **adaptadores de armazenamento**, permitindo que ele se conecte a diferentes tipos de bancos de dados. No exemplo de código acima, a classe *SQLStorageAdapter* permite que o *chatbot* se conecte a bancos de dados SQL. A classe *database* parâmetro é usada para especificar o caminho para o banco de dados que o *chatbot* usará.

No Algoritmo 2, o parâmetro *logic_adapters* é uma lista de **adaptadores lógicos**. No Chatterbot, um **adaptador lógico** é uma classe que recebe uma instrução de entrada e retorna uma resposta a essa instrução. O exemplo apresentado abaixo, utiliza dois adaptadores lógicos: o *TimeLogicAdapter* retorna a hora atual quando a instrução de entrada solicita; o adaptador *MathematicalEvaluation* resolve problemas matemáticos que usam operações básicas.

Algoritmo 2 - Criação de *chatbot* com definição de adaptadores lógicos

```

Bot = Chatbot(
    'Norman',
    storage_adapter='Chatterbot.storage.SQLStorageAdapter',
    logic_adapters=[
        'Chatterbot.logic.MathematicalEvaluation',
        'Chatterbot.logic.TimeLogicAdapter'
    ],
    database_uri='sqlite:///database.sqlite3'
)

```

O trecho de código do Algoritmo 3 apresenta uma conversa em execução com o *chatbot* Norman.

Algoritmo 3 - Sistema de troca de mensagens entre o *chatbot* e o usuário

```

While True:
    try:
        bot_input = bot.get_response(input())
        print(bot_input)
    except (KeyboardInterrupt, EOFError, SystemExit):
        break

```

Neste ponto, o *chatbot* Norman aprenderá a se comunicar enquanto o usuário fala com ele. Ressalta-se que a instrução **break** no código gerado automaticamente por Chatterbot trata três exceções que finalizam o programa. Por exemplo, quando o usuário pressionar ctrl – c gerará automaticamente a exceção *KeyboardInterrupt* (PYTHON SOFTWARE FOUNDATION, 2024). O uso de **break**, nesse caso, não gera risco de *stackoverflow* (estouro da pilha), uma vez que o tratamento da exceção finaliza o programa.

O processo de diálogo entre usuário e Chatbot pode ser melhorado treinando-o com exemplos de conversas existentes. No algoritmo 4 é realizada a importação das funções de *ListTrainer* presentes na ferramenta Chatterbot, então é criada uma instância de *ListTrainer* utilizando como parâmetro o *chatbot* Norman. Um diálogo exemplo é criado por meio de uma caixa de diálogo e usado como parâmetro para a função *train*, a função do Chatterbot responsável por reger o treino do *chatbot*.

O processo de treinamento pode ser executado várias vezes para reforçar as respostas preferenciais a determinadas declarações de entrada. Outra

possibilidade é executar o comando *train* em várias caixas de diálogo para aumentar a amplitude de entradas às quais o *chatbot* pode responder.

Algoritmo 4 - Algoritmo de treinamento da ferramenta Chatterbot

```

From Chatterbot.trainers import ListTrainer

trainer = ListTrainer(bot)

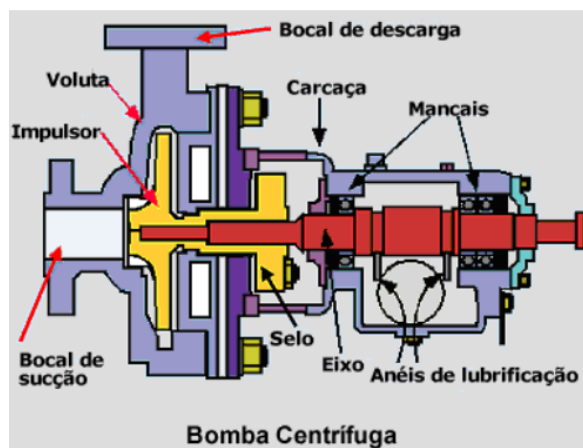
trainer.train([
    'How are you?',
    'I am good.',
    'That is good to hear.',
    'Thank you',
    'You are welcome.'
])

```

2.3 BOMBAS CENTRÍFUGAS

Bombas são turbo máquinas responsáveis por gerar em um fluido regiões de baixa e alta pressão, podendo realizar essa função das mais diversas formas. A bomba mais utilizada no meio industrial é a bomba centrífuga (Figura 4), devido a sua versatilidade (THIN; KHAING; AYE, 2008). A Figura 4 apresenta a estrutura básica de uma bomba centrífuga e suas partes ou componentes, que compõem um conjunto rotativo que se utiliza da força centrífuga para transferir energia para o fluido que percorre o seu interior. Dentro da bomba, seu movimento é ocasionado devido a um motor (geralmente elétrico) que se acopla ao eixo para transmitir uma energia mecânica para o rotor, responsável por transferir a energia para o fluído (SILVA *et al.*, 2017).

Figura 4 - Diagrama de uma bomba centrífuga



Fonte: Silva (2017)

Em análise mais profunda do seu funcionamento, o fluído é levado ao Bocal de sucção, entra no centro do rotor ou impulsor que gira devido ao motor. Assim, o rotor possui pás que arrastam o produto em movimento rotacional, fazendo com que se movimente do centro para as 'bordas', onde é expelido pelo bocal de descarga (SANTOS FILHO, 2018). Os mancais são responsáveis pela lubrificação dos eixos, sendo indispensável que o mesmo esteja com o nível e qualidade de óleo recomendável, e sua troca precisa ocorrer periodicamente. Outra peça importante é o selo, que evita o vazamento do fluído para o mancal, impedindo, assim, a contaminação do óleo (SILVA *et al.*, 2017). Outras preocupações comuns a bomba são a atenção a uma boa fixação, com parafusos bem apertados, presença de vazamentos e ferrugem, situações que podem ser percebidas por inspeção visual (MENGUE; SELITO, 2013).

2.4 PRINCIPAIS MODOS DE FALHA DE BOMBAS CENTRÍFUGAS

2.4.1 Cavitação

Dentro da indústria química, uma preocupação comum em processos de manejo e transporte de produtos líquidos é a formação de bolhas nas tubulações e nas máquinas que o compõem. Esse fenômeno é a cavitação, e ocorre devido a combinação de dois fatores: 1. Processos químicos que elevam a temperatura dos fluídos e materiais tratados; 2. Criação de regiões de baixa pressão causados pela sucção da bomba. Essa combinação de elevação de temperatura e redução de pressão podem iniciar um processo de ebulição do produto, gerando bolhas em seu meio (BINAMA; MUHIRWA; BISENGIMANA, 2016). Essas bolhas em alta velocidade no interior da bomba podem gerar sérios danos estruturais e reduzir drasticamente a vida útil do equipamento. Com a formação das bolhas, podemos encontrar súbitas mudanças de pressão que causam grande impacto no interior da bomba, ocasionando a sua erosão (MCKEE *et al.*, 2011). O fenômeno de cavitação normalmente é detectado por métodos numéricos ou analíticos, dentre eles o mais popular é por meio do NPSH (*Net Positive Suction Head*), o qual utiliza uma série de testes e medições para determinar a carga crítica necessária para evitar o fenômeno (CHUDINA, 2003). Já a sua detecção por observação é mais difícil devido a velocidade e local em que ocorre. Por conta disso, os métodos se focam mais nos efeitos secundários do que no

fenômeno em si, como a distribuição de pressão, medidas de suscetibilidade (BINAMA; MUHIRWA; BISENGIMANA, 2016), emissão de som (CHUDINA, 2003) e corrosão da superfície do rotor que a torna porosa. (JARDIN; RIFFEL; COSTA, 2016)

Para evitar a cavitação, o principal método recomendável é atuar com a bomba acima do NPSH requerido (informação fornecida pelo fabricante) (MCKEE *et al.*, 2011). Porém, caso não seja possível evitar o fenômeno, algumas medidas para reduzir o impacto incluem um bom polimento na superfície do rotor, uso de revestimento de elastômeros e pequeno valor para o ângulo de inclinação das pás na entrada (entre 15° e 30°). (JARDIN; RIFFEL; COSTA, 2016)

2.4.2 Falha de Lubrificação

A lubrificação tem por função reduzir a fricção entre partes móveis permitindo uma movimentação facilitada. Uma falha nesse processo gera uma sobrecarga sobre o sistema causando maior desgaste das peças e um aumento da temperatura durante o processo. Um cuidado sobre a lubrificação aumenta a vida útil da máquina e de seus componentes, se tornando um fator de atenção (MCKEE *et al.*, 2011). Com toda essa importância na devida lubrificação do equipamento, alguns fatores para o devido funcionamento da máquina devem ser considerados sobre o lubrificante: lubrificante adequado para a operação, dosagem correta, contaminação e vida útil do próprio lubrificante (SILVA, 2021).

A falha pode se dar por meio de contaminantes, como água e partículas sólidas, ou pelo aumento de temperatura que aumenta sua viscosidade, reforçando a manutenção e monitoramento constante do óleo de lubrificação. Esse monitoramento ocorre por meio da coleta e análise em laboratório que, além de detectar falhas no óleo, também pode ser eficaz para a detecção de outras falhas em componentes, como rolamentos (SILVA, 2021; CHEN *et al.*, 2022).

2.4.3 Falha de Rolamento

O rolamento é um componente importante para várias máquinas rotativas e a bomba centrífuga não é exceção. Ligado ao rotor da bomba, é responsável por garantir uma melhor movimentação da máquina e, caso falhe, pode provocar um aumento de temperatura, vibração, perda de eficiência e em casos mais extremos, a

parada da máquina (CHEN *et al.*, 2022). Sua falha pode ser causada pela falha de lubrificação, reforçando a importância da boa lubrificação; sobrecarga causada pela operação do sistema fora da sua operação ideal (BEP) (MCKEE *et al.*, 2011); partículas estranhas prensadas nos elementos rotativos, impacto que podem gerar trincas; e, por fim, desalinhamento que podem ser corrigidos pela devida montagem no equipamento em sua área de trabalho (JARDIN; RIFFEL; COSTA, 2016).

2.4.4 Falha no Selo

Os selos mecânicos são estruturas que tem por função evitar a perda de fluídos entre a parte móvel, ou rotativa da bomba, e a parte fixa durante a operação da máquina, de formato cilíndrico que serve de vedação para essa região, sem danificar os demais componentes (RODRIGUES, 2012). A falha do selo pode ser ocasionada devido a uma abertura nas faces lapidadas que permitem a entrada de sólidos entre a superfície e, por esse motivo, o movimento produzido gera um alto desgaste da face rígida. Posteriormente, será responsável por várias falhas de vedação (MCKEE *et al.*, 2011).

2.4.5 Desbalanceamento e Desalinhamento

Desalinhamento ocorre com o desvio do nível ou angulação de uma extremidade do eixo principal, causando um deslocamento na posição do eixo. Já o desbalanceamento diz respeito a uma má distribuição de pesos ao longo da bomba que pode ser causado por incrustação, falha na montagem, qualidade do material, entre outros. Ambos os erros são estruturais podendo ser corrigidos com a devida montagem da bomba e têm como consequência o aumento de vibração do sistema. (CHEN *et al.*, 2022)

2.5 EVOLUÇÃO DOS CONCEITOS DE MANUTENÇÃO

Desde o uso das primeiras ferramentas a prática de manutenção é aplicada, buscando preservá-las pelo maior tempo possível. Este assunto passou a ter maior destaque no século XVI com o surgimento das primeiras máquinas têxteis, onde um profissional chamado de ‘mantenedor’ passa a ser necessário para manter a correta operação das máquinas (CARELLI; MACIEL, 2006).

A partir disso, o assunto de manutenção se apresentou cada vez mais relevante, e agregou maior valor para as empresas, seja por questões ambientais, financeiras ou de segurança. Atualmente, o conceito de manutenção transcende a simples correção de falhas imediatas e reparos de equipamentos no momento em que apresentam defeitos. Inclui também a prática de cuidados sistemáticos e periódicos, com o propósito de prolongar a vida útil dos equipamentos. Essa abordagem proativa visa prevenir a ocorrência de falhas ao detectar e corrigir potenciais anomalias no comportamento dos equipamentos antes que estas se manifestem de forma significativa (CARELLI; MACIEL, 2006; SILVA; NASCIMENTO, 2020)

Dentro do processo de manutenção se destacam três tipos principais: corretiva, preventiva e preditiva; cada uma com a sua especificidade e aplicação dentro da indústria e cuidando de aspectos diferentes da manutenção (CRUZ, 2021).

2.5.1 Manutenção Corretiva

Manutenção corretiva é o conceito mais antigo aplicado na indústria e nos equipamentos. A princípio, este modo de manutenção realiza a operação para reparar o maquinário quando ele se quebra. Porém, com o novo conceito de manutenção para evitar falhas quando elas ocorrem, a manutenção corretiva tem diminuído e aberto espaço para as manutenções, como as preventivas e preditivas, que buscam evitar a quebra do equipamento. Assim, o conceito de manutenção corretiva tem sido abandonado como fator. Entretanto, nem todas as falhas são previstas ou detectadas, e, para estes casos, a manutenção corretiva ainda se faz necessária (SILVA *et al.*, 2017). Este modelo de manutenção é o mais simples e o mais aplicado no dia-a-dia em ambientes domésticos, onde o profissional apenas é acionado quando a televisão quebra ou quando o telefone celular não liga mais. Nos ambientes mais simples, em que os aparelhos apresentam funções independentes, pode ser muito estressante e

custoso manter alguma forma de manutenção constante em certos aparelhos, tornando o modelo corretivo o mais econômico, com exceção nos casos de equipamentos maiores e de maior custo quando ações preventivas são mais simples como em automóveis (Exp. Troca de óleo) e ar-condicionado (Exp. limpeza de filtro).

Porém, quando se trata de um ambiente industrial, com processos complexos e interligados, uma falha pode afetar todo o processo, e as perdas podem superar as economias citadas anteriormente. Primeiramente, a quebra do equipamento gera o que é chamado de ‘parada não programada’, quando há uma interrupção no processo de produção que não havia sido planejada. Essa parada costuma levar mais tempo que paradas programadas e, por não terem sido planejadas, podem gerar atraso em entrega ou falta de estoque. Outro problema deste modo de manutenção tem sido sua aplicação, pois apenas é acionado em momentos em que a avaria se encontra em estado avançado, comprometendo e danificando também outras partes do equipamento. Todos esses fatores, em um contexto industrial complexo e dinâmico, se refletem em perdas financeiras que superam qualquer economia inicial (CRUZ, 2021).

Por fim, manutenções corretivas resultam em menor confiabilidade do processo, uma vez que elas ocorrem repentinamente sem programação, além de todos os problemas já citados. Outro agravante é que este modo de manutenção não combate a raiz do problema, e se limita a lidar com as consequências apenas para colocar a máquina de volta a operação, ignorando a preocupação com quebras posteriores (SILVA *et al.*, 2017).

2.5.2 Manutenção Preventiva

A manutenção preventiva já consiste em prevenir eventuais falhas e danos no equipamento. Esta prevenção é feita por meio de paradas programadas e operações de ajuste e inspeção do equipamento, com ações como troca de óleo e reaperto de parafusos. Com essas operações, a manutenção preventiva visa estender a vida útil do equipamento (CRUZ, 2021). Justamente pelos princípios citados, as grandes vantagens desta manutenção é gerar maior confiabilidade, uma vez que as paradas de manutenção passam a ser programadas e não mais imprevistas, além de poder contar com atuações antes de falhas se agravarem, poupando o equipamento de maiores danos (SILVA *et al.*, 2017).

Apesar de parecer ótima, estas manutenções exigem ser muito bem planejadas para que sua eficiência realmente venha à tona, isso porque o excesso de paradas reduz a porcentagem de tempo efetivo ao tirar o equipamento da produção. Na produção, isso pode ser compensado por uma bomba auxiliar para que o processo produtivo não seja parado, porém, ainda há a redução do aproveitamento da bomba principal. Da mesma forma, caso a frequência seja muito baixa, a prevenção pode se mostrar ineficaz e resultar em quebras, sendo necessário recorrer ao uso da manutenção corretiva (CARELLI; MACIEL, 2006).

2.5.3 Manutenção Preditiva

Durante a década de 1970 ocorre um aumento considerável na demanda de produção de modo que a paralização da produção se tornou uma preocupação frequente, neste conceito nasceu o conceito de manutenção preditiva (SILVA *et al.*, 2017). A manutenção preditiva tem como princípio participar do processo produtivo ao monitorar, em tempo real, a operação e as condições do equipamento, deste modo a manutenção preditiva busca prever as falhas de equipamento antes que elas ocorram e tomar as ações necessárias para reduzir danos ao equipamento e aumentar a confiabilidade do processo (CRUZ, 2021).

Com este princípio, a manutenção preditiva apresenta as mesmas vantagens da parada programada vistas na manutenção preventiva, aumentando a confiabilidade, evitando transtornos e gastos desnecessários, e reduzindo perdas. Além disso, o monitoramento constante permite que as paradas programadas ocorram de acordo com a condição do equipamento, com operações junto de paradas já programadas, ou programando novas de acordo com a necessidade. Isso aumenta o tempo produtivo do maquinário sem prejudicar a sua saúde, tornando um dos modos mais eficientes e eficazes quando se trata de conservar o equipamento (CARELLI; MACIEL, 2006). Apesar de todas essas vantagens, a manutenção preditiva é a que exige maior investimento dentre as três, por conta do grande uso de sensores e ferramentas de análise para manter o supervisionamento, como também a equipe técnica qualificada para realizar as devidas análises e planejar corretamente as operações. Isso a torna a mais custosa e pode se tornar ineficaz, caso a equipe não seja capaz de lidar com os dados, podendo gerar um desperdício de equipamentos pela falta de preparo (SANTOS; BOVÉRIO; SILVA, 2023).

3 MATERIAIS E MÉTODOS

Uma vez que o objetivo geral deste trabalho é o desenvolvimento de um assistente virtual capaz de auxiliar profissionais na manutenção e assistência técnica em bombas centrífugas, a aplicação foi denominada E-MANUEL, um *chatbot* autônomo com métodos de resposta baseado em recuperação, se utilizando de *Deep Learning* e palavras-chave para definir a melhor resposta ao usuário, e domínio fechado, uma vez que o E-MANUEL apenas deve responder perguntas voltadas a manutenção de bombas centrífugas.

Para o desenvolvimento do projeto foi selecionado um modelo de bomba existente no parque industrial e de conhecimento do autor através do estágio supervisionado realizado.

3.1 ESPECIFICAÇÕES TÉCNICAS DA BOMBA KSB-ETA

KSB é uma das principais fornecedoras de bombas no Brasil, incluindo bombas centrífugas para a área industrial. Fundada em 1871, a empresa alemã é especializada em venda e manutenção de bombas, disponibilizando manuais de manutenção e manuais técnicos de seus produtos. Devido a disponibilidade de documentação técnica e a experiência em bombas centrífugas obtida pelo aluno durante o processo de estágio, o modelo escolhido para modelagem neste trabalho foi o ETA (KSB, 2013) da empresa KSB, como apresentada na Figura 5.

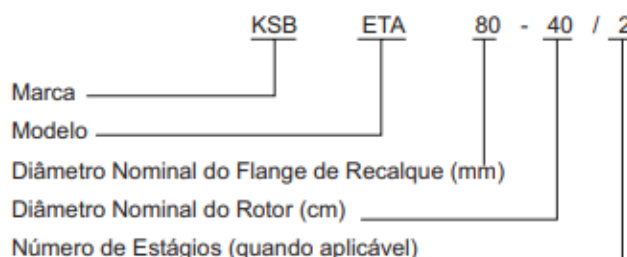
Figura 5 - Bomba KSB ETA



Fonte: KSB (2013)

Como informado no manual técnico, sua aplicação é indicada para bombeamento de líquidos limpos ou turvos como água, óleos e sistema de resfriamento. O código de classificação adotado para este modelo segue a ordem: fabricante, modelo, diâmetro do flange de recalque, diâmetro do rotor e o número de estágios (Figura 6).

Figura 6 - Nomenclatura de bomba centrífuga



Fonte: KSB (2013)

As especificações técnicas para o usuário obter melhor desempenho do equipamento, são (KSB ,2013):

- Temperatura máxima: 140° C (acima de 80° deve conter câmara de resfriamento instalado)
- Vazão máxima: 1800 m³/s
- Elevação máxima: 120m

Para a devida lubrificação da bomba, o lubrificante a ser usado deve seguir as especificações apresentadas na Figura 7.

Figura 7 - Tabela de indicação de lubrificante

**Na falta dos tipos indicados usar SAE 20 ou 30
(Não use HD)**

Velocidade da bomba:	até 1.800 rpm	acima de 1.800 rpm
ATLANTIC	Duro AW 68	Duro AW 46
CASTROL	Hyspin 68	Hyspin 46
ESSO	Teresso 68	Teresso 46
IPIRANGA	Ipitur AW 68	Ipitur AW 46
MOBIL OIL	Mobil D.T.E. - 26	Idem
SHELL	Tellus oil 68	Tellus oil 46
TEXACO	Rigal oil RO 100	Rigal oil RO 68
PETROBRAS	Marbrax TR 52	Marbrax TR 52

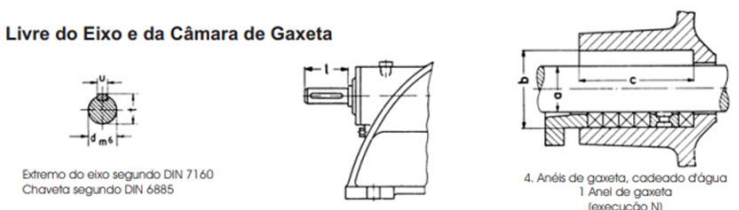
— Verifique na tabela da pág. 11 a quantidade necessária de óleo.

Fonte: KSB (2013)

Seguindo com as especificações da bomba foi necessário, dentro do catálogo, selecionar o diâmetro do flange, que para este modelo pode variar entre 32mm a 300mm, e o diâmetro de rotor, que pode variar de 12cm a 50cm. Deste modo os tamanhos escolhidos foram 125mm de flange e 40cm de rotor, ou seja, a bomba KDB ETA125-40. Com isso novas especificações técnicas foram definidas (Figura 8), que serão importantes para informar o técnico e definir os devidos procedimentos na bomba.

Figura 8 - Especificações baseadas nos diâmetros de flange e rotor

7. Medidas do Extremo Livre do Eixo e da Câmara de Gaxeta



Extremo do eixo segundo DIN 7160
Chaveia segundo DIN 6885

4. Anéis de gaxeta, codados d'água
1 Anel de gaxeta
(execução N)

Medidas em mm

MODELO DA BOMBA	Suporte	ROLAMENTO DE ESFERA DIN 625	PONTA DO EIXO				CÂMARA DE GAXETA			Quant. de Óleo (litro)	Relentores
			Ø dm 6	l	u	t	Ø a	Ø b	c		
125-33 125-40 125-50/2 **) 150-26 150-33 200-23	"C"	6409/C3	42	105	12	45,1	45	65	75	1,2	45 X 62 X 12
150-40 150-50 200-23											

Fonte: KSB (2013)

3.2 PROCESSOS DE MANUTENÇÃO DA BOMBA CENTRÍFUGA KBS-ETA

Alguns dos fatores humanos que podem prejudicar o processo de manutenção adequada da bomba são: a equipe ou um membro da equipe não seguir o procedimento técnico correto, habitualidades (vícios) e falta de treinamento (SILVA, 2016). Visando reduzir estas falhas no processo de manutenção da bomba KSB-ETA o Chatbot E-MANUEL foi projetado para: (1) auxiliar no fornecimento de informações sobre o equipamento e (2) auxiliar o profissional responsável em processos de manutenção padrão. Dentre estes processos uma ação comum a se realizar é o processo de inspeção do equipamento. Esta inspeção pode ocorrer ao final de outras operações ou de forma periódica, a fim de garantir o devido funcionamento do maquinário. Seguindo as orientações de instalação e manutenção de bombas centrífugas no manual técnico da bomba KBS-ETA (KSB, 2013), um *checklist* foi preparado para ser implementado no *chatbot* E-MANUEL:

- Alinhamento do conjunto
- Nível do óleo
- Aperto de parafusos
- Movimentação do eixo
- Acoplamento
- Vazamentos na gaxeta

Além do processo de *checklist*, outros processos foram modelados no E-MANUEL, incluindo processos de pré e pós manutenção que, se realizados de forma indevida, podem gerar novas falhas e retrabalho (SILVA, 2016). São eles: (I) processo de instalação da bomba; (II) processo de montagem e desmontagem da bomba; (III) processo de troca de lubrificante; (IV) processo de troca de rolamento.

I. Processo de Instalação

De acordo com o manual técnico da empresa KSB (KSB, 2013), algumas observações são importantes referente à instalação do equipamento e à sua aplicação: para alguns tamanhos de bomba, além da fixação comum ao cavalete dos mancais por meio de flanges, o apoio deve ser incrementado por pés reforçados e o eixo deve ser protegido por uma bucha instalada na gaxeta.

A base, se de concreto deve estar seca e completamente enrijecida, nivelada por meio de nível de bolha e caso necessário, utilizar encalço para ajuste.

Com a base devidamente alinhada, o próximo passo é chumbar com argamassa evitando sempre de deixar bolhas de ar e buracos, todos os parafusos de ancoragem devem estar firmemente e uniformemente apertados apenas após a pega do cimento.

Na fixação da tubulação, deve-se garantir que ela encoste nos flanges da bomba sem dificuldade. Para evitar a formação de bolha é recomendável que a tubulação de sucção apresente um leve aclave na falange de aspiração da bomba e as válvulas ligadas à sucção devem manter a haste na horizontal. Terminados os procedimentos iniciais, deve-se verificar e garantir uma livre rotação do eixo.

II. Processos de Montagem e Desmontagem

A desmontagem (KSB, 2013) da bomba se inicia na bancada de trabalho do técnico responsável, começando pela remoção da tampa de aspiração e da porca do rotor. O modelo escolhido utiliza suporte C, o que implica também na remoção da arruela de segurança. Após isso, tirar a tampa do cavalete dos mancais do lado do acionamento.

Por fim, o rotor pode ser retirado com pancadas leves no eixo. Em alguns casos o rotor pode apresentar dificuldade de sair do eixo e neste caso é sugerido o uso de solventes de ferrugem para que possa se movimentar com maior facilidade.

Para desmontar o cavalete, inicialmente o aperta-gaxeta deve ser solto para que assim o cavalete possa ser afastado do corpo da bomba. Então deve-se tirar a vareta indicadora de óleo e remover cuidadosamente do eixo. Por fim soltar a tampa do mancal do lado da gaxeta. Para o processo de montagem basta seguir a ordem inversa das instruções.

III. Processo de troca de Lubrificante

Para inserir o lubrificante, os cavaletes e os rolamentos devem ser lavados utilizando gasolina, a lavagem é realizada girando levemente o eixo. Após isso os componentes devem ser secos e então os cavaletes podem ser preenchidos com óleo. O nível do óleo pode ser verificado por uma vareta indicadora de nível de óleo.

IV. Processo de troca de Rolamento

Para trocar o rolamento primeiro o rolamento anterior precisa ser aquecido de modo que o eixo não seja aquecido junto, para então afastar o eixo do anel interno

por meio de pancadas leves. Para inserir o novo rolamento, primeiro este deve ser banhado em óleo quente até a temperatura de 80°C, para então serem assentados no eixo até o encosto.

3.3 BANCO DE DADOS DA EMPRESA

Para que o E-MANUEL gere e disponibilize conhecimento técnico para o usuário, uma das fontes de conhecimento é obtida a partir do banco de dados da empresa. Neste trabalho, foi implementado um pequeno banco de dados para simular essa base de dados técnicos. O ambiente de desenvolvimento utilizado foi o PgAdmin4 que permite gerenciar banco de dados Postgres (THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2024)

No banco de dados implementando, todas as informações técnicas da bomba foram armazenadas numa mesma tabela nomeada **dados_tecnicos** (Algoritmo 5). A primeira coluna intitulada **codigo** corresponde ao código de identificação da bomba na planta industrial, e no banco de dados esse código foi modelado como chave primária (*primary key*). As demais colunas da tabela são de informações encontradas no manual técnico. Na tabela **tm_rotor** refere-se ao diâmetro do rotor em milímetros (mm), **velocidade** é a velocidade de rotação que o rotor opera em Rotação Por Minuto (RPM) e **produto** é a substância com a qual o sistema opera. A seguir é apresentado o código de criação da tabela

Algoritmo 5 - Estrutura para criação da tabela dados_tecnicos

```
CREATE TABLE IF NOT EXISTS dados_tecnicos
( codigo varchar(6) NOT NULL,
  empresa varchar(25),
  modelo varchar(255),
  tm_rotor double precision,
  velocidade integer,
  produto varchar(100),
  CONSTRAINT dados_tecnicos_pkey PRIMARY KEY (codigo) )
```

No preenchimento dos dados no banco de dados, Figura 9, o campo **codigo** é preenchido com uma sequência de 2 letras e 4 algarismos numéricos e as

demais colunas foram preenchidas de acordo com o manual técnico da bomba KBS-ETA.

Figura 9 - Dados da bomba

codigo [PK] character varying (6)	empresa character varying (25)	modelo character varying (255)	tm_rotor double precision	velocidade integer	produto character varying (100)
AB1234	KSB	KSB ETA 125-40	400	3500	Água

Fonte: Acervo pessoal

Foi criada ainda uma tabela **lubrificação** (Algoritmo 6) conectada à tabela **dados_tecnicos** por meio da chave estrangeira **cd_bomba** que referencia a chave primária **codigo**. A chave estrangeira (ou *Foreign Key*) é uma chave da tabela que permite a integração com outras tabelas por meio da chave primária destas outras tabelas (DATASUS, 2019). O código de criação da tabela **lubrificação** é apresentado abaixo, onde o campo **quantidade** refere-se ao volume de lubrificante necessário em Litros (L) e **frequencia_troca** refere-se ao período em dias da troca de óleo.

Algoritmo 6 - Estrutura para criação da tabela lubrificacao

```
CREATE TABLE IF NOT EXISTS lubrificacao
(
    cd_bomba varchar(6) NOT NULL,
    tp_oleo varchar(255) NOT NULL,
    marca varchar(100),
    quantidade double precision,
    frequencia_troca integer,
    CONSTRAINT lubrificacao_cd_bomba_fkey FOREIGN KEY (cd_bomba)
        REFERENCES public.dados_tecnicos (codigo)
)
```

3.4 INSTALAÇÃO E ESTUDO DA FERRAMENTA CHATTERBOT

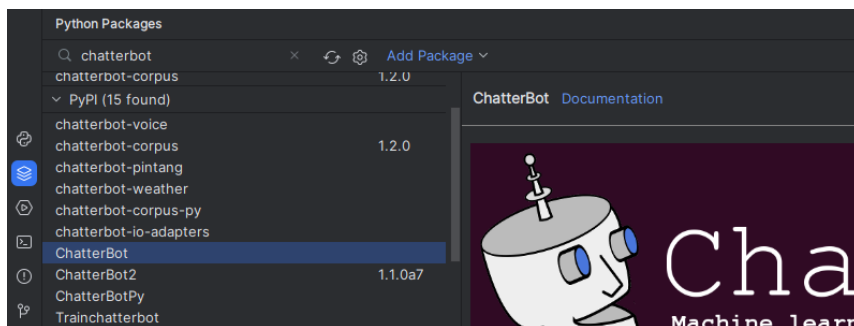
A implementação do assistente virtual foi realizada em linguagem de programação Python devido a sua ampla gama de bibliotecas disponíveis gratuitamente voltadas para desenvolvimento em I.A. e *Deep Learning* bibliotecas importantes para criação de *chatbots*. O ambiente de desenvolvimento utilizado foi a IDE PyCharm (JETBRAINS S.R.O, 2024) e a ferramenta Chatterbot, a qual disponibiliza ferramentas nativas para uso de *Deep Learning* que permitem ao

assistente virtual um constante aprendizado da linguagem humana e uso de diálogos prontos para a criação de seu repertório.

A preparação do ambiente foi iniciada com a instalação da IDE PyCharm. O *download* do executável **PyCharm-professional-2023.3.4.exe** pode ser obtido no site oficial (JETBRAINS S.R.O, 2024).

Na criação do projeto os pacotes Chatterbot foram baixados por meio do próprio PyCharm em 'Python Packages'. A localização pode ser obtida ao pesquisar Chatterbot e a instalação por meio de 'Install package' (Figura 10).

Figura 10 - Python packages



Fonte: Acervo pessoal

Para estudos sobre implementação de *chatbot* na ferramenta, foi criado um *chatbot* teste, seguindo os passos:

- Criação do arquivo *main.py* que rege a troca de mensagens
- Criação do adaptador lógico '*firstAdapter*' em um arquivo *.py*
- Modelagem do *chatbot* em um arquivo *assistente.py*
- Criação do pré-processador em um arquivo *processador.py*
- Criação de um arquivo *.py* com diálogos simples e treinamento do *chatbot* teste

O arquivo *main.py* contém um programa que realiza a interação com o usuário em um looping onde a digitação do usuário é coletada pelo comando *input()*, passado para o *chatbot* teste por meio do comando *getResponse()* e retorna um *statement* contendo a resposta ao usuário e seu texto é transmitido pelo comando *print()*.

O adaptador lógico **MyAdapter** (Algoritmo 7) foi criado no arquivo *firstAdapter.py* e separado em uma pasta feita para adaptadores lógicos intitulada

‘adaptadoresLogicos’. Este adaptador tem a intenção de fornecer uma informação de temperatura, para isso o adaptador busca nas entradas do usuário a *intent* temperatura, caso detectado o adaptador cria um novo *statement* utilizando como texto ‘A temperatura atual é de x°C’ e insere o valor de *confidence* baseado na proximidade da palavra ‘temperatura’.

Algoritmo 7 - Adaptador lógico *MyAdapter*

```

1  confidence = 0.0
2  inpute = input_statement.text
3  frase = inpute.split()
4  for c in frase: # Busca a palavra que mais se assemelha a temperatura
5      |   conf = SequenceMatcher(None, c, 'temperatura').ratio()
6      |   confidence = max(conf, confidence)
7  temperature = 22
8  selected_statement = Statement(text='A temperatura atual é de {}°C'.format(temperature))
9  selected_statement.confidence = confidence
10 return selected_statement

```

Também é importante ressaltar que todo novo adaptador lógico deve implementar a inicialização assim como exemplificado

Algoritmo 8 - Inicialização padrão de adaptadores lógicos

```

class MyAdapter(LogicAdapter):
    def __init__(self, chatbot, **kwargs):
        super().__init__(chatbot, **kwargs)

```

Na inicialização *do* chatbot teste (Algoritmo 9) o primeiro campo se refere ao nome adotado para o *chatbot*, em sequência há a declaração do **storage_adapter**, o adaptador de armazenamento que é responsável por armazenar os diálogos do assistente virtual, então temos os adaptadores lógicos, **logic_adapters**, onde deve ser informado o caminho em que se encontra o adaptador lógico e os argumentos exigidos, e por fim o **database_uri** que referência o banco de dados onde as conversas são armazenadas.

Algoritmo 9 - Inicialização do chatbot teste

```

1 chatbot = ChatBot(
2     "Teste",
3     storage_adapter = "Chatterbot.storage.SQLiteStorageAdapter",
4     preprocessors = [
5         "Processador.pross"
6     ],
7     logic_adapters = [
8         {
9             "import_path" : "Adapters.FirstAdapter.MyAdapter"
10        },
11        {
12            "import_path" : "Chatterbot.logic.BestMatch",
13            "default_response" : "Sinto muito, não sou capaz de ajudar nisso",
14            "maximum_similarity_threshold" : 0.50
15        }
16    ],
17     database_uri = "sqlite:///database.sqlite3"
18 )

```

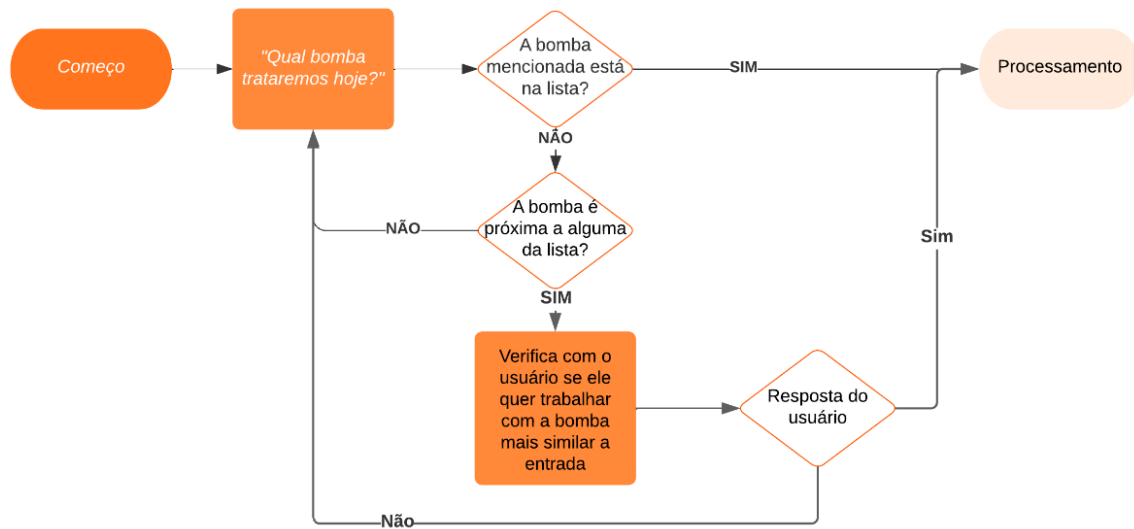
O pré-processador é responsável por realizar um tratamento inicial na mensagem enviada, neste caso ele passa toda a mensagem para *lowercase*, remove caracteres especiais e remove espaços do começo e do final da sentença

3.5 MODELAGEM DA INTERAÇÃO HUMANO-MÁQUINA

Para que o E-MANUEL possa dar início ao suporte na manutenção das bombas centrífugas da planta industrial, inicialmente é necessário informá-lo em qual delas se dará a operação. Ao ser ativado o *chatbot* E-MANUEL pergunta ao usuário o código da bomba a ser trabalhada de modo que seja extraída da resposta do usuário um código válido, assim o assistente armazena este dado para uso posterior em suas funções.

Porém em caso o usuário não passe um código válido o E-MANUEL iniciará um procedimento para a obtenção da informação conforme demonstrado no fluxograma da Figura 11.

Figura 11 - Diagrama de blocos da inicialização da interação



Fonte: autoria própria

Inicialmente o chatbot questiona ao usuário “Qual bomba trataremos hoje?”, as respostas esperadas do usuário são afirmações como: “O código da bomba é AB1234”, “Vamos trabalhar na bomba AB1234”, “Preciso de ajuda com a AB1234”, o usuário também pode apenas digitar o código. Caso o usuário informe um código válido, o E-MANUEL armazena esse código e segue oferecendo ajuda ao usuário. Caso o código informado seja inválido o chatbot buscará por um código próximo dentro da lista cadastrada no banco de dados, esse procedimento foi pensado para casos como erros simples de digitação em relação ao usuário (Exemplo: “AV1234” ao invés de “AB1234”). Ao encontrar o código, E-MANUEL solicita uma confirmação ao usuário, em caso positivo o código encontrado é armazenado e o processamento continua. Em caso negativo, o chatbot solicita que o usuário informe novamente a bomba. Se o código informado não existe, o chatbot volta a solicitar um novo código ao usuário.

A partir de então o usuário pode solicitar ajuda ao E-MANUEL em uma de 4 categorias: informações de dados técnicos, processo de inspeção (*checklist*), guia de processos de manutenção e guia de falhas. A implementação destes processos está distribuída entre adaptador lógico e treinamento com diálogos (*Deep Learning*). Esta distribuição é apresentada no Quadro 1.

Quadro 1 - Relação entre função, modo de implementação e nome do adaptador lógico

Categoria	Técnica	Adaptador lógico
Informações de dados técnicos	Adaptador lógico	BuscaDadosTec BuscaDadosLub
Checklist	Adaptador lógico	Checklist
Ações de manutenção	Deep Learning	BestMach
Guia de falhas	Deep Learning	BestMach

Fonte: autoria própria

O Quadro 1 apresenta ainda os adaptadores lógicos que são utilizados em cada função sendo eles BuscaDadosTec, BuscaDadosLub, Checklist e *BestMach*. O *BestMach* é um adaptador lógico padrão do Chatterbot responsável pelo processo de geração de resposta baseadas em *Deep Learning* (GUNTHER COX REVISION, 2021). Já os demais adaptadores são de autoria própria desenvolvidos para suprir as necessidades específicas do E-MANUEL.

3.5.1 Modelagem do Procedimento de Busca de Dados Técnicos

Para solicitar informações técnicas ao E-MANUEL o usuário pode inserir perguntas como ‘Qual o tamanho do rotor?’ ou ‘Que marca de óleo que a bomba usa?’. Estes questionamentos são tratados pelos adaptadores lógicos **BuscaDadosTec** e **BuscaDadosLub** retornando o valor da informação requisitada por meio de respostas como ‘O rotor mede 40cm’ ou ‘Esta bomba utiliza óleo marbrax tr 52’. Estes adaptadores lógicos utilizam grupos de palavras-chave para extrair o **contexto** e a **intenção** do usuário. O adaptador lógico BuscaDadosTec, busca por palavras-chave de três grupos distintos: palavras que remetam a uma pergunta, palavras que indiquem a *intent* e palavras de contexto. A relação entre adaptadores lógicos, palavras-chave e *intent* é mostrada no Quadro 2.

As informações de características da bomba que o usuário pode pesquisar pelo E-MANUEL são: código da bomba, modelo, fabricante, tamanho de rotor, velocidade de operação, volume de lubrificante, marca do lubrificante, frequência da troca de óleo e fabricante do óleo sendo que **BuscaDadosLub** é responsável por trazer dados relacionados a lubrificação do equipamento enquanto **BuscaDadosTec** está relacionado com os demais dados técnicos.

Quadro 2 - Relação entre adaptadores, palavras-chave e intents

Adaptador lógico	Palavras de contexto	Palavra-chave	Intent
BuscaDadosTec		Rotor	Tamanho do rotor
		Velocidade, rotação	Velocidade de rotação
		Produto, substância	Produto com que opera
		Modelo	Modelo da bomba
		Empresa, fabricante	Empresa fabricante
		Código, numeração	Código da bomba
BuscaDadosLub	Óleo, lubrificante, lubrificação	Frequência, período	Período de troca de óleo
		Fabricante, empresa	Fabricante do lubrificante
		Volume, quantidade	Volume de lubrificante
		Marca, tipo	Marca do lubrificante

Fonte: autoria própria

Identificada a *intent* e o contexto da requisição os adaptadores realizam uma busca no banco de dados para encontrar a informação solicitada e retornam a frase que melhor apresente esse resultado. A relação entre *intent* e resposta fornecida pelos adaptadores é apresentada na Quadro 3 onde '{ }' é substituído pelo dado técnico registrado no banco de dados.

Quadro 3 - Relação entre *intents* e respostas

Intent	Resposta
Tamanho do rotor	O rotor mede { }cm
Velocidade de rotação	Essa máquina opera á { }rpm
Produto com que opera	Esta bomba está operando com { }
Modelo da bomba	Essa máquina é uma bomba é uma { }
Empresa fabricante	A bomba foi feita pela { }
Código da bomba	Essa bomba está registrada como { }
Período de troca de óleo	O óleo precisa ser trocado a cada { } dias
Empresa que fabrica o lubrificante	Essa marca é produzida pela { }
Volume de lubrificante necessário	A bomba precisa de { }L de óleo
Marca do lubrificante	Esta bomba utiliza óleo { }

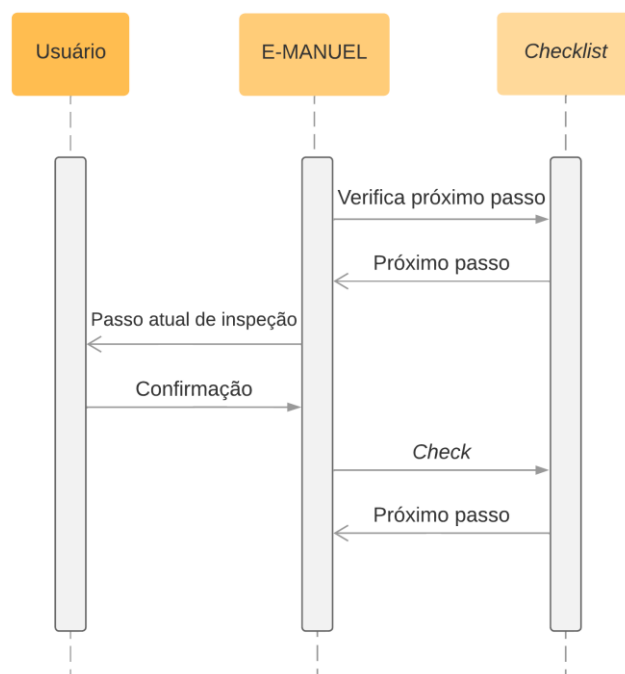
Fonte: autoria própria

3.5.2 Modelagem do Processo de *Checklist*

Para iniciar um processo de *checklist* o usuário pode usar frases como “Vamos fazer a *checklist*” ou “Preciso começar o processo de *checklist* desta bomba” o processo de *checklist* é iniciado pelo adaptador lógico *checklist* pela detecção da palavra-chave e *intent checklist*.

Iniciado o processo, E-MANUEL armazena internamente uma *checklist* com os diferentes itens que devem ser verificados e sua condição de inspeção com 'falso' para itens ainda não verificados e 'verdadeiro' para itens verificados. Então o *chatbot* passa a pergunta ao usuário sobre um item não verificado da *checklist*, podendo receber como resposta de confirmação como "Certo", "Está pronto", "O sistema está alinhado", "Já conferi o óleo" e similares para que seja aceito como realizado (realizar o 'check'), então o E-MANUEL atualiza o *checklist* e retorna o passo seguinte. Este fluxo de informações é representado no diagrama da Figura 12.

Figura 12 - Diagrama de sequência do checklist



Fonte: autoria própria

Para cada item de verificação o E-MANUEL envia uma mensagem personalizada de acordo com o Quadro 4.

Quadro 4 - Relação entre etapa e resposta

Etapa	Resposta
lubrificante	O nível do óleo está adequado?
parafusos	Por favor verifique se os parafusos estão bem apertados
nivelamento	O equipamento está devidamente nivelado?
acoplamento	O motor está devidamente acoplado a bomba?
movimentação	O eixo consegue movimentar livremente?
gaxeta	Tudo certo com a gaxeta?

Fonte: autoria própria

Ao finalizar todos os itens da *checklist* o E-MANUEL informa o usuário que a *checklist* está finalizada e pergunta se ele deseja ajuda em algo mais.

3.5.3 Modelagem do Guia de Ações de Manutenção

Caso o usuário esteja buscando orientação com processos de montagem, desmontagem ou instalação ele pode solicitar orientação ao E-MANUEL por meio de frases como “Preciso de ajuda para desmontar a bomba” ou “Como faço para instalar a bomba?”. Deste modo o E-MANUEL passa o primeiro passo do processo solicitado.

Após cada passo realizado pelo usuário este deve informar ao E-MANUEL descrevendo o último passo realizado de modo que o assistente responda com o passo seguinte e permanecendo nesta mesma troca de mensagens até o processo ser concluído e o E-MANUEL enviar uma frase de finalização.

Em alguns passos o usuário também pode informar se tiver alguma dificuldade e o E-MANUEL poderá passar uma sugestão de ajuda. Os processos que o E-MANUEL pode guiar estão listados e detalhados na seção 3.2.

3.5.4 Modelagem do Guia de Falhas

O usuário também pode informar ao *chatbot* a ocorrência de uma falha encontrada no equipamento. Neste caso o E-MANUEL retorna uma orientação ao usuário sobre como proceder e indica possíveis causas desta falha. Dentre os tipos de falhas mais comuns (seção 2.4) foram implementadas no *chatbot* orientações sobre falha de rolamento e falha de lubrificação.

No caso de **falha de rolamento** E-MANUEL solicita a troca do rolamento e pede a verificação de três fatores que possam ter causado danos no rolamento: o alinhamento, a lubrificação e a carga de trabalho.

No caso de **falha de lubrificação** E-MANUEL solicita detalhes sobre a condição do lubrificante, se ele apresenta impurezas, se está excessivamente quente e se perdeu a viscosidade. Caso o óleo esteja quente é sugerida a verificação do nível de óleo e alinhamento do conjunto. Caso o óleo apresente impurezas é indicada a troca e envio para análise em laboratório especializado, pois pode indicar outras possíveis falhas no equipamento. Caso o óleo tenha perdido a viscosidade o E-MANUEL dirá para realizar a troca, porém já sugere que caso ele esteja ficando muito rápido seja realizado uma reanálise da frequência de troca. Caso o óleo esteja abaixo do nível adequado a orientação é a de completar o nível de lubrificante e inspeção de vazamentos.

3.6 IMPLEMENTAÇÃO DO *CHATBOT* E-MANUEL

Para que o E-MANUEL possa ser acessado e utilizado ele deve primeiramente ser inicializado (Algoritmo 10) definindo nome, adaptadores, preprocessador e local de armazenagem para seus diálogos. Nesta inicialização os adaptadores lógicos BuscaDadosTec e BuscaDadosLub devem receber o código como parâmetro de bomba por meio da função inicializar() que inicia o dialogo inicial para a definição da bomba tratada, retorna o código da bomba e armazena em uma variável global 'codigo' para passar como parâmetro para a inicialização dos adaptadores. O caminho do banco de dados e o adaptador de armazenagem utilizados foram os padrões do Chatterbot e o preprocessador utilizado foi o mesmo desenvolvido para o *chatbot* teste na sessão 3.4.

Algoritmo 10 - Inicialização do chatbot E-MANUEL

```

1 chatbot = Chatterbot.ChatBot (
2     "E-MANUEL",
3     storage_adapter = "Chatterbot.storage.SQLiteStorageAdapter",
4     preprocessors = [
5         "Processador.pross"
6     ],
7     logic_adapters = [
8         {
9             "import_path" : "Adapters.BuscaDadosTec.BuscaDadosTec"
10            "bomba" : inicializar()
11        },
12        {
13            "import_path" : "Adapters.BuscaDadosLumb.BuscaDadosLumb",
14            "bomba": codigo
15        },
16        { "import_path" : "Adapters.Checklist.Checklist" },
17        {
18            "import_path" : "Chatterbot.logic.BestMatch",
19            "default_response" : "Sinto muito, não sou capaz de ajudar nisso",
20        }
21    ],
22     database_uri = "sqlite:///database.sqlite3"
23 )

```

As definições e a função `inicializar()` estão contidas no arquivo `assistente.py`. Os adaptadores lógicos estão em arquivos próprios para cada adaptador e agrupados em uma pasta *Adapters*.

A troca de mensagens é controlada pelo arquivo *main.py* que cria e mantém um looping que coleta a entrada do usuário pelo método `input()`, transfere para o *chatbot*, obtém a resposta e a exibe no terminal. Este *looping* é quebrado quando o usuário utilizar uma palavra de saída (fim, acabou, terminamos, finalizado) ou fechar o programa.

3.6.1 Implementação da busca de dados técnicos para o usuário

Para a interação inicial do E-MANUEL com o usuário foi criado uma função de inicialização que deve se comunicar com o usuário e seguindo o fluxograma apresentado na sessão 3.5 para obter o código da bomba a ser trabalhada, este código será utilizado pelo E-MANUEL como parâmetro de inicialização de

adaptadores lógicos. Finalizada esta interação inicial o usuário poderá acessar as demais funcionalidades do E-MANUEL.

A função de inicialização é acionada assim que o E-MANUEL é iniciado. Para realizar a sua função uma mensagem inicial de boas-vindas é exibido seguido da solicitação do código da bomba. Ao receber a primeira mensagem o E-MANUEL coleta a resposta do usuário e cria uma conexão com o banco de dados. Para realizar a conexão e comandos no banco de dados são utilizados algoritmos (Algoritmo 11) baseados na biblioteca `psycopg2`. Essa biblioteca Python permite realizar *queries* e comandos em bancos de dados PostgreSQL.

A primeira função da biblioteca `psycopg2` cria a conexão com o banco de dados por meio do comando `connect()` e passar esta conexão a uma nova variável fornecendo como parâmetros: usuário, senha e nome do banco de dados. Com a conexão criada é necessário o uso de um cursor criado por meio do método `cursor()`, o cursor tem por função realizar comandos ao banco de dados.

Com as variáveis necessárias para o acesso ao banco de dados criada o E-MANUEL realiza uma busca com a função `execute()` que recebe como parâmetro um comando SQL em forma de *string*, o comando SQL em questão retorna a lista dos códigos das máquinas cadastradas no sistema que deve ser extraída aplicando o comando `fetchall()`.

Algoritmo 11 - Extração de dados do banco de dados

```

1  resposta: str = input('você: ')
2  con = psycopg2.connect(
3      user = "postgres",
4      password = "23071998",
5      database = "Dados_bomba"
6  )
7  cur = con.cursor()
8  cur.execute('SELECT codigo FROM dados_tecnicos')
9  listaBombas = numpy.transpose(cur.fetchall())[0]
10 resp = separacaoPalavras(resposta)
11 resultadoBusca = chanceConter(resp, listaBombas)

```

Extraída a lista de bombas cadastradas, E-MANUEL utiliza a função `chanceConter` (Algoritmo 12) para comparar todas as palavras da entrada do usuário com cada elemento da lista de bombas retornando o código mais próximo e seu grau de similaridade (`inf_usuario`).

Algoritmo 12 - Pseudocódigo da função chanceConter

```

1  INÍCIO
2  |   LEIA (frase, listaPalavras)
3  |   REAL chance ← -1, match
4  |   CADEIA palavraEncontrada
5  |   PARA CADA palavra1 ∈ frase FAÇA
6  |   |   PARA CADA palavra2 ∈ listaPalavras FAÇA
7  |   |   |   match ← grau de proximidade entre palavra1 e palavra2
8  |   |   |   SE match > chance FAÇA
9  |   |   |   |   palavraEncontrada ← palavra2
10 |   |   |   |   chance ← match
11 |   |   FIM SE
12 |   FIM PARA
13 FIM PARA
14 RETORNA chance, palavraEncontrada
15 FIM

```

chanceConter() é uma função interna utilitária do E-MANUEL que recebe uma frase e uma lista de palavras, realiza uma comparação entre cada palavra da frase buscando por aquela que mais se assemelha a um item da lista. uma vez encontrada a função retorna a palavra da lista com maior similaridade e o seu grau de proximidade. O grau de proximidade entre palavras é calculado pela função *SequenceMatcher()* da biblioteca *difflib*.

Retornando para a função de inicialização, caso o código passado pelo usuário não seja válido o E-MANUEL executa um *looping* até que se tenha um código de bomba válido (Algoritmo 13). A cada repetição deste *looping* o E-MANUEL solicita um novo código, extrai novamente a lista de códigos cadastrados no banco de dados e executa a função chanceConter() retornando um novo valor de similaridade.

Caso o valor de similaridade seja 1, isso indica que o código informado pelo usuário é válido, encerrando o *looping*. Caso o valor seja menor que 1 e maior igual a 0.8 o código com maior valor de similaridade será informado para o usuário e será solicitada uma confirmação. Em caso o usuário confirme o código este será armazenado e retornado como código válido e o *looping* é encerrado, caso contrário o E-MANUEL considera o código como inválido.

Algoritmo 13 - Pseudocódigo de primeira interação

```

1  INÍCIO
2  ESCRIVA ("Olá, sou seu assistente virtual E-MANUEL")
3  ESCRIVA("chatbot: Com qual bomba trabalharemos hoje?")
4  LEIA(entrada)
5  conecta com banco de dados
6  bombas[] ← extrai lista bombas cadastradas
7  ENQUANTO (proximidade < 1) E (confirmacao = falso) FAÇA
8      proximidade ← proximidade de codigo o item de bombas[]
9      confirmacao ← falso
10     SE proximidade < 0.8 FAÇA
11         ESCRIVA("Não encontrei essa bomba no banco de dados, tente novamente")
12         LEIA(entrada)
13         conecta com banco de dados
14         bombas[] ← extrai lista bombas cadastradas
15         codigo ← item de bombas[] mais próxima a menção do usuário
16         proximidade ← proximidade de codigo o item de bombas[]
17     SENÃO
18         ESCRIVA("Eu encontrei ",codigo," , é esse que você quer?")
19         LEIA(entrada)
20         SE entrada.afirmativa() = verdadeiro
21             confirmacao ← verdadeiro
22         SENÃO
23             bombas[] ← extrai lista bombas cadastradas
24             codigo ← item de bombas[] mais próxima a menção do usuário
25             proximidade ← proximidade de codigo o item de bombas[]
26         FIM SE
27     FIM SE
28 FIM ENQUANTO
29 RETORNA codigo
30 FIM

```

Quando o usuário deseja alguma informação técnica presente no banco de dados, esta requisição passa para o adaptador lógico BuscaDadosTec. Na inicialização (Algoritmo 14) o adaptador recebe como argumento o código da bomba para futuro uso nas buscas no banco de dados, deste modo o adaptador entende sobre qual bomba deve ser feita a busca. Para que esta informação possa ser passada para o adaptador lógico é utilizado uma *string* interna 'bomba' que armazena este código, o seu valor é definido na função de inicialização.

Algoritmo 14 - Inicialização de BuscaDadosLub

```
class BuscaDadosLub(LogicaAdapter):
    bomba : str

    def __init__(self, chatbot, **kwargs):
        super().__init__(chatbot, **kwargs)
        self.bomba = kwargs.get('bomba')
```

No processamento, a primeira etapa é a conexão com o banco de dados para isso foi utilizada a biblioteca **psycopg2** do Python para consultas por meio de query. Com a conexão criada o adaptador lógico passa a processar a mensagem (Algoritmo 15) buscando por palavras-chave de três grupos distintos: palavras que remetam a uma pergunta, a palavra que indiquem *intent* e palavras de contexto. Ao obter a *intent* o código realiza uma consulta no banco de dados para poder retornar a informação solicitada pelo usuário. Dependendo da solicitação exigida o adaptador lógico se utiliza de um modelo personalizado para cada informação do banco de dados de acordo com o Quadro 3 da sessão 3.5.1.

Algoritmo 15 - Pseudocódigo de BuscaDadosTec

```
1  INÍCIO
2  LEIA (frase, listaPalavras)
3  conecta com banco de dados
4  STR intents[ ] ← lista de intents
5  intencao ← encontra item de intents[ ] mais próxima ao objetivo do usuário
6  retorno ← busca no BD pelo dado correspondente ao valor da intenção
7  STR saida ← frase criada baseada no retorno
8  confidence ← calcula_chance do adaptador logico responder bem o usuário
9  statment ← novo Statement(saida, confidence)
10 RETORNA statment
11 FIM
```

Na linha 8 ocorre o cálculo de confiança pela equação 1. A fórmula foi balanceada experimentalmente por meio de testes em *statements* críticos. O resultado indica a chance do adaptador lógico responder corretamente o usuário.

$$confidence = (chanceInt * 0.6 + chanceDuvida * 0.4 - chanceLub * 0.2) * Ajuste \quad (1)$$

Na fórmula, **chanceInt** é o grau de similaridade da *intent* encontrada, este é o fator de maior relevância no cálculo. Foram utilizados graus de similaridade ao invés de absolutos para que palavras com erros de digitação possam ser reconhecidas.

O segundo fator **chanceDuvida** qualifica quanto a similaridade a palavras que remetem a questionamentos (dúvidas do usuário). Expressões desse tipo demonstram a busca do usuário por informações e distinguem de frases afirmativas. Mesmo assim, o pedido por informações pode chegar para o *chatbot* sem o emprego dessas expressões, como: “preciso do tamanho do rotor” ou “me diga a velocidade”. Devido a isso **chanceDuvida** deve ser usado apenas de forma complementar para tornar o cálculo mais preciso.

O fator **chanceLub** indica a proximidade com a palavra lubrificação, este fator busca identificar o contexto trazido pelo usuário.

O valor máximo que pode ser assumido pela variável **confidence** é 1, com isso os valores de **chanceInt** e **chanceDuvida** devem ser complementares com peso maior para **chanceInt**.

Ajuste é um fator adicionado a confiança dos adaptadores lógicos para evitar interferência com o BeastMach, o adaptador lógico responsável pelos processos de *Deep Learning*. Esse fator se mostrou necessário uma vez que palavras-chave dos adaptadores lógicos eram mencionadas em diálogos treinados em *Deep Learning* de modo que as buscas de informação eram acionadas em momentos indevidos. O valor da variável foi definido realizando testes com as frases mais conflitantes onde as palavras-chave de *intents* eram mencionados em contextos tratados por *Deep Learning* e com algumas frases de busca visando maximizar a taxa de acerto.

Por fim, o algoritmo encerra a conexão por meio do comando *close()* e retorna um novo *Statement* com o texto contendo a informação solicitada, utilizando a proximidade com as palavras de entrada para definir o valor de *confidence* da resposta gerada.

Para a busca por dados sobre lubrificação foi desenvolvido o adaptador lógico **BuscaDadosLub** nos mesmos moldes de **BuscaDadosTec**, porém a busca ocorre na tabela **lubrificacao**. Nesta aplicação os grupos de palavras referentes a pergunta sobre lubrificação permanecem os mesmos, porém as *intents* empregadas

serão agora referentes a buscas que o usuário poderá fazer relacionados a lubrificação, tais como frequência de troca de óleo e marca do lubrificante. A partir de então o processo segue similar ao procedimento para BuscaDadosTec. Ao final, no cálculo de confiança do *Statement* o coeficiente de **chanceLub** passa a ser positivo e não mais negativo como visto em BuscaDadosTec.

3.6.2 Implementação do Adaptador Lógico *checklist*

A última funcionalidade desenvolvida em adaptador lógico é a realização de uma *checklist* que o operador deverá realizar ao fim da execução para garantir as condições de operação.

Esta operação é iniciada assim que o adaptador detectar a *intent* checklist. Diferente dos demais casos, esta detecção inicial é realizada pelo método *canProcess()* característico dos adaptadores lógicos do Chatterbot. Este método retorna um booleano responsável por indicar se o adaptador lógico deve ou não processar a mensagem enviada, sendo este um recurso para poupar processamento da máquina.

O adaptador lógico contém variáveis como: um dicionário '*checkList*' contendo os passos de inspeção e os booleanos referentes ao '*check*', uma *string* '*currentKey*' que armazena o próximo passo a ser executado e um booleano '*processando*' que armazena se o processo de inspeção está em andamento. Quando iniciado (Algoritmo 16) o adaptador lógico inicializa todos os valores booleanos como *False* e passa o primeiro passo para a variável '*currentKey*'.

Algoritmo 16 - Inicialização de *checkList*

```
def __init__(self, chatbot, **kwargs):
    super().__init__(chatbot, **kwargs)
    self.processando = False
    for v in self.checkList.values(): v = False
    self.currentKey = 'lubrificante'
```

Dentro da função *canProcess* (Algoritmo 17) o adaptador analisa a frase e busca uma palavra com grau de compatibilidade de 80% com a palavra *checklist*, caso encontrado e caso o valor de '*processando*' seja *False* a variável '*processando*' recebe

o valor *True* indicando o início do processo. A função retorna o valor de ‘processando’ indicando se o processo de inspeção está em andamento.

Algoritmo 17 - Pseudocódigo do método `canProcess()` do adaptador lógico *checklist*

```

1  INÍCIO
2  |  LEIA (frase, listaPalavras)
3  |  REAL match ← maior grau de proximidade entre palavras de entrada e 'checklist'
4  |  SE match > 0.8 E processando = falso FAÇA
5  |  |  processando ← verdadeiro
6  |  FIM SE
7  |  RETORNA processando
8  FIM

```

Iniciado o processo de inspeção, o adaptador lógico *checklist* passa a processar as mensagens (Algoritmo 18). A variável ‘*currentKey*’ faz o papel de ponteiro referenciando a etapa atual da inspeção. A cada nova afirmação passada pelo usuário, três grupos de *intents* são procuradas, uma afirmativa como ‘sim’ e ‘certo’ que indicará ao E-MANUEL que a etapa está concluída, uma negativa como ‘não’ que deverá anular qualquer afirmação, e referências a etapas do *checklist* caso o usuário passe outro item que não o atual. Caso o usuário se refira a outra etapa sem o emprego do negativo (“não”), E-MANUEL ‘assinala’ a etapa mencionada ao invés da atual.

Algoritmo 18 - Pseudocódigo do adaptador lógico *checklist*

```

1  INÍCIO
2  |   BOOLEANO completo ← True
3  |   CADEIA[] afirmativos ← ['ok','certo','verificado','conferido','feito','sim','está']
4  |   CADEIA verificado ← item de afirmativos[] mais próxima a menção do usuário
5  |   REAL grauVerificado ← maior grau de proximidade entre palavras de entrada e verificado
6  |   CADEIA mencao ← item de checkList.chaves() mais próxima a menção do usuário
7  |   REAL grauMencao ← maior grau de proximidade entre palavras de entrada e mencao
8  |   CADEIA resposta
9  |   SE proximidade < 0.8 FAÇA
10 |       |   currentKey ← mencao
11 |   FIM SE
12 |   SE grauVerificado > 0.8 FAÇA
13 |       |   checkList[currentKey] ← verdadeiro
14 |       |   currentKey ← próximo passo
15 |   FIM SE
16 |   PARA CADA chave ∈ checkList.chaves() FAÇA
17 |       |   completo ← completo E checkList[chave]
18 |   FIM PARA
19 |   SE completo:
20 |       |   processando ← False
21 |       |   resposta ← 'Tudo conferido'
22 |   SENÃO
23 |       |   proximidade ← proximidade de codigo o item de bombas[]
24 |   FIM SE
25 |   RETORNA codigo
26 FIM

```

Caso uma nova etapa não seja mencionada, haja menção positiva (“sim”, “correto”, “certo”, etc.) e não negativa (“não”) E-MANUEL muda o valor do booleano referenciado para *True*, avança etapa e indica esta etapa ao usuário.

3.7 TREINAMENTO DO *CHATBOT* EM AÇÕES DE MANUTENÇÃO

O suporte em ações de manutenção foi implementado no *chatbot* com o adaptador lógico BestMatch, que requer a criação de diálogos para treinamento (Algoritmo 19). Para isso foram criadas bases de diálogos que iniciam com o pedido do usuário por orientações e seguem por uma sequência passo a passo de operação

do *chatbot* e indicação de conclusão por parte do usuário. Ao todo foram utilizados 57 diálogos reunidos no apêndice H.

Algoritmo 19 - Exemplo de caixa de diálogo

```

instalacao1 = [
    "preciso instalar a bomba",
    "ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
    "fundação pronta",
    "certo, agora garanta que a base da bomba esteja devidamente nivelada e então fixe-a bem com argamassa",
    "base pronta",
    "espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
    "parafusos apertados",
    "se as tubulações encostarem sem esforço nas flanges, fixe-as",
    "tubulações fixadas",
    "agora fixe cuidadosamente o acoplamento ao rotor da bomba",
    "acoplamento fixado",
    "ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
    "motor pronto",
    "tudo certo então, algo mais em que eu possa ajudar?"
]

```

A função *train()* do Chatterbot foi utilizada para o treinamento tendo como parâmetros o próprio *chatbot* e os diálogos desenvolvidos e passados para o arquivo *treino.py*. Para limpar a base de dados de treinamento interno do *chatbot*, deve ser inserido o método *chatbot.storage.drop()*.

Durante o treinamento foi percebida uma falha do *chatbot* onde durante o desenvolvimento foi realizado um treinamento utilizando entradas do usuário como 'Feito', porém ao observar o comportamento do E-MANUEL sempre que era utilizada a entrada 'Feito' a resposta permanecia a mesma 'ótimo, agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C', nunca passando ao passo seguinte. Desta forma, foi observado que o uso de sentenças simples como "terminei" ou "já fiz" gerava confusão no assistente sobre como deveria responder por não ter informação de contexto (tema da instrução anterior). Neste caso, o usuário sempre deve usar sentenças completas informando qual foi o passo realizado.

Com os diálogos treinados, o próximo passo envolveu uma melhoria incremental. Com essa finalidade, foram realizados testes com o *chatbot* para orientar nas operações treinadas. Quando ele não respondia apropriadamente, essa falha era anotada. Em seguida, novos diálogos para treinamento foram feitos, mantendo as falas referentes ao *chatbot* inalteradas, porém, substituindo as mensagens do usuário por aquelas que não foram devidamente respondidas pelo programa, e os passos foram repetidos até o *chatbot* não apresentar mais desvios de orientação.

3.8 METODOLOGIA DE TESTES

Nove novas bombas centrífugas foram incorporadas ao banco de dados (Figura 13) para realizar testes abrangentes do sistema E-MANUEL em diferentes cenários. Ao inserir esses dados na tabela `dados_tecnicos`, a convenção de codificação com duas letras seguidas de quatro algarismos foi mantida, garantindo a integridade do código. Os modelos e fabricantes foram selecionados com base em bombas reais, enquanto os demais dados foram cuidadosamente criados para refletir cenários plausíveis.

Figura 13 - Tabela `dados_tecnicos`

	<code>codigo</code> [PK] character varying (6)	<code>empresa</code> character varying (25)	<code>modelo</code> character varying (255)	<code>tm_rotor</code> double precision	<code>velocidade</code> integer	<code>produto</code> character varying (100)
1	ET5678	KSB	KSB Firebloc 32-125	125	3500	Halon
2	NM8379	KSB	Meganorm 125-80-400	400	1450	Alcool
3	TC2709	schneider	BC-23 R 1 1/2	225	960	Água
4	PT1322	DANCOR	37-40 TJM	143	3500	Água
5	MD1120	KSB	Meganorm 200-200-250	250	1160	Alcool
6	AB1234	KSB	KSB ETA 125-40	400	3500	Água
7	XZ0987	KSB	KSB ETA 200-23	230	1750	Água
8	TT2332	KSB	KSB ETA 100-20	200	1800	Condensado
9	GA2005	KSB	KSB ETA 50-26	260	1800	Água
10	JB1722	KSB	KSB ETA 200-23	230	3400	Condensado

Fonte: autoria própria

Para os registros na tabela de lubrificação (Figura 14), foram empregados dados reais referentes à marca e tipo de lubrificante, enquanto os demais dados foram preenchidos com informações fictícias, porém plausíveis.

Figura 14 - Tabela Lubrificação

	cd_bomba character varying (6)	tp_oleo character varying (255)	marca character varying (100)	quantidade double precision	frequencia_troca integer
1	AB1234	marbrax tr 52	PETROBRAS	1.2	365
2	ET5678	marbrax tr 52	PETROBRAS	0.7	90
3	XZ0987	Duro AW 58	ATLANTIC	1.2	330
4	NM8379	CLP-46	Optimal Oil	0.4	175
5	TT2332	Ipitur AW 68	IPIRANGA	0.55	334
6	TC2709	SAE 30	Lubrax	1	90
7	GA2005	Rigal oil RO 100	TEXACO	0.4	345
8	PT1322	CLP-46	Optimal Oil	0.82	300
9	JB1722	Mobil D. T. E. - 26	MOBIL OIL	1.2	365
10	MD1120	CLP-46	Optimal Oil	0.7	84

Fonte: autoria própria

Para analisar a qualidade de um *chatbot* várias métricas podem ser aplicadas, para avaliar a interação com o usuário. Dentre elas, destacam-se (ALHASSAN *et al.*, 2022; VIJAYARAGHAVAN; COOPER, 2020):

Relevância – se refere a capacidade do chatbot em fornecer respostas pertinentes e uteis ao usuário, uma resposta relevante é aquela que fornece informações uteis ao problema apresentado.

Integridade – se refere à consistência e confiabilidade das respostas fornecidas pelo sistema ao longo das interações com o usuário, mantendo uma linha de diálogo coesa e não contraditória.

Precisão – a precisão se refere à capacidade do sistema de fornecer respostas corretas e exatas para as perguntas e solicitações dos usuários fornecendo fatos factualmente corretos.

Tempo de resposta – diz respeito a quanto tempo o programa leva para responder a entrada do usuário.

Taxa de resolução – é uma métrica que indica a taxa com a qual o chatbot está solucionando com sucesso o problema do usuário, sendo medida pela razão entre o número de soluções do chatbot pelo número total de atendimentos.

Uma maneira comum de coletar essas informações é através do *feedback* de voluntários ou usuários que interagem com o *chatbot* e preenchem questionários de satisfação. No entanto, quando não há voluntários disponíveis para testes, é viável

realizar testes automatizados usando conjuntos de dados que simulam interações reais.

Trabalhos anteriores em desenvolvimento de chatbots destacam a taxa de resolução como principal métrica para avaliação de chatbots (ALHASSAN *et al.*, 2022; VIJAYARAGHAVAN; COOPER, 2020).

Para avaliar o chatbot E-MANUEL foram utilizadas as métricas: resolução, relevância e tempo de resposta. Os testes foram realizados com o auxílio da biblioteca *unittest* do Python. *Unittest* é uma biblioteca voltada para automatização de testes unitários em que é possível simular entradas e verificar de forma automática se as saídas obtidas estão de acordo com o esperado. O módulo *unittest* compreende alguns conceitos importantes, são eles:

I. Definição de contexto de teste

Uma definição de contexto de teste representa a preparação necessária pra configurar e executar um ou mais testes, além de quaisquer ações de limpeza relacionadas.

II. Caso de teste

Um caso de teste é uma unidade de teste individual, que verifica uma resposta específica a um determinado conjunto de entradas. O *unittest* fornece uma classe base, *TestCase*, que pode ser usada para criar novos casos de teste.

III. Coleção de Testes

Uma coleção de testes pode ser uma coleção de casos de teste, conjuntos de teste ou ambos. Utiliza-se para agregar testes que devem ser executados juntos.

IV. Test runner

Um *test runner* é um componente que gerencia a execução de testes e fornece o resultado para o usuário. O *runner* pode usar uma interface gráfica, uma interface textual ou um componente que retorna um valor especial para indicar os resultados da execução dos testes.

Os testes foram implementados e alocados no arquivo `testes.py` cujo código fonte se encontra no apêndice I. Na definição de contexto de teste, E-MANUEL é inicializado e o arquivo `treino.py` é executado. Dentre a coleção de testes, cada caso de teste envolve a realização de uma tarefa por parte de E-MANUEL, simulando

entradas e coletando as respostas. O *test runner* deve então realizar toda a coleção de testes e retornar o número e os casos de falha.

Para os casos de busca de dados (Algoritmo 20), apenas uma entrada é utilizada e a saída é comparada de acordo com a frase e informação esperadas.

Ao todo foram realizados 400 testes de busca de dados utilizando-se diferentes formas de solicitação destes dados. As entradas e saídas esperadas estão documentadas no apêndice J.

Algoritmo 20 - Exemplo de teste de busca de dados

```
1 def test_busca_rotor1(self):  
2     resposta = self.chatbot.get_response('Qual o tamanho do rotor?')  
3     self.assertEqual(resposta.text, 'O rotor mede 40.0cm')
```

Para os demais casos que avaliam processos com mais de uma troca de mensagens (Algoritmo 21) são criadas três listas por teste, uma contendo as entradas do usuário que devem ser enviadas em ordem para o E-MANUEL, uma que armazena as respostas coletadas para cada entrada e uma lista de saídas esperadas. Em caso de divergência entre a lista de respostas coletadas e respostas esperadas, o teste retorna falha. Para estes casos foram realizados ao todo 250 testes, as entradas e saídas esperadas estão documentadas no apêndice J.

Em caso de sucesso, considera-se que o E-MANUEL foi capaz de fornecer uma resposta correta conforme o solicitado. No entanto, em caso de falha, a resposta do chatbot é avaliada. Se a resposta incorreta vier do mesmo adaptador lógico ou do mesmo grupo de diálogo de treinamento, considera-se que o E-MANUEL forneceu uma resposta relevante.

Todos os testes foram repetidos para cada bomba centrífuga presente no banco de dados, realizando as adaptações necessárias ao substituir os dados esperados de saída pelos dados registrados no banco de dados. Os relatórios dos testes foram registrados nos apêndices K, L, M, N, O, P, Q, R, S, T, e as análises foram compiladas em uma tabela.

Algoritmo 21 - Exemplo de testes com múltiplas trocas de mensagens

```

1  def test_instalacao_bomba1(self):
2      instalacao1 = [
3          "preciso instalar a bomba",
4          "fundação pronta",
5          "base pronta",
6          "parafusos apertados",
7          "tubulações fixadas",
8          "acoplamento fixado",
9          "motor pronto"
10     ]
11     conversa_esperada = [
12         "ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
13         "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa",
14         "espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
15         "se as tubulações encostarem sem esforço nas flanges, fixe-as",
16         "agora fixe cuidadosamente o acoplamento ao rotor da bomba",
17         "ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
18         "tudo certo então, algo mais em que eu possa ajudar?"
19     ]
20     mensagens_chatbot = []
21     for mensagem in instalacao1:
22         resposta = self.chatbot.get_response(mensagem)
23         mensagens_chatbot.append(resposta.text)
24     self.assertEqual(mensagens_chatbot, conversa_esperada)

```

Para testes de tempo de resposta foi utilizada a função *time()* da biblioteca Time que retorna o tempo em segundos entre o instante atual e a meia-noite do dia 1º de janeiro de 1970, conhecido como o tempo Unix ou *Epoch time*. Para a medição de tempo foram utilizadas uma variável de início, que armazena o momento em que o processo começa, e uma de fim, que armazena o momento em que o processo finaliza sendo a diferença destas duas variáveis a representação do tempo de resposta.

Estes testes foram realizados em um arquivo TesteTempo.py (Apêndice U) que realiza um *looping* realizando a mesma interação 42 vezes para cada categoria de interação (Busca de dados técnicos, processo de inspeção/checklist, guia de processos de manutenção) entre o usuário e E-MANUEL, estes dados são retornados pelo arquivo TesteTempo.py e coletados para análise.

Durante os testes foi observado um atraso (*delay*) após o E-MANUEL receber o código da bomba informada pelo usuário, para estes casos foi realizada uma análise adicional.

4 RESULTADOS E DISCUSSÕES

Por se tratar de um protótipo, não foram implementados elementos de *front-end* em E-MANUEL, assim a troca de mensagens durante a interação é realizada via terminal com indicações do E-MANUEL e do usuário como apresentado na Figura 15.

Figura 15 - Conversa com o E-MANUEL

```
Iniciando assistente ...
chatbot: Olá, sou seu assistente virtual E-MANUEL e vou te ajudar com sua manutenção
chatbot: Com qual bomba trabalharemos hoje?
você: Olá, vamos trabalhar com a AB-1234
chatbot: Eu encontrei AB1234, é esse que você quer?
você: Essa mesmo
chatbot: Então hoje trataremos da bomba AB1234, aguarde um pouco enquanto preparo o ambiente...
chatbot: Ambiente pronto, como posso ajudá-lo?
você: estamos precisando trocar o rolamento
chatbot: para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves
você:
```

Fonte: Acervo pessoal

Realizados todos os testes automatizados, os relatórios foram registrados no apêndice K e compilados de acordo com a relevância e resolução conforme Tabela 1.

Tabela 1 - Distribuição dos testes e resultados

Função	Especificação	Nº de testes	Nº relevância	Nº sucessos	Nº falhas
Busca de dados técnicos	Tamanho de rotor	40	40	40	0
	Velocidade de rotação	40	40	40	0
	Produto de trabalho	40	40	40	0
	Modelo de bomba	40	40	40	0
	Empresa fabricante	40	40	40	0
Busca de dados de lubrificação	código	40	20	20	20
	Frequência de troca	40	40	40	0
	fabricante	40	40	40	0
	Volume necessário	40	40	40	0
	Marca	40	40	40	0
Checklist	Checklist	40	40	30	10
	Instalação	40	40	20	20
	montagem	40	40	40	0
	desmontagem	40	30	30	10
	troca de rolamento	40	40	40	0
Processos de manutenção	desmontar cavalete	10	10	10	0
	falha de rolamento	20	10	10	10
	falha de lubrificação	20	10	20	0
	Total	650	600	580	70

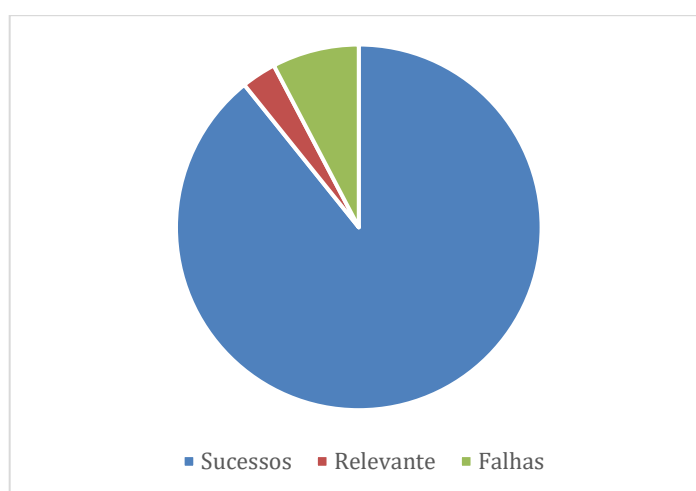
Fonte: autoria própria

Na tabela 1, a coluna N° de testes representa o total de testes realizados em cada categoria (função), a coluna N° relevância representa o número de testes que retornaram respostas relevantes, a coluna N° sucessos representa a quantidade de testes que retornaram as respostas esperadas e a coluna N° falhas representa o total de testes que retornaram falha, ou seja, não retornaram as respostas esperadas.

Os relatórios obtidos pelas funções do Unitest indicaram que para testes com mesma entrada de usuário, porém bombas diferentes, era obtida a mesma saída, assim sendo os mesmos erros e acertos foram observados em todas as bombas.

Os testes que retornaram falso foram em diálogos de busca pelo código da bomba, instruções para instalação da bomba, processo de desmontagem e no processo de *checklist*. A Figura 16 apresenta a taxa de resolução e taxa de relevância obtidas na bateria de testes.

Figura 16 - Distribuição dos Resultados dos Testes



Fonte: autoria própria

Pelos testes podemos observar que o E-MANUEL obteve uma taxa de sucesso de 89,2%, sendo que em apenas 7,7% das interações o E-MANUEL não foi capaz de atribuir resposta dentro do contexto esperado.

Nos casos de erros no *checklist* (Figura 17) observou-se que as respostas fornecidas pelo E-MANUEL permaneceram relacionadas às etapas da inspeção, ou seja, mantiveram-se dentro do adaptador lógico do *checklist*. No entanto, a resposta fornecida representava uma etapa incorreta da inspeção, o que indica que o E-MANUEL compreendeu a necessidade de passar por uma etapa da *checklist*, mas não identificou qual etapa específica.

Figura 17 - Erro no teste do processo de *checklist*

```

Failure
Traceback (most recent call last):
  File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 733, in
test_checklist3
    self.assertEqual(resposta.text, resposta_esperada)
AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar
livremente?'
- O motor está devidamente acoplado a bomba?
+ O eixo consegue movimentar livremente?

```

Fonte: Acervo pessoal

Durante o desenvolvimento, optou-se pelo uso de palavras-chave para extrair as intenções do usuário quando o processamento se encontrava dentro de adaptadores lógicos, porém um dos efeitos negativos encontrados foram conflitos quando era mencionado o ‘nível de óleo’ uma vez que a palavra ‘nível’ em alguns casos acabava por ser relacionada ao nivelamento da máquina ao invés de ser reconhecido como ‘volume do lubrificante’. Isso causou um dos equívocos, mais precisamente no processo de *checklist*, onde tanto o nível de óleo quanto o nivelamento do conjunto bomba-motor foram confundidos.

Uma possível forma de solucionar este problema dentro dos próprios adaptadores lógicos seria, ao invés de usar palavras-chave para entender a intenção do usuário, utilizar exemplos e modelos de entradas para o E-MANUEL se basear. Porém, para o uso desta solução, é imprescindível que todas as possibilidades de diálogos utilizados para a busca de resposta sejam inseridas no adaptador lógico para que assim o usuário sempre tenha a resposta desejada independente da forma que pergunte.

Outros dois erros foram encontrados na busca pelo código da bomba sendo um deles apresentado na Figura 18, em ambos os casos os retornos fornecidos pelo E-MANUEL não se encaixaram com o contexto da busca.

Figura 18 - Erro na busca pelo código da bomba

```
Failure
Traceback (most recent call last):
  File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 104, in
test_busca_codigo3
    self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')
AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está
registrada como AB1234'
- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento
+ Essa bomba está registrada como AB1234
```

Fonte: Acervo pessoal

Outras falhas vistas nas interações ocorreram durante o guia do processo de instalação da bomba gerando erros similares ao apresentado na Figura 19. Nesta situação a resposta esperada era 'espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente', porém a resposta recebida 'se as tubulações encostarem nas flanges sem esforço, fixe-as' é referente a etapa posterior do processo. Em ambos os casos apesar de a resposta estar incorreta o E-MANUEL se manteve no processo de instalação na bomba.

Aqui o problema ocorre devido a dois fatores: primeiramente este processo foi passado ao E-MANUEL via treinamento, porém nestes casos o *chatbot* apenas correlaciona a recente fala do usuário com a respectiva resposta, não guardando contexto ou os demais diálogos desta conversa. Para este caso em específico o E-MANUEL não é capaz de guardar qual etapa e procedimento estão sendo realizados.

Figura 19 - Erro de teste unitário na instalação

```

Failure
Traceback (most recent call last):
  File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 249, in
test_instalacao_bomba2
    self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)
AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[167
chars]0mm'] != ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem nas flanges s[15 chars]e-as'

Diff is 866 characters long. Set self.maxDiff to None to see it.

```

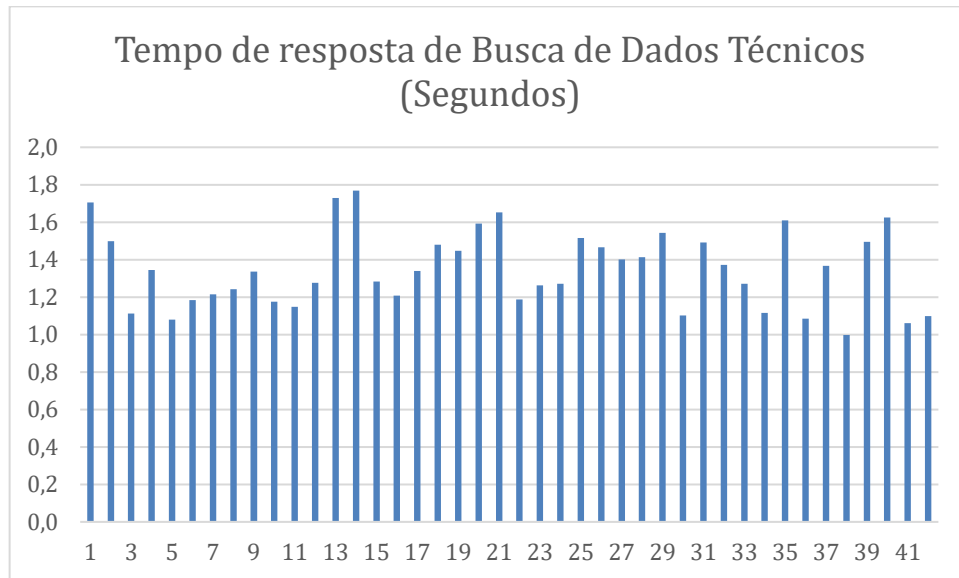
Fonte: Acervo pessoal

O outro fator diretamente correlacionado, diz respeito a falas muito sucintas nas quais o E-MANUEL consegue correlacionar a entrada com múltiplas respostas. Para contornar este problema, o E-MANUEL responde algumas falas como ‘acabei’ e ‘pronto’ com ‘qual foi a última coisa que fizemos?’. Porém falas menores como ‘base fixa’ ainda podem gerar esta confusão.

Uma forma de combater este problema seria tratando estas operações também em adaptadores lógicos, de modo que os contextos possam ser melhor guardados e as informações fornecidas anteriormente se mantenham registradas. As maiores dificuldades desta abordagem é que o desenvolvimento de adaptadores lógicos exige maior tempo de desenvolvimento e abre mão da principal funcionalidade fornecida pelo Chatterbot, o aprendizado de máquina.

Os resultados dos testes de tempo foram separados de acordo com a categoria da interação, o tempo obtido em interações de busca de dados são exibidos no gráfico da Figura 20.

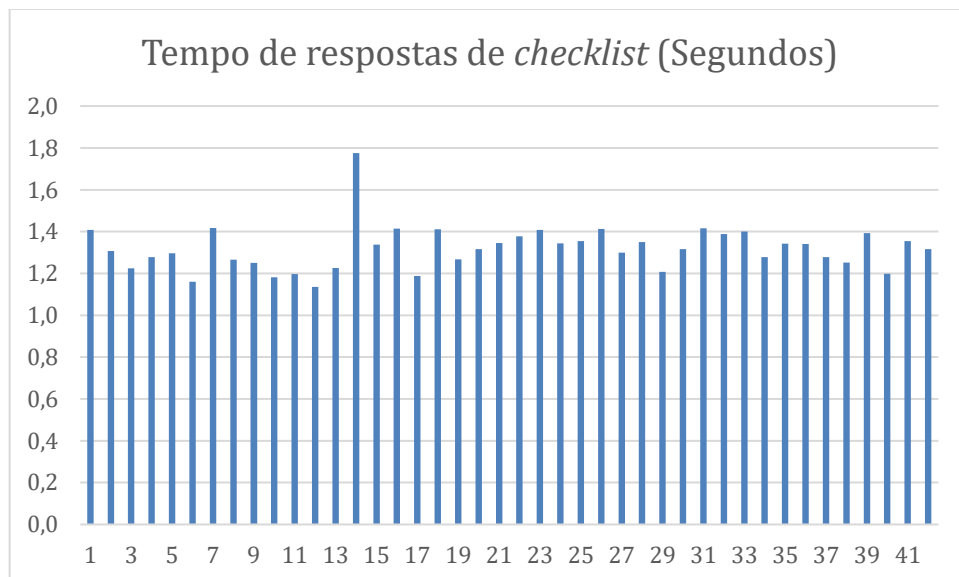
Figura 20 - Dados de tempo de resposta para interações de busca por dados técnicos



Fonte: autoria própria

O tempo médio calculado baseado nos tempos de resposta registrados que o chatbot leva para retornar valores na busca de dados técnicos gira em torno de 1,34 segundos, nunca ultrapassando os 1,8 segundos. Já os dados obtidos para interações de *checklist* são apresentados no gráfico da Figura 21.

Figura 21 - Tempo de resposta para interações relacionadas a *checklist*

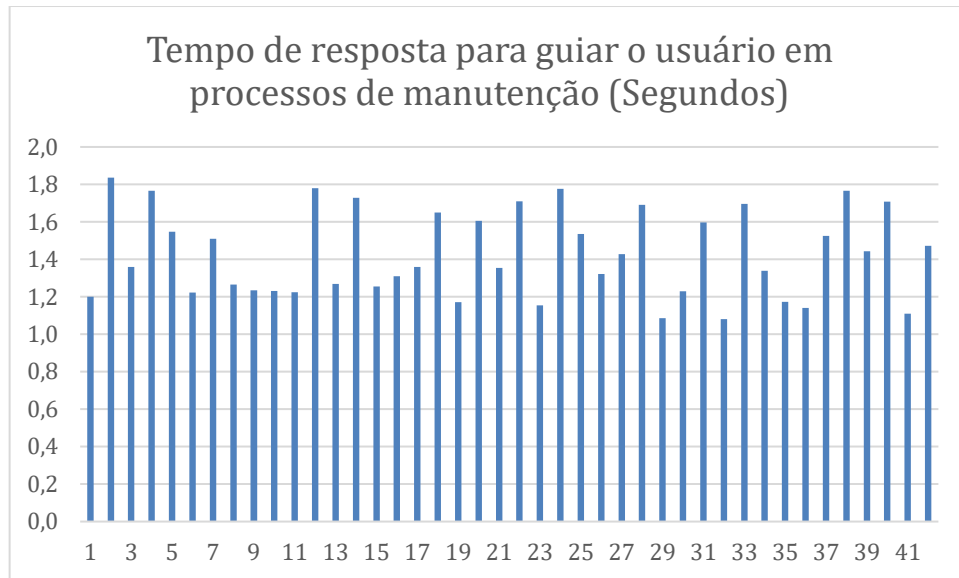


Fonte: autoria própria

Nesta interação é obtida em um tempo médio de 1,31 segundos, apenas 3 centésimos de segundo mais rápido e novamente sem ultrapassar os 1,8 segundos

de tempo de resposta. As interações relacionadas a guiar o usuário em processos de manutenção tiveram os tempos de resposta apresentados no gráfico da Figura 22.

Figura 22 - Tempo de resposta para guiar o usuário em processos de manutenção

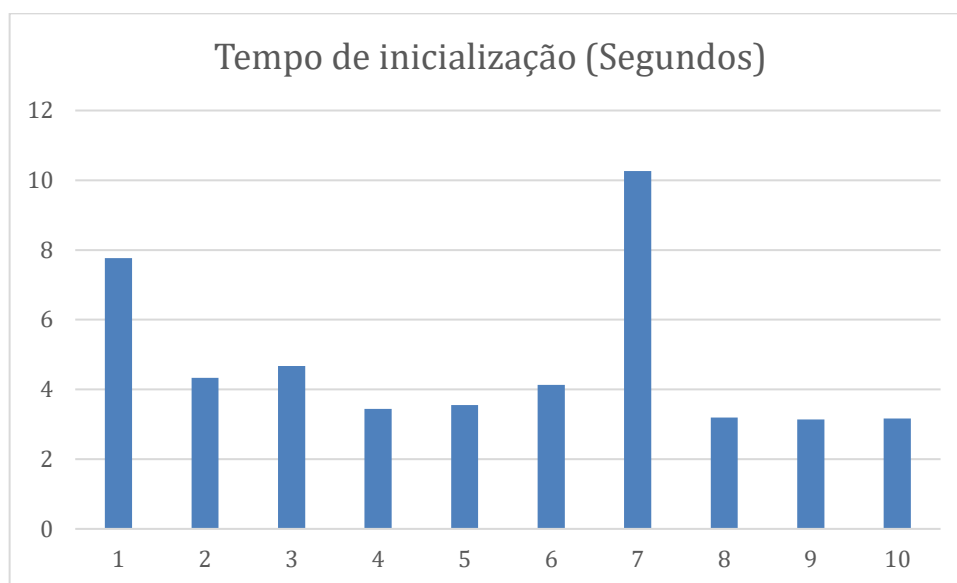


Fonte: autoria própria

Estas interações são as que apresentaram maior tempo de resposta, com uma interação que ultrapassou os 1,8 segundos e com tempo médio de resposta de 1,43 segundos. Com estes dados coletados é possível concluir que o tempo médio de resposta encontrado é de 1,36 para as interações do usuário com o chatbot.

Já a sua inicialização, e consequentemente a sua primeira interação, apresentou os dados coletados no gráfico da Figura 23.

Figura 23 - Tempo de inicialização



Fonte: autoria própria

Esta inicialização representa a ação mais demorada do E-MANUEL, levando sempre mais de 3 segundos para ser finalizada e podendo levar até mesmo acima dos 10 segundos. Para esta interação o E-MANUEL leva em média 4,77 segundos.

Como considerações finais uma funcionalidade cogitada no início do projeto foi a realização de diagnósticos de falha baseado nas entradas fornecidas pelo usuário, porém esta funcionalidade foi descartada, pois este diagnóstico não poderia ser realizado dentro das ferramentas do Chatterbot e teria de ser desenvolvida separadamente, fugindo do escopo original do projeto que é o *chatbot*. Porém uma possível melhoria seria uma integração do E-MANUEL com ferramentas de diagnóstico para expandir a sua utilidade para não apenas guiar na avaliação, mas também fornecer prognósticos das falhas e condições da bomba.

Também, em trabalhos futuros, o E-MANUEL poderia ser aplicado a empresas reais de modo que seu desempenho em manutenção possa ser melhor avaliado, além de o E-MANUEL poder ser alimentado com procedimentos comuns e padrão da empresa e sua busca em banco de dados ser melhor adaptada para bancos de dados empresariais.

5 CONCLUSÃO

A proposta de um sistema de suporte à equipe técnica de manutenção industrial foi implementada na forma de um chatbot e com um pequeno banco de dados fictício. Os testes sobre as interações com o usuário mostraram que E-MANUEL obteve uma taxa de sucesso de 89,2%, e apenas em 7,7% delas não foi capaz de gerar resposta dentro do contexto esperado. Uma das principais funcionalidades do Chatterbot Python é a capacidade de treinar o chatbot com conjuntos de dados específicos. Porém, essa experiência precisa ser mais explorada no domínio da manutenção, uma vez que a base de diálogos para treinamento se mostrou restrita. Um dos motivos foi a indisponibilidade de um banco de dados industrial real. O outro motivo talvez seja pelo caráter da aplicação que se propõe a guiar o usuário a ações corretas e por isso o modelo não permita muitas variações nas respostas. A parte disso, Chatterbot se mostrou uma poderosa ferramenta com a possibilidade do chatbot ser melhorado de forma incremental à medida que novos diálogos expandam sua base de conhecimento. Por fim para uma melhor avaliação e aplicabilidade da ferramenta sugere-se uma futura parceria com indústrias para adequar E-MANUEL para uma situação real de aplicação.

Como futuras melhorias ao E-MANUEL para trabalhos futuros são apontados:

- Transformar o diálogo sobre checklist de bombas centrífugas modelo KSB-ETA em uma única resposta do chatbot, ou seja, na forma de um conjunto de procedimentos sugeridos ao usuário. Dica para implementação:

- Retirar o adaptador logico checklist, inserir o novo diálogo em treino.py e treinar o chatbot por machine learning.

- Transformar a série de diálogos para montagem, desmontagem e instalação de bombas em uma única resposta do chatbot na forma de conjunto de procedimentos indicados ao usuário. Dica para implementação:

- Alterar/Eliminar as caixas de diálogo existentes no arquivo treino.py e treinar o chatbot por machine learning.

- Implementar identificação de contexto no diálogo. Dica para implementação:
 - Implementar no pre-processador (processador.py) busca de palavras-chave (código da bomba por exemplo) e tratar de uma das formas: responder diretamente ao usuário ou passar o contexto via objeto statement para os adaptadores lógicos.
- Implementar interface Web para o usuário com a ferramenta Django.
- Implementar o histórico de manutenção. Dicas para implementação:
 - Criar tabela no banco de dados para armazenamento dos dados históricos de manutenção.
 - Desenvolver novo adaptador logico que busque no BD da empresa, dados de manutenção (por ex. registro de datas das trocas de óleo, volume de óleo adicionado, histórico de incidentes de falhas, histórico de limpeza, calibração, etc.)
 - A partir de dados históricos, sugere-se a implementação de sistema preditivo de manutenção.
 - A partir de dados históricos, sugere-se a implementação de sistema de diagnóstico de falhas.
- Adequar o chatbot para uma planta industrial conhecida ou simulada para testes adicionais.

REFERÊNCIAS

- ADAMOPOULOU, E.; MOUSSIADES, L. Chatbots: history, technology and applications. **Machine learning with applications**, Amsterdam, v. 2, p. 1-18, dec. 2020a. DOI: <https://doi.org/10.1016/j.mlwa.2020.100006>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666827020300062?via%3Dihub>. Acesso em: 30 abr. 2023
- ADAMOPOULOU, E.; MOUSSIADES, L. An overview of chatbot technology. In: IFIP INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE APPLICATIONS AND INNOVATIONS, 16., 2020, Neos Marmaras. **Proceedings** [...]. Cham: Springer, 2020b. p. 373-383. Disponível em: https://link.springer.com/chapter/10.1007/978-3-030-49186-4_31. Acesso em: 31 abr. 2023.
- ALHASSAN, N.A.; ALBARRAK, A.S.; BHATIA, S.; AGARWA, P. A Novel Framework for Arabic Dialect Chatbot Using Machine Learning. **Computational Intelligence and Neuroscience**, v. 2022, p.1844051-11, mar. 2022. Disponível em: <https://doi.org/10.1155/2022/1844051> Acesso em: 10 abr. 2023
- BINAMA, M.; MUHIRWA, A.; BISENGIMANA, E. Cavitation effects in centrifugal pumps-A review. **International Journal of Engineering Research and Applications**, v. 6, n. 5, p. 52-63, 2016.
- CAHN, J. **CHATBOT: Architecture, design, & development**. 2017. Tese (Senior Thesis) - School of Engineering and Applied Science Department of Computer and Information Science, University of Pennsylvania, Philadelphia, 2017.
- CARELLI, A.C.; MACIEL, M. da S. **Manutenção preditiva para tomada de decisão: um caso de bombas centrífugas industriais**. 2006. Trabalho de conclusão de curso (Bacharelado em Engenharia Química) - Escola de Química, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.
- CARVALHO, T.P.; SOARES, F.A.; VITA, R.; FRANCISCO, R.D.P.; BASTO, J.P.; ALCALÁ, S.G.A systematic literature review of machine learning methods applied to predictive maintenance. **Computers & Industrial Engineering**, v. 137, p. 106024, 2019.
- CHANG, C.Y.; HWANG, G.J.; GAU, M.L.. Promoting students' learning achievement and self-efficacy: A mobile chatbot approach for nursing training. **British Journal of Educational Technology**, v. 53, n. 1, p. 171-188, 2022.
- CHEN, L.; WEI, L.; WANG, Y.; WANG, J.; LI, W. Monitoring and predictive maintenance of centrifugal pumps based on smart sensors. **Sensors**, v. 22, n. 6, p. 2106, 2022.
- CHUDINA, M. Noise as an indicator of cavitation in a centrifugal pump. **Acoustical physics**, v. 49, p. 463-474, 2003.
- CRUZ, E.F. da. **MANUTENÇÃO DE BOMBAS HIDRÁULICAS CENTRIFUGAS: como deve ser um plano de manutenção de uma bomba centrífuga, para evitar**

a ocorrência de falhas como aquecimento, quebra e desalinhamento. 2021. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Mecânica) - Centro Universitário do Sul de Minas, Varginha, 2021.

GUNTHER COX REVISION. **About Chatterbot.** [S. l.], c2021a. Disponível em: <https://Chatterbot.readthedocs.io/en/stable>. Acesso em: 31 Mar. 2023

GUNTHER COX REVISION. **Training.** [S. l.], c2021b. Disponível em: <https://chatterbot.readthedocs.io/en/stable/training.html>. Acesso em: 31 Mar. 2023.

GUNTHER COX REVISION. **Logic Adapters.** [S. l.], c2021c. Disponível em: <https://chatterbot.readthedocs.io/en/stable/logic/index.html>. Acesso em: 31 Mar. 2023

HE, M.; HE, D. *Deep Learning* based approach for bearing fault diagnosis. **IEEE Transactions on Industry Applications**, v. 53, n. 3, p. 3057-3065, 2017.

HOANG, D.T.; KANG, H.J. A survey on Deep Learning based bearing fault diagnosis. **Neurocomputing**, v. 335, p. 327-335, 2019.

HUSSAIN, S.; AL-HASHMI, S.H.; MALIK, M.H.; KAZM, S.I.A. Chatbot in E-learning. In: SHS WEB OF CONFERENCES, 156., 2023, Web. **Article** [...]. Les Ulis: EDP Sciences, 2023. p. 1002-5

JARDIM, J.V.A.; RIFFEL, D.B.; DE MORAES COSTA, A.L. Análise de falhas de uma bomba centrífuga. In: CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA, 9., 2016, Fortaleza. **Anais** [...]. Fortaleza: CONEM, p. 0601.

JETBRAINS S.R.O. **PyCharm, o IDE da JetBrains para Python em ciência de dados e desenvolvimento para web.** PyCharm. 2024. Disponível em <https://www.jetbrains.com/pt-br/PyCharm> Acesso em: 07 mar. 2024

KSB. **Manual de serviço:** KSB ETA. N° A1150.8P/4, 2013. Disponível em: http://www.ksb.com.br/ksb-br-pt/tipos.php?codtipo=1&_pag=16. Acesso em: 02 fev 2024.

MCKEE, K.; FORBES, G.; MAZHAR, I.; ENTWISTLE, R.; HOWARD, I.. A review of major centrifugal pump failure modes with application to the water supply and sewerage industries. In: ICOMS ASSET MANAGEMENT CONFERENCE, 2011, Gold Coast. **Proceedings** [...], Gold Coast: Asset Management Council, 2011. Disponível em: <http://hdl.handle.net/20.500.11937/28560> Acesso em: 13 abr. 2023.

MENGUE, D.C.; SELLITTO, M.A. Estratégia de manutenção baseada em funções de confiabilidade para uma bomba centrífuga petrolífera. **Revista Produção Online**, [S. l.], v. 13, n. 2, p. 759–783, 2013. DOI: 10.14488/1676-1901.v13i2.1341. Disponível em: <https://producaoonline.org.br/rpo/article/view/1341>. Acesso em: 13 abr. 2023.

MINISTÉRIO DA SAUDE. **Chave Estrangeira (FK).** DATASUS. Disponível em: <https://datasus.saude.gov.br/glossario/chave-estrangeira-fk/#:~:text=Chave%20estrangeira%2C%20ou%20Foreign%20Key,prim%C3%A1ria%20de%20uma%20outra%20tabela>. Acesso em: 01 mar. 2024

NEUMANN A.T.; ARNDT T.; KÖBIS L.; MEISSNER R.; MARTIN A.; DE LANGE P.; PENGEL N.; KLAMMA R.; WOLLERSHEIM H-W. *Chatbots as a Tool to Scale Mentoring Processes: Individually Supporting Self-Study in Higher Education. Frontiers in Artificial Intelligence*, v. 4, p. 668220, 2021.

PYTHON SOFTWARE FOUNDATION. **Built-in Exceptions**. [S. l.], 2024. Disponível em <https://docs.python.org/3/library/exceptions.html>. Acesso em: 30 Mar. 2024

RODRIGUES, A.J. **Estudo de causas de falhas em selos mecânicos de bombas centrífugas para circulação de óleo térmico**. 2012. Trabalho de conclusão de curso (bacharelado em Engenharia Mecânica) – Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista Júlio de Mesquita Filho, 2012. Disponível em: <http://hdl.handle.net/11449/120812> Acesso em 15 dez. 2023.

SANTOS, A.F.C. dos; BOVÉRIO, M.A.; SILVA, D.P. da. Análise técnica de intervenção corretiva em bomba centrífuga. In: SITEFA, 6., Sertãozinho. **Anais [...]** Sertãozinho: Fatec Deputado Waldyr Alceu Trigo, 2023, p. e614. Disponível em: <https://sitefa.fatecsertaozinho.edu.br/sitefa/article/view/242>. Acesso em: 13 fev. 2024.

SANTOS FILHO, C.A. **Diagnóstico de falha em bomba centrífuga na indústria petroquímica**. 2018. Trabalho de conclusão de curso (Especialização em Gestão da Manutenção) - Centro Universitário SENAI CIMATEC, Salvador, 2018.

SERBAN, C.; SANKAR *et al.* A deep reinforcement learning *chatbot*. **ArXiv**, arXiv:1709.02349v2, 40 f, 2017.

SILVA, C.G. dos S.S. **Desenvolvimento de ferramenta gerencial para cálculo do índice de confiabilidade em manutenção de bombas centrífugas considerando os fatores humanos**. 2016. Dissertação (Mestrado em Ciências em Engenharia Industrial) - Escola Politécnica, Universidade Federal da Bahia, Salvador, 2017.

SILVA, C.N. *et al.* ESTUDO DE MANUTENÇÃO DE BOMBA CENTRÍFUGA COM USO EM INDÚSTRIA. **South American Development Society Journal**, [S.l.], v. 1, n. 1, p. 71 - 87, mar. 2017. ISSN 2446-5763. Disponível em: <http://www.sadsj.org/index.php/revista/article/view/6> . Acesso em: 13 abr. 2023.

SILVA, D.E. **lubrificação industrial análise de óleo lubrificante**. 2021. Trabalho de conclusão de curso (Bacharel em Engenharia Mecânica) - Faculdade Evangélica de Goianésia, Universidade Federal do Amapá, Goianésia, 2021.

SILVA, T.A.F.; NASCIMENTO, G.M. Importância da manutenção preditiva por análise de vibração num conjunto motor e bomba aplicado (um estudo de caso). In: SIMPÓSIO NACIONAL DE CIÊNCIAS E ENGENHARIAS (SINACEN), 5., 2020, Anápolis. **Anais [...]** Anápolis: UniEVANGÉLICA, 2022, p. 112-131.

SUSTO, G.A.; SCHIRRU, A.; PAMPURI, S.; MCLOONE, S.; BEGHI, A. Machine learning for predictive maintenance: A multiple classifier approach. **IEEE transactions on industrial informatics**, v. 11, n. 3, p. 812-820, 2014.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL: The World's Most Advanced Open Source Relational Database**. PostgreSQL. 2024. Disponível em <<https://www.postgresql.org/>> Acesso em: 05 mar. 2024.

THIN, K.C.; KHAING, M.M.; AYE, K.M. Design and performance analysis of centrifugal pump. **World academy of science, engineering and technology**, v. 46, n. 1, p. 422-429, 2008.

THEISSLER, A.; PÉREZ-VELÁZQUEZ, J.; KETTELGERDES, M.; ELGER, G. Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. **Reliability engineering & system safety**, v. 215, p. 107864, 2021.

VIJAYARAGHAVAN, V.; COOPER, J.B.; LEEVINSON J.R. Algorithm inspection for chatbot performance evaluation. **Procedia Computer Science**, v. 171, p. 2267-2274, 2020.

ZHU, Z.; LEI, Y.; QI, G.; CHAI, Y.; MAZUR, N.; AN, Y.; HUANG, X. A review of the application of Deep Learning in intelligent fault diagnosis of rotating machinery. **Measurement**, v. 206, p. 112346, 2023.

APÊNDICE A – CÓDIGO FONTE DA INICIALIZAÇÃO DO CHATBOT

```

import Chatterbot
import numpy
from Utilidades.SeparacaoPalavras import separacaoPalavras, chanceConter
import psycpg2
def inicializar():
    print('chatbot: Olá, sou seu assistente virtual E-MANUEL e vou te ajudar com sua manutenção')
    print('chatbot: Com qual bomba trabalharemos hoje?')
    resposta: str = input('você: ')
    con = psycpg2.connect(
        user="postgres",
        password="23071998",
        database="Dados_bomba" )

    cur = con.cursor()
    cur.execute('SELECT codigo FROM dados_tecnicos')
    listaBombas = numpy.transpose(cur.fetchall())[0]
    resp = separacaoPalavras(resposta)
    resultadoBusca = chanceConter(resp, listaBombas)
    confirmacao = False
    while (resultadoBusca[0] < 1) and not confirmacao:
        if (resultadoBusca[0] < 0.8):
            print('chatbot: Não encontrei essa bomba no banco de dados, por favor tente novamente')
            resposta = input('você: ')
            cur.execute('SELECT codigo FROM dados_tecnicos')
            listaBombas = numpy.transpose(cur.fetchall())[0]
            resp = separacaoPalavras(resposta)
            resultadoBusca = chanceConter(resp, listaBombas)
        else:
            print(f'chatbot: Eu encontrei {resultadoBusca[1]}, é esse que você quer?')
            resposta = input('você: ')
            resp = separacaoPalavras(resposta)
            negativa = ["nao"]
            afirmativa = ["sim", "essa", "isso", 'certo', 'correto', 'esse']
            confirmacao = (chanceConter(resp, afirmativa)[0] > 0.8) and (chanceConter(resp, negativa)[0] < 0.8)
            if not confirmacao :
                cur.execute('SELECT codigo FROM dados_tecnicos')
                listaBombas = numpy.transpose(cur.fetchall())[0]
                resultadoBusca = chanceConter(resp, listaBombas)

    # print(f'desenvolvimento: aprovado')
    con.close()
    print(f'chatbot: Então hoje trataremos da bomba {resultadoBusca[1]}, aguarde um pouco enquanto preparo o ambiente...')
    global codigo
    codigo= resultadoBusca[1]
    return resultadoBusca[1]

chatbot = Chatterbot.ChatBot(
    "E-MANUEL",
    storage_adapter='Chatterbot.storage.SQLStorageAdapter',
    preprocessors=[
        'Processador.pross' ],
    logic_adapters=[
        { 'import_path': 'Adapters.BuscaDadosTec.BuscaDadosTec',
          'bomba': inicializar() },
        { 'import_path': 'Adapters.BuscaDadosLub.BuscaDadosLub',
          'bomba': codigo },
        { 'import_path': 'Adapters.Checklist.Checklist' },
        { 'import_path': 'Chatterbot.logic.BestMatch',
          'default_response': 'Sinto muito, não compreendi do que você precisa', } ],
    database_uri='sqlite:///database.sqlite3'

```

APÊNDICE B – CÓDIGO FONTE DA INTERAÇÃO COM O USUÁRIO

```
from Chatterbot.conversation import Statement

print("Iniciando assistente ...")
from assistente import chatbot
print('chatbot: Ambiente pronto, como posso ajudá-lo?')
entrada = ''
while entrada.lower() not in ('fim', 'acabou', 'terminamos', 'finalizado'):
    try:
        entrada = Statement(text=input('você: '))
        response = chatbot.get_response(entrada)
        print('chatbot:', response)
    except (KeyboardInterrupt, EOFError, SystemExit):
        break
```

Local: main.py

APÊNDICE C – CÓDIGO FONTE DO PRÉPROCESSADOR

```
def pross(statement_input):
    texto = statement_input.text.strip().lower()
    statement_output = statement_input
    statement_output.text = texto
    statement_output.conversation = 'Processado'
    return statement_output
```

Local: processador.py

APÊNDICE D – CÓDIGO FONTE DE FUNÇÕES UTILITÁRIAS

```
from difflib import SequenceMatcher

def separacaoPalavras(frase):
    for x in ["?", "!", ".", ","]:
        if x in frase:
            frase = frase.replace(x, " ")
    palavras = frase.lower().split()
    return palavras

def chanceConter(fraseBusca, comparativo):
    chance = -1
    palavraParecida: str = []
    for palavra1 in fraseBusca:
        for palavra2 in comparativo:
            match = SequenceMatcher(None, palavra1, palavra2).ratio()
            if match > chance:
                chance = match
                palavraParecida = palavra2
            if (chance == 1): break
        if (chance == 1): break
    return [chance, palavraParecida]
```

Local: SeparacaoPalavras.py localizado na pasta 'Utilidades'

APÊNDICE E – CÓDIGO FONTE DO ADAPTADOR LÓGICO CHECKLIST

```

from Chatterbot.conversation import Statement
from Chatterbot.logic import LogicAdapter
from Utilidades.SeparacaoPalavras import separacaoPalavras, chanceConter
from Utilidades.ValoresUteis import ajust

class Checklist(LogicAdapter):

    inicio: bool
    conferencia: bool #indica de o processo foi iniciado ou não
    checklist = {'lubrificante':False, 'parafusos':False, 'nivelamento':False, 'acoplamento':False,
'movimentaca':False, 'gaxeta':False}
    currentKey: str #Passo atual da checklist

    def __init__(self, chatbot, **kwargs):
        super().__init__(chatbot, **kwargs)
        self.conferencia = False
        for v in self.checklist.values(): v = False
        self.currentKey = 'lubrificante'

    def can_process(self, statement):
        frase = statement.text
        palavras = separacaoPalavras(frase)
        if chanceConter(palavras, ['checklist'])[0]>0.8 and not self.conferencia: #Detecta a palavra
checklist
            self.conferencia = True
            self.inicio = True
            for k in self.checklist.keys(): self.checklist[k] = False #inicia o processo de checklist
            return self.conferencia

    def process(self, statement, additional_response_selection_parameters=None):

        completo: bool = True
        afirmativos =
['ok','certo','verificado','conferido','feito','sim','estão','está','apertados','firmes','nivelado']
        palavras =
separacaoPalavras(statement.text.replace('oleo','lubrificante',1).replace('óleo','lubrificante'))
        verificado = chanceConter(palavras, afirmativos)
        mencao = chanceConter(palavras, self.checklist.keys())
        if mencao[0] > 0.8:
            self.currentKey = mencao[1]
            if chanceConter(palavras,'nivelado')[0] > 0.8 and not (mencao[0] > 0.8 and mencao[1] ==
'lubrificante'):
                self.currentKey = 'nivelamento'
            if verificado[0] > 0.8:
                self.checklist[self.currentKey] = True
            resp = Statement(text='Ok')
            for k in self.checklist.keys() :
                completo = completo and self.checklist[k]
                if self.checklist[k] == False :
                    self.currentKey = k
                    break
            if completo:
                self.conferencia = False
                resp.text = 'Tudo conferido'
                resp.confidence = max(mencao[0], verificado[0]) * 0.9
                return resp

        match self.currentKey:
            case 'lubrificante':
                resp.text = 'O nível do óleo está adequado?'
            case 'parafusos':
                resp.text = 'Por favor verifique se os parafusos estão bem apertados'
            case 'nivelamento':
                resp.text = 'O equipamento está devidamente nivelado?'
            case 'acoplamento':
                resp.text = 'O motor está devidamente acoplado a bomba?'
            case 'movimentaca':
                resp.text = 'O eixo consegue movimentar livremente?'
            case 'gaxeta':
                resp.text = 'Tudo certo com a gaxeta?'

        if self.inicio:
            self.inicio = False
            resp.confidence = 1.0
        else:
            resp.confidence = max(mencao[0],verificado[0])*ajust
        return resp

```

Local: Checklist.py localizado na pasta 'Adapters'

APÊNDICE F – CÓDIGO FONTE DE BUSCADADOSLUB

```

from Utilidades.ValoresUteis import ajust

class BuscaDadosLub(LogicAdapter):
    bomba : str

    def __init__(self, chatbot, **kwargs):
        super().__init__(chatbot, **kwargs)
        self.bomba = kwargs.get('bomba')

    def can_process(self, statement):
        return True

    def process(self, input_statement, additional_response_selection_parameters):
        # inicia conexão com o banco de dados postgres (pode ser alterado para outros
        # bancos de dados)
        con = psycopg2.connect(
            user = "postgres",
            password = "23071998",
            database = "Dados_bomba"
        )
        cur = con.cursor()
        inpute = input_statement.text
        frase = separacaoPalavras(inpute)

        # Cálculo da probabilidade de as palavras nota e nome terem sido mencionadas
        perguntas = ["qual", "quanto"]
        contextoOleo = ["oleo", "lubrificação", "lubrificante"]
        informacoes =
["marca", "periodo", "frequencia", "tipo", "fabricante", "empresa", "quanto", "quantidade", "vol
ume"]

        chancePergunta = chanceConter(frase, perguntas) [0]
        chanceOleo = chanceConter(frase, contextoOleo) [0]
        probabilidade = chanceConter(frase, informacoes)

        retorno = ""
        match probabilidade[1]:
            case "marca" | "tipo" :
                cur.execute(f"SELECT tp_oleo FROM lubrificacao WHERE cd_bomba =
'{self.bomba}'")
                resposta = cur.fetchall() [0] [0]
                retorno = f"Essa bomba usa óleo {resposta}"
            case "periodo" | "frequencia" :
                cur.execute(f"SELECT frequencia_troca FROM lubrificacao WHERE cd_bomba =
'{self.bomba}'")
                resposta = cur.fetchall() [0] [0]
                retorno = f"O óleo precisa ser trocado a cada {resposta} dias"
            case "fabricante" | "empresa" :
                cur.execute(f"SELECT marca FROM lubrificacao WHERE cd_bomba =
'{self.bomba}'")
                resposta = cur.fetchall() [0] [0]
                retorno = f"Essa marca é produzida pela {resposta}"
            case "quanto" | "quantidade" | "volume":
                cur.execute(f"SELECT quantidade FROM lubrificacao WHERE cd_bomba =
'{self.bomba}'")
                resposta = cur.fetchall() [0] [0]
                retorno = f"A bomba precisa de {resposta}L de óleo"

        confianca = (0.4*chancePergunta+0.4*chanceInt+0.2*chanceOleo)*ajust

        output_statement = Statement(text=retorno)
        output_statement.confidence = confianca
        con.close()
        return output_statement

```

Local: BuscaDadosLub.py localizado na pasta 'Adapters'

APÊNDICE G – CÓDIGO FONTE DE BUSCADADOSTEC

```
class BuscaDadosTec(LogicAdapter):
    codigo = ''

    def __init__(self, chatbot, **kwargs):
        super().__init__(chatbot, **kwargs)
        self.codigo = kwargs.get('bomba')

    def can_process(self, statement):
        return True

    def process(self, input_statement, additional_response_selection_parameters):

        # inicia conexão com o banco de dados postgres
        con = psycopg2.connect(
            user = "postgres",
            password = "23071998",
            database = "Dados_bomba"
        )
        cur = con.cursor()
        inpute = input_statement.text
        frase = separacaoPalavras(inpute)

        # Cálculo da probabilidade das palavras terem sido mencionadas
        perguntas = ["qual", "quanto"]
        informacoes =
["rotor", "produto", "modelo", "empresa", "velocidade", "rotação", "codigo", "numeração", "substância", "fabr
icante"]
        chancePergunta = chanceConter(frase, perguntas)[0]
        probabilidade = chanceConter(frase, informacoes)

        contextoOleo = ["oleo", "lubrificação", "lubrificante"]
        chanceOleo = chanceConter(frase, contextoOleo)[0]

        retorno = ""
        match probabilidade[1]:
            case "rotor" :
                cur.execute(f"SELECT tm_rotor FROM dados_tecnicos WHERE codigo = '{self.codigo}'")
                resposta = cur.fetchall()[0][0]
                retorno = f"O rotor mede {resposta}cm"
            case "produto" | "substância":
                cur.execute(f"SELECT produto FROM dados_tecnicos WHERE codigo = '{self.codigo}'")
                resposta = cur.fetchall()[0][0]
                retorno = f"Esta bomba está operando com {resposta}"
            case "modelo" :
                cur.execute(f"SELECT modelo FROM dados_tecnicos WHERE codigo = '{self.codigo}'")
                resposta = cur.fetchall()[0][0]
                retorno = f"Essa máquina é uma bomba {resposta}"
            case "empresa" | "fabricante":
                cur.execute(f"SELECT empresa FROM dados_tecnicos WHERE codigo = '{self.codigo}'")
                resposta = cur.fetchall()[0][0]
                retorno = f"A bomba foi feita pela {resposta}"
            case "velocidade" | "rotação":
                cur.execute(f"SELECT velocidade FROM dados_tecnicos WHERE codigo = '{self.codigo}'")
                resposta = cur.fetchall()[0][0]
                retorno = f"Essa máquina opera á {resposta}rpm"
            case "codigo" | "numeração":
                cur.execute(f"SELECT codigo FROM dados_tecnicos WHERE codigo = '{self.codigo}'")
                resposta = cur.fetchall()[0][0]
                retorno = f"Essa bomba está registrada como {resposta}"

        confianca = (0.4*chancePergunta+0.6*chanceInt-0.2*chanceOleo)*ajust

        output_statement = Statement(text=retorno)
        output_statement.confidence = confianca

        con.close()      # Fecha conexão com banco de dados
        return output_statement
```

Local: BuscaDadosTec.py localizado na pasta 'Adapters'

APÊNDICE H – CÓDIGO FONTE DO TREINO DE CHATBOT

```

from Chatterbot.trainers import ListTrainer
from Chatterbot.trainers import ChatterbotCorpusTrainer
from assistente import chatbot

chatbot.storage.drop()                                # Limpar dados de treinamento

trainer = ListTrainer(chatbot)                        # Inicializar treinador

# Início de conversa
cumprimento1 = [
    "oi",
    "olá, como posso ajudar?",
    "não estou precisando de ajuda",
    "tudo bem, se precisar de algo estarei aqui para ajudar"
]
cumprimento2 = [
    "preciso de ajuda",
    "Pode falar, estou aqui para ajudar",
    "você pode me ajudar?",
    "claro, do que precisa?",
    "estou com um problema",
    "pode me contar que te ajudo no que for preciso"
]
cumprimento3 = [
    "bom dia",
    "bom dia, como posso te ajudar hoje?"
]
cumprimento4 = [
    "boa tarde",
    "boa tarde, em que posso ser útil?"
]
cumprimento5 = [
    "boa noite",
    "boa noite, como posso ajudar?"
]
cumprimento6 = [
    'bom dia! como você está?',
    'estou muito bem obrigado. e você?',
    'estou bem também',
    'que bom, tem algo em que eu possa ajudar?'
]
cumprimento1_2 = [
    "olá",
    "olá, como posso ajudar?",
    "não preciso de ajuda",
    "tudo bem, se precisar de algo estarei aqui para ajudar"
]
cumprimento1_3 = [
    'oi!',
    'oi!', que bom ter você aqui! como posso te ajudar?',
    'oi!',
    'nossa, alguém está animado(a)! como posso te ajudar?',
    'oie',
    'olá! como posso ajudar?'
]

# fim de conversa
Agradecimento1 = [
    "obrigado",
    "disponha, estou aqui para ajudar",
    "muito obrigado",
    "imagina, fico feliz em ajudar"
]
Agradecimento2 = [
    "consegui",
    "que bom, espero ter ajudado",
    "deu certo",
    "legal! precisa de algo mais?",
    "não, obrigado",
    "ok, se precisar de mais alguma coisa estarei a disposição"
]
Agradecimento3 = [
    "não deu certo",
    "que pena, sinto muito não ter resolvido seu problema",
    "tchau",
    "até mais, estarei aqui se precisar de mais alguma coisa"
]
Agradecimento4 = [
    'terminei',
    'que bom, algo mais em que eu possa ajudar?',
    'já terminei aqui',
    'Ok, algo mais em que eu possa ajudar?'
]
Agradecimento5 = [
    'muito bem!',

```

```

    'obrigado! faço o meu melhor'
]

# meio de conversa
meio1 = [
    "não foi isso que eu perguntei",
    "Desculpe, tente me perguntar de outra forma"
]
meio2 = [
    'qual o próximo passo?',
    'qual foi a última coisa que fizemos?',
    'não sei te dizer,não tenho memória muito boa desculpa :('
]
meio3 = [
    'o que é a seguir?',
    'depende, onde estamos mesmo?'
]
tipoBomba = [
    'que tipo de bomba é essa?',
    'essa é uma bomba centrífuga'
]
silencio = [
    '',
    '...',
    '.....',
    'então... você quer dizer alguma coisa?',
    'não'
]

# guiar na instalação
instalacao1 = [
    "preciso instalar a bomba",
    "ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
    "fundação pronta",
    "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa",
    "base pronta",
    "espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
    "parafusos apertados",
    "se as tubulações encostarem sem esforço nas flanges, fixe-as",
    "tubulações fixadas",
    "agora fixe cuidadosamente o acoplamento ao rotor da bomba",
    "acoplamento fixado",
    "ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
    "motor pronto",
    "tudo certo então, algo mais em que eu possa ajudar?"
]
instalacao2 = [
    "preciso fazer a instalação da bomba",
    "ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
    "fundação já está limpa",
    "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa",
    "base nivelada",
    "espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
    "bomba parafusada",
    "se as tubulações encostarem sem esforço nas flanges, fixe-as",
    "tubulações fixas",
    "agora fixe cuidadosamente o acoplamento ao rotor da bomba",
    "acoplamento pronto",
    "ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
    "motor fixado",
    "tudo certo então, algo mais em que eu possa ajudar?"
]
instalacao3 = [
    "vamos instalar a bomba",
    "ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
    "tudo limpo",
    "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa",
    "parafusos apertados corretamente",
    "espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
    "parafusos apertados",
    "se as tubulações encostarem sem esforço nas flanges, fixe-as",
    "tubulação fixada na bomba",
    "agora fixe cuidadosamente o acoplamento ao rotor da bomba",
    "acoplado",
    "ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
    "motor acoplado",
    "tudo certo então, algo mais em que eu possa ajudar?"
]
instalacao4 = [
    "vamos começar a instalação",
    "ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
    "podemos fixar a base",
    "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa",
    "pronto para parafusar",

```

```

"espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
"podemos ir para as tubulações",
"se as tubulações encostarem sem esforço nas flanges, fixe-as",
"agora para o acoplamento",
"agora fixe cuidadosamente o acoplamento ao rotor da bomba",
"podemos instalar o motor",
"ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
"tudo pronto",
"tudo certo então, algo mais em que eu possa ajudar?"
]
instalacao5 = [
"com a instalação do equipamento",
"ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
"fundação seca",
"certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com
argamassa",
"base fixa",
"espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
"bomba fixa",
"se as tubulações encostarem sem esforço nas flanges, fixe-as",
"Bomba presa as tubulações",
"agora fixe cuidadosamente o acoplamento ao rotor da bomba",
"acoplamento pronto",
"ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
"bomba acoplada ao motor",
"tudo certo então, algo mais em que eu possa ajudar?"
]
instalacao6 = [
"instalar bomba",
"ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
"a fundação está pronta",
"certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com
argamassa",
"fixei a base",
"espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
"apertei os parafusos",
"se as tubulações encostarem sem esforço nas flanges, fixe-as",
"fixei a tubulação",
"agora fixe cuidadosamente o acoplamento ao rotor da bomba",
"fixei o acoplamento",
"ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
"fixei o motor",
"tudo certo então, algo mais em que eu possa ajudar?"
]
instalacao7 = [
"instalar bomba",
"ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento",
"fundação já está seca e limpa",
"certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com
argamassa",
"base fixada",
"espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente",
"parafusos apertados corretamente",
"se as tubulações encostarem sem esforço nas flanges, fixe-as",
"fixei a tubulação",
"agora fixe cuidadosamente o acoplamento ao rotor da bomba",
"fixei o acoplamento",
"ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
"fixei o motor",
"tudo certo então, algo mais em que eu possa ajudar?"
]

# guia de alinhamento
alinhamentoInspecao = [
'como eu verifico o alinhamento?',
'',
'como sei que está alinhado?',
'o grupo está corretamente alinhado se uma régua, posta sobre as 2 metades do acoplamento e em plano
paralelo ao eixo, tiver em todos os pontos a mesma distância do eixo',
'como eu verifico se está alinhado',
'',
'como saber se o eixo está alinhado?',
'o grupo está corretamente alinhado se uma régua, posta sobre as 2 metades do acoplamento e em plano
paralelo ao eixo, tiver em todos os pontos a mesma distância do eixo',
'como sei se o eixo está alinhado?',
'o grupo está corretamente alinhado se uma régua, posta sobre as 2 metades do acoplamento e em plano
paralelo ao eixo, tiver em todos os pontos a mesma distância do eixo'
]
alinhamentoAjuste = [
'o eixo está desalinhado',
'então porfavor faça o ajuste para evitarmos problemas de desgaste',
'não está alinhado',
'ok, etão realize o realinhamento e depois me avise se está alinhado'
]

# desmontagem da bomba
desmontagem1 = [
'preciso desmontar a bomba, como faço?',

```

```

'primeiro preciso saber, a bomba já está fora da base?',
'a bomba ainda esta na base',
'então primeiro você deve tirá-la da base',
'a bomba já está fora da base',
'ótimo, agora remova a tampa de aspiração',
'tampa de aspiração removida',
'agora solte a porca do rotor',
'porca do rotor retirada',
'muito bem, agora retire a arruela de segurança',
'arruela de segurança retirada',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'tampa do cavalete dos mancais retirada',
'então finalmente retire o rotor, mas use pancadas leves',
'não estou conseguindo retirar o rotor',
'então tente usar líquidos solventes de ferrugem',
'rotor retirado',
'muinto bem, então bomba desmontada'
]
desmontagem2 = [
'como faço para desmontar a bomba?',
'primeiro preciso saber, a bomba já está fora da base?',
'ainda não tirei a bomba da base',
'então primeiro você deve tirá-la da base',
'já tirei a bomba da base',
'ótimo, agora remova a tampa de aspiração',
'removi a bomba de aspiração',
'agora solte a porca do rotor',
'soltei a porca do rotor',
'muito bem, agora retire a arruela de segurança',
'tirei a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'soltei a tampa do cavalete dos mancais',
'então finalmente retire o rotor, mas use pancadas leves',
'o rotor não está saindo',
'então tente usar líquidos solventes de ferrugem',
'consegui tirar o rotor',
'muinto bem, então bomba desmontada'
]
desmontagem3 = [
'como desmonto a bomba?',
'primeiro preciso saber, a bomba já está fora da base?',
'a bomba não está fora da base',
'então primeiro você deve tirá-la da base',
'a bomba foi retirada da base',
'ótimo, agora remova a tampa de aspiração',
'a tampa de aspiração já foi retirada',
'agora solte a porca do rotor',
'porca do rotor solta',
'muito bem, agora retire a arruela de segurança',
'a arruela de segurança já foi retirada',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'a tampa do cavalete dos mancais já está solta',
'então finalmente retire o rotor, mas use pancadas leves',
'não consegui retirar o rotor',
'então tente usar líquidos solventes de ferrugem',
'agora consegui retirar o rotor',
'muinto bem, então bomba desmontada'
]
desmontagem4 = [
'vamos desmontar a bomba',
'primeiro preciso saber, a bomba já está fora da base?',
'ainda não tiramos a bomba da base',
'então primeiro você deve tirá-la da base',
'a bomba está fora da base',
'ótimo, agora remova a tampa de aspiração',
'a bomba já está sem a tampa de aspiração',
'agora solte a porca do rotor',
'o rotor está sem a porca',
'muito bem, agora retire a arruela de segurança',
'já tirei a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'tampa do cavalete do lado do acionamento solta',
'então finalmente retire o rotor, mas use pancadas leves',
'o rotor não quer sair',
'então tente usar líquidos solventes de ferrugem',
'deu certo, tirei o rotor',
'muinto bem, então bomba desmontada'
]
desmontagem5 = [
'preciso desmontar a bomba',
'primeiro preciso saber, a bomba já está fora da base?',
'não, ainda está na base',
'então primeiro você deve tirá-la da base',
'a bomba já está fora da base',
'ótimo, agora remova a tampa de aspiração',
'tampa de aspiração retirada',
'agora solte a porca do rotor',
'já está sem a porca do rotor',

```

```

'muito bem, agora retire a arruela de segurança',
'tirei a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'tampa do cavalete dos mancais do lado do acionamento retirada',
'então finalmente retire o rotor, mas use pancadas leves',
'Não dá, o rotor não sai',
'então tente usar líquidos solventes de ferrugem',
'já soltei o rotor',
'muito bem, então bomba desmontada'
]
desmontagem6 = [
'quero desmontar a bomba',
'primeiro preciso saber, a bomba já está fora da base?',
'não, ainda está na base',
'então primeiro você deve tirá-la da base',
'sim, já está fora da base',
'ótimo, agora remova a tampa de aspiração',
'tirei a tampa de aspiração',
'agora solte a porca do rotor',
'',
'muito bem, agora retire a arruela de segurança',
'já tirei a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'soltei a tampa do cavalete dos mancais do lado do acionamento',
'então finalmente retire o rotor, mas use pancadas leves',
'',
'então tente usar líquidos solventes de ferrugem',
'o rotor já está fora da bomba',
'muito bem, então bomba desmontada'
]
desmontagem7 = [
'poderia me ajudar a desmontar a bomba?',
'claro! primeiro preciso saber, a bomba já está fora da base?',
'',
'então primeiro você deve tirá-la da base',
'',
'ótimo, agora remova a tampa de aspiração',
'',
'agora solte a porca do rotor',
'',
'muito bem, agora retire a arruela de segurança',
'acabei de retirar a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'cavalete dos mancais destampado',
'então finalmente retire o rotor, mas use pancadas leves',
'',
'então tente usar líquidos solventes de ferrugem',
'',
'muito bem, então bomba desmontada'
]
demontcavall = [
'preciso desmontar o cavalete do mancal',
'para desmontar o cavalete primeiro solte o aperta-gaxeta e separe o corpo da bomba do cavalete',
'separei do corpo da bomba',
'tire a vareta indicadora e retire o cavalete cuidadosamente do eixo',
'tirei o cavalete do eixo',
'por fim solte a tampa do mancal do lado da gaxeta',
'soltei a tampa do mancal do lado da gaxeta',
'terminamos de desmontar o cavalete do mancal, mais alguma coisa em que eu possa ajudar?'
]
demontcaval2 = [
'me ajuda a desmontar o cavalete?',
'claro! para desmontar o cavalete primeiro solte o aperta-gaxeta e separe o corpo da bomba do cavalete',
'já separei o cavalete da bomba',
'tire a vareta indicadora e retire o cavalete cuidadosamente do eixo',
'o cavalete já está fora do eixo',
'por fim solte a tampa do mancal do lado da gaxeta',
'soltei a tampa do mancal',
'terminamos de desmontar o cavalete do mancal, mais alguma coisa em que eu possa ajudar?'
]
demontcaval3 = [
'',
'para desmontar o cavalete primeiro solte o aperta-gaxeta e separe o corpo da bomba do cavalete',
'',
'tire a vareta indicadora e retire o cavalete cuidadosamente do eixo',
'',
'por fim solte a tampa do mancal do lado da gaxeta',
'soltei essa tampa do mancal',
'terminamos de desmontar o cavalete do mancal, mais alguma coisa em que eu possa ajudar?'
]
# troca de rolamento
trocaRolamento1 = [
'preciso trocar o rolamento',
'para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves',
'tirei o rolamento',

```

```

'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
'o novo rolamento já está no óleo',
'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
'novo rolamento fixado',
'agora apenas se certifique que o rolamento novo está rodando sem problemas'
]
trocaRolamento2 = [
'vamos trocar o rolamento',
'para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com
pancadas leves',
'tirei o rolamento antigo',
'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
'o rolamento novo está pronto',
'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
'o rolamento novo já está encaixado',
'agora apenas se certifique que o rolamento novo está rodando sem problemas'
]
trocaRolamento3 = [
'vou trocar o rolamento, você pode me ajudar?',
'claro, para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com
pancadas leves',
'o rolamento antigo já está fora do eixo',
'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
'rolamento novo preparado',
'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
'assentei o rolamento novo',
'agora apenas se certifique que o rolamento novo está rodando sem problemas'
]
trocaRolamento4 = [
'',
'para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com
pancadas leves',
'',
'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
'rolamento novo banhado',
'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
'já coloquei o rolamento novo no lugar',
'agora apenas se certifique que o rolamento novo está rodando sem problemas'
]
trocaRolamento5 = [
'preciso trocar o rolamento',
'para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com
pancadas leves',
'tirei o rolamento antigo',
'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
'já preparei o rolamento novo',
'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
'o novo rolamento já está no lugar',
'agora apenas se certifique que o rolamento novo está rodando sem problemas'
]
trocaRolamento6 = [
'',
'para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com
pancadas leves',
'',
'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
'',
'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
'',
'agora apenas se certifique que o rolamento novo está rodando sem problemas'
]

# montar bomba
montar1 = [
'agora vamos remontar a bomba',
'então primeiro insira o rotor de volta',
'rotor já está no lugar',
'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
'tampa do cavalete presa',
'coloque novamente a arruela de segurança',
'arruela de segurança colocada',
'em seguida prenda novamente a porca do rotor',
'porca do rotor presa',
'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo'
]
montar2 = [
'vou montar a bomba',
'então primeiro insira o rotor dentro da bomba',
'rotor inserido',
'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
'prendi a tampa do cavalete dos mancais',
'coloque novamente a arruela de segurança',
'coloquei a arruela de segurança de volta',
'em seguida prenda novamente a porca do rotor',
'rosqueei a porca do rotor de volta',
'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo'
]
montar3 = [

```

```

'preciso remontar a bomba',
'então primeiro insira o rotor de volta',
'inseri o motor',
'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
'prendi a tampa do cavalete dos mancais de volta',
'coloque novamente a arruela de segurança',
'já coloquei de volta a arruela de segurança',
'em seguida prenda novamente a porca do rotor',
'prendi a porca do rotor',
'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo'
]
montar4 = [
'vamos remontar a bomba',
'então primeiro insira o rotor de volta',
'inseri o rotor de volta no lugar',
'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
'a tampa do cavalete dos mancais já está no lugar',
'coloque novamente a arruela de segurança',
'a arruela de segurança está no lugar',
'em seguida prenda novamente a porca do rotor',
'porca do rotor de volta no lugar',
'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo'
]
montar5 = [
'vamos montar a bomba novamente',
'então primeiro insira o rotor de volta',
'o rotor já foi colocado de volta',
'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
'tampa do cavalete dos mancais presa',
'coloque novamente a arruela de segurança',
'a arruela de segurança está colocada',
'em seguida prenda novamente a porca do rotor',
'a porca do rotor está presa',
'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo'
]
montar6 = [
'',
'então primeiro insira o rotor de volta',
'o rotor já foi inserido no lugar',
'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
'prendi a tampa do cavalete dos mancais no lado do acionamento',
'coloque novamente a arruela de segurança',
'',
'em seguida prenda novamente a porca do rotor',
'porca do rotor foi fixada',
'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo'
]
montar7 = [
'',
'então primeiro insira o rotor de volta',
'o rotor já está no lugar',
'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
'a tampa do cavalete dos mancais já está fixada no lado do acionamento',
'coloque novamente a arruela de segurança',
'',
'em seguida prenda novamente a porca do rotor',
'porca do rotor está fixada',
'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo'
]
# modos de falhas
rolamento1 = [
'o rolamento quebrou, o que pode ter acontecido?',
'pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o alinhamento do eixo de
rotação e troque o rolamento',
'o conjunto está desalinhado',
'então por favor realinhe o conjunto',
'o problema não é o óleo e nem o alinhamento',
'então pode ser sobrecarga na bomba'
]
rolamento2 = [
'o que pode ter causado falha no rolamento?',
'pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o alinhamento do eixo de
rotação e troque o rolamento',
'percebi um desalinhamento',
'então por favor realinhe o conjunto',
'não tem nada de errado com a lubrificação ou com o rolamento',
'então pode ser sobrecarga na bomba'
]
rolamento3 = [
'o que pode ter quebrado o rolamento?',
'pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o alinhamento do eixo de
rotação e troque o rolamento',
'o conjunto não está alinhado',
'então por favor realinhe o conjunto',
'não encontrei problema no alinhamento e nem no óleo',
'então pode ser sobrecarga na bomba'
]

```

```

rolamento4 = [
    'o que pode ter ocasionado a falha no rolamento?',
    'pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o alinhamento do eixo de
    rotação e troque o rolamento',
    'Está desalinhado',
    'então por favor realinhe o conjunto',
    'não encontrei problema no alinhamento e nem no óleo',
    'então pode ser sobrecarga na bomba'
]
lubrificacao1 = [
    'o óleo não está bom',
    'por favor analise o óleo e verifique o que tem de errado com ele',
    'o óleo está gasto',
    'então vamos precisar trocar mas se ele está ficando gasto muito cedo você vai precisar fazer uma
    analise mais minuciosa com a equipe técnica',
    'o óleo está com impurezas',
    'então será necessário trocá-lo e realizar uma analise para averiguar a causa desas impurezas',
    'o óleo está muito quente',
    'verifique o nível de óleo na vareta indicadora, o alinhamento do conjunto'
]
lubrificacao2 = [
    'o óleo não está bom',
    'por favor analise o óleo e verifique o que tem de errado com ele',
    'o óleo está gasto',
    'então vamos precisar trocar mas se ele está ficando gasto muito cedo você vai precisar fazer uma
    analise mais minuciosa com a equipe técnica',
    'o óleo está com impurezas',
    'então será necessário trocá-lo e realizar uma analise para averiguar a causa desas impurezas',
    'o óleo está muito quente',
    'verifique o nível de óleo na vareta indicadora, o alinhamento do conjunto'
]
lubrificacao3 = [
    'tem algum problema com a lubrificação',
    'por favor analise o óleo e verifique o que tem de errado com ele',
    'o óleo está velho',
    'então vamos precisar trocar mas se ele está ficando gasto muito cedo você vai precisar fazer uma
    analise mais minuciosa com a equipe técnica',
    'o óleo está sujo',
    'então será necessário trocá-lo e realizar uma analise para averiguar a causa desas impurezas',
    'o óleo está muito quente',
    'verifique o nível de óleo na vareta indicadora, o alinhamento do conjunto'
]
lubrificacao4 = [
    'o óleo parece estar ruim',
    'por favor analise o óleo e verifique o que tem de errado com ele',
    'o óleo está usado',
    'então vamos precisar trocar mas se ele está ficando gasto muito cedo você vai precisar fazer uma
    analise mais minuciosa com a equipe técnica',
    '',
    'então será necessário trocá-lo e realizar uma analise para averiguar a causa desas impurezas',
    '',
    'verifique o nível de óleo na vareta indicadora, o alinhamento do conjunto'
]

# Treinamentos
trainer.train(cumprimento1)
trainer.train(cumprimento2)
trainer.train(cumprimento3)
trainer.train(cumprimento4)
trainer.train(cumprimento5)
trainer.train(cumprimento6)
trainer.train(cumprimento1_2)
trainer.train(cumprimento1_3)

trainer.train(Agradecimento1)
trainer.train(Agradecimento2)
trainer.train(Agradecimento3)
trainer.train(Agradecimento4)
trainer.train(Agradecimento5)

trainer.train(meio1)
trainer.train(meio2)
trainer.train(meio3)
trainer.train(silencio)

trainer.train(instalacao1)
trainer.train(instalacao2)
trainer.train(instalacao3)
trainer.train(instalacao4)
trainer.train(instalacao5)
trainer.train(instalacao6)
trainer.train(instalacao7)

trainer.train(alinhamentoInspecao)

trainer.train(desmontagem1)
trainer.train(desmontagem2)
trainer.train(desmontagem3)

```

```
trainer.train(desmontagem4)
trainer.train(desmontagem5)
trainer.train(desmontagem6)
trainer.train(desmontagem7)
trainer.train(demontcaval1)
trainer.train(demontcaval2)
trainer.train(demontcaval3)

trainer.train(montar1)
trainer.train(montar2)
trainer.train(montar3)
trainer.train(montar4)
trainer.train(montar5)
trainer.train(montar6)
trainer.train(montar7)

trainer.train(trocaRolamento1)
trainer.train(trocaRolamento2)
trainer.train(trocaRolamento3)
trainer.train(trocaRolamento4)
trainer.train(trocaRolamento5)
trainer.train(trocaRolamento6)

trainer.train(rolamento1)
trainer.train(rolamento2)
trainer.train(rolamento3)
trainer.train(rolamento4)
trainer.train(lubrificacao1)
trainer.train(lubrificacao2)
trainer.train(lubrificacao3)
trainer.train(lubrificacao4)

treinador = ChatterbotCorpusTrainer(chatbot)
treinador.train('Chatterbot.corpus.portuguese.greetings')
```

Local: Treino.py

APÊNDICE I – CÓDIGO FONTE DO ARQUIVO TESTES.PY

```
import unittest
from assistente import chatbot

class TestChatbot(unittest.TestCase):

    def setUp(self):
        self.chatbot = chatbot

    #testes de busca por rotor
    def test_busca_rotor1(self):
        resposta = self.chatbot.get_response('Qual o tamanho do rotor?')
        self.assertEqual(resposta.text, 'O rotor mede 40.0cm')
    def test_busca_rotor2(self):
        resposta = self.chatbot.get_response('Quanto mede o rotor?')
        self.assertEqual(resposta.text, 'O rotor mede 40.0cm')
    def test_busca_rotor3(self):
        resposta = self.chatbot.get_response('Preciso do tamanho do rotor')
        self.assertEqual(resposta.text, 'O rotor mede 40.0cm')
    def test_busca_rotor4(self):
        resposta = self.chatbot.get_response('Qual é o rotr?')
        self.assertEqual(resposta.text, 'O rotor mede 40.0cm')

    #testes de busca por velocidade
    def test_busca_velocidade1(self):
        resposta = self.chatbot.get_response('Qual a velocidade da bomba?')
        self.assertEqual(resposta.text, 'Essa máquina opera á 3500rpm')
    def test_busca_velocidade2(self):
        resposta = self.chatbot.get_response('De quanto é a rotação deste equipamento?')
        self.assertEqual(resposta.text, 'Essa máquina opera á 3500rpm')
    def test_busca_velocidade3(self):
        resposta = self.chatbot.get_response('Qual a velocidade da bomba?')
        self.assertEqual(resposta.text, 'Essa máquina opera á 3500rpm')
    def test_busca_velocidade4(self):
        resposta = self.chatbot.get_response('Qual a velosidade de rotaçã?')
        self.assertEqual(resposta.text, 'Essa máquina opera á 3500rpm')

    #testes de busca por produto
    def test_busca_produto1(self):
        resposta = self.chatbot.get_response('Qual o produto que esta bomba opera?')
        self.assertEqual(resposta.text, 'Esta bomba está operando com Água')
    def test_busca_produto2(self):
        resposta = self.chatbot.get_response('Que tipo de produto estamos trabalhando?')
        self.assertEqual(resposta.text, 'Esta bomba está operando com Água')
    def test_busca_produto3(self):
        resposta = self.chatbot.get_response('Com que produto esssa máquina trabalha?')
        self.assertEqual(resposta.text, 'Esta bomba está operando com Água')
    def test_busca_produto4(self):
        resposta = self.chatbot.get_response('Qual é o producto dessa maquina?')
        self.assertEqual(resposta.text, 'Esta bomba está operando com Água')

    #teste de busca por modelo
    def test_busca_modelo1(self):
        resposta = self.chatbot.get_response('Qual o modelo da bomba?')
        self.assertEqual(resposta.text, 'Essa máquina é uma bomba KSB ETA 125-40')
    def test_busca_modelo2(self):
        resposta = self.chatbot.get_response('Preciso do modelo do equipamento')
        self.assertEqual(resposta.text, 'Essa máquina é uma bomba KSB ETA 125-40')
    def test_busca_modelo3(self):
        resposta = self.chatbot.get_response('Você sabe qual é o modelo?')
        self.assertEqual(resposta.text, 'Essa máquina é uma bomba KSB ETA 125-40')
    def test_busca_modelo4(self):
        resposta = self.chatbot.get_response('Qual o modlo mesmo?')
        self.assertEqual(resposta.text, 'Essa máquina é uma bomba KSB ETA 125-40')

    #teste de busca por empresa
    def test_busca_empresa1(self):
        resposta = self.chatbot.get_response('De que empresa é essa bomba?')
        self.assertEqual(resposta.text, 'A bomba foi feita pela KSB')
    def test_busca_empresa2(self):
        resposta = self.chatbot.get_response('Qual a empresa que fabricou essa bomba?')
        self.assertEqual(resposta.text, 'A bomba foi feita pela KSB')
```

```

def test_busca_empresa3(self):
    resposta = self.chatbot.get_response('Preciso saber a empresa que fabricou esse equipamento')
    self.assertEqual(resposta.text, 'A bomba foi feita pela KSB')
def test_busca_empresa4(self):
    resposta = self.chatbot.get_response('Que empresa que vabrca isso?')
    self.assertEqual(resposta.text, 'A bomba foi feita pela KSB')

#teste de busca por codigo
def test_busca_codigo1(self):
    resposta = self.chatbot.get_response('Qual o código dessa bomba mesmo?')
    self.assertEqual(resposta.text, 'Essa bomba está registrada como 123456')
def test_busca_codigo2(self):
    resposta = self.chatbot.get_response('Esqueci o codigo dessa bomba')
    self.assertEqual(resposta.text, 'Essa bomba está registrada como 123456')
def test_busca_codigo3(self):
    resposta = self.chatbot.get_response('Preciso confirmar o codigo da bomba')
    self.assertEqual(resposta.text, 'Essa bomba está registrada como 123456')
def test_busca_codigo4(self):
    resposta = self.chatbot.get_response('Como que é o codgo?')
    self.assertEqual(resposta.text, 'Essa bomba está registrada como 123456')

#teste de busca por marca do oleo
def test_oleo_marca1(self):
    resposta = self.chatbot.get_response('Que tipo de oleo eu uso?')
    self.assertEqual(resposta.text, 'Essa bomba usa óleo marbrax tr 52')
def test_oleo_marca2(self):
    resposta = self.chatbot.get_response('Qual a marca do lubrificante?')
    self.assertEqual(resposta.text, 'Essa bomba usa óleo marbrax tr 52')
def test_oleo_marca3(self):
    resposta = self.chatbot.get_response('Que marca de oleo essa máquina usa?')
    self.assertEqual(resposta.text, 'Essa bomba usa óleo marbrax tr 52')
def test_oleo_marca4(self):
    resposta = self.chatbot.get_response('Que marka de óleo que é mesmo??')
    self.assertEqual(resposta.text, 'Essa bomba usa óleo marbrax tr 52')

#teste de busca por frequencia de troca
def test_oleo_frequencial(self):
    resposta = self.chatbot.get_response('Qual o periodo para troca de óleo?')
    self.assertEqual(resposta.text, 'O óleo precisa ser trocado a cada 365 dias')
def test_oleo_frequencia2(self):
    resposta = self.chatbot.get_response('Com que frequencia eu preciso trocar o óleo?')
    self.assertEqual(resposta.text, 'O óleo precisa ser trocado a cada 365 dias')
def test_oleo_frequencia3(self):
    resposta = self.chatbot.get_response('Preciso saber o periodo para programar a próxima troca de óleo')
    self.assertEqual(resposta.text, 'O óleo precisa ser trocado a cada 365 dias')
def test_oleo_frequencia4(self):
    resposta = self.chatbot.get_response('Qual que é a frquncia da troca?')
    self.assertEqual(resposta.text, 'O óleo precisa ser trocado a cada 365 dias')

#teste de busca por fabricante do oleo
def test_oleo_fabricante1(self):
    resposta = self.chatbot.get_response('qual é a fabricante desse lubrificante?')
    self.assertEqual(resposta.text, 'Essa marca é produzida pela PETROBRAS')
def test_oleo_fabricante2(self):
    resposta = self.chatbot.get_response('Preciso saber que empresa produz esse óleo')
    self.assertEqual(resposta.text, 'Essa marca é produzida pela PETROBRAS')
def test_oleo_fabricante3(self):
    resposta = self.chatbot.get_response('Você sabe me dizer qe empresa é a fabricante desse óleo?')
    self.assertEqual(resposta.text, 'Essa marca é produzida pela PETROBRAS')
def test_oleo_fabricante4(self):
    resposta = self.chatbot.get_response('Mas de qual empesa que é o óloe')
    self.assertEqual(resposta.text, 'Essa marca é produzida pela PETROBRAS')

#teste de busca por volume de oleo
def test_oleo_volumel(self):
    resposta = self.chatbot.get_response('Qual o volume de óleo que eu preciso?')
    self.assertEqual(resposta.text, 'A bomba precisa de 1.2L de óleo')
def test_oleo_volume2(self):
    resposta = self.chatbot.get_response('Quanto de lubrificante a bomba usa?')
    self.assertEqual(resposta.text, 'A bomba precisa de 1.2L de óleo')
def test_oleo_volume3(self):
    resposta = self.chatbot.get_response('Qual a quantidade de óleo que eu preciso mesmo?')
    self.assertEqual(resposta.text, 'A bomba precisa de 1.2L de óleo')

```

```

def test_oleo_volume4(self):
    resposta = self.chatbot.get_response('Qanto de oleo que vai?')
    self.assertEqual(resposta.text, 'A bomba precisa de 1.2L de óleo')

#teste de checklist
def test_checklist1(self):
    conversa = [
        ('Olá, preciso começar o processo de checklist', 'O nível do óleo está
adequado?'),
        ('Sim', 'Por favor verifique se os parafusos estão bem apertados'),
        ('Sim', 'O equipamento está devidamente nivelado?'),
        ('Sim', 'O motor está devidamente acoplado a bomba?'),
        ('Sim', 'O eixo consegue movimentar livremente?'),
        ('Sim', 'Tudo certo com a gaxeta?'),
        ('Sim', 'Tudo conferido')
    ]

    for mensagem, resposta_esperada in conversa:
        resposta = self.chatbot.get_response(mensagem)
        self.assertEqual(resposta.text, resposta_esperada)

def test_checklist2(self):
    conversa = [
        ('Olá, preciso começar o processo de checklist', 'O nível do óleo está
adequado?'),
        ('Tudo certo com o nível de óleo', 'Por favor verifique se os parafusos estão bem
apertados'),
        ('tudo certo com os parafusos', 'O equipamento está devidamente nivelado?'),
        ('Sim, está nivelado', 'O motor está devidamente acoplado a bomba?'),
        ('Está', 'O eixo consegue movimentar livremente?'),
        ('Sim, tudo certo', 'Tudo certo com a gaxeta?'),
        ('Gaxeta conferida', 'Tudo conferido')
    ]

    for mensagem, resposta_esperada in conversa:
        resposta = self.chatbot.get_response(mensagem)
        self.assertEqual(resposta.text, resposta_esperada)

#testes dos diálogos treinados
def test_instalacao_bombal(self):
    instalacao1 = [
        "preciso instalar a bomba",
        "fundação pronta",
        "base pronta",
        "parafusos apertados",
        "tubulações fixadas",
        "acoplamento fixado",
        "motor pronto",
    ]

    conversa_esperada = [
        "ok, primeiro verifique se a fundação está seca e limpa para receber o
equipamento",
        "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a
fixe bem com argamassa",
        "espere a argamassa secar e então aperte os parafusos de ancoragem bem e
uniformemente",
        "se as tubulações encostarem sem esforço nas flanges, fixe-as",
        "agora fixe cuidadosamente o acoplamento ao rotor da bomba",
        "ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
        "tudo certo então, algo mais em que eu possa ajudar?"
    ]

    mensagens_chatbot = []
    for mensagem in instalacao1:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot.append(resposta.text)

    self.assertEqual(mensagens_chatbot, conversa_esperada)

def test_instalacao_bomba2(self):
    instalacao_alterada = [
        "preciso instalar a bomba",
        "fundação está pronta",
        "base fixada",
        "parafusos apertados corretamente",
        "tubulações fixadas",
        "acoplamento preso",
        "motor instalado",
    ]

```

```

    ]

    conversa_esperada_alterada = [
        "ok, primeiro verifique se a fundação está seca e limpa para receber o
equipamento",
        "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a
fixe bem com argamassa",
        "espere a argamassa secar e então aperte os parafusos de ancoragem bem e
uniformemente",
        "se as tubulações encostarem nas flanges sem esforço, fixe-as",
        "agora prenda cuidadosamente o acoplamento ao rotor da bomba",
        "ótimo, agora podemos instalar o motor, mas certifique-se de que está alinhado
corretamente",
        "tudo certo então, algo mais em que eu possa ajudar?"
    ]

    mensagens_chatbot_alteradas = []
    for mensagem in instalacao_alterada:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_alteradas.append(resposta.text)

    self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)
def test_instalacao_bomba3(self):
    instalacao_alterada2 = [
        "vamos começar a instalação",
        "podemos fixar a base",
        "pronto para parafusar",
        "podemos ir para as tubulações",
        "agora para o acoplamento",
        "podemos instalar o motor",
        "tudo pronto",
    ]

    conversa_esperada_alterada2 = [
        "ok, primeiro verifique se a fundação está seca e limpa para receber o
equipamento",
        "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a
fixe bem com argamassa",
        "espere a argamassa secar e então aperte os parafusos de ancoragem bem e
uniformemente",
        "se as tubulações encostarem sem esforço nas flanges, fixe-as",
        "agora fixe cuidadosamente o acoplamento ao rotor da bomba",
        "ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
        "tudo certo então, algo mais em que eu possa ajudar?"
    ]

    mensagens_chatbot_alteradas2 = []
    for mensagem in instalacao_alterada2:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_alteradas2.append(resposta.text)

    self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)
def test_instalacao_bomba4(self):
    instalacao_alterada2 = [
        "vamo instala a bomba",
        "Fundacao seca e lmpa",
        "podemos parafusar",
        "Parafusos apertado",
        "agora é a hora do acoplamento",
        "vamos intalar o motor",
        "tudo pronto",
    ]

    conversa_esperada_alterada2 = [
        "ok, primeiro verifique se a fundação está seca e limpa para receber o
equipamento",
        "certo, agora garanta que a base da bomba esteja devidamente nivelada e então a
fixe bem com argamassa",
        "espere a argamassa secar e então aperte os parafusos de ancoragem bem e
uniformemente",
        "se as tubulações encostarem sem esforço nas flanges, fixe-as",
        "agora fixe cuidadosamente o acoplamento ao rotor da bomba",
        "ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado",
        "tudo certo então, algo mais em que eu possa ajudar?"
    ]

    mensagens_chatbot_alteradas2 = []

```

```

for mensagem in instalacao_alterada2:
    resposta = self.chatbot.get_response(mensagem)
    mensagens_chatbot_alteradas2.append(resposta.text)

self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)

def test_desmontagem_bomba1(self):
    desmontagem1 = [
        'preciso desmontar a bomba, como faço?',
        'a bomba ainda está na base',
        'a bomba já está fora da base',
        'tampa de aspiração removida',
        'porca do rotor retirada',
        'arruela de segurança retirada',
        'tampa do cavalete dos mancais retirada',
        'não estou conseguindo retirar o motor',
        'rotor retirado',
    ]

    conversa_esperada_desmontagem = [
        'primeiro preciso saber, a bomba já está fora da base?',
        'então primeiro você deve tirá-la da base',
        'ótimo, agora remova a tampa de aspiração',
        'agora solte a porca do rotor',
        'muito bem, agora retire a arruela de segurança',
        'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
        'então finalmente retire o rotor, mas use pancadas leves',
        'então tente usar líquidos solventes de ferrugem',
        'muito bem, então bomba desmontada'
    ]

    mensagens_chatbot_desmontagem = []
    for mensagem in desmontagem1:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_desmontagem.append(resposta.text)

    self.assertEqual(mensagens_chatbot_desmontagem, conversa_esperada_desmontagem)

def test_desmontagem_bomba2(self):
    desmontagem4 = [
        'vamos desmontar a bomba',
        'ainda não tiramos a bomba da base',
        'a bomba está fora da base',
        'a bomba já está sem a tampa de aspiração',
        'o rotor está sem a porca',
        'já tirei a arruela de segurança',
        'tampa do cavalete do lado do acionamento solta',
        'o rotor não quer sair',
        'deu certo, tirei o rotor',
    ]

    conversa_esperada_desmontagem4 = [
        'primeiro preciso saber, a bomba já está fora da base?',
        'então primeiro você deve tirá-la da base',
        'ótimo, agora remova a tampa de aspiração',
        'agora solte a porca do rotor',
        'muito bem, agora retire a arruela de segurança',
        'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
        'então finalmente retire o rotor, mas use pancadas leves',
        'então tente usar líquidos solventes de ferrugem',
        'muito bem, então bomba desmontada'
    ]

    mensagens_chatbot_desmontagem4 = []
    for mensagem in desmontagem4:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_desmontagem4.append(resposta.text)

    self.assertEqual(mensagens_chatbot_desmontagem4, conversa_esperada_desmontagem4)

def test_desmontagem_bomba3(self):
    desmontagem5 = [
        'preciso desmontar a bomba',
        'não, ainda está na base',
        'a bomba já está fora da base',
        'tampa de aspiração retirada',
        'já está sem a porca do rotor',
        'tirei a arruela de segurança',
        'tampa do cavalete dos mancais do lado do acionamento retirada',
        'Não dá, o rotor não sai',
    ]

```

```

        'já soltei o rotor',
    ]

    conversa_esperada_desmontagem5 = [
        'primeiro preciso saber, a bomba já está fora da base?',
        'então primeiro você deve tirá-la da base',
        'ótimo, agora remova a tampa de aspiração',
        'agora solte a porca do rotor',
        'muito bem, agora retire a arruela de segurança',
        'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
        'então finalmente retire o rotor, mas use pancadas leves',
        'então tente usar líquidos solventes de ferrugem',
        'muito bem, então bomba desmontada'
    ]

    mensagens_chatbot_desmontagem5 = []
    for mensagem in desmontagem5:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_desmontagem5.append(resposta.text)

    self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)
def test_desmontagem_bomba4(self):
    desmontagem5 = [
        'vamos demontar a bomba',
        'a bomba já tá fora da base',
        'tampa de aspiração retirada',
        'já está sem a porca do rotor',
        'tirei a arruela de segurança',
        'tampa do cavalette dos mancais do lado do acionamento retirada',
        'já soltei o rotor',
    ]

    conversa_esperada_desmontagem5 = [
        'primeiro preciso saber, a bomba já está fora da base?',
        'ótimo, agora remova a tampa de aspiração',
        'agora solte a porca do rotor',
        'muito bem, agora retire a arruela de segurança',
        'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
        'então finalmente retire o rotor, mas use pancadas leves',
        'muito bem, então bomba desmontada'
    ]

    mensagens_chatbot_desmontagem5 = []
    for mensagem in desmontagem5:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_desmontagem5.append(resposta.text)

    self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)
def test_troca_rolamento1(self):
    trocaRolamento1 = [
        'preciso trocar o rolamento',
        'tirei o rolamento',
        'o novo rolamento já está no óleo',
        'novo rolamento fixado'
    ]

    conversa_esperada_troca_rolamento = [
        'para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves',
        'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo a 80°C',
        'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
        'agora apenas se certifique que o rolamento novo está rodando sem problemas'
    ]

    mensagens_chatbot_troca_rolamento = []
    for mensagem in trocaRolamento1:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_troca_rolamento.append(resposta.text)

    self.assertEqual(mensagens_chatbot_troca_rolamento, conversa_esperada_troca_rolamento)
def test_troca_rolamento2(self):
    trocaRolamento3 = [
        'vou trocar o rolamento, você pode me ajudar?',
        'o rolamento antigo já está fora do eixo',
    ]

```

```

        'rolamento novo preparado',
        'assentei o rolamento novo',
    ]

    conversa_esperada_troca_rolamento3 = [
        'claro, para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves',
        'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
        'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
        'agora apenas se certifique que o rolamento novo está rodando sem problemas'
    ]

    mensagens_chatbot_troca_rolamento3 = []
    for mensagem in trocaRolamento3:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_troca_rolamento3.append(resposta.text)

    self.assertEqual(mensagens_chatbot_troca_rolamento3,
conversa_esperada_troca_rolamento3)

    def test_troca_rolamento(self):
        trocaRolamento3 = [
            'vou trocar o rolamento, você pode me ajudar?',
            'claro, para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves',
            'o rolamento antigo já está fora do eixo',
            'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
            'rolamento novo preparado',
            'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
            'assentei o rolamento novo',
            'agora apenas se certifique que o rolamento novo está rodando sem problemas'
        ]

        conversa_esperada_troca_rolamento3 = [
            'claro, para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves',
            'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
            'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
            'agora apenas se certifique que o rolamento novo está rodando sem problemas'
        ]

        mensagens_chatbot_troca_rolamento3 = []
        for mensagem in trocaRolamento3:
            resposta = self.chatbot.get_response(mensagem)
            mensagens_chatbot_troca_rolamento3.append(resposta.text)

        self.assertEqual(mensagens_chatbot_troca_rolamento3,
conversa_esperada_troca_rolamento3)
    def test_troca_rolamento3(self):
        trocaRolamento5 = [
            'preciso trocar o rolamento',
            'tirei o rolamento antigo',
            'já preparei o rolamento novo',
            'o novo rolamento já está no lugar',
        ]

        conversa_esperada_troca_rolamento5 = [
            'para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves',
            'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo á 80°C',
            'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo',
            'agora apenas se certifique que o rolamento novo está rodando sem problemas'
        ]

        mensagens_chatbot_troca_rolamento5 = []
        for mensagem in trocaRolamento5:
            resposta = self.chatbot.get_response(mensagem)
            mensagens_chatbot_troca_rolamento5.append(resposta.text)

```

```

        self.assertEqual(mensagens_chatbot_troca_rolamento5,
conversa_esperada_troca_rolamento5)
    def test_troca_rolamento4(self):
        trocaRolamento5 = [
            'preciso trocar o rolamentu',
            'tirei o rorolamento antigo',
            'já preparei o rolamnto novo',
            'o novo rolamento já está no ligr',
        ]

        conversa_esperada_troca_rolamento5 = [
            'para isso preciso que primero você aqueça o rolamento sem aquecer o eixo e tire o
rolamento com pancadas leves',
            'ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em
óleo á 80°c',
            'muito bem, agora assente o rolamento no eixo até o encosto se necessário com um
tubo',
            'agora apenas se certifique que o rolamento novo está rodando sem problemas'
        ]
        mensagens_chatbot_troca_rolamento5 = []
        for mensagem in trocaRolamento5:
            resposta = self.chatbot.get_response(mensagem)
            mensagens_chatbot_troca_rolamento5.append(resposta.text)
        self.assertEqual(mensagens_chatbot_troca_rolamento5,
conversa_esperada_troca_rolamento5)

    def test_desmontagem_cavalete(self):
        demontcaval2 = [
            'preciso desmontar o cavalete, você pode me ajudar?',
            'já separei o cavalete da bomba',
            'o cavalete já está fora do eixo',
            'tampa do mancal solta',
        ]

        conversa_esperada_desmontagem_cavalete = [
            'para desmontar o cavalete primeiro solte o aperta-gaxeta e separe o corpo da
bomba do cavalete',
            'tire a vareta indicadora e retire o cavalete cuidadosamente do eixo',
            'por fim solte a tampa do mancal do lado da gaxeta',
            'terminamos de desmontar o cavalete do mancal, mais alguma coisa em que eu possa
ajudar?'
        ]

        mensagens_chatbot_desmontagem_cavalete = []
        for mensagem in demontcaval2:
            resposta = self.chatbot.get_response(mensagem)
            mensagens_chatbot_desmontagem_cavalete.append(resposta.text)

        self.assertEqual(mensagens_chatbot_desmontagem_cavalete,
conversa_esperada_desmontagem_cavalete)

    def test_montagem_bomba1(self):
        montarl = [
            'agora vamos remontar a bomba',
            'rotor já está no lugar',
            'tampa do cavalete presa',
            'arruela de segurança colocada',
            'porca do rotor presa',
        ]

        conversa_esperada_montagem_bomba = [
            'então primeiro insira o rotor de volta',
            'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
            'coloque novamente a arruela de segurança',
            'em seguida prenda novamente a porca do rotor',
            'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar
ao campo'
        ]

        mensagens_chatbot_montagem_bomba = []
        for mensagem in montarl:
            resposta = self.chatbot.get_response(mensagem)
            mensagens_chatbot_montagem_bomba.append(resposta.text)

        self.assertEqual(mensagens_chatbot_montagem_bomba, conversa_esperada_montagem_bomba)
    def test_montagem_bomba2(self):
        montarl = [

```

```

        'vamos montar a bomba novamente',
        'o rotor já está no lugar',
        'tampa do cavalete está segura',
        'a arruela de segurança foi colocada',
        'porca do rotor está fixa',
    ]

    conversa_esperada_montagem_bomba = [
        'então primeiro insira o rotor de volta',
        'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
        'coloque novamente a arruela de segurança',
        'em seguida prenda novamente a porca do rotor',
        'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar
ao campo'
    ]

    mensagens_chatbot_montagem_bomba = []
    for mensagem in montar1:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_montagem_bomba.append(resposta.text)

    self.assertEqual(mensagens_chatbot_montagem_bomba, conversa_esperada_montagem_bomba)
def test_montagem_bomba3(self):
    montar4 = [
        'vamos montar a bomba novamente',
        'o rotor já foi inserido no lugar',
        'a tampa do cavalete dos mancais já está fixada no lado do acionamento',
        'a arruela de segurança está colocada',
        'a porca do rotor está presa',
    ]

    conversa_esperada_montagem_bomba = [
        'então primeiro insira o rotor de volta',
        'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
        'coloque novamente a arruela de segurança',
        'em seguida prenda novamente a porca do rotor',
        'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar
ao campo'
    ]

    mensagens_chatbot_montagem_bomba = []
    for mensagem in montar4:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_montagem_bomba.append(resposta.text)

    self.assertEqual(mensagens_chatbot_montagem_bomba, conversa_esperada_montagem_bomba)
def test_montagem_bomba4(self):
    montar1 = [
        'agora vamos remotar a bomba',
        'rotor já está no lugar',
        'tapa do cavalette presa',
        'aruela de segurança colocada',
        'porca do rotor pressa',
    ]

    conversa_esperada_montagem_bomba = [
        'então primeiro insira o rotor de volta',
        'agora prenda a tampa do cavalete dos mancais do lado do acionamento',
        'coloque novamente a arruela de segurança',
        'em seguida prenda novamente a porca do rotor',
        'insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar
ao campo'
    ]

    mensagens_chatbot_montagem_bomba = []
    for mensagem in montar1:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_montagem_bomba.append(resposta.text)

    self.assertEqual(mensagens_chatbot_montagem_bomba, conversa_esperada_montagem_bomba)

def test_falha_rolamento1(self):
    rolamento2 = [
        'o que pode ter ocasionar falha de rolamento?',
        'notei um desalinhamento',
        'não há problema com a lubrificação ou com o rolamento',
    ]

```

```

    ]

    conversa_esperada_falha_rolamento = [
        'pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o
        alinhamento do eixo de rotação e troque o rolamento',
        'então por favor realinhe o conjunto',
        'então pode ser sobrecarga na bomba'
    ]

    mensagens_chatbot_falha_rolamento = []
    for mensagem in rolamento2:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_falha_rolamento.append(resposta.text)

    self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)
def test_falha_rolamento2(self):
    rolamento2 = [
        'O que quebrou o rolamento?',
        'está desalinhado',
        'não achei problema com lubrificação ou rolamento',
    ]

    conversa_esperada_falha_rolamento = [
        'pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o
        alinhamento do eixo de rotação e troque o rolamento',
        'então por favor realinhe o conjunto',
        'então pode ser sobrecarga na bomba'
    ]

    mensagens_chatbot_falha_rolamento = []
    for mensagem in rolamento2:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_falha_rolamento.append(resposta.text)

    self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

def test_lubrificacao1(self):
    lubrificacao1 = [
        'o óleo parece estar ruim',
        'o óleo está muito usado',
        'o óleo está sujo',
        'o óleo está quente demais',
    ]

    conversa_esperada_lubrificacao = [
        'por favor analise o óleo e verifique o que tem de errado com ele',
        'então vamos precisar trocar mas se ele está ficando gasto muito cedo você vai
        precisar fazer uma análise mais minuciosa com a equipe técnica',
        'então será necessário trocá-lo e realizar uma análise para averiguar a causa
        dessas impurezas',
        'verifique o nível de óleo na vareta indicadora, o alinhamento do conjunto'
    ]

    mensagens_chatbot_lubrificacao = []
    for mensagem in lubrificacao1:
        resposta = self.chatbot.get_response(mensagem)
        mensagens_chatbot_lubrificacao.append(resposta.text)

    self.assertEqual(mensagens_chatbot_lubrificacao, conversa_esperada_lubrificacao)
def test_lubrificacao2(self):
    lubrificacao1 = [
        'tem algum problema com o óleo',
        'o óleo está gasto',
        'o óleo tem impurezas',
        'o óleo está muito quente',
    ]

    conversa_esperada_lubrificacao = [
        'por favor analise o óleo e verifique o que tem de errado com ele',
        'então vamos precisar trocar mas se ele está ficando gasto muito cedo você vai
        precisar fazer uma análise mais minuciosa com a equipe técnica',
        'então será necessário trocá-lo e realizar uma análise para averiguar a causa
        dessas impurezas',
        'verifique o nível de óleo na vareta indicadora, o alinhamento do conjunto'
    ]

    mensagens_chatbot_lubrificacao = []

```

```
for mensagem in lubrificacao1:
    resposta = self.chatbot.get_response(mensagem)
    mensagens_chatbot_lubrificacao.append(resposta.text)

self.assertEqual(mensagens_chatbot_lubrificacao, conversa_esperada_lubrificacao)

if __name__ == '__main__':
    unittest.main()
```

APÊNDICE J – ENTRADAS E SAÍDAS ESPERADAS DOS TESTES

Entada	Saída esperada
Qual o tamanho do rotor?	O rotor mede 40.0cm
Quanto mede o rotor?	O rotor mede 40.0cm
Preciso do tamanho do rotor	O rotor mede 40.0cm
Qual é o rotr?	O rotor mede 40.0cm
Qual a velocidade da bomba?	Essa máquina opera á 3500rpm
De quanto é a rotação deste equipamento?	Essa máquina opera á 3500rpm
Qual a velocidade da bomba?	Essa máquina opera á 3500rpm
Qual a velocidade de rotação	Essa máquina opera á 3500rpm
Qual o produto que esta bomba opera?	Esta bomba está operando com Água
Que tipo de produto estamos trabalhando?	Esta bomba está operando com Água
Com que produto essa máquina trabalha?	Esta bomba está operando com Água
Qual é o producto dessa maquina?	Esta bomba está operando com Água
Qual o modelo da bomba?	Essa máquina é uma bomba KSB ETA 125-40
Preciso do modelo do equipamento	Essa máquina é uma bomba KSB ETA 125-40
Você sabe qual é o modelo?	Essa máquina é uma bomba KSB ETA 125-40
Qual o modelo mesmo?	Essa máquina é uma bomba KSB ETA 125-40
De que empresa é essa bomba?	A bomba foi feita pela KSB
Qual a empresa que fabricou essa bomba?	A bomba foi feita pela KSB
Preciso saber a empresa que fabricou esse equipamento	A bomba foi feita pela KSB
Que empresa que fabrica isso?	A bomba foi feita pela KSB
Qual o código dessa bomba mesmo?	Essa bomba está registrada como 123456
Esqueci o codigo dessa bomba	Essa bomba está registrada como 123456
Preciso confirmar o codigo da bomba	Essa bomba está registrada como 123456
Como que é o codgo?	Essa bomba está registrada como 123456
Que tipo de oleo eu uso?	Essa bomba usa óleo marbrax tr 52
Qual a marca do lubrificante?	Essa bomba usa óleo marbrax tr 52
Que marca de oleo essa máquina usa?	Essa bomba usa óleo marbrax tr 52
Que marka de óleo que é mesmo??	Essa bomba usa óleo marbrax tr 52
Qual o periodo para troca de óleo?	O oléo precisa ser trocado a cada 365 dias
Com que frequencia eu preciso trocar o óleo?	O oléo precisa ser trocado a cada 365 dias
Preciso saber o periodo para programar a próxima troca de óleo	O oléo precisa ser trocado a cada 365 dias
Qual que é a frquncia da troca?	O oléo precisa ser trocado a cada 365 dias
qual é a fabricante desse lubrificante?	Essa marca é produzida pela PETROBRAS
Preciso saber que empresa produz esse óleo	Essa marca é produzida pela PETROBRAS
Você sabe me dizer qe empresa é a fabricante desse óleo?	Essa marca é produzida pela PETROBRAS
Mas de qual empesa que é o óloe	Essa marca é produzida pela PETROBRAS
Qual o volume de óleo que eu preciso?	A bomba precisa de 1.2L de oléo
Quanto de lubrificante a bomba usa?	A bomba precisa de 1.2L de oléo
Qual a quantidade de óleo que eu preciso mesmo?	A bomba precisa de 1.2L de oléo
Qanto de oleo que vai?	A bomba precisa de 1.2L de oléo

Observação: Os erros de digitação são propositais.

Entada	Saída esperada
preciso instalar a bomba fundação pronta base pronta parafusos apertados tubulações fixadas acoplamento fixado motor pronto	ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente se as tubulações encostarem sem esforço nas flanges, fixe-as agora fixe cuidadosamente o acoplamento ao rotor da bomba ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado tudo certo então, algo mais em que eu possa ajudar?
preciso instalar a bomba fundação está pronta base fixada parafusos apertados corretamente tubulações fixadas acoplamento preso motor instalado	ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente se as tubulações encostarem sem esforço nas flanges, fixe-as agora fixe cuidadosamente o acoplamento ao rotor da bomba ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado tudo certo então, algo mais em que eu possa ajudar?
vamos começar a instalação podemos fixar a base pronto para parafusar podemos ir para as tubulações agora para o acoplamento podemos instalar o motor tudo pronto	ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente se as tubulações encostarem sem esforço nas flanges, fixe-as agora fixe cuidadosamente o acoplamento ao rotor da bomba ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado tudo certo então, algo mais em que eu possa ajudar?
vamo instala a bomba Fundacao seca e limpa podemos parafusar Parafusos apertado agora é a hora do acoplamento vamos intalar o motor tudo pronto	ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento certo, agora garanta que a base da bomba esteja devidamente nivelada e então a fixe bem com argamassa espere a argamassa secar e então aperte os parafusos de ancoragem bem e uniformemente se as tubulações encostarem sem esforço nas flanges, fixe-as agora fixe cuidadosamente o acoplamento ao rotor da bomba ótimo, agora podemos fixar o motor, mas tenha certeza que está tudo alinhado tudo certo então, algo mais em que eu possa ajudar?
preciso desmontar a bomba, como faço? a bomba ainda está na base a bomba já está fora da base tampa de aspiração removida porca do rotor retirada arruela de segurança retirada tampa do cavalete dos mancais retirada não estou conseguindo retirar o motor rotor retirado	primeiro preciso saber, a bomba já está fora da base? então primeiro você deve tirá-la da base ótimo, agora remova a tampa de aspiração agora solte a porca do rotor muito bem, agora retire a arruela de segurança a seguir solte a tampa do cavalete dos mancais do lado do acionamento então finalmente retire o rotor, mas use pancadas leves então tente usar líquidos solventes de ferrugem muito bem, então bomba desmontada
vamos desmontar a bomba ainda não tiramos a bomba da base a bomba está fora da base a bomba já está sem a tampa de aspiração o rotor está sem a porca já tirei a arruela de segurança tampa do cavalete do lado do acionamento solta o rotor não quer sair deu certo, tirei o rotor	primeiro preciso saber, a bomba já está fora da base? então primeiro você deve tirá-la da base ótimo, agora remova a tampa de aspiração agora solte a porca do rotor muito bem, agora retire a arruela de segurança a seguir solte a tampa do cavalete dos mancais do lado do acionamento então finalmente retire o rotor, mas use pancadas leves então tente usar líquidos solventes de ferrugem muito bem, então bomba desmontada
preciso desmontar a bomba não, ainda está na base a bomba já está fora da base tampa de aspiração retirada já está sem a porca do rotor tirei a arruela de segurança tampa do cavalete dos mancais do lado do acionamento retirada Não dá, o rotor não sai já soltei o rotor	primeiro preciso saber, a bomba já está fora da base? então primeiro você deve tirá-la da base ótimo, agora remova a tampa de aspiração agora solte a porca do rotor muito bem, agora retire a arruela de segurança a seguir solte a tampa do cavalete dos mancais do lado do acionamento então finalmente retire o rotor, mas use pancadas leves então tente usar líquidos solventes de ferrugem muito bem, então bomba desmontada
vamo demontar a bomba a boba já tá fora da base tampa de aspração retirada já está sem a poca do rotr tirei a arruela de segurança tampa do cavalette dos macais do lado do acionamento retirada já soutei o rotor	primeiro preciso saber, a bomba já está fora da base? ótimo, agora remova a tampa de aspiração agora solte a porca do rotor muito bem, agora retire a arruela de segurança a seguir solte a tampa do cavalete dos mancais do lado do acionamento então finalmente retire o rotor, mas use pancadas leves muito bem, então bomba desmontada
preciso trocar o rolamento tirei o rolamento o novo rolamento já está no óleo novo rolamento fixado	para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo a 80°C muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo agora apenas se certifique que o rolamento novo está rodando sem problemas
vou trocar o rolamento, você pode me ajudar? o rolamento antigo já está fora do eixo rolamento novo preparado assentei o rolamento novo	para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo a 80°C muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo agora apenas se certifique que o rolamento novo está rodando sem problemas
preciso trocar o rolamento tirei o rolamento antigo já preparei o rolamento novo o novo rolamento já está no lugar	para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo a 80°C muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo agora apenas se certifique que o rolamento novo está rodando sem problemas

Entrada	Saída esperada
preciso trocar o rolamento tirei o rolamento antigo já preparei o rolamento novo o novo rolamento já está no lugar	para isso preciso que primeiro você aqueça o rolamento sem aquecer o eixo e tire o rolamento com pancadas leves ótimo agora descarte o rolamento antigo devidamente e banhe o novo rolamento em óleo a 80°C muito bem, agora assente o rolamento no eixo até o encosto se necessário com um tubo agora apenas se certifique que o rolamento novo está rodando sem problemas
preciso desmontar o cavalete, você pode me ajudar? já separei o cavalete da bomba o cavalete já está fora do eixo tampa do mancal solta	para desmontar o cavalete primeiro solte o aperta-gaxeta e separe o corpo da bomba do cavalete tire a vareta indicadora e retire o cavalete cuidadosamente do eixo por fim solte a tampa do mancal do lado da gaxeta terminamos de desmontar o cavalete do mancal, mais alguma coisa em que eu possa ajudar?
agora vamos remontar a bomba rotor já está no lugar tampa do cavalete presa arruela de segurança colocada porca do rotor presa	então primeiro insira o rotor de volta agora prenda a tampa do cavalete dos mancais do lado do acionamento coloque novamente a arruela de segurança em seguida prenda novamente a porca do rotor insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo
vamos montar a bomba novamente o rotor já está no lugar tampa do cavalete está segura a arruela de segurança foi colocada porca do rotor está fixa	então primeiro insira o rotor de volta agora prenda a tampa do cavalete dos mancais do lado do acionamento coloque novamente a arruela de segurança em seguida prenda novamente a porca do rotor insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo
vamos montar a bomba novamente o rotor já foi inserido no lugar a tampa do cavalete dos mancais já está fixada no lado do acionamento a arruela de segurança está colocada a porca do rotor está presa	então primeiro insira o rotor de volta agora prenda a tampa do cavalete dos mancais do lado do acionamento coloque novamente a arruela de segurança em seguida prenda novamente a porca do rotor insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo
agora vamos remontar a bomba rotor já está no lugar tapa do cavalete presa arruela de segurança colocada porca do rotor presa	então primeiro insira o rotor de volta agora prenda a tampa do cavalete dos mancais do lado do acionamento coloque novamente a arruela de segurança em seguida prenda novamente a porca do rotor insira novamente a tampa de aspiração e assim a bomba estará pronta para retornar ao campo
o que pode ter ocasionar falha de rolamento? notei um desalinhamento não há problema com a lubrificação ou com o rolamento	pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o alinhamento do eixo de rotação e troque o rolamento então por favor realinhe o conjunto então pode ser sobrecarga na bomba
O que quebrou o rolamento? está desalinhado não achei problema com lubrificação ou rolamento	pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o alinhamento do eixo de rotação e troque o rolamento então por favor realinhe o conjunto então pode ser sobrecarga na bomba
o óleo parece estar ruim o óleo está muito usado o óleo está sujo o óleo está quente demais	por favor analise o óleo e verifique o que tem de errado com ele então vamos precisar trocar mas se ele está ficando gasto muito cedo você vai precisar fazer uma análise mais minuciosa com a equipe técnica então será necessário trocá-lo e realizar uma análise para averiguar a causa das impurezas verifique o nível de óleo na vareta indicadora, o alinhamento do conjunto
tem algum problema com o óleo o óleo está gasto o óleo tem impurezas o óleo está muito quente	por favor analise o óleo e verifique o que tem de errado com ele então vamos precisar trocar mas se ele está ficando gasto muito cedo você vai precisar fazer uma análise mais minuciosa com a equipe técnica então será necessário trocá-lo e realizar uma análise para averiguar a causa das impurezas verifique o nível de óleo na vareta indicadora, o alinhamento do conjunto
Olá, preciso começar o processo de checklist Sim Sim Sim Sim Sim	O nível do óleo está adequado? Por favor verifique se os parafusos estão bem apertados O equipamento está devidamente nivelado? O motor está devidamente acoplado a bomba? O eixo consegue movimentar livremente? Tudo certo com a gaxeta? Tudo conferido
Olá, preciso começar o processo de checklist Tudo certo com o nível de óleo tudo certo com os parafusos Sim, está nivelado Está Sim, tudo certo Gaxeta conferida	O nível do óleo está adequado? Por favor verifique se os parafusos estão bem apertados O equipamento está devidamente nivelado? O motor está devidamente acoplado a bomba? O eixo consegue movimentar livremente? Tudo certo com a gaxeta? Tudo conferido
Vamos fazer o checklist Nível de óleo conferido Parafusos apertados Tudo certo com o nível o eixo está acoplado a bomba consegue, sim Gaxeta pronta	O nível do óleo está adequado? Por favor verifique se os parafusos estão bem apertados O equipamento está devidamente nivelado? O motor está devidamente acoplado a bomba? O eixo consegue movimentar livremente? Tudo certo com a gaxeta? Tudo conferido
checklist sim, está parafusos apertados está nivelado está acoplado sim, sem problemas tudo certo	O nível do óleo está adequado? Por favor verifique se os parafusos estão bem apertados O equipamento está devidamente nivelado? O motor está devidamente acoplado a bomba? O eixo consegue movimentar livremente? Tudo certo com a gaxeta? Tudo conferido

APÊNDICE K – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA AB123456

Ran 65 tests in 311.796s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 104, in test_busca_codigo3
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como AB1234'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento
+ Essa bomba está registrada como AB1234

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 107, in test_busca_codigo4
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como AB1234'

- depende, onde estamos mesmo?
+ Essa bomba está registrada como AB1234

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 733, in test_checklist3
self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?
+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 423, in
test_desmontagem_bomba4

self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'

'primeiro preciso saber, a bomba já está fora da base?'

- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
'ótimo, agora remova a tampa de aspiração',
'agora solte a porca do rotor',
'muito bem, agora retire a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'então finalmente retire o rotor, mas use pancadas leves',

'muinto bem, então bomba desmontada']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 661, in test_falha_rolamento2

self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122 chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'

'então por favor realinhe o conjunto'

['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '

'alinhamento do eixo de rotação e troque o rolamento',

- 'Tudo certo com a gaxeta?',

- 'Tudo certo com a gaxeta?']

+ 'então por favor realinhe o conjunto',

+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '

+ 'condições indicadas']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 249, in test_instalacao_bomba2

self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[167 chars]0mm'] != ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'

'se as tubulações encostarem nas flanges s[15 chars]e-as'

Diff is 866 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteAB.py", line 303, in test_instalacao_bomba4

self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] != ['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'

'se as tubulações encostarem sem esforço n[15 chars]e-as'

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE L – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA ET5678

Ran 65 tests in 270.752s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteET.py", line 104, in test_busca_codigo3
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como ET5678'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como ET5678

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteET.py", line 107, in test_busca_codigo4
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como ET5678'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como ET5678

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteET.py", line 733, in test_checklist3
self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteET.py", line 423, in
test_desmontagem_bomba4

self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'

'primeiro preciso saber, a bomba já está fora da base?'

- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
 'ótimo, agora remova a tampa de aspiração',
 'agora solte a porca do rotor',
 'muito bem, agora retire a arruela de segurança',
 'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
 'então finalmente retire o rotor, mas use pancadas leves',
 'muito bem, então bomba desmontada']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteET.py", line 661, in test_falha_rolamento2

self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122 chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'

'então por favor realinhe o conjunto'

['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '
 'alinhamento do eixo de rotação e troque o rolamento',

- 'Tudo certo com a gaxeta?',

- 'Tudo certo com a gaxeta?']

+ 'então por favor realinhe o conjunto',

+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '

+ 'condições indicadas']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteET.py", line 249, in test_instalacao_bomba2

self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[167 chars]0mm'] != ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'
 'se as tubulações encostarem nas flanges s[15 chars]e-as'

Diff is 866 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteET.py", line 303, in test_instalacao_bomba4
 self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] !=
 ['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'
 'se as tubulações encostarem sem esforço n[15 chars]e-as'

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

Process finished with exit code 1

APÊNDICE M – RELATÓRIO DO UNITTEST PARA A BOMBA GA2005

Ran 65 tests in 305.774s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteGA.py", line 104, in test_busca_codigo3

self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como GA2005'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como GA2005

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteGA.py", line 107, in test_busca_codigo4

self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como GA2005'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como GA2005

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteGA.py", line 733, in test_checklist3

self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteGA.py", line 423, in

test_desmontagem_bomba4

self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'

'primeiro preciso saber, a bomba já está fora da base?'

```
- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
  'ótimo, agora remova a tampa de aspiração',
  'agora solte a porca do rotor',
  'muito bem, agora retire a arruela de segurança',
  'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
  'então finalmente retire o rotor, mas use pancadas leves',
  'muito bem, então bomba desmontada']
```

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteGA.py", line 661, in test_falha_rolamento2

self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122 chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'

'então por favor realinhe o conjunto'

['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '

'alinhamento do eixo de rotação e troque o rolamento',

- 'Tudo certo com a gaxeta?',

- 'Tudo certo com a gaxeta?']

+ 'então por favor realinhe o conjunto',

+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '

+ 'condições indicadas']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteGA.py", line 249, in test_instalacao_bomba2

self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[164 chars]0mm'] != ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'

'se as tubulações encostarem nas flanges s[15 chars]e-as'

Diff is 863 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteGA.py", line 303, in test_instalacao_bomba4

self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] != ['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'

'se as tubulações encostarem sem esforço n[15 chars]e-as'

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE N – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA JB1722

Ran 65 tests in 298.743s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteJB.py", line 104, in test_busca_codigo3

self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como JB1722'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como JB1722

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteJB.py", line 107, in test_busca_codigo4

self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como JB1722'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como JB1722

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteJB.py", line 733, in test_checklist3

self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteJB.py", line 423, in test_desmontagem_bomba4

self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'
'primeiro preciso saber, a bomba já está fora da base?'

- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
'ótimo, agora remova a tampa de aspiração',
'agora solte a porca do rotor',
'muito bem, agora retire a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'então finalmente retire o rotor, mas use pancadas leves',
'muito bem, então bomba desmontada']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PF\ChatbotV1\TesteJB.py", line 661, in test_falha_rolamento2
self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122
chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'
'então por favor realinhe o conjunto'

['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '
'alinhamento do eixo de rotação e troque o rolamento',

- 'Tudo certo com a gaxeta?',
- 'Tudo certo com a gaxeta?']
+ 'então por favor realinhe o conjunto',
+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '
+ 'condições indicadas']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PF\ChatbotV1\TesteJB.py", line 249, in test_instalacao_bomba2
self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[166 chars]0mm'] !=
['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem nas flanges s[15 chars]e-as'

Diff is 865 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteJB.py", line 303, in test_instalacao_bomba4

self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] != ['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'

'se as tubulações encostarem sem esforço n[15 chars]e-as'

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE O – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA MD1120

Ran 65 tests in 269.294s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteMD.py", line 104, in test_busca_codigo3
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como MD1120'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como MD1120

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteMD.py", line 107, in test_busca_codigo4
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como MD1120'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como MD1120

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteMD.py", line 733, in test_checklist3
self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteMD.py", line 423, in
test_desmontagem_bomba4

self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'

'primeiro preciso saber, a bomba já está fora da base?'

- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
'ótimo, agora remova a tampa de aspiração',
'agora solte a porca do rotor',
'muito bem, agora retire a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'então finalmente retire o rotor, mas use pancadas leves',
'muito bem, então bomba desmontada']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PF\ChatbotV1\TesteMD.py", line 661, in test_falha_rolamento2

self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122 chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'

'então por favor realinhe o conjunto'

['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '
'alinhamento do eixo de rotação e troque o rolamento',

- 'Tudo certo com a gaxeta?',

- 'Tudo certo com a gaxeta?']

+ 'então por favor realinhe o conjunto',

+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '

+ 'condições indicadas']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PF\ChatbotV1\TesteMD.py", line 249, in test_instalacao_bomba2

self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[169 chars]0mm'] != ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem nas flanges s[15 chars]e-as'
```

Diff is 868 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteMD.py", line 303, in test_instalacao_bomba4
    self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)
```

```
AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] !=
['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']
```

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem sem esforço n[15 chars]e-as'
```

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE P – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA NM8379

Ran 65 tests in 345.088s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteNM.py", line 104, in test_busca_codigo3
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como NM8379'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como NM8379

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteNM.py", line 107, in test_busca_codigo4
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como NM8379'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como NM8379

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteNM.py", line 733, in test_checklist3
self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteNM.py", line 423, in
test_desmontagem_bomba4

self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'

'primeiro preciso saber, a bomba já está fora da base?'

- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
'ótimo, agora remova a tampa de aspiração',
'agora solte a porca do rotor',
'muito bem, agora retire a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'então finalmente retire o rotor, mas use pancadas leves',
'muito bem, então bomba desmontada']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PF\ChatbotV1\TesteNM.py", line 661, in test_falha_rolamento2

self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122 chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'

'então por favor realinhe o conjunto'

['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '
'alinhamento do eixo de rotação e troque o rolamento',

- 'Tudo certo com a gaxeta?',

- 'Tudo certo com a gaxeta?']

+ 'então por favor realinhe o conjunto',

+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '

+ 'condições indicadas']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PF\ChatbotV1\TesteNM.py", line 249, in test_instalacao_bomba2

self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[169 chars]0mm'] != ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem nas flanges s[15 chars]e-as'
```

Diff is 868 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteNM.py", line 303, in test_instalacao_bomba4
    self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)
```

```
AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] !=
['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']
```

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem sem esforço n[15 chars]e-as'
```

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE Q – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA PT1322

Ran 65 tests in 327.370s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TestePT.py", line 104, in test_busca_codigo3
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como PT1322'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como PT1322

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TestePT.py", line 107, in test_busca_codigo4
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como PT1322'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como PT1322

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TestePT.py", line 734, in test_checklist3
self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TestePT.py", line 423, in
test_desmontagem_bomba4
self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'

'primeiro preciso saber, a bomba já está fora da base?'

```
- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
  'ótimo, agora remova a tampa de aspiração',
  'agora solte a porca do rotor',
  'muito bem, agora retire a arruela de segurança',
  'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
  'então finalmente retire o rotor, mas use pancadas leves',
  'muito bem, então bomba desmontada']
```

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TestePT.py", line 661, in test_falha_rolamento2

self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122 chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'

'então por favor realinhe o conjunto'

```
['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '
 'alinhamento do eixo de rotação e troque o rolamento',
- 'Tudo certo com a gaxeta?',
- 'Tudo certo com a gaxeta?']
+ 'então por favor realinhe o conjunto',
+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '
+ 'condições indicadas']
```

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TestePT.py", line 249, in test_instalacao_bomba2

self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[169 chars]0mm'] != ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem nas flanges s[15 chars]e-as'
```

Diff is 868 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TestePT.py", line 303, in test_instalacao_bomba4
    self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)
```

```
AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] !=
['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']
```

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem sem esforço n[15 chars]e-as'
```

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE R – Relatório de resultado do unittest para a bomba tc2709

Ran 65 tests in 277.553s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTC.py", line 104, in test_busca_codigo3

self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como TC2709'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como TC2709

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTC.py", line 107, in test_busca_codigo4

self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como TC2709'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como TC2709

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTC.py", line 733, in test_checklist3

self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTC.py", line 423, in

test_desmontagem_bomba4

self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'
 'primeiro preciso saber, a bomba já está fora da base?'

- ['então primeiro insira o rotor dentro da bomba',
 + ['primeiro preciso saber, a bomba já está fora da base?',
 'ótimo, agora remova a tampa de aspiração',
 'agora solte a porca do rotor',
 'muito bem, agora retire a arruela de segurança',
 'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
 'então finalmente retire o rotor, mas use pancadas leves',
 'muito bem, então bomba desmontada']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTC.py", line 661, in test_falha_rolamento2
 self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)
 AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122
 chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'
 'então por favor realinhe o conjunto'

['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '
 'alinhamento do eixo de rotação e troque o rolamento',
 - 'Tudo certo com a gaxeta?',
 - 'Tudo certo com a gaxeta?']
 + 'então por favor realinhe o conjunto',
 + 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '
 + 'condições indicadas']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTC.py", line 249, in test_instalacao_bomba2
 self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)
 AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[164 chars]0mm'] !=
 ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'
 'se as tubulações encostarem nas flanges s[15 chars]e-as'

Diff is 863 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTC.py", line 303, in test_instalacao_bomba4

self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] != ['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'

'se as tubulações encostarem sem esforço n[15 chars]e-as'

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE S – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA TT2332

Ran 65 tests in 277.505s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTT.py", line 104, in test_busca_codigo3
    self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')
```

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como TT2332'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como TT2332

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTT.py", line 107, in test_busca_codigo4
    self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')
```

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como TT2332'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como TT2332

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTT.py", line 733, in test_checklist3
    self.assertEqual(resposta.text, resposta_esperada)
```

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTT.py", line 423, in
test_desmontagem_bomba4
```

```
    self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)
```

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'

'primeiro preciso saber, a bomba já está fora da base?'

- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
'ótimo, agora remova a tampa de aspiração',
'agora solte a porca do rotor',
'muito bem, agora retire a arruela de segurança',
'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
'então finalmente retire o rotor, mas use pancadas leves',
'muito bem, então bomba desmontada']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PF\ChatbotV1\TesteTT.py", line 661, in test_falha_rolamento2

self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)

AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122 chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']

First differing element 1:

'Tudo certo com a gaxeta?'

'então por favor realinhe o conjunto'

['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '
'alinhamento do eixo de rotação e troque o rolamento',

- 'Tudo certo com a gaxeta?',

- 'Tudo certo com a gaxeta?']

+ 'então por favor realinhe o conjunto',

+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '

+ 'condições indicadas']

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PF\ChatbotV1\TesteTT.py", line 249, in test_instalacao_bomba2

self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)

AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[166 chars]0mm'] != ['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem nas flanges s[15 chars]e-as'
```

Diff is 865 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteTT.py", line 303, in test_instalacao_bomba4
    self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)
```

```
AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] !=
['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']
```

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
'se as tubulações encostarem sem esforço n[15 chars]e-as'
```

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE T – RELATÓRIO DE RESULTADO DO UNITTEST PARA A BOMBA XZ0987

Ran 65 tests in 278.545s

FAILED (failures=7, successes=58)

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteXZ.py", line 104, in test_busca_codigo3
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'ok, primeiro verifique se a fundação está[36 chars]ento' != 'Essa bomba está registrada como XZ0987'

- ok, primeiro verifique se a fundação está seca e limpa para receber o equipamento

+ Essa bomba está registrada como XZ0987

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteXZ.py", line 107, in test_busca_codigo4
self.assertEqual(resposta.text, f'Essa bomba está registrada como {self.codigo}')

AssertionError: 'depende, onde estamos mesmo?' != 'Essa bomba está registrada como XZ0987'

- depende, onde estamos mesmo?

+ Essa bomba está registrada como XZ0987

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteXZ.py", line 733, in test_checklist3
self.assertEqual(resposta.text, resposta_esperada)

AssertionError: 'O motor está devidamente acoplado a bomba?' != 'O eixo consegue movimentar livremente?'

- O motor está devidamente acoplado a bomba?

+ O eixo consegue movimentar livremente?

Failure

Traceback (most recent call last):

File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteXZ.py", line 423, in
test_desmontagem_bomba4

self.assertEqual(mensagens_chatbot_desmontagem5, conversa_esperada_desmontagem5)

AssertionError: Lists differ: ['então primeiro insira o rotor dentro da b[297 chars]ada'] != ['primeiro preciso saber, a bomba já está f[305 chars]ada']

First differing element 0:

'então primeiro insira o rotor dentro da bomba'

'primeiro preciso saber, a bomba já está fora da base?'

```
- ['então primeiro insira o rotor dentro da bomba',
+ ['primeiro preciso saber, a bomba já está fora da base?',
  'ótimo, agora remova a tampa de aspiração',
  'agora solte a porca do rotor',
  'muito bem, agora retire a arruela de segurança',
  'a seguir solte a tampa do cavalete dos mancais do lado do acionamento',
  'então finalmente retire o rotor, mas use pancadas leves',
  'muito bem, então bomba desmontada']
```

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteXZ.py", line 661, in test_falha_rolamento2
    self.assertEqual(mensagens_chatbot_falha_rolamento, conversa_esperada_falha_rolamento)
AssertionError: Lists differ: ['pod[122 chars]o', 'Tudo certo com a gaxeta?', 'Tudo certo com a gaxeta?'] != ['pod[122
chars]o', 'então por favor realinhe o conjunto', 'en[90 chars]das']
```

First differing element 1:

'Tudo certo com a gaxeta?'

'então por favor realinhe o conjunto'

```
['pode ser desalinhamento ou má lubrificação, por favor verifique o óleo e o '
  'alinhamento do eixo de rotação e troque o rolamento',
- 'Tudo certo com a gaxeta?',
- 'Tudo certo com a gaxeta?']
+ 'então por favor realinhe o conjunto',
+ 'então pode ser sobrecarga na bomba, reavalie se ela está atuando dentro das '
+ 'condições indicadas']
```

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteXZ.py", line 249, in test_instalacao_bomba2
    self.assertEqual(mensagens_chatbot_alteradas, conversa_esperada_alterada)
AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[166 chars]0mm'] !=
['ok,[271 chars]e', 'se as tubulações encostarem nas flanges s[228 chars]ar?']
```

First differing element 3:

'espere a argamassa secar e então aperte o[40 chars]ente'

'se as tubulações encostarem nas flanges s[15 chars]e-as'

Diff is 865 characters long. Set self.maxDiff to None to see it.

Failure

Traceback (most recent call last):

```
File "C:\Users\User\OneDrive\Documentos\PFC\ChatbotV1\TesteXZ.py", line 303, in test_instalacao_bomba4
    self.assertEqual(mensagens_chatbot_alteradas2, conversa_esperada_alterada2)
```

```
AssertionError: Lists differ: ['ok,[271 chars]e', 'espere a argamassa secar e então aperte o[256 chars]ar?'] !=
['ok,[271 chars]e', 'se as tubulações encostarem sem esforço n[212 chars]ar?']
```

First differing element 3:

```
'espere a argamassa secar e então aperte o[40 chars]ente'
```

```
'se as tubulações encostarem sem esforço n[15 chars]e-as'
```

Diff is 759 characters long. Set self.maxDiff to None to see it.

Process finished with exit code 1

APÊNDICE U – CODIGO FONTE DE TESTETEMPO.PY

```

from time import time

from assistente import chatbot

c = 0
TempoBusca = 0
TempoChecklist = 0
TempoGuia = 0
num = 42

#Processo de busca de dados técnicos
print('Processo de busca:')
mas = int(num/2)
for c in range(0, mas):
    #tempo de busca por tamanho do rotor
    inicio = time()
    chatbot.get_response('Qual o tamanho do rotor?')
    fim = time()
    TempoBusca += fim - inicio
    print(fim - inicio)
    #tempo de busca por frequência de troca
    inicio = time()
    chatbot.get_response('Qual a frequência da troca de óleo?')
    fim = time()
    TempoBusca += fim - inicio
    print(fim - inicio)

#Processo de checklist
print('Processo de checklist:')
for c in range(0, int(num/7)):
    #Início do processo
    inicio = time()
    chatbot.get_response('checklist')
    fim = time()
    TempoChecklist += fim - inicio
    print(fim - inicio)
    #Realização do resto do processo
    for c in range(0, 6):
        # Início do processo
        inicio = time()
        chatbot.get_response('Sim')
        fim = time()
        TempoChecklist += fim - inicio
        print(fim - inicio)

#processo de guia
print('processo de guia ')
for c in range(0, num):
    inicio = time()
    chatbot.get_response('o óleo está muito quente')
    fim = time()
    TempoGuia += fim - inicio
    print(fim - inicio)

MediaBusca = TempoBusca/num
MediaChecklist = TempoChecklist/num
MediaGuia = TempoGuia/num

print(MediaBusca)
print(MediaChecklist)
print(MediaGuia)

```

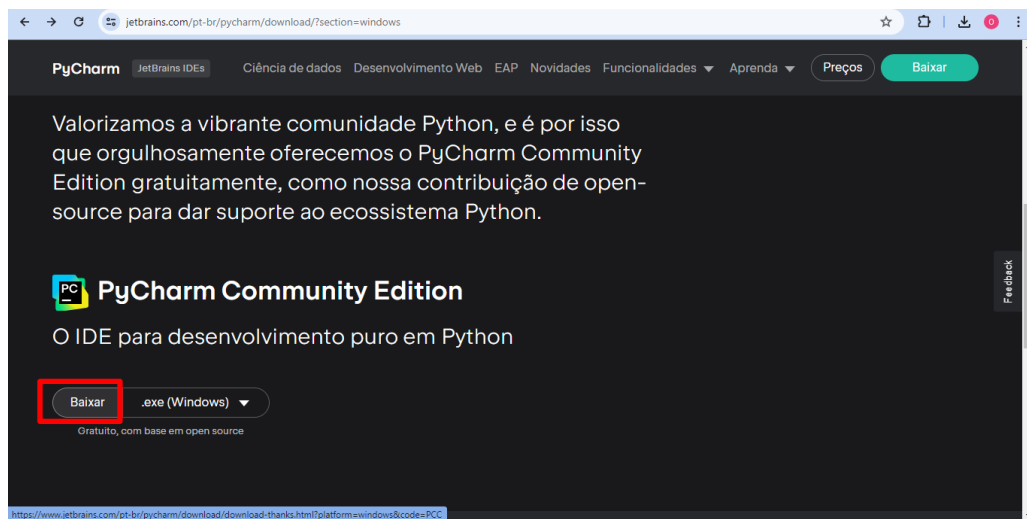
APÊNDICE V – GUIA DE INSTALAÇÃO E CONFIGURAÇÃO DO E-MANUEL E DAS FERRAMENTAS DE DESENVOLVIMENTO

Pycharm

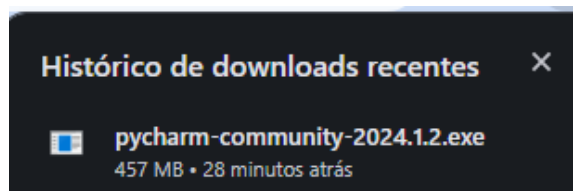
1. Entre no site:

<https://www.jetbrains.com/pt-br/pycharm/download/?section=windows>

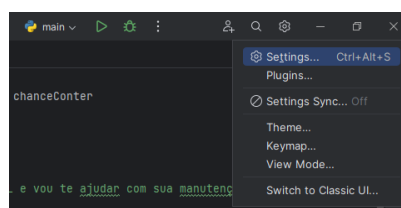
Vá até o final da página em “PyCharm Community Edition” e clique em baixar.



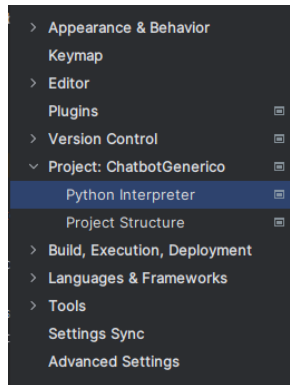
2. Ao terminar de baixar, execute o arquivo “pycharm-community-2024.1.2” ou instalador equivalente que tiver sido baixado.



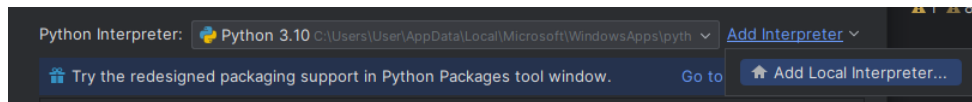
3. E siga o processo de instalação de acordo com o seu sistema operacional
4. Para que o PyCharm possa executar arquivos .py é necessário configurar o interpretador, isso é feito ao clicar em “Settings” no menu superior esquerdo



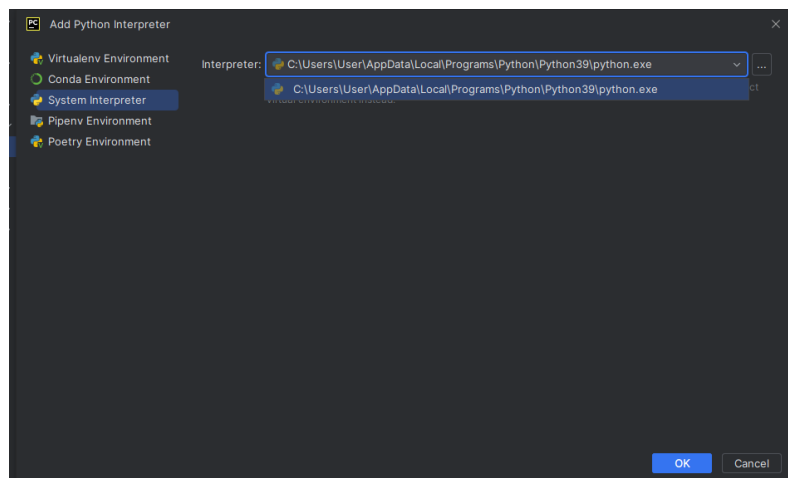
5. Isso abrirá uma nova janela de configuração, então em “Project” abra “Python Interpreter”.



6. Clique em “Add Local Interpreter” em “Add Interpreter”

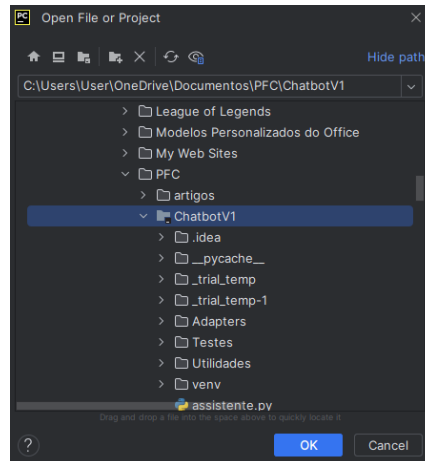


7. Selecione o interpretador e clique em “ok”. Caso não haja um interpretador instalado na máquina o PyCharm realizará a instalação automaticamente.



E-MANUEL

1. Para executar o E-MANUEL basta baixar o arquivo.rar, extrair na pasta de preferência, no PyCharm selecionar “Open Project”, selecionar a pasta do E-MANUEL e clicar em “Ok”



Pacotes Python

1. Para este trabalho os pacotes Python a ser instalados são:
 - a. chatterbot2
 - b. numpy
 - c. spacy
 - d. psycopg2-binary
 - e. pyyaml
 - f. chatterbot-corpus
2. Para realizar a instalação basta no PyCharm acessar “Python Packages” no menu esquerdo, pesquisar os pacotes que deseja instalar, selecionar a versão e clicar em “install”. Para realizar a instalação dos pacotes é recomendável a instalação da versão estável mais recente, evitando versões como “rc” ou “dev”.
3. Para que haja o devido funcionamento do chatterbot também será necessário que dentro do projeto seja executado em um arquivo python os comandos a baixo

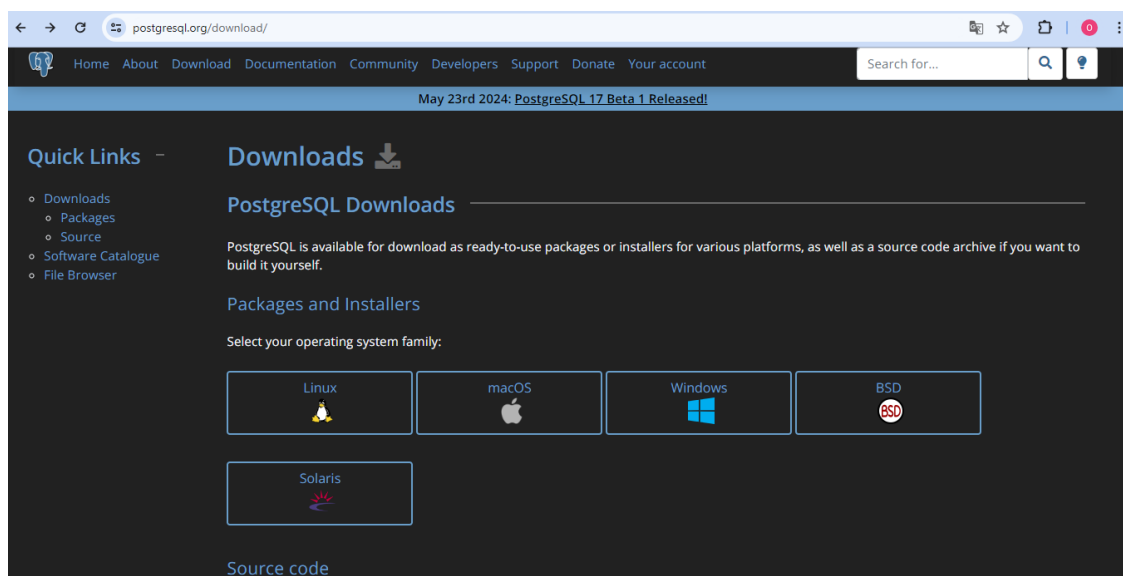
```
from spacy.cli import download

download("en_core_web_sm")

class ENGSM:
    ISO_639_1 = 'en_core_web_sm'
```

PostgreSQL

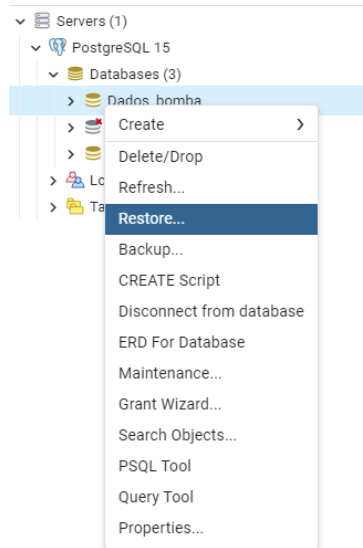
1. Acesse o site <https://www.postgresql.org/download/> e escolha o pacote de acordo com o sistema operacional correspondente sistema operacional e siga os passos indicados.



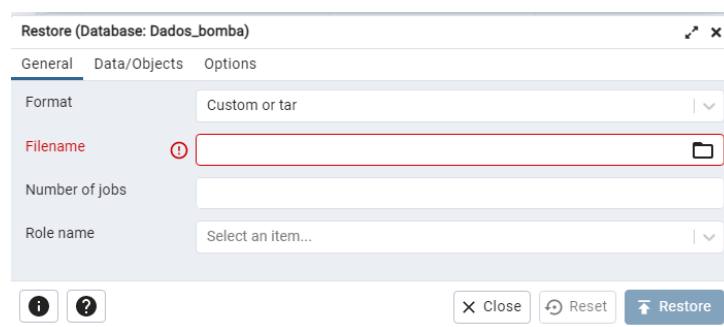
2. Baixe o instalador e execute-o, siga os passos de instalação de acordo com as preferências e seguindo em “next”, porém lembre de gravar a senha “master” que você definir.
3. Para gerenciar o banco de dados é utilizado o PgAdmin4 que por padrão é instalado dentro da pasta PostgreSQL

Importação do Banco de Dados

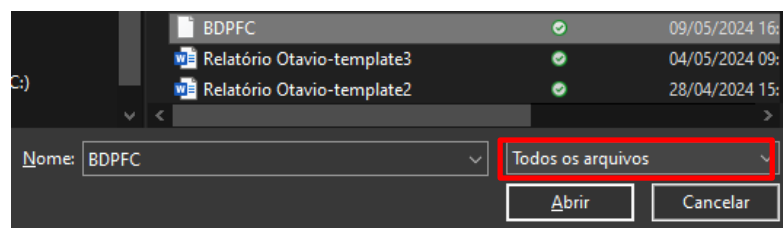
1. Para importar dados e tabelas em um banco de dados basta clicar com o botão direito do mouse no banco de dados de destino e em “Restore”



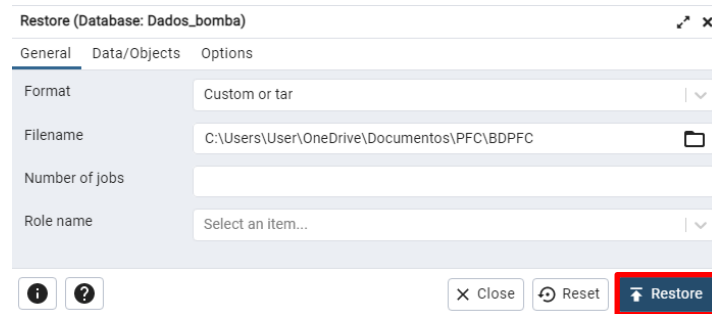
2. Mantenha o formato como “Custom or tar” e em “Filename” clique na pasta a esquerda



3. No explorador busque como “todos os arquivos” pelo arquivo de Backup referente ao projeto



4. Por fim clique em “Restore” e aguarde a finalização do processo



5. Já nos arquivos do E-MANUEL nas pastas assistente.py (linhas 10 a 14), BuscaDadosLumb.py (linhas 24 a 28) e BuscaDadosTec.py (linhas 24 a 28) no comando 'psycopg2.connect' altere os valores de 'password' e 'database' de acordo com as configurações usadas no PostgreSQL/PgAdmin4

```
con = psycopg2.connect(  
    user="postgres",  
    password="23071998",  
    database="Dados_bomba"  
)
```