Relatório A3 – Gestão e Qualidade de Software

Integrantes:

Gabriele Carraro de Oliveira RA: 125111352254

Isabela de Oliveira Ribeiro RA: 125111373163

Isabella Anselmo de Oliveira RA: 125111347710

**Introdução**

O objetivo deste relatório é apresentar e detalhar o processo de criação do sistema escolhido para desenvolvimento: Chatbot. Iremos abordar desde a introdução até a conclusão do projeto. Também serão abordados os estágios de desenvolvimento, os testes realizados antes da entrega do chatbot utilizando TDD (Test Driven Development), com os resultados obtidos em cada etapa do processo de execução.

O Chatbot desenvolvido executa as funções de responder determinadas perguntas relacionadas ao seu código de desenvolvimento, desta maneira, ele pode fornecer informações básicas de forma rápida e eficiente, economizando tempo e esforço dos usuários. Este é apenas um modelo base para ser apresentado aos clientes, após a apresentação, será feito um estudo de caso para quais funções de perguntas e respostas serão implementadas para o Chatbot executar.

Este modelo de Chatbot possui diversas maneiras para ser utilizado, como exemplo:

* Canal de suporte para o cliente, onde pode ajudar e direcionar os usuários a resolverem os seus problemas de maneira simples ou encaminhá-las ao nível de suporte responsável e apropriado para a solicitação;
* Agendamento de Reuniões, o que auxilia aos usuários para consultarem as suas reuniões de maneira ágil, onde também verifica a disponibilidade dos usuários para agendamento e auxilia com lembretes para entrada de reuniões;
* Consultas a informações Internas, como consultar informações dos usuários ao banco de dados, obter informações de departamentos, políticas internas, etc.

O Chatbot não somente irá ajudar nos exemplos citados acima, como também irá contribuir no dia a dia com diversas maneiras de otimizarmos o tempo e ganharmos agilidade nos processos.

**Desenvolvimento**

O desenvolvimento do chatbot foi dividido em diversas fases, incluindo a análise de requisitos, design, implementação e testes. Durante a fase de análise, foram identificados os requisitos funcionais e não funcionais do chatbot, onde os requisitos funcionais são responder às perguntas aleatórias, e informações precisas de data e hora atual, e os requisitos não funcionais estão atrelados ao seu desempenho, confiabilidade, e usabilidade, em que o chatbot deve corresponder respectivamente:

* Desempenho: Responder rapidamente às perguntas dos usuários;
* Confiabilidade: Ser confiável e estar disponível para responder perguntas de forma consistente, sem falhas frequentes ou indisponibilidades prolongadas;
* Usabilidade: O chatbot deve ser intuitivo e fácil de usar, fornecendo respostas claras e compreensíveis aos usuários, mesmo que as perguntas sejam aleatórias.

Com base na análise de requisitos, o design do chatbot foi elaborado, incluindo a definição das intenções (intents) e entidades (entities), a estrutura de diálogo e a escolha da plataforma de desenvolvimento, como exemplo para as entidades e intenções, o Chatbot desenvolvido responde às perguntas aleatórias, data e hora.

Para a implementação do chatbot, foi utilizada a linguagem de programação Java. As intenções e entidades foram configuradas, e o diálogo foi construído com base nas necessidades identificadas durante a fase de análise e também exibido todo o planejamento no Trello.

O chatbot oferece um conjunto de funções para atender às necessidades dos usuários de maneira eficiente, neste exemplo do Chatbot apresentado, ele foi desenvolvido com instruções base na fase de análise para apresentação aos usuários. O chatbot é capaz de fornecer informações precisas referente a data e hora atual, possui a função de exibir aos usuários o seu nome e responde também aos usuários algumas perguntas aleatórias.

Essas funcionalidades combinadas tornam o chatbot uma ferramenta versátil, e capaz de oferecer informações úteis para as necessidades do dia a dia. Ressaltando que este Chatbot é apenas um modelo base para ser apresentado ao cliente, onde apenas ele é incorporado ao discutirmos em reunião com o cliente as necessidades que precisam ser incorporadas ao sistema para atender às suas demandas no dia a dia.

TDD:

Abaixo iremos descrever os resultados dos testes unitários desenvolvidos utilizando a abordagem do TDD (Test Driven Development) para validar a funcionalidade de data, hora e perguntas aleatórias em nosso sistema. O objetivo desses testes é garantir que as implementações estejam corretas e atendam aos requisitos esperados.

A metodologia TDD foi implementada nesta abordagem, pois consiste em escrever os testes antes da implementação do código. Assim, foi possível garantir que os requisitos de data, hora e perguntas aleatórias fossem satisfeitos, e também nos auxiliar para detectar se ocorreu possíveis problemas no desenvolvimento do chatbot.

Para a criação dos testes utilizando TDD (Test Driven Development), criamos uma classe teste em Java, onde o Chatbot irá validar os pontos abaixo:

* Data – Testes na chamada do Chatbot para verificar se ele irá retornar ao que foi perguntado corretamente. Verificamos se a data atual retornada pelo sistema está correta e se é do tipo esperado.
* Hora - Testes na chamada do Chatbot para verificar se ele irá retornar ao que foi perguntado corretamente. Também verificamos se a hora atual retornada pelo sistema está correta e se está no formato esperado.
* Perguntas aleatórias - Testes na chamada do Chatbot para verificar se ele irá retornar ao que foi perguntado corretamente. Avaliamos se as perguntas estão no formato esperado e se atendem aos requisitos de aleatoriedade.
* Testes onde o usuário irá fazer uma pergunta, e o Chatbot não foi programado para responder, e então será retornado uma mensagem de erro.

Execução Chatbot:

Para acessar e executar o Chatbot desenvolvido é necessário certificar-se ter o Java Development Kit (JDK) instalado no seu computador. Caso não possua o JDK especificado, é possível realizar o download a partir do site oficial da Oracle, e para este desenvolvimento do Chatbot foi utilizado o JDK na versão 17. Abra a sua IDE Java de preferência, como o IntelliJ IDEA, Eclipse ou NetBeans e para testar e executar o chatbot, é possível criar uma classe principal (ou método principal em uma das classes existentes) e adicionar código para simular a interação com o usuário. Para os testes, foi utilizado o JUnit na versão 4.9. Onde então, criamos uma classe com a anotação “@teste” para definirmos os métodos de testes para validar a funcionalidade do Chatbot desenvolvido.

**Conclusão:**

O chatbot foi desenvolvido utilizando a linguagem Java para responder às perguntas aleatórias, fornecer informações referente a data e hora, onde foi implementado com sucesso e testado as suas funcionalidades com a equipe de desenvolvimento. Através da validação e desenvolvimento da equipe, o chatbot é capaz de fornecer respostas precisas e atualizadas aos usuários de acordo com o que deseja. Desta maneira, a implementação dessas funcionalidades melhora a experiência do usuário e proporciona interações mais dinâmicas.

Os testes unitários realizados que verificam a funcionalidade de data, hora e pergunta aleatória do chatbot foram bem-sucedidos atribuindo a abordagem TDD. Todas as implementações passaram nos testes, comprovando que a funcionalidade está correta e atende aos requisitos definidos. E todos os testes realizados na abordagem TDD, estão registrados na classe teste.

E com isso, podemos concluir que com base na abordagem TDD, nos resultou para aumentar a confiança na qualidade do código desenvolvido e nos proporcionou também uma redução no risco de bugs e falhas do chatbot.