

Trabalho de Java

*Felipe Bernardes
Felipe Pinheiro
Gabriel Girami
Gustavo Vinhola
Victor Fanfoni
Rafael Chaves*

Sumário

- *Página 3: Destinada à descrição do objetivo como grupo*
- *Link GitHub: [GFV CORP JAVA](#)*

Descrição do objetivo

O nosso objetivo como grupo é criar um registro prévio com informações do cliente (seu cadastro, CPF, telefone) e seu carro utilizando atributos (placa, peso, tamanho, CNH) com os tipos de modais atrelados a uma entidade no sistema de banco de dados, pronto para pegar as informações a qualquer momento, com o intuito de agilizar o atendimento, com um método que verifica se cada informação adequada ao tipo de modal pesado e calculando, garantindo uma assertividade melhor para o tipo de modal em determinada situação, já que as informações do carro já vão estar no sistema, basta acessá-las. A inteligência artificial irá fazer a conversação com o cliente, pedindo uma espécie de cadastro, a fim de localizar todas as especificações do carro, e criaremos um programa na linguagem java que realize essa transmissão de informações da conversa até o banco de dados.

Na classe de decisão de modal, estará o tipo de carroceria do veículo do cliente, a distância que será calculada em outro sistema, o peso do veículo com a carga, possíveis modificações que o usuário fez, e o modal guardando a decisão final.

Para deixar o processo mais transparente, criaremos duas classes para informações do veículo da Porto e seu funcionário, guardando atributos como a placa do veículo de seguros, o nome do funcionário e sua identificação e a outra classe instanciando objetos, evitando também possíveis golpes e garantindo mais segurança ao cliente.

Além disso, vamos implementar uma sistemática de rastreamento em tempo real, que vai aparecer na tela do cliente a partir do momento que o sistema recebeu o tipo de modal adequado para a situação, e irá funcionar parecido com o sistema que o Uber usa para as viagens, para que ele saiba quando o guincho está chegando, ajudando na satisfação do mesmo (Client Experience).

Por fim, teremos uma classe reservada para a geração de protocolo, e outra para envio de todas as informações que foram armazenadas e puxadas pelas outras classes do projeto.

