# **Documentação — Chatbot de Especificações de Carros**

## **1. Visão Geral**

Este chatbot é uma aplicação web interativa onde usuários podem consultar especificações técnicas de veículos ou pedir recomendações de carros conforme critérios (ex: "carros econômicos até 50 mil"). O sistema utiliza inteligência artificial (Google Gemini) para gerar respostas objetivas, sucintas e contextuais, além de sugerir links para compra em grandes portais (OLX, Webmotors, Mercado Livre).

## **2. Funcionalidades**

* Consulta livre: Pergunte sobre qualquer modelo ou categoria (ex: “SUV automático bom e barato”, “Onix 2019 ficha técnica”).
* Respostas organizadas em tópicos, objetivas e sem “encheção de linguiça”.
* Recomendações de veículos baseadas em critérios informados.
* Links diretos para busca do veículo na OLX, Webmotors e Mercado Livre.
* Histórico de chat mantido (o chatbot lembra informações já fornecidas durante a conversa).
* Interface responsiva, moderna e fácil de usar.

## **3. Arquitetura e Arquivos**

O projeto é composto por três partes principais:

### **3.1. Frontend**

* **index.html**: Estrutura da interface do chat.
* **style.css**: Aparência e responsividade.
* **script.js**: Lógica de interação do usuário, histórico e integração com backend.

### **3.2. Backend**

* **server.js**: Servidor Express que integra com a API do Gemini para gerar as respostas inteligentes.

### **3.3. Outros**

* **.env**: Arquivo de configuração (não incluso no repositório por segurança), onde você coloca sua chave GEMINI\_API\_KEY.

## **4. Instalação**

### **Pré-requisitos**

* Node.js instalado (versão 16 ou superior recomendada).
* Conta na Google Cloud para obter a chave da API Gemini.
* (Opcional) Editor de código (ex: VS Code).

### **Passos:**

1. **Clone o projeto ou copie os arquivos para sua máquina.**

**Instale as dependências no backend pelo prompt de comando:** npm install express cors dotenv @google/generative-ai

**Crie o arquivo .env na raiz do projeto backend e adicione sua chave:** GEMINI\_API\_KEY=sua\_chave\_api\_gemini\_aqui

**Execute o servidor backend:** node server.js

1. Por padrão, ele roda na porta 3000.
2. **Abra o arquivo index.html no navegador.**

## **5. Estrutura dos Arquivos Principais**

### **index.html**

* Contém o título, área do chat e campo de input.
* Responsável apenas pela estrutura visual.

### **style.css**

* Define o layout, cores, fontes, responsividade e efeitos do chat.
* Inclui animação suave nas mensagens, destaque para links e adaptações para mobile.

### **script.js**

* Controla o histórico do chat, envia perguntas e histórico para o backend, formata e exibe as respostas (inclusive links).
* Garante usabilidade fluida: Enter envia, exibe carregando, bloqueia botão durante processamento etc.

### **server.js**

* Recebe mensagens e histórico via POST.
* Monta o prompt de forma inteligente, pedindo respostas curtas, objetivas e sempre com links de busca do veículo informado.
* Retorna ao frontend a resposta da IA para ser apresentada ao usuário.

## **6. Como funciona o fluxo**

1. Usuário envia uma pergunta livre no chat (ex: "Carros econômicos até 50 mil").
2. Frontend registra a pergunta e envia ao backend junto com o histórico do chat.
3. Backend monta o prompt, insere o histórico para o Gemini, solicita uma resposta.
4. Gemini retorna resposta objetiva, organizada e com links de busca.
5. Frontend exibe a resposta formatada, incluindo os links clicáveis para OLX, Webmotors e Mercado Livre.
6. Usuário pode continuar perguntando, aproveitando o contexto das mensagens anteriores.

## **7. Customizações Possíveis**

* Adicionar suporte para imagens: Exibir imagens dos carros sugeridos.
* Persistir histórico do usuário: Usar localStorage ou banco de dados.
* Personalizar prompt do Gemini: Para ajustar tom, nível de detalhe ou foco da resposta.
* Internacionalização: Adaptar para outros idiomas.
* Adição de outros marketplaces: Fácil incluir outros links de busca de veículos.

## **8. Exemplo de Uso**

**Pergunta:** Bons carros automáticos até 60 mil reais

**Resposta esperada:**

RECOMENDAÇÕES DE CARROS AUTOMÁTICOS ATÉ 60 MIL

• Honda Fit 2016 — Econômico e confiável

• Toyota Corolla 2014 — Confortável e fácil de revenda

• Hyundai HB20S 2018 — Bom custo-benefício

Links para compra:

- [OLX](https://www.olx.com.br/autos-e-pecas?q=Honda+Fit+2016)

- [Webmotors](https://www.webmotors.com.br/carros/estoque?q=Honda+Fit+2016)

- [Mercado Livre](https://lista.mercadolivre.com.br/Honda-Fit-2016)

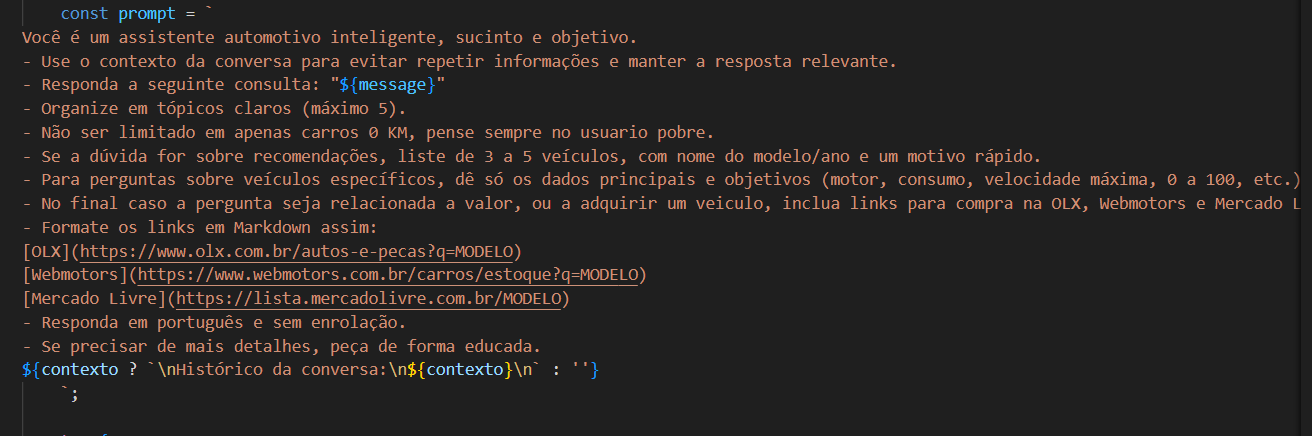
## **9. Dicas e Melhores Práticas**

* Use perguntas claras e específicas para respostas mais relevantes.
* Recarregue a página para iniciar um novo histórico de conversa.
* Sempre verifique os links gerados antes de compartilhar com terceiros.
* O sistema depende da API Gemini — possíveis limites de uso e custos.

## **10. Exemplo de Prompt Customizado (no Backend)**

O prompt enviado para Gemini é construído para garantir:

* Respostas diretas, tópicos claros e nunca prolixo
* Uso de contexto/histórico
* Geração automática dos links de compra



## **11. Suporte e Dúvidas**

* Para dúvidas sobre instalação ou funcionamento, verifique se a chave da API Gemini está ativa.
* Para ajustes de visual ou comportamento, edite os arquivos style.css ou script.js.
* Para modificar a inteligência da resposta, altere o prompt no server.js.
* Para outros problemas, consulte os logs do console no navegador ou terminal.