

Fiz o projeto utilizando a IA do ChatGPT:

1. Fiz o pedido usando a mensagem “construa um sistema de autoatendimento, no qual o cliente interagirá com um terminal de autoatendimento dotado das seguintes funcionalidades: 1. Identificação Opcional: O cliente pode: • Identificar-se via CPF • Cadastrar-se informando nome e e-mail • Permanecer anônimo 2. Seleção de Produtos por Categorias: As categorias são fixas: • Lanche • Acompanhamento • Bebida • Sobremesa Em cada etapa, serão exibidos nome, descrição e preço de cada produto. O cliente pode montar o combo livremente, escolhendo ou não itens de cada categoria. 3. Pagamento Integrado: Após a seleção dos itens, o cliente realizará o pagamento via QR Code do Mercado Pago. Caso a integração real não seja implementada, um mock deve simular essa etapa. 4. Acompanhamento do Pedido: Assim que o pagamento for confirmado, o pedido entrará em um fluxo de status: (a) Recebido (b) Em Preparação (c) Pronto (d) Finalizado O cliente poderá acompanhar o andamento do pedido em um monitor. Quando o pedido estiver "Pronto", será notificado para retirada. Ao ser retirado, o status muda para "Finalizado".”

A IA retornou uma `sistema_autoatendimento.py` inicial.

E um comentário “O sistema de autoatendimento foi desenvolvido com base na descrição fornecida. Ele inclui funcionalidades para identificação de clientes, seleção de produtos, pagamento simulado via QR Code, e acompanhamento do status do pedido. Caso precise de ajustes ou queira expandir alguma funcionalidade, é só me informar!”

2. Depois fiz o pedido “agora adicione uma funcionalidade para que tenha outro terminal no qual, a cozinha receberá os pedidos conforme são confirmados, podendo atualizá-los de "Recebido" para "Em Preparação", depois para "Pronto" e, por fim, "Finalizado" quando o cliente retirá-lo. Esse controle assegura maior eficiência e previsibilidade no preparo”

Que editou a `sistema_autoatendimento.py` inicial adicionando as edições.

E comentou “Adicionei um terminal para a cozinha receber pedidos e atualizá-los conforme avançam no processo de preparo. Agora, há uma rota (`/cozinha`) que retorna os pedidos pendentes, facilitando a visualização e gerenciamento. Se precisar de mais ajustes, me avise!”

3. Depois pedi a terceira edição “agora coloque um Painel Administrativo do Estabelecimento

O estabelecimento terá acesso a um painel para:

1. Gerenciar Clientes: Utilizar dados de identificação para campanhas e promoções.
2. Gerenciar Produtos e Categorias: Adicionar, editar e remover produtos (nome, descrição, imagem, categoria, preço).
3. Acompanhamento de Pedidos: Visualizar pedidos em andamento, atualizando status e observando tempos de espera.”

Na qual a IA editou, e comentou “Adicionei um Painel Administrativo com funcionalidades para gerenciar clientes, produtos e acompanhar pedidos. Agora, o estabelecimento pode visualizar e administrar essas informações diretamente pelo sistema. Se precisar de mais melhorias, me avise!”

4. Pedi “agora crie um Dockerfile para executar a aplicação acima”

A IA respondeu “Aqui está um Dockerfile para executar a aplicação Flask dentro de um contêiner Docker:

Agora, basta criar um arquivo requirements.txt com as dependências do Flask e rodar os comandos:

(docker build -t sistema_autoatendimento .

docker run -p 5000:5000 sistema_autoatendimento)”

5. Pedi a mensagem “agora crie um docker-compose.yml para subir todo o ambiente”

Resposta “Criei um docker-compose.yml que sobe o sistema de autoatendimento junto com um banco de dados PostgreSQL.”

E criou um [Docker-compose.yml](#)

6. Pedi para criar os testes das APIs usando o prompt “crie exemplos de requisições para testar as APIs do Sistema Autoatendimento”
7. Pedi para adaptar para Postman

Resposta “Adicionei exemplos de requisições para testar as APIs do sistema de autoatendimento. Se precisar de mais testes ou ajustes, me avise! 🚀” E criou um arquivo com os testes que está no repositório e depois atualizou com a seguinte mensagem “Adapte os exemplos de requisições para o formato compatível com o Postman. Agora você pode importar esse arquivo JSON diretamente no Postman e testar as APIs. Se precisar de ajustes, me avise! 🚀”

8. Pedi para criar as instruções de execução usando o prompt “agora crie Instruções completas para rodar o projeto localmente, incluindo a ordem de chamadas das APIs, se necessário”

Resposta “Criei um documento com as instruções completas para rodar o projeto localmente, incluindo a ordem recomendada para chamadas das APIs. Se precisar de ajustes, me avise! 🚀”