

Nome do projeto: central de atendimento da uby Autor(a): goimy rondinelli da silveira E-mail: goimyrondinellis@gmail.com

CANVAS			
<div><div>Problema</div><div>Descreva o problema do negócio</div><div>Muitos clientes não têm familiaridade com o aplicativo</div><div>Clientes preferem testar o serviço primeiro pelo WhatsApp</div><div>O processo atual é 100% manual</div><div>Atendentes precisam:</div><div>Ler mensagens no WhatsApp</div><div>Identificar endereço de origem</div><div>Identificar endereço de destino</div><div>Cadastrar a corrida no aplicativo</div><div>Isso gera:</div><div>Demora no atendimento</div><div>Risco de erro humano</div><div>Custo alto com equipe</div><div>Dificuldade de escalar o atendimento</div></div>	<div><div>Como resolver SEM AI</div><div>Descreva o processo atual e solução</div><div>Fluxo operacional: Canal: WhatsApp Business (número oficial). Atendente humano recebe mensagem, segue script padrão: Pergunta 1: “Onde você está? (endereço ou enviar localização)” Pergunta 2: “Para onde deseja ir?” Criar corrida via API do app. Informar motorista ao cliente. Padrões: máximo 2 minutos SLA de resposta inicial , máximo 5 minutos para confirmação . Registro: cada atendimento logado (mensagens, tempo, resultado). Recursos necessários (MVP sem IA): Equipe de atendimento: 12 atendentes</div></div>	<div><div>Como resolver COM AI</div><div>Descreva a solução AI by design</div><div>Principais decisões de design: Não permitir que LLM “decida” dispatch crítico sem validação e de endereço. Human-in-the-loop: bot resolve fluxos simples; atendente recebe histórico e assume exceções.</div><div>Fluxo técnico (MVP recomendado): Cliente envia mensagem no WhatsApp → provider (Twilio / 360dialog / Zenvia). Bot low-code responde com perguntas estruturadas (origem, destino). Usa NLU leve (regex + modelo leve/LLM para entropia). Bot chama APP para validar endereço Se endereço validado: APP criar corrida e obter motorista/ETA. Bot informa nome do motorista, ETA, modelo do carro e número de placa parcial. Se não encontrar motorista, não encontrar endereço ou tiver qualquer dúvida do que está fazendo (nível de confiança inferior a 70%) chama o atendente.</div></div>	<div><div>Para quem?</div><div>Descreva as personas</div><div>Persona primária (alvo do MVP): "João" — 67 anos, mora em cidade pequena, pouco uso de smartphone, liga para a central ou manda mensagem pedindo corrida. Objetivo: pedir corrida sem instalar/usar o app.</div><div>Personas secundárias: Novos usuários do app (sem familiaridade) porque na maioria das vezes nunca usaram um transporte por aplicativo, depois de se acostumarem a usar vão chamar no app porque é mais rápido e mais barato.</div><div>Atendentes da central (usuários do painel de handoff). Gestores de operação (monitoram KPIs). Motoristas (afetados indiretamente: recepção de corridas mais corretas). Ações recomendadas: Realizar 10 entrevistas com João-type por cidade piloto para ajustar linguagem. Testes de usabilidade do fluxo WhatsApp (observacional). Está de acordo com esse público? Quer segmentar por faixa etária ou cidade específica?</div></div>
<div><div>Dados</div><div>Liste dados e fontes</div></div>	<div><div>Ferramentas</div><div>Descreva as ferramentas</div></div>		<div><div>Indicadores de sucesso</div><div>Descreva o(s) indicador(es)</div></div>

Fontes de dados necessárias:

Mensagens do WhatsApp (texto, mídia, áudio, timestamps).

API do app: validação de endereço / busca de motorista / criação de corrida / ETA.

Logs da central (atendimentos por operador, tempo médio).

Dados de motoristas disponíveis (localização, status).

Provedores WhatsApp Business:

360dialog / Twilio / Zenvia — escolha depende de custo, SLA e integração local.

Flow builder / Chatbot:

Typebot / Landbot (low-code, rápido para MVP) — prós: velocidade; contras: flexibilidade para NLU avançada.

Botpress / Rasa (mais técnico) — prós: controle; contras: tempo de

Indicadores propostos:

Tempo médio do primeiro contato até criação da corrida (Tempo de pedido → dispatch)

Baseline: 5 minutos

Meta MVP: reduzir em 40%

Erros operacionais por 100 corridas (ex.: corrida com endereço inválido,