

PlanejaBot: Catálogo de Tarefas para Design e Backlog

Preâmbulo: Da Visão Estratégica ao Backlog Acionável

Este documento traduz o plano estratégico e o roadmap de desenvolvimento do PlanejaBot, detalhados nos materiais de referência ¹, em um catálogo exaustivo de tarefas acionáveis, prontas para serem organizadas em um quadro Kanban como o Trello. O objetivo é fornecer uma fonte única e clara de todo o trabalho a ser feito, sem exceções, para guiar a equipe desde a criação do Produto Mínimo Viável (MVP) até a evolução de longo prazo da plataforma.

As tarefas foram meticulosamente separadas em duas categorias principais, conforme solicitado:

1. **DESIGN:** Engloba todas as atividades relacionadas à experiência do usuário, design conversacional (CUX), identidade visual, comunicação e criação de conteúdo. Em um produto como um chatbot, esta categoria é fundamental, pois define a personalidade do bot, o fluxo dos diálogos e a clareza da interação, sendo o "prompt engineering" uma atividade de design central.¹
2. **BACKLOG:** Contém todas as tarefas técnicas de desenvolvimento, incluindo a implementação de funcionalidades no backend, configuração de infraestrutura, integrações com APIs, gerenciamento de banco de dados e deploy.

Cada tarefa é apresentada com um identificador único, uma descrição detalhada do que precisa ser feito, a fase do projeto a que pertence para fins de priorização, e notas sobre dependências, garantindo que o fluxo de trabalho possa ser sequenciado de forma lógica e eficiente.

Catálogo de Tarefas de DESIGN

Esta seção detalha todas as tarefas criativas e focadas na experiência do usuário. O trabalho aqui define "como" o usuário interage com o PlanejaBot e percebe sua marca.

ID da Tarefa	Nome da Tarefa	Descrição Detalhada (O que precisa ser feito)	Fase do Projeto	Dependências / Notas
D-MVP-01	Definição da Identidade Conversacional e Tom de Voz	Com base nos arquétipos de marca definidos (Sábio, Cuidador e um toque de Mago) ¹ , criar um guia de estilo de conversação. Este documento deve detalhar o tom de voz (Amigável, Empático, Motivador, Inteligente), exemplos de frases, palavras a serem usadas e evitadas, e como o bot deve se comunicar de forma consistente. Este guia será a base para toda a redação de	MVP	Nenhuma. Esta é uma tarefa fundamental que informa muitas outras.

		prompts e mensagens, garantindo que a personalidade e do bot seja coerente em todas as interações. ¹		
D-MVP-02	Design do Fluxo de Conversa para Onboarding e Consentimento LGPD	<p>Mapear visualmente (usando uma ferramenta como Miro ou FigJam) e redigir o texto exato para o fluxo de onboarding do primeiro usuário. O fluxo deve começar com a mensagem de boas-vindas: "Olá! Eu sou o PlanejaBot..."¹, explicar o propósito do bot, solicitar o consentimento explícito para tratamento de dados, fornecer o link para a Política de Privacidade e definir as</p>	MVP	Depende de: D-MVP-01, D-AUX-03.

		respostas para as interações do usuário ("Sim, concordo", "Ler Política de Privacidade", "Não, obrigado(a)"). O design deve ser claro, acolhedor e em total conformidade com a LGPD. ¹			
D-MVP-03	Design do Fluxo de Conversa para o Ciclo Central de Tarefas	<p>Mapear e redigir os diálogos completos para as três interações principais do MVP: 1)</p> <p>Criação de Tarefa: desde o input do usuário em linguagem natural até a confirmação do bot e a pergunta proativa ("Quer adicionar prioridade ou um lembrete?").¹</p> <p>2)</p>	<p>Listagem de Tarefas: para comandos como "Minhas tarefas de hoje". 3)</p> <p>Conclusão de Tarefa: incluindo a mensagem de reforço positivo ("Parabéns! Tarefa '...' marcada como concluída.").¹</p> <p>O design deve focar na clareza e eficiência.</p>	MVP	Depende de: D-MVP-01.

D-MVP-04	Redação das Mensagens de Ajuda e Tratamento de Erros	Redigir a mensagem concisa que o usuário receberá ao digitar "Ajuda", listando as funcionalidades principais do MVP. ¹ Além disso, criar um conjunto de mensagens para tratamento gracioso de erros, como quando o bot não entende a solicitação ("Desculpe, não entendi bem. Você poderia reformular?") para guiar o usuário sem causar frustração. ¹	MVP	Depende de: D-MVP-01.
D-AUX-01	Roteirização e Storyboard do Vídeo de Apresentação (5 min)	Criar um roteiro detalhado, segundo a segundo, para o vídeo de 5 minutos exigido pelo Desafio Liga Jovem. O roteiro deve seguir a	Ancilar (MVP)	Depende de: Todas as tarefas de Design e Backlog do MVP. Esta é uma tarefa de consolidação.

		<p>estrutura de pitch (Problema, Solução, Demo, Mercado, etc.)¹ e ser projetado para cobrir todos os critérios de avaliação do desafio. Criar um storyboard visual para planejar as cenas, especialmente a demonstração do MVP em ação, incluindo a simulação de funcionalidades como os lembretes proativos, se necessário.¹</p>		
D-AUX-02	Design e Conteúdo da Landing Page de Marketing	<p>Projetar o layout (wireframe e design visual) e redigir todo o texto para uma landing page simples (one-page). O design deve ser limpo, responsivo (mobile-first)</p>	Ancilar (MVP)	Depende de: D-MVP-01.

		<p>e alinhado à identidade da marca. O conteúdo deve incluir uma headline clara, descrição do problema/solução, e um Call-to-Action (CTA) proeminente com o link wa.me que inicia diretamente a conversa com o PlanejaBot.¹</p>		
D-AUX-03	<p>Redação da Política de Privacidade e Termos de Uso (Versão Inicial)</p>	<p>Elaborar o texto para a Política de Privacidade e os Termos de Uso, em conformidade e com a LGPD. A linguagem deve ser clara, acessível e transparente, detalhando quais dados são coletados (conteúdo das mensagens, número de telefone), para qual</p>	<p>Ancilar (MVP)</p>	<p>Nenhuma. É um pré-requisito para o D-MVP-02.</p>

		finalidade, como são compartilhados com operadores de dados (ex: Google para API Gemini) e os direitos do usuário. Este é um trabalho de design de comunicação legal. ¹		
D-ST-01	Design das Interações de Gamificação (Pontos e Streaks)	Definir as regras e redigir as mensagens para o sistema de gamificação. Isso inclui: quantos pontos o usuário ganha por tarefa concluída, como um "streak" (sequência) de uso diário é calculado e comunicado, e quais são as mensagens de celebração para quando um usuário atinge um	Curto Prazo	Depende de: D-MVP-01.

		<p>novo streak ou um marco de pontos. O design deve ser motivador e sutil, não intrusivo.¹</p>		
D-ST-02	Design dos Resumos de Produtividade e Personalizados	<p>Projetar o formato e redigir os templates para os resumos semanais de produtividade e. Definir como os dados (ex: número de tarefas concluídas, percentual de avanço em metas) serão apresentados de forma concisa, visualmente agradável (usando emojis) e inspiradora. Exemplo: "Esta semana você concluiu X tarefas e avançou Y% na sua meta Z".¹</p>	Curto Prazo	Depende de: D-MVP-01.
D-ST-03	Design do Fluxo de	Mapear todo o fluxo de	Curto Prazo	Depende de: D-MVP-01.

	<p>Conversa para Integração com Google Calendar</p>	<p>diálogo para a integração com o Google Calendar. Isso inclui o processo de autenticação OAuth 2.0 (como o bot solicita permissão de forma clara e segura), o design dos comandos conversacionais para visualizar eventos do calendário e para criar novos eventos a partir de tarefas existentes no PlanejaBot.¹</p>		
D-MT-01	<p>Design da Funcionalidade de "Parceiro de Responsabilidade"</p>	<p>Projetar todo o fluxo de interação para a funcionalidade de "Accountability Partner". Isso inclui: como um usuário convida outro, como o consentimento para</p>	Médio Prazo	<p>Depende de: D-MVP-01.</p>

		<p>compartilhamento de progresso de metas específicas é obtido de forma explícita e granular de ambos os usuários, e como as atualizações de progresso são comunicadas entre os parceiros de forma motivadora. A privacidade e a clareza são primordiais.¹</p>		
D-MT-02	UX/UI Design do Dashboard Web Simples	<p>Criar os wireframes, protótipos visuais (mockups) e o design final da interface do usuário para o dashboard web. O dashboard deve apresentar estatísticas de produtividade e, progresso de metas e</p>	Médio Prazo	Depende de: D-MVP-01.

		tarefas de forma gráfica (gráficos, barras de progresso) e intuitiva, complementando a experiência do chatbot e oferecendo uma visão mais detalhada dos dados do usuário. ¹		
D-MT-03	Design da Comunicação para o Modelo Freemium	Redigir as mensagens e projetar os pontos de contato (triggers) dentro da experiência do chatbot e do dashboard web que comunicarão a existência e os benefícios da "Versão Premium". O objetivo é explicar o valor dos recursos pagos (ex: IA avançada, integrações) de forma clara e convidativa,	Médio Prazo	Nenhuma.

		em momentos contextuais apropriados, sem prejudicar a experiência do usuário gratuito. ¹		
--	--	---	--	--

Catálogo de Tarefas de BACKLOG

Esta seção detalha todas as tarefas técnicas necessárias para construir, implantar e escalar o PlanejaBot.

ID da Tarefa	Nome da Tarefa	Descrição Detalhada (O que precisa ser feito)	Fase do Projeto	Dependências / Notas
B-MVP-01	Configuração do Ambiente de Desenvolvimento e Repositório Git	Inicializar o repositório Git no GitHub/GitLab. Estruturar o projeto Flask. Configurar o ambiente virtual (venv) e instalar as dependências iniciais (Flask, python-dotenv, requests, gspread, google-generativeai, pandas) em um arquivo requirements.txt . Criar e configurar o	MVP	Nenhuma. Tarefa inicial.

		arquivo .gitignore para excluir arquivos sensíveis e de ambiente (como .env, __pycache__, venv/). ¹		
B-MVP-02	Implementação do Servidor Flask Básico e Endpoint de Webhook	Criar a aplicação Flask principal (app.py). Implementar o endpoint /webhook/whatsapp que lida com as requisições GET (para verificação inicial do webhook da Meta, validando o hub.verify_token) e POST (para receber os payloads JSON com as mensagens do usuário). ¹	MVP	Depende de: B-MVP-01.
B-MVP-03	Integração com a API do WhatsApp (Envio e Recebimento)	Desenvolver as funções para: 1) Parsear o payload JSON recebido no webhook para extrair o número de telefone do remetente (from), o texto da mensagem (body) e o ID da mensagem. 2)	MVP	Depende de: B-MVP-02.

		<p>Construir e enviar requisições POST para a API Graph da Meta (/messages) para responder ao usuário. As credenciais (Access Token, Phone Number ID) devem ser gerenciadas via variáveis de ambiente.¹</p>		
B-MVP-04	<p>Integração com a API Gemini para Processamento de Linguagem Natural</p>	<p>Desenvolver um módulo de serviço (gemini_service.py) para interagir com a API Gemini. A função principal receberá o texto do usuário (e, opcionalmente, o histórico da conversa para contexto), enviará para o modelo Gemini (ex: gemini-1.5-flash-latest) com um prompt estruturado (baseado no design de D-MVP-01), e processará a resposta. A chave de API da Gemini deve ser gerenciada via variável de</p>	MVP	<p>Depende de: B-MVP-01, D-MVP-01.</p>

		ambiente. ¹		
B-MVP-05	Implementação do Backend para o Ciclo de Tarefas (Usando Google Sheets)	Desenvolver a lógica no backend que, após receber a intenção da Gemini, interage com o Google Sheets usando a biblioteca gspread. As funções devem ser capazes de: 1) Adicionar uma nova linha para uma nova tarefa. 2) Ler linhas para listar tarefas por período. 3) Atualizar a coluna "Status" para concluir uma tarefa. Esta abordagem representa uma dívida técnica consciente para acelerar o desenvolvimento do MVP. ¹	MVP	Depende de: B-MVP-04, D-MVP-03.
B-MVP-06	Implementação do Backend para Definição Simples de Metas (Google Sheets)	Desenvolver a lógica para salvar uma nova meta em uma aba específica ("Metas") da planilha do usuário no Google Sheets, de forma similar à funcionalidade de tarefas (B-MVP-05). A	MVP	Depende de: B-MVP-04.

		função deve registrar a descrição e o prazo da meta. ¹		
B-AUX-01	Configuração da Infraestrutura de Implantação na Nuvem (Render.com)	<p>Criar uma conta no Render.com. Conectar o repositório Git. Configurar um novo "Web Service" para a aplicação Python/Flask. Definir o comando de build (pip install -r requirements.txt) e o comando de start (usando gunicorn app:app). Configurar todas as variáveis de ambiente (chaves de API, tokens) no dashboard do Render. A URL pública gerada é essencial para o funcionamento do webhook do WhatsApp.¹</p>	Ancilar (MVP)	Depende de: B-MVP-02. É uma tarefa crítica para a demonstração.
B-AUX-02	Produção e Edição do Vídeo de Apresentação (5 min)	Gravar as cenas planejadas no storyboard (D-AUX-01), incluindo a narração e a demonstração do protótipo	Ancilar (MVP)	Depende de: D-AUX-01, B-AUX-01, e todo o MVP funcional.

		funcional rodando na nuvem. Editar o material, adicionar legendas, trilha sonora e gráficos para criar um vídeo profissional e impactante de até 5 minutos. Fazer o upload para o YouTube e submeter o link na plataforma do Desafio. ¹		
B-AUX-03	Desenvolvimento e Deploy da Landing Page	Usando uma plataforma como Carrd ou desenvolvendo em HTML/CSS simples, construir a landing page projetada em D-AUX-02. Garantir que o link wa.me esteja funcionando corretamente e que a página seja responsiva. Fazer o deploy da página em um serviço de hospedagem estática. ¹	Ancilar (MVP)	Depende de: D-AUX-02.
B-ST-01	Migração do Banco de Dados de Google Sheets para	Tarefa de alta complexidade para pagar a dívida técnica	Curto Prazo	Depende de: Todas as funcionalidades do MVP que

	Firestore	do MVP. Inclui: 1) Projetar a estrutura de coleções e documentos no Firestore (ex: coleção users, subcoleções tasks e goals). 2) Escrever um script de migração único para mover os dados existentes do Google Sheets para o Firestore. 3) Refatorar todo o código de acesso a dados no backend para usar o SDK do google-cloud-firestore em vez do gspread. ¹		usam Google Sheets.
B-ST-02	Implementação de Tarefas Assíncronas com Celery e Redis	Integrar Celery e Redis ao projeto Flask para processar operações lentas (como chamadas à API Gemini) em segundo plano. O endpoint do webhook passará a enfileirar uma tarefa no Celery e responderá imediatamente com um status 200 OK para a API do	Curto Prazo	Depende de: B-ST-01 (idealmente, para rodar sobre a nova infraestrutura de DB).

		WhatsApp. Um worker Celery executará a tarefa e enviará a resposta ao usuário quando estiver pronta, melhorando a responsividade percebida. ¹		
B-ST-03	Implementação da Integração com a API do Google Calendar	Integrar a API do Google Calendar. Implementar o fluxo de autenticação OAuth 2.0 para que os usuários possam conceder permissão de forma segura. Desenvolver as funções de backend para ler eventos do calendário do usuário e para criar novos eventos a partir de tarefas do PlanejaBot, usando a biblioteca de cliente do Google para Python. ¹	Curto Prazo	Depende de: B-ST-01, D-ST-03.
B-ST-04	Implementação da Lógica de Gamificação (Backend)	Desenvolver a lógica no backend para rastrear pontos e streaks. Isso envolve adicionar	Curto Prazo	Depende de: B-ST-01, D-ST-01.

		campos ao modelo de dados do usuário no Firestore (ex: xp_points, current_streak, last_active_date). A lógica deve ser acionada sempre que uma tarefa for concluída ou em uma verificação diária para atualizar os streaks. ¹		
B-ST-05	Implementação da Decomposição Inteligente de Metas (Backend)	Aprimorar o prompt enviado à Gemini ao criar uma meta complexa, instruindo-a especificamente a sugerir um plano de ação com subtarefas e retornar em um formato estruturado (JSON). O backend deve ser capaz de receber essa lista de subtarefas sugeridas, apresentá-la ao usuário e adicioná-las como novas tarefas no Firestore se o usuário concordar. ¹	Curto Prazo	Depende de: B-ST-01.

B-MT-01	Implementação da Entrada de Tarefas por Áudio	Habilitar o processamento de mensagens de voz do WhatsApp. O backend receberá o arquivo de áudio, o enviará para a API Gemini (que possui capacidade de transcrição/processamento de áudio) e processará a resposta da mesma forma que uma mensagem de texto, extraindo a intenção e as entidades da transcrição. ¹	Médio Prazo	Depende de: B-ST-02 (ideal para processamento de áudio assíncrono).
B-MT-02	Implementação da Extração de Tarefas de Imagens	Habilitar o processamento de imagens. O backend receberá uma imagem, a enviará para a API Gemini (usando suas capacidades de visão/OCR) com um prompt para extrair texto e identificar tarefas, e processará a lista de tarefas retornada para o usuário. ¹	Médio Prazo	Depende de: B-ST-02 (ideal para processamento de imagem assíncrono).

B-MT-03	Desenvolvimento do Backend para o Dashboard Web	Criar uma API RESTful segura no backend Flask para servir os dados para o dashboard web. Os endpoints devem ser protegidos (ex: com tokens JWT) e fornecer dados agregados sobre tarefas, metas e produtividade para um usuário autenticado, lendo diretamente do Firestore. ¹	Médio Prazo	Depende de: B-ST-01, D-MT-02.
B-MT-04	Integração com Gateway de Pagamento para a Versão Premium	Pesquisar e selecionar um gateway de pagamento (ex: Stripe, Mercado Pago). Integrar a API do gateway para gerenciar assinaturas (criar, cancelar), processar pagamentos e lidar com webhooks de status de pagamento. O backend precisará de uma lógica para habilitar/desabilitar recursos premium com base no status da assinatura do	Médio Prazo	Depende de: B-ST-01.

		usuário, que será armazenado no Firestore. ¹		
--	--	---	--	--

Referências citadas

1. Funcionalidades PlanejaBot_ Lista Completa e Priorizada_.pdf