# ****AIketing - Documento de Especificação (v9.1 - A Transformação)****

**Data da Última Atualização: 28 de Junho de 2025 Fase Atual: Fase 6 - Replanejamento e Resolução de Bloqueios de Build.**

## 1. Visão Geral do Projeto (Revisada)

* **Objetivo Principal: Desenvolver um SaaS global para capacitar Criadores de Conteúdo e PMEs com ferramentas de marketing autônomas.**
* **Solução Proposta: O AIketing é uma plataforma de marketing baseada em projetos. Usuários criam "Projetos" (ex: "Lançamento do Produto X") e, dentro de cada projeto, utilizam as ferramentas de IA para gerar os diversos conteúdos necessários (posts, anúncios, e-mails), mantendo tudo organizado.**
* **Modelo de Negócio: Assinatura (SaaS) com gateway de pagamento Stripe.**
* **Alcance: Global, com suporte para 5 idiomas (EN, PT, DE, FR, RU).**

## 2. Stack Tecnológico e Automação

* **Arquitetura Geral: Aplicação multi-container orquestrada via docker-compose.**
* **Backend: C# com .NET 8 (ASP.NET Core Web API).**
* **Frontend: Next.js com React (Pages Router), TypeScript e Tailwind CSS.**
* **Banco de Dados: PostgreSQL.**
* **IA (Texto): Modelo Open Source (phi-4-mini) auto-hospedado via Ollama.**
* **Automação de Ambiente: O ambiente de desenvolvimento completo (Frontend, Backend, DB, IA) é iniciado com um único comando: docker-compose up --build.**

## 3. Funcionalidades para a Próxima Versão (v2)

* **🚧 Dashboard de Projetos: A página /dashboard será transformada em uma área para o usuário ver, criar e gerenciar seus projetos.**
* **🚧 Sistema de Projetos:**
  + **Definição: Um Projeto terá nome, descrição, budget (opcional), e uma galeria de imagens e textos gerados.**
  + **Status: Backend precisa de novas tabelas. Frontend precisa de nova UI.**
* **🚧 Gerador de Conteúdo v2 (Baseado em Templates):**
  + **UI: A UI de geração será reformulada. O usuário primeiro escolhe a plataforma (Instagram, Google Ads, etc., com logos) e depois um template de post (ex: "Post de Imagem Única").**
  + **Preview: A UI deve mostrar uma pré-visualização que imita a aparência do post na plataforma escolhida, atualizando em tempo real com as edições.**
* **🚧 Configurações da Conta: Criar uma página /settings para gerenciamento de plano e alteração de senha.**
* **💡 Visão de Futuro (Pós-v2): Implementar o agendador de posts e a integração direta com APIs de redes sociais.**

## 4. Status Atual e Problemas a Serem Resolvidos

* **Status Geral: A implementação do ambiente automatizado com docker-compose está em andamento, mas bloqueada por um erro de build no frontend. As tentativas anteriores de correção falharam.**
* **Problema Bloqueador Atual: Ao executar docker-compose up --build, o processo falha durante a etapa RUN npm run build do Dockerfile do cliente.**
* **Erro Específico Identificado:**
  + **Type error: Expected 2 arguments, but got 1.**
  + **Local: components/PricingSection.tsx, na linha que tenta renderizar a lista de "Features" de um plano.**
* **Análise Honesta: Minhas tentativas anteriores de corrigir este erro foram ineficazes e repetitivas. A causa raiz é um erro de sintaxe na forma como a função de tradução t() está sendo chamada para renderizar um array, um erro que eu introduzi e não consegui resolver. Uma nova abordagem é necessária.**