

Documento de Requisitos do Sistema

Myfitwell - Sistema de Controle de Dietas

Versão: 1.0

Data: Dezembro 2024

Idioma: Português Brasileiro (PT-BR)

1. Visão Geral do Sistema

1.1 Propósito

O **Myfitwell** é uma plataforma integrada para gestão completa de dietas, conectando usuários finais, nutricionistas e fornecedores em um ecossistema de saúde e bem-estar alimentar.

1.2 Objetivos Principais

- Facilitar o controle nutricional pessoal e familiar
- Conectar usuários com nutricionistas qualificados
- Integrar marketplace para compra de produtos e serviços
- Automatizar listas de compras baseadas em dietas
- Acompanhar evolução nutricional e custos
- Garantir segurança e conformidade com LGPD

1.3 Escopo

Sistema web responsivo (mobile-first para usuários) com 4 áreas principais:

- Área Administrativa
- Área do Cliente/Usuário
- Área do Nutricionista
- Área do Marketplace (Fornecedores/Lojas)

2. Perfis de Usuário

2.1 Administrador

- Acesso total ao sistema
- Gestão de usuários e conteúdo

- Dashboards analíticos
- Configurações gerais

2.2 Cliente/Usuário

- Auto cadastro
- Gestão pessoal de dietas
- Controle de estoque e compras
- Agendamento com nutricionistas
- Acesso ao marketplace

2.3 Nutricionista

- Cadastro profissional (CRM)
- Criação de dietas para clientes
- Gestão de agenda e consultas
- Acompanhamento de pacientes
- Presença no marketplace

2.4 Fornecedor/Loja

- Cadastro de estabelecimento
- Gestão de produtos/serviços
- Controle de pedidos e orçamentos
- Dashboard financeiro

2.5 Perfis Múltiplos

Um usuário pode ter múltiplos perfis simultaneamente, com dashboards independentes para cada área.

3. Requisitos Funcionais

3.1 Área Administrativa

RF-ADM-001: Dashboard Administrativo

- **Descrição:** Painel completo com métricas de usuários por perfil, movimentações financeiras, engajamento
- **Filtros:** Por período, perfil, região
- **Visualizações:** Gráficos, tabelas, KPIs

RF-ADM-002: Gestão de Usuários

- **Descrição:** CRUD completo de usuários
- **Funcionalidades:** Busca, filtros, ativação/desativação, auditoria

RF-ADM-003: Cadastro de Produtos Base

- **Descrição:** Produtos genéricos sem marca ou preço
- **Campos:** Nome, categoria, informações nutricionais completas, unidade de medida
- **Fonte:** Tabela TACO/IBGE como referência

RF-ADM-004: Gestão de Unidades de Medida

- **Descrição:** Cadastro completo de unidades para nutrição
- **Exemplos:** g, mg, kg, ml, L, un, porção, colher (sopa), colher (chá), xícara

RF-ADM-005: Tipos de Pedido

- **Descrição:** Configuração de tipos (orçamento, compra direta)
- **Atributos:** Nome, descrição, regras

RF-ADM-006: Status de Pedidos

- **Descrição:** Gerenciamento de status
- **Valores:** Pendente, Efetivado, Cancelado, Devolvido, Enviado, Entregue

RF-ADM-007: Configuração de APIs

- **Descrição:** Cadastro de credenciais para integrações
 - **APIs:** IA (OpenAI/Anthropic), E-mail (SendGrid/AWS SES), OCR, Frete (Correios/Melhor Envio)
-

3.2 Área do Cliente/Usuário

RF-CLI-001: Auto Cadastro

- **Descrição:** Registro autônomo com verificação de e-mail
- **Campos:** Nome completo, e-mail, senha, telefone, data de nascimento
- **Validações:** E-mail único, senha forte, maioridade

RF-CLI-002: Perfil do Usuário

- **Descrição:** Gestão completa do perfil

- **Campos:** Foto, dados pessoais, dados antropométricos
- **Dados Antropométricos:** Peso, altura, IMC (calculado), circunferências
- **Dados de Saúde:** Restrições alimentares, alergias, intolerâncias, condições médicas, nível de atividade física
- **Histórico:** Manter histórico de peso e medidas

RF-CLI-003: Gestão de Endereços

- **Descrição:** CRUD de endereços
- **Tipos:** Residencial, entrega, cobrança
- **Campos:** CEP (com busca automática), logradouro, número, complemento, bairro, cidade, UF
- **Validações:** CEP válido, endereço principal obrigatório

RF-CLI-004: Cadastro de Produtos Específicos

- **Descrição:** Produtos com marca e preço por usuário
- **Campos:** Produto base (referência), marca, preço, quantidade, loja, data de compra
- **Funcionalidades:** Histórico de preços, alertas de variação

RF-CLI-005: Gestão de Dietas

- **Descrição:** Criação e acompanhamento de dietas
- **Estrutura:** Dieta → Refeições → Produtos/Alimentos
- **Refeições:** Café da manhã, lanche da manhã, almoço, lanche da tarde, jantar, ceia, pré-treino, pós-treino
- **Informações:** Objetivo, data início/fim, nutricionista responsável (opcional)
- **Cálculos Automáticos:** Somatória de macros e micros por refeição e total diário
- **Comparação:** Meta vs. Consumido

RF-CLI-006: Controle de Estoque

- **Descrição:** Gestão de produtos em casa
- **Campos:** Produto, quantidade, validade, local de armazenamento
- **Alertas:** Validade próxima, estoque baixo
- **Movimentações:** Entrada (compra), saída (consumo), ajuste

RF-CLI-007: Lista de Compras

- **Descrição:** Geração automática baseada em dieta e estoque

- **Lógica:** (Dieta período X dias) - (Estoque atual) = Lista de compras
- **Funcionalidades:** Adicionar itens extras, marcar como comprado, compartilhar
- **Agrupamento:** Por categoria, loja, prioridade

RF-CLI-008: Grupo Familiar

- **Descrição:** Criação e gestão de grupos
- **Funcionalidades:** Convidar membros (e-mail/link), aceitar/recusar convites, sair do grupo
- **Compartilhamento:** Estoque e lista de compras unificados
- **Privacidade:** Dietas individuais permanecem privadas salvo compartilhamento explícito

RF-CLI-009: Agendamento de Consultas

- **Descrição:** Busca e agendamento com nutricionistas
- **Filtros:** Especialidade, localização, avaliação, preço
- **Processo:** Verificar agenda → Selecionar horário → Confirmar → Pagamento (se aplicável)
- **Notificações:** Confirmação, lembretes, reagendamento

RF-CLI-010: Acompanhamento de Evolução

- **Descrição:** Visualização gráfica da evolução
- **Métricas:** Peso, IMC, medidas, fotos de progresso
- **Compartilhamento:** Permitir acesso ao nutricionista
- **Relatórios:** Exportação em PDF

RF-CLI-011: Atualização de Estoque (OCR)

- **Descrição:** Leitura de cupom fiscal para atualizar preços e estoque
- **Processo:** Upload de imagem → OCR → Identificação de produtos → Confirmação → Atualização
- **Validações:** Correspondência produto base/específico

RF-CLI-012: Pedidos e Orçamentos

- **Descrição:** Solicitação de orçamentos ou compras diretas
- **Tipos:** Baseado em lista de compras, produtos avulsos, serviços
- **Fluxo Orçamento:** Solicitar → Aguardar propostas → Comparar → Escolher → Efetivar
- **Fluxo Compra Direta:** Adicionar ao carrinho → Checkout → Pagamento → Confirmação
- **Endereços:** Múltiplos endereços de entrega por pedido

RF-CLI-013: Receitas Culinárias

- **Descrição:** Cadastro e consulta de receitas
- **Campos:** Nome, ingredientes (com quantidades), modo de preparo, tempo, rendimento, foto
- **Cálculos:** Informações nutricionais automáticas baseadas nos ingredientes
- **Funcionalidades:** Adicionar à dieta, ajustar porções, favoritar

RF-CLI-014: Integração Dieta-Compras

- **Descrição:** Ao efetivar compra, atualizar dieta
- **Lógica:** Produtos comprados → Adicionar ao estoque → Marcar na lista de compras → Refletir na dieta
- **Indicadores:** Mostrar nutrientes disponíveis vs. planejados

RF-CLI-015: Avaliações

- **Descrição:** Avaliar nutricionistas e fornecedores
 - **Regra:** Apenas após consulta/compra efetivada
 - **Campos:** Estrelas (1-5), comentário, data
 - **Cálculo:** Nota média ponderada
-

3.3 Área do Nutricionista

RF-NUT-001: Cadastro Profissional

- **Descrição:** Registro com documentação profissional
- **Campos:** CRM, CPF, CNPJ (opcional), endereço consultório, especialidades, biografia, foto
- **Validações:** CRM válido e ativo
- **Documentos:** Upload de comprovantes

RF-NUT-002: Criação de Dietas

- **Descrição:** Elaborar dietas para pacientes
- **Campos:** Paciente, objetivo, duração, observações, prescrições
- **Produtos:** Escolher produtos base ou específicos
- **Templates:** Salvar modelos de dietas

RF-NUT-003: Gestão de Agenda

- **Descrição:** Configurar disponibilidade

- **Campos:** Dias da semana, horários, duração da consulta, tipo (presencial/online)
- **Bloqueios:** Marcar indisponibilidades (férias, feriados)
- **Recorrência:** Padrões semanais

RF-NUT-004: Consultas

- **Descrição:** Registro de consultas e retornos
- **Campos:** Paciente, data/hora, tipo, anamnese, avaliação, conduta, retorno agendado
- **Prontuário:** Histórico completo do paciente
- **Status:** Agendada, realizada, cancelada, faltou

RF-NUT-005: Acompanhamento de Pacientes

- **Descrição:** Monitorar evolução com consentimento
- **Dados Visualizados:** Dieta seguida, peso, medidas, aderência
- **Comunicação:** Mensagens, ajustes na dieta
- **Alertas:** Desvios significativos, não aderência

RF-NUT-006: Cadastro de Dados do Paciente

- **Descrição:** Coletar dados antropométricos e clínicos
- **Compartilhamento:** Solicitar acesso a dados já preenchidos pelo usuário
- **Consentimento:** Termo de aceite digital (LGPD)

RF-NUT-007: Receitas Profissionais

- **Descrição:** Criar receitas com informações nutricionais
- **Acesso:** Compartilhar com pacientes
- **Biblioteca:** Organizar por categoria, objetivo

RF-NUT-008: Presença no Marketplace

- **Descrição:** Perfil público para agendamentos
 - **Informações:** Especialidades, preços, avaliações, fotos
 - **Destaque:** Opção de anúncio patrocinado (futuro)
-

3.4 Área do Marketplace (Fornecedores/Lojas)

RF-MKT-001: Cadastro de Fornecedor

- **Descrição:** Registro de estabelecimento
- **Campos:** Razão social, CNPJ, endereço, telefone, e-mail, logo, categoria
- **Validações:** CNPJ válido, documentação
- **Tipos:** Supermercado, loja natural, restaurante, loja online

RF-MKT-002: Gestão de Produtos/Serviços

- **Descrição:** CRUD de produtos e serviços
- **Campos Produto:** Nome, descrição, categoria, marca, preço, estoque, fotos, informações nutricionais
- **Campos Marmita/Refeição:** Peso, componentes, informações nutricionais, embalagem, validade
- **Variações:** Tamanhos, sabores
- **Promoções:** Preços especiais, combos

RF-MKT-003: Cálculo de Frete

- **Descrição:** Integração com APIs de frete
- **Entrada:** CEP destino, peso, dimensões
- **Saída:** Opções de entrega (prazo, valor)
- **Transportadoras:** Correios, Melhor Envio, frota própria

RF-MKT-004: Gestão de Pedidos

- **Descrição:** Controle completo de pedidos
- **Status:** Novo, em preparação, pronto, enviado, entregue, cancelado
- **Notificações:** Atualizações para cliente
- **Impressão:** Etiquetas, notas fiscais

RF-MKT-005: Orçamentos

- **Descrição:** Responder solicitações de orçamento
- **Processo:** Receber solicitação → Analisar → Enviar proposta → Aguardar aceitação
- **Validade:** Prazo de validade da proposta

RF-MKT-006: Dashboard Financeiro

- **Descrição:** Visão consolidada de finanças

- **Métricas:** Pedidos por tipo/status, receitas, custos, lucro, ticket médio
 - **Filtros:** Período, produto, canal
 - **Relatórios:** Exportação de dados
-

4. Requisitos Não-Funcionais

4.1 Segurança (RNF-SEG)

RNF-SEG-001: Autenticação

- Autenticação via Supabase Auth
- Suporte a login social (Google, Facebook)
- MFA (autenticação de dois fatores) opcional

RNF-SEG-002: Autorização

- Row Level Security (RLS) em todas as tabelas
- Políticas baseadas em perfis e propriedade de dados
- Acesso granular por funcionalidade

RNF-SEG-003: Criptografia

- HTTPS obrigatório
- Senhas com hash bcrypt
- Dados sensíveis (CPF, CRM) criptografados em repouso

RNF-SEG-004: LGPD

- Termo de consentimento explícito
- Direito ao esquecimento (exclusão de dados)
- Portabilidade de dados (exportação)
- Registro de consentimentos e acessos
- DPO (Data Protection Officer) designado

4.2 Performance (RNF-PERF)

RNF-PERF-001: Tempo de Resposta

- Páginas carregam em < 3s
- APIs respondem em < 500ms (p95)

- Busca de produtos em < 1s

RNF-PERF-002: Escalabilidade

- Suportar 10.000 usuários simultâneos
- Cache de dados estáticos (Redis/Supabase Cache)
- CDN para imagens e assets

RNF-PERF-003: Otimização

- Lazy loading de imagens
- Paginação de listagens
- Índices otimizados no banco

4.3 Usabilidade (RNF-USA)

RNF-USA-001: Design

- Interface minimalista e clean
- Paleta de cores focada em saúde (verde, azul)
- Tipografia legível (mínimo 14px mobile)

RNF-USA-002: Responsividade

- Mobile-first para área do cliente
- Breakpoints: 320px, 768px, 1024px, 1440px
- Touch-friendly (botões mínimo 44x44px)

RNF-USA-003: Acessibilidade

- Contraste WCAG AA
- Navegação por teclado
- Screen reader friendly (ARIA labels)
- Textos alternativos em imagens

RNF-USA-004: UX

- Máximo 3 cliques para ações principais
- Feedback visual imediato
- Mensagens de erro claras e acionáveis
- Onboarding guiado para novos usuários

4.4 Confiabilidade (RNF-CONF)

RNF-CONF-001: Disponibilidade

- 99.5% uptime (SLA)
- Backup diário automático
- Recovery Point Objective (RPO): 24h
- Recovery Time Objective (RTO): 4h

RNF-CONF-002: Auditoria

- Log de todas as operações críticas
- Rastreabilidade de alterações (quem, quando, o quê)
- Retenção de logs por 2 anos

4.5 Compatibilidade (RNF-COMP)

RNF-COMP-001: Navegadores

- Chrome (últimas 2 versões)
- Firefox (últimas 2 versões)
- Safari (últimas 2 versões)
- Edge (últimas 2 versões)

RNF-COMP-002: Dispositivos

- iOS 14+
- Android 10+
- Progressive Web App (PWA) para instalação mobile

5. Modelo de Dados Conceitual

5.1 Entidades Principais

Usuários e Autenticação

- **auth.users** (Supabase Auth - nativa)
- **profiles** (extensão do usuário)
- **user_roles** (perfis: cliente, nutricionista, fornecedor, admin)

- **addresses** (endereços)

Produtos e Nutrição

- **product_bases** (produtos genéricos)
- **nutritional_info** (informações nutricionais)
- **measurement_units** (unidades de medida)
- **product_categories** (categorias)
- **user_products** (produtos específicos do usuário)
- **price_history** (histórico de preços)

Dietas e Alimentação

- **diets** (dietas)
- **meals** (refeições)
- **diet_items** (produtos na dieta)
- **diet_objectives** (objetivos: emagrecer, ganhar massa, etc)
- **recipes** (receitas culinárias)
- **recipe_ingredients** (ingredientes das receitas)

Estoque e Compras

- **inventory** (estoque por usuário/grupo)
- **shopping_lists** (listas de compras)
- **shopping_list_items** (itens da lista)
- **purchase_history** (histórico de compras)

Grupos Familiares

- **family_groups** (grupos)
- **family_members** (membros do grupo)
- **group_invitations** (convites pendentes)

Nutricionistas

- **nutritionists** (dados profissionais)
- **nutritionist_specialties** (especialidades)
- **schedules** (agendas)

- **appointments** (consultas)
- **appointment_types** (tipo de consulta)
- **patient_records** (prontuários)

Marketplace

- **suppliers** (fornecedores/lojas)
- **supplier_products** (produtos de fornecedores)
- **orders** (pedidos)
- **order_items** (itens do pedido)
- **order_status_history** (histórico de status)
- **quotes** (orçamentos)
- **quote_items** (itens do orçamento)
- **shipping_methods** (métodos de envio)

Avaliações

- **reviews** (avaliações)
- **review_responses** (respostas dos avaliados)

Sistema

- **api_configs** (configurações de APIs)
- **audit_logs** (logs de auditoria)
- **consent_logs** (consentimentos LGPD)
- **notifications** (notificações)

5.2 Relacionamentos Principais

```

users (1) —< (N) profiles
profiles (1) —< (N) addresses
profiles (1) —< (N) user_products
profiles (1) —< (N) diets
profiles (1) —< (N) inventory
profiles (N) —< (N) family_groups [many-to-many via family_members]

diets (1) —< (N) meals
meals (1) —< (N) diet_items
diet_items (N) —> (1) product_bases OR user_products

```

product_bases (1) —< (1) nutritional_info
recipes (1) —< (N) recipe_ingredients
recipe_ingredients (N) —> (1) product_bases

profiles (1) —< (N) appointments
nutritionists (1) —< (N) appointments
nutritionists (1) —< (N) schedules

suppliers (1) —< (N) supplier_products
profiles (1) —< (N) orders
orders (1) —< (N) order_items
order_items (N) —> (1) supplier_products

profiles (1) —< (N) reviews
suppliers (1) —< (N) reviews
nutritionists (1) —< (N) reviews

6. Regras de Negócio

RN-001: Cadastro de Usuário

- E-mail deve ser único no sistema
- Senha mínima de 8 caracteres (maiúscula, minúscula, número, especial)
- Verificação de e-mail obrigatória antes de usar o sistema
- Usuário deve ter pelo menos 18 anos

RN-002: Perfis Múltiplos

- Um usuário pode ter múltiplos perfis (cliente + nutricionista + fornecedor)
- Dashboards separados por perfil
- Transição entre perfis sem logout

RN-003: Produtos Base vs Específicos

- Produtos base são globais (cadastro admin)
- Produtos específicos são privados por usuário
- Produtos específicos devem referenciar um produto base
- Informações nutricionais herdadas do produto base (editáveis)

RN-004: Cálculo de Dieta

- Somar todos os nutrientes dos produtos de cada refeição

- Total diário = soma de todas as refeições
- Exibir comparativo: planejado vs consumido
- Alertas para deficiências ou excessos (baseado em objetivos)

RN-005: Lista de Compras

- Lista gerada: (Dieta X dias) - Estoque atual
- Usuário pode adicionar itens extras manualmente
- Se em grupo familiar, considerar estoque compartilhado
- Marcar itens como comprados remove da lista

RN-006: Grupo Familiar

- Apenas um membro pode criar o grupo
- Convites por e-mail com token único (validade 7 dias)
- Membro pode aceitar ou recusar convite
- Membro pode sair do grupo a qualquer momento
- Estoque e lista de compras são compartilhados
- Dietas permanecem privadas (salvo compartilhamento explícito)
- Exclusão do grupo: se criador sair, grupo é dissolvido

RN-007: Agendamento de Consultas

- Nutricionista deve ter agenda configurada
- Horários disponíveis não podem sobrepor consultas existentes
- Cliente pode agendar com antecedência mínima de 24h
- Cancelamento até 12h antes sem cobrança
- Nutricionista pode bloquear datas/horários

RN-008: Avaliações

- Apenas após consulta realizada ou compra entregue
- Uma avaliação por serviço/produto/consulta
- Resposta do avaliado permitida (uma única vez)
- Nota média calculada automaticamente
- Avaliações não podem ser editadas após 7 dias

RN-009: Pedidos e Orçamentos

- Orçamento: múltiplos fornecedores podem enviar propostas
- Prazo de validade do orçamento: 48h (configurável)
- Compra direta: disponível apenas se produto tem estoque
- Ao efetivar pedido, atualizar estoque do fornecedor
- Ao receber pedido, atualizar estoque do cliente
- Frete calculado por CEP destino

RN-010: Atualização por OCR

- OCR extrai: produtos, quantidades, preços, data, loja
- Sistema sugere correspondência com produtos cadastrados
- Usuário valida antes de atualizar
- Atualiza price_history e inventory

RN-011: Integração Dieta-Compras

- Ao marcar item como comprado na lista, adicionar ao estoque
- Estoque atualizado reflete na dieta (nutrientes disponíveis)
- Cálculo: nutrientes consumidos = produtos com estoque

RN-012: Segurança e LGPD

- Termo de uso e política de privacidade obrigatórios no cadastro
- Consentimento explícito para compartilhar dados com nutricionista
- Direito de exportar dados pessoais (formato JSON/PDF)
- Direito de excluir conta (anonimização de dados históricos)
- Logs de acesso mantidos por 2 anos

RN-013: Nutricionista

- CRM deve ser validado (consulta API CFN ou similar)
- Apenas nutricionistas verificados aparecem no marketplace
- Nutricionista pode criar dietas apenas para pacientes com consentimento
- Acesso a dados do paciente requer consentimento ativo

RN-014: Fornecedor

- CNPJ validado (Receita Federal)

- Produtos devem ter informações nutricionais completas
 - Marmitas/refeições devem ter peso e composição detalhados
 - Responsabilidade por atualização de estoque e preços
-

7. Integrações

7.1 APIs de IA

OpenAI / Anthropic Claude

- **Uso:** Geração de dietas com base em objetivos e dados do usuário
- **Entrada:** Perfil usuário (peso, altura, objetivo, restrições), preferências
- **Saída:** Dieta estruturada (refeições, produtos, quantidades)
- **Autenticação:** API Key (cadastrada em api_configs)

7.2 E-mail

SendGrid / AWS SES

- **Uso:** Envio de confirmações, notificações, lembretes
- **Templates:**
 - Boas-vindas
 - Verificação de e-mail
 - Convite grupo familiar
 - Confirmação de consulta
 - Confirmação de pedido
 - Lembrete de consulta
- **Autenticação:** API Key

7.3 OCR

Google Cloud Vision / AWS Textract

- **Uso:** Leitura de cupons fiscais
- **Entrada:** Imagem (JPG/PNG) do cupom
- **Saída:** Texto estruturado (produtos, preços, quantidades)
- **Pós-processamento:** Matching com produtos cadastrados (ML/NLP)

7.4 Frete

Correios / Melhor Envio

- **Uso:** Cálculo de frete em tempo real
- **Entrada:** CEP origem, CEP destino, peso, dimensões
- **Saída:** Opções de entrega (transportadora, prazo, valor)
- **Webhook:** Rastreamento de encomendas

7.5 Pagamentos (Futuro)

Stripe / Mercado Pago

- **Uso:** Processamento de pagamentos
- **Métodos:** Cartão de crédito, PIX, boleto
- **Webhook:** Confirmação de pagamento

7.6 CEP

ViaCEP / Brasil API

- **Uso:** Autocompletar endereço por CEP
 - **Gratuito:** Sim
 - **Cache:** Implementar para reduzir chamadas
-

8. Stack Tecnológico Recomendado

8.1 Backend

Banco de Dados e Backend

- **Supabase** (obrigatório por requisito)
 - PostgreSQL 15+
 - Row Level Security (RLS)
 - Supabase Auth
 - Supabase Storage (imagens)
 - Supabase Functions (serverless)
 - Supabase Realtime (notificações em tempo real)

Linguagens e Frameworks Backend

- **TypeScript** - Type safety
- **Deno** ou **Node.js** - Runtime para Supabase Functions
- **Zod** - Validação de schemas

8.2 Frontend

Framework Principal

- **Next.js 14+** (App Router)
 - React 18+
 - Server Components
 - API Routes
 - Image Optimization
 - SEO-friendly

UI e Estilo

- **Tailwind CSS** - Utility-first CSS (mobile-first)
- **shadcn/ui** - Componentes acessíveis
- **Radix UI** - Primitivos de UI
- **Lucide Icons** - Ícones minimalistas

Estado e Dados

- **React Query (TanStack Query)** - Cache e sincronização
- **Zustand** ou **Jotai** - Estado global leve
- **Supabase JS Client** - Cliente para Supabase

Formulários e Validação

- **React Hook Form** - Gerenciamento de forms
- **Zod** - Validação client-side

8.3 Mobile

PWA (Progressive Web App)

- **Next.js PWA** - Configuração PWA
- **Workbox** - Service workers
- **Web APIs:** Camera (fotos), Geolocation, Notifications

Nativo (Opcional - Futuro)

- **React Native** com Expo
- **Capacitor** - Bridge para funcionalidades nativas

8.4 Infraestrutura

Hospedagem Frontend

- **Vercel** - Hospedagem otimizada para Next.js
- **Cloudflare Pages** - Alternativa com CDN global

CDN e Storage

- **Supabase Storage** - Imagens de produtos, fotos de usuários
- **Cloudflare Images** - Otimização de imagens (opcional)

Monitoramento

- **Sentry** - Error tracking
- **PostHog** ou **Mixpanel** - Analytics
- **Supabase Dashboard** - Métricas de banco

CI/CD

- **GitHub Actions** - Deploy automático
- **Vercel CI** - Integração contínua

8.5 Desenvolvimento

Versionamento

- **Git + GitHub**
- **Conventional Commits**
- **GitFlow** ou **Trunk-based**

Qualidade de Código

- **ESLint** - Linting
- **Prettier** - Formatação
- **Husky** - Git hooks
- **lint-staged** - Pre-commit checks

Testes

- **Vitest** - Unit tests
- **Playwright** - E2E tests
- **React Testing Library** - Component tests

Documentação

- **Storybook** - Catálogo de componentes
- **TypeDoc** - Documentação de código

9. Considerações de Arquitetura Supabase

9.1 Row Level Security (RLS)

Políticas Principais

Tabela: profiles

```
sql

-- SELECT: Usuário vê apenas seu próprio perfil
CREATE POLICY "Users can view own profile" ON profiles
  FOR SELECT USING (auth.uid() = user_id);

-- INSERT: Usuário cria apenas seu próprio perfil
CREATE POLICY "Users can insert own profile" ON profiles
  FOR INSERT WITH CHECK (auth.uid() = user_id);

-- UPDATE: Usuário atualiza apenas seu próprio perfil
CREATE POLICY "Users can update own profile" ON profiles
  FOR UPDATE USING (auth.uid() = user_id);
```

Tabela: diets

```
sql
```

```
-- SELECT: Usuário vê suas dietas OU dietas compartilhadas com ele (nutricionista)
```

```
CREATE POLICY "Users can view own diets" ON diets
```

```
FOR SELECT USING (
```

```
  auth.uid() = user_id OR
```

```
  auth.uid() = nutritionist_id
```

```
);
```

```
-- INSERT: Usuário ou nutricionista autorizado
```

```
CREATE POLICY "Users and nutritionists can create diets" ON diets
```

```
FOR INSERT WITH CHECK (
```

```
  auth.uid() = user_id OR
```

```
  (auth.uid() = nutritionist_id AND has_consent(user_id, auth.uid()))
```

```
);
```

Tabela: inventory (Estoque)

```
sql
```

```
-- SELECT: Usuário vê seu estoque OU estoque do grupo familiar
```

```
CREATE POLICY "Users can view own or family inventory" ON inventory
```

```
FOR SELECT USING (
```

```
  auth.uid() = user_id OR
```

```
  auth.uid() IN (SELECT user_id FROM family_members WHERE family_group_id = inventory.family_group_id)
```

```
);
```

Tabela: orders (Pedidos)

```
sql
```

```
-- SELECT: Cliente vê seus pedidos, fornecedor vê pedidos recebidos
```

```
CREATE POLICY "View orders" ON orders
```

```
FOR SELECT USING (
```

```
  auth.uid() = user_id OR
```

```
  auth.uid() IN (SELECT user_id FROM suppliers WHERE id = orders.supplier_id)
```

```
);
```

9.2 Funções do Banco de Dados

Cálculo Automático de IMC

```
sql
```

```

CREATE OR REPLACE FUNCTION calculate_imc()
RETURNS TRIGGER AS $
BEGIN
    IF NEW.weight IS NOT NULL AND NEW.height IS NOT NULL THEN
        NEW.imc = NEW.weight / (NEW.height * NEW.height);
    END IF;
    RETURN NEW;
END;
$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_calculate_imc
BEFORE INSERT OR UPDATE ON profiles
FOR EACH ROW
EXECUTE FUNCTION calculate_imc();

```

Atualização de Nota Média de Avaliações

```

sql

CREATE OR REPLACE FUNCTION update_average_rating()
RETURNS TRIGGER AS $
BEGIN
    IF TG_OP = 'INSERT' OR TG_OP = 'UPDATE' THEN
        UPDATE suppliers
        SET average_rating = (
            SELECT AVG(rating) FROM reviews WHERE supplier_id = NEW.supplier_id
        )
        WHERE id = NEW.supplier_id;
    END IF;
    RETURN NEW;
END;
$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_update_supplier_rating
AFTER INSERT OR UPDATE ON reviews
FOR EACH ROW
WHEN (NEW.supplier_id IS NOT NULL)
EXECUTE FUNCTION update_average_rating();

```

Validação de Estoque ao Criar Pedido

```

sql

```

```

CREATE OR REPLACE FUNCTION check_inventory_before_order()
RETURNS TRIGGER AS $
DECLARE
    product_stock INTEGER;
BEGIN
    SELECT stock INTO product_stock
    FROM supplier_products
    WHERE id = NEW.product_id;

    IF product_stock < NEW.quantity THEN
        RAISE EXCEPTION 'Estoque insuficiente para o produto %', NEW.product_id;
    END IF;

    RETURN NEW;
END;
$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_check_inventory
BEFORE INSERT ON order_items
FOR EACH ROW
EXECUTE FUNCTION check_inventory_before_order();

```

9.3 Supabase Storage

Buckets Principais

- **profile-pictures** - Fotos de perfil (público)
- **product-images** - Fotos de produtos (público)
- **recipe-images** - Fotos de receitas (público)
- **supplier-logos** - Logos de fornecedores (público)
- **receipts** - Cupons fiscais (privado, RLS)
- **documents** - Documentos de nutricionistas (privado)

Políticas de Storage

sql

-- profile-pictures: usuário pode fazer upload apenas da própria foto

CREATE POLICY "Users can upload own profile picture"

ON storage.objects FOR INSERT

WITH CHECK (

bucket_id = 'profile-pictures' AND
(storage.foldername(name))[1] = auth.uid()::text
);

-- product-images: apenas fornecedores podem fazer upload

CREATE POLICY "Suppliers can upload product images"

ON storage.objects FOR INSERT

WITH CHECK (

bucket_id = 'product-images' AND
EXISTS (SELECT 1 FROM suppliers WHERE user_id = auth.uid())
);

9.4 Supabase Edge Functions

Função: Geração de Dieta com IA

typescript

```

// supabase/functions/generate-diet/index.ts
import { serve } from 'https://deno.land/std@0.168.0/http/server.ts'
import { createClient } from '@supabase/supabase-js'

serve(async (req) => {
  const { userId, objective, restrictions } = await req.json()

  // Buscar dados do usuário
  const supabase = createClient(...)
  const { data: profile } = await supabase
    .from('profiles')
    .select('*')
    .eq('user_id', userId)
    .single()

  // Chamar API de IA
  const aiResponse = await fetch('https://api.openai.com/v1/chat/completions', {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${Deno.env.get('OPENAI_API_KEY')}`,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      model: 'gpt-4',
      messages: [{
        role: 'system',
        content: 'Você é um nutricionista especializado...'
      }, {
        role: 'user',
        content: `Crie uma dieta para: ${JSON.stringify(profile)}, Objetivo: ${objective}`
      }]
    })
  })

  const aiData = await aiResponse.json()
  // Processar e salvar dieta no banco

  return new Response(JSON.stringify({ success: true }), {
    headers: { 'Content-Type': 'application/json' }
  })
})

```

Função: Processamento de OCR

typescript

```
// supabase/functions/process-receipt/index.ts
import { serve } from 'https://deno.land/std@0.168.0/http/server.ts'

serve(async (req) => {
  const { imageUrl } = await req.json()

  // Chamar Google Cloud Vision
  const ocrResponse = await fetch('https://vision.googleapis.com/v1/images:annotate', {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${Deno.env.get('GOOGLE_VISION_KEY')}`,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      requests: [{
        image: { source: { imageUrl: imageUrl } },
        features: [{ type: 'TEXT_DETECTION' }]
      }]
    })
  })

  const ocrData = await ocrResponse.json()

  // Processar texto e identificar produtos
  const products = parseReceiptText(ocrData.responses[0].fullTextAnnotation.text)

  return new Response(JSON.stringify({ products }), {
    headers: { 'Content-Type': 'application/json' }
  })
})
```

9.5 Realtime Subscriptions

Exemplo: Notificações de Pedidos em Tempo Real

typescript

```
// Cliente (Frontend)
const supabase = createClient(...)

supabase
  .channel('orders')
  .on(
    'postgres_changes',
    {
      event: 'INSERT',
      schema: 'public',
      table: 'orders',
      filter: `supplier_id=eq.${supplierId}`
    },
    (payload) => {
      console.log('Novo pedido recebido!', payload.new)
      showNotification('Novo pedido recebido!')
    }
  )
  .subscribe()
```

10. Fluxos de Usuário Principais

10.1 Fluxo: Cadastro e Onboarding

1. Tela de Boas-Vindas

- Apresentação do Myfitwell
- Botões: "Criar Conta" / "Entrar"

2. Registro

- Formulário: Nome, E-mail, Senha
- Aceite de Termos e Política de Privacidade
- Envio de e-mail de verificação

3. Verificação de E-mail

- Link no e-mail → Ativação da conta

4. Onboarding - Passo 1: Perfil Básico

- Upload de foto (opcional)
- Data de nascimento, telefone

5. Onboarding - Passo 2: Dados de Saúde

- Peso, altura (IMC calculado automaticamente)

- Objetivo: Emagrecer, manter, ganhar peso, ganhar massa muscular
- Nível de atividade física

6. Onboarding - Passo 3: Restrições

- Alergias, intolerâncias, restrições alimentares
- Preferências alimentares

7. Onboarding - Passo 4: Endereço

- CEP → Autocomplete
- Complementos

8. Conclusão

- "Bem-vindo ao Myfitwell!"
- Sugestões de próximos passos: criar dieta, buscar nutricionista, explorar marketplace

10.2 Fluxo: Criação de Dieta com IA

1. Acessar "Minhas Dietas"

- Botão: "Criar Nova Dieta"

2. Escolher Método

- "Criar Manualmente" ou "Gerar com IA"

3. Gerar com IA

- Confirmar dados de perfil (peso, altura, objetivo)
- Selecionar duração (7, 14, 30 dias)
- Número de refeições por dia (3-6)
- Preferências adicionais (texto livre)

4. Processamento

- Loading com mensagem "Estamos criando sua dieta personalizada..."
- Chamada à Edge Function

5. Revisão da Dieta Gerada

- Visualizar dieta completa (todas as refeições)
- Totais nutricionais por dia
- Opção de editar refeições/produtos
- Salvar ou Gerar Novamente

6. Confirmação

- Dieta salva com sucesso

- Ativar como "Dieta Atual"
- Gerar lista de compras automaticamente

10.3 Fluxo: Lista de Compras e Compra

1. Visualizar Lista de Compras

- Acesso via menu ou dashboard
- Lista gerada automaticamente: (Dieta - Estoque)
- Itens agrupados por categoria

2. Adicionar Itens Extras

- Botão "+" para incluir produtos não relacionados à dieta

3. Escolher Método de Compra

- "Comprar no Marketplace" ou "Registrar Compra Externa"

4. Comprar no Marketplace

- Selecionar itens da lista
- Buscar fornecedores que tenham os produtos
- Comparar preços e avaliações

5. Adicionar ao Carrinho

- Escolher fornecedor(es)
- Calcular frete por fornecedor
- Consolidar carrinho

6. Checkout

- Revisar pedido
- Escolher endereço de entrega
- Método de pagamento (futuro)
- Confirmar pedido

7. Acompanhamento

- Status do pedido em tempo real
- Notificações de atualização
- Rastreamento de entrega

8. Recebimento

- Marcar como recebido

- Estoque atualizado automaticamente
- Itens removidos da lista de compras

10.4 Fluxo: Agendamento com Nutricionista

1. Explorar Marketplace de Nutricionistas

- Filtros: Especialidade, localização, avaliação, preço
- Cards com foto, nome, CRM, especialidade, nota

2. Visualizar Perfil do Nutricionista

- Biografia, formação, experiência
- Avaliações de pacientes
- Preços de consultas
- Botão "Agendar Consulta"

3. Selecionar Data e Horário

- Calendário com disponibilidade
- Tipo de consulta: Presencial ou Online

4. Preencher Informações

- Motivo da consulta (opcional)
- Compartilhar dados de saúde (consentimento)

5. Confirmação

- Resumo da consulta
- Confirmar agendamento
- Envio de e-mail confirmação

6. Lembretes

- 24h antes: Notificação e e-mail
- 1h antes: Notificação

7. Pós-Consulta

- Nutricionista registra prontuário
- Nutricionista cria/atualiza dieta do paciente
- Paciente recebe notificação de nova dieta
- Paciente pode avaliar consulta

10.5 Fluxo: Acompanhamento de Evolução

1. Registrar Dados

- Menu "Minha Evolução"
- Adicionar novo registro: Peso, medidas, foto (opcional)
- Data automática (hoje)

2. Visualizar Gráficos

- Gráfico de peso ao longo do tempo
- Gráfico de IMC
- Comparação com objetivo

3. Compartilhar com Nutricionista

- Botão "Compartilhar Evolução"
- Selecionar nutricionista (se já tem relação)
- Consentimento de compartilhamento

4. Nutricionista Acessa

- Ver evolução do paciente em tempo real
 - Adicionar comentários/orientações
 - Ajustar dieta conforme progresso
-

11. Wireframes e Telas Principais (Descrição)

11.1 Área do Cliente

Dashboard do Usuário

- **Header:** Logo, nome do usuário, foto, menu
- **Cards de Resumo:**
 - Dieta Atual (nome, objetivo)
 - Peso Atual vs Meta
 - Próxima Consulta
 - Lista de Compras (X itens)
- **Ações Rápidas:**
 - Registrar refeição
 - Adicionar peso

- Ver lista de compras
- **Gráfico de Evolução:** Peso últimos 30 dias

Tela de Dieta

- **Header:** Nome da dieta, período, objetivo
- **Tabs:** Por dia da semana
- **Refeições:** Accordion
 - Café da manhã, Lanche, Almoço, etc.
 - Produtos com quantidades
 - Total de nutrientes por refeição
- **Footer:** Total diário (calorias, proteínas, carboidratos, gorduras)
- **Botões:** Editar, Gerar Lista de Compras, Compartilhar

Tela de Lista de Compras

- **Filtros:** Categoria, comprado/não comprado
- **Lista:** Checkboxes, produto, quantidade, preço estimado
- **Footer:** Total estimado
- **Botões:** Adicionar item, Comprar no Marketplace, Registrar compra manual

Tela de Estoque

- **Busca:** Por nome
- **Filtros:** Categoria, validade próxima
- **Lista/Cards:** Produto, quantidade, validade, local
- **Alertas:** Badge para itens próximos ao vencimento
- **Botões:** Adicionar produto, Ajustar quantidade

Tela de Marketplace

- **Busca:** Barra de busca global
- **Filtros Laterais:** Categoria, fornecedor, faixa de preço, avaliação
- **Grid de Produtos:** Cards com imagem, nome, preço, nota
- **Tabs:** Produtos, Nutricionistas, Marmitas/Refeições

11.2 Área do Nutricionista

Dashboard do Nutricionista

- **Cards de Resumo:**
 - Total de Pacientes
 - Consultas Hoje
 - Consultas Semana
 - Avaliação Média
- **Lista de Próximas Consultas:** Nome paciente, horário, tipo
- **Gráfico:** Consultas por mês
- **Ações Rápidas:** Criar dieta, Ver agenda, Pacientes

Tela de Pacientes

- **Busca:** Por nome
- **Lista/Cards:** Foto, nome, idade, última consulta, status (ativo/inativo)
- **Ações:** Ver prontuário, Ver evolução, Agendar retorno

Tela de Criação de Dieta

- **Formulário:**
 - Selecionar paciente
 - Objetivo
 - Duração
 - Observações
- **Refeições:** Adicionar refeições dinamicamente
- **Produtos:** Buscar e adicionar produtos por refeição
- **Cálculos Automáticos:** Total nutricional em tempo real
- **Botões:** Salvar rascunho, Disponibilizar para paciente, Gerar com IA

Tela de Agenda

- **Calendário:** Semanal/Mensal
- **Horários:** Grid com slots (disponível/ocupado)
- **Ações:** Configurar disponibilidade, Bloquear horários, Ver consultas

11.3 Área do Fornecedor

Dashboard do Fornecedor

- **Cards de Resumo:**
 - Pedidos Pendentes
 - Receita do Mês
 - Ticket Médio
 - Avaliação Média
- **Gráfico:** Vendas últimos 6 meses
- **Lista de Últimos Pedidos:** Cliente, valor, status
- **Ações Rápidas:** Adicionar produto, Ver pedidos, Ver orçamentos

Tela de Produtos

- **Lista/Grid:** Foto, nome, preço, estoque, status (ativo/inativo)
- **Filtros:** Categoria, status
- **Ações:** Adicionar produto, Editar, Desativar
- **Botão de Upload em Massa:** Importar CSV

Tela de Gestão de Pedidos

- **Tabs:** Pendentes, Em preparação, Enviados, Concluídos, Cancelados
- **Lista:** Cliente, produtos, valor, endereço, status
- **Ações:** Atualizar status, Ver detalhes, Imprimir etiqueta
- **Filtros:** Período, status, forma de pagamento

11.4 Área Administrativa

Dashboard Admin

- **Cards de Resumo:**
 - Total Usuários (por perfil)
 - Novos Usuários (semana/mês)
 - Pedidos Totais
 - Receita Total (comissões)
- **Gráficos:**
 - Crescimento de usuários
 - Pedidos por categoria

- Receita por mês
- **Tabelas:** Principais métricas

Tela de Produtos Base

- **Lista:** Nome, categoria, unidade, ações
- **Busca e Filtros:** Categoria, nome
- **Ações:** Adicionar, Editar, Desativar
- **Importação:** CSV de tabela TACO

Tela de Configurações

- **Seções:**
 - APIs (IA, E-mail, OCR, Frete)
 - Tipos de Pedido
 - Status de Pedidos
 - Unidades de Medida
 - Categorias
-

12. Priorização de Features (MVP vs Futuro)

12.1 MVP (Mínimo Produto Viável) - Fase 1

Essencial para lançamento:

- ☒ Cadastro e autenticação de usuários
- ☒ Perfil de usuário com dados de saúde
- ☒ Cadastro de produtos base (admin)
- ☒ Cadastro de produtos específicos (usuário)
- ☒ Criação manual de dietas
- ☒ Visualização de informações nutricionais
- ☒ Geração de lista de compras
- ☒ Controle de estoque básico
- ☒ Cadastro de endereços
- ☒ Área administrativa básica
- ☒ Dashboard simples para cada perfil
- ☒ Segurança básica (RLS)

12.2 Fase 2 - Marketplace e Profissionais

Após validação do MVP:

- ☐ Cadastro de fornecedores
- ☐ Marketplace de produtos
- ☐ Sistema de pedidos e orçamentos
- ☐ Cálculo de frete
- ☐ Cadastro de nutricionistas
- ☐ Agendamento de consultas
- ☐ Sistema de avaliações

12.3 Fase 3 - Recursos Avançados

Expansão de funcionalidades:

- ☐ Geração de dietas com IA
- ☐ OCR para cupons fiscais
- ☐ Grupo familiar
- ☐ Acompanhamento de evolução
- ☐ Receitas culinárias
- ☐ Integração dieta-compras automática
- ☐ Notificações em tempo real (Realtime)
- ☐ Prontuário eletrônico (nutricionistas)

12.4 Fase 4 - Otimizações e Integrações

Melhorias e novos canais:

- ☐ Pagamentos integrados (Stripe/Mercado Pago)
 - ☐ App mobile nativo (React Native)
 - ☐ Dashboard avançado com BI
 - ☐ Relatórios exportáveis (PDF)
 - ☐ Integração com wearables (smartwatches)
 - ☐ Chatbot de suporte
 - ☐ Programa de fidelidade/gamificação
-

13. Métricas de Sucesso (KPIs)

13.1 Métricas de Produto

Aquisição

- Novos usuários por semana/mês
- Taxa de conversão (visitante → cadastro)

- Custo de aquisição por usuário (CAC)

Ativação

- Taxa de conclusão do onboarding
- Tempo médio para criar primeira dieta
- Usuários que adicionam produtos ao estoque (primeiros 7 dias)

Engajamento

- DAU (Daily Active Users) / MAU (Monthly Active Users)
- Frequência de login (dias/semana)
- Dietas criadas por usuário
- Listas de compras geradas
- Produtos adicionados ao estoque

Retenção

- Taxa de retenção (7, 30, 90 dias)
- Churn rate mensal
- Tempo médio de uso (sessão)

Receita (Futuro)

- GMV (Gross Merchandise Value) - volume transacionado
- Receita por usuário (ARPU)
- LTV (Lifetime Value)
- Comissões geradas

13.2 Métricas de Negócio

Marketplace

- Número de fornecedores ativos
- Produtos cadastrados
- Pedidos por dia/mês
- Taxa de conversão (visualização → pedido)
- Ticket médio

Nutricionistas

- Nutricionistas cadastrados
- Consultas agendadas/realizadas
- Taxa de ocupação da agenda
- Avaliação média

Qualidade

- NPS (Net Promoter Score)
 - Avaliação média (nutricionistas/fornecedores)
 - Taxa de devolução/cancelamento
 - Tempo de resposta do suporte
-

14. Cronograma Estimado

Fase 1 - MVP (3-4 meses)

- **Mês 1:** Setup infraestrutura, autenticação, cadastros básicos
- **Mês 2:** Dietas, produtos, cálculos nutricionais
- **Mês 3:** Lista de compras, estoque, dashboards
- **Mês 4:** Testes, ajustes, lançamento beta

Fase 2 - Marketplace (2-3 meses)

- **Mês 5:** Cadastro fornecedores, produtos marketplace
- **Mês 6:** Pedidos, nutricionistas, agendamentos
- **Mês 7:** Avaliações, refinamentos, lançamento

Fase 3 - Recursos Avançados (3-4 meses)

- **Mês 8-9:** IA, OCR, grupos familiares
- **Mês 10-11:** Receitas, evolução, integrações

Fase 4 - Otimizações (Contínuo)

- Melhorias baseadas em feedback
- Novas features
- Expansão de integrações

Total estimado para produto completo: 10-12 meses

15. Riscos e Mitigações

15.1 Riscos Técnicos

Risco	Probabilidade	Impacto	Mitigação
Problemas de performance com muitos usuários	Média	Alto	Load testing, otimização de queries, cache, CDN
Falhas na integração com APIs externas	Alta	Médio	Implementar fallbacks, retry logic, monitoramento
Inconsistência nos cálculos nutricionais	Média	Alto	Testes unitários robustos, validação com nutricionistas
Violação de dados (LGPD)	Baixa	Crítico	Auditoria de segurança, criptografia, políticas RLS rigorosas

15.2 Riscos de Negócio

Risco	Probabilidade	Impacto	Mitigação
Baixa adesão de usuários	Média	Alto	Marketing direcionado, parcerias com nutricionistas
Fornecedores não aderirem	Média	Médio	Comissões atrativas, onboarding simplificado
Nutricionistas preferindo plataformas existentes	Alta	Médio	Diferenciação com IA, integração completa
Concorrência de apps consolidados	Alta	Alto	Foco em nicho, funcionalidades únicas (grupo familiar, IA)

15.3 Riscos Regulatórios

Risco	Probabilidade	Impacto	Mitigação
Mudanças na LGPD	Baixa	Médio	Acompanhamento legislativo, flexibilidade no sistema
Regulamentação de marketplace de saúde	Baixa	Alto	Consultoria jurídica, compliance desde o início
Exigências do CFN sobre prescrição online	Média	Médio	Validação com órgão regulador, termos claros

16. Próximos Passos

16.1 Validação de Requisitos

1. Reunião com Stakeholders

- Apresentar documento

- Coletar feedback
- Validar prioridades

2. Consulta com Nutricionistas

- Validar fluxos de dieta
- Confirmar informações nutricionais necessárias
- Ajustar terminologia

3. Pesquisa com Usuários Potenciais

- Entrevistas qualitativas
- Testes de usabilidade de wireframes
- Validação de proposta de valor

16.2 Detalhamento Técnico

1. Modelagem de Dados Completa

- Diagrama ER detalhado
- Definição de todos os campos
- Relacionamentos e cardinalidades
- **Script de criação do banco de dados (próximo passo)**

2. Arquitetura de Sistema

- Diagrama de componentes
- Fluxo de dados
- Integrações detalhadas

3. Especificação de APIs

- Endpoints REST
- Payloads
- Autenticação/autorização

16.3 Preparação para Desenvolvimento

1. Setup de Ambiente

- Repositório Git
- Projeto Supabase
- CI/CD pipeline

2. Design System

- Paleta de cores
- Tipografia
- Componentes UI
- Guia de estilo

3. Prototipação

- Protótipos de alta fidelidade (Figma)
- Testes de usabilidade
- Ajustes finais

17. Conclusão

Este documento de requisitos estabelece a base completa para o desenvolvimento do **Myfitwell**, um sistema abrangente de controle de dietas que conecta usuários, nutricionistas e fornecedores em uma plataforma integrada.

Diferenciais do Myfitwell:

- **Gestão completa:** Dieta + Estoque + Compras em um só lugar
- **Inteligência artificial:** Geração de dietas personalizadas
- **Grupo familiar:** Gestão compartilhada de estoque e compras
- **Marketplace integrado:** Conecta demanda (lista de compras) com oferta (fornecedores)
- **Evolução rastreável:** Acompanhamento científico de resultados
- **LGPD compliant:** Segurança e privacidade desde a concepção

Tecnologia moderna:

- **Supabase:** Backend robusto e escalável
- **Next.js:** Frontend performático e SEO-friendly
- **Mobile-first:** Experiência otimizada para smartphones
- **PWA:** Instalável como app nativo

O próximo passo é a **criação do script modularizado do banco de dados Supabase**, que será desenvolvido com base nesta especificação completa.

Status: Aguardando aprovação para desenvolvimento