

A3 - Nutrição

Nutrição: Sistema de Planejamento e Avaliação de Dietas

Sites de ideias: <https://numax.com.br/estudantes/>

Requisitos Funcionais

1. Cadastro de Usuários
 - Permitir o cadastro de alunos, professores e administradores.
 - Login e autenticação de usuários.
2. Gestão de Pacientes
 - Permitir que alunos cadastrem e editem pacientes.
 - Armazenar dados básicos (nome, idade, sexo, peso, altura, etc).
3. Registro de Avaliações Nutricionais
 - Permitir o registro de avaliações antropométricas (IMC, circunferências, dobras cutâneas, etc).
 - Permitir o registro de recordatório alimentar (alimentos consumidos, horários, quantidades).
 - Permitir o registro de exercícios físicos praticados.
 - Possibilitar o registro de exames laboratoriais relevantes.*
4. Montagem de Planos Alimentares
 - Permitir que alunos montem planos alimentares personalizados para cada paciente.
5. Acompanhamento e Evolução
 - Registrar consultas de acompanhamento e evolução do paciente fictício.
 - Gerar relatórios de progresso (gráficos de peso, IMC, consumo calórico, etc).
6. Área “blog”.
 - Postagens relacionadas a saúde, receitas, etc...
7. Relatórios
 - Geração de relatórios detalhados para alunos e professores (por paciente, por aluno, por turma).
8. Exportação de Dados
 - Permitir exportar relatórios em PDF ou Excel.

Requisitos Não Funcionais

1. Usabilidade
 - Interface intuitiva, responsiva e fácil de navegar.
 - Suporte a dispositivos móveis (mobile friendly).
2. Segurança
 - Criptografia de senhas e dados sensíveis.
3. Performance
 - Tempo de resposta rápido (preferencialmente < 2 segundos para principais operações).
 - Suporte a múltiplos acessos simultâneos sem degradação perceptível.
4. Confiabilidade
 - Sistema disponível pelo menos 99% do tempo (exceto manutenção).
 - Backup dos dados.
5. Escalabilidade*
 - Possibilidade de adicionar novos módulos (ex: integração com APIs de alimentos, módulo de receitas, etc).
6. Compatibilidade
 - Suporte aos principais navegadores (Chrome, Firefox, Edge, Safari).
7. Documentação
 - Manual de uso para usuários.
 - Documentação técnica para manutenção futura.
8. Manutenibilidade
 - Facilidade em atualizar e dar manutenção.

POSSÍVEIS Tecnologias utilizáveis:

- PWA do [Next.js](#) para instalação do aplicativo
- Backend: Node.js + TypeScript + Express
- Frontend: React + TypeScript + [Next.js](#) + TailwindCSS
- Hospedagem: Firebase (Firestore + Auth) + Vercel (Backend)

PRIMEIRAS APRESENTAÇÕES DO PROJETO

Conceituação e Proposta Inicial

Definição do Ciclo Escolhido

O ciclo de desenvolvimento adotado será o Scrum, sendo escolhido por permitir entregas incrementais, colaboração constante entre equipe e stakeholders, rápida adaptação a mudanças e foco em valor de negócio durante todo o desenvolvimento.

Proposta Inicial da Aplicação

Desenvolver um sistema web para planejamento e avaliação de dietas voltado à área de nutrição. O sistema permitirá:

- Acompanhamento do progresso dos pacientes
- Registro de avaliações nutricionais
- Montagem de planos alimentares personalizados
- Notificações e alertas
- Geração de relatórios e exportação de dados
- Interface responsiva e intuitiva
- Área de blog para postagens relacionadas à saúde e receitas

Funções Obrigatórias

1. Cadastro de Usuários
2. Login e autenticação segura
3. Gestão de pacientes (cadastro, edição e consulta)
4. Registro de avaliações nutricionais (IMC, circunferências, dobras cutâneas, registro alimentar, exercícios físicos)
5. Montagem de planos alimentares personalizados
6. Acompanhamento e evolução do paciente (consultas, gráficos de progresso)
7. Área de blog (postagens sobre saúde e receitas)
8. Relatórios detalhados por paciente; exportação em PDF/Excel
9. Notificações e alertas para usuários

Levantamento de Requisitos

Requisitos Funcionais

- Cadastro e autenticação de usuários
- Cadastro, edição e acompanhamento de pacientes
- Registro de avaliações nutricionais
- Montagem e edição de planos alimentares
- Registro de consumo alimentar e exercícios físicos
- Consultas de acompanhamento, relatórios e gráficos
- Área de blog para postagens
- Exportação de relatórios em PDF/Excel
- Notificações para usuários

Requisitos Não Funcionais

- Usabilidade: Interface intuitiva, responsiva, fácil navegação, suporte a dispositivos móveis
- Segurança: Criptografia de senhas e dados sensíveis, autenticação robusta
- Performance: Tempo de resposta inferior a 2 segundos para operações críticas, múltiplos acessos simultâneos sem degradação
- Confiabilidade: Disponibilidade mínima de 99% (exceto manutenção), backups regulares
- Escalabilidade: Facilidade para adicionar novos módulos (APIs externas, etc)
- Compatibilidade: Suporte aos principais navegadores (Chrome, Firefox, Edge, Safari)
- Documentação: Manual do usuário e documentação técnica
- Manutenibilidade: Facilidade em atualizar e dar manutenção

Requisitos de Qualidade

- Usabilidade: Facilidade de uso, fluxo intuitivo, interface responsiva e acessível em dispositivos móveis
- Segurança: Senhas e dados sensíveis criptografados, proteção contra ataques comuns, autenticação robusta
- Desempenho: Tempo de resposta < 2 segundos para operações críticas, suporte a múltiplos acessos simultâneos
- Confiabilidade: Disponibilidade mínima de 99%, backups automáticos, tolerância a falhas
- Manutenibilidade: Código modular, documentação clara, facilidade de atualização e testes automatizados

Plano de Garantia da Qualidade

Critérios de Aceitação

- Funcionalidades entregues conforme requisitos especificados
- Interface responsiva e intuitiva em todos os dispositivos e navegadores suportados

- Operações principais com tempo de resposta < 2 segundos
- Dados sensíveis protegidos por criptografia
- Cobertura de testes automatizados (>80%)
- Exportação de dados em PDF/Excel funcional
- Relatórios e gráficos exibindo dados corretos e atualizados

Políticas de Versionamento

- Utilização de Git para controle de versão
- Branch's principais: `main` (produção), `develop` (homologação), `feature/*`, `bugfix/*`, `hotfix/*`
- Pull Requests obrigatórios para merge em `main` ou `develop`, com revisão de código

Procedimentos de Revisão de Código

- Todo código novo ou alterado deve passar por revisão (Code Review) por pelo menos um membro da equipe
- Utilização de ferramentas automatizadas (linters, formatadores)
- Revisão inclui verificação de padrões de código, segurança, testes e documentação
- Correções de bugs e melhorias documentadas no histórico do projeto

Estratégias de Testes

- Testes Unitários: Cobertura das principais funções e módulos
- Testes de Integração: Validação do funcionamento entre componentes (frontend, backend, banco)
- Testes de Interface (E2E): Simulação dos fluxos principais do sistema
- Testes de Segurança: Validação da criptografia, autenticação e proteção contra vulnerabilidades
- Testes de Performance: Medição de tempo de resposta e carga
- Testes de Compatibilidade: Execução em múltiplos navegadores e dispositivos

Tecnologias Propostas

- Frontend: React + TypeScript + Next.js + TailwindCSS (PWA)
- Backend: Node.js + TypeScript + Express
- Hospedagem: Firebase (Firestore + Auth), Vercel (Backend)

Entrega para dia 15/10

Modelo de Processo Escolhido

O ciclo de desenvolvimento adotado será o Scrum, sendo escolhido por permitir entregas incrementais, colaboração constante entre equipe e stakeholders, rápida adaptação a mudanças e foco em valor de negócio durante todo o desenvolvimento.