**Documentação Técnica do Projeto "GastroNet"**

**1. Introdução**

Este documento detalha os aspectos técnicos e a metodologia de desenvolvimento do projeto **GastroNet**, uma aplicação web interativa focada em boas práticas de segurança alimentar. O projeto foi concebido para transformar procedimentos operacionais padronizados (POPs) em uma ferramenta digital acessível e educativa, utilizando tecnologias modernas e uma abordagem de desenvolvimento ágil e incremental.

**2. Metodologia de Desenvolvimento**

A metodologia de desenvolvimento adotada seguiu um modelo **híbrido**, combinando elementos de **Design Thinking** para a concepção da solução e um fluxo de trabalho **ágil** para a implementação e entrega contínua.

* **Fase de Concepção (Design Thinking):**
  + **Imersão:** Análise do problema central — a dificuldade de acesso e aplicação das normas de segurança alimentar.
  + **Ideação:** Brainstorming para a criação de uma solução digital que fosse interativa, responsiva e de fácil acesso. A ideia de uma plataforma web com módulos de POPs foi a escolhida.
  + **Prototipagem:** Desenvolvimento de protótipos de baixa fidelidade para definir o layout da página inicial e a estrutura dos módulos.
  + **Testes:** Validação da usabilidade e das funcionalidades básicas com base no feedback recebido, permitindo refinar o design e o fluxo de navegação.
* **Fase de Implementação (Desenvolvimento Ágil):**
  + **Iterações e Sprints:** O desenvolvimento foi dividido em etapas menores. A cada "sprint" (período de trabalho), novos módulos de POPs eram criados e integrados à aplicação.
  + **Refinamento Contínuo:** O feedback foi incorporado ao longo do desenvolvimento, resultando em ajustes como a otimização do layout da página inicial e a correção de bugs.
  + **Integração Contínua (CI):** O uso de ferramentas de automação como o npm run build e o git push garantiu que as versões atualizadas do projeto fossem publicadas de forma automática no GitHub Pages, facilitando a visualização e o teste.

**3. Ferramentas e Tecnologias**

As seguintes ferramentas e tecnologias foram selecionadas para garantir que o projeto fosse robusto, escalável e de fácil manutenção:

* **React:** Biblioteca JavaScript utilizada para a construção da interface do usuário (UI). A escolha do React permitiu a criação de componentes reutilizáveis para cada módulo, tornando o desenvolvimento mais eficiente e organizado.
* **Vite:** Uma ferramenta de build e servidor de desenvolvimento de última geração. O Vite foi fundamental para otimizar o tempo de desenvolvimento, pois oferece um carregamento de módulo instantâneo e um Hot Module Replacement (HMR) ultrarrápido.
* **CSS Modules:** Metodologia de estilização que garante que os estilos CSS de um componente não interfiram em outros. Isso foi essencial para evitar conflitos de estilo e manter o código organizado à medida que novos componentes eram adicionados.
* **Git e GitHub:** Sistema de controle de versão e plataforma de hospedagem de código. O Git permitiu o versionamento do projeto e o trabalho colaborativo, enquanto o GitHub serviu como repositório remoto e, mais importante, como plataforma de hospedagem gratuita através do **GitHub Pages**.
* **GitHub Pages:** Serviço de hospedagem de sites estáticos diretamente a partir de um repositório GitHub. O uso do Pages foi a solução ideal para publicar a aplicação de forma simples e sem custos adicionais, permitindo que ela fosse acessada publicamente.