

## ***Proposta***

O sistema vai ser construído usando Flask como framework em Python no back-end, Firebase para armazenamento de dados e autenticação, e HTML/CSS/JavaScript no front-end.

## ***Tecnologias usadas***

### **Python com Flask**

- Cria aplicações web rapidamente
- Tem uma sintaxe mais clara
- Tem templates prontos pra uso

### **Firebase Firestore**

- Não precisa configurar um servidor de banco de dados
- Oferece armazenamento em nuvem sem complicações
- Funciona bem com aplicações web

## ***Organização do Código***

Vai ser organizado de acordo com os requisitos das pastas em python (tipo a pasta static, template, etc)

## ***PLANO DE IMPLEMENTAÇÃO***

### ***Funcionalidades Básicas***

- Sistema de cadastro e login
- Páginas principais de navegação
- Registro de treinos e exercícios
- Controle de medidas corporais

### ***Funcionalidades Completas***

- Sistema de alimentação
- Controle de água e sono
- Definição de objetivos
- Relatórios e estatísticas

### ***Melhorias***

- Análise de dados
- Interface mais elaborada
- Funcionalidades extras

## ***ESTRUTURA DE DADOS***

### ***Collections no Firebase***

- users: dados dos usuários
- exercicios: catálogo de exercícios
- alimentos: catálogo de alimentos
- Dados dos usuários armazenados por ID único

### ***Vantagens dessa Estrutura***

- Fácil de entender e manter
- Buscas rápidas por usuário
- Dados compartilhados entre usuários
- Expansão simples para novas funcionalidades

### ***Segurança***

### ***Proteções Implementadas***

- Verificação de login em todas as páginas
- Cada usuário só acessa seus dados
- Validação de dados de entrada
- Senhas protegidas pelo Firebase

### ***Regras de Acesso***

- Apenas usuários logados podem acessar o sistema
- Usuários só visualizam e editam seus próprios dados
- Catálogos compartilhados têm controle de edição

### ***Por que Flask?***

O Flask foi escolhido por ser relativamente fácil de mexer, com bons templates e funções. Diferente de frameworks mais complexos, Flask permite entender conceitos web básicos sem muito problema, e é bom para projetos pequenos como este.

### ***Por que esta Arquitetura?***

A arquitetura proposta equilibra simplicidade e funcionalidade, ela permite criar uma aplicação completa sem conhecimentos avançados, enquanto segue práticas modernas de desenvolvimento web.

### ***Viabilidade do Projeto***

#### ***Viabilidade Técnica***

Todas as tecnologias escolhidas são gratuitas para nosso uso e possuem boa documentação para procurar e entender. Não precisam de hardware especializado ou licenças caras.

#### ***Viabilidade de Aprendizado***

O projeto introduz conceitos importantes como:

- Desenvolvimento web básico
- Armazenamento de dados
- Autenticação de usuários
- Interface com o usuário

#### ***Viabilidade de Manutenção***

O código organizado e documentado facilita correções e melhorias futuras, e a estrutura simples permite que outras pessoas entendam e continuem o projeto.