

# Explicação do Projeto Flask com Tema de Futebol

Este projeto foi desenvolvido utilizando o framework Flask, uma ferramenta Python que permite criar aplicações web de maneira eficiente e organizada. A aplicação consiste em três páginas interligadas: uma página inicial, uma página de times e uma página de jogadores.

## Funcionamento Técnico

No Flask, as rotas (`@app.route`) definem os endereços (URLs) que podem ser acessados no navegador. Quando um usuário acessa uma dessas rotas, o Flask executa a função associada e retorna um HTML renderizado através do `render_template`.

A navegação entre páginas é feita através de links no menu, que direcionam o usuário para as diferentes seções do site. Cada clique em um link resulta em uma nova requisição ao servidor Flask, que processa a rota solicitada e retorna o HTML correspondente.

## Estrutura do Projeto

### 1. Arquivo *app.py*

- É o núcleo da aplicação Flask
- Define as rotas principais:
  - `@app.route("/")` → Página inicial
  - `@app.route("/times")` → Página de times
  - `@app.route("/jogadores")` → Página de jogadores
- Cada função retorna o template HTML correspondente
- O `app.run(debug=True)` inicia o servidor local para desenvolvimento

### 2. Sistema de Templates

- Utiliza o Jinja2, mecanismo de templates do Flask
- O `base.html` define a estrutura comum a todas as páginas (cabeçalho, navegação, rodapé)
- As páginas individuais herdam essa estrutura com `{% extends "base.html" %}`
- O conteúdo específico de cada página é inserido através de `{% block content %}`

### 3. Estilização com CSS

- O arquivo `style.css` define a identidade visual com tema de futebol
- Cores verdes em diferentes tonalidades remetem a um campo de futebol
- Elementos visuais como listas de times e cartões de jogadores
- Design responsivo com sombras e transições suaves

## Fluxo de Navegação

- Ao acessar `http://localhost:5000/`, o Flask executa a função `index()` e retorna a página inicial

- Ao clicar em 'Times' no menu, o usuário é direcionado para /times, acionando a função times() que renderiza o template correspondente
- Da mesma forma, o link 'Jogadores' direciona para /jogadores, executando a função jogadores()

## **Aprendizados Principais**

- Compreendi como o Flask atua como controlador central da aplicação
- Entendi a importância dos templates base para evitar repetição de código
- Aprendi como o CSS externo mantém a consistência visual em todas as páginas
- Percebi como a navegação entre páginas no Flask funciona através do sistema de rotas e templates
- Notei a vantagem de separar claramente a lógica da aplicação (app.py) da apresentação (templates)