

# Sistema Passa a Bola

**Repositório GitHub:** [https://github.com/SmoothPath/Python\\_PassaBola.git](https://github.com/SmoothPath/Python_PassaBola.git)

## Equipe SmoothPath

Gabriel dos Santos Cardoso - 562103

Geovana Maria da Silva Cardoso - 566254

Gustavo Torres Caldeira - 561613

Lucas Oliveira Santos - 563613

Mariana Silva do Egito Moreira – 562544

1ESPF – Engenharia de Software (FIAP)

## Visão Geral

O **Passa a Bola** é um sistema local de gerenciamento de usuárias (jogadoras), eventos esportivos e times, com armazenamento persistente em arquivos JSON.

Esse sistema inclui uma funcionalidade de ranking que contabiliza a participação das jogadoras e envia os resultados automaticamente para o site desenvolvido em Node.js + React.

## Funcionalidades Principais

### 1. Gerenciamento de Usuários

- Cadastro de novas jogadoras com validação de e-mail e verificação de duplicidade.
- Login de jogadoras e administradores (com credenciais padrão [admin@passabola.com](mailto:admin@passabola.com) / admin123).
- Armazenamento persistente das informações em `usuarios.json`.

### 2. Gerenciamento de Eventos

- Criação, listagem, atualização e exclusão de eventos esportivos, incluindo nome, local, data e limite padrão de jogadoras por time.
- Cada evento possui uma lista de inscritas e times associados.
- Armazenamento em `eventos.json`.

## CRUD Completo:

- **Editar Usuária:** altera nome ou senha mantendo o mesmo e-mail.
- **Excluir Usuária:** remove completamente a jogadora e suas inscrições.
- **Editar Evento:** permite atualizar nome, local ou data.
- **Excluir Evento:** apaga o evento e os times associados.
- **Listar eventos:** exibe todos os eventos cadastrados com número de inscritas.
- **Cadastrar evento:** permite ao administrador criar novos eventos.
- **Inscriver-se em evento:** permite que a jogadora se inscreva e atualiza automaticamente o ranking.

## 3. Persistência de Dados

O sistema utiliza arquivos JSON para simular um banco de dados local:

- `usuarios.json` – cadastro de jogadoras
- `eventos.json` – lista de eventos e inscrições
- `times.json` – armazenamento dos times formados automaticamente
- `ranking.json` – ranking de jogadoras mais ativas

### Funções principais:

- `carregar_dados()` – lê o conteúdo de um arquivo JSON existente
- `salvar_dados()` – grava os dados no formato JSON com identificação e codificação UTF-8

## 4. Ranking de Jogadoras Mais Ativas

- O ranking é gerado automaticamente sempre que uma jogadora se inscreve em um evento.
- A contagem considera quantos eventos cada jogadora participou.
- Os dados são salvos em `ranking.json` e também enviados via HTTP POST para a API do site:  
<http://localhost:5000/api/ranking>

Caso a API esteja fora do ar, o sistema exibe aviso sem interromper a execução.

O ranking também pode ser visualizado localmente pelo menu.

### Funções principais:

- `gerar_ranking()` – recalcula o ranking com base nas inscrições

- `mostrar_ranking()` – exibe o ranking local ordenado por número de participações

## 5. Formação Automática de Times

- O sistema cria times automaticamente conforme o número de jogadoras definido em cada evento.
- Cada time é formado sequencialmente (as primeiras N jogadoras formam o Time 1, as próximas N o Time 2, etc.).
- Caso o número de inscritas não seja múltiplo do limite, o último time terá menos jogadoras.
- As informações são armazenadas em `times.json`.

### Função principal:

`formar_times(evento)` – percorre a lista de inscritas e cria os grupos automaticamente.

## 6. Menus e Fluxo de Navegação

### Menu da Jogadora:

1. Listar eventos
2. Inscrever-se em evento
3. Ver ranking
4. Logout

### Menu do Administrador:

1. Cadastrar usuário
2. Cadastrar evento
3. Listar eventos
4. Gerar ranking
5. Formar times
6. Logout

### Menu Principal:

1. Cadastrar usuário
2. Login
3. Sair

## 7. Integração com o Site (Node.js + React)

O script Python se comunica com o backend via API REST local, permitindo que o ranking atualizado seja exibido na interface web.

A integração ocorre automaticamente na geração do ranking, por meio de uma requisição:

```
requests.post("http://localhost:5000/api/ranking", json=ranking)
```

## Estrutura de Arquivos

```
passa_a_bola/  
├── main.py  
├── usuarios.json  
├── eventos.json  
├── times.json  
├── ranking.json  
├── contaAdm.txt  
├── membros.txt  
├── README.md  
└── docPython.pdf
```

## Execução

Execute o programa diretamente com:

```
python main.py
```

O sistema abrirá o menu principal, permitindo cadastro, login e navegação pelos recursos de jogadora ou administrador.

## Conclusão

A solução **Passa a Bola** oferece uma plataforma completa de gestão esportiva e engajamento de jogadoras, unindo:

- Sistema local em Python com persistência em JSON;
- Plataforma web em Node.js e React, para exibição e administração dos dados;

- Dispositivo IoT Cinta Monitora, responsável por coletar dados fisiológicos de atletas em tempo real e enviar para o ecossistema FIWARE.

Assim, o conjunto forma um sistema integrado que abrange organização, segurança e monitoramento esportivo, conectando pessoas e tecnologia sob o mesmo propósito.