

SPRINT 3 PASSA A BOLA



por Smoothpath

DETALHES DO ESCOPO EDGE

Objetivo Geral: Cinta Monitora - Passa a Bola

Desenvolver um dispositivo vestível inteligente, integrado a uma arquitetura IoT baseada no FIWARE, para monitoramento fisiológico de atletas em tempo real, garantindo segurança, prevenindo riscos de exaustão e oferecendo insights sobre desempenho esportivo.

Objetivos Específicos

Monitorar batimentos cardíacos simulados por potenciômetro.

Monitorar temperatura corporal simulada por sensor DHT22.

Calcular calorias estimadas a partir dos dados coletados.

Emitir alerta sonoro local (buzzer) quando parâmetros estiverem fora da faixa segura.

Integrar o dispositivo ao FIWARE Orion Context Broker, permitindo consulta em tempo real.

Registrar históricos de dados fisiológicos para análise posterior.

Validar funcionamento em simulação (Wokwi) e em integração com serviços via Postman.

1. No Dispositivo (ESP32 - Cinta)

Leitura contínua dos sensores (simulados pelo potenciômetro e DHT22).

Processamento local para cálculo de calorias gastas.

Ativação do buzzer em condições críticas.

Publicação periódica dos dados no Broker MQTT.

2. Na Camada IoT (FIWARE)

Mosquitto (MQTT Broker): recebe os dados do ESP32.

IoT Agent MQTT: traduz as mensagens em entidades NGSI.

Orion Context Broker: mantém o estado atual das atletas.

MongoDB Internal: banco interno do Orion para persistência básica.

STH-Comet + MongoDB Historical: registro de séries temporais para análise da evolução dos dados.

3. Ferramentas de Validação

Postman: para criar entidades, consultar estados e históricos.

Wokwi: para simular ESP32, sensores e buzzer antes da execução real.

Escopo Não Funcional

Confiabilidade: sistema deve detectar e registrar corretamente valores anormais de temperatura e batimentos.

Escalabilidade: permitir que múltiplos dispositivos sejam monitorados simultaneamente.

Segurança: comunicação via MQTT deve permitir autenticação futura.

Usabilidade: dados devem estar disponíveis de forma clara no Orion e consultáveis via API.

Restrições

Utilização de ESP32 como microcontrolador principal.

Uso do potenciômetro apenas como simulação dos batimentos.

Implementação inicial focada em alerta local sonoro, sem notificações móveis nativas.

Infraestrutura do FIWARE deve rodar em VM Linux com Docker.

Entregas Previstas

Código-fonte documentado para ESP32 (Arduino IDE).

Arquitetura IoT implantada com FIWARE Orion, IoT Agent, MongoDB, STH-Comet.

Coleção Postman com endpoints configurados.

Relatório técnico com resultados de simulação (Wokwi) e integração real.

Benefícios Esperados

Segurança das atletas: prevenção de riscos por sobrecarga física.

Monitoramento contínuo: dados em tempo real acessíveis via FIWARE.

Histórico de desempenho: consultas sobre evolução dos parâmetros.

Base para expansão: possibilidade de adicionar novos sensores ou integrar dashboards de BI.

DETALHES DO ESCOPO PYTHON

Objetivo Geral

Desenvolver um sistema em Python para gerenciar usuários, eventos e times esportivos, com interface em modo texto, aplicando boas práticas de programação e conceitos fundamentais da linguagem.

Objetivos Específicos

Criar um menu principal interativo com opções claras.

Permitir cadastro de usuários (nome, e-mail e senha).

Implementar login de usuário com verificação de credenciais.

Possibilitar a criação de eventos esportivos (tipo de jogo e número de jogadores por time).

Cadastrar times com jogadores, respeitando o limite definido pelo evento.

Exibir relatórios organizados (usuários, eventos, times, partidas).

Garantir que o sistema permaneça em execução até a opção “sair”.

Escopo Funcional

Entrada, Processamento e Saída

Entrada: dados fornecidos pelo usuário (cadastro, login, criação de eventos, etc.).

Processamento: validação, armazenamento em listas/dicionários, regras de limite de jogadores.

Saída: mensagens claras no console (cadastro concluído, login válido/inválido, relatórios formatados).

1. Estruturas de Decisão e Repetição

if/else: validações de login, verificações de limite de jogadores.

while: manter menu ativo até que o usuário escolha sair.

for: percorrer listas de usuários, times e eventos para relatórios.

2. Listas e Dicionários

Lista de usuários (cada usuário como dicionário com nome, e-mail e senha).

Lista de eventos (cada evento como dicionário com tipo e limite de jogadores).

Lista de times (cada time como dicionário com nome e lista de jogadores).

3. Funções

`menu_principal()`

`cadastrar_usuario()`

`login_usuario()`

`criar_evento()`

`cadastrar_time()`

`mostrar_relatorios()`

`main()` para orquestrar a execução.

4. Interface Intuitiva

Textos explicativos em cada etapa.

Confirmações de sucesso ou erro.

Estrutura clara de menus e submenus.

5. Boas Práticas

Nomes de variáveis e funções descritivos.

Comentários explicativos no código.

Indentação e organização consistentes.

Separação de lógica em funções.

Escopo Não Funcional

Código deve ser escrito em Python 3+.

Interface somente em modo texto (console).

Simplicidade e clareza priorizadas sobre sofisticação gráfica.

Estrutura de arquivos organizada:

arquivos .py do código-fonte.

Restrições

Não utilizar banco de dados (apenas listas e dicionários em memória).

Não utilizar bibliotecas externas além das nativas do Python.

Armazenamento dos dados não será persistente (dados são perdidos ao encerrar).

Entregas Previstas

Código-fonte Python com todos os módulos funcionais.

Arquivo integrantes.txt com nome e RM da equipe.

Documento PDF com descrição das funcionalidades implementadas.

Benefícios do Sistema

Prática dos principais conceitos de Python (estruturas de repetição, decisão, listas, dicionários e funções).

Aplicação de boas práticas de programação.

Exercício prático de organização de projeto e documentação.

Sistema simples, mas expansível (pode evoluir para incluir partidas, placares, ranking, etc.).

DETALHES DO ESCOPO WEB

Objetivo Geral

Desenvolver uma aplicação web responsiva e moderna, unindo gestão de atletas, eventos, pontos, e-commerce, voluntariado e doações, com frontend em React + TailwindCSS e backend em Node.js + Express, integrando APIs externas e oferecendo navegação intuitiva.

Objetivos Específicos

- ✓ Criar sistema de cadastro e login seguro (JWT).
- ✓ Implementar sistema de pontos gamificado (“Joga Junto”), incluindo possibilidade de apostar pontos em jogos.
- ✓ Integrar e-commerce (“Camisa 10”) para venda de produtos esportivos.
- ✓ Disponibilizar plataforma de voluntariado (“Voluntários”), com formulário completo e benefícios claros.
- ✓ Oferecer módulo de doações online.
- ✓ Criar painel administrativo para gestão de vendas, doações, inscrições, jogos e voluntários.
- ✓ Integrar API externa de notícias esportivas.
- ✓ Garantir relatórios, responsividade e testes em camadas críticas.

1. Escopo Funcional

1.1 Frontend (React + TailwindCSS)

Layout Base: Navbar, Footer, rotas principais.

Responsividade: desktop, tablet, mobile.

Páginas:

Início:

Botões de Login e Cadastrar.

Seção “História do Passa a Bola”.

Notícias esportivas (API externa) + postagens próprias.

Cadastro/Login (usuário padrão).

Quiz de Boas-Vindas: classificação como Atleta, Técnica ou Olheira.

Painel da Jogadora: foto, nome, pontos, botões de navegação.

Meus Eventos: listagem, detalhes, QR Code para check-in.

Joga Junto (Sistema de Pontos):

Exibir pontos acumulados.

Opção de apostar pontos em jogos/eventos.

Ranking de jogadoras.

Camisa 10 (E-commerce):

Listagem de produtos esportivos.

Carrinho de compras.

Pagamento simulado.

Voluntários:

Chamada principal com título impactante.

Breve explicação do propósito.

Formulário completo: nome, email, telefone, idade, cidade/bairro, disponibilidade, áreas de interesse (checklist), upload de documento/foto/currículo.

Benefícios (certificado, camisa, descontos, networking).

Botão de ação: “Quero ser voluntário 🚀”.

Extras: depoimentos, fotos de eventos, contador de voluntários ativos.

Doações:

Explicação do impacto das doações.

Opções de valor fixo (R\$10, R\$20, R\$50, R\$100).

Pagamento simulado (cartão/pix fake).

Mensagem de agradecimento e recibo.

Painel Administrativo:

Dashboard com visão geral.

Seções: vendas, doações, inscrições de voluntários, jogos, eventos e mapa dos campos.

1.2 Backend (Node.js + Express)

Autenticação: rotas /login e /register com JWT.

Quiz: rota /quiz para salvar tipo de usuária.

Usuário: rota /user/:id para retornar dados da jogadora.

Eventos:

/events → listar eventos disponíveis.

/events/:id → detalhes do evento.

/events/:id/checkin → geração/validação de QR Code.

Sistema de Pontos (Joga Junto):

/points/:id → consultar pontos de uma usuária.

/points/:id/bet → apostar pontos em jogos.

/ranking → ranking das jogadoras.

E-commerce (Camisa 10):

/products → listar produtos.

/cart → gerenciar carrinho.

/checkout → simulação de compra.

Voluntários:

/volunteers → listar voluntários cadastrados.

/volunteers/register → registrar novo voluntário.

Doações:

/donations → listar doações realizadas.

/donations/new → registrar nova doação.

Admin:

/admin/dashboard → resumo de vendas, doações, inscrições.

/admin/events → gerenciar eventos.

/admin/volunteers → gerenciar voluntários.

/admin/donations → gerenciar doações.

/admin/products → gerenciar produtos.

Notícias externas:

/news → integração com API pública de esportes.

2. Escopo Não Funcional

HTML semântico e acessível.

Responsividade garantida com TailwindCSS.

Segurança: JWT para autenticação e controle de acesso.

Performance: dados mockados em arrays no backend.

Escalabilidade: fácil migração para banco de dados real.

3. Restrições

Persistência inicial em memória (sem DB real).
Sistema de pontos, apostas, QR Code e e-commerce simulados.
Integração de doações sem gateway real (somente simulação).

4. Entregas Previstas

Repositório GitHub (monorepo).
Frontend React + TailwindCSS completo.
Backend Node.js + Express com API REST.

Dados de teste:

Usuário padrão (teste@passabola.com / 123@Teste).
Lista de eventos e produtos mockados.
Ranking inicial de jogadoras.
Voluntários de exemplo.
Documentação da API (PDF com endpoints, parâmetros, exemplos request/response).
Relatório de testes críticos.
7. Testes Críticos
Login e cadastro.

Autorização JWT.

Criação e check-in em eventos (QR Code).
Apostas de pontos em jogos.
Fluxo de compra no e-commerce.
Cadastro de voluntários.
Registro de doações.
Dashboard administrativo.
API de notícias funcionando.

5. Benefícios Esperados

Engajamento esportivo com gamificação de pontos.
E-commerce esportivo integrado (geração de receita).
Plataforma de voluntariado para apoiar atletas e eventos.
Canal de doações para sustentabilidade do projeto.
Painel administrativo centralizado para gestão estratégica

BACKLOG FINAL

Prioridade	Área	Funcionalidade	Descrição
P0 - ESSENCIAL MVP	IoT (Cinta)	Leitura de sensores (potenciômetro e DHT22)	Captura contínua de batimentos e temperatura simulados.
P0	IoT (Cinta)	Cálculo de calorias	Processamento local com base nos dados coletados.
P0	IoT (Cinta)	Alerta sonoro (buzzer)	Disparo quando parâmetros saírem da faixa segura.
P0	IoT(FIWARE)	Publicação MQTT → Orion Context Broker	Envio periódico de dados para consulta online.
P0	Python(Console)	Menu principal interativo	Estrutura de navegação para usuários, eventos e times.
P0	Python (Console)	Cadastro e login de usuários	Armazenamento em listas/dicionários (sem DB).
P0	Web (Backend)	Autenticação JWT	Rotas /login e /register seguras.
P0	Web (Frontend)	Páginas Home, Login e Registro	Layout básico com React + TailwindCSS.
P0	Web (Backend)	Listagem de eventos	Rota /events com dados mockados.

BACKLOG FINAL

Prioridade	Área	Funcionalidade	Descrição
P1	IoT (FIWARE)	Histórico de dados (STH-Comet + MongoDB)	Registro de séries temporais para análise.
P1	Python (Console)	Criação de eventos esportivos	Definição de tipo e limite de jogadores.
P1	Python (Console)	Cadastro de times e jogadores	Respeitando limite do evento.
P1	Web (Backend)	Sistema de pontos (Joga Junto)	Rotas /points, /rank e /points/add.
P1	Web (Frontend)	Dashboard inicial	Exibição de eventos, pontos e ranking.
P1	Web (Backend)	Integração com API de notícias esportivas	Exibir feed na Home.

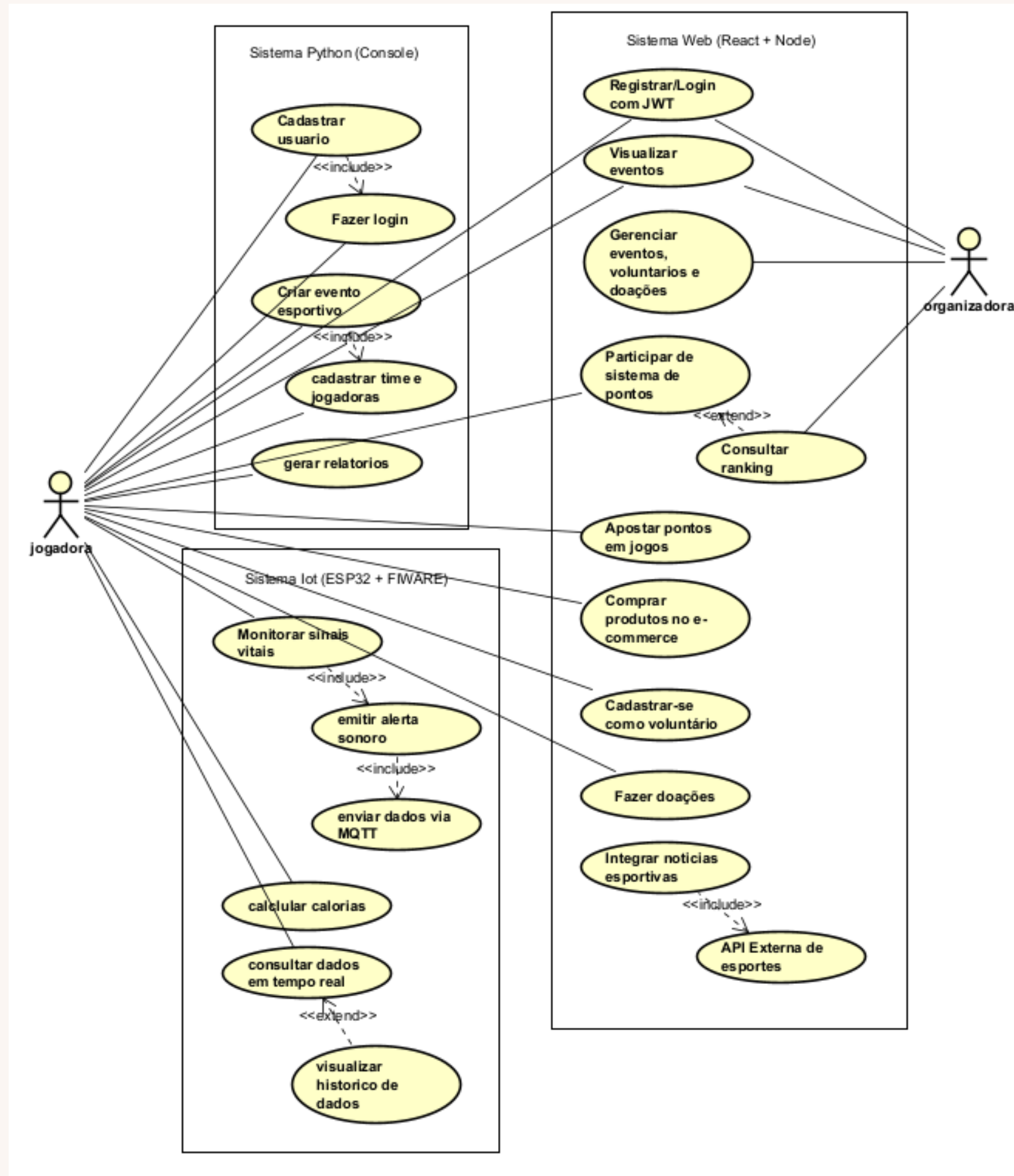
BACKLOG FINAL

Prioridade	Área	Funcionalidade	Descrição
P2 - Valor agregado / Incremento 2	IoT (FIWARE)	Integração com Postman e Webhook	Testes e simulação antes da execução real.
P2	Python (Console)	Relatórios organizados	Listagem de eventos, usuários, times e partidas.
P2	Web (Frontend)	E-commerce (Camisa 10)	Listagem de produtos e carrinho mockado.
P2	Web (Backend)	Rotas de e-commerce	/products, /cart, /products/id.
P2	Web (Frontend)	Módulo de voluntariado	Formulário e listagem de voluntários.
P2	Web (Backend)	Rotas de voluntariado	/volunteers, /volunteers/register.
P2	Web (Frontend)	Módulo de doações	Pix/cartão (simulado) e valores fixos.
P2	Web (Backend)	Rotas de doações	/donations, /donations/register.

BACKLOG FINAL

Prioridade	Área	Funcionalidade	Descrição
P3 - Expansão / Incremento 3	IoT	Autenticação MQTT	Segurança na comunicação.
P3	Python (Console)	Persistência de dados	Evoluir para banco de dados real.
P3	Web (Frontend)	Gamificação avançada	Apostas de pontos em jogos.
P3	Web (Backend)	QR Code para eventos	Rotas /events/qrcode para check-in.
P3	Web (Frontend)	Ranking visual e interativo	Top 10 atletas e pontuação.
P3	Web (Frontend)	Painel administrativo completo	Gestão de vendas, doações, voluntários e eventos.

DIAGRAMA DE CASO DE USO



Integrantes do Grupo e RM

Gabriel dos Santos Cardoso – 562103

Geovana Maria da Silva Castro – 566254

Gustavo Torres Caldeira – 561613

Lucas Oliveira Santos – 563613

Mariana Silva do Egito Moreira – 562544