

Global Solution 2 - 2026

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO
PAULISTA

Grupo 79 - Turing

Andre Ribeiro Leli - RM97780

Daniel Alexandre Barcellos de Brito - RM98185

Marcone Santos Ribeiro - RM552585

Murillo Barbosa Lemos - RM550445

LINK VIDEO PITCH:

<https://youtu.be/inserir>

holo.work

Visão

um espaço de trabalho híbrido leve (2d/pixel), com presença, chat e quadros de tarefas, que integra leituras iot (simuladas) e um agente de recomendação para ritmo saudável e pairing. web/mobile responsivo, cloudflare-first, dados em d1, eventos em kv. sem framework pesado.

Problema

trabalho distribuído fragmenta sinal: status, foco, conforto, saúde. reuniões viram ruído. precisamos de um hub que enxergue contexto e sugira ações práticas sem fricção.

Objetivos

- reduzir atrito de colaboração remota.
- dar visibilidade de conforto/ambiente via iot simulado.
- promover pausas e pairing guiados por dados.
- mvp executável em 1 click com wrangler.

Escopo (v1)

- espaços (rooms) com mapa 2d leve (tiles/png), presença, chat, tasks.
- check-in de humor/energia (1..5) e status (focus, solo, pair, afk).
- ingest de iot: temperatura, ruído, luminosidade (mock) por device.
- agente: recomenda pausas, pairing e sala mais “confortável”.
- auth: e-mail + senha com hash; roles: admin, user.
- admin web local (sem login externo) para CRUD rápido de rooms/devices.

fora do escopo v1: voz, vídeo, 3d, calendário integrado, oauth.

Personas

- dev/design: quer foco, pairing rápido, status claro.
- gestor: quer visibilidade de conforto e carga do time.
- ops: quer quedas/alertas simples de ambiente.

Fluxos principais

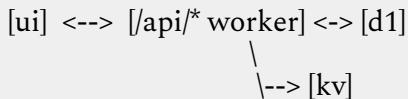
(ver imagem em [#Imagens → Fluxo Principal](#fluxo-principal))

1. login -> escolher room -> ver mapa/presenças -> chat/tasks -> check-in -> agente sugere ação.
2. admin local gera room, device e seeds de mapa.
3. iot_sim envia POST periódico em /api/iot/ingest.

Arquitetura

(ver imagem em [#Imagens → Arquitetura](#arquitetura))

- ui: html+css+js (es modules), render 2d canvas/pixel; pwa básico.
- api: cloudflare worker (ts) com rotas rest, jwt, rate-limit kv.
- dados: d1 (sqlite). kv: sessions, rate, agent-cache, stream buffers.
- iot: script python envia leituras; worker valida device_secret.
- agente: função no worker que lê leituras+check-ins e gera recomendações simples (regras + score).



[iot_sim] -> [/api/iot/ingest]

Modelo de Dados (mínimo)

- users(id, email, pass_hash, role, created_at)
- rooms(id, name, slug, map_seed, created_at)
- room_members(user_id, room_id, joined_at)

- messages(id, room_id, user_id, body, created_at)
- tasks(id, room_id, title, status, assignee_id, created_at)
- checkins(id, user_id, room_id, mood, energy, status, created_at)
- devices(id, room_id, name, kind, secret, created_at)
- readings(id, device_id, t_c, noise_db, lux, created_at)
- suggestions(id, room_id, user_id, kind, text, created_at)

SQL (d1/sqlite)

Códigos SQL

```

create table users (
    id text primary key,
    email text unique not null,
    pass_hash text not null,
    role text not null default 'user',
    created_at integer not null
);
create table rooms (
    id text primary key,
    name text not null,
    slug text unique not null,
    map_seed text,
    created_at integer not null
);
create table room_members (
    user_id text not null,
    room_id text not null,
    joined_at integer not null,
    primary key(user_id, room_id)
);
create table messages (
    id text primary key,
    room_id text not null,
    user_id text not null,
    body text not null,
    created_at integer not null
);
create table tasks (
    id text primary key,
    room_id text not null,

```

```
title text not null,
status text not null default 'todo',
assignee_id text,
created_at integer not null
);
create table checkins (
    id text primary key,
    user_id text not null,
    room_id text not null,
    mood integer,
    energy integer,
    status text, -- focus|solo|pair|afk
    created_at integer not null
);
create table devices (
    id text primary key,
    room_id text not null,
    name text not null,
    kind text not null, -- env
    secret text not null,
    created_at integer not null
);
create table readings (
    id text primary key,
    device_id text not null,
    t_c real,
    noise_db real,
    lux real,
    created_at integer not null
);
create table suggestions (
    id text primary key,
    room_id text not null,
    user_id text,
    kind text not null, -- pause|pair|move
    text text not null,
    created_at integer not null
);
```

API (rest)

Endpoints

(ver imagem em [#Imagens → Endpoints](#endpoints))

- auth
 - POST /api/auth/register {email, password}
 - POST /api/auth/login {email, password} -> {jwt}
 - GET /api/auth/me -> user
- rooms
 - GET /api/rooms
 - POST /api/rooms (admin)
 - GET /api/rooms/:id
 - POST /api/rooms/:id/join
- messages
 - GET /api/rooms/:id/messages?cursor=...
 - POST /api/rooms/:id/messages {body}
- tasks
 - GET/POST /api/rooms/:id/tasks
 - PUT /api/tasks/:id {status, assignee_id}
- checkins
 - POST /api/rooms/:id/checkins {mood, energy, status}
 - GET /api/rooms/:id/checkins?since=ts
- devices/iot
 - POST /api/devices (admin) -> {id, secret}
 - POST /api/iot/ingest {device_id, t_c, noise_db, lux, ts} header: X-Device-Secret
 - GET /api/rooms/:id/readings?window=15m
- agentes
 - GET /api/rooms/:id/suggestions

jwt no header Authorization. rate-limit por ip em kv. argon2id para hash.

Agente (Heurística v1)

(ver imagem em [#Imagens → Agente (Heurística v1)](#agente-heuristica))

- se noise_db>65 e energy<3 -> move para sala X com menor ruído.
- se t_c>27 e mood<=2 -> pausa 5min + água.
- se 2 usuários com status focus>30min no mesmo room e tasks relacionadas -> sugerir pairing.
- se lux<200 -> ajustar iluminação (nota).

Segurança

- hash: argon2id; pepper em secret.
- jwt curto (15min) + refresh opcional em cookie httpOnly (v1 opcional).
- validação de input, limitar body 32KB, sane de html no chat.
- secrets: wrangler secret. no repo só nomes.

NFR

- p50 api < 100ms; p95 < 300ms.
- primeiro byte ui < 200ms via pages.
- ingest suporta 10 req/s no free.

Telemetria

- counters no kv: logins, msgs, tasks_move, sugestões aceitas.

Critérios de sucesso

- demo com 2 rooms, 3 usuários, iot_sim rodando.
- sugestões aparecem reagindo a leituras e checkins.
- zip+pdf+pitch de 3min.

env vars

- JWT_SECRET

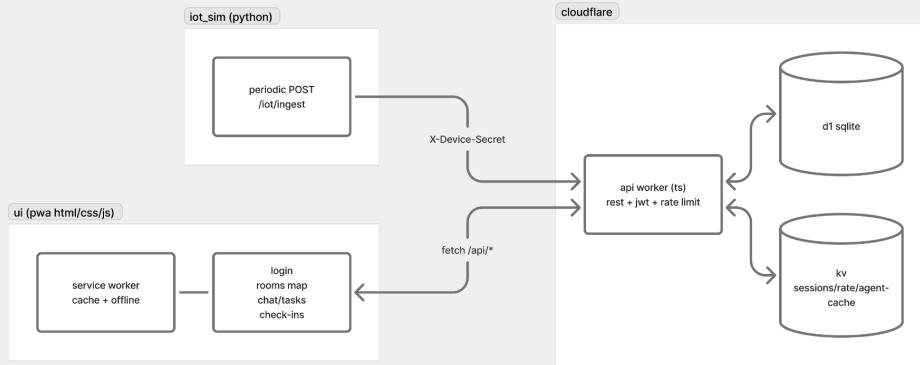
- PEPPER
- DEVICE_INGEST_RATE_MAX (opcional)
- ZAI_API_KEY (geração de assets opcional)

FIAPI mapping

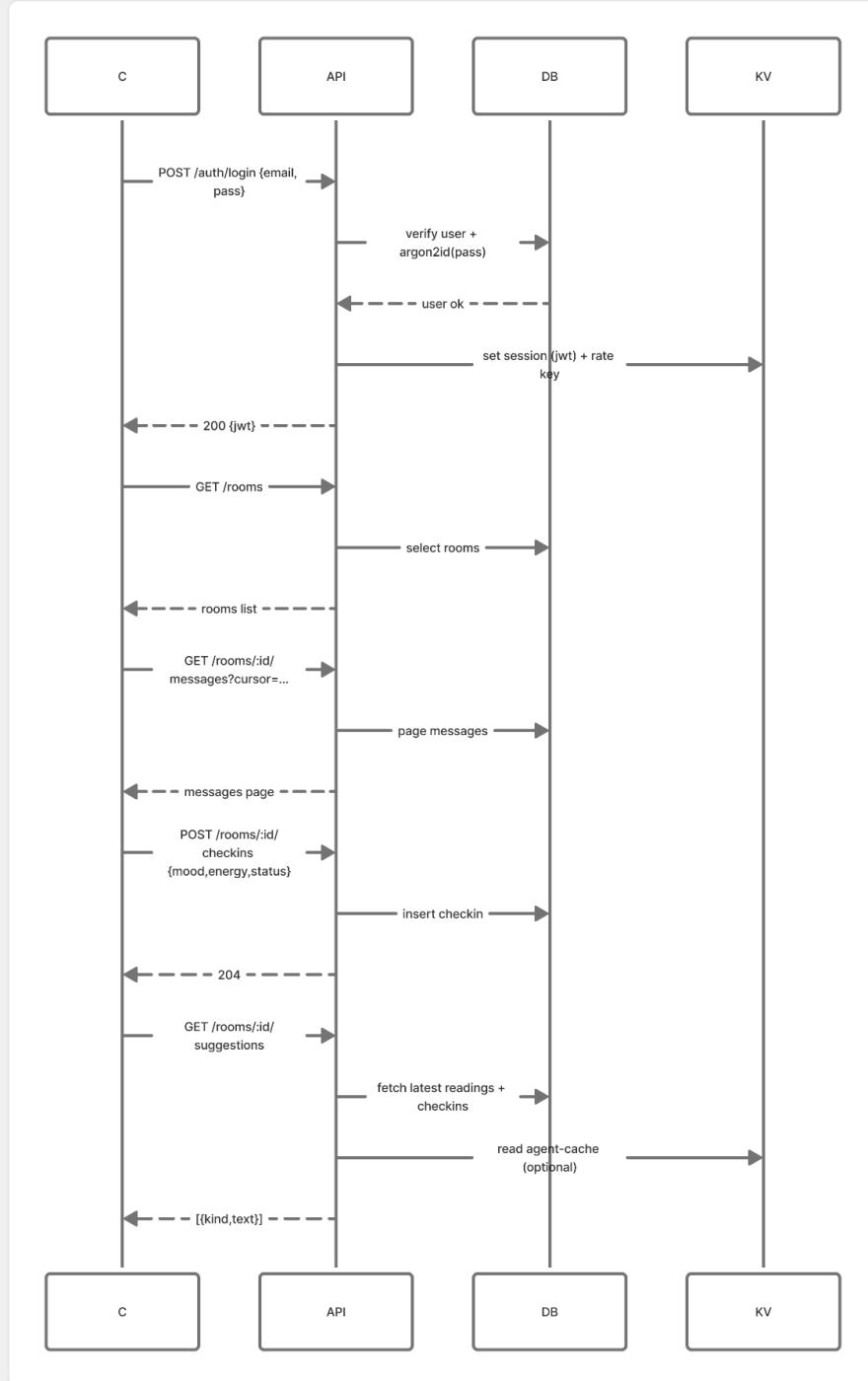
1. banco: sql acima (≥ 4 entidades).
2. api: rotas rest (≥ 5 endpoints).
3. testes: plano e 3 execuções (unit simples + curl evidências).
4. mobile: pwa responsivo 3 telas (login, room, tasks).
5. segurança: hash, roles, validação, rate-limit.
6. iot: iot_sim + ingest + uso em sugestões e painel.

Imagens

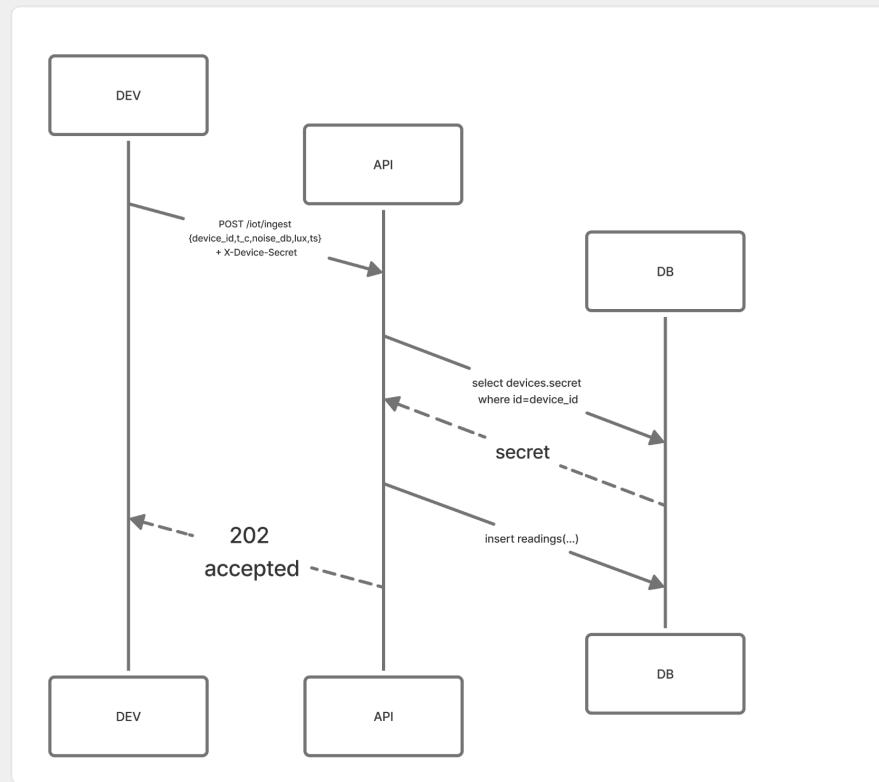
Arquitetura



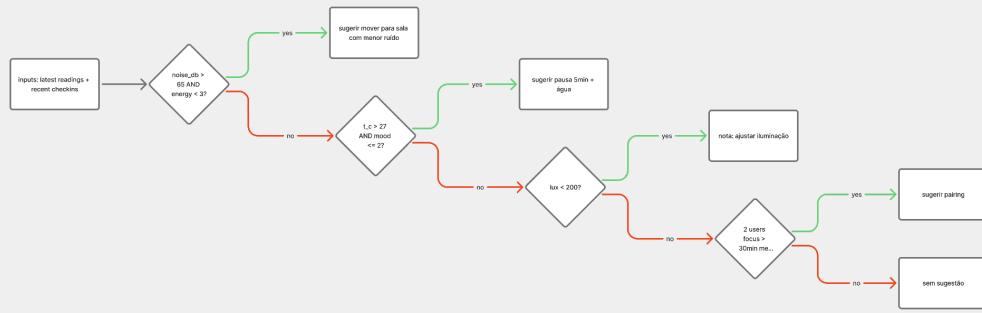
Fluxo Principal



Ingestão IOT



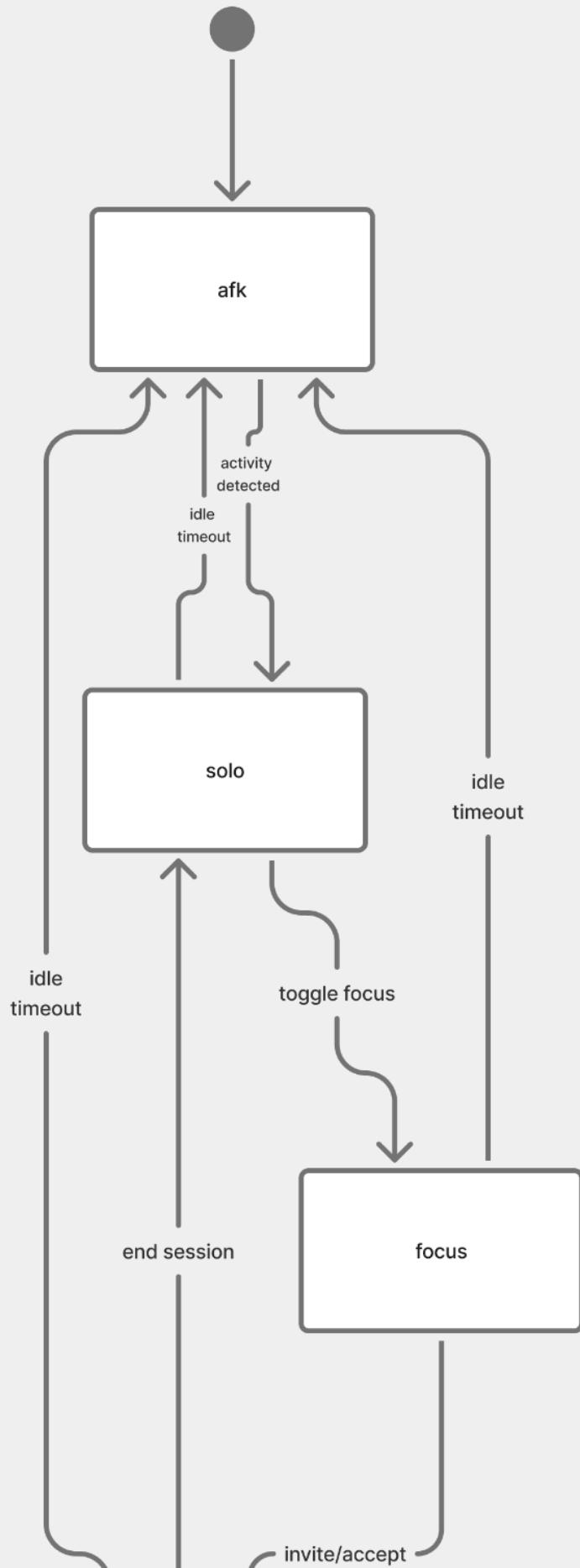
Agente (Heurística v1)



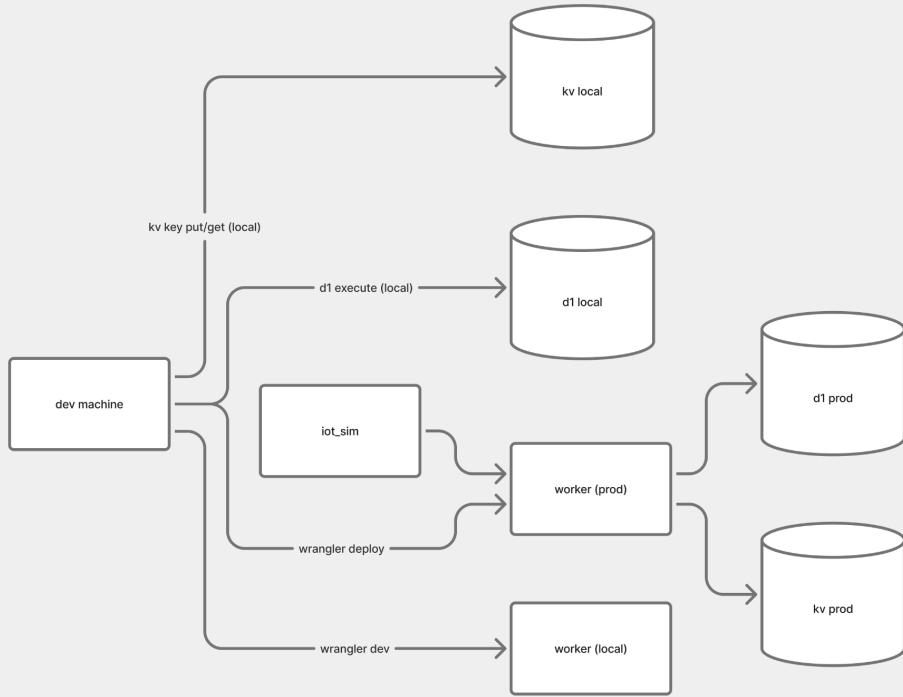
Endpoints



Estados de Presença



Deploy



Figma

Arquitetura

<https://www.figma.com/board/RnpSFs6Rj2zN0iZb2tXIOD/holo.work-%E2%80%94-arquitetura?node-id=0-1&p=f&t=VWsVSS78qQ1gCTcg-0>

Fluxo Principal

<https://www.figma.com/board/NNopmsqV4eYfyfd5nySpRw/holo.work-%E2%80%94-fluxo-principal?node-id=0-1&p=f&t=iG38V2JQwoyfJE1M-0>

Ingestão IOT

<https://www.figma.com/board/mylZk4aj3JyOLpGGkBRISu/holo.work-%E2%80%94-ingest%C3%A3o-iot?node-id=0-1&p=f&t=2DMRjlxAxgXkMxgB-0>

Agente Heurística

<https://www.figma.com/board/3xvZqC9J1M4GLaLdsNPxiQ/holo.work-%E2%80%94-agente--heur%C3%ADstica-v1?-node-id=0-1&p=f&t=vpBNqo9dF7BoIBS6-0>

Endpoints

<https://www.figma.com/board/z6YClh3FALBOUO9w3RJStd/holo.work-%E2%80%94-endpoints?node-id=0-1&p=f&t=pnMyQjh6VPEO661z-0>

Estados de Presença

<https://www.figma.com/board/jNwcCSkh9w1BCVxbgYzbte/holo.work-%E2%80%94-estados-de-presen%C3%A7a?node-id=0-1&p=f&t=a47lx0mnsAQObMfl-0>

Deploy

<https://www.figma.com/board/QAiEky9luSHOt6u4moFLea/holo.work-%E2%80%94-deploy?node-id=0-1&p=f&t=t7kQnDRCOIKkI5cD-0>