LifeOS – La Piattaforma Universale del Benessere 💡 Il “sistema operativo” per la vita quotidiana: salute, lavoro, relazioni, produttività. • Target: persone 25–55 anni, stressate, con poco tempo, che vogliono migliorarsi. • Cosa fa: • Unisce AI predittiva (es. “rischio burnout alto questa settimana”) + piani di azione personalizzati (esercizi, meditazione, nutrizione). • Integra wearable (Garmin, Apple Watch, Oura) e dà un LifeScore giornaliero. • Marketplace integrato: puoi comprare sessioni da coach, trainer, psicologi. • Monetizzazione: • Freemium: base gratis, premium 9€/mese. • Fee 20% su ogni servizio venduto dal marketplace. • Scalabilità: globale, chiunque ha stress, salute, produttività da migliorare. • Perché funziona: è “Canva della vita quotidiana”, semplice, bello e utile.

**🎯 Visione**

LifeOS = **“Sistema operativo della vita”** → un’unica piattaforma che integra salute, produttività e benessere quotidiano, con AI e marketplace.  
Non un’app di fitness, non un’app di meditazione, ma *l’hub centrale della tua vita personale*.

**👤 Target**

* Età: **25–55 anni**
* Profili: lavoratori stressati, manager, freelance, genitori con poco tempo, knowledge workers.
* Problema: mancanza di tempo + overload informativo + difficoltà a mantenere equilibrio.
* Soluzione: LifeOS ti dice cosa fare oggi per stare meglio e lavorare meglio.

**⚙️ Core Features**

1. **AI predittiva**: analizza dati personali (sleep, HRV, calendario, storico abitudini) → segnala rischi (stress, burnout, calo produttività).
2. **LifeScore**: un numero unico che rappresenta il tuo stato di energia e benessere (come il punteggio credito, ma per la vita).
3. **Piani di azione personalizzati**: workout brevi, tecniche di respirazione, micro-routine produttive, suggerimenti alimentari.
4. **Integrazione wearables**: Apple Watch, Garmin, Oura, Fitbit → sincronizzazione automatica.
5. **Marketplace del benessere**: accesso diretto a coach, psicologi, nutrizionisti, corsi e workshop.
6. **Gamification**: badge, streaks, progressi → motivazione continua.

**💰 Monetizzazione**

* **Freemium**:
  + Gratis → tracking base, LifeScore, consigli standard.
  + Premium → 9€/mese → AI avanzata, integrazione wearables, piani personalizzati, report predittivi.
* **Marketplace**: fee 20% su ogni transazione (sessioni coaching, corsi, consulenze).
* **Corporate version**: pacchetto per aziende → LifeOS come benefit per i dipendenti.

**🌍 Scalabilità**

* Problema universale (stress, salute, produttività).
* Modello SaaS + Marketplace → globalizzato facilmente.
* Partnership con brand wellness, palestre, psicologi online → network effect.

**🔑 Perché Funziona**

* Oggi esistono decine di app verticali (meditazione, fitness, agenda).
* **Nessuna** è un *hub centrale* che aggrega **tutto** con un’interfaccia semplice.
* LifeOS diventa il “Canva del benessere”: una dashboard unica, intuitiva e bella.

👉 Prossimo step che ti proporrei:

* definire **MVP** → cosa mettiamo nella prima versione (probabilmente LifeScore + AI consigli base + integrazione con un solo wearable + micro marketplace).
* partiamo B2C: deve essere **semplice, chiaro e subito percepibile dal singolo utente** → l’utente scarica, collega il suo wearable (o inserisce dati base manuali) e *capisce subito perché LifeOS è diverso dalle altre app*.

**🚀 MVP LifeOS (B2C)**

**Obiettivo:** validare che le persone usano l’app ogni giorno e sono disposte a pagare per più insights + marketplace.

**🔑 Feature Core (da avere subito)**

1. **LifeScore giornaliero**
   * Calcolato su 3 dimensioni base:
     + Sonno (qualità, ore, regolarità)
     + Energia fisica (passi, movimento, battito/HRV)
     + Stress percepito (mini check-in giornaliero)
   * Output = numero 0–100, più un “trend” (es. +3 vs ieri).
2. **AI Suggerimenti immediati**
   * In base al LifeScore, l’app ti dice:
     + “Fai una camminata di 10 min oggi”
     + “Respirazione 4-7-8 prima di dormire”
     + “Non caricare troppo lavoro oggi: rischio burnout medio”.
   * 1–2 micro suggerimenti al giorno, facili da eseguire.
3. **Integrazione con 1–2 wearables**
   * Apple Health / Google Fit (copri subito Apple Watch + Android base).
   * (Garmin, Oura ecc. li aggiungi in fase 2).
4. **Check-in rapido**
   * 1 domanda al giorno: “Come ti senti da 1 a 5?”
   * Input manuale minimo per arricchire il modello AI.

**🎁 Feature Nice-to-have (da valutare se tempo/risorse)**

* Mini sezione **Marketplace base**: 3–4 coach/psicologi disponibili, anche finti all’inizio → serve a testare l’interesse.
* **Report settimanale via email**: “Questa settimana il tuo LifeScore è salito di 12 punti”.
* **Gamification base**: streaks (giorni consecutivi di check-in).

**🛠️ Tech Stack MVP**

* **Frontend**: React Native (cross-platform iOS + Android).
* **Backend**: Supabase (auth + db + storage).
* **AI**: regole + modelli leggeri (no GPT pesanti per ora → costi alti), magari un motore di scoring semplice.
* **Integrazioni**: HealthKit (iOS), Google Fit (Android).

**📈 Metriche da validare**

1. **Retention D7 e D30** → la gente torna a usare l’app?
2. **% utenti premium** → 5–10% target.
3. **Engagement medio** → almeno 1 check-in al giorno.
4. **Interesse marketplace** → click su “scopri coach/psicologo”.

**🕑 Timeline MVP**

* UX/UI mockup (LifeScore dashboard semplice).
* Set up Supabase + auth + HealthKit/Google Fit integrazione.
* Implementazione LifeScore + check-in manuale.
* Suggerimenti AI base (if → then).
* Beta test con 50–100 utenti.
* Lancio Marketplace minimale.
* Email report settimanale.
* Lancio pubblico App Store / Play Store.
* Campagna ads + contenuti.

👉 Ora la vera scelta è:

* facciamo un **LifeScore “semplice”** basato su 2–3 dati subito disponibili (tipo passi, sonno e check-in umorale)?
* oppure puntiamo a un **LifeScore “serio”** con wearable più avanzati (HRV, Oura, Garmin), ma rischiamo tempi più lunghi?

Vuoi che ti preparo due versioni del LifeScore (basic vs advanced) così vediamo quale conviene lanciare per MVP?

**ChatGPT ha detto:**

Perfetto 👌 allora ti preparo le due versioni del **LifeScore**: una **Basic** per partire subito (MVP più leggero e veloce da lanciare) e una **Advanced** che puoi sviluppare come fase 2, più solida e scientifica.

**🔹 Versione 1 – LifeScore Basic (MVP)**

**Obiettivo:** dare all’utente un numero subito comprensibile, senza dipendere da wearable avanzati.

**Input dati**

* **Passi giornalieri** (da Google Fit / Apple Health).
* **Ore di sonno** (da app nativa o inserimento manuale).
* **Check-in umore** (scala 1–5).

**Algoritmo semplice (esempio)**

* Sonno = 40% peso → 8h = punteggio pieno, sotto 5h = punteggio minimo.
* Attività (passi) = 30% → target 7.000 passi.
* Umore self-report = 30%.

👉 Output: punteggio 0–100 con emoji (🔴 🟡 🟢) per rendere immediato.

**Vantaggi**

* Subito pronto, dati facili da raccogliere.
* Utente percepisce valore in 2 minuti.
* Costi di sviluppo bassissimi.

**Limiti**

* Non scientificissimo.
* Non misura stress fisiologico vero (HRV, resting HR, ecc.).

**🔹 Versione 2 – LifeScore Advanced (Phase 2)**

**Obiettivo:** trasformare LifeOS in *il vero sistema operativo del benessere*, con dati oggettivi e predittivi.

**Input dati**

* **HRV (Heart Rate Variability)** → stress/recupero.
* **Frequenza cardiaca a riposo**.
* **Sonno (ore + qualità + fasi)** da wearable (Oura, Garmin, Apple Watch).
* **Attività fisica** (calorie attive, passi, VO2max se disponibile).
* **Check-in umorale** (per incrociare dati fisiologici e percepiti).

**Algoritmo avanzato**

* Ponderazione dinamica con AI: il peso delle variabili cambia a seconda del profilo dell’utente (es. se dorme poco, sonno pesa di più).
* Sistema predittivo: “rischio burnout alto nei prossimi 3 giorni” basato su trend HRV + sonno + stress auto-report.

👉 Output:

* Punteggio 0–100 + trend settimanale.
* Insights predittivi (non solo descrittivi).

**Vantaggi**

* Molto più accurato, scientifico e credibile.
* Differenzia LifeOS dalle mille app “di tracking”.

**Limiti**

* Richiede integrazioni più complesse (API Garmin, Oura ecc.).
* Onboarding più difficile → non tutti hanno wearable avanzati.

**🚀 Strategia MVP**

* **Fase 1 (Basic):** lancia subito → validi retention e willingness to pay.
* **Fase 2 (Advanced):** aggiungi HRV, resting HR e stress predittivo → alzi il valore percepito e puoi spingere su premium.

Vuoi che ti preparo una **tabella comparativa con costi, tempi e valore percepito** per scegliere quale LifeScore portare in produzione prima?

**Hai detto:**

Partirei da una via di mezzo tra le due che hai proposto. Un sistema avanzato di input dati con algoritmo avanzato ma senza utilizzare strumenti wearable avanzati

**ChatGPT ha detto:**

Perfetto, Andrea 👍 allora stai dicendo: **non voglio un MVP troppo “giocattolo”**, ma neanche legarmi subito a Garmin/Oura (che ti bloccherebbero con API e sviluppo lungo).

Quindi puntiamo su una **via di mezzo**:

* Input dati **avanzati** (più ricchi del solo “sonno+passi+umore”).
* Algoritmo di scoring **predittivo e dinamico**.
* Tutto basato su fonti accessibili (Apple Health, Google Fit, inserimento manuale).

**🔹 LifeScore “Intermediate” – MVP Potenziato**

**Input dati (senza wearable avanzati)**

* **Sonno** (ore + qualità percepita → check-in 1–5).
* **Attività fisica** (passi + minuti attivi, già in Google Fit/Apple Health).
* **Energia/umore** (check-in giornaliero breve).
* **Stress percepito** (1 domanda tipo “quanto ti senti stressato oggi?”).
* **Frequenza cardiaca media giornaliera** (disponibile anche da smartwatch base).

👉 Così hai già **5 dimensioni**, quindi molto più robusto del Basic.

**Algoritmo (avanzato, senza wearable pro)**

* **Pesi dinamici**: il sistema assegna più importanza alla variabile peggiore (es. se sonno <5h, il sonno pesa di più quel giorno).
* **Trend analysis**: non guardi solo il singolo giorno, ma la direzione (es. 3 giorni di HR medio più alto del solito = alert stress).
* **Scoring finale**:
  + 0–40 = 🔴 “Rischio burnout / livello basso”
  + 41–70 = 🟡 “Equilibrio instabile”
  + 71–100 = 🟢 “Benessere ottimale”

**Output per utente**

1. **LifeScore giornaliero** (numero + colore).
2. **Insight predittivo semplice**:
   * “Trend stress in aumento, attenzione a sonno e pause.”
   * “Energia in crescita, ottimo momento per allenarti di più.”
3. **Suggerimenti AI personalizzati**: micro-azioni da 5–10 minuti → non un piano gigante, ma consigli pratici.

**Vantaggi di questa via di mezzo**

✅ Algoritmo già avanzato e credibile → non sembra un giochino.  
✅ Non dipendi da hardware costoso → più inclusivo e scalabile.  
✅ Onboarding veloce → chiunque può iniziare in 2 minuti.  
✅ Già differenziante rispetto a tutte le app “tracker” standard.

**Limiti**

* Non hai ancora HRV e sonno “scientifico” (servirebbero Oura, Whoop ecc.).
* Quindi la parte “predittiva” è limitata a trend di dati base + self-report.
* Alcuni utenti molto tech-savvy potrebbero chiedere subito l’integrazione wearables pro.

👉 Io direi che questa versione **Intermediate** è perfetta come **MVP di lancio**, perché:

* è abbastanza seria da farti sembrare un prodotto “premium” e non banale;
* è abbastanza leggera da sviluppare in pochi mesi;
* ti permette di testare pricing e retention **senza incasinarti subito con API Garmin/Oura**.

Vuoi che ti disegni una **mappa visuale del flusso utente (onboarding → input → LifeScore → suggerimento)** così vedi subito come presentarlo in app?

**Hai detto:**

La parte migliore é quella dei consigli che vengono dati

**ChatGPT ha detto:**

Assolutamente, la parte dei **suggerimenti personalizzati** è il motore che trasforma LifeOS da “una bella app” a **un vero supporto quotidiano motivante**. E la buona notizia è che esistono evidenze solide su come strutturarla al meglio per aumentare l'engagement e l’adozione nel tempo.

**Perché i micro-consigli funzionano (se fatti bene)**

* **Personalizzazione = percezione di cura**: utenti vogliono sentirsi “capiti”, non ricevere messaggi generici. Il 75 % delle persone desidera soluzioni più personalizzate nella wellness experience [Idomoo Personalized Video](https://www.idomoo.com/blog/personalization-in-health-and-wellness-top-digital-trends/?utm_source=chatgpt.com).
* **Behavioral nudging intelligente**: sistemi avanzati di “nudges” personalizzati (ad esempio basati su GNN o Reinforcement Learning) aumentano attività e coinvolgimento fino a circa il 6–7 % [arXiv+1](https://arxiv.org/abs/2401.10816?utm_source=chatgpt.com).
* **Micro-punteggi e messaggi brevi** sono perfetti per “spezzare i muri” cognitivi della giornata: un consiglio veloce, contestuale, aumenta la probabilità di esecuzione.

**Come strutturare i micro-consigli su LifeOS**

**1. Timing contestuale**

* Utilizza i trend dei dati (es. sonno basso da due giorni, battito medio alto) per “spingere” il consiglio **nel momento di maggior bisogno**.
* Es. “Vedo che dormi poco negli ultimi giorni. Oggi fatti una pausa meditativa di 5 minuti verso metà giornata.”

**2. Linguaggio empatico e motivante**

* Usa tono positivo, empatico, quasi come se fosse un coach amico.  
  Es.: “Bel segnale 🟢 Oggi ti senti più energico: ottimo momento per una camminata veloce di 10 min.”

**3. Brevità + azione concreta**

* Il consiglio dovrebbe essere **brevissimo** e attuabile in 5–10 minuti.  
  Tipo: “Respirazione 4‑7‑8 prima di dormire” o “Un micro-break—stand up, scuoti le spalle.”

**4. Gamification soft**

* Se c’è una streak attiva, puoi premiare l’utente: “Hai fatto check‑in 5 giorni di fila—great job! Oggi un micro-push sarà utile per mantenere la rotta.”

**5. Approccio empatico e rispettoso**

* Evita la pressione. Il consiglio può essere opzionale: “Se ti va, prova questo…” invia il messaggio con tono accogliente e non invadente [biz4group.com](https://www.biz4group.com/blog/best-practices-in-mental-health-design?utm_source=chatgpt.com).

**6. Progressi e piccoli casi studio**

* Inserisci piccole frasi che rendano la vita reale: “Diversi utenti in tua situazione hanno trovato utile...”, “piccoli passi portano a grandi risultati” → costruiscono fiducia ed engagement.

**Sintesi in tabella**

| **Principio** | **Cosa vuol dire in LifeOS** |
| --- | --- |
| **Personalizzazione** | Consigli mirati basati su trend reali sui dati |
| **Tempismo giusto** | Suggerimenti inviati al momento di potenziale bisogno |
| **Micro-azione** | Proposte brevi, immediate, pratiche |
| **Tono empatico** | Copy umano, positivo, non giudicante |
| **Soft gamification** | Streak, piccoli badge e riconoscimenti |
| **Autonomia utente** | Offerta non invadente, con opzione “provalo se vuoi” |

**Esempi pratici di micro-consigli**

* *“Hai dormito solo 5 ore? Rallenta oggi: prenditi 5 min per una micro-meditazione al pomeriggio.”*
* *“Energia in crescita oggi! Se ti va, fai passeggiata breve o respiri profondi tra una call e l’altra.”*
* *“Stai mantenendo lo streak di check-in? Ottimo! Continua così—oggi prova a bere un bicchiere d’acqua extra a metà mattina.”*

Se vuoi, posso anche prepararti una **recipe board grafica** con dei micro-consigli standard, da adattare con i tuoi parametri LifeScore—ti serve?

**Hai detto:**

Bisogna anche insegnare alle persone a usare i consigli perché spesso non sanno da che parte iniziare. Sarà quindi necessario attivare un tutorial dell’attività suggerita (respiro 4/7/8, meditazione 5 minuti, ecc…) dovranno essere guidate quandonutente decide di attivarle

**ChatGPT ha detto:**

Esatto 💡 hai centrato un punto fondamentale: **non basta dire cosa fare, bisogna accompagnare la persona a farlo**.  
Se l’utente legge “respira 4-7-8” e non sa cosa sia → *drop immediato*.  
Quindi serve un **layer educativo + guida pratica in-app**.

**🔹 Struttura della funzionalità “Consiglio + Tutorial”**

1. **Consiglio breve → CTA “Mostrami come”**
   * Esempio: “Rischio stress alto. Prova la tecnica 4-7-8. 👉 [Guida]”
2. **Tutorial guidato step-by-step**
   * Breve intro testuale o video (max 20 sec).
   * Timer/animazione interattiva che accompagna l’utente (es. cerchio che si espande/contrae per la respirazione).
   * Audio opzionale (“inspira… trattieni… espira”).
3. **Conferma/feedback**
   * Alla fine: “Hai completato la tecnica. Come ti senti adesso? 😊😐😞”
   * Questo dato rientra nel LifeScore (feedback loop).

**🔹 Tipologie di tutorial possibili**

* **Respirazione** → animazioni interattive (cerchi, onde).
* **Meditazione 5 min** → audio + countdown semplice.
* **Camminata 10 min** → push con timer e suggerimento “torna quando hai finito per segnare completata l’attività”.
* **Stretching breve** → mini video/gif con esercizio guidato.

**🔹 Vantaggi**

✅ Riduce la barriera iniziale (“non so come si fa”).  
✅ Crea **engagement attivo**: non solo leggere ma eseguire.  
✅ Genera **success experience** → la persona percepisce beneficio immediato, quindi torna.  
✅ Raccoglie dati di completamento → metriche reali sull’efficacia dei consigli.

**🔹 Roadmap implementazione**

1. MVP: **testo + animazione base (respirazione)** → super rapido da implementare.
2. Fase 2: **audio-guided mini-sessions** → come Calm o Headspace, ma integrate nei tuoi consigli.
3. Fase 3: **video brevi (1–2 min)** per esercizi fisici o stretching.

👉 Questo cambia la percezione di LifeOS: da *app che ti dice cosa fare* a **coach virtuale che ti accompagna passo passo**.  
Così l’utente si sente **guidato, non solo consigliato**.

Vuoi che ti prepari un **primo esempio di storyboard UX** per una sessione guidata (es. respiro 4-7-8), così vedi come appare nell’app?

**Hai detto:**

No. Passiamo a come possiamo creare la struttura dei files

**ChatGPT ha detto:**

Ragionato per 6s

Grande. Andiamo dritti al sodo: ti propongo una **struttura modular-monorepo** così condividi logica (LifeScore, motore consigli, tipi) tra app mobile e sito marketing. Tech: **Expo (React Native)** per mobile, **Next.js** per landing/SEO, **Supabase** per auth/db/storage/edge functions.

**Struttura repository**

lifeos/

├─ apps/

│ ├─ mobile/ # App Expo (iOS/Android)

│ │ ├─ app/ # App Router di Expo

│ │ │ ├─ (auth)/

│ │ │ │ ├─ sign-in.tsx

│ │ │ │ └─ sign-up.tsx

│ │ │ ├─ (onboarding)/

│ │ │ │ ├─ welcome.tsx

│ │ │ │ ├─ permissions.tsx # Health/Activity consent

│ │ │ │ └─ baseline-checkin.tsx # primo check-in

│ │ │ ├─ (main)/

│ │ │ │ ├─ index.tsx # Dashboard con LifeScore

│ │ │ │ ├─ checkin.tsx # 1 domanda al giorno

│ │ │ │ ├─ insights.tsx # trend & alert

│ │ │ │ ├─ suggestions/

│ │ │ │ │ ├─ index.tsx # lista suggerimenti del giorno

│ │ │ │ │ └─ [id].tsx # pagina “Mostrami come”

│ │ │ │ └─ settings.tsx

│ │ │ └─ +not-found.tsx

│ │ ├─ components/

│ │ │ ├─ LifeScoreCard.tsx

│ │ │ ├─ TrendSparkline.tsx

│ │ │ ├─ SuggestionCard.tsx

│ │ │ ├─ BreathAnimator.tsx # animazione 4-7-8

│ │ │ └─ TutorialStep.tsx

│ │ ├─ hooks/

│ │ │ ├─ useDailyCheckin.ts

│ │ │ ├─ usePermissions.ts

│ │ │ ├─ useHealthData.ts # wrapper per Apple Health / Google Fit

│ │ │ └─ useNotifications.ts

│ │ ├─ lib/

│ │ │ ├─ supabase.ts # client inizializzato

│ │ │ ├─ health/apple.ts # iOS HealthKit

│ │ │ ├─ health/google.ts # Android Google Fit

│ │ │ └─ storage.ts

│ │ ├─ state/

│ │ │ └─ store.ts # Zustand (user, score, suggestions)

│ │ ├─ assets/ # icone, audio mini-meditazioni

│ │ ├─ constants/

│ │ │ └─ thresholds.ts # soglie passi, sonno, ecc.

│ │ ├─ app.json

│ │ ├─ package.json

│ │ └─ tsconfig.json

│ │

│ └─ web/ # Next.js (marketing + account management base)

│ ├─ app/

│ │ ├─ page.tsx # Hero “LifeOS”

│ │ ├─ pricing/page.tsx

│ │ ├─ legal/

│ │ │ ├─ privacy.md

│ │ │ └─ terms.md

│ │ └─ download/page.tsx # QR / link store

│ ├─ components/

│ │ └─ CTA.tsx

│ ├─ lib/supabase-server.ts

│ ├─ public/

│ ├─ package.json

│ └─ next.config.js

│

├─ packages/

│ ├─ core/ # LOGICA CONDIVISA (agnostica dalla UI)

│ │ ├─ scoring/

│ │ │ ├─ lifeScore.ts # algoritmo “Intermediate”: pesi dinamici + trend

│ │ │ └─ normalizers.ts

│ │ ├─ advice/

│ │ │ ├─ engine.ts # regole → suggerimenti (con priorità & cooldown)

│ │ │ ├─ recipes/

│ │ │ │ ├─ breathing-478.ts

│ │ │ │ ├─ 5min-meditation.ts

│ │ │ │ └─ 10min-walk.ts

│ │ ├─ predictors/

│ │ │ └─ stressTrend.ts # semplici predizioni su 3-7 giorni

│ │ ├─ index.ts

│ │ └─ package.json

│ ├─ ui/ # mini design system RN + web (tamagochi semplice)

│ │ ├─ Button.tsx

│ │ ├─ Card.tsx

│ │ ├─ Typography.tsx

│ │ └─ package.json

│ └─ types/

│ ├─ index.ts # tipi: LifeScore, Suggestion, Checkin, Trend

│ └─ package.json

│

├─ supabase/

│ ├─ migrations/ # SQL schema versionato

│ │ ├─ 0001\_init.sql

│ │ └─ 0002\_events.sql

│ ├─ functions/

│ │ ├─ daily-rollup/index.ts # Edge Function: aggrega metriche/genera consigli

│ │ ├─ send-weekly-report/index.ts # invio email report

│ │ └─ verify-receipts/index.ts # premium receipts (StoreKit/Play)

│ └─ seed/

│ └─ suggestions.json # seed ricette tutorial

│

├─ tools/

│ ├─ scripts/

│ │ ├─ dev:link.sh # collega packages ai progetti

│ │ └─ sync-env.ts # push/pull env locali → Supabase/Vercel

│ └─ github/

│ └─ ci.yml # build mobile lint + build web

│

├─ .env.example

├─ package.json # workspace (pnpm/yarn workspaces)

└─ README.md

**Tabelle (Supabase) – schema iniziale**

**users**

* id (uuid, pk)
* email, name, premium\_until (date), created\_at

**health\_metrics** (giornaliere)

* id, user\_id (fk)
* date
* steps (int), active\_minutes (int)
* sleep\_hours (decimal), sleep\_quality (1–5)
* hr\_avg (int, opzionale)
* mood (1–5), stress (1–5)
* source (enum: manual, apple, google)

**lifescores**

* id, user\_id, date
* score (int 0–100)
* trend\_3d (int), trend\_7d (int)
* flags (jsonb) # es. { "low\_sleep": true, "high\_hr": true }

**suggestions** (catalogo)

* id, key (es. “breathing-478”), title, short\_copy, category
* duration\_sec (int), difficulty (1–3)
* tutorial (jsonb) # step-by-step, cue timing

**user\_suggestions** (storico erogazioni)

* id, user\_id, suggestion\_id, date
* reason (jsonb) # perché consigliato (regola matchata)
* completed (bool), feedback\_mood (1–5), time\_spent\_sec

**events** (per analytics)

* id, user\_id, type, payload (jsonb), created\_at

**Algoritmo “Intermediate” (dove sta cosa)**

* packages/core/scoring/lifeScore.ts
  + **Input:** sleep\_hours, steps, active\_minutes, hr\_avg?, mood, stress
  + **Normalizzazione:** normalizers.ts (ad es. passi → 0–1 con target 7.000)
  + **Pesi dinamici:** aumentano il peso della dimensione peggiore del giorno
  + **Trend:** medie mobili 3/7 giorni (da predictors/stressTrend.ts)
  + **Output:** { score, reasons, flags, trend3d, trend7d }
* packages/core/advice/engine.ts
  + **Regole:** mapping flags/trend → lista suggerimenti ordinati per priorità
  + **Cooldown:** evita di riproporre lo stesso consiglio troppe volte
  + **Context:** fascia oraria, preferenze utente, durata disponibile
* **Edge Function daily-rollup**
  + Cron giornaliero: aggrega metriche di ieri → calcola LifeScore → scrive user\_suggestions per oggi.

**Librerie chiave**

**Mobile (Expo)**

* Auth/DB: @supabase/supabase-js
* Stato: zustand
* Notifiche: expo-notifications
* Health:
  + iOS: react-native-health (via config plugin Expo)
  + Android: react-native-google-fit
* Router: expo-router
* Grafici micro: react-native-svg (+ sparkline minimal)

**Web**

* Next.js (App Router), @supabase/ssr

**Env & Config (esempio .env)**

# Supabase

EXPO\_PUBLIC\_SUPABASE\_URL=

EXPO\_PUBLIC\_SUPABASE\_ANON\_KEY=

# Web (Next)

NEXT\_PUBLIC\_SUPABASE\_URL=

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY=

# Cron (Edge Functions)

SERVICE\_ROLE\_KEY= # solo su server

CRON\_SECRET=

NB: nelle app Expo le chiavi “public” devono essere effettivamente public (anon). Tutto ciò che è sensibile va nelle Edge Functions.

**Flusso dati (giorno tipo)**

1. App legge dati Health (o input manuale) → scrive health\_metrics (oggi).
2. Cron daily-rollup (notte): calcolo LifeScore + genera user\_suggestions per il giorno dopo.
3. La mattina: push notification “Hai 2 micro-consigli pronti”.
4. Utente apre → packages/core/advice filtra/ordina e mostra la top action.
5. Se avvia tutorial → tracking events + eventuale user\_suggestions.completed=true.

**Cartelle “chiave” e perché**

* packages/core/ → è ciò che rende LifeOS *LifeOS*. Portabile ovunque.
* apps/mobile/app/(main)/suggestions/[id].tsx → pagina tutorial guidato (respiro, meditazione, camminata).
* supabase/functions/daily-rollup/ → automazione valore quotidiano (riduce logica client).

**Roadmap dev (setup rapido)**

1. **Repo & workspace**: pnpm workspaces, crea apps/mobile, apps/web, packages/core/types/ui.
2. **Supabase**: crea progetto, applica migrations/0001\_init.sql, seed di 3 ricette tutorial.
3. **Mobile**: onboarding, check-in, lettura sleep/steps dal sistema, dashboard con LifeScore calcolato **in app** (prima versione).
4. **Edge Function**: aggiungi daily-rollup e sposta lì il calcolo giornaliero.
5. **Notification**: invio mattutino + deep link alla top suggestion.
6. **Web**: landing semplice + pricing + download.

**Extra (facoltativi utili)**

* **Feature flag** (es. packages/utils/flags.ts) per attivare Beta features.
* **i18n** (it/en) con dizionari in packages/ui.
* **Analytics**: PostHog/Amplitude (eventi client + edge).
* **QA**: E2E Detox (mobile) e Playwright (web).
* **CI**: GitHub Action che builda packages/core e lint per app.

Se vuoi, nel prossimo messaggio ti preparo:

* il **SQL della migrazione 0001\_init.sql**,
* gli **stub TypeScript** per lifeScore.ts e advice/engine.ts,
* e il **boilerplate dell’Edge Function daily-rollup**.