



# FINCA — Technical Build Spec

*Manual OS → internal tools → PWA → native (only after proof)*

Don't start as a full native app. Start as a manual operating system, become a PWA/private beta, go native only once behaviour is proven. New here. Known here.

## TECHNICAL BUILD SPEC

INTERNAL · CONFIDENTIAL

Manual OS → internal tools → PWA → native (only after proof)

Don't start as a full native app. Start as a manual operating system, become a PWA/private beta, go native only once behaviour is proven. New here. Known here.

**Companion:** FINCA\_Build\_Roadmap (business/ops sequence) · FINCA\_Execution\_OS\_Pack \$2 (CRM fields) · FINCA\_Product\_Demo.html (the 18 PWA screens, already designed). This doc is the **engineering** layer.

Build sequence (don't skip stages)

- 1. **Manual OS (Day 0-60)** — prove behaviour with no app. *Airtable/Supabase table · Tally/Typeform · WhatsApp Business · Stripe payment links · Notion/Softr/Glide member portal · Mailchimp/Beehiiv · Make/Zapier.* Run waitlist, onboarding, RSVP, Come Alone Mode, check-in, recaps, warm intros, perks, Passport status, Atlas, host briefings, partner reporting — **by hand**.
- 2. **Technical foundation (Month 2-3)** — proper backend on **Supabase** (Postgres + Auth + Storage + Edge Functions + Row Level Security).
- 3. **Internal ops dashboard first (Month 2-3)** — **Retool** (or custom admin). Make the *team* powerful before the member app is pretty.
- 4. **PWA / private beta (Month 3-6)** — lightweight, installable, fast to iterate. Native only later.

Recommended stack

| Layer          | Choice                                                                                                                                                                                | Notes                                                            |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| Backend        | <b>Supabase</b> (Postgres, Auth, Storage, Edge Functions, RLS)                                                                                                                        | one platform; RLS enforces privacy                               |
| Frontend       | <b>Next.js + React + Tailwind + Vercel</b> (or <i>FlutterFlow + Supabase for speed/lower cost</i> )                                                                                   | Next.js if you have a serious dev; FlutterFlow for visual/faster |
| Payments       | <b>Stripe</b> (Checkout → payment links first → subscriptions later; webhook into Supabase Edge Functions)                                                                            |                                                                  |
| Maps           | <b>Mapbox</b> (branded Atlas/Living City Map)                                                                                                                                         | Google Maps only if faster initially                             |
| Email          | <b>Resend / Postmark</b> (transactional) · Beehiiv/Mailchimp (newsletter)                                                                                                             |                                                                  |
| Messaging      | Email + <b>WhatsApp manual</b> first → WhatsApp Business API / Twilio later                                                                                                           |                                                                  |
| Analytics      | <b>PostHog</b> (product) · Plausible/GA4 (web) · Supabase/Airtable (ops)                                                                                                              |                                                                  |
| Internal tools | <b>Retool</b> + Supabase Studio + Airtable + Notion/Asana                                                                                                                             |                                                                  |
| AI (later)     | OpenAI/Claude API over <b>curated Finca data only</b> — retrieval from Atlas/Rooms/Hosts/Local Notes; output always one room / one place / one intro / one local note / one next step | not a generic chatbot                                            |

*Avoid Bubble for the long-term core (privacy, maps, event logic, partner dashboards, cross-city identity outgrow it) — prototype only.*

## Core data model (Supabase / Postgres — the start of the Belonging Graph)

Condensed; full field lists in build tickets. Tables:

1. **users** — id · name · email · phone · city · neighborhood · language · newcomer/local/visitor · arrival\_date · profession · interests · social\_comfort · arrival\_style · privacy\_mode · consent\_status · created\_at
2. **member\_profiles** — user\_id · bio · intent · preferred\_room\_types · preferred\_neighborhoods · intro\_preferences · visibility\_settings · regular\_status · passport\_level
3. **rooms** — id · title · room\_type · datetime · venue\_id · host\_id · capacity · description · energy\_tags · first\_timer\_friendly · come\_alone\_friendly · local\_bridge · price · sponsor\_id · status
4. **rsvps** — id · room\_id · user\_id · status · coming\_alone · arrival\_window · intro\_request · dietary · checked\_in · no\_show · payment\_status
5. **checkins** — id · user\_id · room\_id · venue\_id · type · timestamp · privacy\_level
6. **hosts** — id · user\_id · city · host\_level · room\_types · score · notes · training\_status
7. **host\_notes** — id · host\_id · room\_id · user\_id · note\_type · content · visibility
8. **venues/partners** — id · name · category · neighborhood · address · contact · partner\_status · trust\_tags · perk · qr\_code · renewal\_status
9. **partner\_redemptions** — id · user\_id · venue\_id · perk\_id · timestamp · source · value
10. **warm\_intros** — id · requester\_id · recipient\_id · host\_id · reason · status · consent\_a · consent\_b · sent\_at · follow\_up\_status
11. **passport\_activity** — id · user\_id · activity\_type · city · room\_id · venue\_id · intro\_id · weight · timestamp
12. **feedback** — id · user\_id · room\_id · host\_id · venue\_id · felt\_welcomed · come\_alone\_again · intro\_quality · comments · safety\_flag
13. **proof\_library** — id · type · quote · user\_id? · partner\_id? · room\_id · permission\_status · media\_url · created\_at

*RLS on every table; consent + visibility are columns, not afterthoughts. (See the Finca Data Promise — OS Pack §24.14.)*

## Key workflows (build as Edge Functions / automations)

1. **Waitlist → member** — join → onboard → Arrival Style → CRM tag → first-room rec → invite → RSVP.
2. **RSVP → Come Alone Mode** — RSVP → "coming alone" → comfort questions → host dashboard flag → pre-arrival guide → host greeting notes.

3. **Check-in** — arrive → QR/host check-in → attendance saved → Passport updates → host sees who's present → post-event flow fires.
4. **Post-event recap** — event ends → member recap → mark people met → request intro → next-room rec → feedback form.
5. **Warm intro** — request → consent check (both sides) → host sends contextual intro → 7-day follow-up tracked. *No open DMs.*
6. **Partner perk** — open Trusted Place → save/redeem → QR/NFC logs redemption → partner dashboard updates → monthly report generated.
7. **Regular status** — attend → return within 45 days → accept intro / visit partner place → internal Regular score updates → next belonging pathway surfaced.

## Version roadmap (build only what's proven)

| Version                       | Build                                                                                                                                                                                        | Success signal                                                                        |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <b>V-1 Manual MVP</b>         | landing · waitlist · CRM · event pages · payment links · WhatsApp flows · manual Passport/Atlas/intros · partner tracker                                                                     | first 100 qualified · 3 rooms · first perks · first second-visit data                 |
| <b>V0 Internal OS</b>         | admin dashboard · member/room/host/partner views · proof library · weekly metrics                                                                                                            | team runs rooms without chaos; partner reporting exists                               |
| <b>V1 Member PWA</b>          | login · onboarding · Arrival Style · home · rooms · event detail · Come Alone Mode · RSVP · QR check-in · recap · People You Met · Warm Intro · Passport · Atlas · perks · privacy · profile | members use it before/after rooms; Come Alone used; intros requested; return improves |
| <b>V2 Host Tools</b>          | host login · guest list · solo arrivals · suggested intros · check-in · room flow · post-event notes · room score                                                                            | non-founder hosts run rooms; quality consistent                                       |
| <b>V3 Partner Portal</b>      | partner profile · perk manager · redemptions · QR/NFC · monthly report · activation request · renewal                                                                                        | partners see value, renew/upgrade                                                     |
| <b>V4 Living City Map</b>     | trusted places · rooms soon · good-alone · Regulars Hubs · perks · member-loved · privacy-safe rhythm                                                                                        | Finca useful <i>between</i> events                                                    |
| <b>V5 Cross-city Passport</b> | city stamps · cross-city context · Welcome Relay · first host intro + trusted-place set in new city · city launch templates                                                                  | identity portable; second city cheaper/faster                                         |

**Don't build yet:** full native app · AI City Companion · open social feed · public profiles · DMs · follower counts · public live location · complex membership tiers · corporate portal · marketplace · advanced recsys · multi-city infra.

## The three magic moments (center the build on these)

1. **"I'm coming alone."** → *"You're expected. Your host knows you're coming. We'll help with your first hello."*
  2. **After the room** → *"You met three people. Want a warm intro to Ana? Your next room is Thursday's Creative Table. You're one step closer to being known in CDMX."*
  3. **At a partner place** → *"You're at a Finca Trusted Place. Save it, use your perk — or tell us if this could become one of your regular spots."* (This is where Atlas becomes commercial.)
- 
- 

## Team by stage

- **Manual MVP:** founder/CEO · partnerships lead · city/event operator · designer · no-code builder · ops assistant.
- **PWA:** product lead · UX/UI · full-stack or FlutterFlow builder · Supabase/back-end dev · QA · privacy/legal advisor.
- **Platform (later):** CTO · full-stack · mobile eng · data/analytics · lifecycle marketer · trust & safety · city ops.

**Keep internal:** product vision · room quality · host standard · member trust · partner relationships · data/privacy principles · brand voice · city playbook.

**Outsource:** first PWA build · landing page · visual polish · no-code setup · design assets · analytics setup · QA. *Keep one technical advisor close to stop you building the wrong thing.*

---

---

## The exact build order

1. Waitlist → 2. CRM → 3. Partner tracker → 4. Host tracker → 5. Run first rooms manually → 6. Track return + intros → 7. Internal ops dashboard → 8. Manual Passport + Atlas → 9. PWA around proven workflows → 10. Host tools → 11. Partner portal → 12. Living City Map → 13. Cross-city Passport → 14. AI City Companion (last).

*Build only what behaviour has already proven. The app productizes what's working — it doesn't go looking for it.*

*Finca — New here. Known here.*