Livro: a WhatsApp bookkeeping and tax assistant for Brazilian freelancers

What it does

Livro lives inside a Brazilian freelancer's WhatsApp. The freelancer tells it to invoice a foreign client in USDC. Livro builds a Solana Pay link with a fresh receiving address, never reused, and drafts the invoice text. It polls the chain for that payment, and the moment it arrives it looks up the official BACEN PTAX exchange rate for that day, converts the payment to BRL, and writes a permanent, append only ledger entry. It never assumes what the freelancer wants to do with the money next. It asks, or follows an explicit standing preference, and tracks whichever of three paths they choose: convert to BRL, hold as USDC, or allocate into a tokenized Brazilian government bond.

From this ledger it computes the two tax obligations Brazilian law actually imposes on this income (ordinary income tax at receipt, via Carnê-Leão, and capital gains tax at disposal, via weighted average cost basis), proactively reminds the freelancer before deadlines, and flags anything genuinely ambiguous for a human accountant instead of guessing. It exports an accountant ready report on request.

The build was later extended so any freelancer can message one shared WhatsApp Business number and get their own fully isolated agent, paying in prepaid USDC credits with no API key of their own required. That part is documented separately as post-bounty scope.

Who it's for

A Brazilian freelancer operating as an individual, no company, receiving USDC directly into a self custody Solana wallet from clients outside Brazil. Typically no accountant on retainer, currently tracking payments in a spreadsheet, if at all.

ZeroClaw features used

Feature	Use in Livro
Skills	9 skills: draft_invoice, classify_receipt, book_receipt, draft_refund, draft_bond_allocation, register_external_holding, annual_summary, export_for_contador, handle_language_switch
SOPs (cron trigger)	4 pipelines: watch_payment (every 5 min), monthly_reminder (daily), threshold_watch (every 6 hrs), backup_export (weekly)
Built in tools	http_request (Solana RPC, BACEN PTAX), shell (the pure Python engines below), ask_user (disposition prompts), escalate_to_human (ambiguous tax cases), file_read / file_write (the JSONL ledger)
Channels	WhatsApp Web (primary, single tenant), WhatsApp Cloud API (multi-tenant expansion), Telegram (documented fallback)
Risk profile	Supervised, explicit auto_approve / always_ask split

What had to be custom built

Four independent, pure Python packages, invoked from skills and SOPs via the stock shell tool, never a compiled plugin:

- tax_engine: Carnê-Leão brackets, weighted average cost basis, capital gains, IN 1888 threshold watch, BACEN PTAX weekend and holiday fallback. Reproduces Receita Federal's own published worked examples exactly, committed as regression tests.
- rendering: bilingual (pt-BR / English) message rendering, locale correct formatting, deterministic language switch detection, never inferred.

- trust: a deterministic guard that refuses any fund moving instruction arriving from
  anywhere other than the freelancer's own authenticated chat, regardless of phrasing.
- ledger: structural validation dataclasses. Every ledger record type is checked at construction time, so an invalid record cannot be written to disk, let alone acted on.

For the multi-tenant expansion, a small routing and billing service (referred to as the gate) was built to sit in front of ZeroClaw, since Meta allows exactly one webhook callback URL per WhatsApp Business number and ZeroClaw's own workspace isolation has no concept of a sub-tenant boundary. It resolves each sender to their own isolated agent, gates spend on a prepaid credit balance before any model call happens, and acknowledges WhatsApp's webhook immediately while the actual agent turn runs in the background.

Custody tier and threat model

T1, no signing keys, ever. Every fund adjacent action (a refund, a bond allocation) produces an unsigned transaction the freelancer reviews and signs with their own wallet. This is enforced structurally: ledger records cannot be constructed without a real confirmation timestamp, and the trust guard is a deterministic check, not an LLM judgment call.

Threat modeled: a message that is not actually from the freelancer, embedded in a transaction memo, relayed as "the client said," or arriving through any channel other than the freelancer's own authenticated chat, tries to redirect a payment or change a disposition. The guard checks the source before reasoning about the content at all. Only the freelancer's own authenticated chat is ever trusted to carry a fund moving instruction. This is proven with an automated test that reproduces the scenario exactly, not only a manual demo.

Reproduce it

Secrets redacted throughout; every credential below is set as an environment variable, never committed.

What	Link
Setup guide	SETUP.md
Config templates	config/
SOPs	sops/
Skills	shared/skills/livro
Trust guard + tests	trust/
Tax engine + tests	tax_engine/
Ledger + tests	ledger/
Multi-tenant gate	gate/
Full repository	github.com/replicolabs/livro