

Comparativo de ponta a ponta

AvivPay (coreproviders) x Core Banking

Monetarie

Produto, funcionalidade, arquitetura, banco de dados e performance, com evidência empírica da produção AvivPay (mais de 812 mil transações PIX em um dia) e recomendações priorizadas do que absorver no ecossistema Monetarie.

Data: 04 de julho de 2026

Escopo: repositório `/Users/luizpenha/coreproviders` (branch aviv-hml), AWS produção AvivPay (conta 087035285972, sa-east-1, leitura autorizada), monorepo Monetarie (core, cabines PIX/SPB, HML vivo)

Método: 100% empírico. Toda afirmação carrega âncora de evidência: arquivo:linha, métrica CloudWatch/Performance Insights, consulta Logs Insights ou screenshot.

1. Sumário executivo

A produção da AvivPay comprova, em volume real, o formato de trabalho do coreproviders. No dia 03/07/2026 (horário de Brasília) o ambiente processou **812.654 transações PIX** (673.501 recebimentos liquidados e 139.153 envios), contadas uma a uma nos logs de produção. Hoje, 04/07, até as 11h19 já eram 411.383, em ritmo de cerca de 1 milhão no dia. Esse volume roda sobre a mesma linhagem de código do nosso Core (ancestral comum Owem/fluxiq), o que torna cada padrão de escala deles diretamente aproveitável aqui.

No PIX, o nosso modelo é estruturalmente superior, e agora isso está medido dos dois lados. A AvivPay é participante indireto e recebe mensagens da cabine terceirizada CloudPix OnZ por long-poll HTTP (teto de 6 threads do BACEN, ciclo de 5s), com latência de entrada medida em produção de p50 471ms e p99 737ms. A nossa cabine própria, participante direto com mTLS ICP-Brasil e integração interna via NATS JetStream na mesma rede, lê a confirmação pacs.002 em 24 a 28ms (prova de 03/07 em HML). A vantagem é de mais de uma ordem de grandeza e não depende de terceiro no caminho do dinheiro.

Onde eles são melhores: operação sob volume. O coreproviders pagou, em produção, as dores de escala que nós ainda não pagamos: particionamento mensal de 7 tabelas, checkpoint de saldo que removeu 263ms do caminho crítico, caches ETS com invalidação distribuída, 16 filas Oban com cerca de 50 crons de reconciliação e watchdog, observabilidade PromEx + OpenTelemetry, deploy sem 5xx com drain de 90s e uma suíte de benchmark que levou o pipeline de 180 para mais de 1.100 TPS com gate de regressão. O banco deles bateu no teto de 64 ACU e 96% de CPU no pico de ontem: sabemos exatamente o que dói primeiro e como eles trataram.

812.654

PIX IN+OUT EM 03/07
(CONTADO NOS LOGS
PRD)

62.212

PICO DE PIX IN
LIQUIDADOS EM 1
HORA (21H BRT)

471ms

P50 DE ENTRADA VIA
ONZ LONG-POLL
(DELES)

24-28ms

LEITURA PACS.002 NA
NOSSA CABINE (PROVA
03/07)

Direção recomendada (detalhe na seção 8): manter o Monetarie como base canônica, inclusive a cabine PIX própria, e absorver do coreproviders o pacote de escala (particionamento de `transactions`, dedup de webhook por escrita idempotente, checkpoint de saldo, observabilidade, drain de deploy) e um conjunto curto de refinamentos de produto (fee preview no IB, portal do desenvolvedor, breakdown de saldo bloqueado, reconciliador de holds órfãos).

2. Metodologia e fontes

Fonte	O que foi coletado	Ferramenta
AWS PRD AvivPay (conta 087035285972, sa-east-1, profile de leitura autorizado pelo dono)	Topologia ECS/EC2/Aurora/Redis, volumetria por Logs Insights (14,3 GB escaneados por dia consultado), métricas CloudWatch, top queries via Performance Insights	aws cli somente leitura
Repositório coreproviders (branch aviv-hml)	Arquitetura, engenharia de escala, integração OnZ, inventário de produto dos 4 frontends, benchmarks internos	3 agentes de exploração com âncora arquivo:linha
Monorepo Monetarie	Verificação viva do que as Ondas 1-3 (02/07) já absorveram e do que segue em aberto	grep/leitura direta + HML vivo
Matriz de paridade 02/07	193 itens código a código (base pré-existente, não repetida aqui)	docs/reports/2026-07-02-paridade-core-coreproviders.md
Telas	Screenshots oficiais AvivPay do repositório + capturas vivas do admin e IB Monetarie em HML	Playwright

3. Volumetria da produção AvivPay, comprovada

Contagem direta nos logs estruturados de produção (log group `/coreproviders/aviv-prd` , 54,5 GB armazenados), dia 03/07/2026 de 00h00 a 23h59 BRT:

Evento	Padrão de log contado	Total 03/07	Hoje 04/07 até 11h19
PIX IN iniciado (pacs.008 aceito, ACSP enviado)	<code>[TbFirst] Phase 1 OK</code>	673.495	-
PIX IN liquidado (pacs.002 ACCC, depósito efetivado)	<code>[TbFirst] Phase 2: ACCC received</code>	673.501	334.612
PIX OUT enviado (DICT + pacs.008)	<code>[OnZ.Adapter] send_pix</code>	139.153	76.771
Total in + out		812.654	411.383
Mensagens processadas pelo poller (todas)	<code>[POLLER] MSG_OK</code>	1.486.310	-

Evidência: consultas CloudWatch Logs Insights executadas em 04/07 às 11h19 BRT, janela epoch 1783047600 a 1783134000, 14.286 MB escaneados por consulta, status Complete. A conversa de trabalho registra cada query e resultado bruto.

3.1 Perfil de pico e latência de processamento

Métrica (03/07)	Valor	Leitura
Pico horário de PIX IN liquidado	62.212 (21h BRT), 57.981 (20h), 56.109 (19h)	Cerca de 17,3 liquidações/s sustentadas por 1 hora no pico noturno
Latência de processamento por mensagem (poller, 1.486.310 amostras)	média 138ms, p50 99ms, p95 240ms, p99 1.055ms, máx 5,9s	<code>duration_ms</code> do <code>MSG_OK</code> ; handler médio 124ms

4. Infraestrutura e banco da produção AvivPay

4.1 Topologia (lida ao vivo via API AWS)

Componente	Configuração PRD	Observação
Backend Elixir (app :fluxiq)	6 tasks ECS (2 vCPU / 6 GB task; container 1 vCPU / 6 GB), capacity provider EC2 com 6x c7g.xlarge (arm64)	Task-def backend:194 ; deploy pinado maxPct 117 / minHealthy 66
TigerBeetle	Cluster replicado de 3 nós (serviços tigerbeetle-0/1/2), 3x m6g.xlarge	Nosso TB em HML é 1 nó EC2; para produção o modelo deles de 3 réplicas é a referência
Aurora PostgreSQL	17.9 Serverless v2, 2 a 64 ACU, writer + reader, Performance Insights ativo nas duas instâncias	Nós: Aurora 16.6 Serverless v2 0,5 a 4 ACU, instância única
Frontends (admin, banking, merchant, docs)	1 task Fargate cada	UIs são estáticas atrás de nginx, igual a nós
Borda e rede	AWS WAF público (logs de 6,6 GB), OpenZiti tunnel (t4g.small), OPNsense (t3a.small), VPC peering porta 80 com a OnZ	Nossa borda HML é interna via VPN; WAF público será tema de produção

4.2 O banco sob 812 mil transações: onde dói

Métrica do writer (03/07)	Valor	Interpretação
ServerlessDatabaseCapacity	Bateu o teto de 64 ACU em pelo menos 3 janelas horárias (2h, 5h, 10h BRT); média por hora 14 a 17 ACU	Headroom esgotado nos picos; escalar teto ou reduzir carga por query
CPUUtilization	Pico 96,1% (20h BRT), 84,3% (22h)	Coincide com o pico de PIX IN; o writer é o gargalo
DatabaseConnections	287 estáveis o dia todo	Pools fixos (100 writer + 50 reader + 10 batch por 6 tasks, com divisão writer/reader)
CommitThroughput	Pico momentâneo 19.307 commits/s (10h); média por hora 1.400 a 1.755/s	Volume transacional real do core

4.3 Top queries por carga (Performance Insights, dia inteiro 03/07)

#	Query (tokenizada)	Carga média (AAS)	% do total (2,82)	Lição para o Monetarie
1	SELECT ... FROM webhook_deliveries WHERE webhook_id=\$1 AND event_id=\$2 LIMIT 1	0,708	25,1%	O dedup de webhook por SELECT antes do INSERT é o maior consumidor do banco deles. No nosso dedup (Onda 3), usar INSERT ... ON CONFLICT com índice único e cache, nunca SELECT prévio no hot path
2	SELECT COALESCE(SUM(CASE WHEN direction=... THEN amount ...)) FROM transactions t0 WHERE to/from_account_id=\$1	0,293	10,4%	Agregação de saldo/uso on-the-fly sobre tabela grande. É exatamente a dor que o BalanceCheckpoint deles paga; nosso pg_net_available tem o mesmo formato
3	COMMIT	0,273	9,7%	Custo natural de 1,4 a 19 mil commits/s
4	autovacuum: VACUUM public.oban_jobs	0,127	4,5%	Oban em alto volume exige pruner agressivo e monitorar vacuum
5	INSERT INTO message_timeline ...	0,112	4,0%	Timeline de mensagens persistida (nós também temos; particionada lá e cá)
6-10	UPDATE webhook_deliveries; pg_notify; UPDATE oban_jobs; INSERT processed_messages ON CONFLICT DO NOTHING; INSERT webhook_deliveries	0,065 a 0,103	~15%	Webhooks + Oban + idempotência dominam o banco depois do dinheiro em si

Evidência: `aws pi get-resource-metrics` em `db.load.avg` agrupado por `db.sql_tokenized`, janela 03/07 completa, instância `coreproviders-aviv-prd-aurora-writer`.

5. PIX: cabine própria + NATS (nós) x provider OnZ long-poll (eles)

Esta é a comparação central do mandato, agora com medição dos dois lados. O modelo deles funciona (812 mil/dia comprovadas), mas paga pedágio permanente de latência, teto regulatório de canal e cegueira sobre o estado real do BACEN. O nosso elimina as três coisas.

Dimensão	AvivPay via OnZ CloudPix	Monetarie com cabine própria	Veredito
Papel no SPI	Participante indireto (ISPB 04838403 via liquidante 05684234)	Participante direto (ISPB 46026562), mTLS ICP-Brasil verify_peer, XMLDSig assinado no HSM	Monetarie
Recebimento de mensagens	Long-poll puxando da OnZ: máx 6 threads (teto BACEN SPI 2.2.2.10), slots em Redis com lease de 15s, ciclo 5s + stagger 833ms, cursor stateful por conexão TCP (pool Finch dedicada por slot)	Push direto do BACEN no canal ICOM + fan-out interno via NATS JetStream (6 streams, dedup e replay nativos)	Monetarie
Latência de entrada (medida)	p50 471ms, p95 561ms, p99 737ms (PRD 20/04, 1.256 tx/60min; anotado em onz/poller.ex:106-108); p95/p99 acima da meta BACEN de 500ms	Leitura de pacs.002 em 24 a 28ms (CreDtTm até received_at, prova de 03/07); ANS alvo do fluxo 1,6s com monitor dedicado	Monetarie, ~17x
EndToEndId	Gerado pela OnZ na consulta DICT; se não vier, o pagamento falha (dict_e2e_missing)	Gerado pela própria cabine (MessageBuilder.generate_e2e_id/1)	Monetarie
Balde antiscan DICT	Réplica local aproximada; custo das fichas inferido por status HTTP; 429 zera o balde espelhado	Estado real: falamos com o DICT do BACEN diretamente (v2.11.0)	Monetarie
Timeouts do caminho crítico	Inflados pela intermediação: pacs.002 60s, DICT lookup 2,5s com pior caso histórico de 54s	Round-trips diretos: GetEntry 143ms medido, echo 196ms	Monetarie
Robustez operacional do canal	Madura e testada: poison-message policy (5 tentativas + DLQ), inbox durável com processamento out-of-band, monitores de lag/liveness/DLQ/ok-rate por cron, fallback MGMT, re-entrega idempotente por cursor	Parcial: temos StuckOutboundChecker, sweepers e monitor WebSocket, mas não um pacote equivalente de watchdogs com poison policy e ok-rate	AvivPay; absorver
Two-phase de PIX IN	ACSP enviado, intent pendente em Redis (TTL 600s), depósito só no ACCC. Nunca deposita antes da liquidação	Mesmo princípio (herança comum); nossa liquidação é aplicada pela cabine e materializada no Core via NATS	Paridade
Instrumentação por fase	Logs PERF por etapa (POLL_START/END, MSG_RECV end2end_ms, DISPATCH, MSG_OK handler_ms) que permitiram medir tudo desta seção	Temos telemetria, mas não esse padrão de log por fase que torna a produção auto-auditável	AvivPay; absorver

Evidência âncora (deles): backend/lib/fluxiq/services/pix_providers/onz/poller.ex (slots, ciclo, latência, poison), lp_client.ex (TCP affinity, cursor ACK), endpoints/dict.ex:171-196 (timeout 2,5s), runtime.exs:809-815 (pacs.002 60s). **Evidência âncora (nossa):** memória de prova 03/07 (READ_LAG 24-28ms), handoff docs/handoff/2026-06-23-pix-bacen-native-dict-spi-cid-sync.md (GetEntry 200 em 143ms).

Nota honesta sobre NATS nos benchmarks deles: as iterações 005 a 007 da suíte de benchmark do coreproviders usaram pipeline com NATS JetStream (305 a 391 TPS) antes de migrarem o hot path para TB-First direto (659 a 1.129 TPS). O ganho deles veio de tirar o PostgreSQL do caminho quente do depósito, não de remover o NATS. Para nós a conclusão é dupla: manter NATS como espinha de integração (nosso diferencial de rede interna) e, como eles, manter o caminho do dinheiro com o mínimo de I/O síncrono possível.

6. Engenharia de escala deles que devemos absorver

Tudo abaixo foi validado por 812 mil transações/dia em produção e tem âncora no código deles. Terceira coluna: estado verificado hoje no Monetarie.

Padrão	Como funciona lá (evidência)	Estado no Monetarie (verificado 04/07)
Particionamento mensal de 7 tabelas quentes	transactions , inflow/outflow_requests, fee_transactions, fee_split_transactions, message_timeline, audit_logs: RANGE mensal, PK composta (id, started_at), ~16 índices por partição; worker PartitionMaintenance cria 3 meses à frente e confere herança de índices (cron diário 01h)	Errata pós-execução (04/07 à tarde): a transactions do mon_core JÁ É particionada por RANGE (started_at) com PK composta desde a squash migration (DDL no pg_dump, invisível a grep de migrations .exs). Os gaps reais, encontrados e corrigidos no port (commit 91c5348c): 3 tabelas particionadas fora do worker de manutenção, runway de partições com buracos (fee_split_transactions sem partição a partir de jun/2026) e colisão de nomes legados que derrubaria o worker em out/2027
BalanceCheckpoint com prefold noturno	O SUM canônico varria ~14 partições e custava 263ms por chamada sob advisory lock (65% do tempo do writer) ; checkpoint acumulado + delta do dia reduz para 1 partição; prefold às 00h10 BRT (balance_checkpoint.ex:8-11)	Ausente; nosso pg_net_available soma summaries a cada verificação (mesma família da query #2 do Performance Insights deles)
Dedup de webhook barato	Mesmo com o padrão SELECT deles custando 25% do banco, o INSERT usa ON CONFLICT; o gap deles é não ter cacheado o SELECT	Nosso dedup da Onda 3 existe; garantir formato INSERT ... ON CONFLICT + índice único + cache de vistos recentes antes do volume subir
Caches ETS com invalidação distribuída	BalanceCache display 10s + PubSub; UsageCache de limites 15min com chave por período BRT; TxStatusCache com TTL por estado (pending 5s, terminal 300s); GlobalLimitCache e FeeResolver 300s. Todos opt-in por flag, proibidos no caminho de verificação de saldo	Só GlobalLimitCache (port literal). Demais ausentes
Divisão writer/reader	BaseRepo pool 100 no writer; ReadRepo pool 50 no Aurora reader com work_mem 64MB só para analytics; BatchRepo pool 10 para bulk	Pool único no writer; sem ReadRepo. Nosso Aurora HML tem 1 instância; para produção, prever reader + ReadRepo
Advisory lock por conta + I/O fora do lock	pg_advisory_xact_lock(account_id) serializa admissão; flags movem TigerBeetle create_transfers e lookup para fora do lock com shadow-parity antes do flip (balance_check.ex:230-310)	Temos lock e pipeline staged; a disciplina de zero I/O dentro do lock com gate de paridade é o refinamento a copiar
Deploy sem 5xx	Drainer com lameduck 5s + shutdown drain 90s (override do default 15s do ThousandIsland)	Ausente; nossos deploys derrubam conexões em curso
Observabilidade	PromEx + Grafana, OpenTelemetry OTLP, spans por fase do money-path, logs PERF estruturados que permitiram TODA a medição da seção 3 sem acesso ao banco	Sem PromEx/OTel no core; a cabine tem telemetria própria mas não expõe Prometheus. É o gap que mais limita nossa capacidade de provar performance como eles provam
Suíte de benchmark com gate de regressão	16 relatórios versionados: 180 TPS baseline até 960 TPS agregado / 1.129 pico; bench-014 fixa gate de p50/p99 do hot path (±10%); bench-016 mediu flag de limite (p99 de 7.179µs para 16µs)	Ausente; nossos números vêm de provas pontuais em HML. Portar a suíte é barato (mesma linhagem de código)

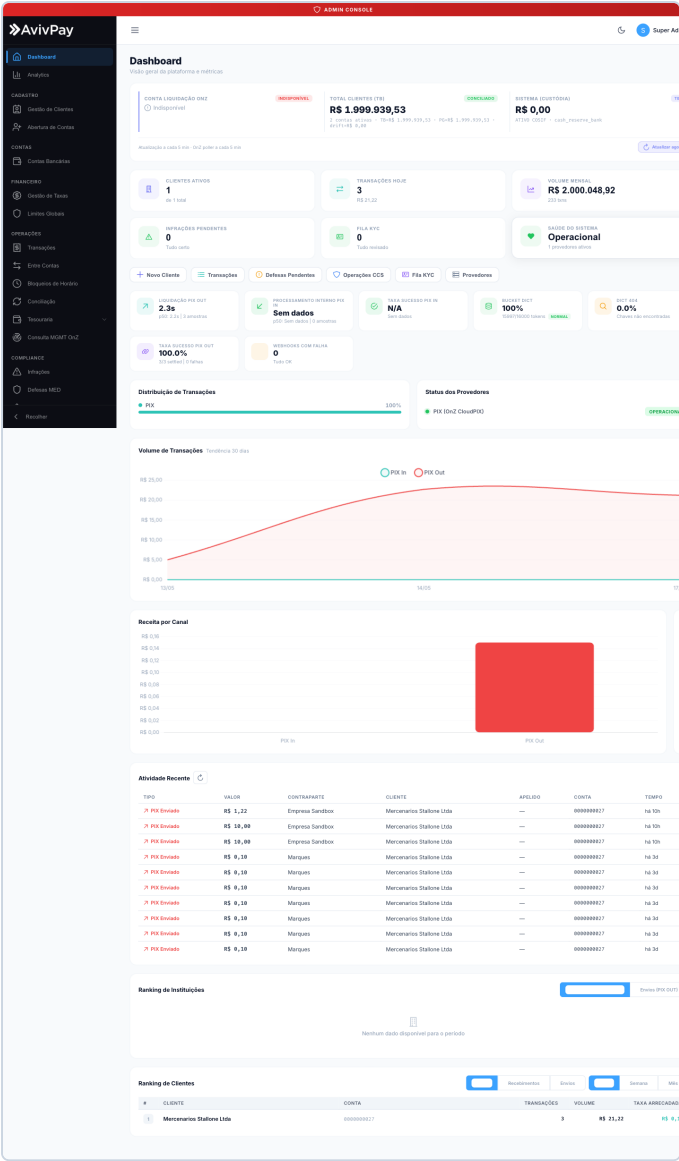
Oban tunado por domínio	16 filas nomeadas (onz_lp_process=6 alinhada ao teto BACEN, pix_in_pg_write=10, pix_out_retry=10 com snooze por bucket DICT), ~50 crons de reconciliação e watchdog, Lifeline 30min	Oban presente com crons; sem a malha de watchdogs nem filas dimensionadas por contrato externo
--------------------------------	---	--

7. Produto e funcionalidade: os dois lados com telas

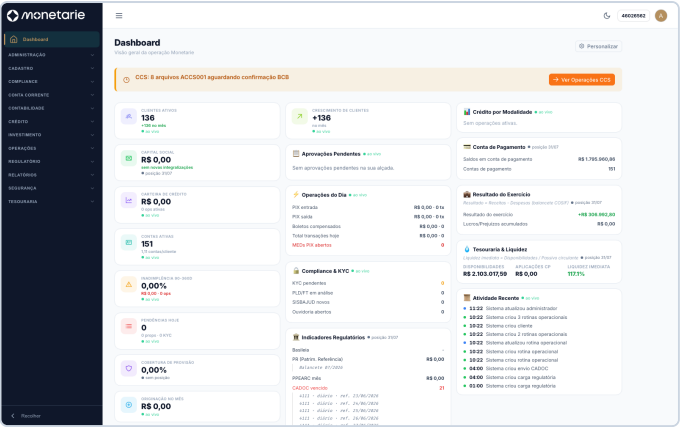
7.1 Visão geral

Domínio	AvivPay	Monetarie	Veredito
Admin console	105 rotas; Caixa Institucional com apropriação de tarifa PIX; dashboard da conta de liquidação OnZ com Chart.js e extrato consolidado; conciliação TB-PG e COSIF em dashboard único (1.822 linhas); gestão de MED com análise de defesa (2.666 linhas); quarentena de PIX OUT com decisão MFA; monitor de fila DICT e bucket dashboard	Admin SCD completo: COSIF com períodos e A=P+Resultado validado ao centavo, SISBAJUD fim a fim, e-Financeira com XMLDSig, DERE/FGCoop/Registrato, tesouraria BCB (compulsório/redesconto), CCS, overdraft com juros/IOF, pacotes de tarifas com franquia, CADOC, PLD vivo	Monetarie em profundidade regulatória; AvivPay em operação de liquidação/MED
Internet Banking	Fee preview com debounce de 350ms, breakdown tarifa + total + limite restante, bloqueio de envio quando valor + tarifa excede saldo; confirmação em 2 etapas; PIX agendado, favoritos, refund; portal do desenvolvedor (API keys, webhooks, eventos, logs) dentro do IB; delegação de acesso à conta	IB com PIX out validado vivo, comprovante com autenticação, extrato com export OFX/CSV, boleto NPC, Pix Automático; sem fee preview nem portal do desenvolvedor	AvivPay em refinamento de UX; absorver os 2 itens
API externa / parceiros	18 endpoints com HMAC-SHA512 por requisição + IP whitelist obrigatória + idempotência; expõe defesa de MED via API; docs públicas em 5 idiomas com Postman e Bruno 100%	Partner API OAuth2 client_credentials versionada com OpenAPI, escopo por parceiro, idempotência por api_key (corrigida na Onda 1); runbook de re-provisionamento pós-ETL	Paridade com estilos diferentes; docs deles são referência de DX
Mobile	Flutter whitelabel real por tenant (cores, marca, CNPJ e endpoint via dart-define); cobre PIX completo, portabilidade de chave e até defesa de MED no app	Flutter presente; rebrand nativo (ícone/splash) ainda pendente desde 06/22	AvivPay

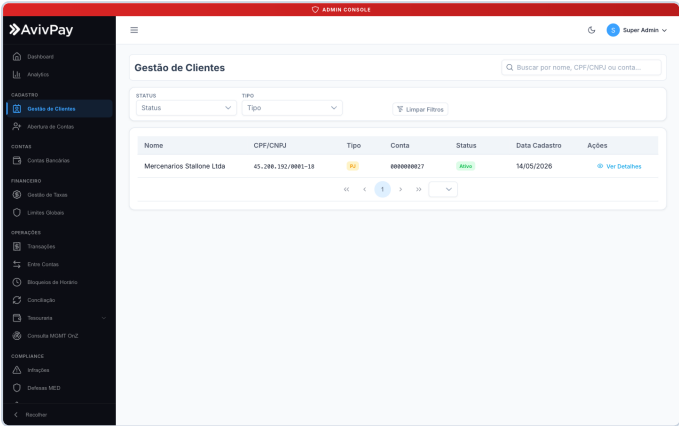
7.2 Comparativo visual (telas oficiais AvivPay x capturas vivas Monetarie HML)



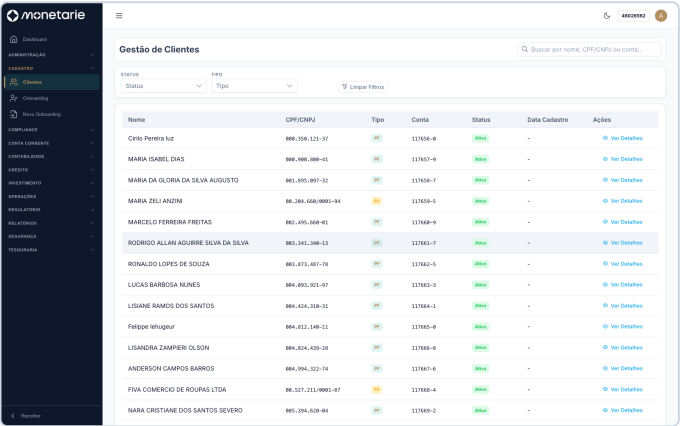
AvivPay admin: dashboard



Monetarie admin: dashboard (HML vivo, 04/07)



AvivPay admin: clientes



Monetarie admin: clientes (HML vivo)

- [Dashboard](#)
- [Análises](#)
- Contas e Saldo
 - [Contas de Clientes](#)
 - [Resumo da Contas](#)
- [Conta PJ](#)
- [Contas Bancárias](#)
- Inscrições
 - [Gestão de Tribos](#)
 - [Lançamentos](#)
 - [Transmissões](#)
 - [Bônus Contas](#)
 - [Respostas do Hotmail](#)
 - [Comissão](#)
 - [Transferência](#)
 - [Cobrança MGMT Out](#)
- [Conveniente](#)
- [Infâncias](#)
- [Canais MDI](#)
- [Recursos](#)

Assim Amarelo

Super Admin

Transações

CLIENTE	Tipo	Status	Data Inicial	Data Final
VALOR MIN (R\$)	VALOR MAX (R\$)			
Min	Max			

☒ Lançar Filtros

ID	Tipo	Valor	Status	Cliente	Data	Ações
7ab0801ac261...	Pix Rec.	- R\$ 1,22	Concluída	Mercenários Statione Ltda	18/05/2026 10:39:30 GMT-3 BRT	
e0f0800c1935...	Pix Rec.	- R\$ 10,00	Concluída	Mercenários Statione Ltda	18/05/2026 10:39:00 GMT-3 BRT	
e0f08015a376...	Pix Rec.	- R\$ 10,00	Concluída	Mercenários Statione Ltda	18/05/2026 10:38:05 GMT-3 BRT	
8a0f087d0826...	Pix Rec.	- R\$ 10,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:01 GMT-3 BRT	
8a0f080a0372...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:01 GMT-3 BRT	
8a0f080a05ae...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:01 GMT-3 BRT	
8a0f080a0602...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:01 GMT-3 BRT	
8a0f080a063f...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:01 GMT-3 BRT	
8a0f081223ad...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:01 GMT-3 BRT	
8a0f081c7026...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f081c8505...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f081c9a72...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f081d8ea7...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f080a0594...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f08020a45...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f080a0609...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f080a056a...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f0801173b...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f080f2a67...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	
8a0f080a4527...	Pix Rec.	- R\$ 0,30	Concluída	Mercenários Statione Ltda	15/05/2026 12:46:00 GMT-3 BRT	

<<

1

2

3

4

5

>>

monetarie

Dashboard

ADMINISTRAÇÃO

SALES

COMPANHIA

COMPANHIA PIX

Contas Correntes

Contas Correntes MTD

Contas Correntes MTD

Contas Correntes PIX

DECT Faturar

Infrações

Multações e DCT

Obrigações CDB

PIX MGMT

PIX OUT em Trânsito

PLUGIT

PIX Lotes

Quotação PIX

Regras de Contas

Relatório Compliance

Relatório PIX

CONTA CORRENTE

CONTABILIDADE

SALES

SALES REPORT

OPERAÇÕES

COMPANHIA

COMPANHIA PIX

Compliance PIX

Realizar

INFRAÇÕES ATIVAS

0

CHARGEBACKS

0

BLOQUEIOS CAUTELARES

0

TOTAL DE COMPLIANCE

100.0%

Eventos

Infrações

Chargebacks

Bloqueios Cautelares

ID	Tipo	Cliente	Status	Valor	Data 1
10 1 2 31					

- Desenvoltos
- ALERT
- Compartilhamento
- Oportunidades CCE
- REGULAMENTO
- SOLICITAÇÃO
- C-Financiamiento
- Reservatório CADUCO
- DMSA
- INFORMACIÓES
- Gestão Financeira
- Métricas
- ADMINISTRACIÓES
- Tela de Acesso
- Página de Anúncio
- Procedimentos
- Visualizar Cliente
- RECURSOS
- IP Whitelist**
- API Keys
- Recursos

ADMIN CONTROL

Super Admin

IP Whitelist

Gerencie as regras do IP permitidas para acesso ao sistema. Regras podem ser sincronizadas com o Cloud Armor.

Modo de Enforcimento do /api/admin

Seu IP detectado 192.168.144.83
Está na whitelist admin? **Sim**

Total whitelisted admin CIDRS
Enforcement plug-in **active**
Cloud Armor border **inactive**

[Ativar enforcimento](#)

Adicione seu IP (192.168.144.83) como escopo antes de ativar enforcimento, para evitar self-lockout.

Filtrar por escopo

Atualizar

Verificar IP

Nova Regra

TOTAL DE REGRAS	ATIVAS	INATIVAS/EXPIRADAS	ESCOPO AM
0	0	0	0

CIDR / IP '1	Descrição	Escopo '1	Status '1	Expira em '1	Criado por	Criado em '1	Ações
							CC ← →

monetarie

Dashboard

ADMINISTRACAO

CONFIGURACAO

USUARIOS

LOGS

INVESTIMENTOS

OPERACOES

RECALCULACAO

RELAZARIOS

RECONSTRUCAO

API Keys

Configuracoes PDI

Monitor

RECONSTRUCAO

API Keys dos Clientes

Buscar por nome, cliente, conta ou api

Status

3 items

Cliente		Nome da Key	Client ID	Permissões	Status	IPs	Último Uso	Criado em	Ações
+		Herbert Santana - chave homolog	<11_75685f4b8356da4799a>	api:read api:write api:delete api:create api:manage	Ativo	3 IPs	-	03/07/2026 21:37:45 GMT-3 BRT	
				api:read api:write api:delete api:create api:manage					
+		chave: @dev-integracao-homolog	<11_551981341e719d743547576>	api:read api:write api:delete api:create api:manage	Ativo	3 IPs	-	03/07/2026 21:36:38 GMT-3 BRT	
				api:read api:write api:delete api:create api:manage					

1

20

AvivPay

Home

PIX

Transferências

Extrato

Operações em Taboão

Comprovantes

CONTAS

Contas Bancárias

Entre Contas

DOCUMENTOS

Extratos

Saldo

Declarações

Transferências

INDICAÇÃO

QR Codes

Open Finance

COMPLIANCE

Compliance

Recalculador

Conta de Pagamento - 8000000027

Mercenários Sta. Lúcia

Ag. 0001

Conta 0000000027

Conta de Pagamento

Boa noite, Mercenários

Mercenários Sta. Lúcia

Entradas do mês

Saldo disponível

Saldos do mês

R\$ 1.000.000,00

R\$ 999.939,53

R\$ 48,92

PIX

Extrato

Receber PIX

Favoritos

Comprovantes

Análisis

MEIs

Taxas

Limites

Ver todos →

Transações recentes

Empresa Sandbox

PIX Envia

Concluído

há 10 horas

- R\$ 1,22

Taxa: R\$ 0,00

Empresa Sandbox

PIX Envia

Concluído

há 10 horas

- R\$ 10,00

Taxa: R\$ 0,00

Empresa Sandbox

PIX Envia

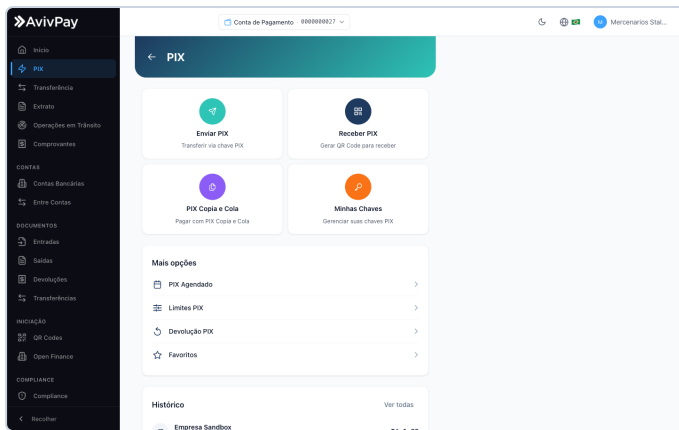
Concluído

há 10 horas

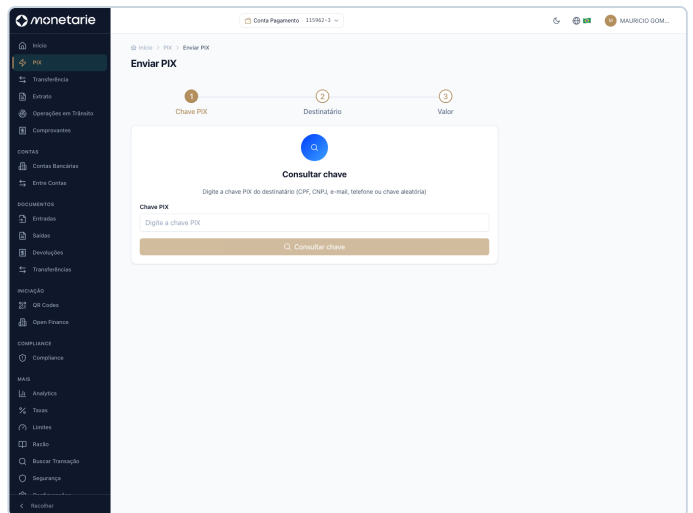
- R\$ 10,00

Taxa: R\$ 0,00

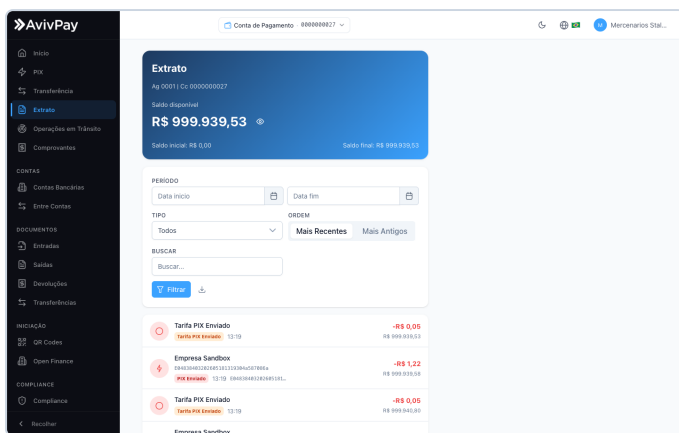
[illegible]



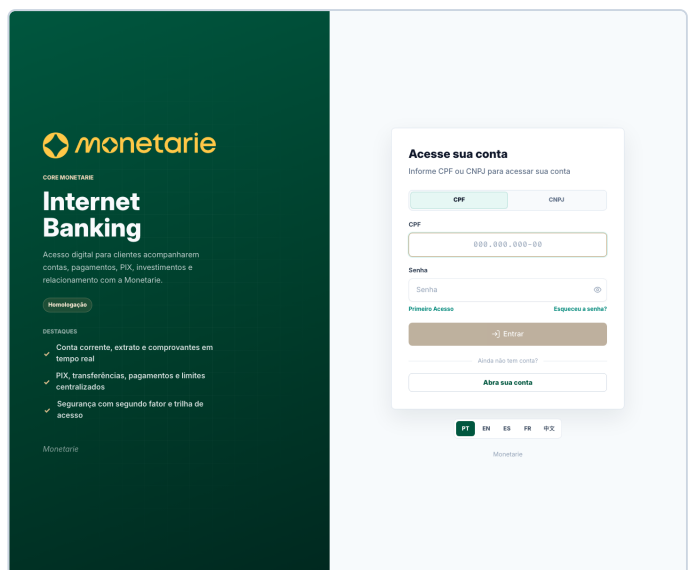
AvivPay IB: área PIX



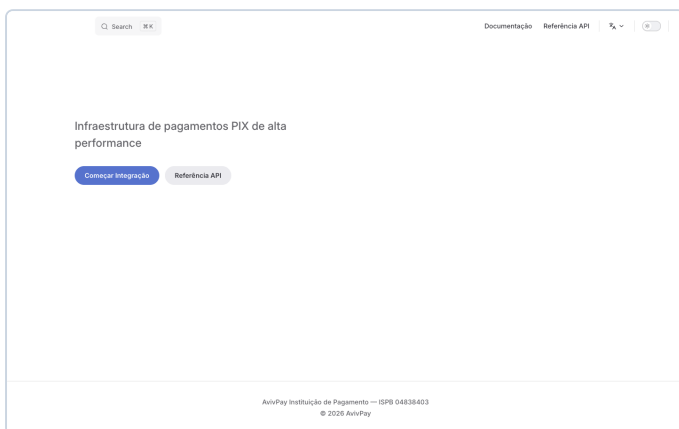
Monetarie IB: envio PIX por chave (evidência 28/06)



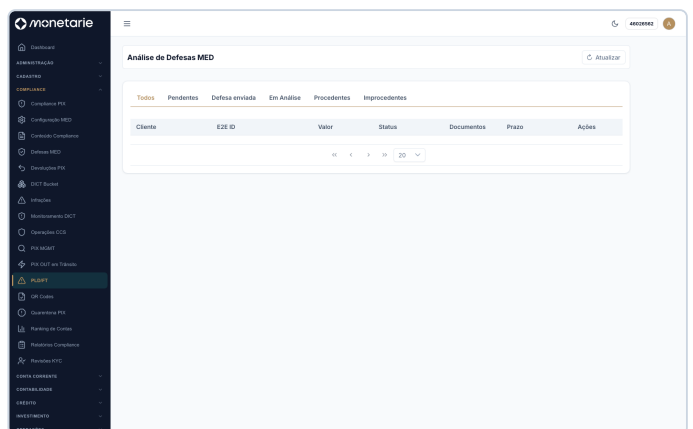
AvivPay IB: extrato



Monetarie IB: comprovante PIX com autenticação (evidência 28/06)



AvivPay: docs públicas da API (5 idiomas)



Monetarie admin: análise de defesa MED (HML vivo)

8. O que já absorvemos e o que recomendo absorver agora

8.1 Já absorvido nas Ondas 1-3 (02/07), verificado hoje no código

- Limites transacional/diário/mensal e janela de horário no PIX OUT e TED:
`core/backend/lib/monetarie/use_cases/payments/outbound_orchestrator.ex:40-150` (LimitCheck + TimeBlockCheck no gate).
- CRC-16 do parser BR Code corrigido (o bug do 6304 duplicado): `use_cases/pix/brcode/parser.ex:30-32`.
- Idempotência de parceiro com escopo por `api_key` (fim do replay cruzado):
`monetarie_web/plugs/idempotency.ex:81-110`.
- PLD vivo pós-liquidação: `workers/compliance/transaction_monitor_worker.ex` chamando `evaluate_transaction`.
- Webhooks com dedup e fail-closed, devolução automática pacs.004, transferência interna mesmo-ISP, polling DICT e timers de claim (handoff 02/07).

8.2 Recomendações priorizadas (cada uma com a dor comprovada que a justifica)

Prio	Recomendação	Justificativa empírica	Esforço
P0	Fechar o stub do export de tarifas do admin (responde "ready" falso com UUID fake) portando o pipeline real Oban + S3 deles, ou remover a tela	Verificado hoje: coreproviders_parity_controller.ex:1183,1309 . Honestidade de UI é regra da casa	Baixo
P1	EXECUTADO 04/07 com errata: a transactions já era particionada (ver seção 6); o trabalho real foi blindar o worker PartitionMaintenance (3 tabelas fora da lista, runway 2026-2028 com buracos fechados, colisão de nomes legados resolvida por checagem de range). Commit 91c5348c	A dor da AVIV continua válida como referência: a query #2 do PI deles (SUM sobre transactions) custa 10% do banco MESMO particionada	Feito
P1	Blindar o dedup de webhook: índice único (webhook_id, event_id), INSERT ... ON CONFLICT, cache curto de vistos	Maior consumidor do banco deles: 25,1% da carga total do writer em 03/07	Baixo
P1	Observabilidade: PromEx + OpenTelemetry no core e na cabine, e logs PERF por fase no padrão deles (MSG_OK handler_ms, end2end_ms)	Toda a seção 3 deste relatório foi possível pelos logs deles; hoje nós não conseguiríamos provar o equivalente sem acesso ao banco	Médio
P1	Malha de watchdogs do canal PIX adaptada ao nosso ICOM/NATS: poison-message policy, monitor de lag fim a fim, liveness do consumer, profundidade de DLQ, ok-rate	O pacote deles (5 monitores + poison policy) é o que mantém 812 mil/dia sem operador; nós temos peças, não a malha	Médio
P1	TigerBeetle em 3 réplicas para produção	PRD deles roda tigerbeetle-0/1/2 em 3x m6g.xlarge; nosso TB é 1 nó (aceitável em HML, não em produção)	Médio (infra)
P1	Produto: breakdown de saldo bloqueado cruzando TB pending + reconciliador de holds MED órfãos	Itens da matriz 02/07 ainda abertos; o reconciliador cobre modo de falha que herdamos do mesmo ancestral	Baixo
P2	BalanceCheckpoint + prefold noturno quando o volume do Core subir	263ms/chamada e 65% do writer foi o custo deles sem isso; nosso formato de query é o mesmo	Médio
P2	Caches opt-in (BalanceCache display 10s, UsageCache 15min, TxStatusCache por estado) com a mesma regra deles: nunca no caminho de verificação de saldo	Padrão comprovado; flags permitem ligar sob medida	Baixo
P2	Graceful drain nos deploys ECS (lameduck + drain 90s) e divisão ReadRepo quando houver Aurora reader	Deploys deles são zero-5xx com 6 tasks pinadas; nossos derrubam in-flight	Baixo
P2	UX do IB: fee preview com debounce e breakdown de tarifa; portal do desenvolvedor no IB	Diferencial visível de produto deles com custo baixo (temos os endpoints de tarifa)	Baixo
P2	Suíte de benchmark com gate de regressão (portar a deles, mesma linhagem)	Foi ela que levou o pipeline deles de 180 a 1.129 TPS com prova a cada passo	Médio

8.3 Regras de port (inegociáveis)

- **Unidade monetária:** o coreproviders usa subcentavos (1 BRL = 10.000) no fio; o contrato Monetarie é centavos na borda. Nenhum código de valores pode ser copiado sem conversão (mesma família do bug de comprovante 100x já corrigido).
- **NATS permanece:** a integração Core-cabines via NATS JetStream é nosso diferencial de latência e desacoplamento na mesma rede; nada do modelo OnZ substitui isso.
- **Cabine própria permanece:** os números da seção 5 encerram a discussão; provider terceirizado custa 17x mais latência e cegueira regulatória.
- **TigerBeetle nunca é resetado** e flags de conta TB são imutáveis; qualquer equalização é por transferência compensatória.

9. Anexo: trilha de evidências

- Identidade AWS de leitura: conta 087035285972, user luizpenha, região sa-east-1 (sts get-caller-identity em 04/07).
- Consultas Logs Insights (log group `/coreproviders/aviv-prd`): contagens Phase 1 OK / Phase 2 ACCC / send_pix, percentis de duration_ms, histograma horário; janela 03/07 BRT epoch 1783047600-1783134000 e 04/07 parcial; ~14,3 GB escaneados por consulta.
- CloudWatch: ServerlessDatabaseCapacity, CPUUtilization, DatabaseConnections, CommitThroughput do writer, período 3600s.
- Performance Insights: db.load.avg por db.sql_tokenized, dia inteiro 03/07.
- ECS/EC2/RDS describe-* para topologia e sizing (backend:194, 6 tasks, c7g.xlarge, TB 3 nós m6g.xlarge, Aurora 17.9 2-64 ACU).
- Código coreproviders: âncoras arquivo:linha citadas em cada seção (poller.ex, lp_client.ex, balance_checkpoint.ex, partition_maintenance.ex, balance_check.ex, delivery_job.ex, endpoints/*, benchmark/docs/bench-result-001..016).
- Código Monetarie: verificações de 04/07 em outbound_orchestrator.ex, brcode/parser.ex, plugs/idempotency.ex, transaction_monitor_worker.ex, migrations de particionamento, coreproviders_parity_controller.ex.
- Screenshots: docs/reports/screenshots/2026-07-04-aviv-comparativo/ (9 telas oficiais AvivPay do repositório + 8 capturas vivas Monetarie).

Relatório interno Monetarie. Gerado em 04/07/2026 a partir de coleta empírica de mesma data. Não contém segredos, credenciais ou dados pessoais de clientes. Os dados de produção AvivPay foram acessados em modo somente leitura com autorização expressa do responsável.