

Manual Definitivo de Engenharia de Dados Financeiros e Conciliação Avançada no Ecosistema Mercado Pago

Introdução: O Imperativo da Integridade Financeira em Escala

No cenário contemporâneo do comércio digital, a integridade dos dados financeiros transcende a mera necessidade contábil; ela se estabelece como o pilar central da sustentabilidade operacional e da conformidade fiscal. À medida que as transações migram para ambientes digitais complexos, a intermediação financeira realizada por gateways de pagamento, como o Mercado Pago, introduz camadas de abstração que desafiam os sistemas tradicionais de Planejamento de Recursos Empresariais (ERPs). O ato de vender não se encerra mais na aprovação do cartão de crédito; inicia-se ali uma jornada de dados que atravessa estados de liquidação, retenções tributárias, disputas de consumidores e consolidação bancária.

Este relatório técnico tem como objetivo desconstruir, com profundidade granular, a arquitetura lógica necessária para sincronizar os registros do Mercado Pago com os livros-razão de um ERP corporativo. Não se trata apenas de garantir que o valor da venda corresponda ao valor recebido, mas de rastrear a genealogia de cada centavo, desde a intenção de compra, passando pela captura, até a efetiva disponibilização dos fundos e sua transferência bancária. A análise aqui apresentada rejeita a superficialidade das validações simples em favor de uma abordagem de "engenharia forense" dos dados, explorando as nuances dos relatórios de liquidação (Settlement Reports), a complexidade dos estados de *chargeback*, a normalização de identidades de pagadores e a matemática precisa do cálculo de recebíveis líquidos.

A premissa fundamental que guiará esta investigação é que a "verdade financeira" reside na interseção precisa entre os logs de eventos transacionais (Webhooks e APIs de Consulta) e os relatórios financeiros consolidados (Relatórios de Liberações). A falha em alinhar essas duas fontes de verdade resulta não apenas em "furos" de caixa, mas em passivos fiscais ocultos e vulnerabilidades operacionais críticas. Através da análise exaustiva da documentação técnica e das práticas de mercado, estabeleceremos os protocolos definitivos para uma integração resiliente, auditável e à prova de falhas.

1. Fundamentos Arquiteturais da Conciliação

Financeira

A conciliação financeira eficiente exige, antes de qualquer linha de código, uma compreensão filosófica sobre a natureza dos dados que fluem entre o adquirente (Mercado Pago) e o sistema de gestão (ERP). Existem dois fluxos paralelos que, embora correlacionados, possuem ciclos de vida, identificadores e finalidades distintos: o fluxo operacional e o fluxo de liquidação.

1.1 A Dicotomia Operacional e Financeira

O primeiro erro comum em integrações de ERP é tratar a notificação de pagamento aprovado como uma confirmação de receita contábil. No ecossistema do Mercado Pago, um pagamento aprovado (status: approved) e creditado (status_detail: accredited) ¹ é um sinal verde para a operação logística, indicando que o risco de crédito foi mitigado e que a mercadoria pode ser liberada. No entanto, do ponto de vista estritamente contábil e financeiro, o dinheiro ainda não existe na conta bancária da empresa; ele reside em um "limbo" virtual, sujeito a regras de liberação (D+0, D+14, D+30), taxas de intermediação e possíveis bloqueios por disputas.

Portanto, a arquitetura de conciliação deve ser bifurcada:

1. **Conciliação Operacional (Tempo Real):** Focada na velocidade. Utiliza Webhooks e consultas à API `/v1/payments/{id}` para atualizar o status do pedido e garantir a experiência do cliente. O objetivo é a liberação do pedido.
2. **Conciliação Financeira (Batch/Assíncrona):** Focada na precisão e na auditoria. Utiliza o Relatório de Liberações (Released Money Report) e a API de Extratos para confirmar o crédito efetivo, deduzir custos e realizar a baixa de títulos a receber. O objetivo é o fechamento de caixa e a conformidade fiscal.

Esta separação permite que o ERP mantenha a agilidade operacional sem comprometer a rigidez contábil. Um pedido pode estar "Pago" no módulo de Vendas, mas o título correspondente no módulo Financeiro permanecerá em "Aberto" ou "Transito" até que o evento de liquidação (settlement) seja processado dias ou semanas depois.

1.2 A Soberania dos Identificadores Únicos

A pedra angular de qualquer estratégia de *match* de dados é a persistência e a integridade dos identificadores únicos. Sem uma chave primária confiável que atravesse os sistemas, a conciliação degrada-se para métodos probabilísticos propensos a erros. A documentação técnica do Mercado Pago fornece três identificadores críticos que devem ser mapeados obrigatoriamente no esquema de banco de dados do ERP.

1.2.1 external_reference: O Elo Perdido

O campo `external_reference` é, sem dúvida, o atributo mais vital para a sanidade da

conciliação.² Ele representa a "Chave Dourada" que conecta o mundo externo ao universo do Mercado Pago. Diferente do ID da transação gerado pelo gateway, o `external_reference` é controlado pelo integrador.

- **Melhores Práticas:** O ERP deve popular este campo com o ID único e imutável do pedido ou da fatura no momento da criação da preferência de pagamento (`/checkout/preferences`) ou do pagamento direto.
- **Risco de Truncamento:** É imperativo notar que este campo é uma String de até 255 caracteres.² No entanto, práticas recomendadas sugerem o uso de IDs alfanuméricos curtos e robustos, evitando caracteres especiais que possam sofrer *encoding* incorreto em relatórios CSV.
- **Imutabilidade:** Uma vez definido na criação do pagamento, o `external_reference` viaja com a transação por todo o seu ciclo de vida, aparecendo nos detalhes do pagamento, nos Webhooks e, crucialmente, nas colunas dos relatórios financeiros.² Se o ERP falhar em enviar este dado, a conciliação automatizada torna-se exponencialmente mais complexa.

1.2.2 id (Payment ID) e SOURCE_ID

O identificador numérico único gerado pelo Mercado Pago para cada intenção de pagamento (ex: 1234567890) é a fonte definitiva de verdade dentro da plataforma. Nos relatórios financeiros e de liberações, este ID é transmutado para a coluna `SOURCE_ID`.²

- **Estratégia de Persistência:** O ERP deve capturar e armazenar este ID imediatamente no primeiro *handshake* de sucesso (seja na resposta síncrona da criação do pagamento ou no primeiro Webhook recebido).
- **Função de Redundância:** Nos casos em que o `external_reference` falha ou vem vazio (como em certas transferências de dinheiro ou pagamentos legados²), o `SOURCE_ID` torna-se a única âncora para rastrear o pagamento. O ERP deve ser capaz de realizar buscas reversas: dado um `SOURCE_ID` encontrado em um relatório financeiro, qual é o pedido interno correspondente?

1.2.3 Identificadores de Agrupamento (order_id)

Em arquiteturas que utilizam o conceito de "Carrinho de Compras" ou "Ordens Comerciais" (Merchant Orders ou Instore Orders para QR Code), surge o `order_id`.³ É crucial distinguir semanticamente este ID. Um único `order_id` pode estar associado a múltiplas tentativas de pagamento (transações falhas, tentativas de fraude, sucesso eventual). A conciliação financeira ocorre no nível da *transação* (o pagamento individual), enquanto a conciliação logística ocorre no nível da *ordem*. O ERP deve modelar essa relação de um-para-muitos (1 Ordem -> N Pagamentos) para evitar duplicidade de receita ou baixas incorretas.

2. Estratégias Avançadas de 'Match' de Dados e

Ingestão

A ingestão de dados para conciliação não pode ser passiva. O ERP deve adotar uma postura ativa e defensiva, assumindo que falhas de rede, *timeouts* e inconsistências temporais ocorrerão. A robustez da conciliação depende da implementação de uma estratégia híbrida de captura de dados.

2.1 A Tríade da Ingestão: Webhooks, Polling e Relatórios

Nenhuma fonte de dados única é suficiente para garantir 100% de integridade. Uma arquitetura resiliente combina três vetores:

1. **Webhooks (Event-Driven):** São os gatilhos de tempo real. Eles informam mudanças de estado (`payment.created`, `payment.updated`, `chargebacks.new`).⁴
 - *Limitação:* Webhooks podem falhar na entrega. O servidor do ERP pode estar *offline*, ou a requisição pode sofrer *timeout*. O Mercado Pago possui política de retentativa, mas não se deve confiar cegamente na entrega sequencial ou garantida.
 - *Aplicação:* Gatilho para iniciar o processo, não para finalizá-lo.
2. **API de Busca e Consulta (Polling de Recuperação):** Para mitigar a perda de Webhooks, o ERP deve implementar rotinas de *sweeping* (varredura).
 - *Endpoint:* `/v1/payments/search`.⁵
 - *Estratégia:* Periodicamente (ex: a cada hora), o ERP busca todos os pagamentos modificados no intervalo (`range=date_last_updated`) para capturar atualizações silenciosas que não dispararam ou falharam no Webhook.
 - *Filtros:* O uso de filtros por `external_reference` permite verificar o status de pedidos específicos que estão "pendentes" há muito tempo no ERP, forçando uma atualização de status.⁵
3. **Relatórios de Liquidação (Source of Truth):** O Relatório de Liberações (Released Money Report) é a autoridade final.
 - *Natureza:* Imutável e auditável. Diferente da API, que reflete o estado atual volátil, o relatório reflete o que foi efetivamente escrito no livro-razão financeiro.
 - *Aplicação:* É aqui que o *match* financeiro definitivo ocorre. O ERP processa este arquivo (geralmente CSV ou JSON) para validar se o dinheiro previsto realmente entrou.⁶

2.2 Algoritmos de Matching Determinístico e Heurístico

O "coração" do módulo de conciliação é o algoritmo que decide se a transação Tx_123 do Mercado Pago corresponde à Fatura Fat_ABC do ERP. Recomendamos uma abordagem em cascata:

Nível 1: Matching Determinístico Forte (Hard Match)

A primeira tentativa de vínculo deve ser exata e baseada em chaves primárias.

- **Lógica:** Buscar no banco de dados do ERP um pedido onde `pedido.id_externo == mp.external_reference`.
- **Validação Secundária:** Se encontrado, verificar se `pedido.id_transacao_gateway == mp.id` (caso já tenha sido capturado anteriormente).
- **Resultado:** Se positivo, o vínculo é automático e seguro. Taxa de confiança: 100%.

Nível 2: Matching Determinístico Reverso

Se o `external_reference` estiver ausente no arquivo de retorno (um cenário possível em certos tipos de fluxo, como pagamentos via link genérico), o sistema deve tentar o vínculo pelo ID da transação.

- **Lógica:** Buscar no ERP um pedido onde `pedido.id_transacao_gateway == mp.source_id` (vindo do relatório) ou `mp.id` (vindo da API).
- **Pré-requisito:** Isso exige que o ERP tenha salvo o ID do Mercado Pago no momento da criação ou na primeira notificação de status pending.

Nível 3: Matching Heurístico (Soft Match)

Quando os identificadores fortes falham (ex: perda de dados, migração de sistema legado, erro manual de cadastro), o ERP deve recorrer à probabilidade.

- **Parâmetros de Correspondência:**
 - **Janela Temporal:** A `date_created` da transação deve estar dentro de um intervalo tolerável (ex: ± 2 horas) da data de criação do pedido no ERP. *Atenção aos fusos horários.*
 - **Valor Monetário:** O `transaction_amount` deve ser idêntico ao valor total do pedido.
 - **Dados do Pagador:** Comparação de `payer.email` ou os últimos 4 dígitos do cartão (`card_initial_number` ou similar).²
- **Resultado:** Este *match* nunca deve ser automático. O sistema deve sinalizar estas transações como "Sugestão de Conciliação" para revisão humana ou aprovação em lote.

2.3 Idempotência e Tratamento de Duplicidades

Em sistemas distribuídos, a entrega de mensagens "pelo menos uma vez" (at-least-once delivery) pode resultar em notificações duplicadas para o mesmo evento. O processamento ingênuo dessas duplicatas pode causar desastres contábeis, como duplicar a receita ou realizar baixas redundantes.

- **Mecanismo de Proteção:** O ERP deve implementar uma verificação de idempotência na entrada. Antes de processar qualquer atualização de status ou linha de relatório, verifica-se: "Este estado já foi processado para este ID?".
- **Uso de X-Idempotency-Key:** Para operações de escrita iniciadas pelo ERP (como criar um estorno), o uso do header `X-Idempotency-Key` é obrigatório.⁷ O ERP deve gerar um UUID único para cada intenção de operação financeira. Se a requisição falhar por *timeout*, o ERP pode reenviar a mesma requisição com a mesma chave, garantindo que o

Mercado Pago não execute a operação duas vezes (ex: não cobrar o cliente duas vezes).

3. Análise Detalhada e Normalização de Dados Financeiros

A captura do valor bruto da venda (transaction_amount) é apenas o início da história financeira. Para uma contabilidade precisa, o ERP deve atuar como um prisma, decompondo esse feixe de luz único em seus espectros componentes: valor líquido, taxas de comissão, custos de financiamento, custos logísticos e encargos tributários.

3.1 A Anatomia do Valor Líquido (Net Received Amount)

A pergunta fundamental do departamento financeiro é: "Quanto dinheiro efetivamente entrou no caixa?". A resposta não é trivial e exige a aplicação de uma fórmula de normalização sobre os dados do Relatório de Liberações. Baseando-se nas colunas documentadas ², a equação de liquidação deve ser modelada no ERP da seguinte forma:

$$\$V_{\text{líquido}} = V_{\text{bruto}} - (\text{Taxa}_{\text{Adquirente}} + \text{Taxa}_{\text{Financiamento}} + \text{Taxa}_{\text{Logística}} + \text{Impostos}) + \text{Ajustes}$$

Vamos analisar cada componente desta equação com base nas especificações técnicas do Mercado Pago:

- **$\$V_{\text{líquido}}$ (SETTLEMENT_NET_AMOUNT / NET_CREDIT):** Este é o valor final creditado no saldo disponível. O ERP deve usar este valor para a conciliação bancária futura.
- **$\$V_{\text{bruto}}$ (TRANSACTION_AMOUNT / GROSS_AMOUNT):** O valor nominal da venda pago pelo consumidor.
- **$\$Taxa_{\text{Adquirente}}$ (MP_FEE_AMOUNT / MKP_FEE_AMOUNT):** Representa a comissão base do Mercado Pago e/ou do Marketplace. O relatório pode agregar isso ou separar o IVA. O ERP deve classificar isso como "Despesa de Venda".
- **$\$Taxa_{\text{Financiamento}}$ (FINANCING_FEE_AMOUNT):** Se o vendedor optou por oferecer parcelamento sem juros (absorvendo o custo), este valor aparecerá aqui.² Contabilmente, isso é uma "Despesa Financeira", distinta da comissão de venda.
- **$\$Taxa_{\text{Logística}}$ (SHIPPING_FEE_AMOUNT):** Refere-se aos custos do Mercado Envio. É crucial notar que este valor é deduzido da venda. Em alguns casos, o frete é pago pelo comprador e apenas "passa" pela conta; em outros, é subsidiado (Frete Grátis). O ERP deve ler este campo para conciliar a conta de "Fretes a Pagar".
- **$\$Impostos$ (TAXES_AMOUNT):** Este é um campo agregador crítico que exige tratamento especial (ver seção 3.2).
- **$\$Ajustes$ (COUPON_AMOUNT):** Descontos aplicados via cupom. A complexidade aqui reside na origem do subsídio. Se o cupom é custeado pelo Mercado Pago (coupon_fee

favorável ao vendedor), ele pode não reduzir o líquido da mesma forma que um desconto do vendedor. O campo `EFFECTIVE_COUPON_AMOUNT` pode indicar o custo real para o vendedor.¹¹

3.2 Tratamento Granular de Impostos e Retenções

A coluna `TAXES_AMOUNT` no relatório consolidado é insuficiente para a conformidade fiscal rigorosa, especialmente em países com sistemas tributários complexos como Brasil (ICMS, PIS, COFINS, ISS) e Argentina (IIBB, IVA). O Mercado Pago atua como agente de retenção na fonte.

- **Decomposição do JSON:** O ERP não deve apenas ler o total. É mandatório processar o detalhamento fornecido frequentemente em colunas JSON auxiliares (`TAXES_DISAGGREGATED` ou `TAX_DETAIL`) ou linhas específicas do relatório.¹⁰
- **Códigos de Imposto:** Os relatórios detalham tipos específicos de retenção, como `tax_payment_iibb` (Imposto sobre Receita Bruta - Argentina) ou `tax_withdholding`.¹²
- **Impacto Contábil:** Cada um desses códigos deve ser mapeado para uma conta contábil específica no ERP (ex: "Impostos a Recuperar" ou "Adiantamento de Impostos"). Se o ERP ignorar esse detalhe e lançar tudo como "Despesa", a empresa perderá créditos fiscais valiosos.

3.3 A Questão Crítica dos Fusos Horários (Timezone)

Um dos maiores geradores de discrepâncias em fechamentos mensais é o desalinhamento temporal. Uma venda realizada às 23:00 do dia 31 de janeiro pode ser registrada no ERP em Janeiro, mas aparecer no relatório do Mercado Pago em Fevereiro se os fusos horários não estiverem sincronizados.

- **Padrão UTC vs. Local:** A API do Mercado Pago geralmente opera em UTC (ISO 8601) ou com *offsets* explícitos.⁵ No entanto, os Relatórios de Liberações gerados podem vir configurados com um fuso horário padrão (ex: GMT-4 para América Latina) dependendo da configuração da conta ou do parâmetro `display_timezone` na geração via API.¹³
- **Estratégia de Normalização:** A prática recomendada para engenharia de dados é normalizar **tudo para UTC** na camada de ingestão.
 1. Ao ler a API, converta `date_created` para UTC antes de salvar.
 2. Ao solicitar relatórios via API, especifique explicitamente o fuso horário desejado ou esteja ciente do padrão da conta.
 3. Ao comparar datas para *matching*, aplique a conversão. Ignorar o *offset* de -3h ou -4h pode mover milhares de transações para o dia errado, quebrando a conciliação diária.

4. Gestão de Exceções: O Ciclo de Vida de Estornos e

Disputas

O "Caminho Feliz" (venda aprovada e liquidada) é fácil de automatizar. A robustez da integração é testada nas exceções: cancelamentos, reembolsos e, o mais temido de todos, o *chargeback*. A lógica do ERP deve ser uma máquina de estados capaz de transitar os pedidos para frente e para trás com base nesses eventos.

4.1 Distinção Técnica: Cancelamento vs. Reembolso

Embora ambos resultem na devolução do dinheiro ao cliente, eles são tecnicamente distintos na API e possuem impactos contábeis opostos.¹⁵

4.1.1 Cancelamento (Cancellation)

- **Definição:** Ocorre quando o pagamento ainda não foi capturado ou liquidado financeiramente (status pending ou in_process). É um "void".
- **Mecanismo:** A transação é anulada antes que o dinheiro troque de mãos efetivamente. O limite do cartão é liberado.
- **Impacto no ERP:** O pedido deve ser simplesmente cancelado/anulado. **Não há lançamento financeiro** de entrada ou saída, pois não houve crédito. Não há taxas a estornar.
- **Gatilho:** API de Cancelamento.

4.1.2 Reembolso (Refund)

- **Definição:** Ocorre após a captura e liquidação (status: approved). O dinheiro já entrou (ou foi agendado para entrar) na conta do vendedor.
- **Mecanismo:** É uma nova transação financeira que retira o valor da conta do vendedor para devolver ao comprador.
- **Impacto no ERP:** Exige um lançamento contábil de **Devolução de Venda** (Débito na Receita).
- **Reembolso Total vs. Parcial:** O ERP deve suportar reembolsos parciais. Se o reembolso for parcial, o status do pedido deve refletir "Parcialmente Devolvido".
- **Rastreabilidade (SOURCE_ID):** Nos relatórios financeiros, o reembolso aparece como uma linha nova com descrição refund. Crucialmente, a coluna SOURCE_ID desta linha de reembolso aponta para o ID do Pagamento Original.⁸ O ERP deve usar este link para encontrar a venda original e atualizar seu saldo.
- **Estorno de Taxas:** Uma complexidade é se o Mercado Pago devolve a taxa (MP_FEE_AMOUNT) proporcionalmente. O ERP não deve "adivinhar" isso calculando percentuais; ele deve ler o valor exato debitado (NET_DEBIT_AMOUNT) na linha de reembolso do relatório para saber o custo real da devolução.

4.2 Protocolos de Gestão de Chargebacks (Disputas)

O *chargeback* é um processo assíncrono e longo que coloca o dinheiro em um estado de

"Bloqueio" (contingência). O ERP deve espelhar esses estados para evitar contar com dinheiro que está congelado. O ciclo de vida documentado ⁴ dita a seguinte lógica:

Fase 1: Notificação e Congelamento (Block)

Quando um cliente contesta a compra no banco:

- **API:** O status do pagamento muda para `charged_back` e o `status_detail` para `in_process`.¹⁸
- **Relatório:** É gerada uma linha de débito com `RECORD_TYPE = block` e descrição `chargeback` ou `dispute`.¹⁰ O valor é retirado do "Saldo Disponível" e movido para "Em Mediação".
- **Ação ERP:** O pedido deve ser bloqueado. O valor financeiro deve ser transferido contabilmente de "Caixa/Banco" para uma conta de "Ativo Bloqueado" ou "Depósito Judicial". Isso impede que a empresa gaste um dinheiro que não possui mais.

Fase 2: Defesa e Documentação

O vendedor envia as provas via API. O status permanece `in_process` ou muda para `review_pending`. O ERP deve monitorar se a documentação foi aceita (`documentation_status`).

Fase 3: Resolução (Resolution)

O veredito final gera dois caminhos possíveis, e o ERP deve estar preparado para ambos:

- **Cenário A: Vitória do Vendedor ou Cobertura (Coverage)**
 - Se o banco decide a favor do vendedor, ou se o Mercado Pago cobre o prejuízo pelo Programa de Proteção (`coverage_applied = true` na API ¹⁸).
 - **Relatório:** Uma linha de `unblock` (desbloqueio) devolve o valor ao saldo disponível. Pode haver também uma linha de `release` específica.
 - **Ação ERP:** Reverter o lançamento de bloqueio. O dinheiro volta a ser receita disponível.
- **Cenário B: Derrota do Vendedor (Débito Definitivo)**
 - Se a decisão for contra o vendedor (`coverage_applied = false`).
 - **API:** Status muda para `charged_back` e detalhe `settled` (liquidado contra o vendedor).¹⁸
 - **Relatório:** O valor bloqueado é consumido definitivamente. Pode aparecer uma linha final de `chargeback` consolidando a perda.
 - **Ação ERP:** Baixar o valor da conta de "Ativo Bloqueado" e lançá-lo como "**Perda Operacional/Chargeback**" (Despesa).

Insight Crítico de Validação: A flag `coverage_applied` na resposta da API de Chargebacks é o diferencial. Mesmo que o status seja `charged_back`, se `coverage_applied` for `true`, o ERP não deve registrar prejuízo financeiro, pois o Mercado Pago absorveu o custo. Ignorar essa flag leva a lançamentos de despesa indevidos.

5. Análise Detalhada dos Dados do Pagador e Prevenção de Fraudes

A conciliação não serve apenas para contabilidade; ela é uma ferramenta poderosa de validação de segurança pós-transacional (anti-fraude). A API do Mercado Pago fornece um objeto payer rico em detalhes que deve ser cruzado com os dados cadastrais do ERP.

5.1 O Objeto payer e a Validação Cruzada

Durante o checkout, o fraudador muitas vezes insere seus próprios dados no cadastro da loja (ERP) para receber o produto, mas utiliza um cartão roubado (cujos dados reais do titular são capturados pelo Mercado Pago e retornados na API).

- **Estrutura de Dados:** O objeto payer contém identification (tipo e número), email, first_name, last_name, e address.²²
- **Estratégia de Match:** O ERP deve comparar o payer.identification.number (CPF/CNPJ) retornado pela API com o CPF/CNPJ cadastrado no pedido do ERP.
 - **Regra de Ouro:** Se (CPF_ERP != CPF_MP_API), levantar um alerta de "Risco de Terceiros". Embora legítimo em casos de presentes, é um indicador de alta correlação com fraudes.

5.2 Normalização de Documentos

A API retorna tipos de documentos variados dependendo do país (CPF, CNPJ, DNI, RUT).

- **Desafio:** O campo pode vir nulo ou com formatação variável.⁵
- **Solução:** O ERP deve normalizar esses dados removendo pontuação (pontos e traços) antes da comparação. Além disso, deve estar preparado para lidar com nulos em casos de pagamentos como "Convidado" (Guest Checkout) ou terminais POS antigos, onde a identificação não é obrigatória.

5.3 Enriquecimento de Dados B2B

Para vendas B2B (Empresas), a conciliação bancária muitas vezes falha porque o pagamento vem de uma conta corporativa diferente da esperada. Utilizar o payer.identification.number (CNPJ) extraído do pagamento permite ao ERP alocar o crédito corretamente, mesmo que o cliente tenha usado a conta de uma filial ou matriz diferente daquela cadastrada no pedido.

6. Conciliação Bancária e Gestão de Saques (Payouts)

A última milha da conciliação é conectar o "mundo virtual" do Mercado Pago ao "mundo real"

do extrato bancário da empresa. O Mercado Pago não deposita venda por venda; ele agrupa valores e realiza transferências em lote ou sob demanda.

6.1 A Lógica de Agrupamento (Batching)

Tentar conciliar uma venda individual (R\$ 100,00) diretamente com o extrato bancário é impossível, pois ela estará somada a outras vendas em um depósito único de R\$ 5.432,10.

- **O Elo de Ligação (Payout ID):** Nos relatórios de liberação, quando o dinheiro sai da conta Mercado Pago para o banco, é gerada uma linha do tipo withdrawal ou payout.²
- **Colunas Chave:**
 - SOURCE_ID ou ID da linha de Payout: Este é o identificador do lote de transferência.
 - NET_DEBIT_AMOUNT: O valor total retirado da conta virtual.
- **Descrição no Extrato Bancário:** No extrato do banco receptor (Itaú, Bradesco, etc.), o depósito aparecerá com uma descrição genérica (ex: "MERCADOPAGO", "TED 12345"). Infelizmente, a descrição no banco nem sempre carrega o ID do Payout de forma clara.

6.2 Algoritmo de Reconciliação de Payouts

1. **Identificar o Depósito:** O ERP lê o extrato bancário e encontra um crédito de R\$ X na data D.
2. **Buscar no Relatório MP:** O ERP busca no Relatório de Liberações do Mercado Pago, nas datas D ou D-1 (considerando tempo de compensação bancária), uma linha de payout cujo valor absoluto (NET_DEBIT_AMOUNT) seja igual a R\$ X.
3. **Vínculo e Baixa:**
 - Se o valor bate, o ERP vincula o Depósito Bancário ao Payout ID.
 - Todas as vendas (settlements) e reembolsos que compuseram o saldo disponível antes desse payout são, por transitividade, consideradas "Liquidadas no Banco".
 - O ERP então baixa os títulos do "Contas a Receber (Mercado Pago)" e debita a conta "Banco (Itaú/Bradesco)".

6.3 Relatórios de Extrato Bancário

O Mercado Pago oferece uma funcionalidade específica para gerar relatórios a cada saque (Available Balance report gerado por saque).²⁴ Configurar o ERP para consumir este relatório específico (execute_after_withdrawal = true) simplifica drasticamente a conciliação, pois o arquivo conterá exatamente as transações que compuseram aquele saque específico, eliminando a necessidade de adivinhar quais vendas entraram no lote.

7. Cenários Avançados e Melhores Práticas de Implementação

7.1 Integração com Mercado Pago Point (POS)

Para transações físicas via maquininha (Point), a lógica é similar, mas com nuances.

- **store_id e pos_id:** O ERP deve capturar e armazenar qual terminal (poi_id) realizou a venda.¹¹ Isso é vital para redes de lojas físicas que precisam conciliar o caixa por filial.
- **Conciliação de Turno:** Diferente do online, o varejo físico fecha o caixa por turno. A conciliação deve agrupar as vendas pelo user_id (operador) ou pos_id para bater com a "Z" da maquininha.

7.2 Tratamento de Assinaturas (Subscriptions)

Em modelos de recorrência, o ID do "Plano" ou "Assinatura" (preapproval_id) é persistente, mas cada cobrança mensal gera um novo payment_id.

- **Risco:** O ERP não pode depender apenas do ID da assinatura. Ele deve tratar cada recorrência como uma nova transação financeira independente, vinculada ao cliente pai. Os relatórios de assinatura²⁵ detalham as datas de liberação de cada parcela (money_release_date), permitindo o fluxo de caixa projetado.

7.3 Tratamento de Erros e Códigos de Rejeição

O ERP não deve apenas registrar "Erro". Ele deve interpretar o status_detail para ações corretivas automatizadas¹:

- cc_rejected_call_for_authorize: Instruir o cliente a ligar no banco. Não cancelar o pedido imediatamente; manter em *hold*.
- cc_rejected_insufficient_amount: Sugerir outro meio de pagamento.
- cc_rejected_high_risk: Bloquear preventivamente o cadastro do usuário no ERP para revisão de segurança.
- pending_contingency: Não liberar mercadoria, mas não cancelar. Monitorar por até 2 dias úteis.

Conclusão

A conciliação entre o Mercado Pago e sistemas ERP não é uma tarefa estática de comparação de planilhas; é um processo dinâmico de engenharia de dados. A integridade financeira depende da capacidade do sistema de orquestrar identificadores únicos (external_reference), normalizar fusos horários, decompor taxas e impostos com precisão cirúrgica, e gerenciar o ciclo de vida não-linear de disputas e estornos.

Ao adotar o **Relatório de Liberações** como a única fonte da verdade contábil e implementar as lógicas de *match* determinístico e validação cruzada detalhadas neste manual, as organizações podem transformar a conciliação de um centro de custo operacional em um ativo estratégico de inteligência financeira e segurança. A arquitetura aqui proposta garante não apenas que "as contas fechem", mas que cada transação conte uma história financeira

completa, auditável e transparente.

Works cited

1. Collection creation results - Response handling - Mercado Pago ..., accessed January 9, 2026,
<https://www.mercadopago.com.br/developers/en/docs/checkout-api-payments/response-handling/collection-results>
2. Report fields - Account balance - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-api/additional-content/reports/account-money/report-fields>
3. Integration model - Checkout API (via Orders) - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-api-orders/integration-model>
4. Chargebacks - Resources - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.br/developers/en/docs/checkout-pro/chargebacks>
5. Search payments - Payments - Mercado Pago Developers, accessed January 9, 2026,
https://www.mercadopago.com.ar/developers/en/reference/payments/_payments_search/get
6. Mercado Pago's split payment solution sales report - Reports - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-pro/additional-content/reports/sales-report/introduction>
7. Create payment - Payments - Mercado Pago Developers, accessed January 9, 2026,
https://www.mercadopago.com.ar/developers/en/reference/payments/_payments/post
8. Refund order - Orders - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/reference/in-person-payments/point/orders/refund-order/post>
9. Create payment - Payments - Mercado Pago Developers, accessed January 9, 2026,
https://www.mercadopago.com.pe/developers/en/reference/payments/_payments/post
10. Report fields - Available money - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.br/developers/en/docs/woocommerce/additional-content/reports/available-money/report-fields>
11. Report fields - Released money - Mercado Pago Developers, accessed January 9, 2026,

- <https://www.mercadopago.com.mx/developers/en/docs/subscriptions/additional-content/reports/released-money/report-fields>
12. Report fields - Released money - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/subscriptions/additional-content/reports/released-money/report-fields>
 13. Generate report - Account balance - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-pro/additional-content/reports/account-money/generate>
 14. Generation through API - Released money - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.br/developers/en/docs/checkout-api-payments/additional-content/reports/released-money/api>
 15. Configure refunds and cancellations - Additional settings - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-pro/additional-settings/refunds-and-cancellations>
 16. Refunds and cancellations - Payment management - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.co/developers/en/docs/checkout-api-payments/payment-management/cancellations-and-refunds>
 17. Refunds and cancellations - Payment management - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.br/developers/en/docs/checkout-api-orders/refunds-cancellations>
 18. Manage chargebacks - Chargebacks - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-pro/chargebacks/manage>
 19. Chargebacks - Resources - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-pro/chargebacks>
 20. Report fields - Available money - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/woocommerce/additional-content/reports/available-money/report-fields>
 21. Get chargeback - Chargebacks - Mercado Pago Developers, accessed January 9, 2026,
https://www.mercadopago.com.co/developers/en/reference/chargebacks/_chargebacks_id/get
 22. Utilities - Industry data - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.co/developers/en/docs/checkout-api-payments/additional-content/industry-data/utilities>

23. Retail - Industry data - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-api-payments/additional-content/industry-data/retail>
24. Generating per withdrawal - Available money - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/woocommerce/additional-content/reports/available-money/withdrawal>
25. Report fields - Account balance - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.co/developers/en/docs/subscriptions/additional-content/reports/account-money/report-fields>
26. Transaction status - Main statuses - Mercado Pago Developers, accessed January 9, 2026,
<https://www.mercadopago.com.ar/developers/en/docs/checkout-api-orders/payment-management/status/transaction-status>