MCP SERVER | NO CODE | CLOUD HOSTED

Mercado Pago MCP

Charge cards, generate Pix codes, and manage refunds instantly.

Mercado Pago connects your AI agent directly to Brazil's top payment gateway. You can generate instant Pix QR codes, securely charge credit cards (Visa, Mastercard), handle customer records, and issue full or partial refunds—all from a natural conversation with zero manual dashboard clicks.

A+ Quality Score 100/100

pix-payments | qr-code-payments | credit-card-processing | refund-management | checkout-api

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/mercado-pago

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

Mercado Pago MCP

12 tools available
Cloud-hosted on Vinkius

Need to manage sales in Brazil? This MCP connects your AI agent to Mercado Pago, letting you run the entire financial workflow through simple chat commands. You can create instant Pix payments by generating scannable QR codes for the customer, or process traditional charges using credit cards like Visa and Mastercard.

It handles more than just charging: you can register new customers and save their card details for one-click reorders. Need to fix an order? Simply trigger a refund, partial or full. Furthermore, if a transaction fails or you need reporting, you don't have to guess; the MCP allows you to search for specific payments using filters like email address or status.

By connecting this capability through Vinkius, your AI client gets immediate access to Brazil's leading payment tools. This means e-commerce managers and small business owners can run their entire finance stack—from capturing customer data to issuing refunds—without ever touching a developer console.

Core Capabilities

01 — Generate Instant Pix Payments

Creates the necessary QR code data for customers to pay immediately using Brazil's popular Pix system.

02 — Process Credit Card Transactions

Charges various cards (Visa, Mastercard, Elo) securely, handling payment details and generating checkout links.

03 — Manage Customer Data

Registers new users and saves their card information to facilitate quick, repeat payments.

04 — Issue Refunds

Processes full or partial refunds on any completed payment using the transaction ID.

05 — Track Payments

Searches and retrieves detailed records of past payments using criteria like customer email or status.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/mercado-pago** — connect your AI agent in three steps.

01 First, subscribe to this MCP and provide your unique Mercado Pago Access Token.

02 Next, tell your AI client exactly what payment action you need—for example, 'Generate a Pix QR code for R$150.'

03 Finally, the agent executes the request using the integrated tools and returns confirmation details or data.

The bottom line is that your AI client acts as a direct financial worker, handling payment logic without you writing any boilerplate code.

Built For

This MCP is built for anyone who handles money flow in an e-commerce setting. If you're tired of manually logging into multiple dashboards to track failed payments or generate refund codes, this is for you.

E-commerce Operations Manager

Managing daily sales requires issuing refunds and checking payment status quickly so that customer service doesn't stall.

Fintech Developer

Needs to integrate local Brazilian payment methods like Pix into an AI-driven workflow without building the entire API layer from scratch.

Small Business Owner

Generates simple web checkout links and tracks all transactions directly through chat when managing sales remotely.

What Changes When You Connect

01 Start checking payment status right away. Instead of navigating multiple screens to reconcile sales, you can use the search_payments tool to find all transactions by email or ID in one prompt.

02 Handle cash-flow issues without delay. If a customer needs a partial credit, simply calling refund_payment lets your agent issue it instantly, logging the transaction details for accounting.

03 Speed up checkouts dramatically. You can use create_pix_payment to generate QR codes on demand, enabling instant payments that don't rely on card processing.

04 Improve customer data integrity. When a user signs up or makes an order, you can use create_customer and create_card to save their details securely for future one-click purchases.

05 Build checkout flows easily. Instead of hardcoding payment logic, you can use create_preference to generate dynamic links that redirect customers to the correct payment portal.

Real-World Applications

Handling a Failed Payment

A customer calls because their card was declined. Instead of making them manually repeat all details, you ask your agent to search_payments using their email address to find the last successful transaction and then use create_card to save their billing info for a quick retry.

Processing Bulk Returns

Your warehouse manager needs to issue refunds for 10 items sold yesterday. You tell your agent, which uses refund_payment, to process the full or partial returns across multiple payment IDs without needing to open ten different screens.

Setting up a New Sales Channel

A small business owner needs to accept payments from an online partner. They ask their agent to use create_preference, which generates the necessary secure checkout link for immediate deployment across the web.

Reconciling Daily Sales

It's end of day and you need a report on all successful transactions from the last 24 hours. You instruct your agent to search_payments, filtering by status 'approved', pulling together all necessary payment IDs.

Patterns to Avoid

Assuming Payment Success

✗ AVOID

A developer writes code that assumes a charge worked and moves on. They never check the status, leading to lost revenue because they didn't confirm payment.

✓ INSTEAD

Always use search_payments or get_payment after initiating a transaction. This confirms the final status and prevents failed payments from being treated as successful.

Forgetting Pix Payments

✗ AVOID

The system only handles card charges, ignoring local payment methods. The business loses access to customers who prefer instant QR code payments.

✓ INSTEAD

Use create_pix_payment whenever a customer opts for an immediate, non-card method of payment in Brazil.

Recreating Customer Data

✗ AVOID

Every time a user orders something, the system prompts them to re-enter their full name and card number. This is bad UX and adds friction.

✓ INSTEAD

Use create_customer first, then create_card. The MCP saves this data, allowing you to reference saved details for one-click payments.

The Right Fit

You should use this MCP if your core business function revolves around accepting money or managing the lifecycle of a payment in Brazil. Specifically, if you need to charge cards (create_payment), manage local Pix methods (create_pix_payment), or handle post-sale adjustments like refunds (refund_payment). Don't use this if your primary task is inventory management, user profile creation unrelated to billing, or complex reporting that doesn't involve transaction IDs. If you only need a simple contact list, look at general database connectors; if you need payment processing, stick with the tools here.

Tracking down lost money after a sale is always a headache.

Right now, when something goes wrong—a customer says their charge failed or they just can't find yesterday's payment receipt—you have to jump through hoops. You log into the portal, search by email in one tab, check status codes in another, and then copy-paste IDs between five different screens just to piece together what happened.

With this MCP, your agent does the legwork for you. Instead of clicking around, you simply ask: 'Show me all approved payments from user@email.com.' You get a clear, actionable list right back in chat, letting you confirm payment details or identify exactly which transactions need attention.

Creating secure checkout links with Mercado Pago MCP

Before this, setting up a web checkout usually meant hardcoding the payment gateway API call directly into your site's backend code. You had to manage tokens and redirection logic manually every time you needed a new link.

Now, you just ask for it. The agent uses create_preference to generate that secure, ready-to-use payment link instantly. It takes away the entire process of manual API configuration.

Mercado Pago: 12 Tools for Financial Ops

Use these tools to handle the full spectrum of financial tasks, from generating instant payment QR codes to managing customer records and issuing accurate refunds.

#	TOOL	DESCRIPTION
01	`create_card`	Saves a customer's payment details so the card can be used for future payments.
02	`create_customer`	Registers an entirely new user account within the Mercado Pago system.
03	`create_payment`	Processes a payment using card details or other methods for immediate charges.
04	`create_preference`	Generates a secure, temporary link that customers can use to complete an online purchase checkout.
05	`get_customer`	Retrieves specific details about an existing customer account.
06	`get_payment`	Fetches all detailed information for a single payment using its unique ID.
07	`get_preference`	Retrieves the current details of a generated checkout link or preference.
08	`get_identification_types`	Lists all valid identification formats used in Brazil, such as CPF and CNPJ.
09	`list_payment_methods`	Shows every payment method currently accepted by the gateway in Brazil.
10	`create_pix_payment`	Generates a unique QR code and associated data for an instant Pix payment request.
11	`refund_payment`	Initiates a refund, allowing you to partially or fully return funds on a completed payment.
12	`search_payments`	Searches across all transactions using filters like email address, status, or ID for reconciliation.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U Create a Pix payment for R$150 for a customer.

Pix payment created! Status: pending. QR Code generated for customer to scan and pay instantly.

U Refund payment 12345678 fully.

Full refund processed successfully! Status: refunded.

U Show me all approved payments for user@email.com.

Found 3 approved payments: 1. R$ 50.00 (Visa). 2. R$ 120.00 (Pix). 3. R$ 85.50 (Mastercard).

Frequently Asked Questions

01 **How do I use Mercado Pago with Pix payments?**

You initiate a Pix transaction using create_pix_payment. The MCP returns QR code data that you can give to the customer, enabling instant payment without relying on credit cards.

02 **Can I refund payments if they were made with Pix?**

Yes. You use the refund_payment tool and provide the original payment ID. The MCP handles the refund process regardless of whether the initial payment method was a card or Pix.

03 **Do I need to know the customer's email when searching payments?**

No, but it helps. You can use search_payments with various filters, including ID, status (like 'pending'), or even just a date range, so you don't have to guess what information you have.

04 **What is the difference between create_payment and create_preference?**

create_payment charges a card immediately. create_preference generates a secure link that directs the user to an external page where they can complete the payment themselves.

05 **How do I save customer cards using Mercado Pago MCP?**

You use the create_customer tool first, then use create_card. This saves their details securely within the system for future one-click transactions.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"mercado-pago": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

Mercado Pago is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Mercado Pago. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Mercado Pago MCP
Server ID	`019d75d2-9be1-727a-926b-b02ccda689c1`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/mercado-pago.