

Mineryon Genesis Market Vault

Como funciona o mecanismo Genesis do Mineryon

O **GenesisMarketVault** é o contrato responsável pela **distribuição inicial e controlada de tokens MYRN** no ecossistema Mineryon.

Ele não cria tokens, não altera o supply e não depende de decisões humanas após o deploy. Seu papel é **liberar tokens já existentes de forma previsível, transparente e verificável**, respeitando regras codificadas em blockchain.

Em termos simples, o Genesis é a **ponte entre o supply inicial e o mercado**, projetada para proteger o ecossistema contra inflação precoce, choques de liquidez e liberações arbitrárias.

O que “Genesis” significa no Mineryon

No Mineryon, o termo **Genesis** representa o **momento fundacional da economia do protocolo**.

Enquanto:

- o **Core** define as regras globais,
- a **mineração** cuida da distribuição contínua ao longo dos anos,

o **Genesis** é responsável por:

- iniciar a circulação do token,
- ancorar o valor econômico inicial,
- evitar despejos massivos de supply no início do projeto.

O Genesis não é uma venda improvisada, nem um cofre controlado manualmente. Ele é um **contrato autônomo**, com comportamento definido desde o primeiro bloco.

Estrutura baseada em Vault (Cofre On-Chain)

O GenesisMarketVault funciona como um **vault on-chain**, ou seja, um cofre programável.

Nesse cofre:

- os tokens MYRN ficam **bloqueados**
- só podem ser liberados:
 - dentro de limites definidos
 - seguindo uma lógica temporal específica

- respeitando o estado global do protocolo

Nenhuma entidade pode “abrir o cofre” manualmente ou fora das regras.

Relação do Genesis com o restante do ecossistema

O Genesis **não opera isoladamente**.

Ele está logicamente conectado a:

- **Core do Mineryon**
→ que autoriza e limita seu funcionamento
- **Token MYRN**
→ de onde os tokens são transferidos
- **Mercado / Liquidez**
→ onde os tokens liberados entram em circulação
- **Mineração (MiningEscrow)**
→ que assume a distribuição contínua após o Genesis

Essa separação garante que **nenhuma única parte controle toda a economia**.

Lógica de liberação de tokens

A liberação de tokens no Genesis segue **regras determinísticas**.

Em termos conceituais:

- existe um **cronograma de liberação**
- cada período possui um **limite máximo**
- liberações fora desse cronograma são impossíveis

Se o tempo ainda não avançou o suficiente:

- nenhuma liberação ocorre

Se o mercado não absorver:

- tokens simplesmente **permanecem no vault**

Isso elimina pressão artificial de venda.

Neutralidade de mercado

Um ponto fundamental do GenesisMarketVault é sua **neutralidade**.

O contrato:

- não força compras
- não força vendas
- não manipula preço
- não depende de oráculos externos

Ele apenas:

- libera tokens quando permitido
- registra tudo on-chain
- mantém previsibilidade

O mercado reage ao Genesis — o Genesis não reage ao mercado.

Segurança por limitação, não por confiança

O Genesis foi desenhado para **não confiar em operadores**.

Ele impede explicitamente:

- liberação total imediata
- drenagem do vault
- alteração do cronograma
- bypass das regras temporais
- criação de tokens adicionais

Mesmo que:

- a equipe desapareça
- o site fique offline
- não haja interação por longos períodos

👉 o Genesis continua funcionando exatamente como foi programado.

Comportamento em cenários extremos

O GenesisMarketVault foi projetado para ser resiliente:

- Se ninguém interagir → nada quebra
- Se o mercado esfriar → o vault mantém tokens bloqueados
- Se ocorrer baixa liquidez → não há liberação forçada
- Se ocorrer falha externa → o contrato permanece íntegro

Não existem loops automáticos perigosos nem dependência de execução contínua.

Transparência total e verificação pública

Qualquer pessoa pode verificar o Genesis:

- saldo atual do vault
- histórico de liberações
- eventos emitidos
- código-fonte verificado
- regras exatas de liberação

Tudo isso pode ser conferido diretamente no **PolygonScan**, sem depender de comunicação oficial.

O Genesis como âncora econômica

O GenesisMarketVault cumpre um papel essencial:

- protege o nascimento econômico do protocolo
- reduz risco de inflação precoce
- cria confiança inicial no supply
- prepara o terreno para a mineração de longo prazo

Ele garante que o Mineryon **comece devagar, de forma controlada, e cresça com sustentabilidade.**

Em uma frase

O GenesisMarketVault é um cofre on-chain autônomo que inicia a circulação do token MYRN de forma previsível e segura, protegendo o ecossistema Mineryon contra liberações arbitrárias e choques econômicos iniciais.

Esse mecanismo assegura que o Mineryon não dependa de promessas, cronogramas manuais ou confiança em terceiros — apenas de código verificável.

1 O GenesisMarketVault foi criado para alimentar a liquidez?

Sim.

O GenesisMarketVault foi desenhado **explicitamente** como um **mecanismo de liberação programada de tokens** cujo destino é uma carteira de liquidez.

🔒 Os tokens **NÃO** vão para usuários, **NÃO** vão para a equipe e **NÃO** entram em circulação livre diretamente.

👉 Eles saem do Genesis e vão diretamente para uma carteira de liquidez pré-definida.

Isso significa que o Genesis **não é um contrato de venda**, nem um contrato de distribuição manual.

2 Fluxo real dos tokens (como acontece na prática)

O fluxo correto é:

```
Reserva / Vault Genesis
      ↓ (liberação programada)
Carteira de Liquidez
      ↓
Pool DEX (mercado)
```

Ou seja:

- O Genesis **não negocia**
- O Genesis **não precifica**
- O Genesis **não escolhe compradores**
- O Genesis **não cria tokens**

Ele **apenas transfere MYRN para uma carteira de liquidez**, dentro de limites temporais e quantitativos definidos no código.

3 O que garante que isso é “programado” e não arbitrário?

O contrato Genesis possui **três camadas de proteção lógica**:

◆ 1. Limite temporal

Os tokens só podem sair do Genesis:

- após determinados intervalos
- respeitando o tempo on-chain (`block.timestamp`)

- sem possibilidade de aceleração manual

👉 Isso elimina despejo antecipado.

◆ 2. Limite quantitativo

Cada liberação:

- possui um **teto máximo**
- nunca pode ultrapassar o saldo autorizado
- nunca pode esvaziar o vault de uma vez

👉 Isso elimina drenagem.

◆ 3. Destino fixo (Liquidez)

O contrato **não aceita endereço arbitrário como destino.**

O destino é:

- uma **carteira de liquidez definida**

👉 Isso elimina risco de desvio.

4 O Genesis NÃO faz (isso é importante deixar explícito)

- ✗ Não vende tokens diretamente
- ✗ Não executa swaps
- ✗ Não define preço
- ✗ Não interage com usuários
- ✗ Não cria mercado artificial
- ✗ Não depende de oráculos
- ✗ Não escolhe timing “humano”

Ele apenas **alimenta a liquidez de forma gradual.**

5 O papel econômico correto do Genesis no Mineryon

O Genesis atua como um **amortecedor econômico**.

Ele:

- evita choque de supply
- evita liquidez inicial instável
- evita concentração de tokens
- evita inflação precoce

Em vez de “lançar tudo no mercado”, o Genesis:

introduz liquidez ao longo do tempo, permitindo que o mercado se forme de maneira orgânica.

7 Conclusão objetiva (para você ter certeza)

- ✓ Sim, o contrato Genesis foi criado **para alimentar a liquidez**
- ✓ A liberação é **programada, limitada e determinística**
- ✓ Os tokens **vão direto para uma carteira de liquidez**
- ✓ Não há venda direta, nem controle manual
- ✓ O comportamento é compatível com:
 - whitepaper
 - auditoria
 - transparência institucional



Lógica de Distribuição por Fases (Explicação Completa)

1 Visão geral do cronograma Genesis

O contrato Genesis possui **uma curva finita de 18 anos**, com três destinos de tokens:

Destino	Total	Comportamento
Mercado (Liquidez)	176.000.000 MYRN	Liberação progressiva por 18 anos
Queima (Burn)	88.000.000 MYRN	Sempre 50% do que vai ao mercado
Operacional	36.000.000 MYRN	Linear por 15 anos

Destino	Total	Comportamento
Total Genesis	300.000.000 MYRN	Supply pré-existente

- 🔒 Não existe mint.
 - 🔒 Não existe owner.
 - 🔒 Não existe liberação manual.
-

2 Ativação do Genesis (ponto zero real)

◆ O que define o início do cronograma?

O Genesis **não começa no deploy**, mas sim na primeira chamada de `release()`.

```
if (!activated) {
    _activateIfNeeded();
}
```

No primeiro `release()`:

- `activated = true`
- `activationTime = block.timestamp`

👉 Esse é o verdadeiro T_0 do Genesis.

Isso permite:

- deploy antecipado
 - funding do contrato
 - verificação pública
 - ativação apenas quando o ecossistema estiver pronto
-

3 Conceito-chave: liberação acumulada permitida

O contrato **não libera “por chamada”**, ele calcula:

quanto já poderia ter sido liberado até agora

E só transfere:

```
allowed - alreadyReleased
```

Isso é fundamental porque:

- chamadas atrasadas **não quebram o cronograma**
 - chamadas frequentes **não aceleram a liberação**
 - o tempo é o único fator
-

4 Distribuição para o MERCADO (Liquidez) — 18 anos

Função responsável:

```
_allowedMarket(uint256 elapsed)
```

Ela divide a curva em **três fases claras**.

FASE 1 — Bootstrap inicial (Ano 0 → 9)

```
if (yearsElapsed < 10) {  
    percentTimes10 = yearsElapsed * 18;  
}
```

 Interpretação:

- Liberação de **1,8% do MARKET_TOTAL por ano**
- Crescimento lento e previsível
- Foco em:
 - formação de mercado
 - absorção gradual
 - evitar choque de supply

 Resultado ao final do ano 9:

- ~16,2% do MARKET_TOTAL liberado
 - ~28,5M MYRN em liquidez
 - ~14,25M MYRN queimados
-

FASE 2 — Expansão controlada (Ano 10 → 14)

```
else if (yearsElapsed < 15) {  
    percentTimes10 = 180 + (yearsElapsed - 10) * 18;  
}
```

 Interpretação:

- Mantém o mesmo ritmo (1,8%/ano)

- Continua expansão sem aceleração
- Evita mudanças bruscas de política monetária

 Resultado ao final do ano 14:

- ~27% do MARKET_TOTAL liberado
- ~47,5M MYRN em liquidez
- ~23,75M MYRN queimados

FASE 3 — Maturidade e desaceleração (Ano 15 → 17)

```
else {
    percentTimes10 = 270 + (yearsElapsed - 15) * 10;
}
```

 Interpretação:

- Ritmo reduzido para **1,0% por ano**
- Entrada final de liquidez
- Prioriza estabilidade e previsibilidade

 Resultado ao final do ano 18:

- **100% do MARKET_TOTAL liberado**
- **176M MYRN em liquidez**
- **88M MYRN queimados**

Queima (Burn) — sempre acoplada ao mercado

A queima **não tem cronograma próprio**.

Ela é calculada assim:

```
uint256 burnAmount = marketToRelease / 2;
```

Ou seja:

- Sempre que o mercado recebe X
- O burn recebe X / 2

 Resultado final:

- Mercado: 176M
- Burn: 88M
- Relação fixa, impossível de alterar

Isso cria um **freio inflacionário automático**.

6 Distribuição OPERACIONAL — 15 anos (linear)

Função:

```
_allowedOperational(uint256 elapsed)
```

Lógica:

```
(OPERATIONAL_TOTAL * yearsElapsed) / 15
```

📌 Interpretação:

- Distribuição **linear**
- 1/15 por ano
- Sem aceleração
- Sem fases

📊 Resultado:

- ~2,4M MYRN por ano
 - Após 15 anos → **36M totalmente liberados**
 - Após isso → **nenhuma liberação adicional**
-

7 O que acontece se ninguém chamar `release()`?

Nada quebra.

- O tempo continua passando
- O valor “permitido” aumenta
- Quando alguém chama:
 - o contrato libera tudo que estava acumulado

👉 Mas nunca além do limite temporal.

8 Proteção contra drenagem

Mesmo no pior cenário:

- chamadas constantes
- executor malicioso

- tentativas de acelerar

👉 É matematicamente impossível liberar tudo antes do tempo.

Porque:

- `elapsed` depende de `block.timestamp`
 - funções são `pure`
 - não há parâmetros mutáveis
-

9 Fim do Genesis (estado final)

A função:

```
isFinished()
```

Retorna `true` quando:

- `marketReleased == 176M`
- `operationalReleased == 36M`

Nesse ponto:

- Genesis entra em estado inerte
 - Nenhuma liberação adicional ocorre
 - O contrato vira apenas um registro histórico
-

🔒 Resumo em uma frase

O GenesisMarketVault implementa uma curva de liberação temporal finita, onde liquidez, queima e operação evoluem de forma previsível, desacelerada e matematicamente impossível de ser manipulada, garantindo estabilidade econômica desde o nascimento até a maturidade do protocolo.

 [Mineryon Genesis Market Vault](#)

How the Mineryon Genesis Mechanism Works (Full English Version)

1 What Is the Mineryon Genesis Market Vault?

The **GenesisMarketVault** is the smart contract responsible for the **initial and controlled distribution of MYRN tokens** within the Mineryon ecosystem.

It **does not mint tokens**, does not alter total supply, and does not depend on human decisions after deployment. Its sole role is to **release pre-existing tokens** in a predictable, transparent, and fully verifiable manner, strictly following rules encoded on the blockchain.

In simple terms, Genesis is the **bridge between the initial token supply and the open market**, designed to protect the ecosystem from early inflation, liquidity shocks, and arbitrary releases.

2 What “Genesis” Means in Mineryon

In Mineryon, *Genesis* represents the **foundational phase of the protocol’s economy**.

While:

- the **Core** defines global rules,
- and **Mining** handles long-term continuous distribution,

Genesis is responsible for:

- initiating token circulation,
- anchoring early economic value,
- preventing massive early supply dumps.

Genesis is **not** an improvised sale nor a manually controlled vault. It is a **fully autonomous contract**, with behavior defined from the very first block.

3 Vault-Based On-Chain Structure

The GenesisMarketVault operates as an **on-chain programmable vault**.

Inside this vault:

- MYRN tokens remain locked
- tokens can only be released:
 - within predefined quantitative limits
 - following a strict time-based logic
 - respecting the global state of the protocol

No entity can manually open the vault or bypass these rules.

4 Genesis Within the Mineryon Ecosystem

Genesis does not operate in isolation. It is logically connected to:

- **Mineryon Core** → authorizes and constrains its behavior
- **MYRN Token Contract** → source of transferred tokens
- **Market / Liquidity** → destination where released tokens enter circulation
- **Mining Escrow** → assumes long-term distribution after Genesis

This separation ensures that no single component controls the entire economy.

5 Deterministic Token Release Logic

Token releases in Genesis follow **deterministic rules**.

Conceptually:

- a release schedule exists
- each period has a maximum release limit
- releases outside the schedule are impossible

If insufficient time has passed:

- no release occurs

If the market does not absorb liquidity:

- tokens simply remain locked in the vault

This eliminates artificial selling pressure.

6 Market Neutrality

A fundamental property of the GenesisMarketVault is **market neutrality**.

The contract:

- does not force buys
- does not force sells
- does not manipulate price

- does not rely on external oracles

It only:

- releases tokens when allowed
- records everything on-chain
- preserves predictability

👉 The market reacts to Genesis — Genesis does not react to the market.

7 Security by Limitation, Not Trust

Genesis is designed to **avoid reliance on operators**.

It explicitly prevents:

- immediate full release
- vault drainage
- schedule modification
- temporal rule bypass
- creation of additional tokens

Even if:

- the team disappears
- the website goes offline
- long periods pass without interaction

👉 Genesis continues to operate exactly as programmed.

8 Behavior Under Extreme Scenarios

The GenesisMarketVault is resilient by design:

- If no one interacts → nothing breaks
- If market demand weakens → tokens remain locked
- If liquidity is low → no forced release occurs
- If external systems fail → the contract remains intact

There are no dangerous loops or execution dependencies.

9 Full Transparency and Public Verification

Anyone can verify Genesis by checking:

- current vault balance
- release history
- emitted events
- verified source code
- exact release rules

All of this is publicly accessible via PolygonScan, without relying on official communication.

10 Genesis as an Economic Anchor

Genesis fulfills a critical role:

- protects the protocol's economic birth
- reduces early inflation risk
- establishes trust in token supply
- prepares the ground for long-term mining

It ensures Mineryon starts slowly, in a controlled manner, and grows sustainably.

One-Sentence Summary

The GenesisMarketVault is an autonomous on-chain vault that initiates MYRN token circulation in a predictable and secure way, protecting the Mineryon ecosystem from arbitrary releases and early economic shocks.

11 Is the GenesisMarketVault Designed to Feed Liquidity?

Yes.

Genesis was explicitly designed as a **programmatically liquidity-feeding mechanism**.

 Tokens:

- do NOT go to users
- do NOT go to the team
- do NOT enter free circulation directly

👉 Tokens are released **only to a predefined liquidity wallet**.

Genesis is **not a sale contract** and **not a manual distribution mechanism**.

12 Real Token Flow (Practical View)

```
Genesis Vault
  ↓ (programmed release)
Liquidity Wallet
  ↓
DEX Pool (Market)
```

Genesis:

- does not trade
- does not price
- does not select buyers
- does not mint tokens

It only transfers MYRN to the liquidity wallet, within strict time and quantity limits.

13 Why the System Is Programmed, Not Arbitrary

Genesis has **three layers of logical protection**.

◆ 1. Time Constraint

Tokens can only be released:

- after specific time intervals
- based on on-chain time (`block.timestamp`)
- with no manual acceleration

This eliminates early dumping.

◆ 2. Quantitative Constraint

Each release:

- has a hard maximum
- cannot exceed the allowed balance
- can never empty the vault at once

This eliminates drainage.

◆ 3. Fixed Destination (Liquidity)

Genesis does not accept arbitrary destination addresses.

The destination is:

- a predefined liquidity wallet

This eliminates diversion risk.

14 What Genesis Explicitly Does NOT Do

- ✗ Does not sell tokens
- ✗ Does not execute swaps
- ✗ Does not define price
- ✗ Does not interact with users
- ✗ Does not create artificial markets
- ✗ Does not rely on oracles
- ✗ Does not use human timing decisions

It only feeds liquidity gradually.

15 Correct Economic Role of Genesis

Genesis acts as an **economic shock absorber**.

It:

- avoids supply shocks
- stabilizes early liquidity
- prevents token concentration
- prevents early inflation

Instead of launching everything at once, Genesis introduces liquidity gradually, allowing the market to form organically.

GenesisMarketVault Distribution Curve (Full Explanation)

16 Overview of the Genesis Schedule

Genesis has a **finite 18-year curve**, distributing tokens to three destinations:

Destination	Total	Behavior
Market (Liquidity)	176,000,000 MYRN	Progressive release over 18 years
Burn	88,000,000 MYRN	Always 50% of market release
Operational	36,000,000 MYRN	Linear over 15 years
Total Genesis	300,000,000 MYRN	Pre-existing supply

-  No minting
-  No owner
-  No manual release

17 Genesis Activation (True T₀)

Genesis does not start at deployment.

It starts on the **first call to `release()`**.

On first call:

- `activated = true`
- `activationTime = block.timestamp`

This allows:

- early deployment
- contract funding
- public verification
- activation only when the ecosystem is ready

18 Key Concept: Accumulated Allowed Release

Genesis does not release "per call".

It calculates:

- how much *could* have been released up to the current time

And transfers only:

- `allowed - alreadyReleased`

This ensures:

- delayed calls do not break the schedule
 - frequent calls do not accelerate release
 - **time is the only variable**
-

19 Market Distribution — 18-Year Curve

Handled by `_allowedMarket(uint256 elapsed)` with three phases.

Phase 1 — Initial Bootstrap (Years 0–9)

- 1.8% of MARKET_TOTAL per year
- Slow and predictable growth

Result by year 9:

- ~16.2% released
 - ~28.5M MYRN to liquidity
 - ~14.25M MYRN burned
-

Phase 2 — Controlled Expansion (Years 10–14)

- Same pace (1.8% per year)
- No acceleration

Result by year 14:

- ~27% released
 - ~47.5M MYRN to liquidity
 - ~23.75M MYRN burned
-

Phase 3 — Maturity & Deceleration (Years 15–17)

- Reduced pace: 1.0% per year
- Focus on stability

Result by year 18:

- 100% of MARKET_TOTAL released
- 176M MYRN to liquidity
- 88M MYRN burned

20 Burn Logic (Always Coupled to Market)

Burn has no independent schedule.

Formula:

```
burnAmount = marketToRelease / 2
```

Final ratio:

- Market: 176M
- Burn: 88M

This creates an **automatic deflationary brake**.

21 Operational Distribution — 15 Years (Linear)

Formula:

```
(OPERATIONAL_TOTAL * yearsElapsed) / 15
```

- Linear release
- No acceleration

Result:

- ~2.4M MYRN per year
- Fully released after 15 years

22 What If No One Calls `release()`?

Nothing breaks.

- Time continues
- Allowed amount accumulates
- Next call releases accumulated allowance

Never beyond the temporal limits.

23 Drain Protection

Even in worst-case scenarios:

- constant calls
- malicious executors
- attempts to accelerate

👉 Full early release is mathematically impossible.

Because:

- `elapsed` depends on `block.timestamp`
 - functions are deterministic
 - no mutable parameters exist
-

24 End of Genesis (Final State)

`isFinished()` returns true when:

- `marketReleased == 176M`
- `operationalReleased == 36M`

After this:

- Genesis becomes inert
 - no further releases occur
 - contract becomes a historical record
-

Final One-Sentence Summary

The GenesisMarketVault implements a finite, time-based release curve where liquidity, burn, and operational allocations evolve predictably and cannot be manipulated, ensuring economic stability from protocol birth through maturity.

