

# **SnowRail: tesorería ejecutable para pagos condicionales on-chain**

## **Resumen ejecutivo**

SnowRail es infraestructura de tesorería ejecutable: permite definir intents financieros (qué pagar, cuánto y bajo qué condiciones) y ejecutarlos automáticamente en blockchain cuando las condiciones se cumplen. Su núcleo combina custodia programable, separación entre decisión y ejecución, señales/oráculos en tiempo real y privacidad opcional mediante ZK. SnowRail no es un bot, ni un mixer, ni un dashboard DeFi: es un motor de ejecución de pagos condicionales orientado a operaciones B2B y tesorería. Integra una interfaz universal para coordinación (MCP) y pagos condicionales para servicios/APIs (x402), con liquidación verificable on-chain. Como extensión natural, habilita acceso a economía tokenizada y RWA mediante conectores abstractos que modelan rendimiento, liquidez y riesgo sin depender de una sola cadena. El resultado es un sistema que reduce fricción operativa, mejora control de riesgo y habilita nuevas formas de automatización financiera en producción.

## **1) Problema**

Los pagos y operaciones de tesorería en entornos B2B suelen quedar atrapados entre dos extremos:

- Manualidad costosa: aprobaciones humanas, procesos por tickets, conciliación tardía, errores operativos y demoras.
- Automatización riesgosa: scripts o bots que ejecutan sin una capa robusta de control, límites y verificaciones.

Con la llegada de agentes de IA y flujos machine-to-machine, el problema se agrava:

- Los agentes requieren pagar por datos, herramientas y APIs de forma continua.
- Los equipos B2B necesitan control estricto sobre qué se paga, cuándo y bajo qué condiciones.
- La ejecución debe ser verifiable y audit-able, pero sin exponer información sensible en todos los casos.

La brecha central: hace falta una capa de tesorería que sea programable, verifiable y segura para operar sin intervención humana constante, manteniendo controles de riesgo y, cuando se requiere, privacidad.

## 2) Qué es SnowRail

SnowRail es infraestructura de tesorería ejecutable para definir y ejecutar pagos condicionales on-chain bajo condiciones verificables.

Qué es:

- Un motor que convierte políticas financieras en intents ejecutables.
- Un sistema que evalúa condiciones externas (señales/oráculos) y ejecuta pagos solo cuando corresponde.
- Una capa de control para operaciones repetitivas y críticas (pagos, desembolsos, settlement) con trazabilidad on-chain.

Qué no es:

- No es un bot de trading.
- No es un mixer como producto principal.
- No es un dashboard DeFi.
- No es una simple billetera multisig con automatizaciones.

En una frase: SnowRail convierte tesorerías en sistemas que pueden ejecutar pagos on-chain de forma automática, bajo reglas verificables y con controles de riesgo.

## 3) Principios de diseño

- Separación de funciones (decidir vs ejecutar): diferencia entre evaluar si un pago debe ocurrir y la acción de ejecutarlo, reduciendo el riesgo de automatizaciones “sin frenos”.
- Custodia programable: fondos administrados mediante reglas y permisos que limitan qué movimientos son válidos.
- Condiciones verificables: reglas basadas en señales que pueden validarse antes de ejecutar (estados, rangos, ventanas de tiempo, confirmaciones).
- Automatización segura por defecto: la prioridad son controles, límites y trazabilidad, no “velocidad” sin gobernanza.
- Privacidad opcional: módulo disponible cuando el caso lo exige (salarios, grants, pagos sensibles).
- Filosofía chain-agnostic: interoperabilidad como requisito, no como extra.

## 4) Arquitectura conceptual (alto nivel)

A nivel conceptual, SnowRail puede entenderse como cuatro capas:

- Capa de intención: donde se define el intent financiero (destino, monto, condiciones, límites).
- Capa de control: donde se valida el intent contra políticas, permisos y señales externas.
- Capa de ejecución: donde un agente ejecutor dispara la transacción cuando el intent es válido.
- Capa de verificación: donde la red confirma, registra y permite auditar el resultado.

Actores típicos (conceptuales):

- Operador B2B/DAO: define políticas e intents.
- Agente ejecutor: observa condiciones, decide en base a reglas y ejecuta cuando corresponde.
- Señales/oráculos: aportan datos externos necesarios para validar condiciones.
- Capa on-chain: liquida, registra eventos y habilita verificación.

## 5) Flujo operativo (intent → validación → autorización → ejecución → verificación)

Diagrama conceptual descrito en texto:

Intent: Entrada: "Pagar X a Y si se cumple condición Z, dentro de límites L". Salida: intent registrado como objetivo financiero con fondos y reglas asociadas.

Validación: Entrada: intent + políticas + señales/oráculos. Proceso: comprobar restricciones (límites, tiempo, estados, permisos, condiciones). Salida: intent marcado como elegible o no elegible.

Autorización: Entrada: intent elegible. Proceso: aplicar control de riesgo y separación de funciones (por ejemplo, requerir confirmación de política, permitir solo rutas aprobadas). Salida: autorización para ejecutar (o bloqueo/espera).

Ejecución: Entrada: autorización. Proceso: agente ejecutor envía transacción de liquidación. Salida: transacción enviada con parámetros definidos por el intent.

Verificación: Entrada: resultado on-chain. Proceso: confirmar estado final, registrar trazabilidad y emitir evidencia para conciliación. Salida: intent completado (o fallido) con prueba verificable.

## 6) Módulos del sistema

### 6.1 Payment intents

Los intents son la unidad principal de trabajo. Representan una instrucción financiera expresada como:

- Qué: acción (pagar, desembolsar, distribuir, liquidar).
- Cuánto: monto y activo.
- A quién: destino o beneficiario.
- Cuándo: ventana temporal, disparador o condición.
- Bajo qué reglas: límites, permisos y validaciones.

Los intents estandarizan operaciones repetitivas y las transforman en ejecuciones automáticas verificables.

## **6.2 Custodia programable y control de riesgo**

Este módulo convierte la tesorería en un sistema gobernado por reglas:

- Límites por operación y por periodo.
- Destinos permitidos y acciones válidas.
- Controles de exposición (por ejemplo, límites de riesgo o de lockup).
- Separación de funciones: la decisión se apoya en políticas y la ejecución obedece únicamente a intents válidos.

La meta es reducir el riesgo de ejecución autónoma sin control, manteniendo operación automatizada.

## **6.3 Oráculos y señales en tiempo real**

Muchas condiciones requieren datos externos o señales:

- Precios y rangos para pagos condicionados.
- Estados de cumplimiento (por ejemplo, hitos de un grant o delivery de un servicio).
- Disponibilidad o validación de una contraparte.

Las señales/oráculos se usan como entradas de validación, no como “decisiones”: habilitan o bloquean intents según reglas predefinidas.

## **6.4 Privacidad ZK opcional**

La privacidad se incorpora como módulo opcional para casos que requieren confidencialidad:

- Payroll/contractors donde la trazabilidad pública puede ser indeseable.
- Pagos sensibles (negociaciones, proveedores, estrategias).
- Grants o desembolsos con requisitos de privacidad.

Conceptualmente, se busca ejecutar y verificar sin exponer vínculos innecesarios entre origen y destino, manteniendo auditabilidad donde sea requerido por compliance.

## **6.5 RWA y economía tokenizada (extensión)**

RWA es una extensión natural del core: una tesorería ejecutable no solo paga, también asigna capital.

Implementación conceptual:

- Conectores abstractos: conexión a activos tokenizados mediante integraciones encapsuladas, evitando dependencia de un proveedor único.
- Modelo de activo: cada RWA se trata como un módulo con señales de rendimiento, liquidez y riesgo.
- Intents de asignación: intents para asignar, reinvertir o rebalancear capital hacia RWA bajo reglas verificables.

- Colateral productivo: uso de RWA como colateral cuando el caso lo admite, buscando eficiencia de capital sin romper controles.
- Trazabilidad: decisiones y outcomes verificables on-chain a nivel de intent y resultado.

En síntesis: RWA no es un producto aparte; es un destino y herramienta financiera que el motor puede operar bajo reglas, igual que un pago.

## 6.6 MCP y x402 (interfaz y pagos condicionales)

MCP (entrada universal): actúa como interfaz estandarizada para que sistemas, agentes y aplicaciones describan tareas financieras y consulten estados. A nivel conceptual, es capa de coordinación: una forma uniforme de expresar intents y consultar resultados.

x402 (pagos condicionales / paywalls para APIs): habilita el patrón pagar para acceder a un recurso (por ejemplo, un endpoint o una API). SnowRail lo incorpora para que agentes o sistemas puedan:

- Pagar por consumo de datos/herramientas con control.
- Encadenar pagos con condiciones (pagar solo si se valida entrega o cumplimiento).
- Mantener trazabilidad y conciliación sobre gastos machine-to-machine.

## 7) Casos de uso

- Pagos por APIs/datos (agentic workflows): un agente necesita datos o cómputo. SnowRail define un intent condicionado: pagar solo si la respuesta cumple condiciones. Se reduce fraude, reintentos y gasto no controlado.
- Payroll y contractors: pagos recurrentes con ventanas y límites, con privacidad opcional. Se define una vez y el motor ejecuta con evidencia verificable para conciliación.
- Grants por hitos: desembolsos condicionados a hitos verificables. Reduce fricción, acelera ciclos y mantiene control de riesgo sobre desembolsos.
- Operaciones cross-border de tesorería: movimientos recurrentes entre entidades, proveedores o filiales bajo reglas, con ejecución automática y conciliación más rápida.
- Colateral productivo con RWA: asignación a activos tokenizados bajo reglas de liquidez/riesgo; rebalanceo/reinversión por señales. El colateral puede mantenerse productivo cuando aplica, sin perder trazabilidad.

## 8) Métricas/KPIs de adopción

KPIs sugeridos:

- Número de intents creados y ejecutados por periodo.
- Tasa de éxito de ejecución (intents completados vs fallidos) y causas principales.
- Tiempo medio desde intent hasta verificación final (latencia operativa).
- Volumen liquidado bajo intents (por categoría: pagos, payroll, grants, asignación).
- Número de integradores/partners activos y casos en producción.
- Retención de organizaciones (intents recurrentes por mes).
- Reducción de intervención manual (conteo de aprobaciones, tickets, conciliaciones).

- Cobertura de controles: porcentaje de intents operando con políticas y límites definidos.

## **9) Consideraciones de seguridad (conceptual)**

- Privilegio mínimo: cada componente opera con permisos restringidos a su función.
- Separación de funciones: la ejecución obedece a intents válidos; la evaluación se sostiene en políticas.
- Límites y restricciones: topes por operación, por periodo y por tipo de acción.
- Observabilidad: evidencia verificable del outcome de cada intent para conciliación.
- Dependencias externas: señales/oráculos como entradas sujetas a validación, no autoridad absoluta.
- Privacidad con límites: ZK se usa cuando corresponde, sin sacrificar control y evidencia sobre ejecución.

## **10) Próximos pasos (pilotos y partnership)**

Ruta típica de evaluación sin entrar en implementación:

- Selección de caso de uso: elegir un flujo con alto costo operativo o alto riesgo (payroll, grants, pagos por APIs, treasury ops, asignación a RWA).
- Definición de políticas e intents: documentar reglas mínimas (límites, condiciones, ventanas, destinos permitidos y señales necesarias).
- Piloto controlado: alcance acotado, criterios claros de éxito, revisión periódica con KPIs.

Criterios de readiness:

- Señales disponibles para condiciones verificables.
- Políticas de riesgo definidas (límites, permisos).
- Requisitos de conciliación (qué evidencia requiere finanzas/operaciones).
- Necesidades de privacidad (si aplica).

Escalamiento: expandir a más flujos cuando el piloto muestre estabilidad, control y trazabilidad consistente.

## **Equipo**

CTO: Juan Carlos Quiceno Vasquez — jquiceva@gmail.com — X @Cyber\_paisa

Arquitecto de software: Luis Fernando — guffenix@gmail.com — X @AKAwolfcito

Product manager: Daniel Luque — daniluque@colombiablockchain.co — X @DaniLuqueTV

Dev: Josue Marin — josuemarin2009@hotmail.com — X @josuearayamarin

Dev: Kevin Brenes — kevin.brenes0220@gmail.com — X @KevinMembr95392

## Glosario

- Intent: instrucción financiera declarativa con condiciones y límites.
- Tesorería ejecutable: tesorería capaz de ejecutar operaciones automáticamente bajo reglas verificables.
- Custodia programable: control de fondos mediante reglas y permisos definidos, con ejecución restringida.
- Separación de funciones: diseño que distingue evaluar/decidir de ejecutar/liquidar.
- Oráculo / señal: fuente de datos externos usada para validar condiciones.
- Liquidación on-chain: ejecución final registrada y verificable en blockchain.
- ZK: técnica criptográfica para probar propiedades sin revelar datos subyacentes.
- RWA: activos del mundo real representados como tokens para rendimiento o colateral.
- Chain-agnostic: filosofía de diseño que evita dependencia de una sola cadena para operar.
- MCP: interfaz estandarizada para coordinar tareas/consultas entre sistemas y agentes.
- x402: patrón de pagos condicionales para acceso a recursos (por ejemplo, APIs) con verificación de pago.
- Control de riesgo: límites, validaciones y permisos que restringen ejecuciones no deseadas.