

Convención de grafo y semántica de pesos (MVP Fase 1)

Objetivo

El objetivo de este documento es fijar de manera operativa la **semántica de los pesos** en el grafo que será utilizado por el agente y, en función de ello, definir la regla de actualización y de poda basada en la curvatura de Forman-Ricci. La elección correcta de la semántica es crítica porque determina cómo se interpretan las operaciones de *estirar* o *contraer* aristas durante el flujo de Ricci y cuándo debe suprimirse una arista como ruido.

A continuación se exponen las decisiones adoptadas, la justificación teórica (con citas a fuentes primarias) y las implicaciones prácticas para el `RicciCleaner` y el `RAS` descritos en el plan inicial.

1. Tipo de grafo y clases de aristas

Nodos

Para el MVP se utilizará un grafo ponderado $G = (V, E)$ cuyos nodos representan distintos tipos de objetos dentro del agente:

- **MEMORY_CHUNK**: fragmentos o *chunks* de memoria semántica (e.g., notas, citas, resúmenes).
- **TOOL**: herramientas concretas accesibles por el agente (buscadores, RAG local, calculadora, etc.).
- **EXPERT**: sub-agentes con funciones especializadas (p. ej. verificador, planificador, redactor).
- **CONCEPT** (opcional): conceptos o entidades recurrentes si se decide emplear un grafo semántico más fino.

Aristas

Las aristas $e = (u, v)$ del grafo poseen diferentes roles. Para cada arista se almacenan:

- `type`: categoría de la conexión (`SIMILARITY`, `CO_OCCURRENCE`, `SUCCESS_LINK`, `CONSTRAINT`).
- `d` (**distancia/peso**): valor positivo que representará la *distancia/costo de conexión* entre los nodos (ver §2).
- `age`: antigüedad o tiempo desde su creación/última actualización.
- `locked` (**booleano**): indica si la arista es crítica y no debe ser podada (`CONSTRAINT`).

Las aristas `CONSTRAINT` codifican reglas simbólicas o rutas obligatorias (Simbólico en RSI) y no deben eliminarse; el flujo de Ricci sólo ajustará aristas de tipo `SIMILARITY`, `CO_OCCURRENCE` y `SUCCESS_LINK`.

2. Semántica de los pesos

2.1 Alternativas y decisión

Existen dos formas naturales de interpretar el peso w de una arista:

1. **Peso como fuerza de confianza (más alto es más fuerte):** se utiliza en grafos semánticos donde valores altos indican que la conexión es fiable o relevante.
2. **Peso como distancia/costo (más alto es más lejano):** valores altos indican mayor costo para atravesar la arista, por lo que caminos con pesos altos son menos preferidos.

La regla de actualización propuesta por el documento ($\kappa < 0$ “estira” y $\kappa > 0$ “contrae” la arista) implica multiplicar el peso por un factor mayor a 1 para curvaturas negativas y por un factor menor a 1 para curvaturas positivas. Bajo la interpretación de “fuerza”, esto sería contradictorio: una arista con curvatura negativa aumentaría su fuerza, cuando normalmente deseamos debilitarla.

Por tanto, **adoptamos la convención de que el peso d es un valor de distancia o costo**. Así:

- Un peso bajo (p. ej. $d = 0,1$) indica que la arista conecta fuertemente a los nodos (preferible para el encaminamiento).
- Un peso alto (p. ej. $d = 5$ o d cercano al percentil 95) indica que la arista es costosa y probablemente representa ruido o un puente no deseado.
- El *router* utilizará pesos como costos en algoritmos de camino más corto o búsqueda de vecinos cercanos, de modo que se seleccionen rutas con bajo costo.

Implicación práctica: la función de actualización del `RicciCleaner` multiplicará d por $(1 + \eta|\kappa|)$ cuando $\kappa < 0$ y por $(1 - \eta\kappa)$ cuando $\kappa > 0$. De esta forma, aristas con curvatura negativa se *alargan* y se penalizan, y aquellas con curvatura positiva se acortan y son más atractivas para el router. Tras aplicar el flujo de Ricci, las aristas cuyo costo supere un umbral d_{\max} serán podadas (eliminadas) por el `RicciCleaner`.

2.2 Justificación teórica

La curvatura de Forman–Ricci se define de manera natural en grafos ponderados. Para un grafo con pesos positivos en aristas w_e y pesos en vértices w_v , el artículo de Estrada et al. (2025) define la curvatura de una arista $e = (v, w)$ como ¹:

$$F_e = w_e \left(\frac{w_v}{w_e} + \frac{w_w}{w_e} - \sum_{e_l \sim v} \frac{w_v}{\sqrt{w_e w_{e_l}}} - \sum_{e_l \sim w} \frac{w_w}{\sqrt{w_e w_{e_l}}} \right).$$

En un grafo *no ponderado*, es decir con $w_v = w_w = w_e = 1$, esta expresión se simplifica a ²

$$F_e = 4 - k_v - k_w,$$

donde k_v y k_w son los grados de los vértices. Así, **aristas que unen nodos de alto grado** ($k_v + k_w > 4$) **presentan curvatura negativa**, lo cual sugiere que funcionan como “puentes” entre regiones densas y pueden contribuir a la propagación excesiva (*over-squashing*).

El mismo artículo resalta que la curvatura de Forman–Ricci es computacionalmente más eficiente que la de Ollivier porque no requiere resolver un problema de transporte óptimo y puede definirse en grafos dirigidos y ponderados ³.

3. Regla de actualización de pesos y poda

Sea $\kappa(e)$ la curvatura de Forman–Ricci calculada para la arista e según la fórmula escogida (ver §4). Sean $d(e) > 0$ su costo actual y $\eta > 0$ el paso de tiempo del flujo. Entonces se define el *Ricci flow discreto* para cada arista no bloqueada como:

1. **Aristas con curvatura negativa** ($\kappa(e) < 0$): aumentar el costo para penalizar la arista:

$$d(e) \leftarrow d(e) \times (1 + \eta |\kappa(e)|).$$

1. **Aristas con curvatura positiva** ($\kappa(e) > 0$): disminuir el costo para premiar la conexión:

$$d(e) \leftarrow d(e) \times (1 - \eta \kappa(e)).$$

1. **Normalización (opcional)**: después de cada iteración del flujo se pueden normalizar los costos para evitar explosiones o colapsos numéricos (p. ej. rescalar todos los $d(e)$ de modo que su media sea 1).
2. **Poda**: tras completar t iteraciones del flujo, se eliminan todas las aristas e no bloqueadas que satisfacen $d(e) \geq d_{\max}$. La elección de d_{\max} puede basarse en un valor absoluto (p. ej. 10) o, de forma más robusta, en un percentil alto (p. ej. el 95 %) de la distribución de pesos actual. **Las aristas marcadas como `locked` no se podan** aunque superen el umbral, ya que representan reglas o caminos críticos.

Esta regla refleja la metáfora geométrica descrita en el documento original: aristas con curvatura negativa se estiran y eventualmente se cortan, expulsando ruido hacia la periferia, mientras que aristas con curvatura positiva se contraen y se consolidan como “caminos preferentes” dentro del grafo.

4. Selección de la fórmula de curvatura (avance del Paso 2)

Para completar el MVP se debe elegir la variante de Forman–Ricci a implementar. Existen varias versiones (no ponderada, ponderada estándar, variantes con triángulos/poliedros, versiones de comunicabilidad). Para el prototipo inicial se propone lo siguiente:

1. **Versión no ponderada (proxy sencillo)**: usar la fórmula $F_e = 4 - k_v - k_w$ ² para grafos simples. Esta versión es barata de calcular ($O(|E|)$) y captura la heterogeneidad de grados local.
2. **Versión ponderada con pesos de vértice**: emplear la fórmula general (8) del artículo de Estrada et al. ¹, interpretando los pesos de vértice w_v como 1 (o como importancia de cada nodo si se dispone de esa información) y los pesos de arista $w_e = d(e)$.
3. **Decisión operativa**: comenzar con la versión no ponderada para validar el *Ricci-cleaning* en un grafo estático. Posteriormente se implementará la versión ponderada y se comparará su impacto en la precisión del router y en la eliminación de ruido.

5. Parámetros iniciales

Se recomiendan los siguientes valores por defecto, ajustables durante los experimentos:

Parámetro	Valor inicial	Comentario
η (tamaño de paso)	0,05	Pequeño para evitar oscilaciones; se ajusta entre 0,01–0,1
t (número de iteraciones)	1–3	En grafos densos, pocas iteraciones bastan para acentuar curvaturas
d_{\max} (umbral de poda)	Percentil 95 de $d(e)$	Adapta el corte al grafo actual
<input type="checkbox"/> locked	<input checked="" type="checkbox"/> True para aristas <input type="checkbox"/> CONSTRAINT	Impide eliminar rutas simbólicas

6. Riesgos y mitigaciones

- **Riesgo de cortar aristas útiles:** al tratar curvaturas negativas como ruido, se pueden eliminar puentes que unen comunidades legítimas. Se mitigará controlando el umbral d_{\max} , manteniendo un registro `removed_edges` para restaurar aristas y comparando métricas de enrutado antes y después.
- **Riesgo de saturar el grafo:** si η es demasiado grande o no se normalizan los pesos, los costos pueden crecer descontroladamente. Se mitigará normalizando pesos y acotando el multiplicador en cada actualización.
- **Dependencia de la definición de curvatura:** la versión no ponderada puede ser insuficiente para grafos con pesos heterogéneos. Se comparará con la versión ponderada en la fase de benchmarking.

7. Conexión con el RAS y el verificador

El `RASGate` utilizará la entropía o incertidumbre del turno para decidir cuándo ejecutar el `RicciCleaner`. Cuando la entropía sea alta o existan señales de posible alucinación (p. ej. falta de evidencia), se activará el modo `CLEAN_AND_ROUTE` y se aplicará el flujo de Ricci. Tras la limpieza, el `Router` seleccionará nodos y caminos usando los costos ajustados. Si se detectan afirmaciones factuales sensibles, el `VerifierPolicyGate` elegirá aplicar CoVe o PRM dependiendo del modo (`FAST`, `CLEAN_AND_ROUTE`, `VERIFY`).

Conclusión: esta especificación fija que los pesos de las aristas del grafo representan distancias o costos de tránsito, justifica la adopción de la curvatura de Forman-Ricci para detectar conexiones ruidosas y define una regla consistente para estirar, contraer y podar aristas en función de la curvatura. Asimismo, deja encaminado el próximo paso: elegir e implementar la fórmula de curvatura (no ponderada vs ponderada) y comparar su impacto en el desempeño del agente.

¹ ² ³ `formanricci_communicability_curvature_of_graphs_and_networks.pdf`

https://www.cambridge.org/core/services/aop-cambridge-core/content/view/1D39CA14E9513387FB1E353105F57E20/S0956792525000014a.pdf/formanricci_communicability_curvature_of_graphs_and_networks.pdf