

VEHICLE-SUPRA

Supra-Expert Autonomous Relational Agent

Escuela primaria · Educación universitaria · Nivel de doctorado (PhD)

Autor	Roberto Borda Milan — VEHICLE Systems Lab
Framework base	Pirámide de Borda Milan + VEHICLE v1.1
DOI	10.5281/zenodo.19981738
Repositorio	github.com/vehiclesystemslab/VEHICLE-SUPRA
Versión código	v0.2.2 (GitHub-Ready)
Fecha	Mayo 2026 · Santa Cruz de la Sierra, Bolivia
Evento objetivo	GITEX Global 2026 · Expand North Star · Dubai

Este documento acompaña la demostración interactiva VEHICLE-SUPRA (simulation demo) presentada como visualización HTML/Canvas con física matemática real derivada del código fuente del repositorio oficial. Cada nivel educativo usa la Pirámide de Borda Milan como estructura conceptual unificadora.

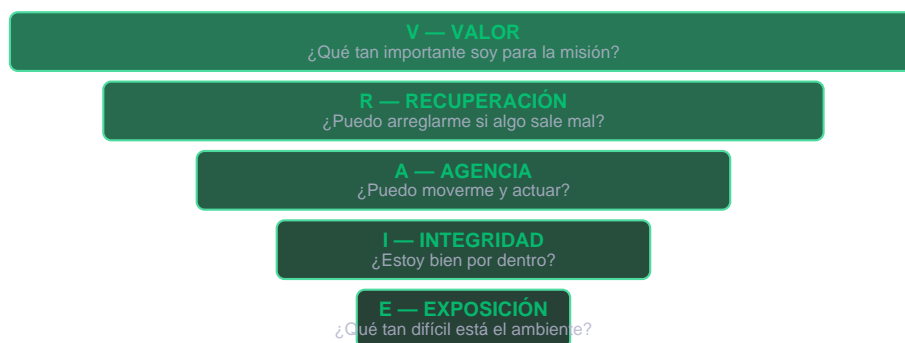
NIVEL 1 — ESCUELA PRIMARIA

¿Cómo funciona un robot que se arregla solo en la Luna?

Imagina que envías robots pequeños a la Luna para explorar y buscar agua. Pero la Luna es peligrosa — hay mucho polvo, hace mucho frío y calor, y a veces los robots se rompen. ¿Qué pasa si no hay nadie cerca para arreglarlos?

VEHICLE-SUPRA es la inteligencia que vive dentro del robot. Le enseña a hacerse cinco preguntas todo el tiempo — y a actuar según las respuestas.

La Pirámide de Borda Milan — Las 5 preguntas del robot



Cada bloque de la pirámide es una pregunta que el robot se hace a sí mismo. La base (Exposición) es lo más básico; la punta (Valor) es lo más especial.

¿Qué hace el robot cuando algo sale mal?

1. MIDE su tensión

Como cuando te duele la barriga y sabes que algo está mal. El robot calcula un número llamado $T(X)$ — si es grande, hay problema.

2. SE ARREGLA SOLO

El robot usa su operador de recuperación (como su médico interno) para volver a estar bien poco a poco.

3. LLAMA A SU GEMELO

Si no puede arreglarse, crea una copia de sí mismo — un robot gemelo — que sigue trabajando mientras el original descansa.

4. NUNCA SE DETIENE

El sistema siempre sigue funcionando. Como cuando en la escuela un compañero te ayuda si tú te enfermaste.

Los colores del robot — ¿Cómo saber si está bien?

COLOR	NOMBRE	¿QUÉ SIGNIFICA?
■ Verde	A0 — Normal	El robot está perfecto. Todo funciona bien.
■ Amarillo	A1 — Presión	Algo difícil pasando afuera. El robot está alerta.
■ Naranja	A3 — Problema	Algo está mal por dentro. El robot se está arreglando.
■ Rojo	A5 — Crisis	El robot llama a su gemelo para que lo ayude.

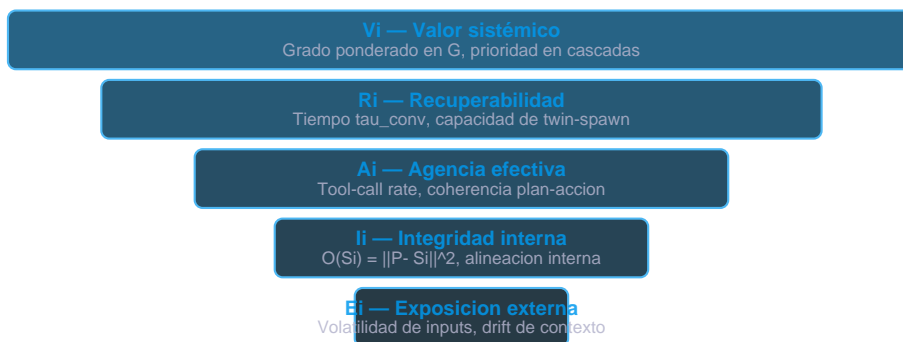
El sistema VEHICLE-SUPRA fue creado por Roberto Borda Milan en Bolivia y será demostrado en Dubai en diciembre de 2026, en la feria de tecnología más grande del mundo.

NIVEL 2 — EDUCACIÓN UNIVERSITARIA

Arquitectura E.I.A.R.(V) y el operador de recuperación gobernada

VEHICLE-SUPRA modela cada agente de IA como un nodo relacionamente estructurado con cinco dimensiones medidas continuamente. En lugar de tratar al agente como una caja negra que produce outputs, la arquitectura lo describe mediante un vector de estado en el hipercubo unitario y una función de tensión que guía su comportamiento.

La Pirámide de Borda Milan — Vector de estado $S_i(t)$



$S_i(t) = (E_i, I_i, A_i, R_i, V_i)$ pertenece a $[0,1]^5$ — estado completo del agente en tiempo t .

La función de tensión $T(X)$

El indicador central del sistema es $T(X)$, una función escalar que mide simultáneamente la discrepancia relacional entre agentes vecinos y la incoherencia interna de cada nodo:

```

T(X)    =    T_ext(X)  +  T_int(X)

T_ext(X) = SUM_{(i,j) en E}  omega_ij * ||S_i - S_j||^2
          --> Discrepancia entre agentes conectados en la red

T_int(X) = lambda * SUM_{i en N}  ||P - S_i||^2
          --> Incoherencia interna (desequilibrio dimensional)
  
```

El operador de recuperación V_{op}

Cuando $T(X)$ supera el umbral T_{limit} , el agente no colapsa ni espera intervención humana. Ejecuta el operador de proyección gobernada, que lo mueve hacia el estado coherente más cercano dentro de la región admisible K :

Gradiente local:

$$\nabla_{S_i} T(X) = \sum_j \omega_{ij} (S_i - S_j) + \lambda \cdot P_- \cdot S_i$$

Operador de proyeccion:

$$V_{op}(S_i) = P_K[S_i - \eta \cdot \nabla T(X)]$$

Actualizacion con relajacion:

$$S_i(t+1) = (1 - \alpha) \cdot S_i(t) + \alpha \cdot V_{op}(S_i(t))$$

Taxonomía de regímenes A0–A6

REGIMEN	NOMBRE	CONDICION	PROTOCOLO
A0	Estable	$T_{int} < 0.15, T_{ext} < 0.20$	Operacion normal
A1	Presion ext	T_{ext} creciente, T_{int} OK	Validacion reforzada
A2	Recuperando	$T(X)$ decreciente post-Vop	Monitor N ciclos
A3	Incoherencia	$O(S_i) > 0.15$, A_i alto, I_i bajo	V_{op} inmediato
A4	Aislado	$\deg_G(i) = 0$	Standby / inspect
A5	Crisis	$O(S_i) > 0.28$, no converge	Twin-node spawn
A6	Cascada	T_{ext} explota multi-nodo	Isolate + rebuild

El protocolo de nodo gemelo (Twin-node)

Si V_{op} no converge en N ciclos (por defecto 20), el nodo entra en régimen A5 y ejecuta el protocolo de replicación: crea un nodo gemelo inicializado en su último estado coherente verificado, transfiriendo la carga operacional mientras el nodo original continúa su recuperación en background.

Condicion de replicacion: $\tau_{conv} > \text{replication_threshold}$ (20 ciclos)

Estado gemelo: $S_j = S_i^*$ (ultimo snapshot coherente)

Red actualizada: $G' = G \cup \{j\}$, redistribucion de aristas

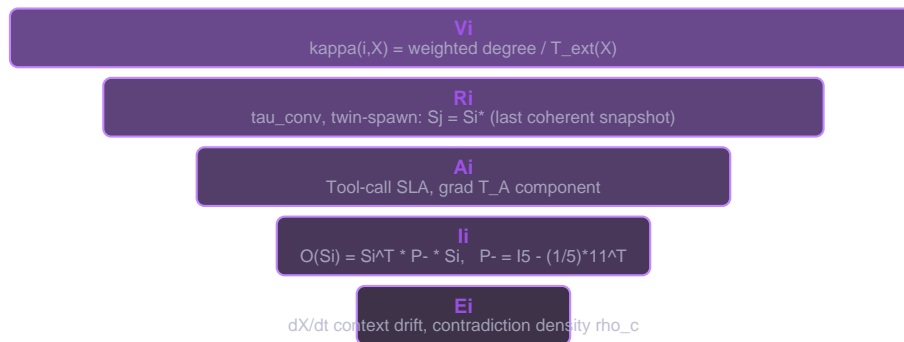
Resultado: $T(X)$ total decrece; sistema nunca se detiene.

NIVEL 3 — DOCTORADO (PhD)

Formalismo matemático completo de VEHICLE-SUPRA

El framework VEHICLE-SUPRA implementa una arquitectura de recuperación gobernada por proyección sobre un espacio de estados estructurado. Lo que sigue es la especificación matemática completa derivada directamente del código fuente del repositorio v0.2.2.

1. Espacio de estados y operador de centrado



Cada dimensión de la pirámide corresponde a una coordenada del vector de estado con su operador de medición formal y su gradiente en $T(X)$.

Estado estructurado: $S_i(t) = (E_i, I_i, A_i, R_i, V_i)$ en $[0,1]^5$

Operador de centrado P^- :

$$P^- = I_5 - (1/5) * 11^T$$

donde I_5 es la identidad 5×5 y 11^T la matriz de unos

Incoherencia interna $O(S_i)$:

$$O(S_i) = ||P^- * S_i||^2 = S_i^T * P^- * S_i$$

Nodo perfectamente balanceado (dim iguales): $O = 0$

Alta dispersion dimensional: O alto \rightarrow regimen A3/A5

2. Región admisible K y condición de coherencia

Region admisible:

$$K = \{ S \text{ en } [0,1]^5 : O(S) \leq T_{\text{limit}} \}$$

T_{limit} configurable via `SupraNodeConfig.t_limit` (default: 0.30)

Condicion de coherencia: nodo coherente ssi S_i en K

$$O(S_i) \leq T_{\text{limit}} \iff ||P^- * S_i||^2 \leq T_{\text{limit}}$$

3. Función de tensión total $T(X)$

Red $G = (N, E)$ con pesos $\omega_{ij} > 0$:

$$T_{\text{ext}}(X) = \sum_{(i,j) \in E} \omega_{ij} * ||s_i - s_j||^2$$

$$T_{\text{int}}(X) = \lambda * \sum_{i \in N} s_i^T * P^- * s_i$$

$$T(X) = T_{\text{ext}}(X) + T_{\text{int}}(X)$$

Gradiente respecto a s_i :

$$\nabla_{s_i} T(X) = \sum_{j:(i,j) \in E} \omega_{ij} * (s_i - s_j) + \lambda * P^- * s_i$$

Primer termino: correccion relacional

Segundo termino: correccion de incoherencia interna

4. Operador de proyección P_K y V_{op}

Proyeccion onto K (implementation: binary search 64 iteraciones):

$$P_K(z) = \operatorname{argmin}_{s \in K} ||s - z||^2$$

Si z ya en K : $P_K(z) = \text{clip}(z, [0,1]^5)$

Si no: busqueda sobre μ s.t. $O(z - \mu * P^- * z) = T_{\text{limit}}$

Operador de correccion gobernada V_{op} :

$$V_{op}(s_i; X) = P_K[s_i - \eta * \nabla_{s_i} T(X)]$$

η = paso de descenso (default: 0.10)

Actualizacion con relajacion convexa:

$$s_i(t+1) = (1 - \alpha) * s_i(t) + \alpha * V_{op}(s_i(t); X)$$

α en $(0,1]$ (default: 0.30) -- convergencia controlada

5. Criterio de convergencia y protocolo Twin-Node

Convergencia local: nodo i recuperado ssi $O(s_i(t+1)) \leq T_{\text{limit}}$

Convergencia global: todos los nodos en K

Si $\tau_{\text{conv}}(i) > \text{replication_threshold}$ (20 ciclos):

--> Activar protocolo Twin-Node para nodo i

$s_j = s_i^*$ (ultimo snapshot con $O(s_i^*) \leq T_{\text{limit}}$)

$G' = G \cup \{\text{nodo } j\}$

Redistribucion de aristas: transferir carga de i a j

Nodo i : continua recovery en background

Sistema: nunca se detiene (zero unplanned downtime)

6. Clasificación formal de regímenes A0–A6

REGIMEN	NOMBRE	CONDICION	PROTOCOLO
A0	Estable	$T_{int} < 0.15, T_{ext} < 0.20$	Operacion normal
A1	Presion ext	T_{ext} creciente, T_{int} OK	Validacion reforzada
A2	Recuperando	$T(X)$ decreciente post-Vop	Monitor N ciclos
A3	Incoherencia	$O(S_i) > 0.15, A_i$ alto, I_i bajo	V_{op} inmediato
A4	Aislado	$deg_G(i) = 0$	Standby / inspect
A5	Crisis	$O(S_i) > 0.28$, no converge	Twin-node spawn
A6	Cascada	T_{ext} explota multi-nodo	Isolate + rebuild

7. Parámetros de configuración (SupraNodeConfig)

Parámetro	Default	Descripción
<code>t_limit</code>	0.30	Umbral de coherencia: $O(S_i) \leq t_limit$ define K
<code>t_int_warning</code>	0.15	Umbral A3: incoherencia interna alta
<code>t_int_crisis</code>	0.28	Umbral A5: crisis, activa twin-spawn
<code>t_ext_warning</code>	0.20	Umbral A1: presión relacional externa
<code>eta</code>	0.10	Paso de descenso gradiente en V_{op}
<code>alpha</code>	0.30	Parámetro de relajación ($0 < \alpha \leq 1$)
<code>replication_threshold</code>	20	Ciclos sin convergencia -> twin-spawn
<code>lam (lambda)</code>	1.0	Peso de T_{int} relativo a T_{ext} en $T(X)$

Referencia: Borda Milan, R. (2026). *The Borda Milan Pyramid and the VEHICLE Formula-as-Architecture*. VEHICLE Systems Lab / Zenodo. DOI: 10.5281/zenodo.19981738. Repositorio: github.com/vehiclesystemslab/VEHICLE-SUPRA (MIT License)