

Manual de Implementación

OctoMatrix como Middleware de Validación de Eventos

1. Propósito del Documento

Este manual describe los pasos técnicos y conceptuales necesarios para implementar **OctoMatrix** como un middleware de validación de eventos dentro de una arquitectura basada en productores y consumidores. El documento está diseñado para entornos académicos, de laboratorio y prototipos controlados, priorizando claridad arquitectónica sobre optimizaciones de producción.

OctoMatrix **no reemplaza** mecanismos de seguridad existentes; actúa como una **capa adicional de correlación y observabilidad** entre comportamiento de red e intención de aplicación.

2. Alcance de la Implementación

La implementación descrita cubre:

- * Validación de eventos a nivel de aplicación usando modelos entrenados (archivo `*.pkl`).
- * Correlación básica entre identidad de red (IP/MAC) y acciones observadas.
- * Integración lógica con aplicaciones existentes (ej. Parchate Pereira).

Fuera de alcance:

- * Inspección profunda de paquetes (DPI).
- * Reemplazo de sistemas IAM, ACLs o firewalls.
- * Ambientes productivos de alta criticidad.

3. Componentes del Sistema

3.1 Aplicación (Parchate Pereira)

<https://github.com/SPotes22/Pereira-Explorer-AI-Parchate-Pereira>

The screenshot displays the Parchate Pereira application interface, which includes the following sections:

- Estadísticas de Eventos:** Shows a total of 8 events, with 7 being free and 1 paid. It also provides a breakdown by category: baile (1), cine (1), cultura (3), feria (1), gastronomia (1), and musica (1). A button to "Actualizar Stats" is present.
- Búsqueda Avanzada:** A search bar containing "alert(1)" and several filters: Música, Cine, Gastronomía, Teatro, and Ferias. Buttons for "Buscar Eventos" and "Ver Todos" are also shown.
- Eventos Reales de Pereira:** A section titled "Resultados para 'alert(1)'" which states "No se encontraron eventos".
- HelpBoy con Data Real:** A section showing the message "alert(1)". A button to "Obtener Recomendación Inteligente" is available.
- Premium MBA Service:** A section titled "Premium MBA Service" with the subtext "Gemini AI + 82% context match filter". It shows a message about alternative music festivals and a button to "Obtener Recomendación Premium".
- Recomendaciones:** Two recommended events are listed:
 - Festival Internacional de Poesía Luna de Locos:** XIX edición del festival internacional de poesía. Date: 2025-08-25 18:00. Location: Teatro Área Cultural Cámara de Comercio de Pereira. Status: Gratis.
 - Convivencia Eje Rock 2025:** Festival internacional de rock con bandas locales e internacionales.

<https://pereira-explorer-ai-parchate-pereira.onrender.com/>

| |
|--|
| Dec 31 |
| 04:04:52 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:04:52 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:04:57 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:04:57 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:02 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:02 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:07 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:07 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:09 PM [rhrkz] ✓ Data cargada: 8 eventos procesados |
| 04:05:09 PM [rhrkz] 127.0.0.1 - - [31/Dec/2025:21:05:09 +0000] "GET /api/events/search?q=--admin HTTP/1.1" 200 71 "https://pereira-explorer-ai-parchate-pereira.onrender.com/" "Mozilla/5.0 (X11; CrOS x86_64 14541.0.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36" |
| 04:05:12 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:12 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:12 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:12 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:17 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:17 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:22 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:22 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:24 PM [rhrkz] ✓ Data cargada: 8 eventos procesados |
| 04:05:24 PM [rhrkz] 127.0.0.1 - - [31/Dec/2025:21:05:24 +0000] "GET /api/events/search?q=alert(1) HTTP/1.1" 200 72 "https://pereira-explorer-ai-parchate-pereira.onrender.com/" "Mozilla/5.0 (X11; CrOS x86_64 14541.0.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36" |
| 04:05:27 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:27 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:32 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:32 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:37 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:37 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:42 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:42 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:42 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:42 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |
| 04:05:47 PM [rhrkz] 10.203.25.224 - - [31/Dec/2025:21:05:47 +0000] "GET / HTTP/1.1" 200 20823 "-" "Render/1.0" |

La aplicación cliente es responsable de:

- * Generar eventos de negocio.
- * Consumir decisiones de validación emitidas por OctoMatrix.
- * No contiene lógica de seguridad predictiva.

Archivos de referencia:

- * `app.py`
- * `brain.py`

3.2 OctoMatrix Core

OctoMatrix es el componente encargado de:

- * Cargar el modelo entrenado (`.pkl`).
- * Recibir eventos normalizados.
- * Evaluar patrones y emitir decisiones (válido / sospechoso).

Archivo de referencia:

- * `octomatrix_poc_moe_owasp.py`
- <https://www.kaggle.com/code/santiagopotes/octomatrix-poc-moe-owasp/log>

3.3 Modelo de Machine Learning

El modelo entrenado:

- * Fue generado offline.
- * Presenta una precisión superior al 82%.
- * No se entrena en tiempo real.

El modelo se carga en memoria al iniciar OctoMatrix.

4. Flujo General de Implementación

1. La aplicación genera un evento.
 2. El evento se normaliza (estructura JSON o dict).
 3. El evento se envía a OctoMatrix.
 4. OctoMatrix evalúa el evento usando el modelo.
 5. Se retorna una decisión.
 6. La aplicación actúa según la decisión.
-

5. Requisitos del Entorno

5.1 Software

- * Python 3.9+
- * Librerías estándar de ML (según entrenamiento previo)
- * Entorno virtual recomendado

5.2 Hardware

- * CPU estándar (no GPU requerida)
 - * Mínimo 4 GB RAM
-

6. Preparación del Entorno

1. Crear entorno virtual.
2. Instalar dependencias.
3. Ubicar el archivo ` `.pkl` en la ruta definida.
4. Verificar permisos de lectura.

7. Inicialización de OctoMatrix

Al iniciar OctoMatrix:

- * Se carga el modelo en memoria.
- * Se validan estructuras internas.
- * El sistema queda en modo escucha.

Recomendación: inicializar una sola vez por proceso.

8. Integración con la Aplicación

La aplicación debe:

- * Invocar a OctoMatrix como servicio o módulo.
- * Enviar eventos estructurados.
- * Interpretar la respuesta sin lógica adicional.

OctoMatrix **no decide acciones finales**, solo emite evaluaciones.

9. Manejo de Resultados

Las respuestas posibles incluyen:

- * Evento válido.
- * Evento sospechoso.

La aplicación decide:

- * Registrar.
 - * Bloquear.
 - * Alertar.
-

10. Buenas Prácticas

- * Mantener el modelo desacoplado del código.
 - * No modificar el `.pkl` en caliente.
 - * Registrar decisiones para análisis posterior.
 - * No asumir falsos positivos como errores.
-

11. Limitaciones Conocidas

- * Dependencia de calidad del entrenamiento.
- * No detección de ataques completamente nuevos.
- * No reemplaza validaciones determinísticas.

12. Extensiones Futuras

- * Correlación con identidad de red (MAC/IP).
 - * Integración con Kafka.
 - * Uso de métricas SNMP.
 - * Soporte multi-modelo.
-

13. Consideraciones Académicas

Este manual está diseñado para facilitar:

- * Reproducción del experimento.
- * Evaluación técnica en laboratorio.
- * Discusión arquitectónica.

No pretende ser una guía de despliegue comercial.

14. Cierre

OctoMatrix debe entenderse como una **capa experimental de validación**, útil para explorar nuevas formas de correlacionar comportamiento, identidad y eventos sin comprometer la arquitectura existente.