

Industrial Event Standard v2.0

Especificación técnica del protocolo de eventos que **todos los agentes deben implementar** para publicar eventos a la plataforma CTRL+HACK.

Principios

- Todos los eventos comparten una estructura base obligatoria
 - Los datos específicos del agente van dentro del bloque `data`
 - No se permiten campos arbitrarios fuera de la estructura establecida
 - Nomenclatura en `snake_case` en todos los módulos
 - Todos los timestamps en formato ISO 8601 (UTC)
-

Estructura del evento

```
{
  "event_id": "string (requerido)",
  "timestamp": "string ISO 8601 UTC (requerido)",
  "platform_version": "string (requerido)",
  "module": {
    "id": "string (requerido)",
    "version": "string (requerido)"
  },
  "asset": {
    "asset_id": "string (requerido)",
    "asset_type": "string (opcional)",
    "plant_id": "string (opcional)",
    "area_id": "string (opcional)",
    "line_id": "string (opcional)",
    "location": "string (opcional)"
  },
  "event": {
    "type": "string (requerido)",
```

```

    "category": "string (opcional)",
    "severity": "string (opcional)"
  },
  "data": { "...campos del módulo..." },
  "metadata": { "...campos contextuales..." }
}

```

Campos requeridos (validados por el API)

Campo	Tipo	Descripción
<code>event_id</code>	string	Identificador único del evento (UUID recomendado)
<code>timestamp</code>	string	Formato ISO 8601 UTC, ej: <code>2026-05-23T14:30:00Z</code>
<code>platform_version</code>	string	Versión de la plataforma: <code>"2.0.0"</code>
<code>module.id</code>	string	ID del módulo en snake_case, ej: <code>"vision_quality_inspection"</code>
<code>module.version</code>	string	Versión semántica del módulo, ej: <code>"1.0.0"</code>
<code>asset.asset_id</code>	string	ID del activo físico o lógico, consistente entre módulos
<code>event.type</code>	string	Nombre del evento en snake_case, ej: <code>"defect_detected"</code>

Un evento sin alguno de estos campos es rechazado por el API con HTTP 400.

Campos opcionales recomendados

Campo	Valores válidos
<code>event.category</code>	<code>quality</code> , <code>productivity</code> , <code>maintenance</code> , <code>energy</code> , <code>safety</code> , <code>configuration</code> , <code>system</code>
<code>event.severity</code>	<code>low</code> , <code>medium</code> , <code>high</code> , <code>critical</code>
<code>metadata.shift</code>	<code>"A"</code> , <code>"B"</code> , <code>"C"</code>
<code>metadata.operator_id</code>	ID del operador

<code>metadata.production_order</code>	Orden de producción activa
--	----------------------------

Tipos de evento

Tipo	Descripción	Ejemplos
<code>event_driven</code>	Ocurre cuando sucede algo específico	<code>defect_detected</code> , <code>machine_stopped</code> , <code>anomaly_confirmed</code>
<code>state_update</code>	Actualización de estado del activo	<code>machine_status_changed</code> , <code>device_registered</code>
<code>periodic_metrics</code>	Reporte periódico de métricas	<code>metrics_snapshot</code> , <code>energy_reading</code> , <code>kpi_report</code>

Endpoints de la plataforma

Método	Endpoint	Descripción
<code>POST</code>	<code>/api/v1/events</code>	Publicar un evento
<code>GET</code>	<code>/api/v1/events</code>	Consultar eventos (params: <code>start</code> , <code>end</code> , <code>module_id</code> , <code>asset_id</code>)
<code>GET</code>	<code>/api/v1/health</code>	Verificar conectividad con la plataforma

Autenticación: header `x-api-key: <tu-api-key>`

Ejemplos completos válidos

Los tres ejemplos a continuación validan correctamente contra el schema del API.

ERP — Stock bajo mínimo

```
{
  "event_id": "3f9a1c2d-0e4b-4f8a-9c3d-7b2e5f1a8c4d",
  "timestamp": "2026-05-23T14:30:00Z",
  "platform_version": "2.0.0",
  "module": { "id": "erp_inventory_monitor", "version": "1.0.0" },
  "asset": { "asset_id": "ALMACEN-01", "asset_type": "warehouse",
"plant_id": "PLT-JUAREZ-01" },
  "event": { "type": "stock_below_minimum", "category":
"productivity", "severity": "high" },
  "data": {
    "sku": "TORNILLO-M8-ACERO",
    "current_stock": 120,
    "minimum_stock": 500,
    "auto_order_generated": true
  }
}
```

Visión Artificial — Defecto detectado

```
{
  "event_id": "7c4e2a1f-8b3d-4e9f-a0c2-5d6b8e3f9a1c",
  "timestamp": "2026-05-23T15:12:33Z",
  "platform_version": "2.0.0",
  "module": { "id": "vision_quality_inspection", "version":
"1.0.0" },
  "asset": { "asset_id": "CAM-LINE2-01", "asset_type": "camera",
"line_id": "LINE-2" },
  "event": { "type": "defect_detected", "category": "quality",
"severity": "medium" },
  "data": {
    "defect_type": "surface_scratch",
    "confidence": 0.94,
    "inference_latency_ms": 22
  }
}
```

```
}  
}
```

AI & ML — Falla predicha

```
{  
  "event_id": "b1d4f7e2-3a9c-4b8e-0d5f-2c7a1e9b4f6d",  
  "timestamp": "2026-05-23T16:05:47Z",  
  "platform_version": "2.0.0",  
  "module": { "id": "aiml_failure_predictor", "version": "1.0.0"  
},  
  "asset": { "asset_id": "MOTOR-BANDA-03", "asset_type": "motor",  
"line_id": "LINE-1" },  
  "event": { "type": "failure_predicted", "category":  
"maintenance", "severity": "high" },  
  "data": {  
    "failure_type": "bearing_wear",  
    "probability": 0.87,  
    "estimated_hours_to_failure": 14  
  }  
}
```

Errores comunes

Error	Causa	Solución
HTTP 400 - event_id missing	Campo requerido ausente	Generar UUID antes de enviar
HTTP 400 - timestamp format invalid	Formato incorrecto	Usar ISO 8601 con Z, ej: 2026-05-23T14:30:00Z
HTTP 401 - Unauthorized	API key inválida o ausente	Incluir header x-api-key

HTTP 400 - additionalProperties	Campo extra fuera del schema	Solo campos del schema están permitidos en <code>module</code> , <code>asset</code> y <code>event</code>
------------------------------------	---------------------------------	--

Los archivos JSON completos están en [../assets/examples/](#) .