

FICHA TÉCNICA

AITHANEX AI

Plataforma para crear, orquestar y ejecutar agentes IA

Tipo:
Ficha técnica

Producto:
AITHANEX AI

Sistema:
AITHANEX

Audiencia:
Ingeniería, instalación, integración y soporte

Versión del documento:
1.0

Fecha de revisión:
2026-10-04

Tipo de producto

Plataforma para crear, orquestar y ejecutar agentes IA. Incluye ruteo entre modelos LLM (modelos grandes de lenguaje), RAG (recuperación de conocimiento autorizado para responder con contexto) y MCP (protocolo para conectar herramientas y acciones externas).

Aporta asistencia, análisis y automatización supervisada sobre los datos, documentos y procesos del sistema **AITHANEX**.

Función dentro del sistema

AITHANEX AI permite definir, desplegar y ejecutar agentes IA especializados.

Estos agentes operan sobre datos, documentos, permisos y eventos de **AITHANEX**.

Los agentes actúan como intermediarios inteligentes:

- Acceden a bases de conocimiento mediante RAG: documentación, históricos, conversaciones y telemetría.
- Rutean tareas entre modelos LLM según reglas de coste, privacidad, latencia o complejidad.
- Se integran con herramientas externas mediante MCP, APIs, webhooks y sistemas corporativos.
- Se usan en **AITHANEX Things** para consultar datos operativos, asistir al operador y ejecutar acciones controladas.
- Se usan en **AITHANEX Customer service** para aportar inteligencia a bots de atención, clasificación y respuestas asistidas.

Componentes principales

- **Diseñador de agentes.** Crea agentes con rol, instrucciones, permisos, fuentes y modelo asociado.
- **Diseñador de flujos.** Define flujos visuales que pueden usar agentes, reglas, eventos, APIs o acciones externas.
- **Motor de ejecución.** Ejecuta agentes en tiempo real. Gestiona concurrencia, reintentos, errores y tiempos de espera.
- **Orquestador IA.** Coordina RAG, MCP, selección de modelo, permisos, ejecución y auditoría en servidor.
- **Servidor RAG.** Gestiona recuperación de conocimiento, índices, fuentes autorizadas y contexto para agentes.
- **Servidor MCP.** Expone herramientas, conectores y acciones externas bajo permisos y trazabilidad.
- **Enrutador de modelos.** Selecciona el modelo según reglas de coste, privacidad, latencia, disponibilidad y complejidad.
- **Consola RAG / MCP / LLM.** Interfaz de administración para configurar fuentes, herramientas, modelos, cuotas y pruebas.
- **Informes de uso.** Muestra consumo de tokens, conversaciones, ejecuciones, costes estimados y uso por modelo.

Arquitectura de RAG, MCP y LLM

La arquitectura se organiza alrededor de una consola de administración y un orquestador IA.

Desde la consola se configuran fuentes de conocimiento, herramientas, modelos, cuotas y políticas. El orquestador aplica esa configuración en cada ejecución y coordina recuperación de contexto, uso de herramientas, selección de modelo, auditoría e informes de uso.

Datos

- **Agentes.** Creación, gestión y versionado de agentes personalizados por proyecto o caso de uso.
- **Modelos LLM.** Soporte para Claude, GPT, Llama y otros modelos (extensible por configuración).
- **Ruteo.** Reglas definidas por cliente: latencia, coste, precisión y política de privacidad.
- **Contexto RAG.** Documentación, históricos AITHANEX, datos en tiempo real y bases de datos externas.
- **Integraciones MCP.** APIs, webhooks, sistemas externos y herramientas de análisis.
- **Flujos de trabajo.** Flujos visuales que coordinan agentes, eventos, reglas, APIs y acciones externas.
- **Permisos.** Control granular: quién crea agentes, quién ejecuta acciones y sobre qué recursos.
- **Observabilidad.** Tokens, conversaciones, ejecuciones, errores, coste estimado y uso por modelo.
- **Límites configurables.** Tiempos de espera, reintentos, límites de coste y cuota de ejecuciones por agente o proyecto.

Casos de uso principales

1. Inteligencia para AITHANEX Customer service

Objetivo verificable: Aportar inteligencia a bots de atención, clasificación y respuesta asistida.

Técnica:

- Entrada: conversación, canal, contacto, historial y contexto de atención.
- RAG: consulta documentación, base de conocimiento y respuestas autorizadas.
- Enrutador de modelos: selecciona el modelo según complejidad, coste y política del proyecto.
- Derivación: transfiere la conversación a un operador humano cuando el flujo lo requiere.

Control operativo:

- Escalada: deriva a un operador humano si la confianza es baja o se solicita.
- Auditoría: registra consulta, fuentes, modelo, tokens y resultado.

Ejemplo técnico:

Paso	Ejemplo
Entrada	Consulta de cliente desde un canal conectado.
Contexto RAG	Documentación de producto, políticas de soporte e historial de conversación.
Ruteo	Selección del modelo adecuado para respuesta o clasificación.
Salida bot	Respuesta sugerida, etiqueta, prioridad o solicitud de datos.
Derivación	Transferencia a agente humano con historial y contexto.

2. Control asistido en AITHANEX Things

Objetivo verificable: Consultar datos operativos y ejecutar acciones controladas sobre dispositivos cuando el despliegue lo habilita.

Técnica:

- Entrada: telemetría, estado, alarma, activo, usuario y permisos.
- Consulta: el agente interpreta datos disponibles en AITHANEX Things.
- Decisión: el flujo valida reglas, permisos y límites de actuación.
- Ejecución: la acción se lanza como comando, configuración, incidencia o llamada externa.
- Registro: se guardan fuentes, modelo, tokens, usuario, acción y resultado.

Ejemplo técnico:

Paso	Ejemplo
Consulta	"¿Qué cargadores tienen alarmas activas?"
Contexto	Estados, alarmas, ubicación y últimos eventos desde Things.
Sugerencia	Priorizar equipos con alarma crítica o pérdida de comunicación.
Acción controlada	Crear incidencia, notificar técnico o ejecutar comando autorizado.
Auditoría	Registro de usuario, agente, modelo, fuentes, tokens y resultado.

3. Automatización con flujos de trabajo y MCP

Objetivo verificable: Coordinar procesos que combinan reglas, agentes, APIs y sistemas externos.

Técnica:

- Flujo: evento, regla o solicitud inicia el workflow.
- Agente: interpreta contexto si el paso lo requiere.
- MCP: ejecuta llamadas autorizadas a herramientas externas.
- Control: valida permisos, límites, errores y resultado.
- Observabilidad: registra ejecución, modelo, tokens, coste estimado y trazabilidad.

Ejemplo técnico:

Paso	Ejemplo
Evento	Alarma crítica en una instalación.
Evaluación	Consulta histórico y reglas de escalado.
Integración	Crea ticket, avisa al equipo y actualiza sistema externo.
Control	Verifica permisos y registra cada acción.
Resultado	Incidencia trazable con contexto técnico y fuentes.

Enlace técnico

- **AITHANEX Things.** Aporta telemetría, alarmas y estados como contexto para agentes; recibe acciones controladas sobre dispositivos.
- **AITHANEX Customer service.** Usa agentes para bots, clasificación, respuestas asistidas y derivación a operadores humanos.
- **AITHA Core.** Publica datos y eventos que sirven como contexto operativo para agentes y flujos de trabajo.

- **AURA X36.** Aporta eventos operativos, seguridad e integración industrial para análisis y automatización.
- **Herramientas externas.** APIs, webhooks y conectores gestionados mediante MCP con permisos y trazabilidad.
- **AITHANEX base.** Permisos, históricos, auditoría, usuarios y trazabilidad común del sistema.

Referencias

- AITHANEX: <https://aitha.es/es/plataforma/>
- AITHANEX AI: <https://aitha.es/es/plataforma/aitha-ai/>
- AITHANEX Things: <https://aitha.es/es/plataforma/things/tecnica/>
- AITHANEX Customer service: <https://aitha.es/es/plataforma/customer-service/tecnica/>
- AITHA Core: <https://aitha.es/es/aitha-core/tecnica/>
- AURA X36: <https://aitha.es/es/productos/aura-x36/tecnica/>
- AITHANEX: <https://aitha.es/es/plataforma/tecnica/>

AITHANEX AI