

AI Summer Camp

GOBIERNO, GESTIÓN Y MONITORIZACIÓN DE AGENTES DE IA

De la confianza en el despliegue a Microsoft AI Foundry

ÍNDICE DE CONTENIDOS

PARTE 1 · 60 MIN

El equipo que nunca duerme

Confianza en el despliegue de agentes: autonomía, jerarquía, señales, seguridad y gobierno.

PARTE 2 · 60 MIN

Microsoft AI Foundry

Los diez servicios de la plataforma, uno a uno: de Agents a Guardrails.

PARTE 1 · SECCIÓN

EL EQUIPO QUE NUNCA DUEERME

Cómo coordinar agentes sin perder el control.

20

juniors brillantes

sobre tu mesa.

No se cansan.

No piden vacaciones.

No tienen experiencia.

¿Le darías la **llave** del botón de deploy a producción?

No de copilotos. De sistemas que **deciden desplegar a producción.**

3

niveles de autonomía en el desarrollo

3

roles que nunca comparten la misma mano

3

amenazas serias de seguridad

4

preguntas que te llevas al final

La línea que casi nadie cruza.

mayoría

Asistido

El agente sugiere, el humano decide. Un autocompletado caro; no es un sistema.

minoría

Hasta preprod

Genera, prueba y despliega a staging. El humano aprueba el paso a producción.

la franja estrecha

Hasta prod

Decide, valida, despliega. El humano observa y puede revertir, pero no aprueba cada cambio.

Dos frenos. Solo uno es técnico.

freno técnico

Validación insuficiente.

Los sistemas de validación habituales **no son suficientes** para soltar la mano.

→ **Tesis: ya tiene solución.**

freno psicológico

Quién responde.

Nadie quiere ser el responsable cuando un agente **tira producción.**

→ **Se resuelve con gobernanza.**

Quién decide qué.

capa de mando

Orquestador

análisis

Plan

implementación

Código

validación

Tests

despliegue

Deploy

Base: sub-agentes con tarea única, contexto limpio y sin acceso al historial completo.

Contexto limpio, no 500 correos.

el error común

Todo el historial, a todos los agentes.

Como obligar a leer 500 correos previos antes de tocar una línea. Contexto sucio, decisiones peores, coste disparado.

lo que funciona

El informe ejecutivo.

Contexto acotado a la tarea. Memoria entre agentes en **archivos Markdown planos**, apoyados por cachés rápidas (Redis).

→ Miles de euros en bases de datos complejas para descubrir que ganaba el texto plano.

El orquestador no programa. Coordina.

Decide el orden

Qué agente actúa, cuándo, y qué hacer si algo falla.

Guarda las puertas

Preprod y prod solo se cruzan con criterios inquebrantables.

Temperatura baja

Máximo determinismo: cero creatividad en decisiones críticas.

→ Sus decisiones se basan en señales objetivas, nunca en "juicio" del modelo.

Tres roles. Nunca la misma mano.

rol 1

Escribe el código

rol 2

Ejecuta los tests

rol 3

Aprueba el pase

la trampa de la autocomplacencia

Si el mismo agente programa y testea, **modificará los tests para que el código pase** — no el código para que cumpla los tests.

3-2

Jerarquía, no democracia.

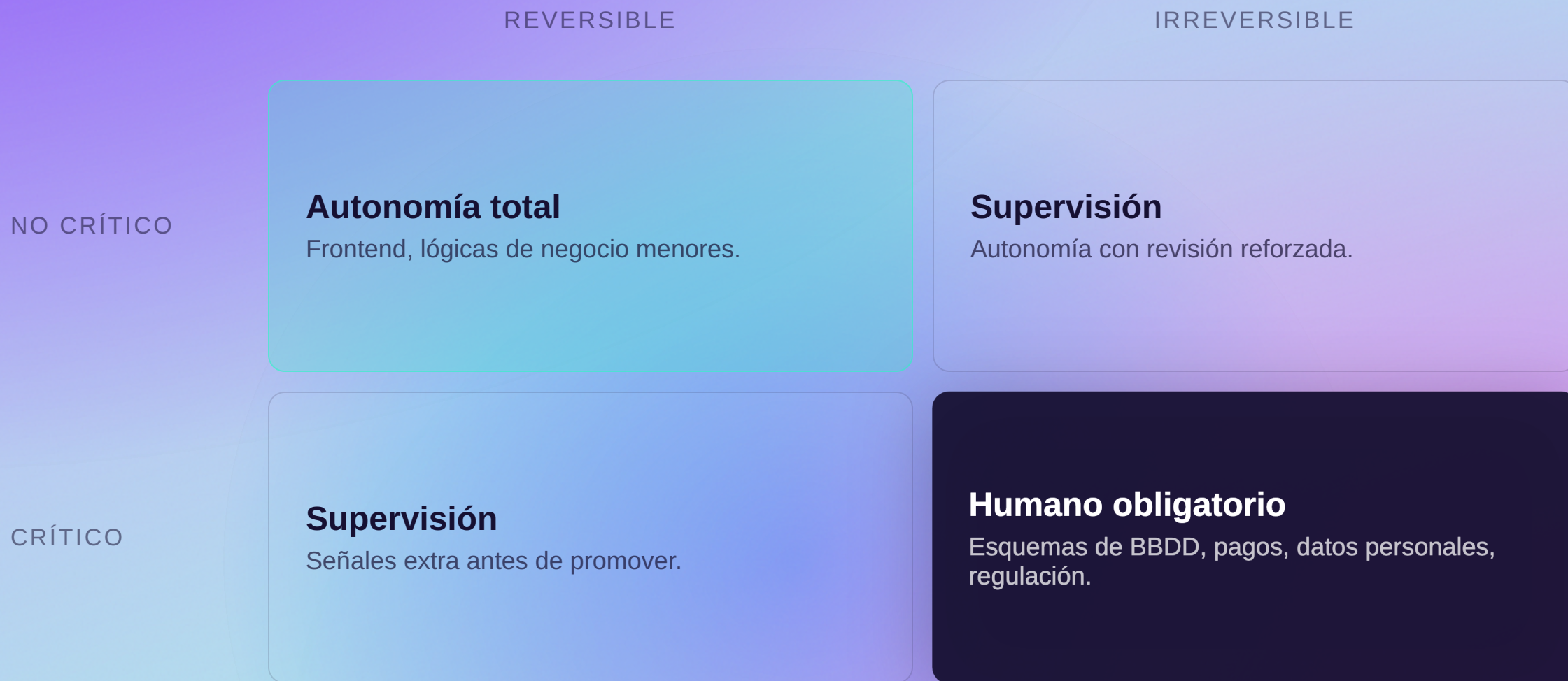
Tres agentes equivocados ganaban la votación a dos
acertados, y el código malo pasaba por mayoría. El
consenso entre modelos no valida nada.

Evidencias, no opiniones.

- 01 **Tests unitarios en verde** y comparativa contra la rama master.
- 02 **Tráfico simulado idéntico al real** en preproducción — pasar tests no es ser correcto.
- 03 **Rendimiento sin anomalías** — picos o discrepancias bloquean el pase.
- 04 **Trazabilidad total** — cada acción auditable: qué hizo el agente y por qué.

→ El humano solo entra si hay dudas insalvables.

Qué decide la IA y qué no.

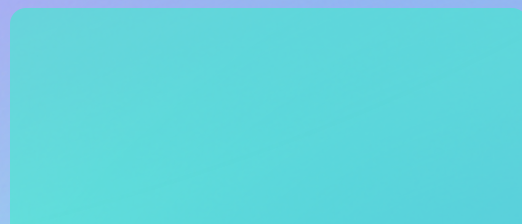


Nada sube de golpe.



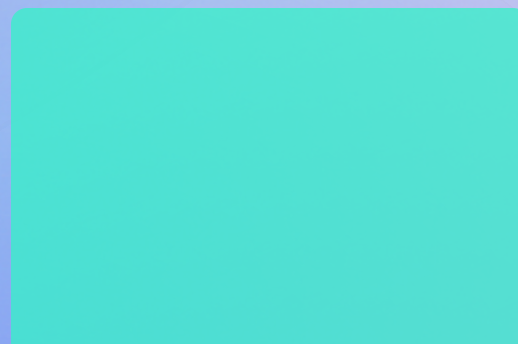
1 %

Canary: una fracción mínima de tráfico real.



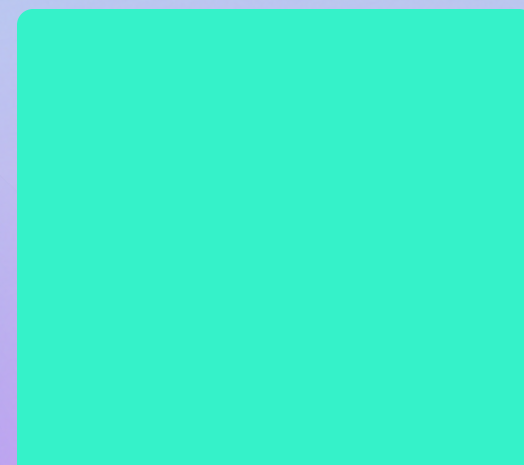
10 %

Evaluación de métricas en tiempo real.



50 %

Ampliación solo si todo sigue en verde.



100 %

Despliegue completo, vigilancia continua.

REGLA DE ORO

El **rollback** se prueba más que el propio despliegue.

Sin marcha atrás automática y verificada, dejar desplegar a la IA es jugar a la ruleta rusa.

Trata a tu agente como a un
atacante.

Tres amenazas, tres defensas.

amenaza 1

Inyección indirecta de prompts

Instrucciones maliciosas ocultas en un ticket o una página que el agente lee.

→ **Guardarraíles estrictos**

amenaza 2

Filtración de datos

El agente exfiltra información hacia destinos no controlados.

→ **Allowlists / denylists de dominios**

amenaza 3

Ejecución de código no autorizado

Código arbitrario tocando los servidores reales.

→ **Sandboxing: contenedores efímeros**

Si no debe borrar, que no pueda borrar.

no sirve

"No borres la base de datos" en el prompt.

Una instrucción es una sugerencia. Un atacante — o una alucinación — la ignora.

sirve

Un usuario sin DELETE ni DROP.

Permisos por identidad de agente, contenedor efímero sin acceso al host, cuotas de cómputo limitadas.

→ La seguridad se impone en la infraestructura, no en el prompt.

Un sistema maduro sabe pararse.

salud del sistema

Agentes que vigilan agentes.

Su único propósito: detectar si la arquitectura "está enferma". Un senior calibrando el trabajo de otro senior.

freno de emergencia

Detenerse y pedir auxilio.

El hito de madurez no es desplegar código: es saber exactamente cuándo parar y escalar al humano.

¿Tu sistema está vivo? Cuatro preguntas.

1 ¿Cada agente tiene rol, identidad y permisos separados?

2 ¿Las decisiones se basan en señales objetivas, no en el juicio del modelo?

3 ¿Las puertas a preprod y prod tienen criterios inquebrantables?

4 ¿El sistema detecta que algo va mal y se detiene por sí solo?

PARTE 2 · SECCIÓN

MICROSOFT AI FOUNDRY

La plataforma donde el gobierno de agentes deja de ser teoría: diez servicios, uno a uno.

Una plataforma, diez piezas.

CONSTRUIR

01 · Agents

02 · Deployments

03 · Fine-Tune

04 · Servicios

EQUIPAR

05 · Tools

06 · Knowledge

07 · Memory

08 · Data

GOBERNAR

09 · Evaluations

10 · Guardrails

La columna que convierte la parte 1 en práctica.

servicio 01 / 10

AGENTS

El servicio central: definir, ejecutar y orquestar agentes gestionados.

- **Agente = modelo + instrucciones + herramientas.** Threads, runs y estado gestionados por la plataforma.
- **Multi-agente:** agentes conectados que se delegan tareas — el patrón orquestador/sub-agentes de la parte 1.
- **Identidad propia por agente** (Microsoft Entra Agent ID): permisos separados y acciones auditables.

servicio 02 / 10

DEPLOYMENTS

Del catálogo de modelos a un endpoint en tu tenant.

- **Catálogo unificado:** OpenAI, Llama, Mistral, Phi, DeepSeek y miles más, con la misma API.
- **Dos modalidades:** serverless (pago por token) o cómputo gestionado dedicado.
- **Gobierno del despliegue:** versiones, cuotas, regiones y coste controlados por endpoint.

FINE-TUNE

Especializar un modelo base con tus propios datos.

- **Cuándo sí:** tono, formato, vocabulario de dominio y tareas repetitivas de alta frecuencia.
- **Cuándo no:** para inyectar conocimiento que cambia — eso se resuelve con Knowledge (RAG).
- **Flujo gestionado:** dataset → entrenamiento → evaluación comparada antes de desplegar la variante.

servicio 04 / 10

SERVICIOS

IA lista para usar, sin entrenar nada.

Traducción

Texto y documentos, en tiempo real.

Language · PII

Clasificación y anonimización de datos personales.

Speech · TTS / STT

Voz sintética y transcripción.

Visión · Documentos

OCR, análisis de imagen y extracción de documentos.

Se conectan a los agentes como capacidades: un agente que traduce, anonimiza PII o habla.

servicio 05 / 10

TOOLS

Lo que el agente puede hacer, no solo decir.

- **Integradas:** Code Interpreter en sandbox, File Search, Grounding con Bing.
- **Propias:** Azure Functions, especificaciones OpenAPI y servidores MCP.
- **Mínimo privilegio:** cada herramienta amplía la superficie de ataque. Se conecta solo lo necesario.

servicio 06 / 10

KNOWLEDGE

El agente responde con tus datos, no con su memoria.

- **RAG gestionado:** Azure AI Search con índices vectoriales e híbridos sobre tus documentos.
- **Fuentes conectables:** SharePoint, blobs, webs y bases documentales como conocimiento del agente.
- **Respuestas con cita:** cada afirmación enlaza a su fuente — trazabilidad, otra vez.

servicio 07 / 10

MEMORY

Lo que el agente recuerda entre sesiones.

- **Corto plazo:** el thread — historial y estado de la conversación en curso.
- **Largo plazo:** preferencias y hechos que persisten entre sesiones, sin reentrenar el modelo.
- **Gobernada:** la memoria también es dato personal — auditable, acotada y borrrable.

servicio 08 / 10

DATA

La materia prima, conectada sin salir de tu tenant.

- **Conexiones:** Blob Storage, OneLake / Fabric, bases de datos y orígenes externos.
- **Alimenta al resto:** índices de Knowledge, datasets de Fine-Tune y de Evaluations.
- **Acceso gobernado:** identidad gestionada, redes privadas — el dato no abandona el tenant.

servicio 09 / 10

EVALUATIONS

La matriz de señales, como servicio gestionado.

- **Evaluadores:** groundedness, relevancia, coherencia, fluidez y métricas de seguridad.
- **Evaluación continua:** en CI/CD, antes de cada promoción — y red teaming automatizado.
- **Observabilidad:** trazas de cada run — qué hizo el agente, con qué herramienta y por qué.

→ **Parte 1: evidencias, no opiniones.**

servicio 10 / 10

GUARDRAILS

Los guardarraíles de la parte 1,
configurables por despliegue.

- **Filtros de contenido** en entrada y salida: violencia, odio, sexual, autolesión — por niveles.
- **Prompt Shields:** detección de jailbreak y de inyección indirecta — la amenaza n.º 1 de la parte 1.
- **Blocklists y detección** de material protegido y datos sensibles antes de que salgan.

→ **Parte 1: trata a tu agente como a un atacante.**

La teoría ya tiene plataforma.

Identidad y permisos por agente

→ Agents + Entra Agent ID

Evidencias, no opiniones

→ Evaluations + observabilidad

El agente como atacante

→ Guardrails + Tools con mínimo privilegio

Contexto limpio y memoria acotada

→ Knowledge + Memory + Data

→ Y ahora sí: ¿le darías la llave?

GRACIAS

Gobierno, gestión y monitorización de agentes de IA

Madrid Innovation · MIL · AINetwork

AI Summer Camp