GUÍA PRÁCTICA

Edición de octubre de 2026

Del piloto a producción en Azure

Checklist y mapa para llevar tu app de IA generativa a producción en una landing zone corporativa

DENTRO

- **12** dimensiones del mapa de brecha
- **20** casillas de go-live para marcar
- **4** reglas para el camino de tickets
- **7** capas de validación y quién firma

Carlos I. Pérez-Sechi
18 años en tecnología · Doctor en Ciencia de Datos (UCM)

EL PATRÓN

Por qué se atascan los pilotos

22%

de las organizaciones ha escalado la IA en varias unidades de negocio.

Gartner, 1 de septiembre de 2026

25%

de las organizaciones ha pasado a producción el 40% o más de sus pilotos.

Deloitte, State of AI in the Enterprise, 3 de marzo de 2026

La IA acelera el POC. No acelera la parte empresarial.

1. **El prototipo cuesta días.** Con asistentes de código o vibe coding, una sola persona monta en un sandbox un piloto que funciona y gusta, con endpoints públicos y claves en un .env.

2. **La fecha del anuncio llega antes que el plan.** Dirección compromete el lanzamiento sin entornos, sin SLA y sin saber quién mantendrá el código.

3. **La industrialización es un segundo proyecto.** Identidad, red, secretos, IaC, aprobaciones: nadie lo había presupuestado y, a veces, la plataforma de destino aún no existe.

LA REGLA DE LAS TRES FASES

Las 3 fases y qué prometer en cada una

	1 Validar	2 MVP	3 Industrializar
USUARIOS	Un grupo pequeño y elegido	Usuarios reales de un área	Toda la organización
PROMESA	Demostrar valor con casos reales	Aprender de usuarios reales y mejorar	Operar a escala, con soporte
SLA	✕ **Ninguno**	✕ **No se promete**	✓ **Firmado, con RTO y RPO**
SEGURIDAD	Datos acotados y ninguna clave en el código	Ya en la landing zone, con identidad gestionada	Aprobación de ciberseguridad y defensa en profundidad

MVP no significa SLA. El MVP sirve para aprender de usuarios reales. El SLA llega con la industrialización.

Traza una línea roja: necesitamos esto y no otra cosa.

Pide lo mínimo que te desbloquea: un resource group con rol Contributor.

DIAGNÓSTICO

Mapa de brecha: 12 dimensiones

	DIMENSIÓN	PREGUNTA CLAVE	✓	–	×
1	**Identidad**	¿Hay alguna API key o connection string?	○	○	○
2	**Red**	¿Qué está expuesto a internet y qué salida está permitida?	○	○	○
3	**Secretos**	¿Cuántos secretos hay y quién puede leerlos?	○	○	○
4	**IaC**	¿Se puede recrear el entorno desde el repositorio?	○	○	○
5	**CI/CD**	¿El pipeline despliega sin secretos, con identidad federada?	○	○	○
6	**Entornos**	¿Hay desarrollo, UAT y producción con datos y permisos separados?	○	○	○
7	**Observabilidad**	¿Quién recibe la alerta si se cae a las 3 de la mañana?	○	○	○
8	**Requisitos no funcionales**	¿Están definidos backups, SLA, RTO, RPO y la clasificación del dato?	○	○	○
9	**Calidad medible**	¿Cuál es el score hoy?	○	○	○
10	**Coste**	¿Sabes cuánto cuesta este caso de uso y tiene cuota?	○	○	○
11	**Aprobaciones**	¿Quién firma cada capa de validación y cuándo?	○	○	○
12	**Dueño y soporte**	¿Quién mantiene el código si su autor se va?	○	○	○

✓ Resuelto – A medias × Pendiente

Cada pendiente es trabajo que tu plan aún no tiene.

GO-LIVE

Checklist de go-live (1/2): identidad, secretos y red

Identidad

- [ ] **Una Managed Identity por caso de uso**
  Nunca una identidad compartida por todos
- [ ] **RBAC solo sobre sus propios recursos**
  Asignado en Terraform, no a mano
- [ ] **Autenticación Entra en PostgreSQL y Storage**
  Sin connection strings con contraseña

Secretos

- [ ] **Secretos externos en Key Vault, con secretRef**
  Llegan como variables del contenedor, nunca en el código
- [ ] **Contrato de secretos: nombre y formato**
  Entre equipos se comparte el nombre, nunca el valor

Red y exposición

- [ ] **Private endpoints en datos y modelos**
  También en Key Vault: nada público por defecto
- [ ] **Salida a internet denegada por defecto**
  Cada endpoint externo, justificado en la arquitectura
- [ ] **Reglas de firewall por FQDN**
  Solo los destinos declarados, por nombre
- [ ] **TLS forzado en todas las conexiones**
  Incluida la base de datos
- [ ] **Recursos huérfanos eliminados**
  Ningún recurso del piloto olvidado con acceso público

GO-LIVE

Checklist de go-live (2/2): operación

Observabilidad

- [ ] **La telemetría llega a Application Insights**
  Comprobado en producción, no supuesto
- [ ] **Alertas configuradas en Azure Monitor**
  Errores, latencia y consumo de tokens
- [ ] **Una persona de guardia recibe la alerta**
  Con nombre y horario, también a las 3 de la mañana

Continuidad

- [ ] **Backups configurados y restaurados**
  Un backup que nunca has restaurado es una hipótesis
- [ ] **SLA acordado con negocio**
  Llega con la industrialización, no con el MVP
- [ ] **RTO y RPO definidos**
  Cuánto puedes estar caído y cuántos datos perder

Entrega

- [ ] **Entornos de desarrollo, UAT y producción**
  Con datos y permisos separados
- [ ] **IaC en Terraform**
  El entorno se recrea desde el repositorio
- [ ] **CI/CD con identidad federada**
  El pipeline despliega sin secretos

Dueño y soporte

- [ ] **Dueño y nivel de soporte declarados**
  Antes de desplegar, no tras el primer incidente

LANDING ZONE

Caso ilustrativo. Northwind Traders es una empresa ficticia

El camino oculto: tickets

Una hora de código. Semanas de tickets.

El ticket que bloqueaba a todos

Aprovisionamiento de recursos

× Eliminado

2 Sustituido por una PR

Burocráticos

Los abre y persigue negocio

- Alta en inventario de aplicaciones
- App registration en Entra ID
- 3 Certificado interno — ⇉ En paralelo
- Certificado externo — ⇉ En paralelo
- 4 Registro DNS — ⇉ En paralelo
- Firewall por FQDN — ⇉ En paralelo
- Ruta en App Gateway — ⇉ En paralelo

1

Técnicos

Pull request de Terraform, sin ticket

- Resource groups por entorno
- Managed Identities
- RBAC
- Namespace en Container Registry
- Key Vault y referencias a secretos
- PostgreSQL con autenticación Entra
- Entorno de Container Apps

1. **Separa burocracia e ingeniería.** Los tickets burocráticos los abre y persigue negocio; lo técnico se resuelve en código.
2. **Sustituye el ticket que bloquea por una PR.** Infraestructura revisa una pull request de Terraform en vez de atender un ticket.
3. **Paraleliza lo manual.** Certificados, DNS, firewall y gateway se lanzan a la vez, no en cadena.
4. **Despliega antes de pedir el DNS.** La URL solo existe tras el primer despliegue.

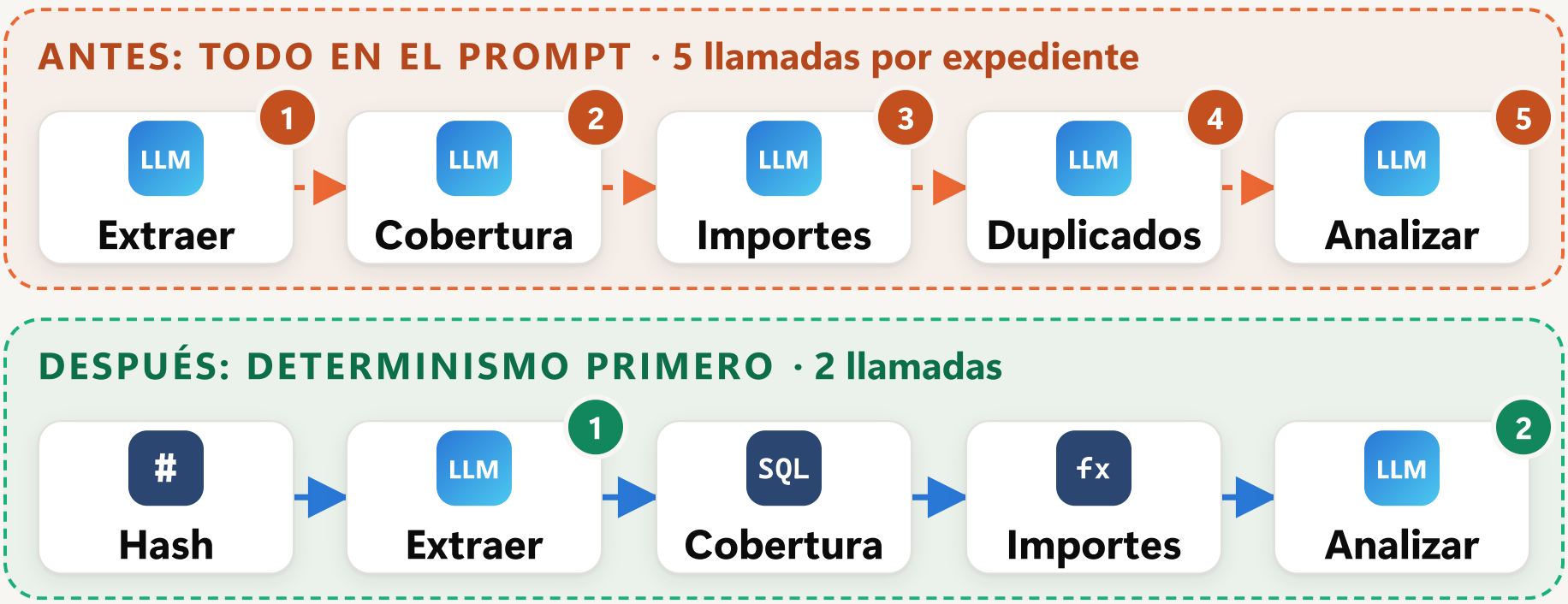
Carlos I. Pérez-Sechi
Guía «Del piloto a producción en Azure»

FINOPS

Caso ilustrativo. Contoso Seguros es una empresa ficticia

Coste por diseño

Si una decisión es determinista, **no se la pidas a un LLM.**



1 **Determinismo primero**
Lo que resuelve una consulta o un cálculo sale del prompt.

2 **Tres palancas**
Número de llamadas, tamaño y esfuerzo de cada una.

3 **Modelo por complejidad**
Ligero para extraer, razonador para analizar. Criticidad no es complejidad.

4 **Cuenta de volumen**
Hazla antes de optimizar. No degrades un modelo validado por poco.

5 **Hash antes de inferir**
Si el fichero ya se evaluó, devuelve el resultado anterior.

6 **AI gateway con cuotas**
Azure API Management delante de los modelos: cuota y coste por caso de uso.

Carlos I. Pérez-Sechi
Guía «Del piloto a producción en Azure»

CALIDAD

Calidad medible

Si me piden una IA fiable al 100%,
primero pregunto: **¿cuál es el score hoy?**

1. **Qué es correcto** — Firmado por negocio
2. **Golden dataset** — Casos y resultado esperado
3. **Tests en cada PR** — Alucinación y regresión
4. **Juez nocturno** — Un modelo revisa el día
5. **Score por versión** — Prompts como código

`golden-dataset.xlsx` — Plantilla para copiar

Caso	Entrada	Esperado	Variación	Pass/Fail
C-001				☐ Pass ☐ Fail
C-002				☐ Pass ☐ Fail
C-003				☐ Pass ☐ Fail
C-004				☐ Pass ☐ Fail

Go

El score supera el umbral acordado (por ejemplo, 95%) y ningún caso crítico falla.

No-go

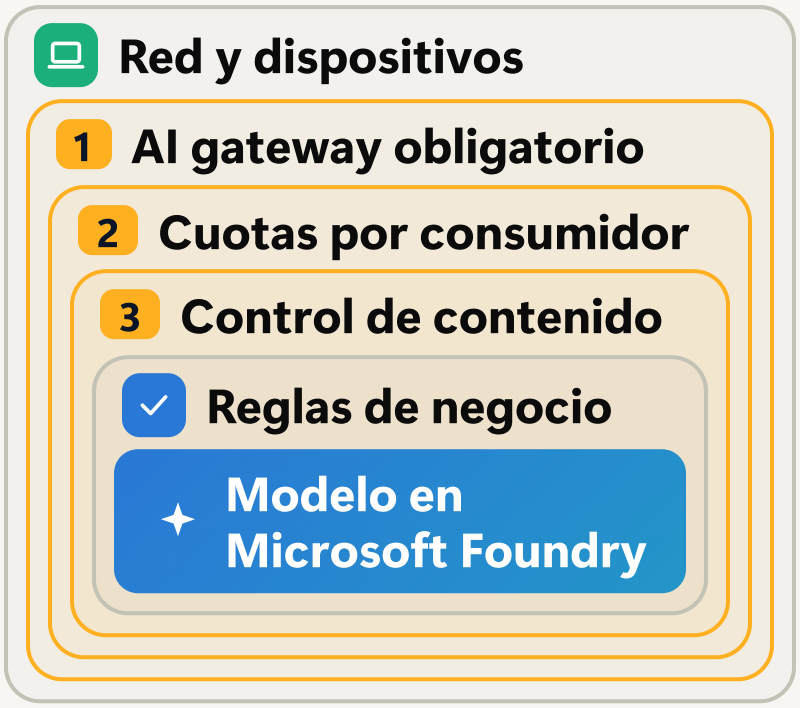
Falla un caso crítico o el score baja respecto a la versión anterior.

El umbral se fija con negocio antes de medir, no después de ver el resultado.

APROBACIONES

Ciberseguridad y comité: qué llevar

Defensa en profundidad



Qué llevar a ciberseguridad

- Diagrama de contexto y contenedores
- Clasificación del dato
- Certificación de IA responsable
- Defensa en profundidad para los LLM
- Salidas a internet declaradas

Un ADR no es una arquitectura.
El ADR dice qué servicios usar. La arquitectura dice cómo encajan.

CAPA	QUIÉN FIRMA (ROL)	QUÉ LLEVAR
Alto nivel	Arquitectura empresarial	Diagrama de contexto
Integración	Autoridad de diseño	Diagrama plataforma con plataforma
Seguridad	Ciberseguridad	Clasificación del dato y defensa en profundidad
Funcional	Sponsor de negocio	Criterios de aceptación
Operaciones	Operaciones e infraestructura	Monitorización, soporte y runbook
Calidad	QA o calidad	Golden dataset y umbral go/no-go
Registro del caso de IA	Gobierno de IA o riesgo	Uso previsto y perfil de riesgo

Ejemplo: los nombres de las capas y quién firma cambian en cada empresa. Mapéalo el primer día.

AI ACT

AI Act, artículo 50, por rol

No soy abogado: traduzco requisitos a arquitectura y controles. La interpretación legal es de tu asesor.

2 ago 2026 — Ya aplica

Artículo 50 aplicable y **no aplazado**.

2 dic 2026 — Proveedores

Límite para el marcado legible por máquina (50.2) de sistemas generativos comercializados antes del 2 ago 2026.

2 dic 2027 Alto riesgo, Anexo III — Aplazado

2 ago 2028 Alto riesgo, Anexo I — Aplazado

Reglamento (UE) 2026/1744, en vigor desde el 27 de julio de 2026.

Proveedor

Art. 50.1 y 50.2. Si construyes tu propio agente, también lo eres.

Responsable del despliegue

Art. 50.3 y 50.4. Ya aplica, sin periodo de gracia.

CÓMO LO CONVIERTO EN CONTROLES

1 Control que cubre el 100%

Primero, un aviso de etiquetado que aparece siempre.

→

2 Detección automática

Después, y medida: un detector que falla en silencio da falsa seguridad.

¿TIENES UN PILOTO CONCRETO?

Siguiente paso

Diagnóstico piloto a producción

2 semanas

Mido la distancia real de tu piloto a producción en tu landing zone de Azure y trazo el plan más corto y seguro.

- ✓ Mapa de brecha
- ✓ Camino con tickets
- ✓ Plan por fases y fechas
- ✓ Arquitectura objetivo en Azure
- ✓ Checklist de go-live
- ✓ Revisión rápida de coste

Empieza por una llamada de 30 minutos

en mi agenda o escríbeme por mensaje privado.

Carlos I. Pérez-Sechi

perez-sechi

18 años en tecnología · Doctor en Ciencia de Datos (UCM)

- Un asistente basado en IA para miles de usuarios, en Microsoft Foundry.
- Un chatbot RAG para más de 40.000 empleados, con evaluación continua.

01 Optimizamos flujos de trabajo

02 Hacemos realidad conceptos

03 Escalamos el despliegue

perez-sechi.com · Sígueme en LinkedIn (/in/perez-sechi) y TikTok (@perezsechi)