

Informe de Arquitectura Técnica: Sistema de Orquestación AI-BPO

1. Resumen Ejecutivo

Se propone una solución de backend basada en un microservicio asíncrono diseñado para automatizar el procesamiento de solicitudes de clientes. La arquitectura utiliza un enfoque **Data-Driven**, donde el comportamiento del Agente de IA se adapta dinámicamente según la configuración de cada empresa almacenada en una base de datos NoSQL, garantizando escalabilidad y mantenibilidad sin necesidad de despliegues constantes de código.

2. Componentes y Responsabilidades

A. Capa de Aplicación (FastAPI + LangChain)

- **FastAPI:** Actúa como el núcleo del microservicio, gestionando solicitudes HTTP de forma asíncrona para maximizar el bajo alta concurrencia, además genera automáticamente la documentación utilizando swagger.
- **LangChain & Google Gemini:** El orquestador de IA utiliza el modelo Gemini-1.5-Flash para clasificar solicitudes, asignar prioridades y generar respuestas estructuradas. Se implementa **Structured Output** mediante modelos de Pydantic para asegurar que la salida del LLM cumpla siempre con el esquema JSON requerido, y schemas para garantizar que se distingan las variables de resultado y respuestas. En este caso no fue necesario utilizar InMemorySaver para la memoria del agente.

B. Capa de Datos (MongoDB / DocumentDB)

- **Colección empresas:** Almacena metadatos críticos, categorías permitidas, reglas de prioridad y rutas de delegación.
- **Colección casos:** Repositorio de auditoría donde se persiste cada interacción, facilitando la trazabilidad y la detección de solicitudes duplicadas.

C. Herramientas de Integración (Tools)

- Módulos desacoplados que permiten al agente interactuar con servicios externos (ej. consulta de API para Mensajería del Valle). Esta capa permite que las capacidades del agente crezcan de forma modular, en este ejercicio solo fue necesario crear una tool.

3. Desacoplamiento y Escalabilidad

Estrategia de Desacoplamiento

- **Lógica de Negocio vs. Código:** Las reglas de cada empresa no residen en el código (hardcodeado), sino que se inyectan en el prompt del sistema desde la base de datos. Esto permite que el componente de IA sea agnóstico al cliente procesado.
- **Modularidad de Tools:** Las herramientas de integración residen en una capa independiente, lo que permite testear y actualizar conectores externos sin afectar el núcleo del orquestador.

Soporte al Crecimiento

- **Crecimiento en Clientes:** La infraestructura permite integrar una nueva compañía simplemente añadiendo un documento de configuración en MongoDB.
- **Crecimiento en Categorías:** El sistema soporta N categorías por empresa. Al estar tipadas como una lista en el JSON de configuración, el agente ajusta su espectro de clasificación en tiempo real.
- **Escalabilidad Horizontal:** Al ser un servicio Stateless, el backend puede replicarse en múltiples instancias sin conflictos, ya que toda la persistencia reside en la base de datos distribuida.

4. Propuesta de Despliegue Cloud (AWS)

Para garantizar un entorno de producción de misión crítica, se propone el despliegue en la región **us-east-1** utilizando instancias de cómputo elástico:

Cómputo y Balanceo

- **Amazon EC2 & Auto Scaling Group (ASG):** La aplicación se despliega en instancias EC2 distribuidas en múltiples Zonas de Disponibilidad. El ASG utiliza una Amazon Machine Image (AMI) preconfigurada para lanzar nuevas instancias automáticamente ante incrementos en la demanda (CPU/RAM), garantizando que el servicio escale según el tráfico.
- **Application Load Balancer (ALB):** Punto único de entrada que distribuye el tráfico de forma balanceada entre las instancias activas del ASG, realizando health checks constantes para asegurar que solo las instancias saludables reciban peticiones.

Red y Seguridad

- **AWS WAF (Web Application Firewall):** Capa de seguridad perimetral para mitigar ataques como inyecciones o bots antes de llegar al balanceador.
- **VPC con Subredes Privadas:** Las instancias EC2 de la aplicación y la base de datos residen en subredes privadas, protegidas de la internet pública.
- **NAT Gateways:** Redundantes por Zona de Disponibilidad, permiten que las instancias EC2 consulten de forma segura la API de Google Gemini mientras permanecen en una red privada.

Persistencia

- **Amazon DocumentDB (Cluster Multi-AZ):** Base de datos gestionada y compatible con MongoDB. Se configura con un cluster de alta disponibilidad (instancia primaria y réplicas) distribuido en diferentes zonas para asegurar resiliencia ante fallos regionales.

5. Justificación de Seguridad

El diseño implementa el principio de **Defensa en Profundidad**:

1. **Validación de Esquemas:** Uso de Pydantic para prevenir el procesamiento de datos malformados.
2. **Aislamiento de Red:** Security Groups que restringen el tráfico entrante a la base de datos únicamente desde el Security Group de las instancias EC2.
3. **Gestión de Secretos:** Uso de variables de entorno protegidas para el manejo de API Keys y credenciales de acceso a las bases de datos.