Informe de Arquitectura: Conversational E-commerce Backend

1. Visión General del Sistema

El sistema está diseñado bajo un paradigma de **Arquitectura Híbrida**, separando estrictamente la orquestación conversacional de la lógica de negocio. El diseño prioriza la **determinación y la fiabilidad** sobre la generación estocástica, utilizando el procesamiento de lenguaje natural (LLM) como una capa de mejora opcional y no como una dependencia crítica.

2. Organización por Capas y Componentes

A. Exposición y Entrada (API Layer)

- **Archivo:** app/main.py (FastAPI)
- **Responsabilidad:** Actúa como una "capa delgada" que expone endpoints estratégicos (/start, /chat, /checkout/submit).
- **Diseño:** No contiene lógica de negocio. Su función es adaptar las peticiones entrantes al motor de ejecución y transformar el estado interno en un ui_payload simplificado para el consumo del frontend.

B. Motor Conversacional y Gestión de Estado

- **ChatEngine (app/engine/service.py):** Es el controlador central. Gestiona el ciclo de vida de la sesión, la persistencia temporal de los datos y la ejecución del grafo.
- **Estado de la Conversación (app/engine/state.py):** Funciona como la **fuente única de verdad**. Define el contrato común (carrito, datos de envío, flags de UI y control de flujo) sobre el cual operan todos los nodos de LangGraph.
- **Post-procesamiento de UX (app/engine/response.py):** Centraliza las reglas de presentación (ej. limpieza de formato en checkout), independizando la lógica conversacional de la política de respuesta al usuario.

C. Orquestación con LangGraph

- **Grafo de Estados (app/graph/builder.py):** Define la topología de la conversación mediante nodos y aristas condicionales.
- **Router Determinista (routing/selectors.py):** Un selector final que previene bucles infinitos y asegura que el flujo transicione correctamente basándose en el estado actual, garantizando una experiencia de usuario predecible.

D. Inteligencia y Procesamiento (Routing por Reglas)

- **Cerebro Determinista (app/graph/routing/rules/*):** El sistema utiliza un conjunto de reglas de negocio que analizan el mensaje del usuario antes de cualquier procesamiento externo. Esto permite gestionar salidas, inicios de checkout y ayuda sin latencia ni costes de API.
- **Parsers Heurísticos (app/utils/*):** Implementan lógica de extracción de slots (cantidades, IDs de producto) de forma robusta, permitiendo que el sistema funcione al 100% incluso en entornos sin conexión a modelos de lenguaje.

E. Capa de Dominio y Tools

- **Services (app/services/*):** Centralizan la lógica pura de e-commerce (cálculo de totales, filtrado de catálogo por aroma/precio/público).
- **Tools (app/tools/*):** Operaciones atómicas que mutan el estado. Son funciones puras y testeables que actúan como puente entre el grafo conversacional y los servicios de dominio.

3. Estrategia de IA: LLM Encapsulado

El sistema implementa una integración de **OpenAI** (app/llm/*) diseñada como una estrategia de "mejora progresiva":

- **Fallback Automático:** Si el LLM está desactivado (LLM_ROUTER_ENABLED=false) o su confianza es inferior al umbral (LLM_MIN_CONFIDENCE), el sistema utiliza automáticamente los parsers deterministas.
- **Aislamiento:** El LLM solo se encarga de la clasificación de intención y extracción de entidades; nunca toma decisiones directas sobre la ejecución de pagos o inventario, eliminando riesgos de alucinación.

4. UI y Frontend

- **Gradio UI:** Se incluye una capa de interfaz (gradio_chat.py) independiente de la lógica del backend, facilitando las pruebas de usuario y la validación de flujos de checkout y formularios de contacto.
- **Centralización de Copy:** Todos los mensajes están centralizados en app/ux/copy.py, permitiendo el soporte multi-idioma y garantizando una voz de marca consistente en todos los canales.

5. Conclusión de Diseño

Esta arquitectura permite que el proyecto sea:

1. **Económico:** Minimiza llamadas a APIs externas.
2. **Mantenible:** La lógica de "qué decir" está separada de "cómo vender".
3. **Escalable:** Es sencillo añadir nuevas reglas o nodos al grafo sin romper la lógica de carrito o catálogo existente.