

Escenario: arquitectura para un sistema de reservas de vuelos

Prompt 1 — Modelado de dominios

Tengo que diseñar una arquitectura distribuida para un sistema de reservas de vuelos. ¿Qué módulos o bounded contexts debería considerar? Quiero que me ayudes a definirlos y priorizarlos.

Prompt 2 — Diseño de eventos

Proponme los eventos clave del dominio. Usa formato verbo-pasado como 'flight.scheduled' o 'booking.confirmed'. Dame también el servicio que los emite y qué otros los consumen.

Prompt 3 — Contratos de eventos

Genera la estructura JSON para el evento 'booking.confirmed', incluyendo los campos recomendados y buenas prácticas para trazabilidad.

Prompt 4 — Topología de servicios

Basado en los eventos anteriores, propón una topología distribuida de servicios. Quiero saber qué servicios existen, cómo se comunican (sincronía vs eventos) y qué base de datos usa cada uno.

Prompt 5 — Validación de decisiones arquitectónicas

¿Crees que el servicio de pagos debe emitir su propio evento al confirmarse el pago o debe simplemente responder al frontend? Evalúa ventajas y riesgos de ambas opciones.

Prompt para diagrama Mermaid

Genera un diagrama en formato Mermaid.js para el siguiente flujo: - OrderService crea una orden y emite 'order.created'. - PaymentService escucha ese evento y responde con 'payment.succeeded'. - InventoryService reacciona a 'payment.succeeded' y descuenta stock.

Resultado esperado:

```
graph TD
  OrderService -->|order.created| EventBus
  EventBus --> PaymentService
  PaymentService -->|payment.succeeded| EventBus
  EventBus --> InventoryService
```