

Escenario propuesto: Plataforma de pedidos de comida a domicilio

Similar a Glovo, Uber Eats o Rappi.

Paso 1: Definición del flujo de negocio (nivel alto)

Usuario:

1. Realiza un pedido desde la app
2. El sistema valida los datos y disponibilidad
3. El pago se procesa
4. Se asigna un repartidor
5. El restaurante recibe el pedido
6. El pedido se entrega

Paso 2: Usamos la IA para ayudarnos a identificar módulos y eventos

Prompt para ChatGPT:

Estoy diseñando una arquitectura distribuida para una plataforma de pedidos de comida. ¿Qué módulos o servicios sugerirías? ¿Qué eventos de dominio deberían emitirse y por quién?

Respuesta esperada (resumida):

Servicios:

- OrderService
- PaymentService
- DeliveryService
- RestaurantService
- NotificationService

Eventos:

- order.placed
- payment.succeeded
- delivery.assigned
- restaurant.notified

- order.delivered

Paso 3: Definimos eventos clave (IA + sentido común)

```
JSON
// order.placed
{
  orderId: "abc123",
  userId: "user42",
  items: [{ id: "burger01", qty: 2 }],
  total: 18.5,
  createdAt: "2025-09-10T12:00:00Z"
}

// payment.succeeded
{
  orderId: "abc123",
  paymentId: "pmt001",
  method: "card",
  paidAt: "2025-09-10T12:01:10Z"
}

// delivery.assigned
{
  orderId: "abc123",
  driverId: "drv88",
  eta: "15 min"
}
```

Paso 4: Diseñamos el flujo completo de eventos

```
None
graph TD
    A[User] --> B[OrderService]
    B -->|order.placed| EB(EventBus)
    EB --> C[PaymentService]
    C -->|payment.succeeded| EB
    EB --> D[DeliveryService]
    EB --> E[RestaurantService]
    D -->|delivery.assigned| EB
    EB --> F[NotificationService]
```

💡 Puedes usar herramientas como Mermaid.js, IA generativa o diagramas de secuencia para visualizarlo.

Paso 5: Documentamos contratos y consumidores

Ejemplo:

Evento	Publica	Consume
order.placed	OrderService	PaymentService, RestaurantService
payment.succeeded	PaymentService	DeliveryService, NotificationService
delivery.assigned	DeliveryService	NotificationService

Paso 6: IA como copiloto de decisión

Prompt para IA:

¿Crees que debería usar comunicación síncrona entre OrderService y PaymentService, o emitir un evento 'order.placed'? Evalúa pros y contras.

✅ Esto fuerza a la IA a razonar contigo sobre trade-offs arquitectónicos.

Ejercicio práctico (para el alumno)

Repite este mismo proceso con otro caso:

- Plataforma de reservas de hoteles
- Sistema de logística y paquetería
- Plataforma de educación en línea con inscripciones y pagos