

# Ejercicios Técnicos – Ingeniería de Prompts

## Ejercicio 1 – Normalización y Validación de Órdenes (Chain of Thought)

### Problemática

Se recibe un JSON con órdenes que contienen datos inconsistentes: tipos incorrectos, valores negativos y formatos irregulares. Se debe normalizar la estructura, validar reglas de negocio y generar un resumen calculado únicamente con órdenes válidas.

### Objetivo del ejercicio

Construir un prompt que obligue al modelo a:

- 1) Analizar reglas de validación antes de calcular totales.
- 2) Separar órdenes válidas de inválidas.
- 3) Generar un JSON estructurado y verificable.

### Entrada

```
{  
  "orders": [  
    {  
      "orderId": 101,  
      "items": [  
        { "sku": " abc-1 ", "qty": "2", "unitPrice": "10.50" },  
        { "sku": "xyz-9", "qty": 1, "unitPrice": 5 }  
      ]  
    },  
    {  
      "orderId": "102A",  
      "items": [  
        { "sku": "abc-1", "qty": 0, "unitPrice": 10.5 }  
      ]  
    },  
    {  
      "orderId": 103,  
      "items": [  
        { "sku": " LMN-2", "qty": -3, "unitPrice": 7.25 },  
        { "sku": "pqr-7", "qty": 4, "unitPrice": "0" }  
      ]  
    }  
  ]  
}
```

### Criterios verificables

- orderId convertido a string.
- sku normalizado (trim + uppercase).
- qty convertido a entero y  $\geq 1$ .
- unitPrice convertido a number  $\geq 0$ .
- Solo órdenes válidas deben participar en el cálculo de totales.
- grossTotal debe ser exactamente 26.00.
- skuCounts debe ser {"ABC-1": 2, "XYZ-9": 1}.

### Salida esperada

```
"summary": {  
    "validOrdersCount": 1,  
    "invalidOrdersCount": 2,  
    "grossTotal": "26.00",  
    "skuCounts": {  
        "ABC-1": 2,  
        "XYZ-9": 1  
    }  
}
```

## Ejercicio 2 – Transformación Determinística de Logs (Few-Shot)

### Problemática

Se deben transformar registros de log a un formato de auditoría. El modelo debe seguir reglas estrictas de normalización y mantener el orden original.

### Objetivo del ejercicio

Construir un prompt few-shot que enseñe al modelo el patrón correcto de transformación y genere una salida exactamente verificable.

### Entrada

```
{  
  "logs": [  
    { "ts": "2026-02-16T10:20:30Z", "user": " Alice ", "event": "LOGIN_OK" },  
    { "ts": "2026-02-16T10:25:00Z", "user": "Bob", "event": "FILE_UPLOAD" },  
    { "ts": "2026-02-16T11:00:00Z", "user": "ALICE", "event": "LOGIN_FAIL" }  
  ]  
}
```

### Criterios verificables

- userId debe estar en lowercase y sin espacios.
- date debe extraerse en formato YYYY-MM-DD.
- actionCode debe mapearse correctamente.
- El orden debe mantenerse.
- La salida debe ser únicamente JSON.

### Salida esperada

```
{  
  "auditRecords": [  
    { "date": "2026-02-16", "userId": "alice", "actionCode": "A01" },  
    { "date": "2026-02-16", "userId": "bob", "actionCode": "A10" },  
    { "date": "2026-02-16", "userId": "alice", "actionCode": "A02" }  
  ]  
}
```

## Ejercicio 3 – Deduplicación de Contactos (Tree of Thoughts)

### Problemática

Se requiere identificar contactos duplicados con variaciones en nombre, email y teléfono. El modelo debe analizar múltiples estrategias antes de decidir cuál aplicar.

### Objetivo del ejercicio

Forzar al modelo a:

- 1) Generar tres estrategias distintas.
- 2) Compararlas.
- 3) Elegir una.
- 4) Producir clusters finales verificables.

### Entrada

```
{  
  "contacts": [  
    { "id": "c1", "name": "Ana López", "email": "ana.lopez@gmail.com", "phone": "+52 81 1234  
5678" },  
    { "id": "c2", "name": "ANA LOPEZ", "email": "ana.lopez+promo@gmail.com ", "phone":  
"8112345678" },  
    { "id": "c3", "name": "Ana L.", "email": "ana.lopez@gmail.com", "phone": "81 1234 5678" },  
    { "id": "c4", "name": "Luis Perez", "email": "luisp@empresa.com", "phone": "+52 55 0000  
0000" },  
    { "id": "c5", "name": "Luis Pérez", "email": "luisp@empresa.com ", "phone": "5500000000"  
}  
  ]  
}
```

### Criterios verificables

- Deben generarse 3 estrategias realmente distintas.
- Debe existir una recomendación justificada.
- Los clusters correctos son:  
k1: c1, c2, c3  
k2: c4, c5

### Salida esperada (clusters finales)

```
{  
  "clusters": [  
    { "clusterId": "k1", "contactIds": ["c1", "c2", "c3"] },  
    { "clusterId": "k2", "contactIds": ["c4", "c5"] }  
  ]  
}
```