# Desafío Técnico – IA Engineer

## Caso: Sistema Multi-Agente para Detección de Fraude Ambiguo en Transacciones Financieras

### 1. Contexto de Negocio

Las instituciones financieras enfrentan el reto de detectar fraudes en transacciones que presentan señales ambiguas: montos inusuales, horarios no habituales, dispositivos desconocidos o patrones de comportamiento atípicos.

El objetivo de este reto es construir una **web App (Backend + Frontend)** que implemente un **Sistema Multi-Agente de Detección de Fraude**, capaz de:

- Analizar transacciones en tiempo real.
- Evaluar señales internas (comportamiento, monto, horario, país, dispositivo).
- Consultar políticas internas mediante RAG (base vectorial).
- Buscar inteligencia externa sobre amenazas recientes (búsqueda web gobernada).
- Orquestar agentes para tomar decisiones rápidas y trazables.
- Incluir revisión humana (Human-in-the-loop) cuando la decisión lo requiera.

### 2. Datos sintéticos

Se proporcionan datos sintéticos, sin datos personales reales:

- **Transactions**

| transaction_id | customer_id | amount | currency | country | channel | device_id | timestamp | merchant_id |
|---|---|---|---|---|---|---|---|---|
| **T-1001** | CU-001 | 1800.00 | PEN | PE | web | D-01 | 2025-12-17T03:15:00 | M-001 |
| **T-1002** | CU-002 | 9500.00 | PEN | PE | mobile | D-02 | 2025-12-17T23:45:00 | M-002 |

- **customer_behavior**

| customer_id | usual_amount_avg | usual_hours | usual_countries | usual_devices |
|---|---|---|---|---|
| **CU-001** | 500.00 | 08-20 | PE | D-01 |
| **CU-002** | 1200.00 | 09-22 | PE | D-02 |

- **fraud_policies:**


```
[
  {
    "policy_id": "FP-01",
    "rule": "Monto > 3x promedio habitual y horario fuera de rango → CHALLENGE",
    "version": "2025.1"
  },
  {
    "policy_id": "FP-02",
    "rule": "Transacción internacional y dispositivo nuevo → ESCALATE_TO_HUMAN",
    "version": "2025.1"
  }
]
```


## 3. Requerimientos Técnicos

Construir una **web App (Backend + Frontend)** que:

- Procese los archivos entregados y consolide la información por transacción y cliente.
- Analice señales clave: monto, horario, país, dispositivo, patrón de comportamiento.
- Genere al menos cuatro escenarios por transacción:
  - **APPROVE**: transacción legítima.
  - **CHALLENGE**: requiere validación adicional.
  - **BLOCK**: bloqueo por sospecha de fraude.
  - **ESCALATE_TO_HUMAN**: revisión humana obligatoria.
- Utilice un **equipo multi-agente** orquestado:
  - **Transaction Context Agent**: analiza señales internas.
  - **Behavioral Pattern Agent**: compara con el historial del cliente.
  - **Internal Policy RAG Agent**: consultas políticas internas vía base vectorial.
  - **External Threat Intel Agent**: busca amenazas recientes en la web (gobernada).
  - **Evidence Aggregation Agent**: reúne todas las evidencias.
  - **Debate Agents**: Pro-Fraud vs Pro-Customer.

    - **Decision Arbiter Agent**: toma la decisión.
    - **Explainability Agent**: genera explicación para cliente y auditoría.
- Implemente **Human-in-the-loop** con cola de casos y audit trail.
- Calcule la **confianza** y registre señales, evidencias internas (RAG) y externas (web).
- Genere un **informe explicativo en lenguaje natural** por transacción, usando IA Generativa.
- Lenguaje de programación preferente: Python.
- Despliegue en la nube (preferentemente **Azure**, se acepta **AWS**).
- Utilice frameworks de agentes/orquestación (Ejemplos: Azure AI Agent Framework, LangChain, Agent SDK, Bedrock Agents, etc.).
- Si falta información, asuma bajo su mejor criterio.

## 4. Criterios de Evaluación

| Criterio | Puntaje |
|---|---|
| Diseño de arquitectura multiagente y orquestación | 25 pts |
| - Claridad en la definición de roles de agentes y su interacción | |
| - Uso de un framework de orquestación (ej. Azure AI Agent Framework, Agent SDK, LangChain, Bedrock Agents, etc.) | |
| Implementación del flujo multiagente | 20 pts |
| - Correcta secuencia y colaboración entre agentes | |
| - Manejo de handoffs u otro patrón y agregación de evidencias | |
| Integración de fuentes internas (RAG) y externas (web search) | 15 pts |
| - Uso efectivo de RAG para políticas internas | |
| - Consulta de inteligencia externa relevante y gobernada | |
| Trazabilidad y auditabilidad del proceso de decisión | 15 pts |
| - Registro de señales, evidencias, rutas de agentes y decisiones | |
| - Generación de explicaciones claras para cliente y auditoría | |
| Despliegue en la nube y buenas prácticas DevOps | 10 pts |
| - Infraestructura como código, CI/CD, seguridad de secretos | |
| - Uso de servicios gestionados (Azure/AWS) y justificación técnica | |
| Documentación, presentación y claridad del README | 10 pts |
| - Instrucciones claras de despliegue y uso | |
| - Ejemplos de uso, evidencia de funcionamiento y trazabilidad | |
| Creatividad, valor agregado y mejoras sobre lo solicitado | 5 pts |
| - Propuestas innovadoras, visualizaciones, mejoras UX, etc. | |

## 5. Entregables

- Repositorio en GitHub con el código fuente y README. enviar al siguiente correo: enriqueinca@bcp.com.pe. Plazo de entrega 48 horas después de recibir el desafío.
- Instrucciones de ejecución local o enlace a demo desplegada.
- Informe generado por IA para al menos 4 transacciones (cada tipo de decisión).
- Evidencia de trazabilidad y audit trail.
- (Opcional) Video corto explicando la solución (máx. 5 min).

### Ejemplo de Flujo Multi-Agente

1. **Llega una transacción**: T-1002, monto 9500 PEN, horario 23:45, dispositivo D-02.
2. **Transaction Context Agent** detecta monto y horario fuera de lo habitual.
3. **Behavioral Pattern Agent** confirma que el cliente suele operar con montos menores y en otro horario.
4. **Internal Policy RAG Agent** recupera la política FP-01: "Monto > 3x promedio y horario fuera de rango → CHALLENGE".
5. **External Threat Intel Agent** encuentra alerta reciente de fraude en ese merchant vía web search gobernada.
6. **Evidence Aggregation Agent** reúne señales y evidencias.
7. **Debate Agents** discuten: Pro-Fraud (sospecha) vs Pro-Customer (historial limpio).
8. **Decision Arbiter Agent** decide: CHALLENGE.
9. **Explainability Agent** genera explicación para el cliente y auditoría, citando reglas y fuentes.
10. Si la confianza es baja o hay señales contradictorias, **ESCALATE_TO_HUMAN** y se registra en la cola HITL.

- **Formato de Respuesta por Evaluación:**

```
{
  "decision": "CHALLENGE",
  "confidence": 0.65,
  "signals": ["Monto fuera de rango", "Horario no habitual", "Alerta externa"],
  "citations_internal": [
    { "policy_id": "FP-01", "chunk_id": "1", "version": "2025.1" }
  ],
  "citations_external": [
    {
      "url": "https://<URL de referencia>",
```

```
      "summary": "Alerta de fraude reciente en el merchant"
    }
  ],
  "explanation_customer": "La transacción requiere validación adicional por monto y horario inusual.",
  "explanation_audit": "Se aplicó la política FP-01 y se detectó alerta externa. Ruta de agentes: Context → RAG → Web → Debate → Decisión."
}
```


***¡Tengo la certeza de que podrás con este desafío!***