PoC4 - Validación de Historias de Usuario con IA

# 🛠️ Diseño Técnico

## 📐 Arquitectura general

El sistema está compuesto por tres capas integradas:  
- Jira Cloud (origen del evento)  
- Azure Function (procesamiento intermedio)  
- Copilot Studio (evaluación con IA)  
  
El flujo se inicia con un webhook de Jira que activa la función en Azure. La función extrae la historia de usuario, la envía vía HTTP POST a un flujo en Power Automate que invoca a un agente IA en Copilot Studio, y luego registra el resultado de nuevo en el issue en Jira a través de su API REST.

## 📩 Webhook de Jira

- Se configura para eventos `issue\_created` y `issue\_updated`  
- URL destino: Azure Function endpoint (HTTP Trigger)  
- Payload enviado:  
 - issue.key  
 - issue.fields.description  
 - issue.fields.issuetype.name  
 - issue.fields.project.key

## ⚙️ Azure Function

- Endpoint: HTTP Trigger anónimo  
- Lenguaje: Python 3.10  
- Función:  
 1. Validar que el issue sea tipo Story  
 2. Extraer y preparar descripción  
 3. Enviar JSON al webhook de Power Automate (Copilot Studio)  
 4. Recibir `score` y `comentario`  
 5. Actualizar campo personalizado del issue con API de Jira

## 📤 Payload enviado a Copilot Studio

{  
 "historia": "Como usuario, quiero guardar mis datos para no tener que ingresarlos cada vez."  
}

## 📥 Respuesta esperada desde Copilot

{  
 "score": 85,  
 "comentario": "La historia está casi completa, pero falta el objetivo ('para')."  
}

## 🔒 Seguridad y autenticación

- Azure Function usa autenticación anónima en entorno de pruebas  
- Webhook de Power Automate configurado como HTTP con token secreto si es necesario  
- La API de Jira se invoca con token personal (PAT) gestionado por secreto en Azure

## 📋 Logs y seguimiento

- Logs en Application Insights  
- Registro de entradas recibidas y resultados enviados a Jira  
- Manejo de errores por validación, timeout o respuesta inválida del agente IA