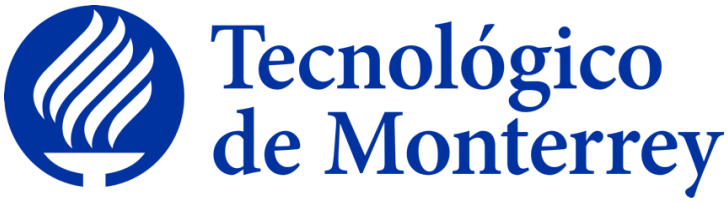
# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY




## Entrega Final

### Procesamiento de Lenguaje Natural – NLP

Presenta (Equipo 80)

| | |
|---|---|
| Arturo Jain Delgadillo | A01794992 |
| Ramiro Martin Jaramillo Romero | A01171251 |
| Owen Jáuregui Borbón | A01638122 |
| José Ashamat Jaimes Saavedra | A01736690 |

A 25 de junio de 2025

# Primera parte

## Propuesta José Ashamat:

Una de las problematicas más grandes dentro de mi trabajo, es la prolongada y compleja curva de aprendizaje de las normas, reglamentos y politicas diversas que existen y que se presentan en el onboarding. Son disitintos documentos compartidos por el área de RRHH y que sirven para describir procesos como solicitud de permisos, vacaciones o politicas de ausencia, administrativos, entre otros. Estos documentos, pocas veces son analizados a profundidad y generalmente olvidados cuando se requiere hacer un proceso.

Existe una persona en el área de recursos humanos que si bien, su funcion no es esa, se encarga de resolver todas las preguntas relacionadas a estos documentos, como "¿Si tengo 4 años en la agencia, cuantos días de vacaciones tengo derecho?", "Cómo funciona el programa cazatalento?", "Con quien me puedo acercar para revisar un tema de esta politica?", "Cuál es el proceso de solicitud de viáticos?".

Se propone realizar un agente RAG, asistente de recursos humanos de la empresa a la que pertenezco, este agente, contará con toda la información de estos documentos realizados por las distintas áreas (finanzas, RRHH, seguridad, etc), almacenados en confluence y será capaz de responder información acerca de esos documentos en un formato de tipo "chat".

Se considera una arquitectura en la nube de Google (dentro de los free tier existentes), para ello, se requiere:

Cloud Storage: Un bucket single-región para el build del front end (Vue JS) < 1GB (dentro de los free tier de Google)

Cloud Run: Dos contenedores, uno para la BBDD vectorial (que almacenará Embbedings y Documentos) y el otro para el agente RAG. El contenedor del agente RAG, no requiere muchas replicas, ni una memoria mayor a 512Mi, únicamente es una aplicación sencilla FastApi con la arquitectura en forma de grafo (LangGraph) con el framework LangChain, por lo que, el uso de este, entra dentro del free tier de Google.

Para el contenedor de la BBDD, ChromaDB (open source), se requiere un poco más de procesamiento, por lo que la memoria sería de 1Gi, este contenedor si puede generar un costo (pues excede el procesamiento del free tier de Google), sin embargo, la configuración de este deberá tener una réplica máxima de 1 y una réplica mínima de 0, por lo que si el contenedor no se usa, no tendrá instancias activas. Esto puede generar un coldstart insignificante para los propósitos de este proyecto.

LLM: Gemini 2.0 Flash, costo calculado a partir del millón de tokens de entrada y de salida. Debido a que es un agente con contexto y consciente del historial del chat, los prompts pueden llegar a ser pesados, considerando una cantidad de 4 caracteres por token, tenemos hasta 250,000 tokens para no romper el free tier de Google. Cada millón de token extra en texto, tiene un costo de .10USD.

Emmbedings model: VertexIA text-multilingual-embedding-002, costo $0.0002USD / 1,000 caracteres de entrada.

---

# Propuesta Arturo Jain Delgadillo

**AMI – Conclusión: Proyecto de Inteligencia Artificial Clínica Avanzada**

**Problemática a resolver:**
La redacción de la conclusión médica es una de las etapas más críticas y sensibles del proceso diagnóstico. Actualmente, depende totalmente del médico especialista, lo que implica una alta carga cognitiva, riesgo de errores por omisión o inconsistencias, y una amplia variabilidad en estilo y claridad. Además, para el paciente, la conclusión suele ser difícil de comprender por el uso de tecnicismos, lo que genera incertidumbre o malinterpretaciones.

**Tipo de datos y fuente:**
AMI se basa en múltiples fuentes de datos ya disponibles en la plataforma NUBIX:

- Datos demográficos del paciente.
- Tipo y motivo del estudio.
- Descripción médica realizada por el especialista.
- Estudios DICOM adjuntos.
- Texto clínico estructurado y dictado de voz (transcripción ASR).

**Departamentos involucrados:**

- **IA y Ciencia de Datos:** Diseño, entrenamiento y ajuste de los modelos.
- **Equipo Médico:** Validación clínica, alineación con estándares médicos y corrección de estilos.
- **Producto y UX:** Integración en la plataforma NUBIX y experiencia de usuario.

- **TI y Seguridad:** Protección de la información y cumplimiento de normativas (HIPAA, ISO 27001).
- **Soporte y Customer Success:** Capacitación a usuarios y retroalimentación para mejoras continuas.

**Modelos de IA que se pueden utilizar:**

- **Generación automática de texto clínico:** Modelos tipo GPT finetuneados en corpus médicos.

**Entregables del proyecto:**

1. **Módulo de conclusión automática para especialistas**, integrado en el flujo de trabajo de NUBIX.

**Tecnología y seguridad:**

- Infraestructura sobre AWS/OCI/GCP con cifrado AES.
- Cumplimiento con normativas HIPAA, ISO 27001 y FISMA.

**Tiempos y costos estimados:**

- **Fase 1 (Prototipo):** 6 semanas – Módulo funcional con generación básica de conclusiones.
- **Fase 2 (Versión clínica):** 12 a 14 semanas – Integración con flujo médico, validación con médicos.
- **Fase 3 (Despliegue completo):** 3 a 4 meses
- **Costo estimado:** $350,000 a $600,000 MXN según complejidad, infraestructura y validación clínica. El modelo de negocio puede seguir el esquema “pago por uso” como el resto de NUBIX, o integrarse como valor agregado a ciertos planes.

**Conclusión:**
AMI Conclusión representa una innovación sustancial en el flujo clínico digital. Automatiza una de las etapas más críticas del diagnóstico médico, mejora la calidad y coherencia del informe, reduce el tiempo operativo del médico y empodera al paciente con información clara y comprensible. Esta solución convierte la experiencia médica en un proceso más eficiente, humano y centrado en la comprensión. AMI no solo es una herramienta de inteligencia artificial, sino una extensión del compromiso de NUBIX con la excelencia médica, la accesibilidad del conocimiento y la transformación del sector salud.

# Propuesta Owen Jáuregui

**Amazon - Predicción y prevención de devoluciones**

**Problemática que trata de resolver:**

Amazon enfrenta millones de devoluciones cada año, lo que representa una pérdida significativa en términos logísticos, operativos y de satisfacción del cliente. Muchas devoluciones se deben a problemas prevenibles como descripciones incorrectas, expectativas mal gestionadas o patrones de comportamiento del usuario.

**Objetivo:** Diseñar un sistema de software basado en IA que prediga la probabilidad de devolución de un producto en tiempo real antes de la compra, y proponga acciones preventivas (alertas, recomendaciones, validaciones adicionales) para reducir la tasa de devoluciones.

**Tipo de datos involucrados:**

- **Datos de transacciones de compra**: fecha, usuario, producto, método de pago.
- **Historial de devoluciones**: razones de devolución, fechas, tipo de producto.
- **Descripciones de productos**: texto, imágenes, especificaciones técnicas.
- **Reseñas y calificaciones de usuarios**: texto, estrellas, categorías.
- **Datos de atención al cliente**: chats, tickets, reclamos.

**Origen de los datos:**

- **Bases de datos internas de Amazon** (historial de órdenes, devoluciones, reseñas).
- **Sistemas de gestión de inventarios y logística**.
- **Amazon Customer Support** – logs de interacciones y reclamos.
- **Amazon Retail Intelligence APIs** – para enriquecer con datos externos del mercado.

**Áreas involucradas:**

- **Ingeniería de Software / Backend Development**
- **Data Science y Machine Learning**
- **UX/UI y Customer Experience**
- **Logística y Operaciones**
- **Producto / Gerencia Comercial**
- **Legal (por el uso de datos personales)**

**Modelos de IA a utilizar**

- **LightGBM**: Para la predicción de la devolución usando datos estructurados.
- **BERT**: Para el procesamiento de las reseñas y usar comentarios como indicadores de devolución.

**Entregable del proyecto:**

- **Sistema integrado en el frontend de Amazon** que:
    - Muestra alertas personalizadas: "Este producto tiene una alta tasa de devolución por talla, verifica tu medida".
    - Recomienda productos alternativos si la predicción de devolución es alta.
- **Dashboard interno** para operaciones, con métricas de predicción de devolución por categoría.
- **API REST interna** para servir el modelo a los diferentes sistemas de Amazon (web, móvil, etc.).

**Tecnologías requeridas:**

- **Lenguajes:** Python (ML, backend), JavaScript/TypeScript (frontend).
- **Infraestructura:** AWS (SageMaker, Lambda, API Gateway, S3, DynamoDB, CloudWatch).
- **Frameworks ML:** Scikit-learn, TensorFlow, PyTorch.
- **Bases de datos:** DynamoDB.
- **Herramientas de visualización:** Amazon QuickSight.

**Tiempos y costos aproximados:**

Alrededor de 6 meses y una inversión inicial de $100,000 dólares.

# Propuesta Martin Jaramillo

## Introducción

En la industria automotriz, las etapas de validación de sistemas eléctricos representan una fase crítica para asegurar la calidad, seguridad y cumplimiento normativo de los vehículos. Durante estas pruebas, se recopilan grandes volúmenes de datos provenientes de redes CAN, dataloggers, reportes técnicos y documentación interna. Sin embargo, persisten diversos retos, entre los que destacan la detección de fallas intermitentes y la constante

actualización de los documentos de validación (process standards), que frecuentemente requieren la adaptación de los planes de prueba. En este contexto, la aplicación de técnicas de Inteligencia Artificial (IA), incluyendo el Procesamiento de Lenguaje Natural (PLN), se plantea como una solución viable para optimizar estos procesos y aumentar su trazabilidad y eficiencia.

## Descripción del Problema

Durante las pruebas de validación de sistemas eléctricos en vehículos de desarrollo, se presentan dos desafíos principales. Por un lado, las fallas intermitentes en la red CAN pueden generar reprocesos, retrasos en el cronograma de pruebas y un uso ineficiente del tiempo de ingeniería. Estas fallas son difíciles de predecir con herramientas convencionales, ya que pueden depender de combinaciones de eventos en el tiempo y condiciones de prueba. Por otro lado, los documentos conocidos como process standards, que establecen las pruebas eléctricas requeridas para cada módulo, sufren cambios constantes debido a actualizaciones normativas o de diseño. Actualmente, no existe una solución automatizada que traduzca estos cambios en modificaciones concretas de los planes de prueba, lo que puede llevar a ejecutar pruebas desactualizadas o incompletas. Este proyecto busca implementar modelos de IA que aborden ambos problemas de forma integral.

## Tipo de datos involucrados:

1. **Logs de red CAN:**
    - ID de mensajes, timestamps, frecuencia, errores, datos de voltaje y corriente por nodo.
2. **Documentos de process standard (versiones anteriores y actuales):**
    - Archivos PDF, Word, o Excel que contienen la descripción de pruebas eléctricas requeridas por módulo.
3. **Reportes de validación manual:**
    - Bitácoras de pruebas, resultados de ejecución, evidencias fotográficas o gráficas.
4. **Historial de cambios/documentación técnica:**
    - Notas de ingeniería, tickets de cambio, logs de versiones (change control).
5. **Datos contextuales del vehículo en pruebas:**
    - Condiciones de temperatura, vibración, humedad, duración de pruebas, lote de producción.

## Origen de los datos:

- **Logs CAN:** extraídos desde herramientas como **CANalyzer**, **CANoe** o dataloggers instalados en vehículos prototipo.
- **Process standards:** provienen del sistema de gestión documental interno de la empresa (como **Teamcenter**, **Confluence**, **Windchill**, etc.).
- **Reportes de validación:** generados por ingenieros de validación usando formularios estándar o tickets en plataformas como **Jira** o **eQMS**.
- **Historial de cambios:** almacenado en repositorios internos de control de versiones o PLM.
- **Datos del vehículo:** adquiridos por sensores durante pruebas o sistemas de adquisición de datos (DAQ).

## Áreas o departamentos involucrados:

1. **Ingeniería de Validación Eléctrica:** ejecutores de pruebas y usuarios principales del sistema.
2. **Diagnóstico y Sistemas Eléctricos:** soporte en interpretación de fallas e integración con arquitectura.
3. **IT / Data Science / IA interna:** desarrollo del modelo y almacenamiento de datos.
4. **Calidad / PM / Ingeniería de Producto:** interesados en los reportes generados y trazabilidad.
5. **Design Release Engineers:** proveedores y gestores de los process standards.

## Modelos de IA que se podrían utilizar para la solución:

Para abordar la predicción de fallas en la red CAN, se propone el uso de modelos de series temporales como LSTM (Long Short-Term Memory), que permiten identificar patrones complejos en secuencias de mensajes. Complementariamente, los autoencoders serían útiles para detectar anomalías en las señales eléctricas asociadas, mientras que algoritmos supervisados como Random Forest y XGBoost podrían emplearse para clasificar eventos de falla con alta precisión. Para interpretar los resultados y entender qué variables o señales contribuyeron a cada predicción, se utilizaría la metodología SHAP, orientada a la explicabilidad de modelos.

Por otro lado, para automatizar el análisis de documentos técnicos y actualizaciones en los process standards, se emplearían técnicas de Procesamiento de Lenguaje Natural (PLN). En este caso, modelos tipo Transformer como BERT o RoBERTa permitirían extraer instrucciones relevantes desde textos técnicos no estructurados. Además, se aplicarían enfoques de detección de cambios semánticos entre versiones de documentos, apoyados

por herramientas de comparación de texto y embeddings lingüísticos. También se utilizaría Named Entity Recognition (NER) para identificar entidades clave como módulos, parámetros o procedimientos, así como clasificadores de texto para categorizar las pruebas según su tipo, criticidad o área funcional.

## Entregables del Proyecto

El proyecto entregará un modelo predictivo entrenado capaz de identificar fallas probables en la red CAN con base en datos históricos. Adicionalmente, se desarrollará un sistema automatizado que analice y traduzca los cambios en los process standards en actualizaciones claras para los planes de validación. Ambos resultados estarán integrados en un panel interactivo que permita la visualización de riesgos, cambios y recomendaciones, junto con un reporte técnico completo para su uso en auditorías internas y revisiones de ingeniería.

## Tecnologías Requeridas

- Lenguajes de programación: Python (NumPy, pandas, scikit-learn, TensorFlow o PyTorch)
- Herramientas de análisis de red: CANalyzer, CANoe
- Procesamiento documental: OCR, SpaCy, HuggingFace Transformers
- Almacenamiento de datos: PostgreSQL, MongoDB
- Visualización: Power BI o Tableau
- Infraestructura: Servidor local o nube (AWS, Azure), Git, Jira

## Tiempos Estimados

- Recolección y limpieza de datos: 2 semanas
- Desarrollo de modelos IA y PLN: 3 semanas
- Integración de dashboard y automatización: 2 semanas
- Pruebas piloto y validación con usuarios: 2 semanas
- Documentación final y ajustes: 1 semana

**Total estimado:** 10 semanas

## Costos Aproximados

- Licencias y herramientas (Power BI/Tableau, OCR, almacenamiento): $1,000 USD
- Infraestructura y servidores: $500 USD

- Horas de ingeniería (data science, validación, documentación): $6,000 USD

**Costo total estimado:** $7,500 USD

---

---

# Modelo LLM a utilizar: Chat GPT o3

Se utilizará debido a su capacidad de razonamiento que permite profundizar más e incluso hacer investigación en línea dentro de su cadena de pensamiento para complementar la porpuesta.

---

---

# Segunda parte

## Prompt Versión 1

¿Cómo aplico IA en la validación de sistemas eléctricos de un vehículo?

## Prompt Versión 2

Dispongo de logs CAN, reportes manuales y 'process standards' en PDF. ¿Cómo usar IA para detectar fallas intermitentes y actualizar planes de prueba?

## Prompt Versión 3

Como ingeniero de validación eléctrica, necesito:

1. Fases del proyecto (recolección, entrenamiento, pruebas, despliegue)
2. Modelos IA sugeridos (LSTM, autoencoder, BERT...)
3. Roles implicados (Validación, Data Science, IT...)
4. Cronograma con hitos
5. Entregables (dashboard, reporte)

## Prompt Versión 4

Actúa como consultor de IA y elabora una propuesta de proyecto que incluya:

- Problema y objetivos (fallas CAN + actualización automática de standards)

- Fases detalladas (datos, modelado LSTM+autoencoder, PLN BERT, pruebas, despliegue)
- Modelos y herramientas (tensorflow LSTM, HuggingFace RoBERTa, OCR, PostgreSQL)
- Roles y departamentos involucrados
- Cronograma con horas y coste aproximado
- KPIs (precisión de fallo, tiempo de reproceso, velocidad de actualización)
- Entregables concretos (panel interactivo, business case)

## Prompt Versión 5

Como consultor de IA en automotriz, genera una propuesta de proyecto que incluya:

1. **Descripción**: detección de fallas intermitentes en logs CAN y adaptación automática de 'process standards'.
2. **Fases**:
    a. Recolección/limpieza de datos CAN y documentación
    b. Entrenamiento LSTM (TensorFlow-Keras) + autoencoder para anomalías
    c. Análisis de texto con RoBERTa (OCR + SpaCy) y detección semántica de cambios
    d. Pruebas piloto, despliegue y plan de mantenimiento
3. **Modelos y tech stack**: versiones concretas y porqué (p.ej. LSTM v2.5, RoBERTa-base, PostgreSQL, AWS EC2).
4. **Roles**: Validación Eléctrica, Data Science/IT, Calidad, Producto, Legal.
5. **Cronograma**: hitos semanales con horas y coste estimado.
6. **KPIs**: precisión de detección, reducción de reprocesos, TTR de actualización.
7. **Riesgos y mitigación**: calidad de datos, sesgos ML, cambios normativos.
8. **Ejemplos de prompts**:
    a. Extraer requisitos de un PDF de standards
    b. Comparar semánticamente dos versiones
    c. Analizar secuencias CAN y marcar anomalías
9. **Plantillas**:
    a. Esquema de business case (PowerPoint)
    b. Mockup de dashboard Power BI" | • Cubre absolutamente todos los requisitos y añade:

- Gestión de riesgos
- Prompts de ejemplo
- Plantillas de entregables

# Respuesta Prompt Versión 5

# Propuesta de Proyecto

## Detección de fallas intermitentes en logs CAN y adaptación automática de “process standards”

**Consultor:** Equipo IA Automotriz

**Fecha:** 25 de junio de 2025

## 1. Descripción del proyecto

La industria automotriz genera volúmenes crecientes de datos CAN (Controller Area Network) y documentación técnica regulatoria. Las fallas intermitentes —difíciles de replicar y, por lo tanto, costosas— se ocultan en secuencias extensas de mensajes. Este proyecto propone un sistema basado en IA que:

1. Detecte anomalías en tiempo (sub-segundos) dentro de los logs CAN mediante modelos LSTM y autoencoders.
2. Analice automáticamente los cambios semánticos en los ‘process standards’ (normativas internas y externas) para actualizar parámetros de diagnóstico.
3. Ajuste los parámetros de detección y las reglas de validación eléctrica de forma continua (MLOps) con mínima intervención humana.

**Objetivos SMART**

- *Precisión* ≥ 95 % (F1) en detección de anomalías intermitentes.
- Reducir reprocesos en laboratorio un 30 % en 6 meses.
- TTR (*Time-to-Review*) de actualización normativa ≤ 48 h tras la publicación de un nuevo estándar.

## 2. Alcance y entregables

| Fase | Entregable principal |
|---|---|
| 1. Recolección y limpieza de datos | Dataset CAN etiquetado + repositorio de estándares digitalizados |
| 2. Modelado y entrenamiento | Pesos TensorFlow (.h5) + pipeline Autoencoder-LSTM |
| 3. Análisis semántico | Servicio REST RoBERTa + spaCy + motor de reglas |
| 4. Prueba piloto | Informe de validación y dashboard Power BI |
| 5. Despliegue | Infraestructura IaC (Terraform) + playbook de MLOps |
| 6. Mantenimiento | Plan de soporte y métricas en producción |

## 3. Fases, actividades y duración

### 3.1 Fase 1 – Recolección / limpieza de datos CAN y documentación (Semanas 1-3)

- Inventario de buses CAN y extracción histórica (30 GB).
- Estandarización de formatos DBC y sincronización de sellos de tiempo.
- OCR de PDF/DWG → JSON (tesseract-ocr 5.3 + pdfplumber 0.11).
- **Horas:** 240 (80 h/sem).

### 3.2 Fase 2 – Entrenamiento LSTM + Autoencoder (Semanas 4-7)

- Arquitectura LSTM v2.5 (2 × 512 gru, dropout 0.3) + Autoencoder denso (latent = 128).
- Framework TensorFlow-Keras 2.15, Python 3.11.
- HPO con Keras-Tuner y 4 × GPU A10G en AWS EC2 g5.
- **Horas:** 320.

### 3.3 Fase 3 – Análisis de texto con RoBERTa (Semanas 4-6, en paralelo)

- Fine-tuning RoBERTa-base (transformers 4.41) sobre corpus automotriz.
- Detección de cambios semánticos (similaridad > 0.85) y entidades normativas.
- **Horas:** 160.

### 3.4 Fase 4 – Pruebas piloto (Semanas 8-9)

- Integración con banco de pruebas eléctrica.
- Comparativa A/B frente a proceso actual.
- **Horas:** 120.

### 3.5 Fase 5 – Despliegue (Semanas 10-11)

- Contenedores Docker en AWS EKS + PostgreSQL 16 (Aurora) + S3.
- IaC con Terraform 1.8 + GitHub Actions CI/CD.
- **Horas:** 120.

### 3.6 Fase 6 – Plan de mantenimiento (Semana 12 + continuo)

- Retraining trimestral automático.
- Monitor de drifts (Evidently 2.3) y alertas PagerDuty.
- **Horas (setup):** 40. **Horas anuales O&M:** 480.

## 4. Modelos y stack técnico

| Capa | Herramienta & versión | Motivo de elección |
|---|---|---|
| Procesamiento CAN | **Python 3.11**, pandas 2.2, numpy 2.0 | Ecosistema maduro y soporte amplio |
| Deep Learning | **TensorFlow-Keras 2.15**, **LSTM v2.5**, Autoencoder custom | Rendimiento GPU, API estable |
| NLP | **RoBERTa-base**, HuggingFace transformers 4.41 | Transfer learning en español/inglés; tamaño manejable |
| PLN pipeline | **spaCy 3.7**, **tesseract-ocr 5.3** | Eficiencia y precisión en OCR + NER |
| Base de datos | **PostgreSQL 16** (AWS Aurora) | ACID, JSONB, escalabilidad |
| Infraestructura | **AWS EC2 g5**, **EKS**, **S3**, **Terraform 1.8** | Elasticidad y DevOps automatizado |
| Dashboard | **Power BI Service (jun-2025)** | Integración con Azure/AWS SQL, visual storytelling |

## 5. Roles y responsabilidades

| Rol | Área | Responsabilidades clave |
|---|---|---|
| Validación Eléctrica | Ingeniería de Pruebas | Definir métricas de fallo, validar señales CAN, aceptar pilotos |
| Data Science / IT | CoE Analytics | Modelado DL/NLP, MLOps, automatización |
| Calidad | Calidad Corporativa | Aseguramiento de procesos, auditoría ISO 26262 |
| Producto | Gestión de Programas | Priorización de features, ROI, roadmap |
| Legal & Compliance | Legal | Evaluar implicaciones regulatorias (GDPR, UNECE R155/156) |

## 6. Cronograma, horas y coste estimado

| Semana | Hito | Horas | Costo (*USD, tarifa promedio 120 $/h*) |
|---|---|---|---|
| 1 | Kick-off & capturas iniciales | 80 | 9 600 |
| 2 | ETL CAN + OCR estándares | 80 | 9 600 |
| 3 | Dataset final y data lake | 80 | 9 600 |
| 4 | Arquitectura modelo + HPO start | 80 | 9 600 |
| 5 | Entrenamiento iteración 1 | 80 | 9 600 |
| 6 | Entrenamiento iteración 2 | 80 | 9 600 |
| 7 | Validación métrica interna | 80 | 9 600 |
| 8 | Integración banco de pruebas | 60 | 7 200 |
| 9 | Informe piloto & dashboard | 60 | 7 200 |
| 10 | Contenedorización y IaC | 60 | 7 200 |
| 11 | Puesta en producción | 60 | 7 200 |
| 12 | Handover + capacitación | 40 | 4 800 |
| **Total** | **—** | **860** | **103 200** |

*Nota:* Coste de infraestructura AWS no incluido (~4 400 $/mes g5 + Aurora + tráfico).

## 7. KPIs clave

1. **Precisión de detección de fallas (F1)** ≥ 95 %.
2. **Reducción de reprocesos** (%) vs. línea base.
3. **TTR de actualización normativa** (h).
4. **Cobertura de standards analizados** (# docs vs. total).
5. **Drift modelado** (divergencia KL < 0.05).

## 8. Gestión de riesgos y mitigación

| Riesgo | Impacto | Prob. | Mitigación |
|---|---|---|---|
| Calidad insuficiente de datos CAN | Falsos positivos | Alta | Reglas de validación, sensores redundantes |
| Sesgos en datos de entrenamiento | Model bias | Media | Muestreo estratificado, fairness metrics |
| Cambios normativos abruptos | Desfase en parámetros | Media | Monitoreo continuo de UNECE, retraining rápido |
| Ciberseguridad (R155) | Brechas de datos | Baja | Hardening EKS, escaneo imágenes |
| Dependencia de AWS GPU | Costos ↑ | Baja | Reservas 1 año, Spot instances |

## 9. Ejemplos de prompts de uso

1. **Extraer requisitos de un PDF de standards**

Analiza el PDF adjunto y lista los requisitos numerados de la sección 4.2 en formato JSON {id, texto, nivel}.

2. **Comparar semánticamente dos versiones**

Compara “standard_v3_2024.pdf” y “standard_v4_2025.pdf” y destaca cambios con similaridad < 0.85.

3. **Analizar secuencias CAN y marcar anomalías**

Detecta fallas intermitentes en el log can_log_2025-06-24.csv y resume las top-5 con score > 0.9.

# 10. Plantillas de entregables

## 10.1 Esquema Business Case (PowerPoint)

1. **Título & resumen ejecutivo**
2. Problema y oportunidad
3. Análisis de costos (TCO) vs. beneficios
4. Solución propuesta (Arquitectura, IA, flujos)
5. KPIs y ROI
6. Riesgos y plan de mitigación
7. Roadmap y siguientes pasos

## 10.2 Mockup Dashboard Power BI

- **Página 1 – KPIs principales**: tarjetas F1, reprocesos, TTR.
- **Página 2 – Timeline de fallas**: gráfico de dispersión tiempo-vs-anomalía.
- **Página 3 – Heatmap estándares**: similaridad vs. fecha publicación.
- **Filtros**: vehículo, ECU, fecha, estándar.

# 11. Próximos pasos

1. Aprobación del presupuesto (Dirección Producto & Finanzas).
2. Firma de SOW y NDA con áreas Legal y Calidad.
3. Puesta en marcha de Fase 1 (Semana 1).

**Contacto:** consultoria-ia@empresa.com | ext. 2025

# Conclusión

Esta actividad representó una valiosa oportunidad para aplicar los conocimientos adquiridos en el curso de Procesamiento de Lenguaje Natural (PLN) en contextos reales y diversos, alineados con nuestras áreas profesionales. Cada integrante del equipo propuso una idea original que abordó problemáticas concretas desde distintos sectores: recursos humanos, salud, comercio electrónico y validación automotriz. Este enfoque multidisciplinario enriqueció el proceso, al evidenciar la amplitud de aplicaciones del PLN en la industria.

La propuesta seleccionada para su desarrollo detallado, centrada en la **detección de fallas intermitentes en redes CAN y la automatización del análisis de process standards en la industria automotriz,** permitió explorar en profundidad una solución técnica robusta, basada en modelos LSTM, autoencoders y Transformers como RoBERTa. A lo largo del proceso se utilizó de manera estratégica la ingeniería de prompts con LLMs (en este caso, ChatGPT o3) para guiar el diseño de cada fase del proyecto, lo que facilitó una estructuración clara, medible y viable, incluyendo cronogramas, KPIs, entregables y planes de mitigación de riesgos.

Este ejercicio no solo reforzó nuestra comprensión técnica del PLN, sino también nuestras habilidades en gestión de proyectos de IA, integración multidisciplinaria y pensamiento estratégico. Como resultado, el documento generado puede servir como un punto de partida sólido para la implementación real del proyecto dentro de un entorno empresarial. En resumen, esta actividad nos permitió conectar la teoría con la práctica, aplicar IA de manera responsable y útil, y fortalecer nuestras capacidades como futuros líderes en inteligencia artificial aplicada.