



“Deep Learning y Natural Language processing para generar un chatbot”

Sprint 2

Bernardo Corona Domínguez

NAO ID: 1085

Ciudad de México, México

12 de junio de 2025

## 1. Backlog.

### A) Historias de usuario.

A continuación, las 5 (cinco) historias de usuario:

1. **Como seguidor del canal de astrología**, quiero interactuar con un chatbot inteligente que responda mis preguntas sobre predicciones astrológicas en tiempo real, para obtener respuestas rápidas y precisas.
2. **Como creadora de contenido (Amira Rashid)**, quiero poder actualizar regularmente el chatbot con las preguntas frecuentes de mis seguidores, para mejorar continuamente la calidad y precisión de las respuestas.
3. **Como desarrollador**, quiero contar con scripts documentados y accesibles para facilitar futuras actualizaciones y mantenimientos al modelo del chatbot.
4. **Como administrador de la aplicación web**, quiero monitorear el desempeño del chatbot en tiempo real, para asegurar su eficiencia operativa y experiencia de usuario óptima.
5. **Como visitante casual de la página**, quiero encontrar una interfaz amigable e intuitiva del chatbot, para poder realizar consultas sin complicaciones técnicas.

### B) Tablas de registro.

Historia de Usuario	Requerimientos Específicos
<b>Como seguidor del canal de astrología, quiero interactuar con un chatbot inteligente.</b>	- Interfaz interactiva web desarrollada con Flask y HTML. - Integración del modelo NLP y Deep Learning para respuestas automáticas.
<b>Como creadora de contenido (Amira Rashid), quiero actualizar regularmente el chatbot.</b>	- Mecanismo sencillo para añadir nuevas preguntas frecuentes. - Script que permita reentrenar el modelo de TensorFlow rápidamente.

<b>Como desarrollador, quiero scripts documentados y accesibles para futuras actualizaciones.</b>	- Código Python organizado en módulos claros y documentados. - Comentarios explícitos para el mantenimiento futuro.
<b>Como administrador, quiero monitorear el desempeño del chatbot.</b>	- Sistema de registro de logs y métricas de desempeño. - Panel básico de administración para monitoreo en tiempo real.
<b>Como visitante casual, quiero una interfaz amigable para realizar consultas fácilmente.</b>	- Diseño claro e intuitivo del front-end HTML/CSS. - Funcionamiento ágil del backend Flask para consultas rápidas.

#### Lista Priorizada.

Requerimientos	Etapas	Estimación de Tiempo	Entregables
<b>Scripts en Python para procesamiento NLP (SpaCy) y creación de bolsa de palabras.</b>	Fase 1 - Tokenizado	2 semanas	Scripts funcionales para procesamiento inicial de datos.
<b>Desarrollo del modelo de Deep Learning (TensorFlow).</b>	Fase 2 - Modelo DL	3 semanas	Modelo entrenado, probado y documentado con TensorFlow.
<b>Aplicación Web en Flask y diseño HTML para interfaz de usuario.</b>	Fase 3 - Front-end	2 semanas	Aplicación funcional con interfaz web y backend Flask.
<b>Módulo de monitoreo del desempeño (logs y métricas).</b>	Fase 4 - Monitoreo	1 semana	Panel funcional básico para monitoreo operativo del chatbot.

<b>Video demostrativo de interacción chatbot-usuario para presentación final.</b>	Fase 5 - Demostración	1 semana	Archivo MP4 mostrando interacción y eficacia del chatbot.
---	-----------------------	----------	---

## Lista de requerimientos

Historias de Usuario	Requerimientos Específicos y Técnicos
<b>Como seguidor del canal de astrología, quiero interactuar con un chatbot inteligente.</b>	- Interfaz web atractiva y responsiva.- Procesamiento de consultas en tiempo real utilizando NLP y TensorFlow.- Respuestas claras y precisas adaptadas a contextos astrológicos específicos.
<b>Como creadora de contenido (Amira Rashid), quiero actualizar regularmente el chatbot.</b>	- Interfaz o script sencillo para actualizar preguntas frecuentes y respuestas predeterminadas.- Sistema de entrenamiento incremental del modelo NLP.
<b>Como desarrollador, quiero scripts documentados y accesibles para futuras actualizaciones.</b>	- Scripts en Python organizados por funcionalidades: ingestión de datos, entrenamiento de modelo, despliegue.- Documentación exhaustiva en comentarios y README.md para facilitar el desarrollo colaborativo.
<b>Como administrador, quiero monitorear el desempeño del chatbot.</b>	- Sistema automático de logging que capture consultas, respuestas y tiempos de respuesta.- Herramienta de visualización en tiempo real del rendimiento del modelo (porcentaje de aciertos, latencia).
<b>Como visitante casual, quiero una interfaz amigable para realizar consultas fácilmente.</b>	- Diseño web intuitivo con formulario de entrada simple para consultas.- Tiempo de respuesta inferior a 2 segundos.- Instrucciones sencillas de uso visibles claramente en la página.

**Tabla 2. Lista de Requerimientos Priorizados**

Requerimiento	Etapas	Estimación de Tiempo	Entregable
<b>Desarrollo del modelo de Deep Learning (TensorFlow) para clasificación de intenciones.</b>	Fase 2 - Modelo DL	3 semanas	Modelo entrenado, probado y documentado listo para predicción.
<b>Generación y almacenamiento de archivos de salida para predicción (weights/modelo, vectorizadores, mapeo de etiquetas, etc.).</b>	Fase 2 - Output de modelo	3 días	Archivos .h5, .pkl y/o .json que faciliten la carga y uso del modelo.
<b>Documentación del proceso de entrenamiento y uso del modelo.</b>	Fase 2 - Documentación	2 días	Sección README ampliada, guía de uso y comentarios en el código fuente.

<b>Pruebas de validación del modelo con datos de ejemplo.</b>	Fase 2 - Validación	2 días	Resultados de validación (matriz de confusión, métricas de desempeño).
<b>Preparación de scripts para reutilización y reentrenamiento futuro.</b>	Fase 2 - Scripts futuros	2 días	Scripts documentados y listos para actualización con nuevos datos.