Desarrollo y personalización de un chat de inteligencia artificial basado en modelo largo de lenguaje

Autor

Esp. Ing. Jorge Hernán Cuenca Marín

Director del trabajo

Dr. Mg. Ing. Camilo Torres

Este plan de trabajo ha sido realizado en el marco de la asignatura Gestión de Proyectos entre marzo y abril de 2024.

Tabla de contenido

IMPORTANTE: No borrar las consignas en cada una de las cuatro secciones de este documento, de forma tal que el jurado tenga claro qué es lo solicitado en cada caso, así como el significado de los símbolos y colores utilizados.

Revisión	Cambios realizados	Fecha
1.0	Creación del documento	27/10/2024

1. Breve resumen del trabajo realizado hasta la fecha

Elabore un detalle del estado del proyecto a la fecha. **Utilicé como mínimo dos páginas completas y como máximo tres páginas**. Explique muy brevemente en qué consiste su Trabajo Final, **aunque esa información esté más detallada en el Plan de Trabajo al cual su Jurado también tiene acceso**. Incluya imágenes y tablas según considere apropiado. **Indique con claridad por qué estima que podrá completar todos los faltantes** (o al menos la gran mayoría) **antes del inicio del Taller de Trabajo Final.**

El proyecto en curso titulado Desarrollo y Personalización de un Chat de Inteligencia Artificial Basado en un Modelo Largo de Lenguaje tiene como objetivo principal crear un sistema interactivo que utilice un modelo de lenguaje largo (LLM) para proporcionar respuestas precisas y contextualizadas a las preguntas de los usuarios. Este sistema se ha desarrollado utilizando varias tecnologías avanzadas como FastAPI, PostgreSQL, Django, y técnicas de finetuning para modelos de lenguaje LLaMA. El presente documento describe el estado actual del proyecto, los avances logrados y las razones por las cuales se estima que se podrán completar los faltantes antes del inicio del Taller de Trabajo Final.

El trabajo final se centra en el desarrollo de un chat inteligente que interactúa con los clientes de Bitlink, proporcionando respuestas rápidas y precisas sobre los servicios de la empresa. La arquitectura del sistema incluye un backend robusto implementado en FastAPI, que maneja la recepción de preguntas y el envío de respuestas generadas por el modelo LLM de Bitlink, previamente entrenado y ajustado utilizando técnicas estándar de finetuning.

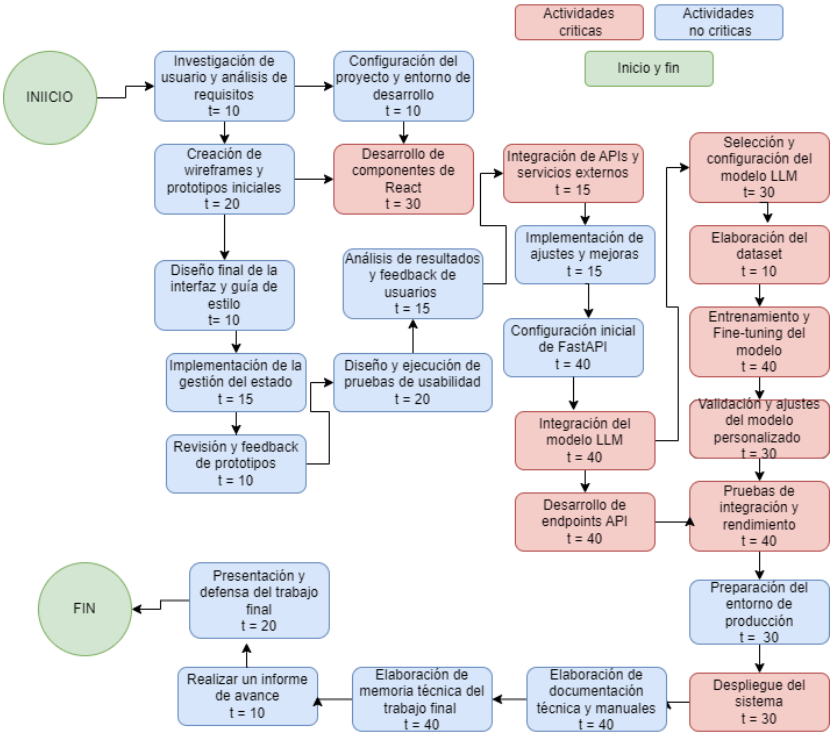

Figura 1: Arquitectura del Sistema

Avances del Proyecto

Desarrollo de la API con FastAPI

Hasta la fecha, se ha desarrollado una API utilizando FastAPI que tiene los siguientes endpoints:

1. Recepción de Preguntas y Respuestas: Este endpoint permite recibir preguntas de los usuarios y regresar respuestas generadas por el LLM de Bitlink. El proceso incluye la aplicación de una técnica de Retrieval-Augmented Generation (RAG) utilizando un documento PDF con información específica de Bitlink.

2. Actualización del Documento para RAG: Otro endpoint permite actualizar el documento PDF utilizado para RAG. Este endpoint recibe un nuevo PDF y reemplaza el documento anterior, permitiendo la actualización continua de la información.

Interacción con el Frontend

El frontend, ya desarrollado, permite a los clientes interactuar con la API a través de un chat. Este chat facilita la comunicación entre los usuarios y la inteligencia artificial, permitiendo una atención rápida y

eficiente. Además, el sistema presenta dos respuestas posibles al cliente, quien puede seleccionar la que considere más adecuada.

Gestión de Preguntas y Respuestas con Django y PostgreSQL

La base de datos PostgreSQL gestiona las preguntas y respuestas preconfiguradas por Bitlink. Utilizando el framework Django, se ha desarrollado un panel de administración donde los técnicos de Bitlink o el gerente pueden observar el historial de interacciones de las personas con el chat, editar respuestas y gestionar la actualización del documento para RAG.

Actualización Automatizada del RAG

Cada cierto tiempo, Bitlink puede actualizar el documento utilizado para RAG. Desde una vista en Django, se recopila el historial de preguntas y respuestas de los usuarios, se convierte en un PDF y se envía al endpoint correspondiente de la API para reemplazar el documento anterior. Este proceso asegura que la información se mantenga actualizada y que las respuestas del modelo de lenguaje sean cada vez más precisas.

2. Avance en las tareas

a) Indicar a continuación para cada una de las tareas su estado de situación según su criterio, utilizando verde si considera que es satisfactorio, amarillo si considera que es insatisfactorio por sobrecostos y/o demoras, y rojo si lo considera muy insatisfactorio por sobrecostos y/o demoras.
Si a la fecha de completar este informe no está previsto que la tarea haya comenzado entonces deje la celda correspondiente en blanco, sin pintarla con ningún color.

En subcelda inferior izquierda colocar:

- ** si los recursos u horas utilizadas fueron o están siendo muy inferior a lo planificado.
- * si los recursos u horas utilizadas fueron o están siendo inferior a lo planificado.
- $ si los recursos u horas utilizadas fueron o están siendo de acuerdo a lo planificado.
- $$ si los recursos u horas utilizadas fueron o están siendo superior a lo planificado.
- $$$ si los recursos u horas utilizadas fueron o están siendo muy superior a lo planificado.

En subcelda inferior derecha colocar:

- -- si la tarea se ejecutó o se está ejecutando mucho más rápido de lo previsto
- - si la tarea se ejecutó o se está ejecutando más rápido de lo previsto
- = si la tarea se ejecutó o se está ejecutando en el tiempo previsto.
- + si la tarea se ejecutó o se está ejecutando con demoras.
- ++ si la tarea se ejecutó o se está ejecutando con demoras muy significativas.

IMPORTANTE: Indicar con borde grueso las tareas que forman parte del camino crítico

1.0 Investigación de usuarios y an´alisis de requisitos		1.1 Creación de wireframes y prototipos iniciales		1.2 Revisión y feedback de prototipos		1.3 Diseño final de la interfaz y guia de estilo Y otra más	
$	=	$	=	$$$	=	*	=
1.4 Configuración del proyecto y entorno de desarrollo		1.5 Desarrollo de componentes de React		1.6 Integración de APIS y servicios externos		1.7 Implementación de la gestión del estado	
**	--	*	-	$	=	$	=
1.8 Diseño y ejecución de pruebas de usabilidad		1.9 Analisis de resultados y feedback de usuarios		1.10 Implementación de ajustes y mejoras			
$	=	$	=	$	=		
2.0 Configuración inicial de FastAPI		2.1 Integración del modelo LLM		2.2 Desarrollo de endpoints API		2.3 Pruebas de integración y	

						rerendimiento	
**	-	$	=	$	=	$	=
3.0 Selección y configuración del modelo LLM		3.1 Elaboración del dataset		3.3 Entrenamiento y fine-tuning del modelo		3.4 Validación y ajustes del modelo personalizado	
$$	++	$	+	$	=	$	=
4.0 Preparación del entorno de producción		4.1 Despliegue del sistema		4.2 Elaboración de documentación técnica y manuales			
**	++	$	=	$	=		
5.0 Elaboración de la memoria técnica del trabajo final		5.1 Elaboración del informe de avance		5.2 Presentación y defensa del trabajo final			
$	=	$	=				

3. Cumplimiento de los requerimientos

a) Indicar a continuación para cada uno de los requerimientos el estado de situación según su criterio, utilizando verde si considera que ya se ha cumplido, amarillo si considera que aún no se a cumplido pero se podrá cumplir, y rojo si considera que aún no se ha cumplido y tiene dudas si se podrá cumplir.

Si considera que es necesario modificar los requerimientos respecto a los indicados en la planificación inicial entonces incluya acá los requerimientos actualizados, **marcando en negrita** aquellos que son nuevos o se han modificado.

Req #1: El sistema debe permitir a los usuarios interactuar mediante lenguaje natural con el chatbot en la plataforma de BITLINK.
Req #2: La interfaz del usuario, creada con React.js, debe ser responsiva y adaptativa a diferentes dispositivos y tamaños de pantalla.

Req #3: El backend, desarrollado con FastAPI, debe gestionar las solicitudes de forma asincrónica para garantizar tiempos de respuesta rápidos.
Req #4: El sistema debe integrar un modelo de lenguaje pre entrenado con la capacidad de ser afinado (fine-tuning) para ajustarse a los datos específicos de BITLINK.
Req #5: El usuario debe poder recibir respuestas precisas y contextuales basadas en el conocimiento personalizado de BITLINK que se hace con el RAG.
Req #6: Debe existir una documentación completa del código fuente y del sistema.
Req #7: Los manuales de usuario y mantenimiento deben estar disponibles y ser fáciles de entender (prioridad menor).
Req #8: Se deben realizar pruebas unitarias para cada componente del sistema.
Req #9: Se deben realizar pruebas de integración para asegurar la interoperabilidad entre el frontend y el backend.
Req #10: La interfaz debe ser intuitiva y guiar al usuario a través de la funcionalidad del chat de forma natural.
Req #11: El sistema debe ser capaz de integrarse con APIs existentes de BITLINK sin afectar sus operaciones actuales.
Req #12: El sistema debe guardar de manera efectiva la información de las preguntas y respuestas en la base de datos.

4. Gestión de riesgos

a) Indicar a continuación para cada uno de los riesgos el estado de situación según su criterio, utilizando verde si considera que el riesgo ya no se manifestará o es muy improbable que se manifieste, amarillo si considera que es posible que es improbable que el riesgo se manifieste o si se manifiesta estima que será fácilmente controlado, y rojo si considera que es muy probable que el riesgo se manifieste y que no pueda ser controlado fácilmente.

Si considera que es necesario modificar los riesgos respecto a los presentados en la planificación inicial entonces incluya acá los riesgos actualizados, **marcando en negrita** aquellos que son nuevos o se han modificado, e indicando para ellos los valores de S, O y RPN, junto con su respectiva justificación.

Riesgo #1: falta de precisión en el reconocimiento de consultas del chatbot.
Riesgo #2: retrasos en el cronograma del proyecto .
Riesgo #3: problemas de integración de tecnologías.
Riesgo #4: falta de adopción del sistema por los usuarios finales.
Riesgo #5: dificultades en la personalización y entrenamiento del modelo LLM.