

Módulo de procesamiento speech to text para reconocimiento de tareas en proyectos mediante modelos LLM

Salomon David Saenz Giraldo
Dpto. de Ingeniería de Sistemas
Universidad del Norte
Barranquilla, Colombia
sdsaenz@uninorte.edu.co

Samuel David Constante Lopez
Dpto. de Ingeniería de Sistemas
Universidad del Norte
Barranquilla, Colombia
sconstante@uninorte.edu.co

Sebastian Barandica Oquendo
Dpto. de Ingeniería de Sistemas
Universidad del Norte
Barranquilla, Colombia
sebastianbarandica@uninorte.edu.co

Abstract– This project focuses on the development of a speech-to-text processing module for task recognition in projects using LLM models based on team meeting recordings. It leverages advanced tools such as Google Cloud Speech-to-Text, Deepgram, Gemini AI, and Vertex AI to obtain accurate transcriptions, which are then used to generate structured tasks that are stored in a database and accessible via a web interface. The adopted methodology includes the integration of large language models (LLM) and supervised learning to enhance the accuracy of transcriptions and task categorization. The results demonstrate a significant improvement in project management and tracking by automating the recognition and structuring of tasks, thus optimizing team productivity. This approach provides a solid foundation for future implementations in tools related to project management.

Keywords– *Speech-to-text, Task recognition, Project management, Large language models (LLM), Transcription accuracy, Supervised learning, Gemini AI, Vertex AI, Deepgram.*

I. INTRODUCCIÓN

La gestión de proyectos sigue siendo un reto significativo para los líderes de equipos, quienes deben coordinar diversas actividades, recursos y miembros del equipo para cumplir con los objetivos establecidos de manera eficiente. En un entorno de trabajo cada vez más dinámico y colaborativo, la necesidad de herramientas que optimicen y faciliten estos procesos es cada vez más urgente. Las metodologías ágiles, cuando se implementan de manera adecuada y ajustada a las necesidades del equipo, se han consolidado como una de las estrategias más efectivas para asegurar el éxito en la ejecución de proyectos, promoviendo flexibilidad y adaptabilidad frente a los cambios (Schwaber & Sutherland, 2020).

A su vez, los avances en las tecnologías de transcripción

automática de voz han revolucionado la documentación de reuniones, permitiendo transformar las grabaciones de audio en textos con una precisión cada vez mayor, lo que mejora considerablemente la eficiencia en la captura de información relevante (Miao et al., 2020). Estas tecnologías, junto con los progresos en el procesamiento de lenguaje natural (PLN), facilitan no solo la interpretación y estructuración del contenido transcrito, sino también la automatización de la asignación de tareas, lo cual optimiza la gestión de proyectos mediante la conversión de la información verbal en acciones concretas (Touvron et al., 2023).

Sin embargo, a pesar de estos avances, persiste el desafío de transformar de manera automática y eficiente las conversaciones de las reuniones en tareas asignables. Con el objetivo de cerrar esta brecha, el presente proyecto se centra en el desarrollo de un módulo innovador de gestión de tareas, basado en tecnologías de transcripción automática y procesamiento de lenguaje natural, que será capaz de procesar grabaciones de audio de reuniones, generar transcripciones precisas y asignar automáticamente las tareas identificadas a los miembros del equipo.

Este sistema propuesto no solo permitirá la transcripción eficiente de las reuniones, sino que también utilizará IA para estructurar la información de manera que facilite la asignación de tareas relevantes. A través de un proceso que integra tanto la transcripción como la categorización de las tareas discutidas, se logrará una optimización del flujo de trabajo y de la coordinación entre los miembros del equipo.

El propósito de este trabajo es desarrollar un prototipo de aplicación web que automatice la gestión de tareas mediante el uso de inteligencia artificial, con un enfoque particular en la asignación automática de tareas a partir de las transcripciones de las reuniones. Este prototipo se diseñará para ajustarse al flujo de trabajo específico del equipo, proporcionando una herramienta que no solo promueva la eficiencia en la administración de tareas, sino que también apoye la implementación

de metodologías ágiles en proyectos de software (Pellerin et al., 2013). De esta forma, el sistema propuesto busca reducir los errores y malentendidos, mejorando la productividad y fomentando una gestión más fluida y precisa de los proyectos en curso.

II. PROBLEMA

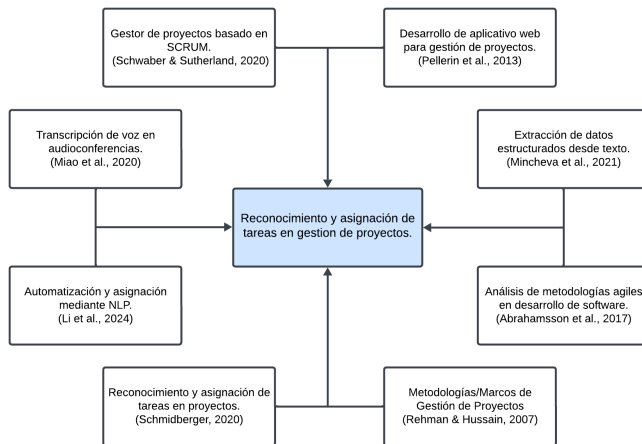


Figura 1: Mapa del problema.

En el entorno dinámico de trabajo en equipo, las reuniones son fundamentales para coordinar esfuerzos, definir tareas y tomar decisiones. Sin embargo, la conversión de estas discusiones en acciones concretas a menudo enfrenta desafíos significativos, como la pérdida de información importante, malentendidos sobre las asignaciones y la falta de seguimiento efectivo. Estos problemas pueden afectar la productividad y la eficacia del equipo.

El problema central que aborda este proyecto es la necesidad de mejorar la captura, procesamiento y gestión de la información generada en las reuniones de equipo para asegurar que las tareas y asignaciones se implementen de manera precisa y eficiente. Como podemos ver en la Figura 1, se identifican tres áreas clave:

Primero, el reconocimiento de voz y la transcripción de las discusiones en tiempo real son fundamentales, ya que las reuniones no siempre se documentan adecuadamente, lo que resulta en la pérdida de detalles cruciales. Contar con un sistema que transcriba automáticamente las conversaciones asegura que toda la información relevante se capture y esté disponible para su procesamiento posterior.

En segundo lugar, la simple transcripción de una reunión no garantiza la claridad en las asignaciones de tareas. Es necesario procesar esta información para extraer las tareas y asignaciones específicas. Aquí es donde entra en juego el uso de procesos de análisis de lenguaje natural con I.A como el explicado por Uchida (2024) a través de su examinación

de potencial en modelos LLMs. Que justados mediante fine-tuning o el uso de otras técnicas de NLP como las explicadas por Mincheva (2021), que permiten una comprensión más profunda del contexto. Identificando con precisión las tareas asignadas y los responsables al interpretar el contexto y las implicaciones de las discusiones. Esto es crucial para transformar la información bruta de la transcripción en datos estructurados y accionables.

Finalmente, una vez procesada la información, es crucial que esta se integre en un gestor de tareas que centralice y organice todas las asignaciones, plazos y responsables. Además, este gestor debe estar alineado con los principios de Scrum (Schwaber & Sutherland, 2020), garantizando que las tareas se gestionen de acuerdo con los ciclos de sprint. El cual debe facilitar no solo la visualización clara de las tareas, sino también la actualización de las mismas, asegurando que el equipo esté alineado y que no se pierda ninguna tarea asignada. Evidenciando el proceso completo en la Figura 2.

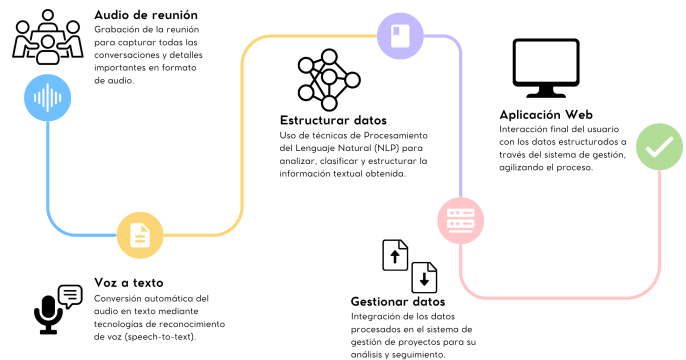


Figura 2: Diagrama general.

Este enfoque integral es esencial para superar los desafíos actuales en la administración de tareas dentro de los equipos, aprovechando las tecnologías avanzadas para transformar las reuniones en acciones concretas y mejorar la coordinación del equipo.

III. JUSTIFICACIÓN

Este proyecto provee un método eficiente para conectar y facilitar el proceso de conversión y síntesis de la información de las reuniones en entornos laborales, teniendo en cuenta la creciente necesidad de optimizar y automatizar la asignación de tareas en el marco de desarrollo de un proyecto. Actualmente, los sistemas de gestión de proyectos tradicionales enfrentan limitaciones significativas, especialmente en lo que respecta a la documentación y la asignación de tareas. Las metodologías convencionales, aunque efectivas en ciertos aspectos, a menudo requieren una integración manual que puede resultar en ineficiencias y errores (Rehman & Hussain, 2007). Este trabajo aborda estas deficiencias mediante la implementación de una solución innovadora que combina transcripción automática de

reuniones y modelos de lenguaje avanzados para la asignación de tareas.

La integración de tecnologías emergentes en la gestión de proyectos ofrece una ventaja sustancial en términos de eficiencia y precisión. Un ejemplo relevante que encontramos en la literatura es la implementación de Miao et al. (2020), con la cual quedó demostrado el potencial y la ventaja que ofrece el uso del reconocimiento de voz en audioconferencias corporativas, a través del cual se realizaban transcripciones en tiempo real, se identificaban a los oradores y en post-procesado se compilaban los resultados en un acta que se cargaba en una base de datos de gestión de reuniones. Además, el uso de modelos de lenguaje avanzados, como el explicado en el trabajo de Touvron et al. (2023) sobre Llama 2, permitirá optimizar la asignación de tareas al adaptar la asignación a las necesidades específicas del equipo y el contexto del proyecto, proporcionando una base robusta para trabajar en la mejora de la gestión de tareas de un proyecto.

El impacto de este software se manifiesta en su capacidad para automatizar y agilizar la gestión de tareas, proporcionando una administración más fluida y adaptada a las necesidades del equipo. Mientras que los métodos tradicionales de gestión, incluidos los enfoques ágiles como Scrum, han demostrado ser efectivos (Adi, 2015), la combinación de transcripción automática y modelos de lenguaje ofrece una administración más precisa y en tiempo real. Este enfoque no solo cierra la brecha existente entre la documentación y la asignación de tareas, sino que también mejora la organización y el seguimiento de proyectos, lo que tiene implicaciones significativas para la productividad y coordinación de equipos en diversos sectores.

Desde una perspectiva teórica, este proyecto aporta una contribución importante al llenar vacíos en el conocimiento sobre la aplicación de tecnologías emergentes en la gestión de proyectos. La propuesta ofrece una solución innovadora y eficiente, combinando transcripción automática y asignación de tareas, y estableciendo un nuevo estándar para la gestión de proyectos colaborativos.

IV. OBJETIVOS

IV-A. *Objetivo General*

Modelar, diseñar e implementar un módulo de gestor de tareas para proyectos que integre la transcripción automática de reuniones y la asignación de tareas utilizando modelos de lenguaje natural, con el fin de optimizar la gestión y el seguimiento de proyectos en entornos colaborativos.

IV-B. *Objetivos Específicos*

1. Elaborar la revisión sistemática de la literatura relacionada con gestión de tareas, speech-to-text, grandes modelos de lenguaje, procesamiento de lenguaje natural y estructuración de datos.
2. Desarrollar un módulo de transcripción que capture y convierta las conversaciones de las reuniones en texto, utilizando modelos avanzados de reconocimiento de voz.
3. Implementar un prototipo de software que sugiera la creación y asignación de tareas de acuerdo a las decisiones y acuerdos tomados durante las reuniones, basado en un sistema de análisis de lenguaje natural usando LLM (modelo extenso de lenguaje).
4. Validar el prototipo del gestor, la coherencia y el impacto en la eficiencia de la gestión de tareas.

V. METODOLOGÍA DE INVESTIGACIÓN

1. Fase de requerimientos e investigación:
En esta primera fase se define el problema principal del proyecto, que es automatizar la conversión de reuniones en tareas concretas, en relación a este se identifican las palabras clave acorde al contexto del proyecto. Además, se realiza una revisión exhaustiva y sistemática de la literatura y se analizan soluciones existentes para establecer los requerimientos funcionales y no funcionales.
2. Fase de diseño:
En esta fase se diseña una solución basada en los requerimientos definidos, incluyendo la interfaz de usuario y la estructura del sistema. Se determinan los componentes necesarios para realizar de forma eficiente las transcripciones y gestionar de manera adecuada la asignación de tareas. Durante esta etapa, se definen los frameworks y los servicios de infraestructura a utilizar en el proyecto.
3. Fase de desarrollo e implementación:
En esta fase se desarrolla el prototipo del software siguiendo una metodología ágil. Se implementan los módulos clave para garantizar el funcionamiento correcto de las transcripciones y del post-procesado de la información para establecer tareas correctamente. Asimismo, se mejora el prototipo a lo largo de ciclos de trabajo para asegurar la funcionalidad y usabilidad del gestor.
4. Fase de despliegue y pruebas:
Por último, en esta fase se despliega el aplicativo para validarlo y realizar pruebas exhaustivas que aseguren el correcto funcionamiento del prototipo y cada uno de sus módulos, ajustados al alcance establecido inicialmente.

VI. METODOLOGÍA DE DESARROLLO

La metodología de desarrollo para este proyecto se basa en Scrum, estructurada en cuatro sprints, los cuales son los

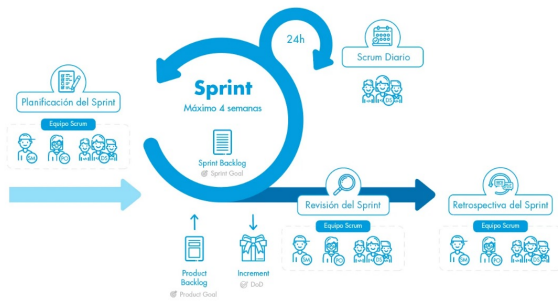


Figura 3: Metodología de referencia - *Scrum*.

siguientes:

Sprint 1: Se define el problema y se establecen los requerimientos funcionales y no funcionales, así como los diagramas de arquitectura y la interfaz de usuario básica. Se genera un documento de especificación y un backlog del proyecto. Duración estimada: 2 semanas.

Sprint 2: Se desarrolla e integra el módulo de transcripción automática, realizando pruebas iniciales para evaluar su precisión. El resultado esperado es un módulo funcional y un informe de pruebas. Duración estimada: 4 semanas.

Sprint 3: Se implementa el módulo de asignación de tareas mediante procesamiento de lenguaje natural y se realizan pruebas de integración. Se busca obtener un flujo de trabajo completo que vincule transcripción y asignación de tareas. Duración estimada: 4 semanas.

Sprint 4: Se lleva a cabo la integración final del sistema, pruebas funcionales y de usuario, así como la mejora de rendimiento y la solución de errores. Al final, se produce un prototipo completamente funcional y se prepara la documentación para su entrega. Duración estimada: 2 semanas.

VII. MARCO TEÓRICO

El reconocimiento de voz es una tecnología esencial para la automatización en la administración de tareas, especialmente en el contexto de reuniones y conferencias. La capacidad de transcribir automáticamente conversaciones y asignar tareas basadas en la identificación de hablantes permite que las discusiones se conviertan en acciones sin la intervención manual, lo que minimiza los errores y acelera el flujo de trabajo. Miao et al. (2020) presentan un sistema integrado de transcripción de voz y reconocimiento de hablantes que permite extraer información clave de las reuniones. Este tipo de soluciones optimiza la captura de datos al generar transcripciones en tiempo real y distinguir entre diferentes participantes.

Por otro lado, estudios como el de Maruri et al. (2021)

han profundizado en cómo los cambios contextuales en la voz pueden influir en las interacciones de reuniones remotas. El análisis de estos cambios es crucial para mejorar la precisión en la interpretación de la intención del hablante y, por ende, en la asignación de tareas. Estos trabajos subrayan la importancia de contar con sistemas que no solo reconozcan el contenido verbal, sino que también identifiquen los matices del discurso para facilitar la colaboración y la comunicación en entornos distribuidos.

Trabelsi et al. (2022) también realizaron una evaluación de motores avanzados de reconocimiento de voz, mostrando que los algoritmos más eficientes son capaces de manejar una variedad de lenguajes y contextos, lo cual es relevante para proyectos que operan en ambientes multilingües o con variabilidad en la calidad de audio. Este enfoque es fundamental para asegurar que el prototipo desarrollado sea lo suficientemente robusto como para manejar las dinámicas propias de equipos diversos.

El análisis de texto, especialmente a través de técnicas de procesamiento de lenguaje natural (PLN), desempeña un papel crucial en la automatización de tareas. La integración de modelos de lenguaje avanzados, como Llama 2 y GPT, permite interpretar conversaciones con precisión y convertirlas automáticamente en acciones concretas. Estos modelos destacan por su capacidad de procesar grandes volúmenes de datos textuales y generar salidas coherentes, lo que resulta particularmente útil para extraer información estructurada de discusiones en reuniones. De manera similar, Brown et al. (2020) analizan cómo los modelos de lenguaje más recientes pueden comprender y ejecutar tareas basadas en instrucciones mínimas, lo que es fundamental para optimizar la administración de proyectos de manera automatizada.

Además, se destaca la importancia de utilizar PLN para crear automáticamente tareas a partir de entradas de voz (Schmidberger, 2020). Estos sistemas no solo transcriben el contenido hablado, sino que lo analizan en profundidad para extraer patrones y generar problemas, acciones o recordatorios de manera automática. En un contexto más amplio, Mincheva et al. (2021) investigan cómo el PLN puede extraer datos estructurados de textos no estructurados, lo que es particularmente relevante para convertir actas de reuniones o discusiones informales en acciones concretas y tareas organizadas.

Jiang (2012) también discute técnicas de extracción de información, subrayando su utilidad en la gestión de proyectos al permitir que grandes volúmenes de texto sean analizados automáticamente para detectar información clave. Esta capacidad resulta esencial en el contexto de reuniones frecuentes, donde es importante detectar temas recurrentes y asignar tareas automáticamente basadas en lo discutido.

La gestión de proyectos ha evolucionado significativamente con la introducción de metodologías ágiles, las cuales pro-

mueven ciclos cortos de trabajo, flexibilidad en los procesos y una alta participación del equipo. Scrum es una de las metodologías más utilizadas y ha demostrado ser eficaz para administrar proyectos de software de manera iterativa, permitiendo ajustes continuos y mejorando la productividad (Adi, 2015; Schwaber & Sutherland, 2020). La implementación de Scrum en la administración de tareas implica la creación de backlogs, la asignación dinámica de tareas y reuniones de seguimiento, lo que permite que el equipo esté constantemente alineado con los objetivos del proyecto.

Los trabajos de Rehman y Hussain (2007) comparan diversas metodologías y marcos de gestión de proyectos, concluyendo que las metodologías ágiles, en particular Scrum, ofrecen mayor adaptabilidad en entornos complejos. En este sentido, la integración de tecnologías como el reconocimiento de voz y el PLN en sistemas de gestión de proyectos contribuye a mejorar la asignación de tareas y reducir la carga administrativa.

Además, Pellerin et al. (2013) señalan que el uso de software especializado para la gestión de proyectos puede influir directamente en el rendimiento de los mismos, mejorando tanto la eficiencia como la coordinación. Esto refuerza la idea de que la automatización mediante herramientas avanzadas no solo es deseable, sino que puede ser un diferenciador clave en proyectos modernos.

Existen varios proyectos y soluciones tecnológicas que abordan desafíos similares al de este proyecto. El estudio de Ramin et al. (2020) discute cómo la gestión de proyectos modernos va más allá del código y abarca otros aspectos críticos, como la colaboración, la comunicación y la toma de decisiones basadas en datos. Este enfoque holístico se alinea con el objetivo de este proyecto de integrar datos y conversaciones en tiempo real, transformándolos en acciones concretas mediante la automatización.

Otra referencia clave es el trabajo de Sahraeian y Van Compernelle (2017), quienes analizan el reconocimiento de voz multilingüe. Este tipo de tecnología es esencial para proyectos que requieren operar en contextos globales y manejar equipos que hablen diferentes idiomas. De manera similar, Marcolin et al. (2021) proponen un marco para mejorar la toma de decisiones mediante el análisis de texto, utilizando datos textuales para mejorar la comprensión de discusiones y facilitar la asignación de tareas. Este enfoque basado en datos se relaciona directamente con el objetivo del proyecto propuesto, ya que optimiza la administración de tareas y mejora la coordinación.

Finalmente, Fanni et al. (2023) describen cómo el PLN ha sido utilizado para estructurar informes médicos, lo que subraya la aplicabilidad de estas tecnologías en diversos campos, incluyendo la gestión de proyectos. Estos proyectos similares proporcionan valiosas lecciones sobre cómo integrar PLN y reconocimiento de voz para transformar conversaciones en

acciones, lo cual es el núcleo del presente proyecto.

VIII. MARCO CONCEPTUAL

Gestión de proyectos:

La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo. Se logra mediante la aplicación e integración adecuadas de los 42 procesos de la dirección de proyectos, agrupados lógicamente, que conforman los 5 grupos de procesos. Estos 5 grupos de procesos son: Iniciación, Planificación, Ejecución, Seguimiento y Control, Cierre (Project Management Institute [PMI], 2021).

Metodología ágil:

Según Abrahamsson et al. (2017) metodología ágil (que hace referencia a "la cualidad de ser ágil, estar listo para el movimiento, agilidad, actividad, destreza en el movimiento") es un método de desarrollo de software que intenta ofrecer una respuesta a la comunidad empresarial, que pide procesos de desarrollo de software más ligeros, rápidos y ágiles. Esto es especialmente cierto en el caso de la volátil y creciente industria del software de Internet, así como en el emergente entorno de las aplicaciones móviles.

Speech-to-text:

La conversión de discurso a texto, según Trivedi et al. (2018), es el proceso de convertir palabras pronunciadas en textos escritos. Este proceso es sinónimo de reconocimiento de voz, aunque el término 'reconocimiento de voz' se utiliza para describir un proceso más amplio de comprensión del habla. El STT sigue los mismos principios y pasos que el reconocimiento de voz, aplicando diferentes combinaciones de técnicas en cada etapa.

Grandes modelos de lenguaje (LLM):

Los grandes modelos de lenguaje se definen como sistemas de inteligencia artificial entrenados con un vasto corpus de datos, que incluye miles de millones de palabras provenientes de artículos, libros y otros contenidos disponibles en línea (Thirunavukarasu et al., 2023). Recientemente, Tang et al. (2023) han observado un creciente interés en los LLM, impulsado por su destacado rendimiento y su capacidad para comprender el lenguaje natural. En consecuencia, los investigadores han comenzado a investigar el potencial de estas herramientas para asistir a expertos en diversas áreas, entre estas la gestión de proyectos.

Procesamiento de lenguaje natural (NLP):

Según la definición de Liddy (2001), el procesamiento de lenguaje natural se define como un enfoque computacional para el análisis de textos, fundamentado en un conjunto de teorías y tecnologías. Como señalan Brown et al. (2020), el enfoque inicial del NLP se centraba en el aprendizaje de representaciones y el diseño de arquitecturas orientadas a tareas específicas. No obstante, este paradigma ha sido sustituido por un enfoque más generalizado que prioriza el pre-entrenamiento y utiliza arquitecturas no específicas para tareas particulares. Esta evolución ha llevado a avances significativos en diversas problemáticas que enfrentaba el NLP, tales como la comprensión de lectura y la respuesta a preguntas. A pesar de que el enfoque inicial es agnóstico respecto a las tareas, es imprescindible realizar un posterior refinamiento utilizando un conjunto de datos, lo que permite al modelo adaptarse y ejecutar la tarea deseada.

Datos no estructurados:

Los datos no estructurados son información que no está organizada según un modelo o esquema de datos preestablecido y, por lo tanto, no se puede almacenar en una base de datos relacional tradicional o RDBMS. El texto y los archivos multimedia son dos tipos comunes de contenido no estructurado. Muchos documentos comerciales no están estructurados, al igual que los mensajes de correo electrónico, los videos, las fotos, las páginas web y los archivos de audio (MongoDB).

Datos estructurados:

Los datos estructurados tienen un formato estandarizado que permite tanto al software como a las personas acceder a estos de forma eficaz. Por lo general, se trata de datos tabulares con filas y columnas que definen claramente sus atributos. Las computadoras pueden procesar eficazmente los datos estructurados en busca de información dado que se trata de información cuantitativa (AWS).

IX. ARQUITECTURA LÓGICA DE LA SOLUCIÓN

La arquitectura lógica del sistema de gestión de proyectos está diseñada para procesar archivos de audio en formatos .mp3 y .wav, convirtiéndolos en transcripciones mediante un servicio de **Speech to Text**. Estas transcripciones son analizadas por un módulo de **Procesamiento de Lenguaje Natural** para identificar, actualizar o generar nuevas tareas en el sistema, basándose en el contexto del proyecto. La interfaz web permite a los usuarios interactuar con el sistema, mientras que una **API** gestiona la comunicación entre el **frontend** y los servicios de **backend**, incluyendo operaciones **CRUD** y la interacción con la base de datos, donde se almacenan todas las tareas y asignaciones de los proyectos.

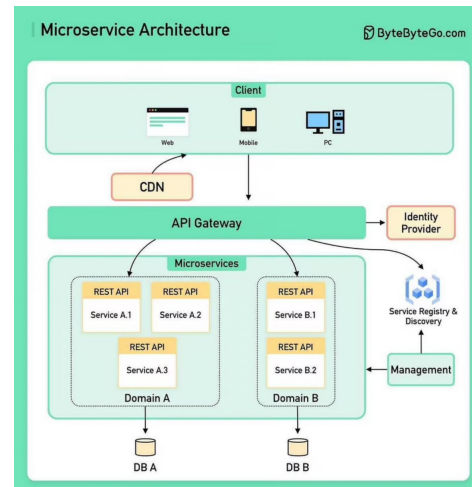


Figura 4: Arquitectura lógica de referencia.

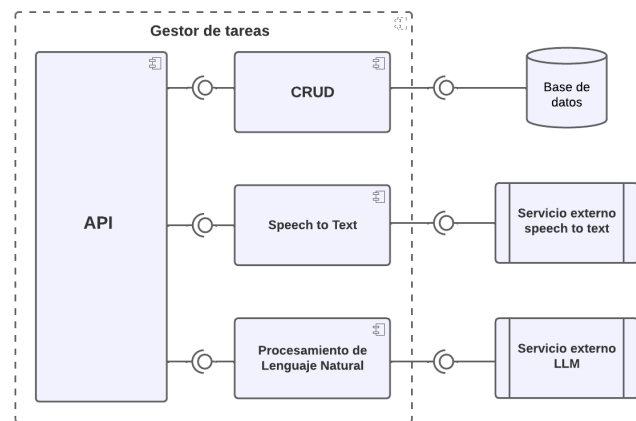


Figura 5: Arquitectura lógica de la solución.

X. ARQUITECTURA FÍSICA DE LA SOLUCIÓN

La arquitectura física del sistema de gestión de proyectos se basa en una API desarrollada con **FastAPI** en Python, que gestiona la lógica de negocio y proporciona puntos de acceso para interactuar con el sistema, permitiendo realizar operaciones **CRUD** y conectarse a una base de datos **MySQL**, donde se almacenan las tareas y asignaciones de los proyectos. Los archivos de audio en formatos .mp3 y .wav se almacenan inicialmente en un **Cloud Storage Bucket** de Google, desde donde se procesan mediante dos servicios de transcripción de voz a texto: **Google Cloud Speech-to-Text v2** y **Deepgram**, que generan versiones independientes de las transcripciones. Estas transcripciones se envían a un modelo de lenguaje grande (**Gemini**) para unificarlas, creando una única versión que combina lo mejor de ambas y facilita la estructuración de las conversaciones. Finalmente, la transcripción unificada es procesada por **Google Vertex AI Studio**, que analiza y transforma el contenido en datos estructurados, identificando y definiendo las tareas del proyecto, las cuales se almacenan

en la base de datos. La interacción con los usuarios ocurre a través de una interfaz web conectada a la API, que permite gestionar tareas, consultar información y actualizar el estado de los proyectos.

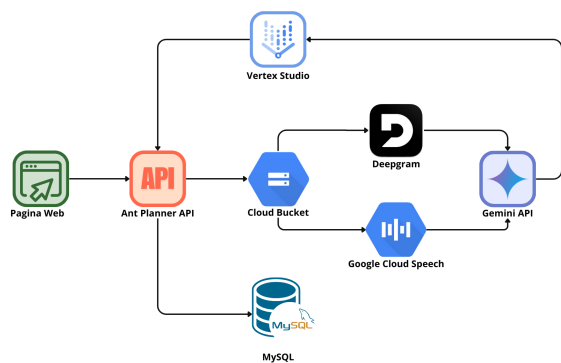


Figura 6: Arquitectura física de la solución.

XI. PROTOTIPO

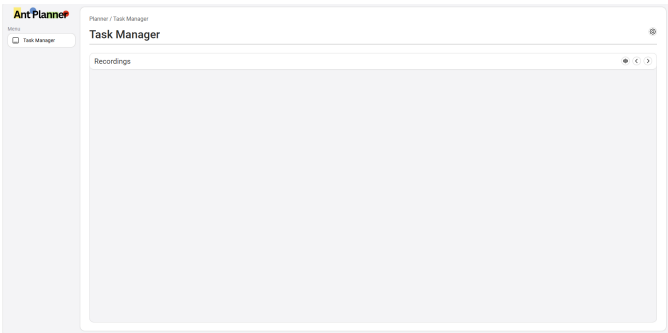


Figura 7: Gestor de tareas - pantalla inicial.

En esta pantalla se evidenciarán las tareas reconocidas en el audio cargado.

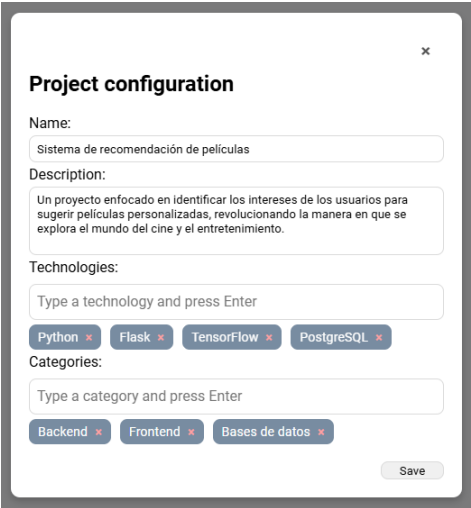


Figura 8: Gestor de tareas - configuración de proyecto.

En este popup se ingresará la información relevante del proyecto.

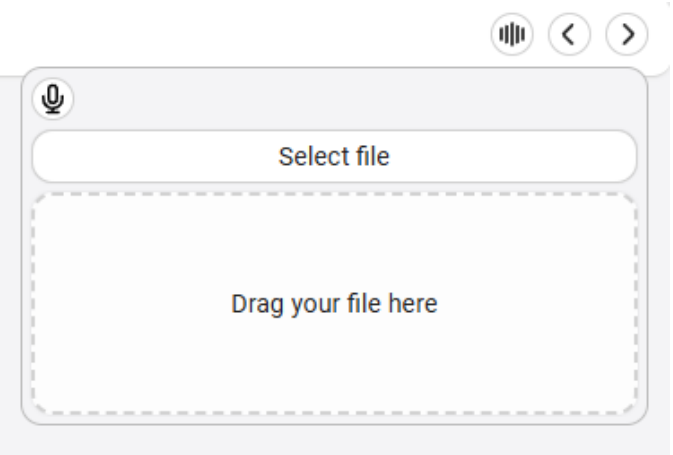


Figura 9: Gestor de tareas - cargar/grabar audio.

En este botón se cargará el archivo del audio en formato .wav o se podrá grabar directamente la reunión presionando el micrófono.

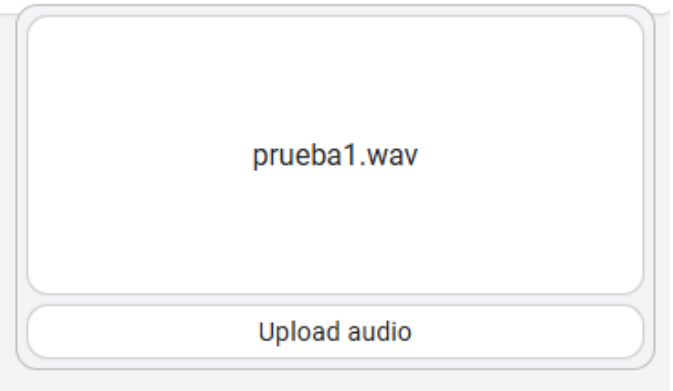


Figura 10: Gestor de tareas - subir audio.

Posteriormente se subirá el audio al bucket para que se almacene y luego se pueda procesar mediante inteligencia artificial.

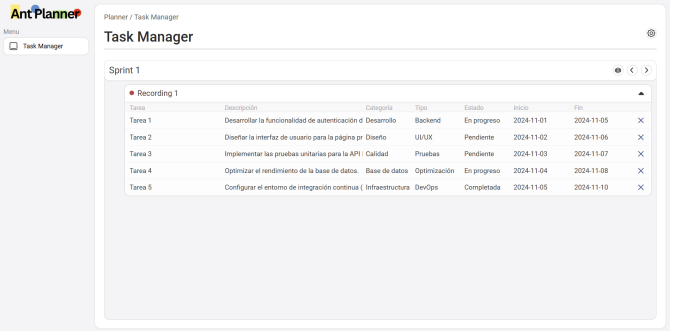


Figura 11: Gestor de tareas - lista de tareas.

Como resultado se tendrá el listado de tareas extraídas del audio, las cuales se encuentran asociadas al sprint en el cual fue cargada o grabada la reunión, además se muestra la información relacionada a la tarea, con la posibilidad de modificar y corregir cualquier dato.

XII. TABLA DE EVALUACIÓN

Característica	Definición o descripción	AD
Understandability	¿Fácil de comprender?	5
Documentation	¿Documentación de usuario completa, apropiada y bien estructurada?	5
Buildability	¿Fácil de construir en un sistema compatible? (Close-Open)	4
Installability	¿Fácil de instalar en un sistema compatible?	
Learnability	¿Fácil de aprender a usar sus funciones?	5
Identity	¿La identidad del proyecto / software es clara y única?	5
Copyright	¿Es fácil ver quién posee el proyecto / software?	
Licensing	Adopción de la licencia apropiada?	
Governance	¿Fácil de entender cómo se ejecuta el proyecto y cómo se gestiona el desarrollo del software?	
Community	¿Evidencia de comunidad actual / futura?	5
Accessibility	¿Evidencia de capacidad de descarga actual / futura?	
Testability	¿Fácil de probar la corrección de funciones caja negra?	
Portability	¿Utilizable en múltiples plataformas?	
Supportability	¿Evidencia de soporte para desarrolladores actuales / futuros?	5
Analysability	¿Fácil de entender a nivel fuente?	5
Changeability	¿Fácil de modificar y aportar cambios a los desarrolladores?	5
Evolvability	¿Evidencia de desarrollo actual / futuro?	4
Interoperability	¿Interoperable con otro software requerido / relaciona	

Figura 12: Tabla de evaluación del prototipo.

XIII. CONCLUSIÓN

Este proyecto logró integrar tecnologías avanzadas para transformar la gestión de proyectos, facilitando la conversión de información no estructurada, como grabaciones de audio, en tareas organizadas y listas para su seguimiento. El módulo desarrollado automatizó procesos clave, desde la carga y transcripción de audios hasta la asignación estructurada de tareas, superando las limitaciones de los sistemas tradicionales al minimizar la intervención manual y los posibles errores asociados.

Mediante la combinación de herramientas como Google Cloud Speech V2 y Deepgram, se obtuvo una transcripción precisa del contenido de las reuniones, mientras que la implementación de Gemini permitió generar una versión fiel a la conversación real. Posteriormente, el uso de Vertex AI, entrenado con ejemplos específicos, estructuró estas transcripciones en tareas categorizadas que se almacenaron en una base de datos MySQL, lista para su visualización y gestión en el front-end mediante un CRUD. Este proceso demostró ser altamente eficiente, proporcionando al equipo la capacidad de ajustar y personalizar la información generada según las necesidades del proyecto.

El sistema desarrollado no solo optimizó la documentación de reuniones, sino que también automatizó la asignación de roles y tareas, estableciendo una conexión directa entre las discusiones del equipo y las acciones concretas a realizar. Este enfoque resolvió desafíos comunes en la organización y el seguimiento de proyectos, incrementando la productividad y mejorando la coordinación entre los miembros del equipo.

La experiencia adquirida durante el desarrollo confirmó la capacidad de las tecnologías emergentes para transformar procesos tradicionales, aportando soluciones innovadoras y prácticas. Este proyecto demostró que es posible automatizar la gestión de proyectos de manera precisa y adaptada al contexto de cada equipo, sentando las bases para futuras integraciones con herramientas más amplias, como gestores de proyectos.

En resumen, el trabajo realizado estableció un nuevo estándar en la optimización de flujos de trabajo, maximizando la eficiencia operativa y ofreciendo una solución que no solo cumplió con los objetivos planteados, sino que también abrió posibilidades para futuras aplicaciones en entornos laborales modernos.

REFERENCIAS

- [1] Miao, K., Biermann, O., Miao, Z., Leung, S., Wang, J., & Gai, K. (2020). Integrated parallel system for audio conferencing voice transcription and speaker identification. 2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS), 1-8. <https://doi.org/10.1109/HPBDIS49115.2020.9130598>
- [2] Maruri, H.A.C., Aslan, S., Stemmer, G., Alyuz, N., Nachman, L. (2021) Analysis of Contextual Voice Changes in Remote Meetings. Proc. Interspeech 2021, 2521-2525, [doi: 10.21437/Interspeech.2021-1932](https://doi.org/10.21437/Interspeech.2021-1932)
- [3] Adi, P. (2015). Scrum Method Implementation in a Software Development Project Management. International Journal Of Advanced Computer Science And Applications, 6(9). <https://doi.org/10.14569/ijacsa.2015.060927>
- [4] Ken Schwaber & Jeff Sutherland (2020.)The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game. <https://scrumguides.org/scrum-guide.html>
- [5] PMBOK® Guide (Seventh Edition). (2021). Project Management Institute (PMI). <https://www.pmi.org/standards/pmbok>
- [6] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., . . . Scialom, T. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv.org. <https://arxiv.org/abs/2307.09288#>
- [7] Schmidberger, F. (2020). Automated issue creation using voice recognition and natural language processing (Bachelor's thesis). <https://d-nb.info/123727074X/34>
- [8] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020, 28 mayo). Language Models are Few-Shot Learners. arXiv.org. <https://arxiv.org/abs/2005.14165>
- [9] Mincheva, Z., Vasilev, N., Nikolov, V., & Antonov, A. (2021). Extracting Structured Data from Text in Natural Language. International Journal Of Intelligent Information Systems, 10(4), 74. <https://doi.org/10.11648/j.ijis.20211004.16>
- [10] Sahraeian, R., & Van Compernelle, D. (2017). Crosslingual and Multilingual Speech Recognition Based on the Speech Manifold. IEEE/ACM Transactions On Audio Speech And Language Processing, 25(12), 2301-2312. <https://doi.org/10.1109/taslp.2017.2751747>
- [11] Ramin, F., Matthies, C., & Teusner, R. (2020). More than Code. Association For Computing Machinery. <https://doi.org/10.1145/3387940.3392241>
- [12] Rehman, A. U., & Hussain, R. (2007). Software Project Management Methodologies/Frameworks Dynamics «A Comparative Approach».

2007 International Conference On Information And Emerging Technologies. <https://doi.org/10.1109/iciet.2007.4381330>

- [13] Marcolin, C. B., Becker, J. L., Wild, F., Behr, A., & Schiavi, G. (2021). Listening to the voice of the guest: A framework to improve decision-making processes with text data. *International Journal Of Hospitality Management*, 94, 102853. <https://doi.org/10.1016/j.ijhm.2020.102853>
- [14] Benzeghiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes, T., Jouvét, D., Fissore, L., Laface, P., Mertins, A., Ris, C., Rose, R., Tyagi, V., & Wellekens, C. (2007). Automatic speech recognition and speech variability: A review. *Speech Communication*, 49(10-11), 763-786. <https://doi.org/10.1016/j.specom.2007.02.006>
- [15] Jiang, J. (2012). Information Extraction from Text. En Springer eBooks (pp. 11-41). https://doi.org/10.1007/978-1-4614-3223-4_2
- [16] Uchida, S. (2024). Using early LLMs for corpus linguistics: Examining ChatGPT's potential and limitations. *Applied Corpus Linguistics*, 100089. <https://doi.org/10.1016/j.acorp.2024.100089>
- [17] Preston, S., Wei, M., Rao, R., Tinn, R., Usuyama, N., Lucas, M., Gu, Y., Weerasinghe, R., Lee, S., Piening, B., Tittel, P., Valluri, N., Naumann, T., Bifulco, C., & Poon, H. (2023). Toward structuring real-world data: Deep learning for extracting oncology information from clinical text with patient-level supervision. *Patterns*, 4(4), 100726. <https://doi.org/10.1016/j.patter.2023.100726>
- [18] Trabelsi, A., Warichet, S., Aajaoun, Y., & Soussilane, S. (2022b). Evaluation of the efficiency of state-of-the-art Speech Recognition engines. *Procedia Computer Science*, 207, 2242-2252. <https://doi.org/10.1016/j.procs.2022.09.534>
- [19] Pellerin, R., Perrier, N., Guillot, X., & Léger, P. (2013). Project Management Software Utilization and Project Performance. *Procedia Technology*, 9, 857-866. <https://doi.org/10.1016/j.protcy.2013.12.095>
- [20] Qiu, Q., Tian, M., Tao, L., Xie, Z., & Ma, K. (2024). Semantic information extraction and search of mineral exploration data using text mining and deep learning methods. *Ore Geology Reviews*, 165, 105863. <https://doi.org/10.1016/j.oregeorev.2023.105863>
- [21] Fanni, S. C., Romei, C., Ferrando, G., Volpi, F., D'Amore, C. A., Bedini, C., Ubbiali, S., Valentino, S., & Neri, E. (2023). Natural language processing to convert unstructured COVID-19 chest-CT reports into structured reports. *European Journal Of Radiology Open*, 11, 100512. <https://doi.org/10.1016/j.ejro.2023.100512>
- [22] Li, Y., Liu, Y., Zhang, J., Cao, L., & Wang, Q. (2024). Automated analysis and assignment of maintenance work orders using natural language processing. *Automation In Construction*, 165, 105501. <https://doi.org/10.1016/j.autcon.2024.105501>
- [23] Ghadage, Y. H., & Shelke, S. D. (2016). Speech to text conversion for multilingual languages. *IEEE Explore*. <https://doi.org/10.1109/iccsp.2016.7754130>
- [24] Verma, S., Jain, K., & Prakash, C. (2020). An Unstructured to Structured Data Conversion using Machine Learning Algorithm in Internet of Things (IoT). *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3563389>
- [25] Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017, 25 septembre). Agile Software Development Methods: Review and Analysis. *arXiv.org*. <https://arxiv.org/abs/1709.08439>
- [26] Tang, R., Han, X., Jiang, X., & Hu, X. (2023, 8 marzo). Does Synthetic Data Generation of LLMs Help Clinical Text Mining? *arXiv.org* <https://arxiv.org/abs/2303.04360>
- [27] Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., & Ting, D. S. W. (2023). Large language models in medicine. *Nature Medicine*, 29(8), 1930-1940. <https://doi.org/10.1038/s41591-023-02448-8>
- [28] Liddy, E.D. 2001. Natural Language Processing. In *Encyclopedia of Library and Information Science*, 2nd Ed. NY: Marcel Dekker, Inc. <https://surface.syr.edu/istpub/63/>
- [29] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020, 28 mayo). Language Models are Few-Shot Learners. *arXiv.org*. <https://arxiv.org/abs/2005.14165>
- [30] Trivedi, A., Pant, N., Shah, P., Sonik, S., & Agrawal, S. (2018). Speech to text and text to speech recognition systems: A review. *IOSR Journal of Computer Engineering*, 20(2), 36-43. <https://paperswithcode.com/paper/speech-to-text-and-text-to-speech-recognition>
- [31] MongoDB. (s.f.). What is unstructured data? <https://www.mongodb.com/resources/basics/unstructured-data>
- [32] What is Structured Data? - Structured Data Explained - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/what-is/structured-data/>