# ATHENUS: Un Chatbot Empresarial con Procesamiento de Lenguaje Natural para la Gestión de Documentos Internos

Kevin Fabio Ramos López, Juan David Ramos López, Juan Sebastian Vargas Castañeda, Julián Steven Vega Daza
Universidad Nacional de Colombia, Sede Bogotá

**En el entorno empresarial, la gestión eficiente de la documentación interna es fundamental para la toma de decisiones y el cumplimiento de procesos. Sin embargo, el acceso y consulta de estos documentos puede ser complejo, especialmente en organizaciones con múltiples departamentos y niveles de restricción. Para abordar este desafío, presentamos Athenus, un chatbot empresarial basado en inteligencia artificial que permite a los empleados consultar información de documentos internos de manera segura y eficiente.**

*Index Terms*—**Chatbot, Procesamiento de Lenguaje Natural, Seguridad de Datos, Inteligencia Artificial, Aplicaciones Empresariales.**

## I. INTRODUCCIÓN

EN las organizaciones modernas, la documentación interna juega un papel clave en la gestión eficiente de procesos, cumplimiento normativo y toma de decisiones estratégicas [1]. Sin embargo, el acceso y consulta de estos documentos pueden volverse tareas complejas y poco eficientes, especialmente en empresas con múltiples departamentos y niveles jerárquicos. La información suele estar fragmentada en distintos sistemas, dificultando su disponibilidad inmediata y segura para los empleados que la requieren.

El problema se agrava cuando no existen mecanismos adecuados para garantizar que cada usuario acceda solo a la información que necesita según su rol. Esto puede generar ineficiencias, pérdida de tiempo en búsquedas manuales e incluso riesgos de seguridad en la exposición de datos sensibles [2].

### A. Motivación para el desarrollo de Athenus

Para abordar estos desafíos, surge la necesidad de un sistema inteligente que permita a los empleados consultar información relevante de manera rápida y segura, respetando las restricciones de acceso según el nivel y departamento de cada usuario. Athenus nace como una solución innovadora basada en inteligencia artificial, diseñada para optimizar la consulta de documentos internos dentro de las empresas.

A diferencia de los chatbots genéricos, Athenus no solo responde preguntas generales, sino que está especializado en la documentación empresarial, con un modelo de acceso restringido que protege la información sensible y garantiza que cada usuario consulte únicamente los documentos autorizados según su rol y departamento.

### B. Objetivos del proyecto

El desarrollo de Athenus tiene como objetivos principales:

- Facilitar la consulta de documentos internos mediante un chatbot basado en inteligencia artificial.
- Implementar un sistema de niveles de acceso que garantice la seguridad y confidencialidad de la información.
- Mejorar la eficiencia en la gestión documental dentro de las empresas, reduciendo el tiempo de búsqueda de información.
- Ofrecer una solución escalable y adaptable a diferentes organizaciones y estructuras empresariales.

### C. Descripción del funcionamiento

Athenus es una aplicación móvil desarrollada en Kotlin que se conecta a un backend encargado de la autenticación de usuarios, gestión de empresas y administración de permisos de acceso. El chatbot utiliza un modelo de lenguaje especializado para responder consultas sobre documentos internos, asegurando que solo los usuarios con los permisos adecuados puedan acceder a cada nivel de información.

El sistema implementa una jerarquía de accesos por departamento, donde cada usuario tiene un nivel determinado que define qué documentos puede consultar. Por ejemplo, un analista de tecnología con nivel 1 podrá acceder solo a documentos de su área en ese nivel, mientras que un CTO con nivel 7 tendrá acceso a todos los documentos de su departamento hasta dicho nivel.

Con esta solución, Athenus mejora la productividad empresarial, optimiza la gestión del conocimiento y fortalece la seguridad de la información, proporcionando una herramienta eficaz para la administración de documentos internos.

## II. TRABAJOS RELACIONADOS

### A. Tecnologías similares

El uso de modelos de lenguaje natural (LLMs, por sus siglas en inglés) ha revolucionado la forma en que las empresas automatizan la gestión del conocimiento. Tecnologías como ChatGPT (OpenAI), Claude (Anthropic), BERT (Google) y DeepSeek han demostrado su capacidad para procesar y generar texto de manera coherente, lo que permite su aplicación en chatbots empresariales.

Varias plataformas comerciales han incorporado estos modelos en sus soluciones, entre ellas:

- **IBM Watson Assistant:** Utiliza inteligencia artificial para responder preguntas y automatizar procesos, con integración en múltiples plataformas empresariales [3].
- **Dialogflow (Google Cloud AI):** Permite crear asistentes conversacionales personalizados, aunque su enfoque es generalista y no está diseñado específicamente para la gestión documental empresarial [4].
- **Azure OpenAI Service:** Ofrece acceso a modelos como GPT-4 para desarrollar chatbots y asistentes virtuales en entornos corporativos [5].
- **Amazon Bedrock:** Proporciona infraestructura para implementar modelos de lenguaje en aplicaciones personalizadas dentro del ecosistema de AWS [6].

Aunque estos modelos y plataformas tienen aplicaciones empresariales, su implementación generalmente implica la gestión de datos en servidores externos, lo que puede generar preocupaciones en términos de **seguridad y privacidad de la información corporativa**.

### *B. Comparación con Soluciones Existentes*

Athenus se diferencia de las soluciones comerciales al especializarse en la **consulta segura de documentación interna dentro de una empresa**. Aunque utiliza un modelo de lenguaje similar a ChatGPT, Claude, BERT o DeepSeek, su enfoque es más **específico y controlado**, permitiendo:

- **Acceso basado en roles y niveles de autorización:** A diferencia de los chatbots generalistas, Athenus implementa un sistema de jerarquía de acceso. Un analista con nivel 1 solo podrá consultar documentos dentro de su departamento con esa restricción, mientras que un director tendrá un nivel más alto de acceso.
- **Procesamiento de documentos internos:** El modelo está ajustado para responder preguntas basadas en documentos internos de la empresa, evitando información irrelevante o respuestas fuera de contexto.
- **Infraestructura privada y controlada:** A diferencia de soluciones como ChatGPT o Claude, que dependen de servidores externos, Athenus permite a las empresas ejecutar el modelo en servidores privados, garantizando que la información nunca salga del entorno corporativo.
- **Seguridad y privacidad:** Mientras que plataformas como Watson o Dialogflow almacenan datos en la nube, Athenus asegura que cada empresa mantenga control total sobre sus datos, minimizando riesgos de filtración.

En resumen, Athenus aprovecha la potencia de los modelos de lenguaje avanzados, pero los adapta a un **entorno empresarial seguro y especializado**, garantizando que cada usuario acceda únicamente a la información que le corresponde, sin comprometer la privacidad ni la confidencialidad de los datos.

## III. Metodología

El desarrollo de Athenus sigue una arquitectura cliente-servidor que integra una aplicación móvil, un backend para la gestión de usuarios y empresas, y un modelo de lenguaje ajustado a las necesidades de cada organización. El sistema ha sido diseñado para garantizar **seguridad, escalabilidad y control de acceso a la información**.

### *A. Arquitectura del Sistema*

Athenus está compuesto por tres capas principales:

- **Frontend:** Aplicación móvil en Kotlin para Android.
- **Backend:** API basada en Django para la gestión de usuarios, autenticación y control de acceso.
- **Modelo de Lenguaje:** Implementación con Ollama, asegurando que las consultas a los documentos internos se realicen con acceso restringido.

Dependiendo de los requerimientos del cliente, el sistema puede desplegarse **en la nube** o en **infraestructura local**, permitiendo a cada empresa mantener control sobre su información.

#### *1) Frontend: Aplicación Móvil*

La aplicación móvil está desarrollada en **Kotlin** y proporciona una interfaz intuitiva para los empleados de la empresa. Entre sus principales características se encuentran:

- **Inicio de sesión seguro:** con autenticación basada en credenciales.
- **Interfaz de chat** donde los usuarios pueden realizar consultas sobre documentos internos.
- **Compatibilidad con diferentes dispositivos Android** para facilitar su adopción en la empresa.

#### *2) Backend: API y Gestión de Usuarios*

El backend ha sido desarrollado en **Django** y expone una **API REST** para la gestión de usuarios, autenticación y control de acceso. Sus principales funcionalidades incluyen:

- **Autenticación y autorización:** mediante credenciales de usuario.
- **Gestión de empresas y usuarios** con jerarquías de acceso.
- **Despliegue flexible:** permitiendo a los clientes optar por una instalación en la nube o en un servidor privado dentro de su infraestructura.

#### *3) Modelo de Lenguaje: Acceso a Documentos Internos*

El núcleo de Athenus es un **modelo de lenguaje basado en Ollama**, optimizado para procesar consultas sobre documentación interna de cada empresa. La seguridad en el acceso a la información se garantiza mediante:

- **Restricción por departamentos y niveles de acceso**.
- **Procesamiento local o en servidores privados**.
- **Indexación y preprocesamiento de documentos**.

## IV. Implementación

### *A. Desarrollo de la aplicación móvil.*

La aplicación móvil de Athenus ha sido desarrollada en **Kotlin**, con algunas partes en **Java** para mantener compatibilidad con ciertos módulos y bibliotecas. Su arquitectura sigue el patrón **MVVM (Model-View-ViewModel)** para mejorar la separación de responsabilidades y facilitar el mantenimiento.

#### *1) Flujo de la Aplicación*

El flujo de la aplicación consta de dos actividades principales:

- **Pantalla de Inicio de Sesión**
  - Valida las credenciales del usuario a través de la API REST del backend.

  - Implementa autenticación segura y persistencia de sesión.
  - Maneja errores de autenticación, como credenciales inválidas o cuentas inactivas.
- **Pantalla Principal del Chatbot**
  - Una vez validado el login, el usuario accede a la pantalla principal, donde puede interactuar con el chatbot.
  - La interfaz de chat permite enviar preguntas relacionadas con los documentos internos de la empresa.
  - El chatbot responde utilizando el modelo de lenguaje basado en **Ollama**, asegurando que solo se muestren respuestas según el nivel de acceso del usuario.
- **Módulo de Subscripción**
  - Desde la pantalla principal, los usuarios pueden acceder a una opción donde se muestran los diferentes planes de suscripción.
  - Esta sección está diseñada para administradores o usuarios con permisos especiales, quienes pueden gestionar la suscripción de la empresa y visualizar los beneficios de cada plan.

Este enfoque garantiza una experiencia de usuario fluida y segura, asegurando que cada empleado solo acceda a la información correspondiente según sus permisos dentro de la empresa.

### *B. Configuración del backend y seguridad de datos*

El backend de Athenus está desarrollado en **Django** utilizando **Python** como lenguaje principal. La API REST permite gestionar la autenticación de usuarios, el control de acceso y la comunicación con el modelo de lenguaje, asegurando la privacidad de los documentos internos de cada empresa.

#### *1) Arquitectura y Entidades*

El backend sigue una arquitectura modular y escalable, donde cada entidad representa un elemento clave dentro del sistema:

- **Companies**: Representa a las empresas registradas en la plataforma. Cada empresa tiene su propio espacio aislado de documentos y usuarios.
- **Employees**: Usuarios registrados dentro de una empresa con distintos niveles de acceso a la documentación interna.
- **LLMModels**: Instancias del modelo de lenguaje configuradas para cada empresa, asegurando que la información procesada sea exclusiva de la organización.
- **Messages**: Registra las interacciones entre los empleados y el chatbot, permitiendo auditoría y mejora del sistema.
- **User**: Modelo base para la gestión de autenticación y permisos de los empleados dentro del sistema.
- **API de Login**: Maneja la autenticación basada en tokens, proporcionando acceso mediante **TokenAuth**.

#### *2) Seguridad de Datos*

Dado que Athenus maneja documentación confidencial, se han implementado medidas de seguridad estrictas para garantizar la protección de la información:

- **Autenticación:** Uso de TokenAuth para hacer peticiones.
- **Control de Acceso Basado en Roles:** Cada usuario tiene niveles de acceso según su rol y departamento dentro de la empresa, asegurando que solo pueda consultar documentos autorizados.
- **Cifrado de Datos:** Se emplea **SHA256** para cifrar información sensible almacenada en la base de datos.
- **Comunicación Segura:** Todas las solicitudes entre la aplicación móvil y el backend utilizan **HTTPS** para evitar interceptación de datos.
- **Aislamiento de Datos por Empresa:** Cada empresa opera en un entorno segregado, impidiendo que usuarios de una organización accedan a información de otra.
- **Registro de Actividades:** Se mantiene un log de todas las interacciones con el chatbot para auditoría y seguridad.

#### *3) Opciones de Despliegue*

El backend de Athenus puede desplegarse en distintas infraestructuras según las necesidades de cada empresa:

- **Despliegue en la Nube:** Utilizando plataformas como AWS, Google Cloud o Azure para garantizar escalabilidad y alta disponibilidad.
- **Infraestructura Local:** Algunas empresas pueden optar por ejecutar el sistema en servidores privados dentro de sus instalaciones, asegurando un mayor control sobre los datos.

Gracias a esta configuración, Athenus garantiza la seguridad, privacidad y disponibilidad de la información, brindando a las empresas una solución confiable para la gestión de documentación interna.

### *C. Procesamiento de datos y respuestas*

El sistema procesa los documentos dividiéndolos en fragmentos solapados para optimizar la búsqueda. A cada fragmento se le genera una representación numérica (embedding) que captura sus características semánticas, y esta representación se almacena en una base de datos vectorial.

Para responder a cada consulta se emplea un enfoque híbrido que integra diversas técnicas de búsqueda y recuperación de información:

1) **Búsqueda vectorial (semántica):** Se utiliza la similitud entre embeddings para identificar fragmentos de texto relevantes en función del significado y el contexto de la consulta.
2) **Búsqueda BM25L:** Se implementa el algoritmo BM25L, que evalúa la coincidencia de términos entre la consulta y los documentos, asignando ponderaciones específicas basadas en la frecuencia y distribución de las palabras.
3) **Búsqueda por palabras clave (fallback):** Si tanto la búsqueda vectorial como la BM25L no generan resultados satisfactorios, se recurre a una búsqueda basada en palabras clave para extraer fragmentos que contengan términos relevantes.

El proceso se desarrolla de la siguiente manera:

1) **Fragmentación y representación del contenido:** Cada documento se divide en fragmentos solapados para optimizar la búsqueda. A cada fragmento se le genera

una representación numérica (embedding) que captura las características semánticas del texto, y esta representación se almacena en una base de datos vectorial.

2) **Búsqueda híbrida:** Para responder a cada consulta se implementan tres estrategias:
   - **Búsqueda exacta (sml25L):** Se buscan coincidencias literales en el texto.
   - **Búsqueda semántica:** Se utiliza la similitud de los embeddings para identificar documentos relevantes basados en el significado y el contexto de la consulta.
   - **Búsqueda por palabras clave (fallback):** Si las dos búsquedas anteriores no arrojan resultados satisfactorios, se recurre a una búsqueda basada en palabras clave.
3) **Reordenamiento de resultados:** Los resultados obtenidos se combinan asignando pesos relativos a cada método de búsqueda. Posteriormente, se utiliza un modelo *CrossEncoder* para reordenar los documentos según su relevancia respecto a la consulta.
4) **Integración con el historial y generación de respuestas:** El sistema utiliza el historial de conversación para enriquecer la consulta y proporcionar contexto adicional. La información relevante se integra en un *prompt* que se envía a un modelo de lenguaje, encargado de generar una respuesta completa y coherente. Además, se implementa un mecanismo de memoria a corto plazo que permite mantener la continuidad en las interacciones.
5) **Retroalimentación del usuario:** Se almacena la retroalimentación del usuario sobre la respuesta obtenida, lo que posibilita futuras mejoras en el proceso de recuperación y en la generación de respuestas.

Este enfoque integral y modular combina técnicas tradicionales de búsqueda con avanzadas metodologías de procesamiento semántico, asegurando que, independientemente de la consulta, se proporcione una respuesta precisa y contextualizada.

**Control de acceso:** Además, cada rol cuenta con su propia base de datos vectorial, lo que evita el acceso a información restringida.

## V. Resultados y Evaluación

### A. Desempeño del chatbot en pruebas reales

El control de acceso mediante bases de datos vectoriales asignadas por roles ha demostrado ser eficaz; por ejemplo, al utilizar el rol de contabilidad, el modelo recupera únicamente los datos financieros pertinentes, sin exponer información de otros departamentos (véase Figura 1). Sin embargo, se ha identificado que los modelos compactos tienden a ”alucinar” (inventar datos) cuando el contexto resulta insuficiente o poco claro. Para mitigar este problema, se sugiere realizar un fine-tuning con ejemplos de respuestas inadecuadas y añadir capas adicionales de procesamiento del lenguaje natural, de modo que el sistema se abstenga de responder sin datos pertinentes.

Adicionalmente, al evaluar modelos que incorporan cadenas de pensamiento se han obtenido respuestas de mayor calidad y precisión, aunque a costa de un mayor tiempo de procesamiento (véase Figura 2). Este compromiso entre precisión y velocidad representa un desafío a abordar en futuras optimizaciones.

Finalmente, la experiencia ha demostrado que, aunque inicialmente se pensó que modelos como LLaMA en sus versiones compactas serían suficientes, es necesario recurrir a modelos con mayor número de parámetros y capacidades avanzadas para mejorar la precisión y coherencia de las respuestas. Más allá de la estructura tradicional de un RAG (Retrieval Augmented Generation), resulta fundamental incorporar etapas adicionales —como el preprocesamiento y la validación del contexto, la gestión del historial de conversación y el establecimiento de umbrales de confianza— para que el sistema funcione de manera óptima como asistente y ofrezca una experiencia de usuario consistente.

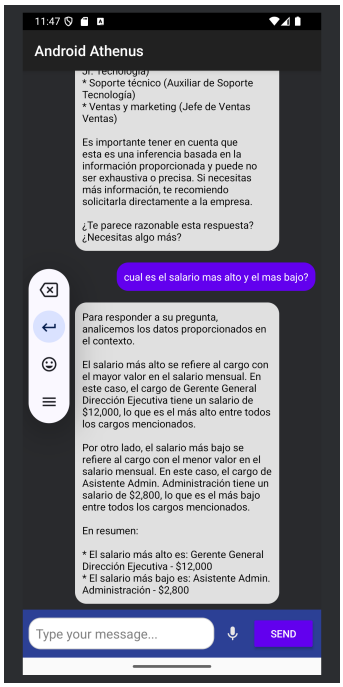


Fig. 1. Acceso restringido a datos de contabilidad mediante el rol correspondiente.

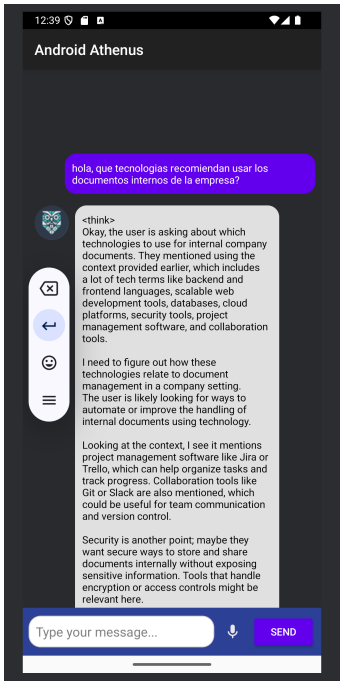


Fig. 2. Ejemplo de cadena de pensamiento con DeepSeek-r1.

## VI. Conclusiones y Trabajo Futuro

- La estrategia de control de acceso basada en bases de datos vectoriales asignadas por roles ha resultado eficaz para proteger la información sensible. Sin embargo, se ha observado que el modelo, especialmente en versiones compactas, tiende a generar respuestas inventadas (”alucinaciones”) cuando el contexto es insuficiente o poco claro. Para mitigar esta limitación, se propone realizar un fine-tuning con ejemplos de respuestas inadecuadas y agregar capas adicionales de procesamiento del lenguaje natural, de modo que el sistema se abstenga de responder cuando no haya información pertinente. Además, se plantea la implementación de un sistema de pagos que permita acceder a funcionalidades avanzadas y garantice un control más riguroso del acceso a la información.
- El uso de herramientas de aprendizaje de máquina en la nube y en el dispositivo aumentan las capacidades y caracterísitas de las aplicaciones móviles. Particularmente con whisper y LLMs en Athenus se logró una interacción más natural con el usuario.

### A. Resumen de logros y limitaciones del sistema

- **Logros:**
  - Realiza búsquedas integradas combinando métodos vectoriales, BM25L y búsqueda por palabras clave.
  - Limita el acceso a documentos según roles, evitando el acceso a información restringida.
  - Gestiona sesiones y utiliza el contexto del chat para generar respuestas coherentes.
  - Permite la transcripción de voz a texto on device, integrando esta funcionalidad en el proceso de consulta.
- **Limitaciones:**
  - Los modelos pequeños pueden generar respuestas inventadas (”alucinaciones”).
  - El manejo del contexto en chats extensos aún presenta desafíos en términos de precisión.
  - La integración de múltiples fuentes de información puede dar lugar a inconsistencias.

### B. Posibles mejoras y futuras actualizaciones

- Mejorar los LLMs mediante fine-tuning y actualización de arquitecturas para reducir las alucinaciones.
- Añadir etapas adicionales de procesamiento del lenguaje natural para refinar la relevancia y precisión de las respuestas.
- Optimizar el manejo del contexto del chat en el sistema de recuperación de información.
- Implementar un sistema de pagos que permita acceder a funcionalidades avanzadas y gestione de forma diferenciada el acceso a la información.

## Referencias

[1] M. C. Ungan, “Standardization through process documentation,” *Business Process Management Journal*, vol. 12, no. 2, pp. 135–148, 2006.
[2] X. Shu, D. Yao, and E. Bertino, “Privacy-preserving detection of sensitive data exposure,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1092–1103, 2015. doi: 10.1109/TIFS.2015.2398363
[3] IBM, “IBM Watson,” IBM, 2025. [Online]. Available: https://www.ibm.com/watson. [Consultado: 20-Feb-2025].
[4] Google, “Dialogflow,” Google Cloud, 2025. [Online]. Available: https://dialogflow.cloud.google.com/. [Consultado: 20-Feb-2025].
[5] Microsoft, “Azure OpenAI Service,” Microsoft Azure, 2025. [Online]. Available: https://azure.microsoft.com/es-es/products/ai-services/openai-service. [Consultado: 20-Feb-2025].
[6] Amazon Web Services, “Amazon Bedrock,” AWS, 2025. [Online]. Available: https://aws.amazon.com/bedrock/. [Consultado: 20-Feb-2025].