|  |
| --- |
|  |

Modelamiento y documentación de software

Proyecto: GenIA

Versión 1.0

|  |  |  |
| --- | --- | --- |
|  |  | 05-2025 |

Ficha del documento

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Autor o autores** |
| 23 de mayo de 2025 | 1.0 | Lina Lizeth Londoño Marín, Leandro Rivera Ríos, Balmer Valencia Banguero |

Contenido

[Ficha del documento 2](#_Toc160517267)

[Contenido 3](#_Toc160517268)

[1 Introducción 5](#_Toc160517269)

[1.1 Propósito 5](#_Toc160517270)

[1.2 Alcance 5](#_Toc160517271)

[1.3 Personal involucrado 6](#_Toc160517272)

[1.4 Definiciones, acrónimos y abreviaturas 6](#_Toc160517273)

[1.5 Referencias 6](#_Toc160517274)

[1.6 Resumen 6](#_Toc160517275)

[2 Descripción general 7](#_Toc160517276)

[2.1 Perspectiva del producto 7](#_Toc160517277)

[2.2 Funcionalidad del producto 7](#_Toc160517278)

[2.3 Características de los usuarios 7](#_Toc160517279)

[2.4 Restricciones 7](#_Toc160517280)

[2.5 Evolución previsible del sistema 8](#_Toc160517281)

[3 Requisitos específicos 8](#_Toc160517282)

[3.1 Requisitos comunes de las interfaces 8](#_Toc160517283)

[3.1.1 Interfaces de usuario 8](#_Toc160517284)

[3.1.2 Interfaces de hardware 9](#_Toc160517285)

[3.1.3 Interfaces de software 9](#_Toc160517286)

[3.1.4 Interfaces de comunicación 9](#_Toc160517287)

[3.2 Requisitos funcionales 9](#_Toc160517288)

[3.2.1 Requisito funcional 1 9](#_Toc160517289)

[3.2.2 Requisito funcional 2 9](#_Toc160517290)

[3.2.3 Requisito funcional 3 10](#_Toc160517291)

[3.2.4 Requisito funcional n 10](#_Toc160517292)

[3.3 Requisitos no funcionales 10](#_Toc160517293)

[3.3.1 Requisitos de rendimiento 10](#_Toc160517294)

[3.3.2 Seguridad 11](#_Toc160517295)

[3.3.3 Fiabilidad 11](#_Toc160517296)

[3.3.4 Disponibilidad 11](#_Toc160517297)

[3.3.5 Mantenibilidad 11](#_Toc160517298)

[3.3.6 Portabilidad 11](#_Toc160517299)

[3.4 Modelo de calidad 11](#_Toc160517300)

[3.5 Otros requisitos 12](#_Toc160517301)

[4 Apéndices 12](#_Toc160517302)

# Introducción

Esta Especificación de Requisitos de Software (SRS) describe los requisitos para el desarrollo de DataGenie , una solución basada en chatbot diseñada para optimizar el acceso y el análisis de datos para Bancolombia SA. El documento cumple con el estándar IEEE Std 830-1998 y ofrece una visión general completa del propósito, el alcance, las funcionalidades y las limitaciones del sistema. Sirve como guía para el equipo de desarrollo en el diseño e implementación del sistema, y ​​para que las partes interesadas validen que el sistema satisface sus necesidades

## Propósito

**Propósito del documento :** Definir los requisitos funcionales y no funcionales de DataGenie, una solución de software que permite a los usuarios interactuar con bases de datos relacionales (MySQL) a través de consultas en lenguaje natural, reduciendo la dependencia de los analistas técnicos y mejorando la eficiencia operativa.

**Público objetivo :** Este documento está dirigido a:

* Equipo de desarrollo (programadores, diseñadores y probadores).
* Partes interesadas del proyecto en Bancolombia SA, incluidos analistas de integración de datos y usuarios comerciales.

## Alcance

**Identificación del producto :** DataGenie (AI-DG)

**Funcionalidades :**

* Permita a los usuarios consultar bases de datos utilizando lenguaje natural a través de una interfaz gráfica intuitiva construida con Next.js.
* Genere consultas SQL automáticamente en función de las entradas del usuario.
* Conéctese a bases de datos MySQL para recuperar y mostrar datos en tiempo real.
* Proporcionar visualización de datos básicos (por ejemplo, tablas, gráficos simples).
* Registrar las interacciones de los usuarios para su análisis y mejora continua.

**Exclusiones :**

* Creación o mantenimiento de bases de datos.
* Soporte para consultas avanzadas de Big Data.
* Integración con herramientas externas de Business Intelligence (BI).
* Escalabilidad para uso masivo basado en la nube.

**Beneficios :** DataGenie simplifica el acceso a los datos para usuarios no técnicos, reduce los tiempos de respuesta a las consultas y permite a los analistas centrarse en tareas estratégicas, mejorando la eficiencia operativa de Bancolombia.

## Personal involucrado

|  |  |
| --- | --- |
| Nombre | Lina Lizeth Londoño Marín |
| Rol | Frontend Developer |
| Categoría profesional | Student/Engineer |
| Responsabilidades | Design interface, test functionality |
| Información de contacto | lina.londono@uao.edu.co |
| Aprobación | Si |

|  |  |
| --- | --- |
| Nombre | Balmer Valencia Banguero |
| Rol | Backend Developer |
| Categoría profesional | Student/Engineer |
| Responsabilidades | Develop backend, integrate LLM and database |
| Información de contacto | balmer.valencia@uao.edu.co |
| Aprobación | Si |

|  |  |
| --- | --- |
| Nombre | Leandro Rivera Ríos |
| Rol | Lead Full Stack Developer |
| Categoría profesional | Student/Engineer |
| Responsabilidades | Develop backend and Frontend |
| Información de contacto | leandro.rivera@uao.edu.co |
| Aprobación | Si |

## Definiciones, acrónimos y abreviaturas

|  |  |
| --- | --- |
| **Término** | **Definición** |
| DataGenie | Nombre de la herramienta de interacción de datos basada en chatbot. |
| SQL | Lenguaje de consulta estructurado, utilizado para consultar bases de datos relacionales. |
| Next.js | Un marco React para crear interfaces de usuario. |
| FastAPI | Un marco de Python para crear API. |
| LLM | Modelo de lenguaje grande, utilizado para el procesamiento del lenguaje natural. |
| NLQ | Consulta en lenguaje natural, consultas realizadas en lenguaje natural en lugar de código. |

## Referencias

|  |  |  |  |
| --- | --- | --- | --- |
| **Titulo** | **Ruta** | **Fecha** | **Autor** |
| Documentación de FastAPI | https://fastapi.tiangolo.com/ | Accedido en 2025 | Tiangolo |
| Documentación de MySQL | https://dev.mysql.com/doc/ | Accedido en 2025 | Corporación Oracle |
| Next.js: El marco de React | https://nextjs.org/ | Accedido en 2025 | Vercel |
| Streamlit Documentation | https://docs.streamlit.io/ | May 2025 | Streamlit |
| Groq API Documentation | https://console.groq.com/docs/api-reference | May 2025 | Groq Inc. |
| DataGenie GitHub Repository | https://github.com/LeoR22/ai\_chatbot\_mysql | May 2025 | Leandro Rivera Ríos et al. |
| Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks | https://arxiv.org/abs/2005.11401 | May 2025 | Lewis et al. |
| Python Documentation | https://docs.python.org/3/ | May 2025 | Python Software Foundation |
| ISO/IEC 9126 Software Quality Model | https://www.iso.org/standard/22749.html | May 2025 | ISO/IEC |
| Ley 1581 de 2012 - Colombia Data Protection | https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49908 | May 2025 | Government of Colombia |
| MySQL Connector/Python Documentation | https://dev.mysql.com/doc/connector-python/en/ | May 2025 | Oracle Corporation |
| Streamlit Community Forum | https://discuss.streamlit.io/ | May 2025 | Streamlit Community |
| Building Conversational AI with Python | https://www.oreilly.com/library/view/building-conversational-ai/9781492033530/ | May 2025 | O’Reilly Media |

## Resumen

Este documento SRS detalla los requisitos de DataGenie, una herramienta basada en chatbot para Bancolombia SA que optimiza el acceso a los datos. El documento se estructura en una introducción, propósito, alcance, personal, definiciones, referencias, descripción general y requisitos específicos. Incluye requisitos funcionales y no funcionales detallados, descripciones de la interfaz y restricciones para guiar el desarrollo y la validación.

# Descripción general

## Perspectiva del producto

DataGenie es un producto de software independiente que se integra con las bases de datos MySQL existentes de Bancolombia. Utiliza un enfoque de Recuperación-Generación Aumentada (RAG) para procesar consultas en lenguaje natural, generar consultas SQL y recuperar datos. El sistema emplea un frontend basado en Streamlit para la interacción del usuario, un backend Python/FastAPI para el procesamiento y un LLM Groq para la generación de consultas.

## Funcionalidad del producto

**Procesamiento de consultas en lenguaje natural** : los usuarios pueden ingresar consultas en lenguaje natural (por ejemplo, "¿Cuántos empleados hay en la tabla de empleados?") y el sistema genera las consultas SQL correspondientes.

**Conectividad de base de datos :** se conecta a bases de datos MySQL utilizando credenciales proporcionadas por el usuario (host, puerto, nombre de usuario, contraseña, nombre de base de datos).

**Recuperación y visualización de datos :** recupera datos y muestra los resultados en tablas o gráficos simples.

**Registro de interacciones :** almacena las interacciones del usuario en MySQL para su análisis y mejora del sistema.

**Aprendizaje basado en retroalimentación :** utiliza la retroalimentación para optimizar la generación de consultas y la precisión de las respuestas

## Características de los usuarios

|  |  |
| --- | --- |
| Tipo de usuario | Usuarios empresariales |
| Formación | No técnico, centrado en los negocios |
| Habilidades | Conocimientos básicos de informática, sin conocimientos de SQL. |
| Actividades | Consulta de datos para informes y toma de decisiones |

|  |  |
| --- | --- |
| Tipo de usuario | Analistas de integración de datos |
| Formación | Experiencia técnica en SQL |
| Habilidades | SQL avanzado, habilidades de análisis de datos |
| Actividades | Validar consultas, realizar análisis avanzados |

## Restricciones

**Herramientas de desarrollo :** Streamlit para frontend, Python/FastAPI para backend, Groq LLM para procesamiento de consultas.

**Soporte de base de datos :** Limitado a MySQL en la versión inicial.

**Conectividad :** Requiere acceso a Internet para bases de datos basadas en la nube e integración de API LLM.

**Alcance :** Excluye la creación y el mantenimiento de bases de datos y las consultas avanzadas de Big Data.

**Seguridad :** Debe operar dentro de la intranet y VPN de Bancolombia.

## Evolución previsible del sistema

Integración con tipos de bases de datos adicionales (por ejemplo,PostgreSQL, Mongodb).

Visualización de datos mejorada (por ejemplo, gráficos avanzados, paneles de control).

Escalabilidad para uso de gran volumen basado en la nube.

# Requisitos específicos

|  |  |  |  |
| --- | --- | --- | --- |
| Id | RF | | |
| Número de requisito |  | | |
| Nombre de requisito |  | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito |  | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

## Requisitos comunes de las interfaces

### Interfaces de usuario

**Título :** "Conexión a Base de Datos"

**Campos** : Host (por ejemplo, localhost), Puerto (por ejemplo, 3303), Nombre de usuario (por ejemplo, root), Contraseña, Nombre de la base de datos (por ejemplo, Chat).

**Botón :** “Conectar” para iniciar la conexión a la base de datos.

Comentarios : muestra mensajes como “Conexión exitosa” o “Error en la conexión”.

**Estilo :** Diseño limpio e intuitivo con una combinación de colores moderna (por ejemplo, azul y blanco para la marca Bancolombia).

**Título :** "Chat con MySQL Integrado con Groq"

**Mensaje de bienvenida** : "Hola, soy tu asistente SQL. Pregúntame cualquier cosa sobre tu base de datos".

**Campo de entrada :** para consultas en lenguaje natural (por ejemplo, “¿Cuántos empleados tiene la tabla empleados?”).

**Salida :** Muestra la consulta generada (por ejemplo, SELECT COUNT(\*) AS NumEmpleados FROM empleados; ), la explicación y los resultados en una tabla o gráfico.

Estilo : Fácil de usar, con texto claro y diseño minimalista.

### Interfaces de hardware

NA : El sistema no interactúa directamente con componentes de hardware específicos más allá de los dispositivos informáticos estándar (PC, servidores) que alojan las bases de datos y las aplicaciones.

### Interfaces de software

**MySQL :** se conecta a bases de datos MySQL mediante controladores JDBC/ODBC estándar.

**Propósito :** Ejecutar consultas SQL y recuperar datos.

Formato : Sintaxis SQL estándar, JSON para intercambio de datos.

**API LLM** (por ejemplo, Groq) : se integra con un modelo de lenguaje grande para el procesamiento del lenguaje natural.

**Propósito :** Convertir consultas en lenguaje natural a SQL/NoSQL.

**Formato :** API REST con cargas útiles JSON.

### Interfaces de comunicación

**Protocolo :** HTTP/HTTPS para la comunicación entre el frontend de Next.js, el backend de FastAPI y las bases de datos.

**Seguridad :** SSL/TLS para transmisión segura de datos.

**Acceso a la base de datos** : utiliza puertos de base de datos estándar (por ejemplo, 3303 para MySQL).

## Requisitos funcionales

### Requisito funcional 1

|  |  |
| --- | --- |
| **Identificación** | **RF1** |
| Número 1 | Número 1 |
| Nombre Conexión a la base de datos | Nombre Conexión a la base de datos |
| Tipo Requisito | Tipo Requisito |
| Fuente Requisitos del Proyecto (Capítulo 1) | Fuente Requisitos del Proyecto (Capítulo 1) |
| Prioridad Alta/Esencial | Prioridad Alta/Esencial |

**Descripción** : El sistema debe permitir a los usuarios conectarse a bases de datos MySQL mediante credenciales (host, puerto, nombre de usuario, contraseña y nombre de la base de datos) a través de una interfaz gráfica. Tras la conexión, se muestra un mensaje de confirmación; de lo contrario, se muestra un mensaje de error.

### Requisito funcional 2

|  |  |
| --- | --- |
| **Identificación** | **RF2** |
| Número | 2 |
| Nombre | Procesamiento de consultas en lenguaje natural |
| Tipo | Requisito |
| Fuente | Requisitos del Proyecto (Capítulo 1) |
| Prioridad | Alta/Esencial |

**Descripción :** El sistema debe procesar consultas en lenguaje natural (p. ej., "¿Cuántos empleados hay en la tabla de empleados?") y generar las consultas SQL/NoSQL correspondientes mediante un LLM integrado mediante API. La consulta generada y su explicación se muestran al usuario.

### Requisito funcional 3

|  |  |
| --- | --- |
| **Identificación** | **RF1** |
| Número | 3 |
| Nombre | Recuperación y visualización de datos |
| Tipo | Requisito |
| Fuente | Requisitos del Proyecto (Capítulo 1) |
| Prioridad | Medios/Deseado |

**Descripción :** El sistema debe ejecutar las consultas generadas en la base de datos conectada y mostrar los resultados en una tabla o un gráfico simple (p. ej., un gráfico de barras para datos de ventas). Las consultas no válidas deben generar un mensaje de error con sugerencias de corrección.

### Requisito funcional n

|  |  |
| --- | --- |
| **Identificación** | **RF1** |
| Identificación | RF4 |
| Número | 4 |
| Nombre | Registro de interacciones |
| Tipo | Requisito |
| Fuente | Requisitos del Proyecto (Capítulo 1) |

**Descripción :** El sistema debe registrar las interacciones del usuario (consultas, resultados, comentarios) en una base de datos MySQL para el análisis y la mejora continua del rendimiento del sistema.

## Requisitos no funcionales

### Requisitos de rendimiento

El sistema debe manejar hasta 50 usuarios simultáneos con tiempos de respuesta menores a 7 segundos para el 95% de las consultas.

La ejecución de consultas en bases de datos debe completarse dentro de los 10 segundos para consultas estándar.

### Seguridad

Utilice SSL/TLS para todas las transmisiones de datos.

Almacene las credenciales de la base de datos de forma segura mediante encriptación.

Registrar las actividades del usuario con fines de auditoría, a las que sólo puede acceder personal autorizado.

Ejecución solo mediante VPN.

Solo puede ser ejecutado con la intranet.

### Fiabilidad

El sistema debe alcanzar un tiempo medio entre fallos (MTBF) de al menos 30 días.

Los mecanismos de recuperación de errores deben garantizar que no haya pérdida de datos durante la ejecución de la consulta.

### Disponibilidad

El sistema debe estar disponible el 99,5% del tiempo durante el horario comercial (de 8:00 a 18:00 horas).

### Mantenibilidad

Las tareas de mantenimiento (por ejemplo, actualizar modelos LLM, corregir errores) deben ser realizadas por el equipo de desarrollo.

Se deben generar registros semanales de las interacciones del usuario para el análisis del rendimiento.

### Portabilidad

El sistema debe ejecutarse en navegadores web estándar (Chrome, Firefox) y ser compatible con servidores Windows y Linux.

El uso de Next.js y FastAPI garantiza la independencia de la plataforma para el frontend y el backend.

## Modelo de calidad

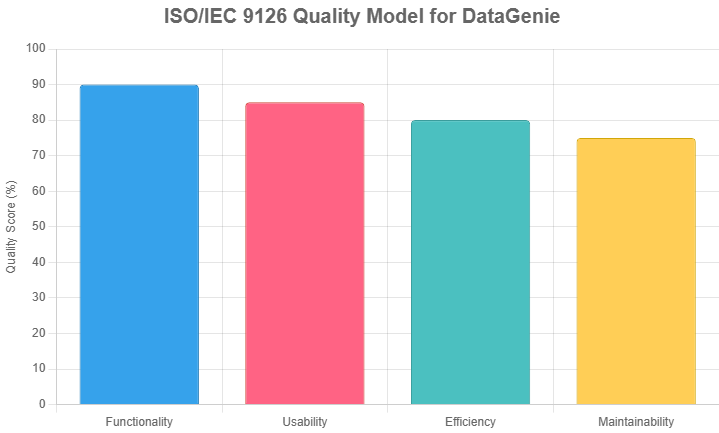
El modelo de calidad sigue la norma ISO/IEC 9126 , centrándose en:

Funcionalidad : Generación precisa de consultas y recuperación de datos.

Usabilidad : Interfaz intuitiva para usuarios no técnicos.

Eficiencia : tiempos de respuesta rápidos y bajo uso de recursos.

Mantenibilidad : Estructura de código modular para actualizaciones fáciles.



## Otros requisitos

Requisitos culturales : La interfaz debe estar alineada con la marca de Bancolombia (por ejemplo, colores, logotipo).

Requerimientos Legales : Cumplimiento de la normativa de protección de datos (ej. Ley 1581 de 2012 de Colombia sobre protección de datos personales).

# Apéndices

**Apéndice A: Prototipo de baja fidelidad** (Ver "Capitulo 1 \_Anexo\_Solucion\_PrototipoBajoNivel.pdf" para wireframes).

**Apéndice B: Storyboard del Problema** (Ver "Capitulo 1 \_Anexo\_Problema\_SituacionActual.pdf").

**Apéndice C: Storyboard del Proceso Futuro** (Ver "Capitulo 1 \_Anexo\_Solucion\_SituacionFutura.pdf").