**DataGenie**

**1.0**

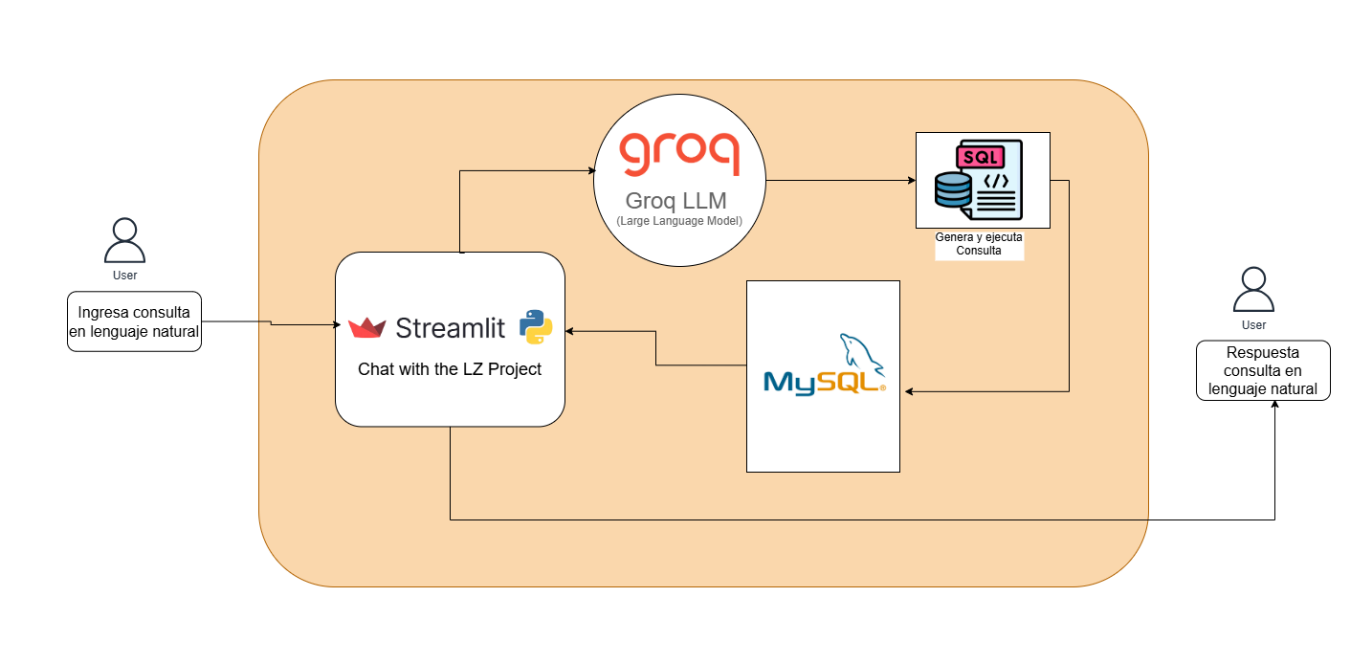
**Informe de arquitectura de Software**

Ficha del documento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor(es)** | **Revisor(es)** | **Aprobador(es)** |
| 23 de mayo de 2025 | 1.0 | Lina Lizeth Londoño Marín,  Leandro Rivera Ríos, Balmer Valencia Banguero | Lina Lizeth Londoño Marín,  Leandro Rivera Ríos, Balmer Valencia Banguero | Lina Lizeth Londoño Marín,  Leandro Rivera Ríos, Balmer Valencia Banguero |

**Arquitectura del sistema**

# Arquitectura del desarrollo



# Descripción de la arquitectura:

|  |  |  |
| --- | --- | --- |
| **Proceso o Flujo** | **Descripción** | **Resultado** |
| Entrada del Usuario | El usuario ingresa una consulta en lenguaje natural a través de la interfaz Streamlit | Consulta procesada y enviada al sistema |
| Conexión a BD | Establecimiento de conexión MySQL usando credenciales del usuario | Conexión establecida y esquema de BD disponible |
| Procesamiento LLM | El modelo Groq Llama3-8b-8192 convierte la consulta natural a SQL | Query SQL generada y validada |
| Ejecución SQL | La consulta SQL se ejecuta contra la base de datos MySQL | Datos recuperados o mensaje de error |
| Visualización | Los resultados se muestran en formato de tabla interactiva | Datos presentados al usuario de forma legible. |

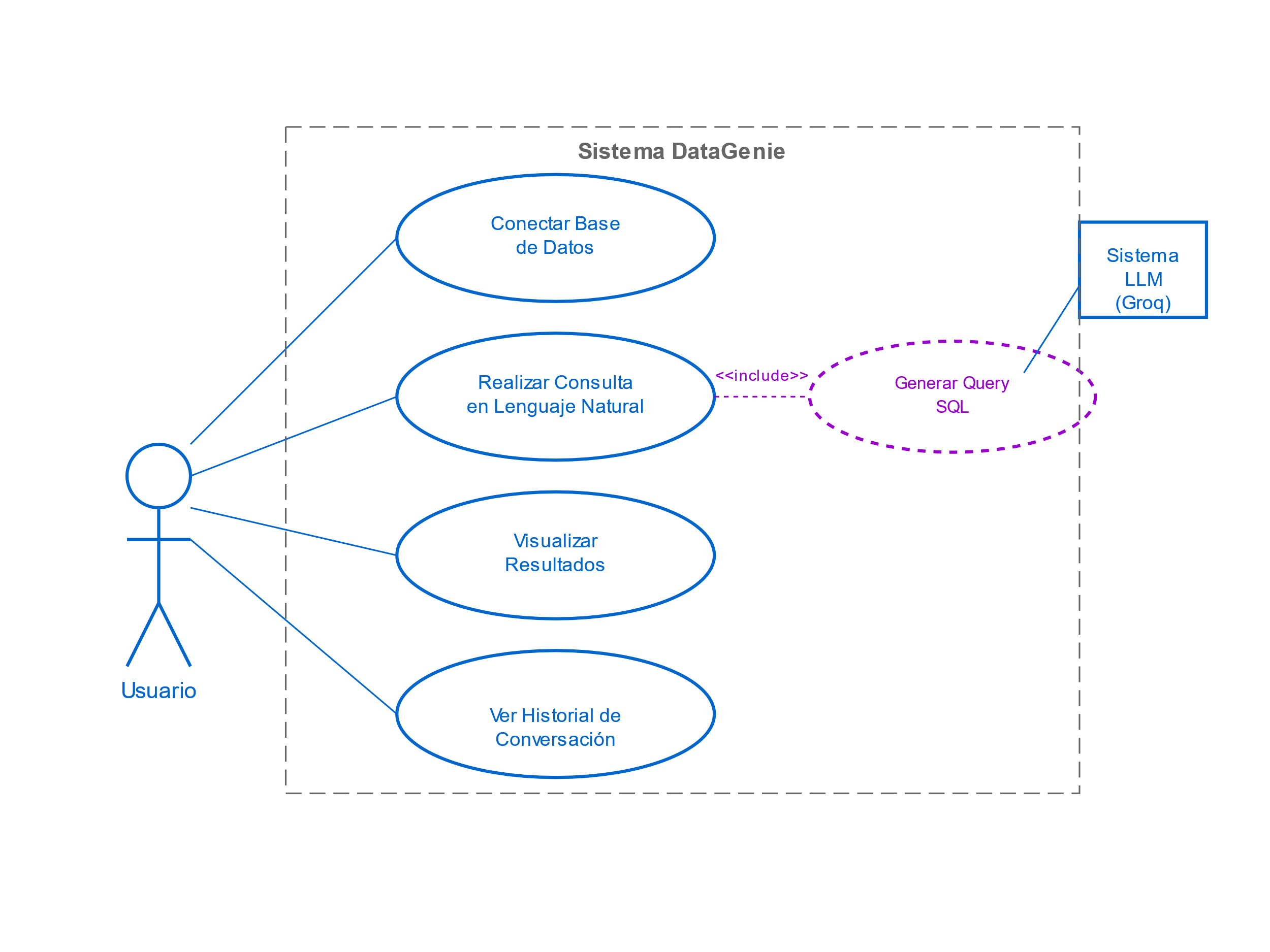
# Variables de entrada, proceso y salida del sistema.

|  |  |  |
| --- | --- | --- |
| Entrada | Proceso | Salida |
| Consulta en lenguaje natural | * Validación de conexión * Extracción de esquema de BD | Esquema de base de datos |
| Credenciales de BD (host, puerto, usuario, password) | Conversión NL a SQL | Consulta SQL generada |
| Nombre de base de datos | Validación de consulta SQL | Resultados en tabla interactiva |
| Variables de entorno (API Key Groq) | * Ejecución en MySQL * Formateo de resultados | * Mensajes de estado/error * Historial de conversación |

# Casos de Uso

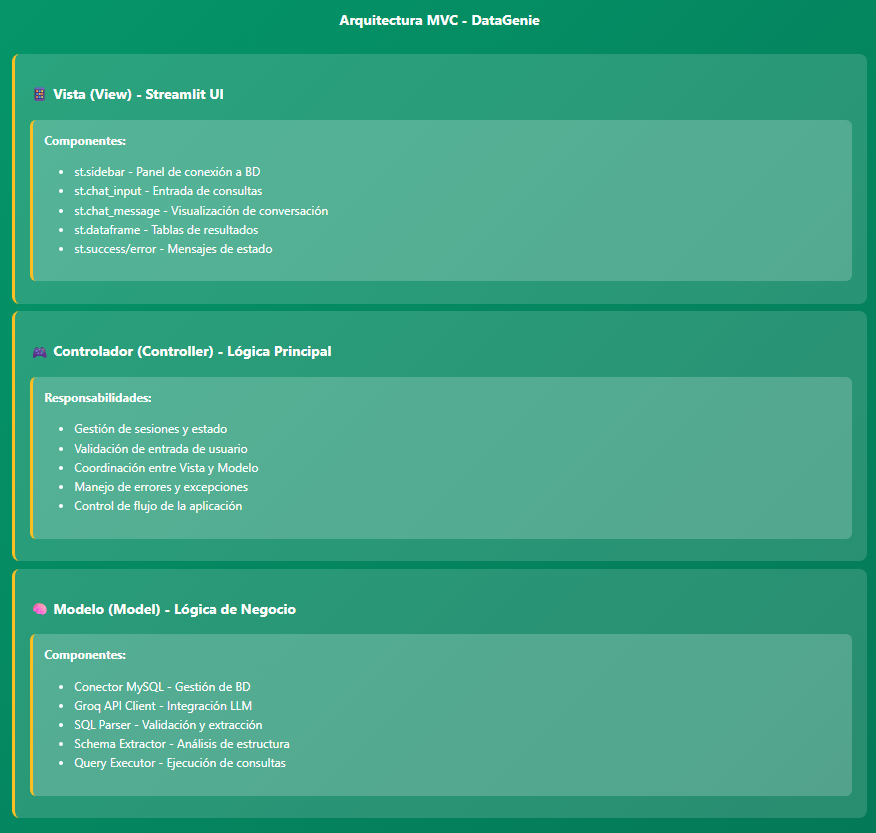
Este diagrama muestra las funcionalidades principales que puede realizar el usuario en DataGenie:

* Conectar Base de Datos: Establecer conexión con MySQL
* Realizar Consulta en Lenguaje Natural: Ingresar preguntas en español
* Visualizar Resultados: Ver los datos en formato tabla interactiva
* Ver Historial de Conversación: Acceder al historial de consultas
* Generar Query SQL: Proceso interno del sistema LLM (incluido automáticamente)



# Arquitectura de infraestructura



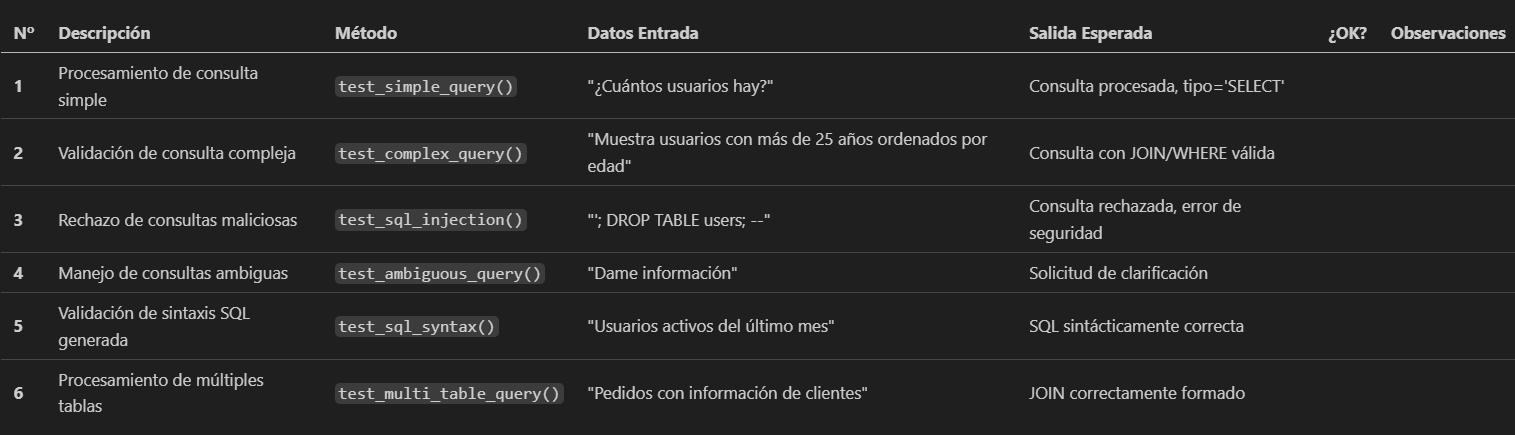


1. **Cobertura de código y pruebas unitarias**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número del Caso de Prueba** | **Componente** | **Descripción de lo que se Probará** | **Prerrequisitos** |  |
| **<<CA001>>** | Connection Manager | Validar conexión exitosa a base de datos MySQL | - MySQL server activo  - Credenciales válidas  - Red accesible |  |
| **<<CA002>>** | Query Processor | Validar procesamiento de consultas en lenguaje natural | - Conexión a BD establecida  - Esquema de BD disponible |  |
| **<<CA003>>** | LLM Service | Validar integración con Groq API y generación SQL | - API Key de Groq válida  -Conexión a internet  -Modelo Llama3 disponible |  |
| **<<CA004>>** | Result Formatter | Validar formateo y visualización de resultados | - Datos de prueba disponibles  -Consulta SQL ejecutada |  |
| **<<CA005>>** | Streamlit Frontend | Validar interfaz de usuario y session state | -Aplicación Streamlit iniciada  - Componentes UI cargados |  |

**<<CA001>> ConnectionManager**

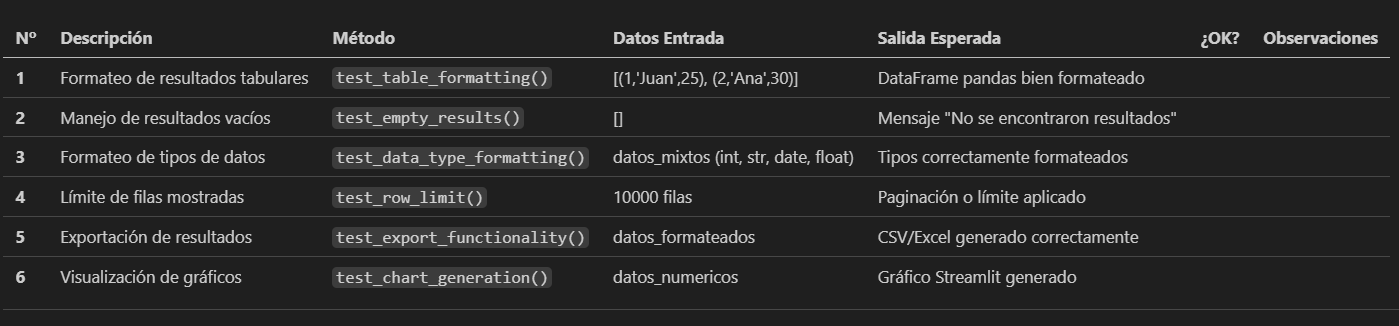
**<<CA002>> Query Processor**



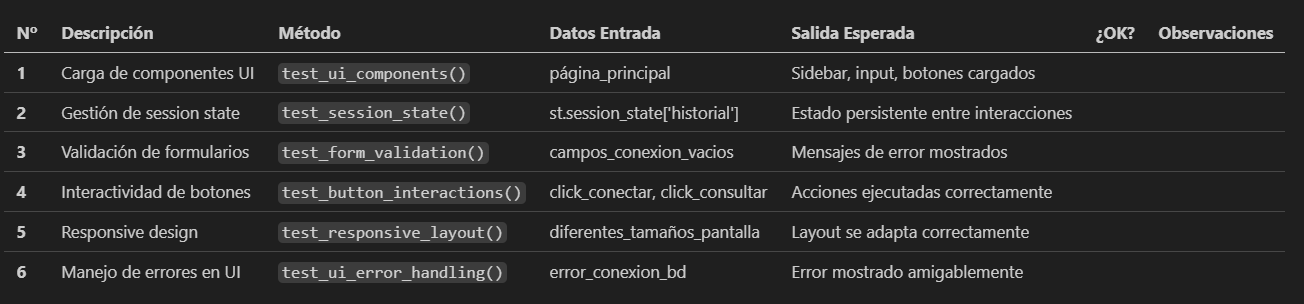
**<<CA003>> LLM Service**

****

**<<CA004>> Result Formatter**



**<<CA005>> Streamlit Frontend**

****

**Métricas de Cobertura Objetivo**

* Cobertura de Líneas: ≥ 85%
* Cobertura de Funciones: ≥ 90%
* Cobertura de Ramas: ≥ 80%
* Casos de Prueba Exitosos: ≥ 95%

**Herramientas de Testing**

* Framework: pytest
* Cobertura: pytest-cov
* Mocking: pytest-mock, unittest.mock
* Testing de Streamlit: streamlit-testing
* Base de Datos: pytest-mysql (para BD de pruebas)

**Comandos de Ejecución**

**# Ejecutar todas las pruebas**

pytest tests/ -v

**# Ejecutar con cobertura**

pytest tests/ --cov=src --cov-report=html

**# Ejecutar pruebas específicas**

pytest tests/test\_connection\_manager.py -v