

INSTITUTO TECNOLÓGICO DE CULIACÁN – TECNM



Sistema experto en análisis estadístico de datos hidrometeorológicos
mediante agentes inteligentes para monitoreo climático.

TESIS

OBTENER EL GRADO DE
INGENIERO EN SISTEMAS COMPUTACIONALES

POR:

Christopher Chaparro Castillo

DIRECTOR DE TESIS:

DR. Zuriel Dathan Mora Felix

Culiacán Sin, A 16 De Agosto Del 2026



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Culiacán

DIVISION DE ESTUDIOS PROFESIONALES
ASUNTO: **Autorización de Impresión**

Culiacán, Sinaloa, 19/08/2026 18:25

C. CHRISTOPHER CHAPARRO CASTILLO
PASANTE(S) DE LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
P R E S E N T E.

De acuerdo con fallo emitido por la Comisión Revisora Integrada por:

Cédula Profesional

Presidente:	Doctorado en Ciencias de la Ingeniería Zuriel Dathan Mora Felix	14236234
Secretario:	Doctor en Ciencia y Tecnología con Especialidad en Ingeniería Ambiental Jesús Gabriel Rangel Peraza	8327423
Vocal:	Maestría en Ciencias en Ciencias Computacionales Arcelia Judith Bustillos Martínez	13118410

Y considerando que reúne los requisitos establecidos en la Opción: **TITULACIÓN INTEGRAL: TESIS PROFESIONAL "SISTEMA EXPERTO EN ANALISIS ESTADISTICO DE DATOS HIDROMETEOROLÓGICOS MEDIANTE AGENTES INTELIGENTES PARA EL MONITOREO CLIMÁTICO"**, damos a usted nuestra autorización para que proceda a continuar con su proceso de titulación.

Sin otro particular, quedo de usted.

ATENTAMENTE
Excelencia en Educación Tecnológica®

CONCEPCIÓN MENDOZA ROSALES
LIC. CONCEPCIÓN MENDOZA ROSALES
JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

Educación
Secretaría de Educación Pública
INSTITUTO TECNOLÓGICO DE CULIACÁN
DEPARTAMENTO DE
DIVISIÓN DE ESTUDIOS PROFESIONALES



2026
año de
Margarita
Maza



Juan de Dios Bátiz 310 Pte. Col. Guadalupe C.P. 80220, Culiacán, Sinaloa
Tel. 667-454-0100 **tecnm.mx** **www.culiacan.tecnm.mx**

Dedicatoria

Dedico esta tesis a dios, a mis padres que sin ellos no hubiera tenido esta hermosa oportunidad en mi vida de crecer profesionalmente y ser quien soy ahora, muchísimas gracias, padres por sus enseñanzas y apoyo ultra amoroso siempre estaré agradecido con ustedes los quiero muchísimo. A mis hermanos que siempre me apoyaron a seguir adelante muchas gracias, hermanos. A mi novia, por todo el apoyo, amor y cariño incondicional que me has brindado te quiero muchísimo.

Agradecimientos

A mi madre Dolores Janeth Castillo y a mi padre Lorenzo Chaparro, por brindarme esta gran oportunidad de crecer no solo como profesional sino como persona y su amor que siempre me dieron para mantenerme fuerte.

A mis hermanos Jesús Manuel Chaparro Y Kevin Alexis Chaparro, por apoyarme siempre en los momentos de adversidad más difíciles y ayudarme a nunca darme por vencido.

A los profesores y doctores el Dr. Zuriel Dathan Mora Félix Y al Dr. Jesús Gabriel Rangel Peraza por brindarme esta valiosa oportunidad de desarrollarme como profesional y ser alguien mejor en la vida.

A mi novia, La Ing. Emma Daniela Portillo López, por todo el apoyo, amor y cariño incondicional que me ha brindado a lo largo de esta maravillosa aventura. Gracias por ser mi motor, por inspirarme a ser mejor cada día y por estar a mi lado siempre. No tengo palabras suficientes para expresar lo agradecido que estoy por tenerte en mi vida. Estoy profundamente orgulloso de ti, y sin duda, eres lo mejor que me ha pasado. Te quiero muchísimo.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Christopher Chaparro Castillo.

Culiacán, Sinaloa, México, 2026

Resumen

En tiempos muy cercanos, las redes hidrometeorológicas generan cantidades enormes de registros, vitales para comprender el tiempo y orientar resoluciones clave en áreas como control del agua, cultivo y alerta de catástrofes. No obstante, aprovechar esos datos con eficacia resulta difícil debido a archivos complicados, contenido faltante o incorrecto, y la exigencia de destrezas técnicas en código y números para analizarlos bien.

Las inteligencias artificiales han modificado radicalmente nuestra manera de interactuar y transitar dentro del entorno digital. Pese a sus notables avances en la comprensión del habla humana, presentan inconvenientes al no lograr conectar directamente con los repositorios oficiales de clima y agua. Por ello, sus respuestas pueden resultar inexactas, viejas o sin fuente clara, apoyándose en conocimientos previos en vez de buscar información actualizada sobre la marcha.

Ante tal circunstancia, el estudio actual sugiere desarrollar una plataforma cognitiva basada en modelo extenso de lenguaje (LLM), apta para interpretar requerimientos escritos en habla diaria y transformarlos en acciones ejecutables dentro de repositorios hidrometeorológicos. Con ese propósito, se integra un agente autónomo dotado de instrumentos especializados para manejar información, llevar a cabo estudios preliminares de datos (EDA) y emplear métodos cuantitativos tales como asociación lineal y ajuste de tendencias.

Asimismo, se ha creado un diseño web para facilitar el diálogo entre el operador y la plataforma, logrando formular consultas con rapidez y obtener soluciones precisas, legibles y repetibles. La plataforma garantiza que las soluciones se basen en hechos verificables, elevando por tanto la seguridad respecto al contenido entregado.

Por ende, dicha iniciativa contribuye a bajar barreras técnicas para obtener registros meteorológicos, a hacer que los flujos de estudio funcionen mejor y a permitir elecciones con mayor base sólida. De esta forma, se promueve una aplicación más directa y fácil del material hidrometeorológico, aspecto que beneficia igual a los especialistas que a aquellos sin formación técnica profunda.

Palabras Clave

- Modelos de lenguaje (LLM)
- Agentes de inteligencia artificial
- Bases de datos hidrometeorológicas
- Consulta en lenguaje natural (NLQ)
- Análisis exploratorio de datos (EDA)
- Extracción, Transformación y Carga (ETL)
- Preprocesamiento de datos
- Correlación estadística
- Regresión lineal
- Generación aumentada por recuperación (RAG)
- Trazabilidad de datos
- Reproducibilidad
- Aplicaciones web

Índice

1. Introducción	1
1.1. Planteamiento del problema	1
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. Hipótesis	2
1.4. Justificación	2
1.5. Organización del documento	3
2. Marco Teórico	4
2.1. Sistemas hidrometeorológicos	4
2.2.1. Datos climatológicos	4
2.2. Inteligencia artificial	5
2.3. Teoría agentica	5
2.3.1. Agentes de Inteligencia Artificial	5
2.3.2. LLM (Large Language Models)	6
2.3.3. Prompt	7
2.3.4. Prompt Engineering	7
2.3.5. RAG (Retrieval-Augmented Generation)	8
2.3.6. Uso de herramientas (Tools)	8
2.3.7. Arquitectura multi agente	9
2.3.8. Base de datos vectorial	9
2.3.9. Integración de LLM con bases de datos	9
2.3.10. Sistemas de consulta en lenguaje natural (NLQ)	10
2.4. Análisis de datos	10
2.4.1. ETL	11
2.4.2. Preprocesamiento de datos	12
2.4.3. Calidad de datos	12
2.4.4. Análisis Exploratorio de Datos (EDA)	13
2.4.5. Correlación estadística	13

2.4.6. Regresión lineal	13
2.4.7. Regresión múltiple	14
2.4.8. Trazabilidad y reproducibilidad	14
2.5. Tecnologías web	15
2.5.1. Aplicaciones web	15
2.5.2. API	15
2.5.3. React.js	16
2.5.4. Next.js	16
3. Estado del arte	17
3.1. Plataformas de datos hidrometeorológicos	17
3.1.1. CONAGUA	18
3.1.2. NOAA	19
3.1.3. Google Earth	20
3.2. Herramientas y aplicaciones basadas en interacción con datos climáticos	21
3.2.1. The Weather Channel	21
3.2.2. AccuWeather	22
3.2.3. Apple Weather	22
3.3. Integración con inteligencia artificial	23
3.3.1. LLMs	24
3.3.2. LangChain	24
3.3.3. Metodología RAG	26
3.4. Análisis cuantitativo entre plataformas	27
4. Metodología de desarrollo	29
4.1. Desarrollo del pipeline de datos	30
4.2. Metodología de desarrollo e inteligencia artificial	31
4.2.1. Metodología de desarrollo	31
4.2.2. Inteligencia artificial	32
4.3. Proceso de desarrollo de software	32
4.4. Análisis	34
4.4.1. Análisis de requerimientos	34
4.4.2. Actores	37
4.4.3. Diagrama de contexto	37

4.4.4. Arquetipos	39
4.5 Diseño	40
4.5.1. Arquitectura	40
4.5.2. Modelo de datos	46
4.5.3. Interfaz Grafica	46
4.6. Implementación	50
4.6.1. React	51
4.6.2. Next.js	51
4.6.3. Componentes	51
4.6.4. LangChain	52
4.6.5. LangGraph	52
4.6.6. Integración con el modelo LLaMA 3.3 70B Versatile	52
4.7. Depuración y pruebas	53
4.7.1. Pruebas	54
4.7.2. Dificultades	55
5. Experimentos Y Resultados	56
5.1. Interacción con el usuario	56
5.2. Análisis de resultados	59
5.2.1. Consulta de estaciones climatológicas	59
5.2.2. Resultado del análisis de regresión	60
5.2.3. Generación de reportes EDA	62
5.2.4. Evaluación del flujo completo	63
5.2.5. Resultados generales	64
6. Conclusión y trabajo futuro	65
6.1. Conclusión	65
6.2. Trabajo futuro	66
Bibliografías	67
Anexos	70
A. Manual De Usuario	70
A.1. Objetivo del manual	70
A.2. Acceso al proyecto	70
A.3. Pantalla principal	71

A.4. Realizar una consulta	72
A.5. Consultar estadísticas climatológicas	73
A.6. Solicitar análisis de correlación	74
A.7. Solicitar análisis de regresión	75
A.8. Generar un reporte EDA	76
A.9. Visualización del reporte	77
A.10. Ocultar llamadas a herramientas	78
B. Manual Tecnico	79
B.1. Objetivo	79
B.2. Componentes del sistema	79
B.3. Requisitos técnicos	80
B.4. Instalación del agente	80
B.5. Configuración del agente	81
B.6. Preparación del dataset	82
B.7. Ejecución del agente	83
B.8. Ejecución y visualización de reportes EDA	83
B.9. Generación anticipada de reportes	84
B.10. Instalación y ejecución de la interfaz web	84
B.11. Orden recomendado de ejecución	85
B.12. Capacidades técnicas del agente	85
B.13. Variables climatológicas disponibles	85
B.14. Estructura principal del proyecto	86
B.15. Pruebas automatizadas	87
B.16. Logs y trazabilidad	87
B.17. Consideraciones técnicas	87
B.18. Solución de problemas comunes	88
B.19. Cierre del sistema	88

Índice de figuras

Figura 2 - 1. La arquitectura transformer.....	6
Figura 2 - 2. Ejemplo de entrada y salida de un prompt	7
Figura 2 - 3. El sistema RAG vs sistemas tradicionales	8
Figura 2 - 4. Arquitectura agente simple y arquitectura multi agente	9
<i>Figura 2 - 5. Conexión de datos a través de formatos vectoriales</i>	<i>10</i>
Figura 2 - 6. Arquitectura API REST	15
Figura 3 - 1. Plataforma Conagua	18
Figura 3 - 2. Plataforma NOAA.....	19
Figura 3 - 3. Plataforma Google Earth	20
Figura 3 - 4. Interfaz Aplicación The Weather Channel	21
Figura 3 - 5. Interfaz Aplicación AccuWeather.....	22
Figura 3 - 6. Interfaz Plataforma Apple Weather	23
Figura 3 - 7. Arquitectura agentes llm	24
Figura 3 - 8. Modulos De LangChain	25
Figura 3 - 9. Proceso de RAG	26
Figura 4 - 1. Pipeline de datos del sistema	30
Figura 4 - 2. Modelo Iterativo e Incremental	33
Figura 4 - 3. Diagrama de contexto del sistema Nivel 1	38
Figura 4 - 4. Diagrama de arquetipos del sistema	39
Figura 4 - 5. Diagrama de contenedores del sistema	41
Figura 4 - 6. Diagrama de componentes del sistema	43
Figura 4 - 7. Componentes del agente IA	45
Figura 4 - 8. Vista general de la interfaz del proyecto	47
Figura 4 - 9. Ejemplo de interacción conversacional con el agente	48
Figura 4 - 10. Solicitud de generación de reporte HTML desde la interfaz.....	49
Figura 4 - 11. Visualización del reporte EDA en formato HTML.....	50
Figura 4 - 12. Ejemplo de componente en React	51
Figura 4 - 13. Ejecución del proyecto	53
Figura 5 - 1. Ejecución del prompt	56
Figura 5 - 2. Ejecución del prompt	57
Figura 5 - 3. Ejecución del prompt	58
Figura 5 - 4. Archivo HTML generado.....	58
Figura 5 - 5. Consulta de estaciones climatológicas disponibles mediante el agente ..	59
Figura 5 - 6. Resultado del análisis de regresión lineal	60

Figura 5 - 7. Representación de la regresión lineal de precipitación respecto al mes para la estación el dorado	61
Figura 5 - 8. Generación de un reporte EDA en formato HTML	62
Figura 5 - 9. Reporte de análisis exploratorio de datos generado para la estación 25161 (El dorado)	62
Figura A - 1. Pantalla principal de la aplicación	70
Figura A - 2. Sección historial de chats	71
Figura A - 3. Sección escribirle al agente	72
Figura A - 4. Resultado estaciones disponibles.....	72
Figura A - 5. Resultado promedio de lluvia de la estación acatitan	73
Figura A - 6. Resultado promedio de lluvia de la estación acatitan	74
Figura A - 7. Resultado correlación entre precipitación y temperatura máxima en la estación el dorado.....	74
Figura A - 8. Resultado regresión lineal de precipitación contra mes en la estación el dorado	75
Figura A - 9. Regresión múltiple de precipitación contra mes y año en la estación Acatitan	76
Figura A - 10. Enlace para el archivo HTML generado	77
Figura A - 11. Visualización del reporte EDA generado por el agente	78
Figura B - 1. Componentes principales del agente	80
Figura B - 2. Proceso ETL utilizado para preparar el dataset climatológico	83

Índice de tablas

Tabla 3 - 1. Tabla comparativa del estado del arte.....	27
Tabla 4 - 1. Tabla requisitos funcionales del sistema.	35
Tabla 4 - 2. Tabla requisitos no funcionales del sistema.....	36
Tabla 4 - 3. Tabla requisitos de calidad del sistema.....	36
Tabla 4 - 4. Tabla restricciones del sistema.	37
Tabla 4 - 5. Tabla restricciones del sistema.	54
Tabla B - 1. Tabla requisitos técnicos.....	80
Tabla B - 2. Tabla capacidades técnicas del agente.	85
Tabla B - 3. Tabla variables climatológicas disponibles.	85
Tabla B - 4. Tabla directorios del proyecto.....	86

Capítulo 1

1. Introducción

En este capítulo se presenta la formulación del problema, así como los objetivos tanto generales como específicos del proyecto. También se expone la hipótesis, la justificación y se define la estructura básica del documento.

1.1. Planteamiento del problema

Los sistemas hidrometeorológicos han sido esenciales desde hace mucho tiempo para seguir y estudiar cómo se comporta el clima, ya que permiten registrar datos como lluvia, temperatura y evaporación. Esta información es muy importante para áreas clave, como la agricultura, el manejo del agua, la prevención de desastres y el desarrollo urbano. No obstante, a pesar de que hay bases de datos oficiales muy completas, como la Base de Datos Climatológica de la Comisión Nacional del Agua (CONAGUA) y del Sistema Meteorológico Nacional (SMN), acceder y utilizar esta información de manera efectiva presenta varios problemas. Los datos a menudo están en formatos técnicos difíciles (archivos planos, estructuras tabulares grandes), lo que complica su consulta rápida y exige conocimientos específicos en análisis estadístico y programación para poder interpretarlos correctamente. Esta situación crea una discrepancia entre la cantidad de información disponible y cómo se utiliza realmente. Individuos sin experiencia, responsables de tomar decisiones e incluso expertos pueden tener problemas para:

- Obtener información histórica específica de manera rápida.
- Entender resultados estadísticos de un modo sencillo.
- Reconocer tendencias o relaciones entre diferentes datos.
- Crear informes que se puedan reproducir de forma efectiva.

Como consecuencia, el potencial de los datos hidrometeorológicos oficiales no es plenamente aprovechado, lo que limita la toma de decisiones basada en evidencia y reduce la eficiencia en los procesos de análisis climático.

Aunque los modelos actuales de inteligencia artificial han demostrado avances significativos en la comprensión del lenguaje natural, pero presentan limitaciones importantes en este dominio. En particular, estos modelos no cuentan con acceso

directo, estructurado ni actualizado a bases de datos oficiales específicas como la de CONAGUA. Por ello, sus respuestas pueden carecer de precisión, trazabilidad o respaldo en datos reales, al basarse principalmente en conocimiento general previamente entrenado y no en consultas dinámicas sobre información verificada.

Ante este escenario, surge la necesidad de desarrollar un sistema inteligente que permita consultar, analizar e interpretar datos hidrometeorológicos mediante lenguaje natural, integrando modelos de lenguaje con técnicas estadísticas automatizadas. Dicho sistema deberá garantizar la precisión, trazabilidad y reproducibilidad de los resultados, facilitando así el acceso a información relevante y mejorando la toma de decisiones basada en datos confiables.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un sistema experto que permita la adecuada gestión del análisis estadístico de datos hidrometeorológicos mediante diversas técnicas de Inteligencia Artificial.

1.2.2. Objetivos Específicos

- 1) Diseño y modelado de prompts para transferencia de conocimiento.
- 2) Desarrollo de agentes del sistema.
- 3) Proceso de transferencia de la información.
- 4) Desarrollo de chatbot.

1.3. Hipótesis

La implementación de un sistema de consulta hidrometeorológica basado en un modelo de lenguaje grande (LLM) mejora la accesibilidad, interpretación y eficiencia en el análisis de datos climatológicos oficiales en comparación con métodos tradicionales de consulta manual.

1.4. Justificación

El estudio de información hidrometeorológica es crucial para decidir en campos como el manejo del agua, la agricultura y la protección contra desastres. No obstante, la dificultad de los archivos gubernamentales y la necesidad de habilidades especializadas restringen su uso para muchos individuos.

Frente a esto, la creación de un programa apoyado en modelos de lenguaje (LLM) posibilita simplificar la obtención de estos datos mediante preguntas en lenguaje

cotidiano, disminuyendo los obstáculos técnicos y haciendo más ágil la comprensión de los detalles.

A diferencia de las soluciones de inteligencia artificial existentes, que carecen de conexión directa a orígenes oficiales concretos, esta iniciativa sugiere incorporar un asistente inteligente vinculado a archivos fidedignos, capaz de ejecutar interrogantes y evaluaciones estadísticas de manera automática.

Así, el sistema ayuda a optimizar el estudio de datos climáticos, permitiendo reconocer tendencias y producir información valiosa tanto para especialistas como para público general, robusteciendo las decisiones basadas en hechos.

1.5. Organización del documento

La tesis está dividida en 6 capítulos, los cuales se describen a continuación:

Capítulo 2. Este capítulo presenta el marco teórico de la tesis, los temas principales necesarios para comprender el trabajo realizado.

Capítulo 3. Este capítulo presenta el estado del arte, es decir, la información disponible sobre el tema tratado.

Capítulo 4. Este capítulo describe el proceso de desarrollo del sistema de software producido y las tecnologías necesarias para implementarlo.

Capítulo 5. Este capítulo exhibe los resultados de la evaluación de efectividad del software desarrollado.

Capítulo 6. Este capítulo expone las conclusiones alcanzadas y propone trabajo futuro para esta tesis.

Capítulo 2

2. Marco Teórico

En este capítulo se presenta el marco teórico, el cual reúne los conceptos, teorías y antecedentes que fundamentan la investigación. Su finalidad es sustentar el estudio desde una perspectiva científica y contextualizar el problema dentro del conocimiento existente.

2.1. Sistemas hidrometeorológicos

Los sistemas hidrometeorológicos son conjuntos de herramientas, procesos y fuentes de información destinados a la recopilación, procesamiento, análisis y monitoreo de variables relacionadas con el ciclo hidrológico y las condiciones atmosféricas (Sit et al., 2020).

Los sistemas hidrometeorológicos tienen la finalidad de proporcionar información que apoye en la gestión de los recursos hídricos y la prevención de riesgos naturales, el monitoreo climático y la toma de decisiones ambientales, para ello permiten analizar entre otras variables la precipitación, la temperatura, la humedad relativa, la evaporación, la velocidad del viento y el escurrimiento superficial (Sit et al., 2020).

Los tipos de sistemas hidrometeorológicos más comunes pueden ser las redes de estaciones meteorológicas, los sistemas de monitoreo hidrológico de cuencas, los sistemas de alerta temprana de inundaciones y sequías o las plataformas de consulta climática empleadas por organizaciones gubernamentales como la Comisión Nacional del Agua (CONAGUA).

2.2.1. Datos climatológicos

Los datos climáticos son los registros y observaciones recogidos sobre el clima de una región durante un período dado. Los datos se recogen mediante estaciones meteorológicas, sensores, satélites o cualquier tipo de instrumento de medición y permiten caracterizar el clima que fue y analizar este momento de la variabilidad climática a lo largo del tiempo (World Meteorological Organization [WMO], 2023).

Los datos climatológicos son indispensables para el estudio de los fenómenos atmosféricos, la gestión de los recursos naturales, la planificación agrícola, la prevención del riesgo a los eventos extremos, el desarrollo de los modelos de predicción climática. Su interpretación permite identificar patrones, tendencias y anomalías que ayuden a una mejor comprensión de las condiciones ambientales y sus interacciones y posibles consecuencias sobre la sociedad y los ecosistemas.

Los datos climatológicos más relevantes son la temperatura del aire, la precipitación, la humedad relativa, la presión atmosférica, la velocidad y la dirección del viento, la radiación solar y la evaporación. Son datos que suelen ser utilizados por organismos estatales, centros de investigación y sistemas de monitoreo ambiental para generar información que soporte la toma de decisiones, así como para la elaboración de materiales de formación en diferentes sectores.

2.2. Inteligencia artificial

La inteligencia artificial (IA) es un campo de la informática que se dedica a crear sistemas que pueden llevar a cabo tareas que, por lo general, necesitarían inteligencia humana. Esto incluye cosas como aprender, razonar, resolver problemas, reconocer patrones y entender el lenguaje natural (Russell & Norvig, 2021).

La meta primordial de la inteligencia artificial es diseñar sistemas que puedan examinar datos, tomar determinaciones y producir respuestas basadas en la información que se les proporciona. Dado el progreso en la potencia de cálculo y la cantidad de datos disponibles, la IA se ha incorporado en distintos ámbitos como la medicina, la enseñanza, las finanzas, la industria, el transporte y las ciencias del medio ambiente. Esto ha facilitado la automatización de procesos y ha optimizado la toma de decisiones.

Algunas de las divisiones más importantes de la inteligencia artificial son el aprendizaje automático (Machine Learning), el aprendizaje profundo (Deep Learning), la visión por computadora, los sistemas expertos y el procesamiento del lenguaje natural (Natural Language Processing o NLP). Estas herramientas capacitan a los sistemas para interpretar información intrincada, aprender de lo vivido anteriormente y ajustarse a circunstancias nuevas.

2.3. Teoría agentica

2.3.1. Agentes de Inteligencia Artificial

Los agentes de inteligencia artificial se pueden definir como entidades de software autónomas que se diseñan para llevar a cabo tareas dirigidas a objetivos dentro de entornos digitales definidos. Estos agentes se distinguen por poder percibir entradas, ya sean estructuradas o no, por razonar sobre la información del contexto y por poder iniciar acciones para lograr metas concretas, actuando a menudo en lugar de usuarios humanos u otros subsistemas.

A diferencia de los scripts de automatización habituales, que se adhieren a flujos de trabajo predeterminados, los agentes de IA demuestran inteligencia reactiva y cierto nivel de adaptabilidad, lo que les posibilita interpretar entradas cambiantes y modificar sus respuestas según sea necesario (Sapkota et al., 2025).

2.3.2. LLM (Large Language Models)

Raschka (2024) en su libro, define a los LLM como modelos de redes neuronales profundas que se han desarrollado en los últimos años. Inauguraron una nueva era para el procesamiento del lenguaje natural (PLN) entrenado para predecir el siguiente token (palabra/subpalabra) en una secuencia. En la práctica, los LLM:

- Interpretan lenguaje natural (preguntas del usuario).
- Generan texto estructurado (p. ej., JSON, reportes).
- Ayudan a explicar resultados estadísticos.

2.3.2.1. Transformers

Según Raschka (2024), la arquitectura transformer consta de dos submódulos: un codificador y un decodificador, como se aprecia en la figura 2 - 1.

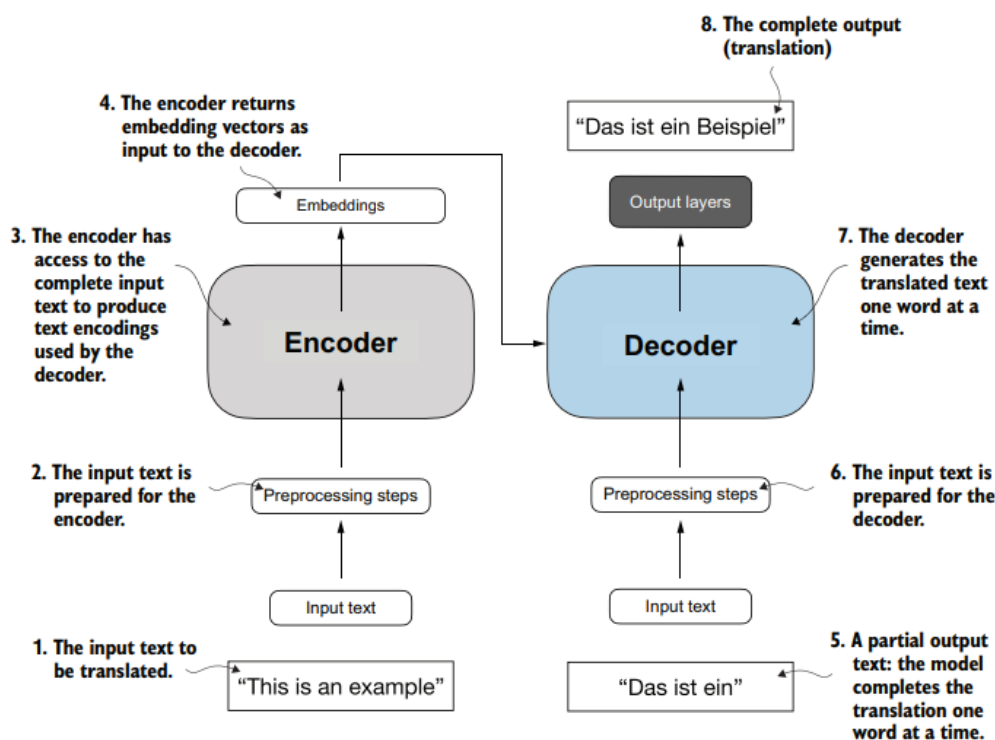


Figura 2 - 1. La arquitectura transformer (Raschka, 2024).

El módulo **codificador** procesa el texto de entrada y lo codifica en una serie de representaciones numéricas o vectores que capturan la información contextual de la entrada. El módulo **decodificador** toma estos vectores codificados y genera el texto de salida. En una tarea de traducción, por ejemplo, el codificador convertiría el texto del

idioma de origen en vectores, y el decodificador decodificaría estos vectores para generar texto en el idioma de destino.

Tanto el codificador como el decodificador están compuestos por muchas capas conectadas mediante un mecanismo denominado autoatención (self-attention).

2.3.3. Prompt

Un prompt es la entrada que proporcionas, normalmente texto, cuando interactúas con un modelo de IA como ChatGPT o Midjourney. El prompt funciona como un conjunto de instrucciones que el modelo utiliza para predecir la respuesta deseada (Taylor, 2024).

Input:

```
Can I have a list of product names for a pair of shoes that  
can fit any foot size?
```

Output:

```
Certainly! Here's a list of product names for a pair of shoes  
that can fit any foot size:
```

```
UniFit SoleStrides
```

Figura 2 - 2. Ejemplo de entrada y salida de un prompt (Taylor, 2024).

En la figura 2 – 2, demuestra que incluso un *prompt* simple puede producir resultados satisfactorios con un esfuerzo mínimo. Conforme los modelos de lenguaje continúan evolucionando, la probabilidad de obtener respuestas adecuadas desde el primer intento aumenta considerablemente. En tareas esporádicas o de uso único, donde no se requiere repetir el proceso, un enfoque basado en instrucciones simples puede ser suficiente para alcanzar el resultado esperado.

2.3.4. Prompt Engineering

La ingeniería de prompts (Prompt Engineering) es el proceso de descubrir indicaciones que produzcan resultados útiles o deseados de manera confiable (Taylor, 2024), No se trata solo de “hacer una pregunta”, sino de formularla de manera estratégica:

especificando el contexto, el objetivo, el formato de respuesta, el tono y cualquier restricción necesaria. Un buen prompt puede marcar una gran diferencia en la calidad del resultado, ya que guía al modelo para interpretar correctamente la intención del usuario.

2.3.5. RAG (Retrieval-Augmented Generation)

La generación aumentada por recuperación (RAG) se ha propuesto como un nuevo marco para la IA que busca integrar conocimiento adicional, como datos organizacionales, y generar resultados que puedan vincularse a ese conocimiento, donde se observa en la figura 2 - 3.

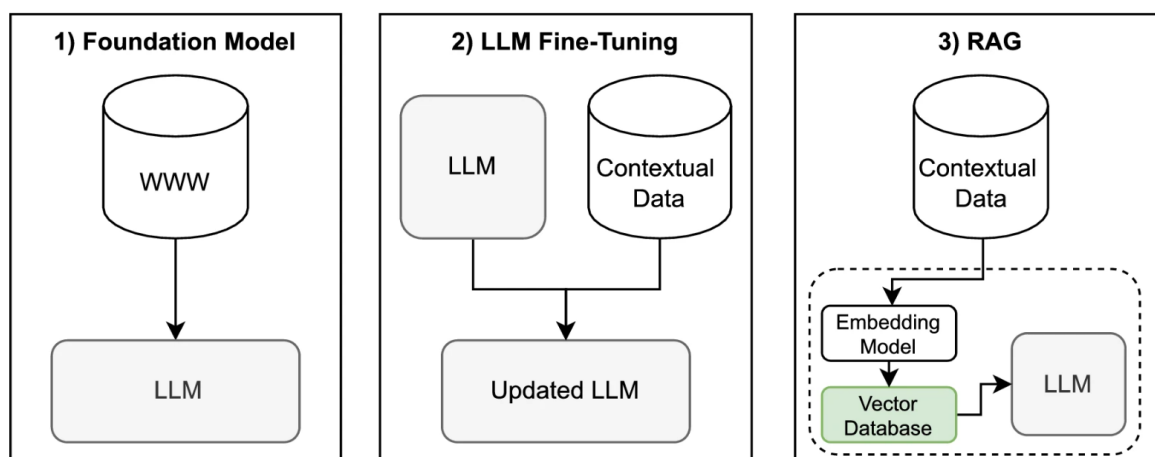


Figura 2 - 3. El sistema RAG vs sistemas tradicionales (Klesel & Wittmann, 2025).

Esto permite a los usuarios acceder a información dentro de una organización y reduce el riesgo de alucinaciones. Esta nueva arquitectura ofrece avances importantes en comparación con las arquitecturas anteriores y presenta nuevos desafíos para la investigación y el ámbito académico (Klesel & Wittmann, 2025).

2.3.6. Uso de herramientas (Tools)

En el contexto de los agentes de inteligencia artificial, las herramientas (tools) representan cualquier función que el modelo puede invocar. Estas permiten que el agente interactúe con fuentes de datos externas, ya sea extrayendo o enviando información hacia dichas fuentes.

Un ejemplo de una persona de agente y sus herramientas asociadas es un redactor profesional de contratos. A este agente se le asigna una persona que define su rol y los tipos de tareas que debe realizar. Además, se le proporcionan herramientas

relacionadas con agregar notas a un documento, leer un documento existente o enviar un correo electrónico con el borrador final (Masterman et al., 2024).

2.3.7. Arquitectura multi agente

Estas arquitecturas involucran dos o más agentes, donde cada agente puede utilizar el mismo modelo de lenguaje o un conjunto de diferentes modelos de lenguaje. Los agentes pueden tener acceso a las mismas herramientas o a herramientas distintas, y cada uno generalmente cuenta con su propia personalidad (persona), como vemos en la figura 2 - 4.

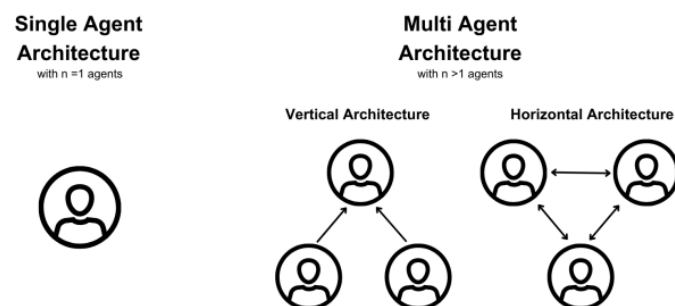


Figura 2 - 4. Arquitectura agente simple y arquitectura multi agente (Masterman et al., 2024).

Las arquitecturas multi agente pueden presentar una gran variedad de organizaciones con distintos niveles de complejidad. En este trabajo, se dividen en dos categorías principales: vertical y horizontal. Es importante tener en cuenta que estas categorías representan dos extremos de un espectro, ya que la mayoría de las arquitecturas existentes se sitúan en algún punto intermedio entre ambos (Masterman et al., 2024).

2.3.8. Base de datos vectorial

Las bases de datos vectoriales almacenan embeddings de texto, lo que permite una recuperación y procesamiento eficientes. Estas bases de datos respaldan sistemas de Generación Aumentada por Recuperación (RAG) al permitir la obtención de información relevante durante la generación de respuestas.

Todos estos embeddings almacenados, dentro de un enfoque RAG, combinan métodos basados en recuperación y generación para producir resultados más precisos y con mayor riqueza contextual que los modelos tradicionales (Tahseen et al., 2025).

2.3.9. Integración de LLM con bases de datos

La integración de modelos de lenguaje de gran escala (LLM) con bases de datos representa un avance significativo en el desarrollo de sistemas inteligentes orientados a la consulta y análisis de información. A diferencia de los modelos tradicionales, que generan respuestas basadas únicamente en conocimiento previamente entrenado, los

LLM integrados con fuentes de datos permiten realizar consultas dinámicas sobre información actualizada y verificable en la figura 2 – 5 podremos visualizarlo.

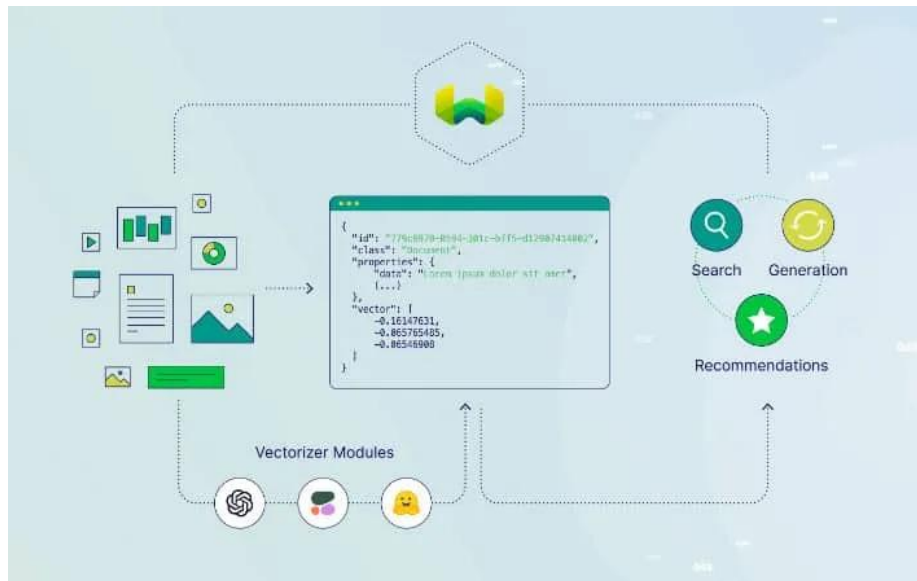


Figura 2 - 5. Conexión de datos a través de formatos vectoriales (Fuente: elaboración propia)

Este método aprovecha la habilidad de los LLM para entender el lenguaje natural y convertirlo en acciones concretas, como peticiones a bases de datos, selección de datos o realización de análisis. Así, estos modelos actúan de puente entre el usuario y los sistemas de datos, haciendo más fácil acceder a información complicada sin requerir conocimientos técnicos especializados (Lewis et al., 2020).

Una de las estrategias más importantes en esta unión es la de emplear arquitecturas que mezclan modelos generativos con métodos de búsqueda de información, como la Generación Aumentada por Recuperación (RAG). Con estos sistemas, el modelo puede consultar fuentes externas mientras crea sus respuestas, lo que eleva la exactitud, disminuye las invenciones y ofrece resultados fundamentados en información verídica (Zaharia et al., 2022).

2.3.10. Sistemas de consulta en lenguaje natural (NLQ)

Los sistemas de consulta en lenguaje natural (Natural Language Querying, NLQ) permiten a los usuarios interactuar con bases de datos mediante lenguaje humano, eliminando la necesidad de utilizar lenguajes formales como SQL. Su meta fundamental consiste en convertir el deseo del operador en preguntas organizadas capaces de correr dentro de repositorios de información.

Con el progreso de los modelos lingüísticos masivos (LLM), las estructuras NLQ han cambiado mucho, avanzando en entender el contexto, la duda y la meta del usuario. Estas herramientas dejan crear preguntas mejores y flexibles para áreas distintas, ayudando a tocar datos difíciles sin pedir saber técnico profundo (Rajkumar et al., 2022).

En dicho marco, las estrategias actuales fusionan modelos generativos junto a procedimientos operativos aplicados sobre repositorios de datos auténticos, facilitando no únicamente la decodificación de consultas formuladas en lenguaje natural, sino además la puesta en marcha de acciones tales como selección, suma total y examen de información. Esta expansión extiende las potencias de los sistemas NLQ hacia situaciones más intrincadas, tal como el estudio preliminar de conjuntos de datos y la creación espontánea de documentos informativos (Li et al., 2023).

2.4. Análisis de datos

El análisis de datos es el proceso de inspeccionar, transformar e interpretar conjuntos de datos con el objetivo de obtener información útil, identificar patrones y apoyar la toma de decisiones. Mediante la aplicación de técnicas estadísticas, computacionales y analíticas, es posible convertir grandes volúmenes de datos en conocimiento que facilite la comprensión de fenómenos y la resolución de problemas (Provost & Fawcett, 2023).

Actualmente, el análisis de datos es utilizado en diversos sectores como la salud, la industria, las finanzas, la investigación científica y las ciencias ambientales. Su aplicación permite detectar tendencias, realizar predicciones y generar información estratégica para la toma de decisiones basada en evidencia.

2.4.1. ETL

Una definición simple de **Extracción, Transformación y Carga (ETL)** podría ser: “el conjunto de procesos utilizados para obtener datos de sistemas OLTP (Procesamiento de Transacciones en Línea) y trasladarlos hacia un almacén de datos” (Yulianto, 2019). Este proceso se compone de tres fases principales: extracción, transformación y carga.

2.4.1.1. Extracción

La etapa de **Extracción (Extract)** implica en obtener los datos desde una o varias fuentes de origen. Estas fuentes pueden incluir bases de datos, archivos de texto, hojas de cálculo, servicios web, sensores, sistemas externos o repositorios institucionales.

Durante esta etapa, el objetivo principal es recopilar la información necesaria sin modificar significativamente su contenido original. También es importante identificar el formato, estructura, volumen y procedencia de los datos, ya que estas características determinarán cómo deberán procesarse posteriormente.

2.4.1.2. Transformación

El paso de **Transformación (Transform)** implica tratar los elementos recolectados para hacerlos uniformes, organizados y listos para ser estudiados. Este momento resulta muy relevante cuando los materiales vienen de orígenes distintos o muestran irregularidades en cómo se presentan.

En medio del proceso transformador se ejecutan actividades tales como limpiar registros dobles o erróneos, corregir formatos, convertir tipos de datos, tratar valores ausentes, normalizar nombres y estandarizar variables.

2.4.1.3. Carga

La etapa de Carga (Load) implica guardar los registros modificados hacia su ubicación definitiva. Dicha ubicación podría constituir una base de datos, un repositorio de datos, una arquitectura distribuida o un esquema de archivo diseñado para facilitar el tratamiento y estudio posterior.

Lo que se busca en esta fase es guardar los resultados trabajados bajo una estructura ordenada, fácil de encontrar y con buen rendimiento. Según requiera el entorno técnico, la introducción de material puede hacerse de modo total, sustituyendo todo lo previo, o de manera progresiva, sumando solo lo reciente.

2.4.2. Preprocesamiento de datos

El preprocesamiento de datos permite crear conjuntos de datos precisos y útiles al corregir problemas como errores, valores incompletos u otras inconsistencias. Los métodos más utilizados se clasifican en tres principales categorías: limpieza de datos, transformación de datos y reducción de datos.

Cuando no se aplican técnicas de preprocesamiento a los datos en bruto, o cuando se emplean métodos inadecuados, las técnicas de análisis de datos por sí solas no logran alcanzar un rendimiento satisfactorio (Çetin & Yıldız, 2022).

2.4.3. Calidad de datos

La calidad de los datos se refiere al grado en que un conjunto de datos cumple con los requisitos necesarios para ser considerado adecuado para su uso en análisis, toma de decisiones y generación de conocimiento. Un alto nivel de calidad de datos es fundamental para asegurar resultados confiables, dado que datos incorrectos, incompletos o inconsistentes pueden afectar significativamente la precisión de los análisis (Batini & Scannapieco, 2020).

2.4.4. Análisis Exploratorio de Datos (EDA)

El Análisis Exploratorio de Datos (EDA) es un enfoque utilizado para analizar conjuntos de datos. Principalmente, el EDA emplea métodos de visualización de datos y, en muchos casos, modelos estadísticos para: (i) evaluar la estructura general de un conjunto de datos, (ii) obtener resúmenes descriptivos de la información y (iii) proporcionar una base para la formulación de modelos.

El EDA incluye la verificación de la calidad de los datos, el cálculo de estadísticas descriptivas y la visualización de los datos. La calidad de los datos se evalúa en función de errores, valores atípicos (outliers) y datos faltantes (Galatro & Dawe, 2023).

2.4.5. Correlación estadística

La correlación estadística es una medida que permite determinar la relación existente entre dos variables, indicando el grado en que estas cambian de manera conjunta. Dicho principio resulta clave al examinar datos, pues permite detectar modelos, evoluciones y vínculos potenciales existentes en una colección de cifras.

En especial, la correlación lineal simple describe vínculo entre dos magnitudes cuantitativas, donde modificación en una puede implicar alteración en la segunda. Si ambas crecen o decrecen al mismo tiempo, se afirma que hay correlación positiva; por otro lado, si una sube mientras la baja, la correlación resulta negativa.

Para medir dicha vinculación, se aplican índices de asociación, tal cual el índice de Pearson, cuya magnitud varía desde menos uno hasta uno positivo. Una cifra próxima a uno denota una alta asociación directa; una cifra cercana a menos uno señala una alta asociación inversa; mientras que una cifra próxima a cero apunta hacia la falta de vínculo lineal entre ambas magnitudes (James et al., 2021).

2.4.6. Regresión lineal

La regresión lineal simple hace honor a su nombre: es un enfoque muy sencillo para predecir una respuesta cuantitativa Y a partir de una sola variable predictora X . Este modelo asume que existe una relación aproximadamente lineal entre X y Y .

Matemáticamente, esta relación se puede expresar como:

$$Y \approx \beta_0 + \beta_1 X$$

El símbolo “ $\approx$ ” puede interpretarse como “se modela aproximadamente como”. En ocasiones, esta expresión se describe diciendo que estamos regresando Y sobre X (o Y en función de X) (James et al., 2023).

2.4.7. Regresión múltiple

La regresión múltiple es una extensión de la regresión lineal simple que permite modelar la relación entre una variable dependiente cuantitativa y dos o más variables independientes. Este enfoque se usa mucho en el análisis de datos, porque permite tener en cuenta varios factores a la vez, lo que da modelos más exactos y representativos de fenómenos complicados (James et al., 2021).

Matemáticamente, la regresión múltiple se expresa como:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

donde Y es la variable dependiente, $X_1, X_2, \dots, X_n$ son las variables independientes, y $\beta_0, \beta_1, \dots, \beta_n$ representan los coeficientes del modelo. Cada coeficiente indica el efecto individual de una variable independiente sobre la variable dependiente, manteniendo constantes las demás variables.

Este tipo de modelo es especialmente útil en contextos donde múltiples variables influyen en un resultado, como en el análisis hidrometeorológico, donde factores como la temperatura, la precipitación, la humedad y la evaporación pueden estar interrelacionados. La regresión múltiple permite identificar la contribución relativa de cada variable y analizar su impacto conjunto.

En aplicaciones modernas, la regresión múltiple también se utiliza como base para modelos más avanzados en aprendizaje automático, sirviendo como una herramienta fundamental para la predicción, el análisis de tendencias y la toma de decisiones basada en datos (Montgomery et al., 2021).

2.4.8. Trazabilidad y reproducibilidad

La trazabilidad y la reproducibilidad son principios clave en el análisis de datos, sobre todo en situaciones donde la fiabilidad y la transparencia de los resultados son cruciales. La trazabilidad se refiere a la capacidad de rastrear el origen de los datos, así como cada uno de los procesos y transformaciones aplicadas durante su análisis. Por su parte, la reproducibilidad implica que un análisis pueda ser replicado por otros usuarios o sistemas, obteniendo los mismos resultados bajo las mismas condiciones (Gundersen & Kjensmo, 2021).

2.5. Tecnologías web

2.5.1. Aplicaciones web

Es un software que se ejecuta en el navegador web. Las empresas tienen que intercambiar información y proporcionar servicios de forma remota. Utilizan aplicaciones web para comunicarse con los clientes cuando lo necesiten y de una forma segura. Las funciones más comunes de los sitios web, como los carros de compra, la búsqueda y el filtrado de productos, la mensajería instantánea y los canales de noticias de las redes sociales, tienen el mismo diseño que las aplicaciones web. Le permiten acceder a funcionalidades complejas sin la necesidad de instalar o configurar un software (Amazon Web Services, s. f.).

2.5.2. API

Las API son una forma sencilla de conectarse a, integrarse con y extender un sistema de software. Más específicamente, las API se utilizan para construir sistemas de software distribuidos, cuyos componentes están débilmente acoplados, se puede apreciar mejor en la figura 2 - 6.

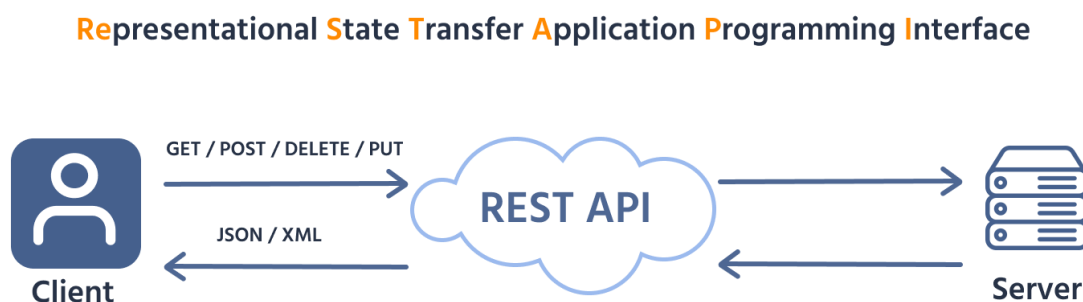


Figura 2 - 6. Arquitectura API REST (Kiesler & Schiffner, 2023).

Normalmente se implementan como servicios web y proporcionan recursos de datos a través de una pila tecnológica web. Las aplicaciones típicas que utilizan API incluyen aplicaciones móviles, aplicaciones en la nube, aplicaciones web o dispositivos inteligentes .

Los desarrolladores también consideran importante que una API pueda extenderse e integrarse en un framework existente. Sin embargo, no se identificaron tendencias claras entre los criterios, ya que intervienen muchos factores al momento de elegir una API. Por lo tanto, se requiere más investigación para proporcionar orientación a los (futuros) desarrolladores de software sobre cómo pueden mejorar el diseño de sus APIs (Kiesler & Schiffner, 2023).

2.5.3. React.js

React, también conocido como React.js o ReactJS, es una biblioteca de frontend basada en JavaScript para construir interfaces de usuario o componentes de UI. React es gestionado por Facebook, una comunidad de desarrolladores y empresas.

En un estudio realizado por Venkat en 2021, se explicó que la tecnología de desarrollo web antes de 2015 se centraba en la creación de scripts y el renderizado. En ese momento, HTML y CSS se utilizaban para el frontend, mientras que el backend utilizaba PHP (Lazuardy & Anggraini, 2022).

2.5.4. Next.js

El framework Next.js es un marco de trabajo basado en React que cuenta con una gran variedad de características, por lo que se utiliza como tecnología en la construcción de aplicaciones frontend (Lazuardy & Anggraini, 2022).

Capítulo 3

3. Estado del arte

En este capítulo se analizan diversas plataformas, herramientas y trabajos de investigación relacionados con la gestión, consulta y análisis de datos hidrometeorológicos. El propósito es determinar las tecnologías y los métodos que se emplean hoy en día para acceder, procesar y mostrar datos climáticos, además de valorar sus atributos, beneficios y restricciones principales.

De igual modo, se examinan las propuestas que se basan en inteligencia artificial, modelos de lenguaje y sistemas para preguntar en lenguaje natural con el fin de simplificar el acceso a grandes cantidades de información. También, se estudian las soluciones dirigidas al análisis de datos, la creación de informes y la ayuda en la toma de decisiones a partir de datos ambientales y meteorológicos.

Por último, se detectan ámbitos con potencial y puntos que se pueden mejorar en las plataformas y las herramientas que se han analizado. Este estudio posibilita fijar los beneficios competitivos y las contribuciones del proyecto, en especial al integrar agentes de inteligencia artificial para consultar, analizar y generar automáticamente información hidrometeorológica a partir de fuentes oficiales.

3.1. Plataformas de datos hidrometeorológicos

Las plataformas de datos hidrometeorológicos, vistas como sistemas especializados, abordan la recopilación, almacenamiento, procesamiento, consulta y representación gráfica de datos relacionados con variables climáticas e hidrológicas. Es mas se estiman una herramienta de decision primordial en el manejo de recursos hidricos, la agricultura, la proteccion civil, la investigacion cientifica y la observacion ambiental. Porque hacen posible el acceso a vastas cantidades de datos historicos y sobre las condiciones presentes (Vaughn et al. , 2023).

De igual manera, las plataformas de datos hidrometeorologicos sirven como una base para crear sistemas más sofisticados enfocados en el analisis de datos, la prediccion de fenomenos meteorológicos y la generación de conocimiento mediante tecnicas de inteligencia artificial y aprendizaje automatico. Entre las aplicaciones integradas en estas clases de sistemas se encuentran: modelos de prediccion climatica, sistemas de alerta temprana, plataformas de apoyo a la toma de decisiones y asistentes inteligentes para buscar informacion meteorologica (DePodwin et al. , 2023).

En este marco, disponemos de varias plataformas que nos dan acceso a datos hidrometeorológicos públicos y científicos. Las gestionadas por la Comisión Nacional del Agua CONAGUA y el Servicio Meteorológico Nacional SMN aquí en México resaltan, al igual que las ofrecidas por la National Oceanic and Atmospheric Administration NOAA de los Estados Unidos. Instituciones así proveen acceso a historiales, observaciones climáticas, datos climatológicos, y productos especiales que se usan bastante en estudios ambientales.

Por otro lado, plataformas tales como Google Earth representan una fuente auxiliar de información geoespacial, que facilita ver fenómenos ambientales y poner en contexto espacial los datos que se recopilan de estaciones meteorológicas y de monitoreo. Dada su habilidad para visualizar, integrar informaciones geográficas y analizar el territorio, bastantes estudios ven en estas herramientas un recurso de gran valor para la investigación de fenómenos del clima y del medio ambiente.

Por lo tanto, examinar estas plataformas es interesante para este estudio, puesto que revela las características actuales y los obstáculos en cuanto a acceso, entendimiento y uso de la información. Estas consideraciones funcionan como base para la creación del proyecto, el cual busca simplificar la revisión y estudio de datos hidrometeorológicos, utilizando inteligencia artificial y procesamiento de lenguaje natural.

3.1.1. CONAGUA

En México, la CONAGUA funciona como la fuente principal para información hidrometeorológica, ofreciendo archivos históricos de estaciones climáticas que abarcan datos de precipitación, temperatura y evaporación. Esta plataforma representa el antecedente más importante para el sistema, dado que la información empleada por el sistema se extrae directamente de estos archivos, la plataforma se aprecia en la figura 3 - 1.



Figura 3 - 1. Plataforma Conagua (Fuente: Captura de pantalla de la plataforma CONAGUA, tomada en 2026.).

En el caso específico de los datos históricos que se usan en el proyecto las variables de precipitación, evaporación, temperatura máxima y temperatura mínima, El SMN indica que provienen de las estaciones climatológicas convencionales de la Red Nacional de CONAGUA. Esos registros son reportados por los Organismos de Cuenca y Direcciones Locales y se almacenan en las bases de datos climatológicas del SMN. Actualmente su portal permite consultar información histórica de más de 5,000 estaciones, aunque no todas siguen operando.

No obstante, acceder a esta información y utilizarla demanda procedimientos manuales y experiencia técnica, lo que restringe su disponibilidad para quienes no poseen conocimientos especializados .

3.1.2. NOAA

La National Oceanic and Atmospheric Administration (NOAA) es una de las principales fuentes de datos climáticos abiertos a nivel internacional. Ofrece mucha información sobre el clima a través de servicios en la web, aplicaciones de programación y bases de datos históricos de muchas estaciones en todo el mundo. Lo que destaca de la NOAA es que pone a disposición de todos los datos de manera estandarizada, lo que facilita su uso en diferentes sistemas de análisis, como vemos en la figura 3 – 2.

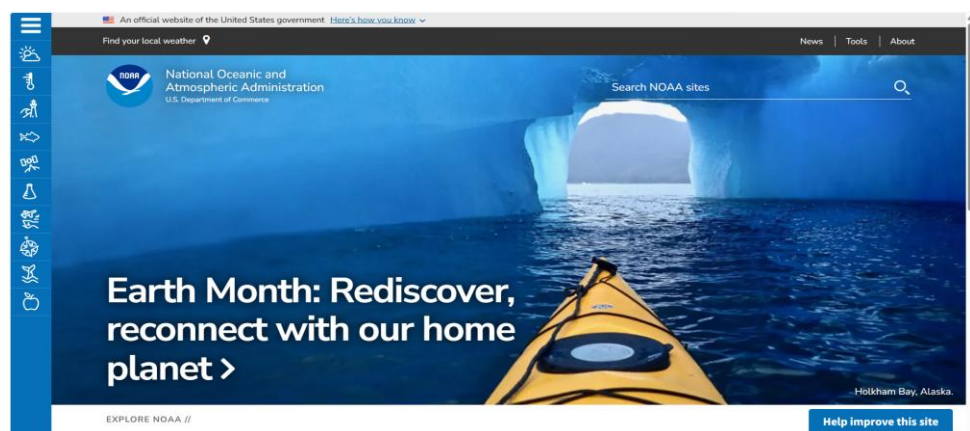


Figura 3 - 2. Plataforma NOAA (Captura de pantalla de la plataforma NOAA, tomada en 2026.).

La NOAA recopila sus datos de una red vasta y global de sistemas de observación masivos. En cuanto a los datos del clima y tiempo atmosférico, esta agencia fusiona estaciones terrestres, satélites, radares, globos meteorológicos, aeronaves, buques y boyas oceánicas entre otras informaciones disponibles. Diariamente la NOAA maneja cientos de millones de observaciones para crear sus pronósticos y variados productos meteorológicos.

Sin embargo, para acceder a estos datos, es necesario tener conocimientos técnicos en programación y saber manejar aplicaciones de programación, lo que puede ser un obstáculo para personas que no son expertas en estas áreas.

En cuanto a al proyecto, la NOAA es un precedente importante en términos de datos abiertos. Sin embargo, el proyecto busca avanzar aún más al incorporar capacidades de análisis automático y permitir consultas en lenguaje natural. Esto significa que el objetivo es buscar como hacer que los datos sean más accesibles y fáciles de analizar para todos, no solo para los expertos en programación.

3.1.3. Google Earth

Por otro lado, plataformas en la nube como Google Earth han cambiado la forma en que analizamos datos geospaciales. Ahora es posible procesar grandes cantidades de información gracias a la infraestructura distribuida.

Google Earth nos permite acceder a conjuntos de datos globales y ofrece herramientas para hacer análisis espaciales y temporales a gran escala. Esto es muy útil en estudios ambientales y climáticos, podemos verlo directamente como se ve en la figura 3 - 3.

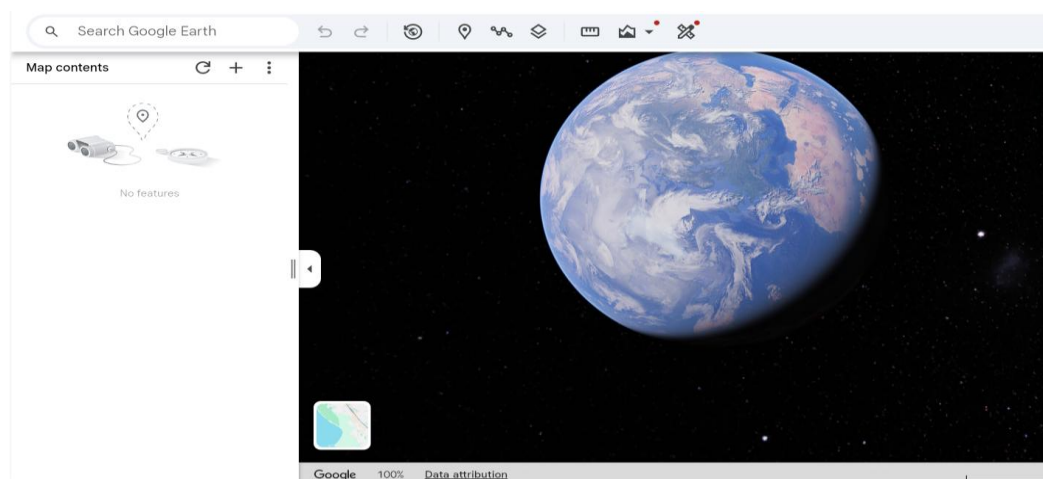


Figura 3 - 3. Plataforma Google Earth (Fuente: Captura de pantalla de la plataforma Google Earth, tomada en 2026.).

Google Earth obtiene su información principalmente de imágenes satelitales, fotografías aéreas, modelos 3D, datos geospaciales y Street View. Google explica que sus imágenes de satélite y aéreas son capturadas por cámaras instaladas en satélites y aeronaves, y cada imagen corresponde a una fecha y hora específicas.

Sin embargo, usar esta plataforma es complicado porque requiere conocimientos avanzados en programación y manejo de datos geospaciales. Esto hace que sea difícil de usar para personas sin formación técnica.

En este contexto, el proyecto retoma la idea de procesar datos de manera eficiente, pero de forma simplificada. Se enfoca en la interacción a través de lenguaje natural y en el análisis automatizado sin necesidad de conocimientos especializados.

3.2. Herramientas y aplicaciones basadas en interacción con datos climáticos

En los últimos años, se ha visto grandes avances en el desarrollo de aplicaciones web y sistemas interactivos. Estos han mejorado la manera en que las personas acceden y consultan información hidrometeorológica. Antes, solo había plataformas tradicionales para visualizar esta información. Pero ahora, tenemos sistemas más dinámicos. Estos sistemas incluyen interfaces conversacionales, asistentes virtuales y servicios inteligentes que se enfocan en el usuario.

3.2.1. The Weather Channel

The Weather Channel desarrollada por IBM es una aplicación la cual proporciona información meteorológica en tiempo real, incluyendo pronósticos diarios, alertas climáticas y visualizaciones gráficas del estado del tiempo, como se ve en la figura 3 - 4. Esta aplicación se caracteriza por su interfaz amigable y accesible para el usuario promedio, permitiendo consultar rápidamente condiciones como temperatura, probabilidad de lluvia y velocidad del viento (Vaughn et al., 2023).

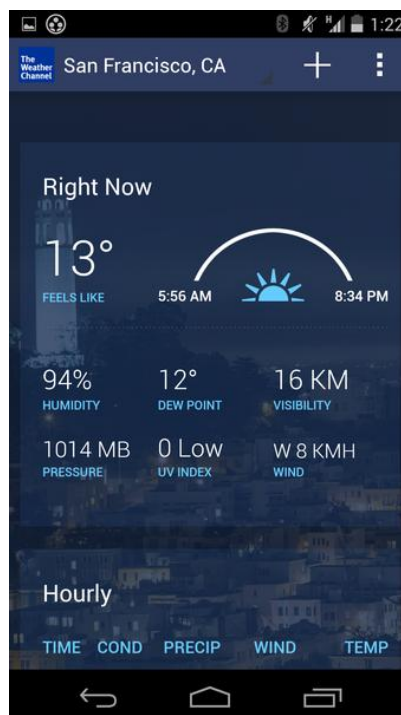


Figura 3 - 4. Interfaz Aplicación The Weather Channel (Fuente: The Weather Channel).

The Weather Channel obtiene su información a través de The Weather Company, que integra datos de múltiples fuentes meteorológicas públicas y privadas. Entre ellas se

incluyen radares, satélites, estaciones meteorológicas oficiales, estaciones meteorológicas personales, aeronaves, boyas y otras redes de observación.

No obstante, su funcionalidad se centra en la presentación de datos ya procesados, sin ofrecer herramientas para el análisis histórico o la exploración de patrones complejos. En este sentido, aunque facilita el acceso a la información, no permite al usuario realizar análisis personalizados como los que propone en el proyecto, mediante consultas en lenguaje natural.

3.2.2. AccuWeather

AccuWeather (DePodwin et al., 2023), es un servicio que ayuda a predecir el tiempo de manera avanzada. Utiliza sus propios modelos y tiene una forma especial de medir la temperatura, llamada “RealFeel Temperature”, que intenta mostrar cómo realmente se siente el calor o el frío. También ofrece mapas, gráficos y notificaciones personalizadas para que la gente pueda obtener la información que necesita de una manera más fácil, como podemos visualizar en la figura 3 - 5.

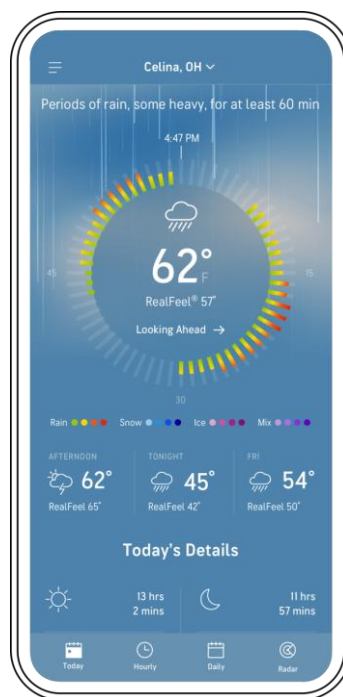


Figura 3 - 5. Interfaz Aplicación AccuWeather (DePodwin et al., 2023).

AccuWeather obtiene y procesa información meteorológica a partir de múltiples fuentes. La compañía misma señala que incorpora información de gobiernos, asociados meteorológicos, observaciones desde tierra, embarcaciones, aviones, informes colaborativos, satélites y radares de múltiples naciones, sumando una vasta compilación de modelos de predicción.

No obstante, emulando otras apps de índole parecida, su principal fin es la pronosticación del clima y la exhibición de datos en tiempo real y a futuro, empero no

confiere al usuario la potestad de manipular datos pretéritos ni de realizar disecciones estadísticas exhaustivas.

A un costado, la iniciativa aspira a constituirse cual apéndice de apoyo, inyectando análisis automáticos y la facultad de escudriñar en registros cronológicos.

3.2.3. Apple Weather

Apple Weather es una aplicación que viene con iOS y mucha gente la usa. Esta herramienta muestra información sobre el clima de manera fácil de entender, con gráficos modernos, animaciones y predicciones detalladas. Lo que más llama la atención de Apple Weather es que presenta la información de forma clara, como el pronóstico por hora, la calidad del aire, las precipitaciones y las condiciones del entorno; podemos ver la aplicación en la figura 3 - 6.



Figura 3 - 6. Interfaz Plataforma Apple Weather (Fuente: Captura de pantalla propia de la aplicación Apple Weather, tomada en 2026).

Apple Weather obtiene su información combinando modelos meteorológicos, datos de estaciones, radares y servicios meteorológicos oficiales de distintos países. Apple identifica entre sus fuentes a organismos como NOAA/National Weather Service, Environment and Climate Change Canada, Deutscher Wetterdienst, ECMWF/The Met Office, Japan Meteorological Agency y Météo-France.

Asimismo, Apple incorpora al Servicio Meteorológico Nacional (SMN) de CONAGUA entre sus fuentes, particularmente para datos sobre alertas meteorológicas severas.

No obstante, al igual que otras aplicaciones, se centra en presentar información ya procesada y en asegurar una buena experiencia para el usuario, pero no le ofrece la posibilidad de realizar análisis personalizados con datos históricos ni de formular preguntas complejas de forma natural.

3.3. Integración con inteligencia artificial

Últimamente la inteligencia artificial, cambió la forma en que los humanos se relacionan con la información. Grandes modelos de lenguaje como GPT-4 y LLaMA han resultado, y mucho, de ayuda. Tienen esta capacidad de tomar preguntas dichas en lenguaje común, y luego soltar respuestas que parecen sacadas de un libro, basándose en datos que ya estaban organizados.

3.3.1. LLMs

Los Modelos de Lenguaje de Gran Escala (LLMs, por sus siglas en inglés) son sistemas complejos de inteligencia artificial capaces de comprender, generar y traducir el lenguaje humano. Al analizar grandes cantidades de datos textuales, estos modelos aprenden patrones lingüísticos que les permiten realizar tareas como redacción, conversación y resumen como vemos en la figura 3 - 7. Los agentes basados en LLMs (agentes LLM) amplían aún más estas capacidades, permitiéndoles procesar interacciones con usuarios y ejecutar operaciones complejas en diversos entornos de tareas (Yan et al., 2025).

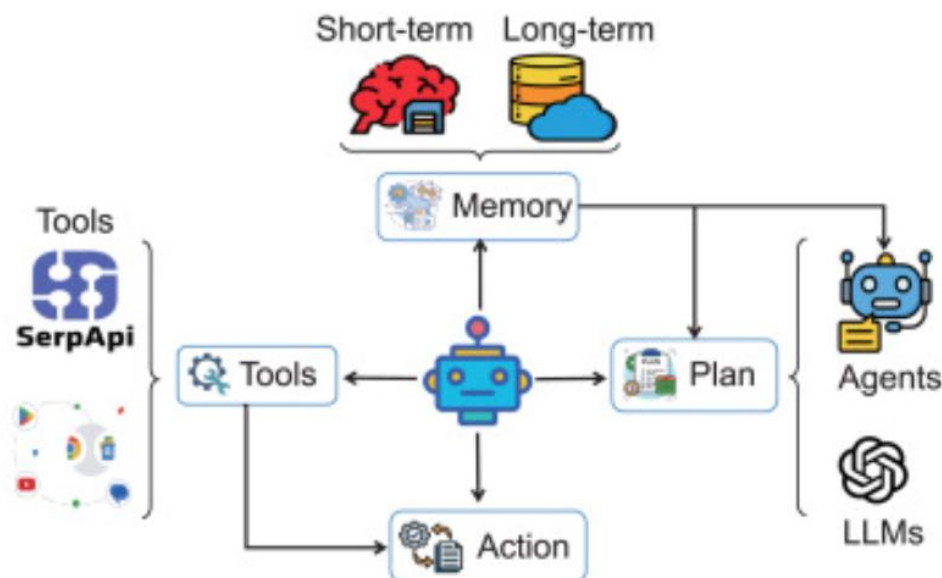


Figura 3 - 7. Arquitectura agentes llm (Yan et al., 2025).

3.3.2. LangChain

El uso de frameworks como LangChain que es un marco de trabajo de rápida evolución que ofrece un enfoque versátil y modular para desarrollar aplicaciones impulsadas por

modelos de lenguaje de gran tamaño (Citar). Mediante LangChain, los desarrolladores pueden simplificar etapas complejas del ciclo de vida de las aplicaciones tales como: el desarrollo, la puesta en producción y el despliegue; facilitando la creación de aplicaciones escalables, con estado y conscientes del contexto. Además, proporciona herramientas para manejar modelos de chat, integrar generación aumentada por recuperación (RAG) y ofrecer interacciones seguras con APIs. El desarrollo se vuelve factible, logrando así el despliegue rápido de soluciones sofisticadas basadas en LLM en diversos dominios, como se observa en la figura 3.8.

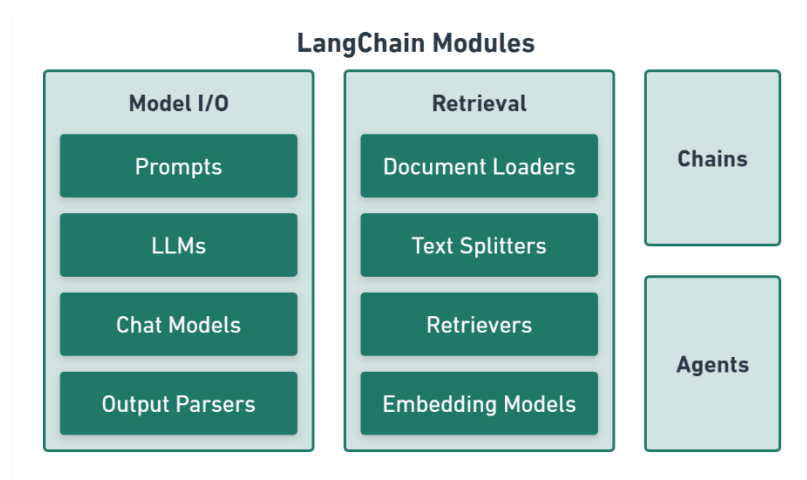


Figura 3 - 8. Módulos De LangChain (LangChain, 2024).

No obstante, a pesar de sus ventajas, el uso de LangChain también introduce ciertos desafíos. Su enfoque modular y altamente integrable puede incrementar la complejidad del sistema, especialmente en aplicaciones que requieren múltiples herramientas y flujos de ejecución (Citar) Asimismo, la interacción con APIs externas y el manejo de datos sensibles plantean posibles riesgos de seguridad, lo que hace necesario implementar mecanismos de control y validación adecuados. En términos de escalabilidad, el uso intensivo de modelos de lenguaje y servicios externos puede implicar costos computacionales elevados, lo cual debe ser considerado en el diseño del sistema (LangChain, 2024).

El uso del framework de desarrollo LangChain representa un componente fundamental para la construcción del agente conversacional. Este framework permite integrar el modelo de lenguaje con herramientas específicas, como el pipeline de procesamiento de datos hidrometeorológicos, facilitando la ejecución de análisis a partir de consultas en lenguaje natural. De esta manera, el agente no se limita a generar respuestas, sino que actúa como un orquestador que selecciona y ejecuta las herramientas necesarias para obtener resultados basados en datos reales.

3.3.3. Metodología RAG

Dentro de las metodologías actuales en inteligencia artificial, destaca el enfoque conocido RAG, el cual ha emergido como una solución viable para mejorar la precisión y relevancia de las respuestas generadas por modelos de lenguaje. Este enfoque se basa en la integración de mecanismos de recuperación de información externa dentro del proceso de generación de texto, permitiendo que el modelo no dependa únicamente del conocimiento aprendido durante su entrenamiento, sino que pueda acceder a datos actualizados y específicos en tiempo de ejecución, como vemos en la figura 3 - 9.

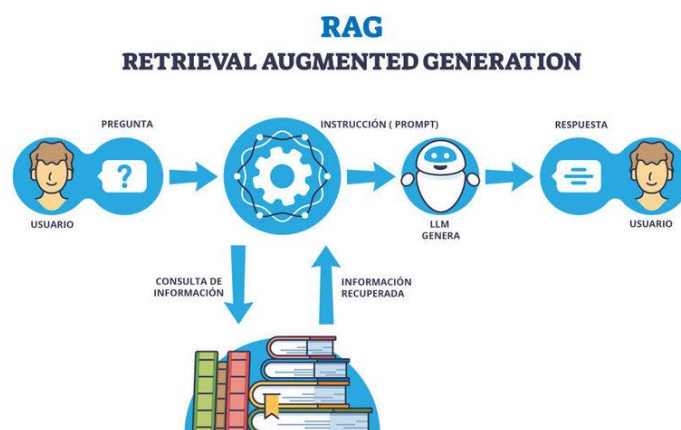


Figura 3 - 9. Proceso de RAG (Fuente: Elaboración propia).

A diferencia de los enfoques tradicionales, donde los modelos de lenguaje generan respuestas basadas únicamente en patrones aprendidos, RAG introduce una capa adicional de recuperación de información que enriquece el contexto utilizado por el modelo. Esto resulta particularmente relevante en aplicaciones donde la precisión de los datos es crítica, ya que permite reducir la generación de información incorrecta y mejorar la consistencia de las respuestas (Arslan et al., 2024).

En el contexto del análisis hidrometeorológico, el uso de RAG adquiere especial relevancia, ya que permite integrar datos provenientes de fuentes confiables, como registros de estaciones climatológicas, con capacidades de interpretación en lenguaje natural. En el proyecto, este enfoque se refleja en la utilización de herramientas que recuperan datos reales procesados mediante el pipeline del sistema, los cuales son posteriormente interpretados por el agente conversacional para generar respuestas precisas y fundamentadas.

3.4. Análisis cuantitativo entre plataformas

A continuación, se presenta una comparación entre diferentes plataformas, aplicaciones y sistemas que tienen que ver con el análisis y la consulta de datos hidrometeorológicos. El objetivo es identificar lo que pueden hacer, lo que no pueden hacer y cómo se relacionan con la propuesta en el proyecto como se aprecia en la Tabla 3-1.

Tabla 3 - 1. Tabla comparativa del estado del arte.

Sistema / Plataforma	Tipo	Fuente de datos	Análisis de datos	Interacción	Limitaciones	Relación con el proyecto
CONAGUA	Plataforma de datos	Estaciones meteorológicas nacionales	No integrado (requiere procesamiento externo)	Consulta manual	Datos en bruto, requiere limpieza y conocimiento técnico	Fuente principal de datos utilizada en el proyecto
NOAA	Plataforma de datos	Estaciones globales, APIs	Limitado (requiere herramientas externas)	Técnica (APIs)	Acceso complejo para usuarios no técnicos	Referente de datos abiertos y estructurados
Google Earth	Plataforma en la nube	Datos geoespaciales globales	Avanzado (requiere programación)	Técnica (scripts)	Alta complejidad de uso	Inspiración en procesamiento eficiente de grandes datos
The Weather Channel	Aplicación móvil	Datos meteorológicos procesados	No (solo visualización)	Usuario amigable	No permite análisis histórico ni personalizado	Ejemplo de accesibilidad, no de análisis
AccuWeather	Aplicación móvil	Modelos propios + datos climáticos	Limitado (enfocado a predicción)	Usuario amigable	No permite manipulación de datos	Referente en predicción y experiencia de usuario

Apple Weather	Aplicación móvil	Datos meteorológicos integrados	No (visualización avanzada)	Intuitiva	No permite análisis ni consultas complejas	Referente en UX/UI moderna
Proyecto Propuesto	Sistema propuesto	Datos CONAGUA procesados	Completo (EDA, correlación, regresión)	Lenguaje natural (agente IA)	Dependencia de calidad de datos	Integra datos, análisis y conversación en un solo sistema

En conclusión, se identificaron áreas de oportunidad relacionadas con la accesibilidad de la información, la complejidad técnica de algunas herramientas y la necesidad de contar con mecanismos que permitan a usuarios no especializados obtener información útil de manera sencilla y confiable. En este contexto, el proyecto busca integrar las ventajas de las plataformas de datos hidrometeorológicos y las tecnologías modernas de inteligencia artificial para ofrecer una solución capaz de consultar, analizar e interpretar información climatológica mediante una interfaz conversacional.

Una vez analizados los antecedentes tecnológicos y las soluciones existentes, en el siguiente capítulo se presenta el desarrollo de la propuesta, describiendo la arquitectura, diseño e implementación del sistema, así como los componentes que permiten su funcionamiento.

Capítulo 4

4. Metodología de desarrollo

En este capítulo se describe la metodología utilizada para el desarrollo del proyecto, desde la obtención de los datos hidrometeorológicos hasta la generación de respuestas mediante inteligencia artificial.

Así mismo se explican las metodologías, arquitecturas y tecnologías utilizadas para su construcción, además se presentan los componentes que forman parte del sistema.

Se muestran los enfoques utilizados para organizar el desarrollo. También se describe cómo se integró la inteligencia artificial, la arquitectura del sistema y las herramientas tecnológicas seleccionadas.

Además, se describe cómo funciona el sistema en general. Esto incluye el flujo de procesamiento de datos y cómo los usuarios interactúan con el sistema a través del agente conversacional.

Este capítulo ayuda a entender cómo la propuesta del proyecto se convierte en una solución funcional. Integra el procesamiento de datos hidrometeorológicos, análisis automatizados y capacidades de interacción en lenguaje natural en una sola arquitectura.

4.1. Desarrollo del pipeline de datos

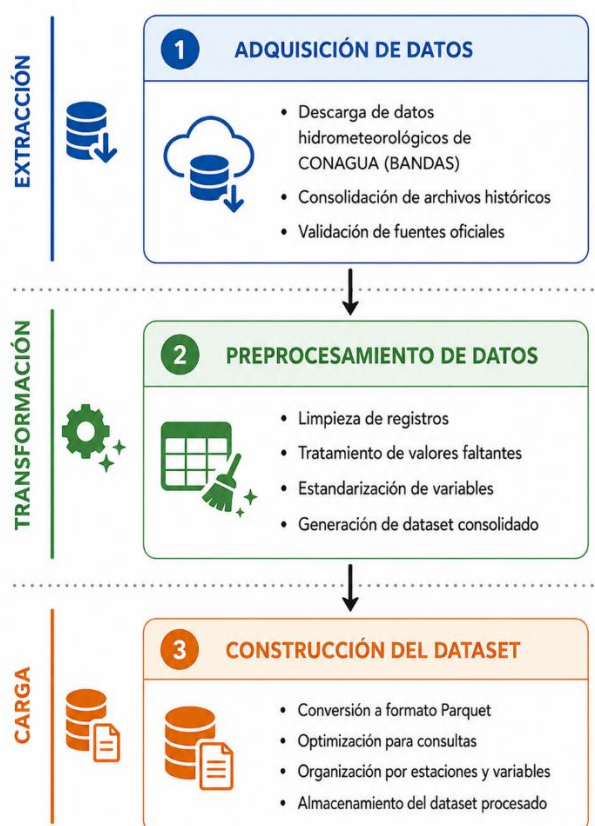


Figura 4 - 1. Pipeline de datos del sistema (Fuente: Elaboración propia).

La Figura 4 - 1 ilustra el proceso ETL (Extracción, Transformación y Carga) que se utilizó para preparar los datos que el proyecto emplea. Dicho proceso consta de tres etapas principales. Durante la etapa de **extracción**, se adquieren los datos hidrometeorológicos de las fuentes oficiales de CONAGUA en formato txt y se consolidan los archivos históricos de las diversas estaciones climatológicas que se incluyeron en el proyecto.

Acto seguido, en la fase de **transformación**, los datos recolectados pasan por un preprocesamiento que abarca la depuración de registros, el manejo de valores ausentes y la uniformización de las variables. Con estas acciones se consigue un conjunto de datos unificado y estructurado de forma pareja, lo que disminuye las disparidades que podrían mermar los análisis subsecuentes.

Por último, en la fase de **carga**, el conjunto de datos ya preprocesado se convierte y guarda en formato **Parquet**. Este formato facilita la organización de la información por estación y variable, además de agilizar las consultas subsiguientes. El archivo que se genera se convierte en la fuente de datos trabajada que luego utilizan las herramientas de análisis y el agente de inteligencia artificial del proyecto.

4.2. Metodología de desarrollo e inteligencia artificial

El proyecto se desarrolló combinando dos enfoques: la organización del proceso de construcción del software y el diseño e implementación de componentes de inteligencia artificial. Esto permitió estructurar tanto la gestión del proyecto como la creación de un sistema inteligente capaz de procesar y analizar datos hidrometeorológicos.

Se utilizó una metodología de desarrollo ágil. Esto facilitó la implementación incremental de funcionalidades, lo que permitió validar continuamente el comportamiento del sistema y realizar ajustes según los requerimientos del proyecto. Este enfoque fue fundamental para gestionar la complejidad del sistema, que integra múltiples componentes, como el procesamiento de datos, la generación de reportes y la interacción mediante lenguaje natural.

El sistema también incorpora una metodología de inteligencia artificial basada en el uso de modelos de lenguaje de gran escala y arquitecturas orientadas a agentes. Esto permite que el sistema no solo procese datos, sino que también interprete consultas en lenguaje natural y ejecute análisis de forma automatizada. La combinación de estos elementos da lugar a un sistema híbrido que integra capacidades de análisis de datos con interacción conversacional.

4.2.1. Metodología de desarrollo

Para desarrollar el proyecto, se usó una metodología llamada Scrum. Esta metodología ayudó a trabajar de manera organizada, paso a paso, y a entregar partes del sistema poco a poco. Así pudimos probar y mejorar cada parte antes de seguir adelante.

El proyecto tiene muchas partes, como manejo de datos, inteligencia artificial y servicios en línea. Por eso, dividimos el trabajo en períodos de tres semanas llamados “sprints”. En cada sprint, hacíamos un plan, implementábamos nuevas funcionalidades, las probábamos y revisábamos si funcionaban bien.

Antes de empezar cada sprint, definíamos qué necesitábamos hacer desde el punto de vista del usuario. Estas “historias de usuario” nos ayudaron a priorizar lo más importante y a enfocarnos en esas partes primero.

Desarrollamos varias de estas historias, como un sistema para procesar datos del clima y agua, métricas para ver la calidad de los datos, un chat con inteligencia artificial y un flujo completo de principio a fin. Construimos estas partes poco a poco en diferentes sprints.

Cada sprint incluía pruebas para asegurarnos de que todo funcionara correctamente. Esto nos permitió encontrar errores a tiempo y mejorar la calidad del sistema antes de seguir adelante. A medida que avanzábamos, agregamos pruebas automáticas para asegurarnos de que todo el sistema funcionara bien.

Scrum ayudó a mantener todo organizado, a adaptarnos a cambios y a construir el sistema paso a paso. Fue especialmente porque integramos muchas tecnologías y necesitábamos probar y mejorar constantemente.

4.2.2. Inteligencia artificial

El sistema utiliza una forma de inteligencia artificial que combina modelos de lenguaje con herramientas para analizar datos. Esto permite que el sistema entienda el lenguaje natural y procese datos estructurados. De esta manera, el sistema puede interpretar preguntas y realizar análisis de forma automática.

El sistema tiene un componente central llamado agente conversacional. Este agente recibe preguntas del usuario, analiza lo que se está pidiendo y decide qué acciones tomar para resolver la solicitud. A diferencia de otros modelos de lenguaje, este agente no genera información por sí solo, sino que utiliza herramientas externas que trabajan con datos reales.

El sistema sigue un ciclo de tres etapas: percepción, decisión y acción. Primero, el sistema recibe la entrada del usuario. Luego, el modelo de lenguaje interpreta la pregunta y selecciona la herramienta adecuada. Finalmente, se ejecuta la herramienta, se procesan los datos y se genera una respuesta en lenguaje natural basada en los resultados.

Además, el sistema utiliza un enfoque que se basa en datos reales para generar respuestas. En lugar de buscar documentos, el sistema accede a datos hidrometeorológicos previamente procesados y los utiliza para generar respuestas. Esto ayuda a reducir la generación de información incorrecta y a mejorar la confiabilidad del sistema.

El uso de herramientas especializadas, como el procesamiento de datos y los módulos de análisis estadístico, permite separar la lógica de negocio del modelo de lenguaje. Esto asegura que los resultados se generen a partir de información verificable y mejora la robustez del sistema.

En resumen, el sistema combina inteligencia artificial con análisis de datos hidrometeorológicos. Esto permite que el sistema no solo entienda preguntas en lenguaje natural, sino que también realice procesos complejos de análisis de forma automática.

4.3. Proceso de desarrollo de software

El sistema se desarrolló de una manera estructurada, combinando el ciclo de vida del software con un enfoque ágil basado en Scrum. Esto significa que el desarrollo fue iterativo e incremental, lo que permitió construir el sistema de manera progresiva. En cada paso, se validaron los componentes y se facilitó la adaptación a los requerimientos del proyecto.

A diferencia de los métodos tradicionales, donde las fases del ciclo de vida del software se siguen de forma lineal, en este proyecto se adoptó un enfoque iterativo. En este enfoque, las etapas de análisis, diseño, implementación y pruebas se repiten en múltiples ciclos. Cada ciclo, que duraba tres semanas, permitió desarrollar una parte funcional del sistema, que se evaluaba antes de avanzar a la siguiente, veamos la figura 4 - 2.

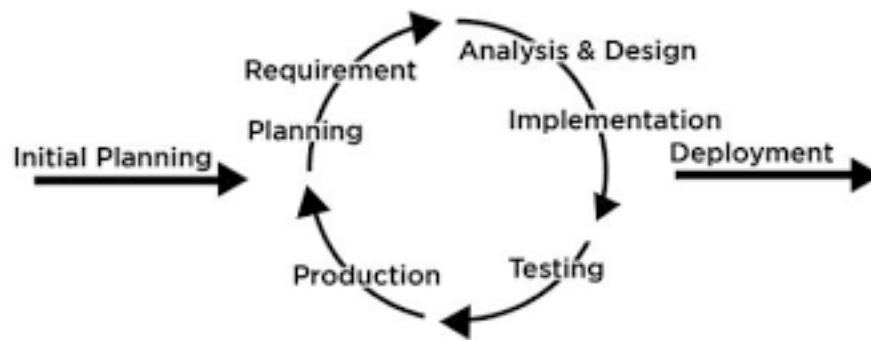


Figura 4 - 2. Modelo Iterativo e Incremental (Fuente: Elaboración propia).

El proceso comenzó con la fase de análisis de requerimientos, donde se identificaron las necesidades del sistema, como el procesamiento de datos hidrometeorológicos, el análisis estadístico y la interacción mediante lenguaje natural. Sin embargo, en lugar de definir todos los requerimientos de forma rígida desde el inicio, se refinaron de manera incremental en cada ciclo, lo que permitió una mayor flexibilidad en el desarrollo.

En la fase de diseño, se definió la arquitectura del sistema y la organización de sus componentes. El diseño no se estableció de forma definitiva en una sola etapa, sino que evolucionó con el tiempo, a medida que se integraban nuevos módulos, como el agente conversacional, el pipeline de datos y los mecanismos de generación de reportes. Esto permitió ajustar la arquitectura en función de las necesidades reales del sistema.

Durante la fase de implementación, el desarrollo se llevó a cabo de manera incremental, construyendo funcionalidades específicas en cada ciclo. Por ejemplo, en los primeros ciclos, se desarrollaron componentes básicos de procesamiento de datos, mientras que en etapas posteriores se integraron capacidades más avanzadas, como el agente basado en inteligencia artificial y el flujo end-to-end de análisis. Esto permitió tener versiones funcionales del sistema desde etapas tempranas del desarrollo.

La fase de integración se realizó de manera continua, en lugar de ser una etapa final aislada. En cada ciclo, los componentes desarrollados se integraron progresivamente, asegurando su compatibilidad y correcto funcionamiento dentro del sistema. Esto permitió validar el flujo completo del sistema de manera temprana, reduciendo riesgos asociados a la integración tardía.

Finalmente, la fase de pruebas y validación se llevó a cabo de forma iterativa en cada ciclo de desarrollo. Se implementaron pruebas unitarias para verificar componentes individuales y pruebas integrales para validar el funcionamiento del sistema completo. Esto permitió detectar errores de manera temprana y asegurar la calidad del sistema en cada incremento entregado.

4.4. Análisis

En este apartado se habla del análisis del sistema. El objetivo es encontrar los elementos necesarios para diseñarlo e implementarlo. Esta etapa permite definir qué debe hacer el sistema, quiénes interactúan con él, cuáles son sus limitaciones y cómo se relaciona con su entorno.

El análisis se divide en cuatro partes principales. Primero, se describen los requisitos del sistema, como lo que debe hacer, lo que no debe hacer, la calidad que debe tener y las limitaciones que tiene. Luego, se identifican las personas que interactúan con el proyecto. Después, se muestra un diagrama que permite ver el sistema de manera general. Finalmente, se describen los patrones de funcionamiento del sistema.

Esta parte es importante porque permite entender lo que el sistema necesita antes de definir cómo estará estructurado y qué tecnologías se utilizarán para implementarlo.

4.4.1. Análisis de requerimientos

El sistema se diseñó con base en un conjunto de requisitos que establecen su funcionamiento, rendimiento y limitaciones operativas.

Estos requisitos se dividen en varias categorías:

- Requisitos funcionales
- Requisitos no funcionales
- Requisitos de calidad
- Restricciones

Cada una de estas categorías desempeña un papel crucial en la configuración y el desempeño del sistema.

Requisitos funcionales

Los requisitos funcionales definen cómo el sistema se comporta cuando el usuario hace algo. También definen lo que el sistema muestra al usuario. En la Tabla 4-1, se enumeran los requisitos funcionales del sistema.

Requerimiento	Descripción	Prioridad
RF-01	El sistema debe mantener un historial de conversaciones por usuario.	Alta
RF-02	El sistema debe permitir consultar datos hidrometeorológicos mediante lenguaje natural.	Alta
RF-03	El sistema debe ejecutar análisis exploratorio de datos (EDA).	Alta
RF-04	El sistema debe calcular estadísticas (media, mediana, mínimo, máximo, desviación estándar).	Alta
RF-05	El sistema debe permitir análisis por rangos de fechas.	Alta
RF-06	El sistema debe generar reportes en formato HTML o Markdown.	Media
RF-07	El sistema debe integrar un agente conversacional basado en IA.	Alta
RF-8	El sistema debe permitir visualizar resultados de análisis en lenguaje natural.	Alta
RF-9	El sistema debe permitir consultar estaciones meteorológicas por nombre.	Alta

Tabla 4 - 1. Tabla requisitos funcionales del sistema.

Requisitos no funcionales

Los requisitos no funcionales definen cómo funciona un sistema, describiendo sus cualidades, características y restricciones (seguridad, rendimiento, usabilidad) en lugar de sus tareas específicas. En la Tabla 4-2, se enumeran los requisitos no funcionales del sistema.

Requerimiento	Descripción	Prioridad
RNF-01	El sistema debe responder en tiempos adecuados (< 5 segundos en consultas promedio).	Alta
RNF-02	El sistema debe ser modular para facilitar mantenimiento.	Alta
RNF-03	El sistema debe ser interoperable entre React Y Python.	Alta
RNF-04	El sistema debe asegurar comunicación segura entre servicios (HTTPS, tokens).	Alta
RNF-05	El sistema debe ser tolerante a fallos en servicios externos.	Media

Tabla 4 - 2. Tabla requisitos no funcionales del sistema.

Requisitos de calidad

Los requisitos de calidad se definen como un conjunto de requisitos para comprobar si el proyecto cumple con la calidad del software esperada, En la Tabla 4-3, se enumeran los requisitos de calidad del sistema.

Requerimiento	Descripción	Prioridad
RC-01	El sistema debe garantizar precisión en los análisis de datos.	Alta
RC-02	El sistema debe evitar generación de información incorrecta (alucinaciones).	Media
RC-03	El sistema debe mantener trazabilidad mediante logs y eventos.	Alta
RC-04	El sistema debe permitir reproducibilidad de análisis.	Media
RC-05	El sistema debe ser mantenible y extensible.	Alta

Tabla 4 - 3. Tabla requisitos de calidad del sistema.

Restricciones

Las restricciones se definen como se delimita el alcance de las funcionalidades del sistema, En la Tabla 4-4, se enumeran las restricciones del sistema.

Requerimiento	Descripción	Prioridad
R-01	Uso obligatorio de datos provenientes de CONAGUA.	Alta
R-02	Dependencia de modelos de lenguaje configurados.	Media
R-03	Limitaciones de recursos computacionales.	Media
R-04	Integración con Python	Alta
R-05	El sistema debe operar sobre datos previamente procesados (parquet).	Alta

Tabla 4 - 4. Tabla restricciones del sistema.

4.4.2. Actores

El único actor presente en el sistema es el usuario, que interactúa directamente con el chatbot para poder obtener datos y análisis hidrometereológico todo dependiente a lo que solicite.

4.4.3. Diagrama de contexto

La relación entre los actores y la representación del movimiento de datos está expuesta en el esquema de contexto del sistema que se muestra en la Figura 4 - 3.

NIVEL 1: Contexto

Diagrama de contexto del sistema
ConAlgua - ChatBot Meteorológico

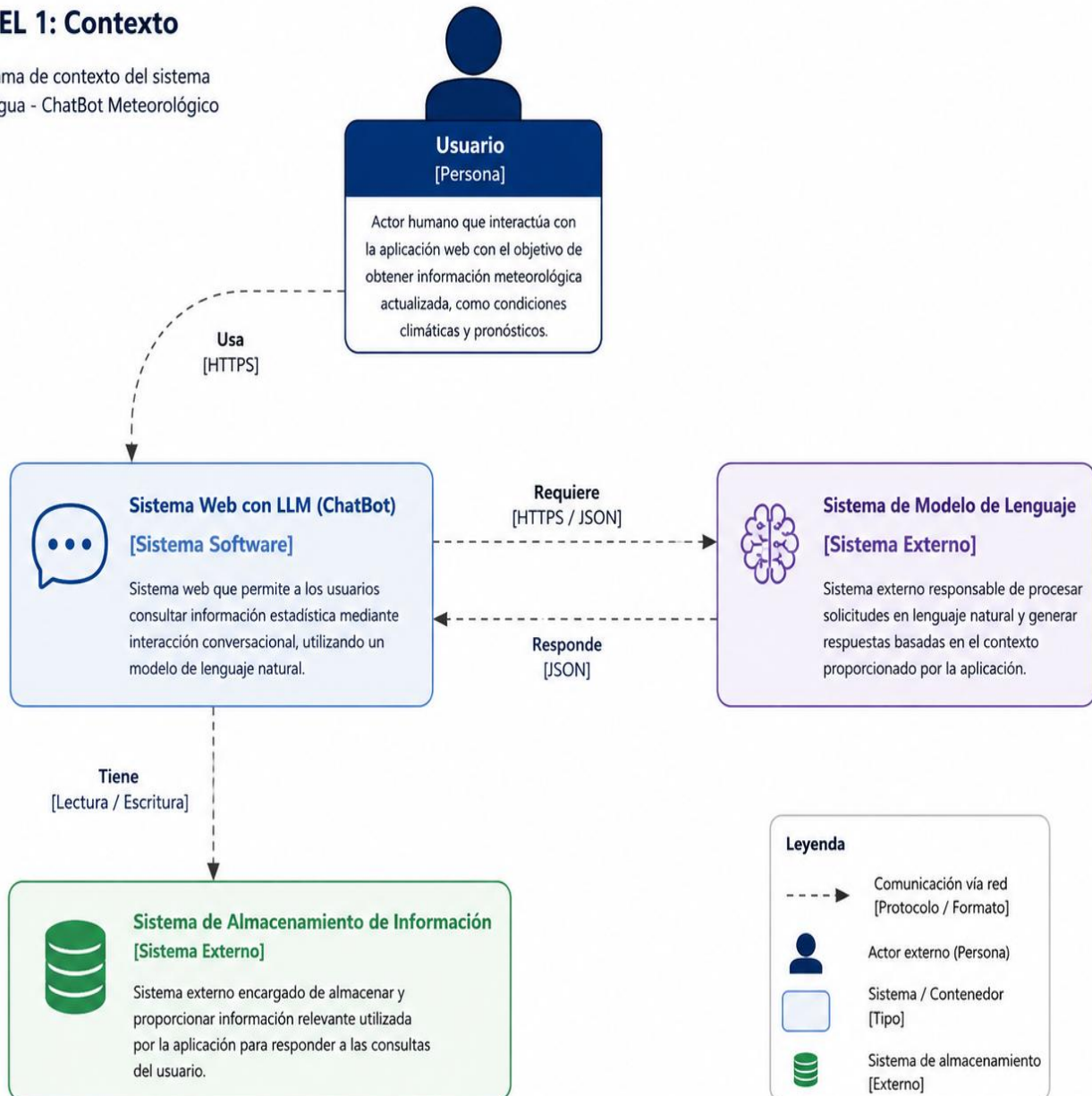


Diagrama de Contexto - Sistema Web con LLM (ChatBot) ConAlgua

Figura 4 - 3. Diagrama de contexto del sistema Nivel 1 (Fuente: Elaboración propia).

El usuario interactúa con el sistema mediante una interfaz conversacional tipo chatbot, a través de la cual realiza consultas en lenguaje natural orientadas al análisis de datos hidrometeorológicos. Estas consultas son interpretadas por el sistema para identificar la intención del usuario y transformar la entrada en una serie de acciones internas, tales como la ejecución de procesos de análisis, recuperación de información y generación de resultados.

Este enfoque permite establecer una interacción intuitiva y dinámica, en la cual el usuario puede acceder a funcionalidades complejas sin requerir conocimientos técnicos especializados, facilitando la exploración y comprensión de los datos mediante un modelo de comunicación basado en lenguaje natural.

4.4.4. Arquetipos

El sistema está formado por varios arquetipos básicos que muestran cómo interactúan y se procesan las cosas dentro de él, como vemos en la figura 4 – 4.

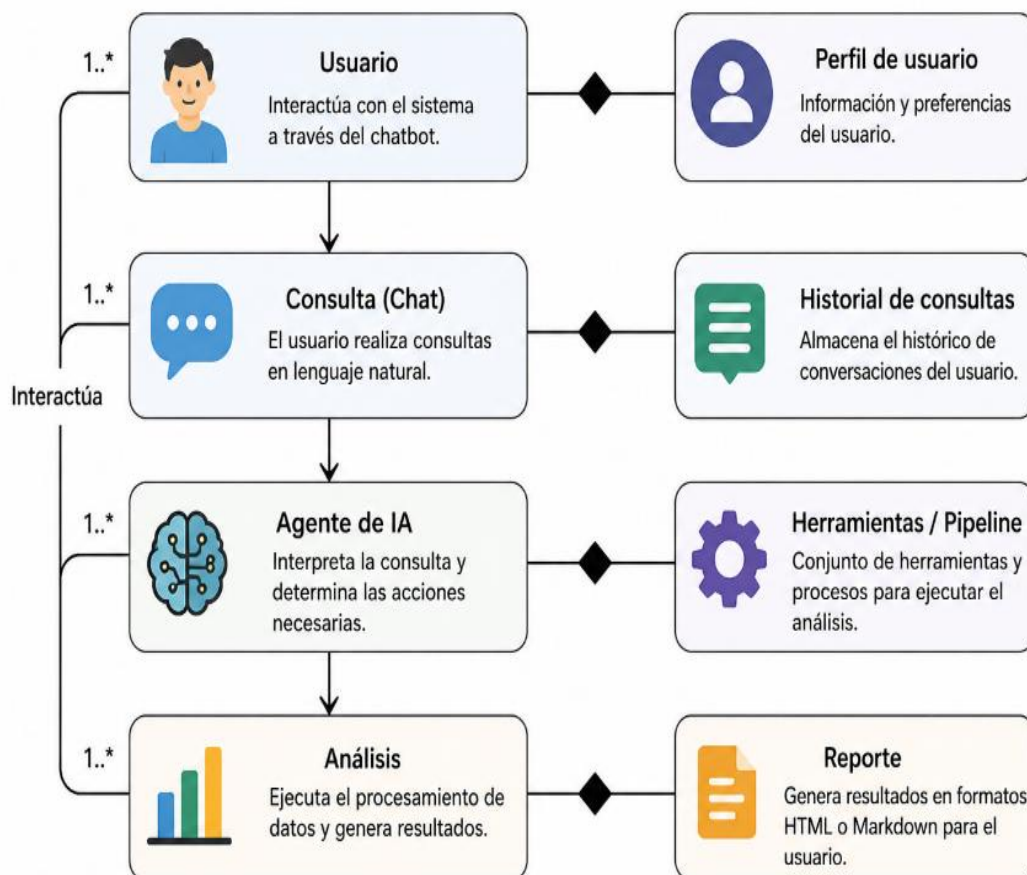


Figura 4 - 4. Diagrama de arquetipos del sistema (Fuente: Elaboración propia).

El arquetipo más importante es el del Usuario, que se comunica con el sistema haciendo preguntas en un lenguaje normal a través de una ventana de chat. Estas preguntas crean algo llamado Consulta, que el Agente de Inteligencia Artificial procesa para entender lo que el usuario quiere.

Este agente es como un director que usa las herramientas necesarias para solucionar la petición, creando el arquetipo de Análisis. Este análisis usa datos ordenados y crea informes que el usuario puede ver.

El sistema también incluye cosas como el Perfil de usuario y el Historial de consultas, que ayudan a recordar lo que ha pasado y a mejorar la experiencia del usuario.

El sistema funciona de la siguiente manera:

- El usuario interactúa con el sistema mediante consultas en lenguaje natural.
- El Agente de Inteligencia Artificial procesa la Consulta para entender la intención del usuario.
- El agente ejecuta las herramientas necesarias para resolver la solicitud, generando el arquetipo de Análisis.
- El Análisis se apoya en fuentes de datos estructurados y produce reportes que pueden ser consultados por el usuario.
- El sistema incorpora elementos complementarios como el Perfil de usuario y el Historial de consultas para mejorar la experiencia del usuario.

4.5 Diseño

En esta sección, se presenta el diseño del sistema. Este diseño incluye la arquitectura propuesta, el modelo de datos y la interfaz gráfica de la aplicación.

El sistema se compone de varios elementos principales. Estos elementos están organizados de manera que interactúan entre sí.

La interacción entre los elementos permite el procesamiento y análisis de datos hidrometeorológicos. Estos datos se procesan mediante un enfoque conversacional basado en inteligencia artificial.

4.5.1. Arquitectura

Para diseñar el proyecto, se optó por una arquitectura modular que separa las responsabilidades, cuya estructura se puede apreciar en la Figura 4-5. El sistema se organiza en distintos módulos con funciones concretas: la interfaz web, el agente de inteligencia artificial, las herramientas para analizar datos, el espacio donde se guarda la información climática y el servicio externo de modelo de lenguaje.

Así, las tareas de presentación, procesamiento y análisis de datos quedan bien diferenciadas. La interfaz web es la que interactúa con el usuario, mientras que el agente, construido con LangGraph, se encarga de gestionar las peticiones y elegir las herramientas adecuadas para los análisis. A su vez, las herramientas de análisis acceden a los datos climáticos que ya han sido procesados y guardados en formato Parquet. Por último, el agente se comunica con un modelo de lenguaje externo a través de la API de Groq para entender las peticiones y elaborar respuestas en lenguaje natural.

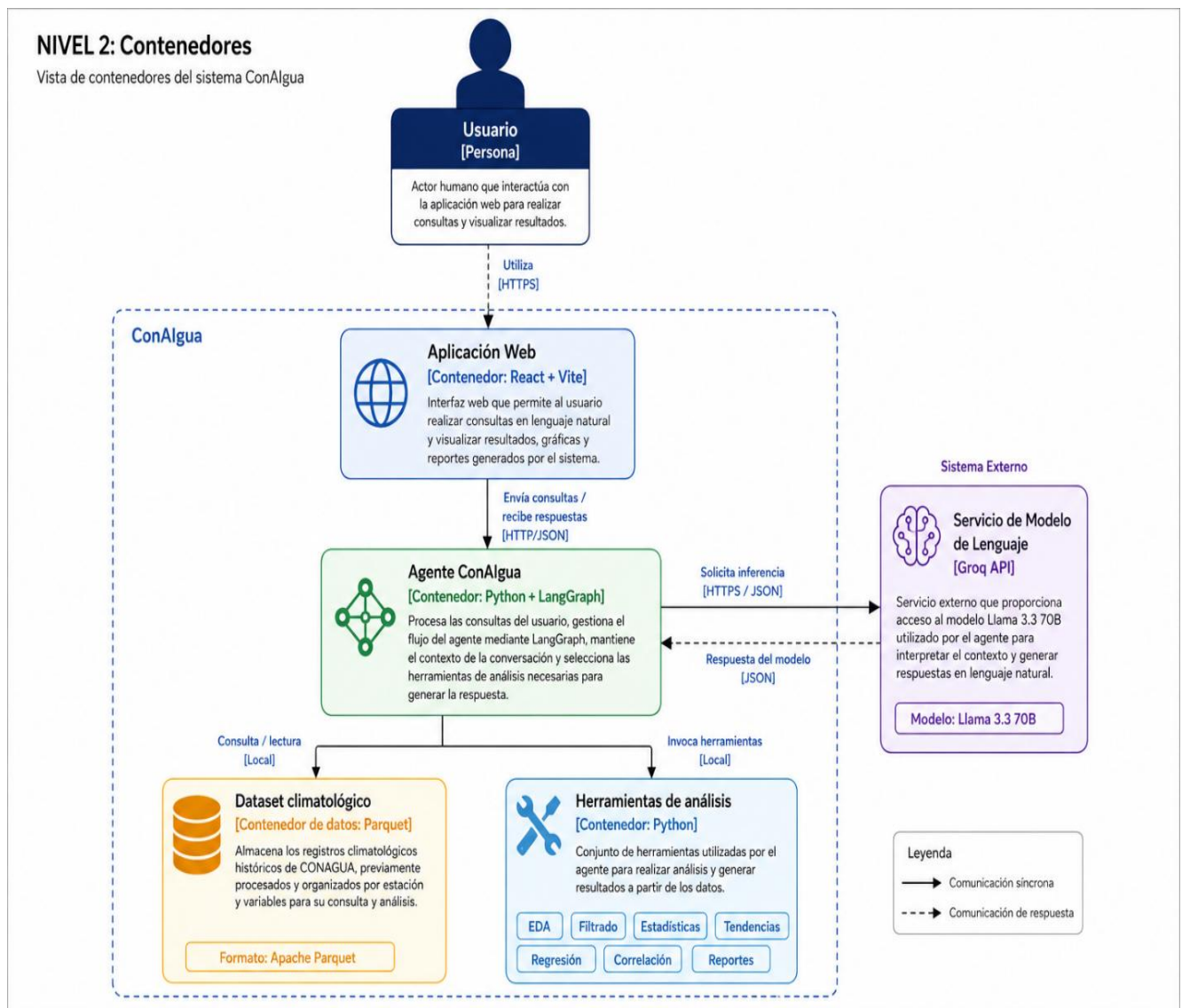


Figura 4 - 5. Diagrama de contenedores del sistema (Fuente: Elaboración propia).

El **frontend** del proyecto se ha creado con **React y Next.js**. Este componente es el que da la cara, la interfaz gráfica, a través de la cual el usuario habla con el sistema usando un chatbot. Desde aquí, el usuario puede preguntar lo que necesite sobre información del clima en lenguaje normal y ver las respuestas y los resultados que saca el sistema.

Usar React y Next.js ayuda a montar la interfaz a base de piezas que se pueden usar en varios sitios y a tener bien ordenada la parte de cómo se muestra todo en la aplicación. Así, es más fácil mantener el sistema y añadirle cosas nuevas poco a poco.

La mayor parte del trabajo de procesar las preguntas lo hace el **agente de inteligencia artificial de proyecto**, que está hecho en **Python** y se maneja con **LangGraph**. El agente recibe lo que le pide el usuario, lleva el control de cómo se va haciendo todo y decide qué herramientas usar según lo que le pregunten.

Para hacer todos los análisis, el agente tiene un montón de herramientas hechas en Python que le permiten hacer cosas como explorar los datos (EDA), seleccionar registros, sacar estadísticas básicas, ver tendencias, hacer regresiones, ver correlaciones y crear informes. Estas herramientas operan directamente sobre los datos del clima.

La información que usa el sistema está guardada en unos datos en formato **Parquet**. Estos datos se han sacado de los registros del clima de CONAGUA, pasándolos por un proceso de extracción, transformación y carga. Este formato mantiene los datos bien ordenados y hace más fácil buscar información de estaciones, épocas y variables concretas.

Además, el agente se comunica con un modelo de lenguaje externo a través de la API de Groq. En la implementación actual se está utilizando el modelo Llama 3.3 70B, que es capaz de interpretar las consultas formuladas en lenguaje natural y ayuda a generar las respuestas que se le muestran al usuario.

De este modo, la arquitectura del sistema se estructura en cuatro componentes principales: la interfaz web, creada con React y Next.js; el agente de inteligencia artificial, que se apoya en LangGraph; las herramientas de análisis; y el conjunto de datos climatológicos guardado en formato Parquet. También se recurre al servicio externo de Groq para acceder al modelo de lenguaje, tal como se ilustra en la Figura 4-6.

NIVEL 3: Componentes

Descomposición del contenedor Agente ConAlgua

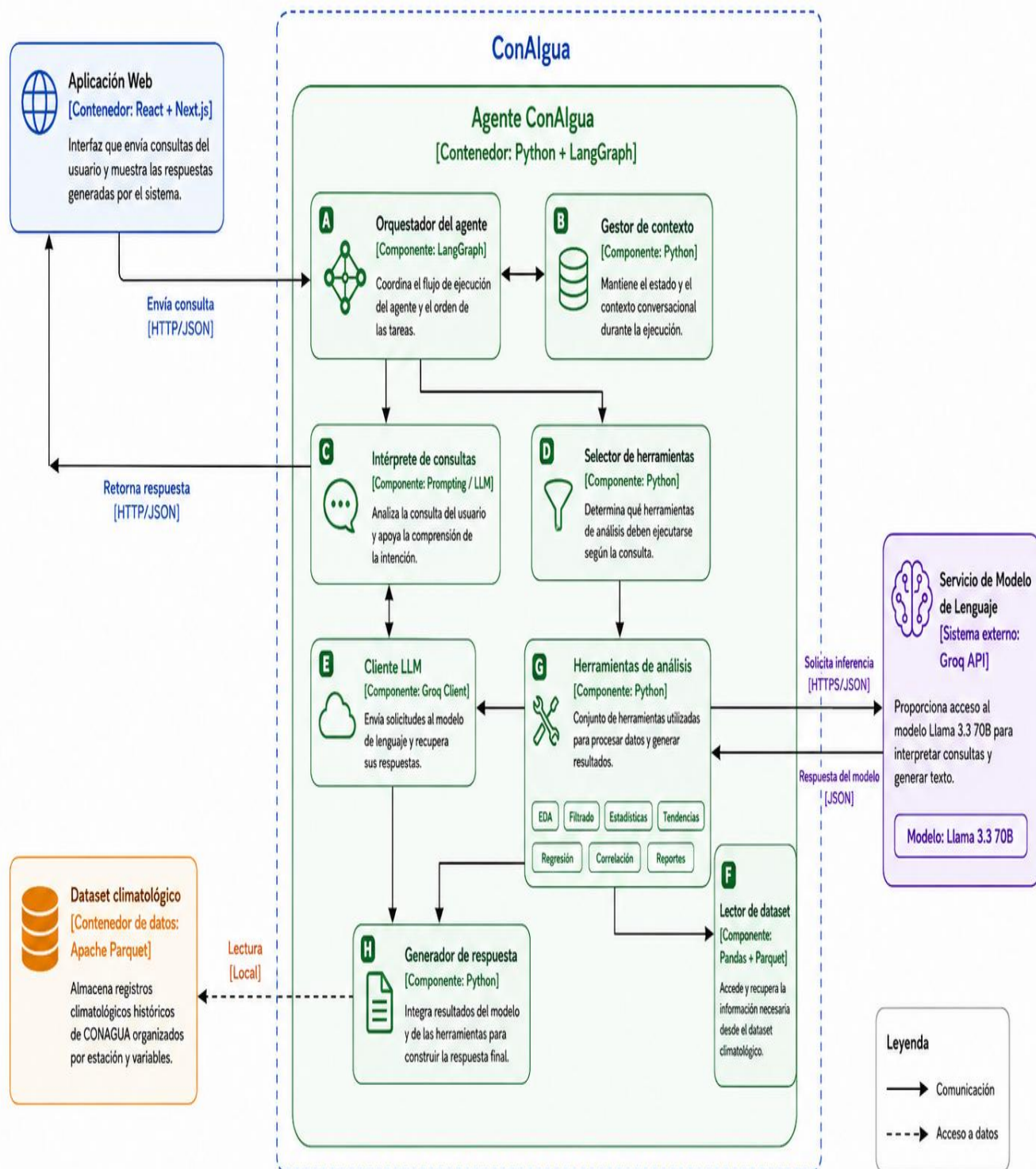


Figura 4 - 6. Diagrama de componentes del sistema (Fuente: Elaboración propia).

El componente de inteligencia artificial del proyecto se desarrolló en Python y usa LangGraph para definir y coordinar cómo se ejecuta el agente. Elegimos Python por su gran ecosistema de bibliotecas para procesar y analizar datos, además de su compatibilidad con las herramientas que usamos en el proyecto, como pandas, LangChain y ydata-profiling.

El agente es la parte que procesa las preguntas que el usuario hace en lenguaje natural. Con cada pregunta, analiza qué busca el usuario y decide qué herramientas debe usar para conseguir la información necesaria. Estas herramientas sirven para hacer operaciones con los datos del clima, como seleccionar registros, calcular estadísticas básicas, explorar datos, encontrar tendencias, correlaciones, hacer regresiones y generar informes.

El backend (LangGraph) se encarga de coordinar estas operaciones, permitiendo organizar el flujo del agente y manejar la ejecución de las distintas herramientas según lo que necesite cada consulta. Además, el agente mantiene el contexto durante la conversación y emplea un modelo de lenguaje para entender las preguntas del usuario y dar respuestas claras basándose en los resultados.

Para procesar la información, las herramientas del agente acceden al conjunto de datos del clima que ya habíamos procesado y guardado en formato Apache Parquet. Así, las respuestas se basan en los datos históricos de CONAGUA y no solo en lo que sabe el modelo de lenguaje en general. El agente también se comunica con el modelo Llama 3.3 70B a través de la API de Groq, transmitiendo el contexto requerido para entender las preguntas y producir las respuestas en lenguaje natural. La aplicación web, creada con React y Next.js, posibilita que el usuario interactúe con este proceso y observe los resultados que produce el sistema. La estructura interna de este componente se muestra en la Figura 4-7.

ARQUITECTURA DEL AGENTE IA ConAlgua

Agente Implementado con LangGraph

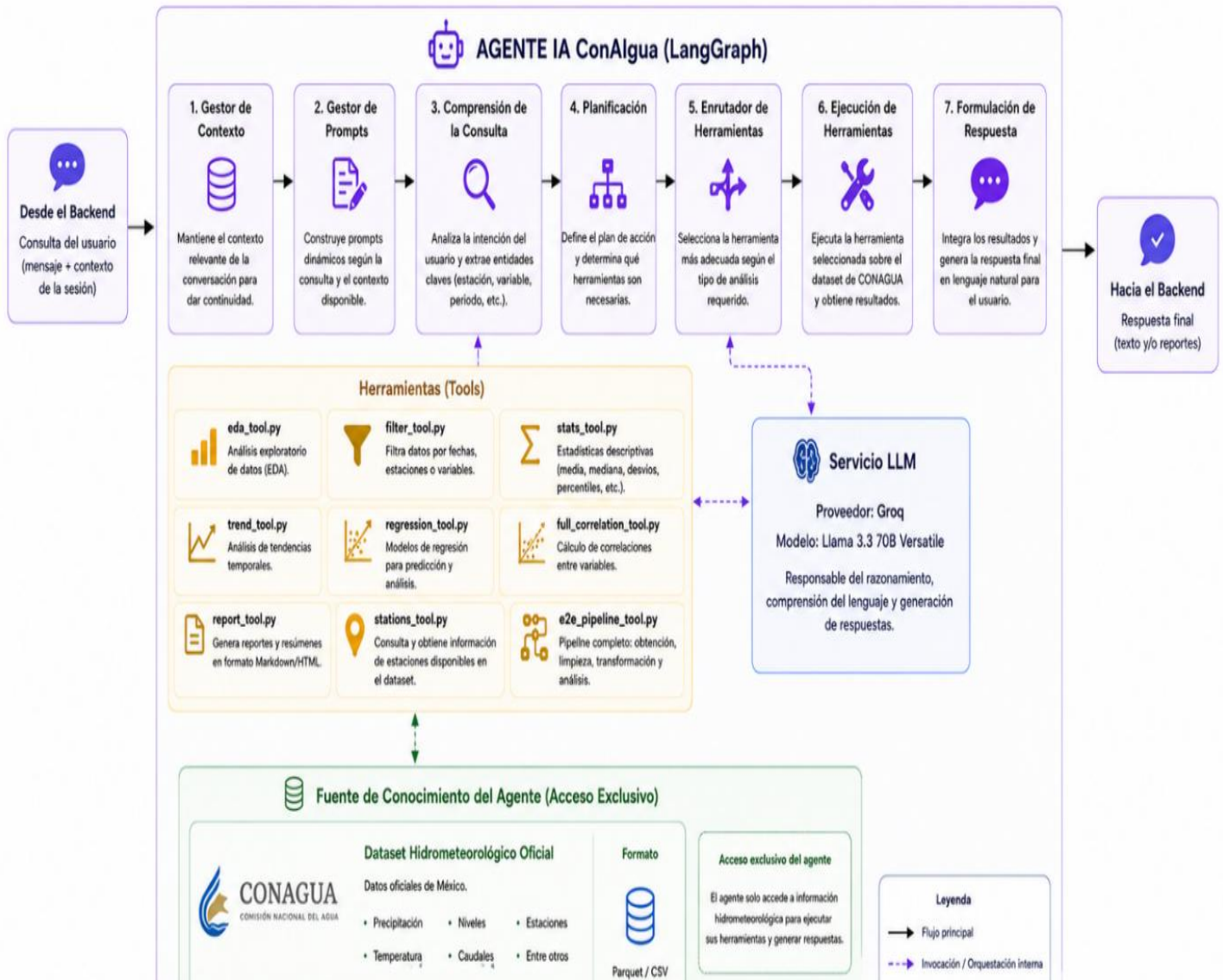


Figura 4 - 7. Componentes del agente IA (Fuente: Elaboración propia).

4.5.2. Modelo de datos

El modelo de datos del proyecto está diseñado para guardar y examinar datos climáticos históricos. A diferencia de una arquitectura transaccional común, la versión actual no necesita guardar información de usuarios, sesiones o historiales de chat en una base de datos, ya que el objetivo fundamental del sistema es facilitar la consulta y el análisis de los registros climáticos que ya han sido procesados.

La información que usa el sistema viene de los archivos históricos de CONAGUA y pasa primero por un proceso de extracción, transformación y carga (ETL). En esta etapa, los registros se limpian, se uniformizan y se juntan para luego guardarlos en un conjunto de datos con formato Parquet.

El conjunto de datos incluye la información de las diferentes estaciones climáticas contempladas en el proyecto. Cada registro mantiene los datos necesarios para identificar la estación y el período al que corresponde la observación, además de las variables climáticas más importantes usadas en los análisis, como precipitación, evaporación, temperatura máxima y temperatura mínima.

En términos generales, la información se estructura con los siguientes componentes:

Estación: Identifica la estación climática de donde proviene el registro.

Fecha: Sirve para establecer el período de cada observación.

Año y mes: Facilitan la agrupación y el análisis temporal de los registros.

Precipitación: Indica la cantidad de lluvia registrada durante el período correspondiente.

Evaporación: Contiene los valores de evaporación disponibles para la estación.

Temperatura máxima: Registra la temperatura más alta observada.

Temperatura mínima: Registra la temperatura más baja observada.

4.5.3. Interfaz Grafica

La interfaz gráfica del proyecto se diseñó para que la interacción fuera fácil, clara y centrada en buscar información del clima usando lenguaje natural. Se implementó con React y Next.js, lo que permitió fue crear una aplicación web dinámica donde el usuario puede hablar con el agente de inteligencia artificial a través de una interfaz de chat.

El diseño visual de la interfaz es minimalista, enfocándose en que el contenido se lea bien y quitando cosas que pudieran distraer al usuario de lo importante. Como se ve en la Figura 4-8, la aplicación tiene dos partes principales: un panel a la izquierda para el historial de chats y una sección central donde ocurre la conversación con el agente. En esta primera pantalla también está el espacio para que el usuario escriba sus preguntas, un botón para enviarlas y una opción para ocultar las llamadas internas a herramientas, todo para que la experiencia sea más limpia.

Capturas de pantalla

Vista general de la interfaz



Chat con el agente

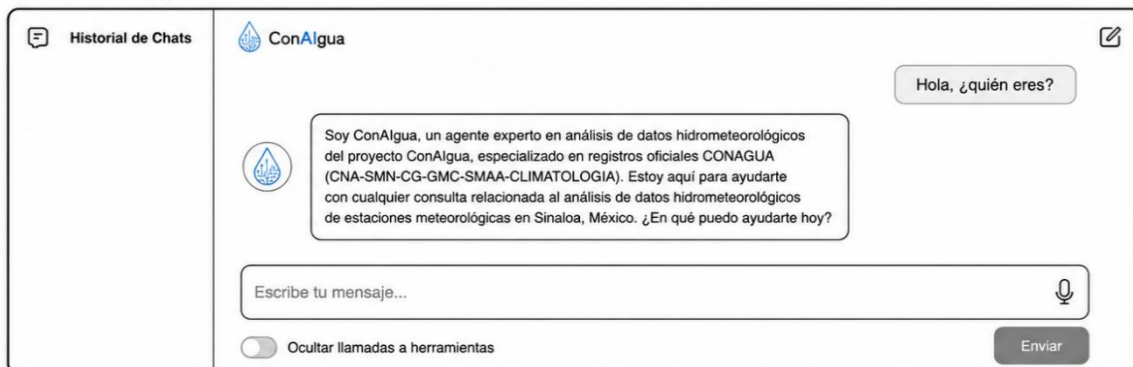


Figura 4 - 8. Vista general de la interfaz del proyecto (Fuente: Elaboración propia).

La Figura 4-9 muestra cómo funciona la interfaz cuando se hace una consulta normal. Aquí, el usuario puede preguntar sobre las estaciones de clima que hay y obtener respuestas del agente en lenguaje natural. En este ejemplo, el sistema muestra una lista de las estaciones registradas y luego responde a una petición para hacer un análisis de regresión lineal simple entre la lluvia y el mes para una estación en particular. Esta interacción prueba que la interfaz no solo deja intercambiar mensajes, sino que también muestra de forma clara los resultados de los análisis que hace el agente.



Figura 4 - 9. Ejemplo de interacción conversacional con el agente (Fuente: Elaboración propia).

Además de contestar preguntas analíticas, la interfaz deja pedir la generación de informes. Tal como se ve en la Figura 4-10, el usuario puede solicitar al sistema que cree un informe en formato HTML para una estación concreta. Una vez que la petición es procesada, el agente envía una confirmación escrita y da un enlace para abrir el informe que se ha generado. De este modo, la interfaz hace más fácil acceder a productos analíticos más completos sin que el usuario tenga que salir del entorno de conversación.

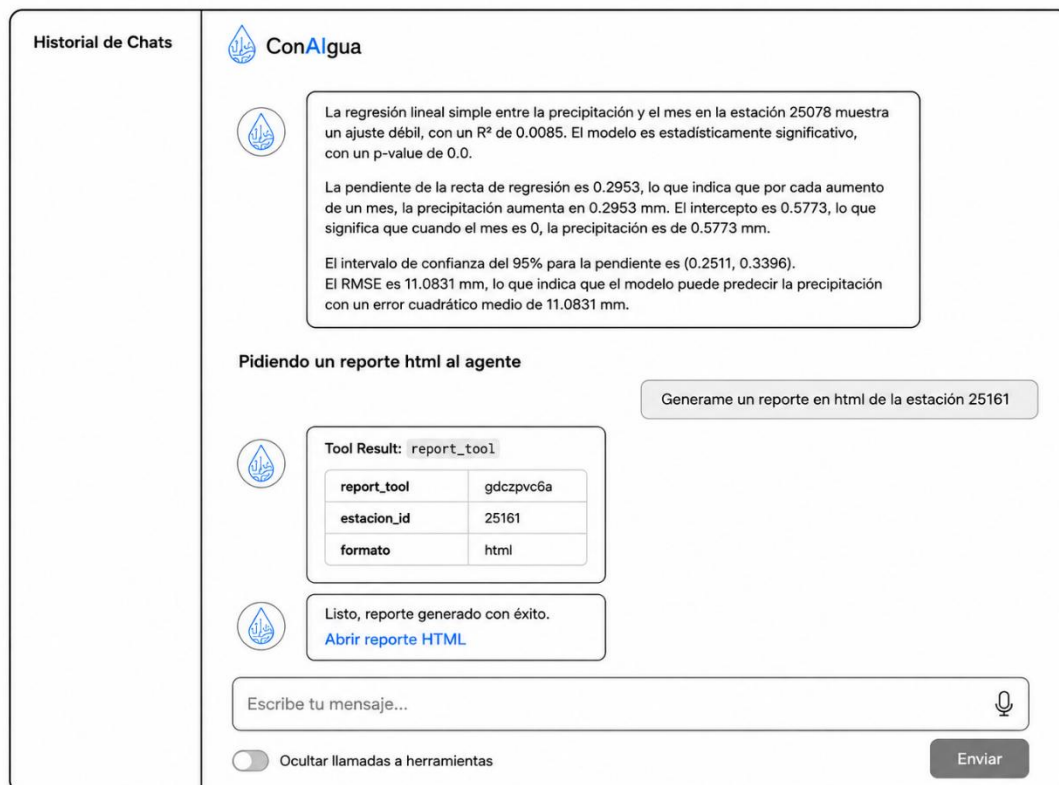


Figura 4 - 10. Solicitud de generación de reporte HTML desde la interfaz (Fuente: Elaboración propia).

Por otro lado, la Figura 4-11 muestra el informe EDA en formato HTML. En esta pantalla se ven partes dedicadas al análisis exploratorio de datos, como un resumen general del conjunto de datos, la clasificación de variables y el examen de atributos concretos. Este tipo de resultado completa la experiencia de conversación al dar una forma más ordenada y detallada de la información que el sistema ha analizado.

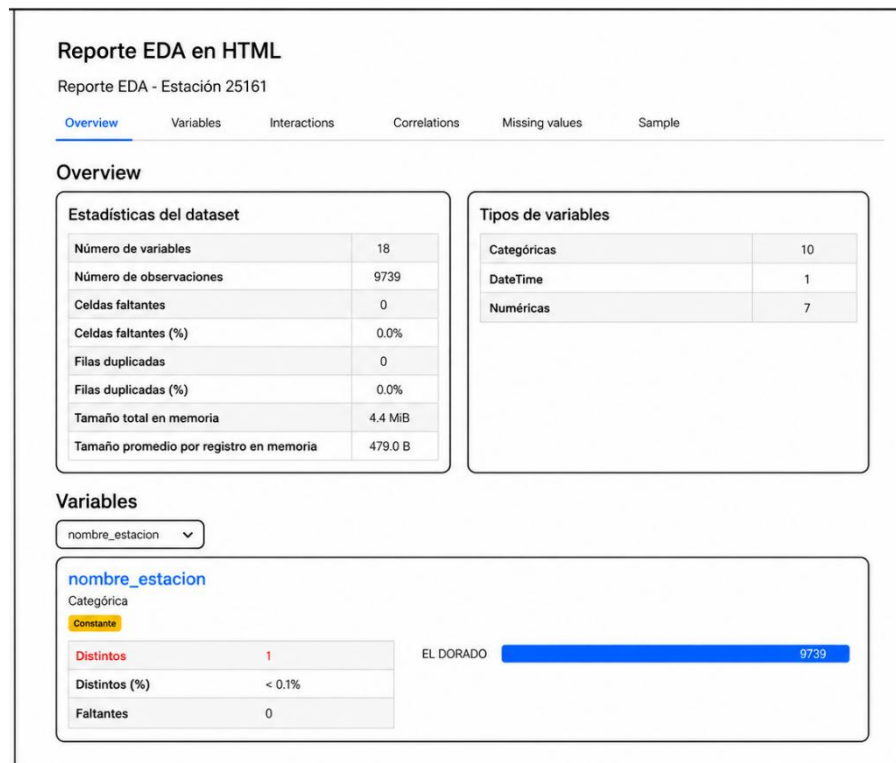


Figura 4 - 11. Visualización del reporte EDA en formato HTML (Fuente: Elaboración propia).

4.6. Implementación

El proyecto se ejecutó como una aplicación web enfocada en buscar y examinar datos del clima usando lenguaje común. La parte visual se hizo con React y Next.js, mientras que el componente de inteligencia artificial se programó en Python, apoyándose en LangGraph y LangChain para controlar su funcionamiento y conectarlo con las herramientas de análisis.

Para generar las respuestas, se conectó el modelo LLaMA 3.3 70B Versatile a través de la API de Groq. El manejo de la información climática se lleva a cabo con herramientas desarrolladas en Python y librerías como pandas y ydata-profiling, usando principalmente un conjunto de datos ya procesado en formato Apache Parquet.

Además, se puso en marcha un proceso ETL para sacar, modificar y meter los registros históricos del clima que se obtuvieron de CONAGUA. Con base en esta información, el sistema puede hacer selecciones, cálculos estadísticos, identificar patrones, ver relaciones, realizar regresiones y producir informes de análisis exploratorio de datos en formato HTML.

4.6.1. React

Para construir la interfaz gráfica del proyecto, se optó por React. Su arquitectura, que se basa en componentes, hizo posible el desarrollo de una interfaz conversacional dinámica. A través de ella, el usuario puede hacer consultas en lenguaje natural y ver las respuestas que genera el agente. Asimismo, React se empleó para implementar elementos como el área de conversación, el campo donde se escriben las consultas, los mensajes del usuario, las respuestas del agente, el panel para revisar el historial de chats y el acceso a los reportes que genera el sistema.

4.6.2. Next.js

Para el desarrollo de la aplicación web, se usó Next.js como framework. Su integración con React facilitó la organización de la estructura del frontend a través de páginas y componentes, sentando las bases para la implementación de la interfaz gráfica. El framework permitió integrar los diversos elementos visuales de la aplicación y establecer la comunicación necesaria entre la interfaz y el agente de inteligencia artificial. Así, las consultas del usuario se pueden enviar al agente y las respuestas obtenidas se muestran directamente en el entorno conversacional.

4.6.3. Componentes

La interfaz gráfica se desarrolló utilizando componentes que se pueden reutilizar, lo que permitió dividir sus distintas funciones en elementos independientes. Un ejemplo de esta estructura se puede observar en la Figura 4-12.

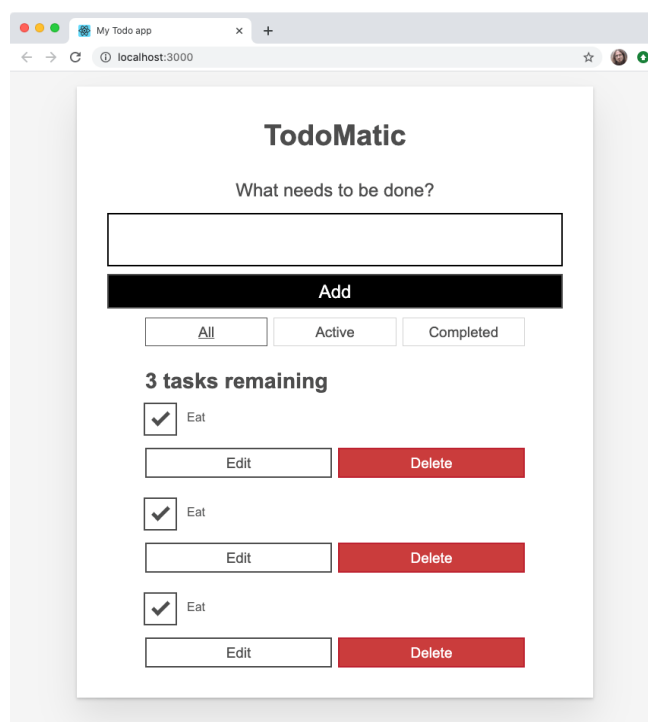


Figura 4 - 12. Ejemplo de componente en React (Fuente: Captura tomada de un proyecto).

Entre los componentes más importantes que se implementaron están el área de conversación del chatbot, el campo para ingresar consultas, los mensajes del usuario, las respuestas del agente, el panel del historial de chats y los elementos para acceder a los reportes generados.

4.6.4. LangChain

Se recurrió a LangChain como una herramienta auxiliar para enlazar el modelo de lenguaje con las diversas herramientas de las que dispone el agente.

Gracias a su implementación, fue posible especificar qué herramientas podían emplearse al procesar una consulta y simplificar su conexión con el modelo de lenguaje. Estas herramientas posibilitan la ejecución de tareas concretas, tales como la filtración de datos, el cálculo de estadísticas, el análisis exploratorio, la identificación de tendencias, la regresión, el establecimiento de correlaciones y la elaboración de informes.

De este modo, LangChain actúa como un componente clave en la conexión entre las capacidades del modelo de lenguaje y las operaciones que se llevan a cabo con los datos climatológicos.

4.6.5. LangGraph

LangGraph se empleó como el marco principal para la creación y la gestión del agente de inteligencia artificial del proyecto.

Esta tecnología facilita la definición de la secuencia de pasos que se siguen al procesar una consulta. El agente recibe la petición del usuario, la interpreta y decide qué herramientas son necesarias para obtener la información pertinente.

LangGraph también posibilita conservar el estado necesario durante la ejecución y coordinar la interacción entre el modelo de lenguaje, las herramientas de análisis y los resultados obtenidos a partir del conjunto de datos.

4.6.6. Integración con el modelo LLaMA 3.3 70B Versatile

Para el procesamiento de consultas en lenguaje natural, se conectó el modelo LLaMA 3.3 70B Versatile a través de la API de Groq.

El agente emplea el modelo de lenguaje para descifrar las peticiones del usuario, discernir la finalidad de la consulta y ayudar a escoger las herramientas apropiadas. Más adelante, los resultados que arrojan estas herramientas sirven de contexto para elaborar una respuesta inteligible en lenguaje natural.

Los cálculos estadísticos y las operaciones que se realizan con la información climática no dependen exclusivamente de lo que sabe el modelo de lenguaje. Estos se llevan a cabo con herramientas diseñadas precisamente para consultar y procesar el conjunto de datos del proyecto, lo que garantiza que las respuestas se basen en los registros climáticos existentes.

La integración a través de Groq posibilita cursar las peticiones al modelo de lenguaje desde el agente y obtener las respuestas requeridas para proseguir con el flujo establecido mediante LangGraph.

4.7. Depuración y pruebas

Durante el desarrollo del proyecto se llevaron a cabo actividades de depuración y pruebas para comprobar el correcto funcionamiento de los distintos componentes del sistema, abordando desde el procesamiento inicial de los datos hasta la interacción del usuario con el agente de inteligencia artificial, el cual opera satisfactoriamente tal como se observa en la Figura 4-13.

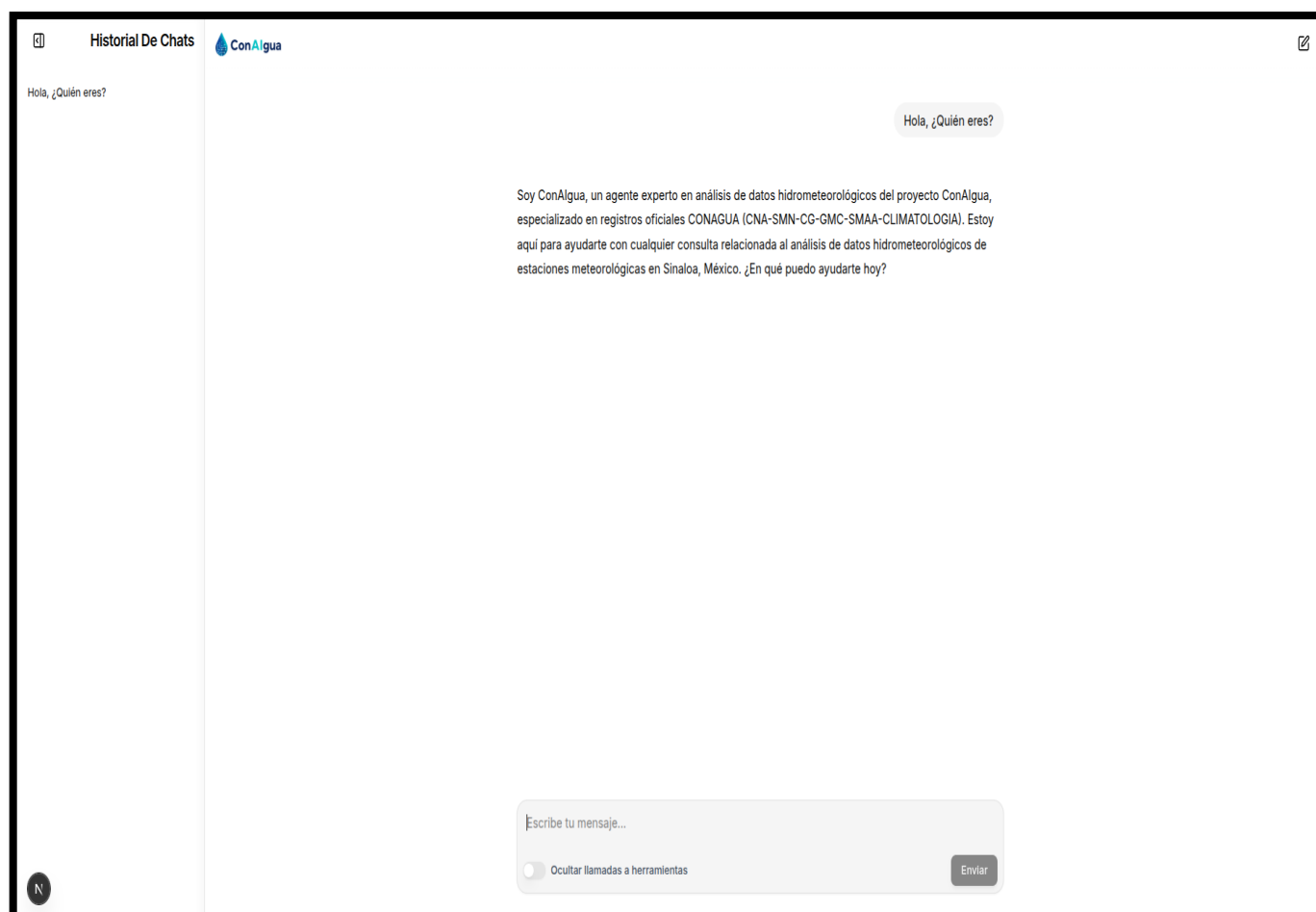


Figura 4 - 13. Ejecución del proyecto (Fuente: Captura tomada del proyecto).

4.7.1. Pruebas

Para poder evaluar cómo funciona el proyecto, llevamos a cabo diversas pruebas tanto en la interfaz gráfica como en el agente de inteligencia artificial y en las herramientas de análisis. Estas pruebas consistieron en hacerle preguntas en lenguaje natural, pedirle que realizara distintas operaciones estadísticas y comprobar que los resultados se generaran y vieran correctamente.

A lo largo de este proceso, detectamos algunos puntos que necesitaban ser modificados. Principalmente, esto se relacionó con la forma en que interactúan el usuario y el agente, cómo funcionan las herramientas de análisis y cómo se presenta la información en la interfaz. Los problemas más importantes que encontramos y las soluciones que aplicamos se detallan en la Tabla 4-5.

Comentario	Acción realizada
El agente debía identificar correctamente las estaciones climatológicas disponibles.	Se verificó la lectura del dataset y se ajustó la herramienta encargada de consultar y mostrar las estaciones disponibles.
Algunas consultas requerían seleccionar una herramienta específica de análisis.	Se ajustaron las instrucciones del agente y la definición de las herramientas para mejorar su selección de acuerdo con la intención de la consulta.
Se necesitaba comprobar el funcionamiento de los análisis estadísticos.	Se realizaron consultas de prueba para estadísticas descriptivas, tendencias, correlaciones y regresiones, comparando los resultados obtenidos con los datos procesados.
La generación de reportes HTML debía realizarse a partir de una estación solicitada por el usuario.	Se verificó la ejecución de la herramienta de reportes y la creación correcta del archivo HTML correspondiente a la estación seleccionada.
Los reportes HTML generados necesitaban ser accesibles desde la interfaz.	Se habilitó el acceso al reporte mediante un enlace generado después de completar correctamente el análisis.

Tabla 4 - 5. Tabla restricciones del sistema.

4.7.2. Dificultades

Durante el desarrollo del proyecto surgieron varios inconvenientes, sobre todo en lo que respecta al manejo de los datos, la forma en que se integraron las herramientas de análisis y cómo se comunicaban las distintas partes del sistema.

- Uno de los primeros retos fue la gran variedad de formatos en los archivos históricos de climatología. Esto se debía a que las estructuras de los registros, los valores que faltaban y los formatos de las variables eran distintos entre sí. Para solucionar esto, creamos un proceso de preprocesamiento que se encargó de limpiar, uniformizar y agrupar la información antes de guardarla en el conjunto de datos final, que se almacenó en formato Parquet.
- Otro problema surgió al integrar las herramientas de análisis dentro del agente. Como cada consulta puede necesitar operaciones distintas, tuvimos que definir bien las herramientas y sus parámetros. Así, el agente pudo elegir la operación correcta según lo que el usuario quería.
- También hubo complicaciones al configurar el entorno donde correría el agente, especialmente con la organización de los módulos de Python y cómo se importaban. Estas cuestiones se resolvieron ajustando la configuración del proyecto y las rutas que se usaban al ejecutar LangGraph.
- La creación y visualización de los informes EDA en HTML fue otro punto que requirió atención. Tuvimos que asegurarnos de que los archivos se crearan bien y de que estuvieran accesibles para poder verlos desde la interfaz a través de un servidor local.
- Además, al integrar el modelo de lenguaje aparecieron mensajes sobre servicios externos de rastreo. Como estas funciones no eran importantes para este prototipo, desactivamos el envío de rastreos externos y solo mantuvimos la comunicación necesaria con el modelo LLaMA 3.3 70B Versatile a través de Groq.

Capítulo 5

5. Experimentos Y Resultados

En este capítulo se exponen los experimentos que se llevaron a cabo para valorar el desempeño del proyecto, así como las conclusiones a las que se llegó. La evaluación tiene en cuenta cómo responde el agente a distintos tipos de preguntas y el grado de satisfacción de los usuarios al interactuar con la aplicación web.

5.1. Interacción con el usuario

El proyecto se maneja a través de una interfaz de conversación creada con React y Next.js. El usuario escribe su pregunta en lenguaje natural y el agente decide qué operaciones necesita para conseguir la información que pide.

Esta interacción se probó con varios tipos de preguntas, desde las más generales sobre los datos que tenemos, hasta las que necesitan usar herramientas estadísticas. Así pudimos verificar todo el proceso, desde la interfaz, pasando por el agente, hasta los datos del clima.

Una de las primeras cosas que se probó fue pedir los datos de las estaciones que están en el sistema. Si le preguntas algo como:

“¿Qué estaciones tienes disponibles?”

El agente revisa el conjunto de datos y muestra algunas de las estaciones que están disponibles, y también dice la cantidad total de estaciones que se han registrado. Así, el usuario puede saber la información que existe antes de pedir un análisis concreto como se aprecia en la Figura 5-1.

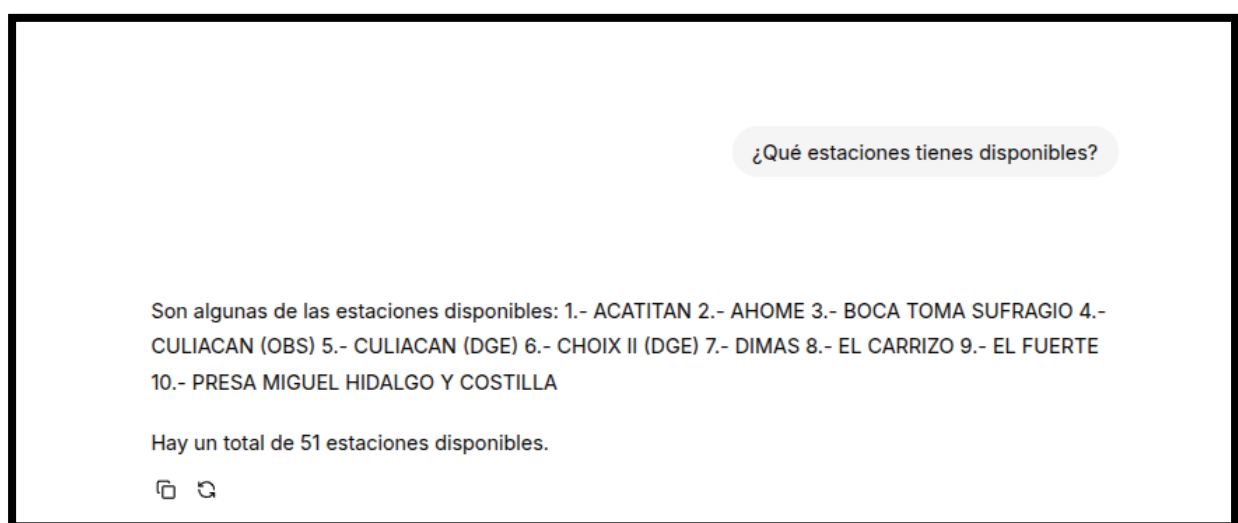


Figura 5 - 1. Ejecución del prompt (Fuente: Captura tomada del proyecto).

Más adelante, se llevaron a cabo consultas que requieren procesamiento estadístico. Por ejemplo:

“Regresión lineal de precipitación contra mes en la estación 25078.”

En esta situación, el agente reconoce que la solicitud se trata de un análisis de regresión, elige la herramienta adecuada, extrae del conjunto de datos los registros pertenecientes a la estación y lleva a cabo el análisis. Para concluir, muestra al usuario los resultados conseguidos, los cuales abarcan métricas como el coeficiente de determinación (R^2), la pendiente, la intersección, el intervalo de confianza y el error cuadrático medio (RMSE) como se aprecia en la Figura 5-2.

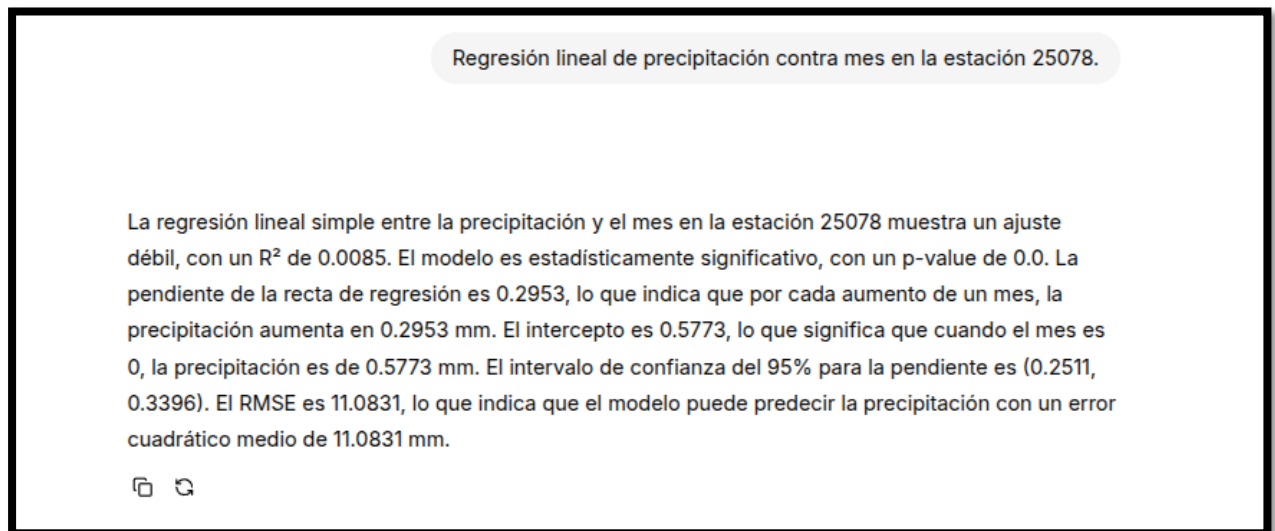


Figura 5 - 2. Ejecución del prompt (Fuente: Captura tomada del proyecto).

Un experimento adicional implicó pedir directamente la creación de un informe a través de una indicación del tipo:

“Genera un reporte en HTML de la estación 25161.”

El agente identifica la estación y el formato que se pide, luego utiliza la herramienta que genera el reporte EDA y, después, le da al usuario un enlace para verlo como se observa en la Figura 5-3.

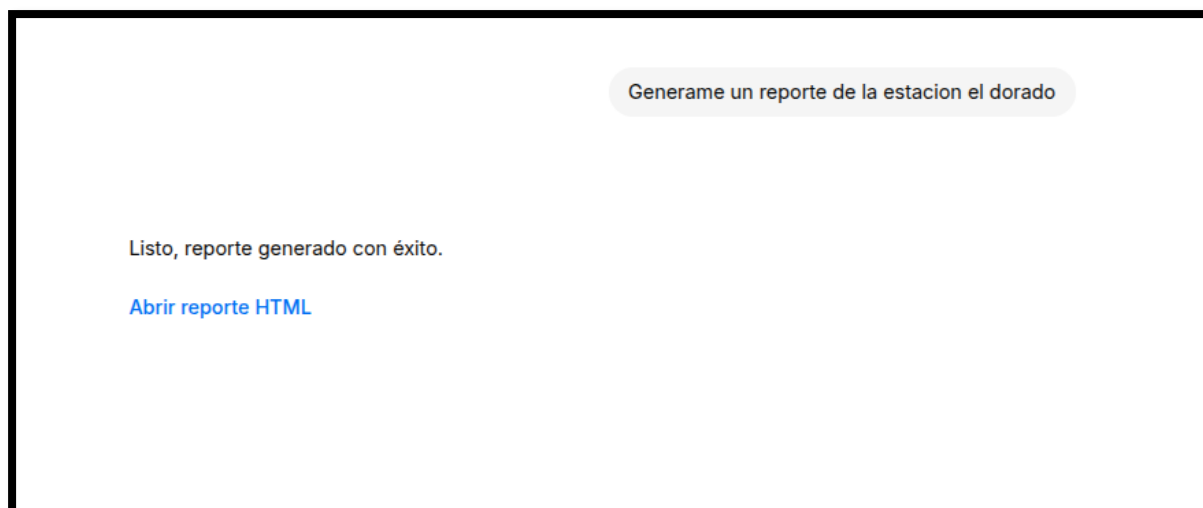


Figura 5 - 3. Ejecución del prompt (Fuente: Captura tomada del proyecto).

El reporte incluye información detallada del conjunto de datos de la estación elegida, como la cantidad de observaciones, los tipos de variables, los valores que faltan, cómo se distribuyen las variables, las correlaciones y otros aspectos que se usaron en el análisis exploratorio como se aprecia en la Figura 5-4.

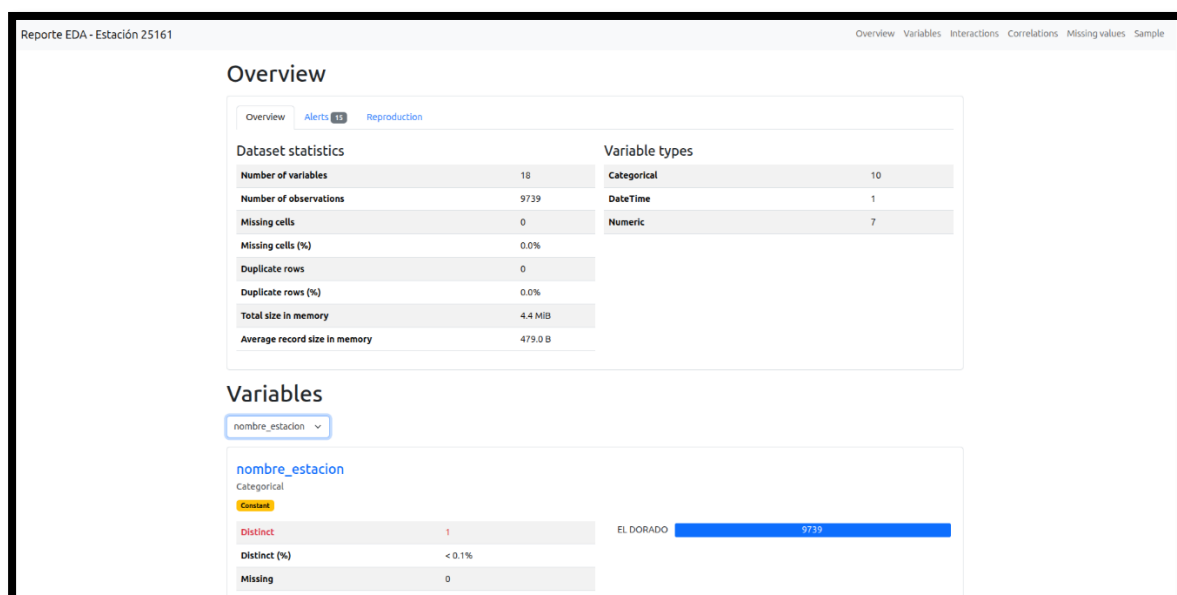


Figura 5 - 4. Archivo HTML generado (Fuente: Captura tomada del proyecto).

Estos ejemplos muestran que para usar el sistema no es necesario que el usuario sepa comandos específicos, lenguajes de programación o pasos estadísticos. Las peticiones se pueden hacer directamente con lenguaje normal y el agente se encarga de saber qué hacer para responderlas.

5.2. Análisis de resultados

Los resultados que se obtuvieron durante las pruebas permitieron constatar que los componentes principales del proyecto funcionaban correctamente y que eran capaces de procesar distintos tipos de consultas formuladas en lenguaje natural. A la hora de evaluar, tomamos en cuenta consultas vinculadas a la disponibilidad de estaciones climatológicas, la ejecución de análisis estadísticos y la generación de informes para análisis exploratorio de datos.

5.2.1. Consulta de estaciones climatológicas

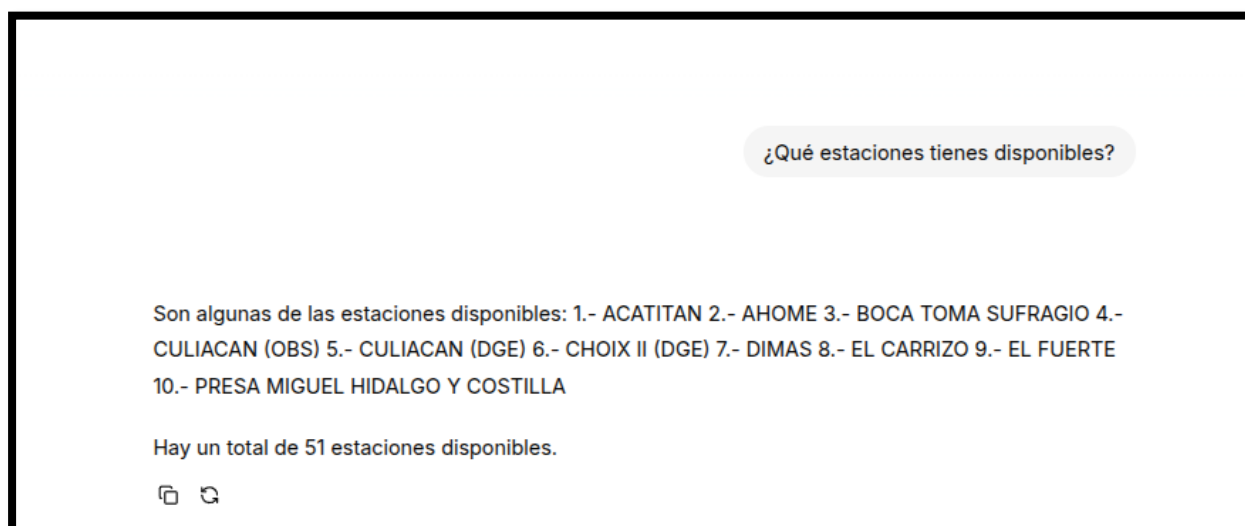
Una de las primeras pruebas que hicimos fue pedirle al agente información sobre las estaciones meteorológicas que teníamos en el conjunto de datos. El usuario preguntó:

“¿Qué estaciones tienes disponibles?”

El agente tramitó la solicitud y extrajo la información pertinente del conjunto de datos climatológicos, presentando una selección de las estaciones disponibles y señalando que el sistema dispone de un total de 51 estaciones.

Dicho resultado validó la capacidad del agente para reconocer consultas sobre la disponibilidad de información y acceder al conjunto de datos sin necesidad de que el usuario conozca de antemano la organización interna de los archivos.

La interacción resultante se ilustra en la Figura 5-5.



*Figura 5 - 5. Consulta de estaciones climatológicas disponibles mediante el agente
(Fuente: Captura tomada del proyecto).*

La prueba demuestra que el usuario puede explorar inicialmente la información disponible utilizando lenguaje natural. Esto facilita que posteriormente pueda seleccionar una estación específica para realizar análisis más profundos.

5.2.2. Resultado del análisis de regresión

Para evaluar la ejecución de herramientas estadísticas se realizó una consulta solicitando una regresión lineal simple:

“Regresión lineal de precipitación contra mes en la estación 25078.”

El agente identificó que la solicitud requería la ejecución de una herramienta de regresión, recuperó los datos asociados a la estación seleccionada y realizó el análisis correspondiente.

Como resultado se obtuvieron los siguientes valores:

- Coeficiente de determinación (R^2): 0.0085.
- Pendiente: 0.2953.
- Intercepto: 0.5773.
- Intervalo de confianza del 95 % para la pendiente: (0.2511, 0.3396).
- RMSE: 11.0831 mm.

La respuesta presentada por el agente puede observarse en la Figura 5-6.

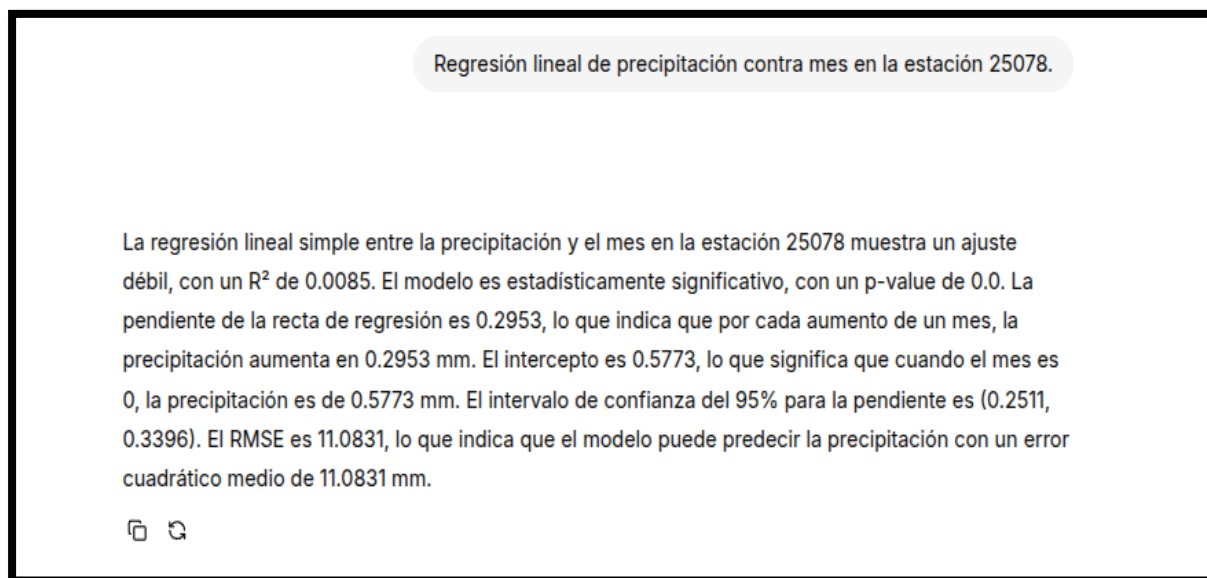


Figura 5 - 6. Resultado del análisis de regresión lineal (Fuente: Captura tomada del proyecto).

El valor de R^2 conseguido revela que la correlación lineal entre el mes y la precipitación es bastante precaria, dado que el modelo solo da cuenta de una porción mínima de la variabilidad registrada en la precipitación. A pesar de que el análisis informa de una pendiente positiva de 0.2953, el bajo coeficiente de determinación subraya que el mes, analizado de forma independiente mediante un modelo lineal, no es suficiente para esclarecer las variaciones detectadas en la precipitación.

Asimismo, el RMSE obtenido fue de 11.0831 mm, lo que se traduce en la magnitud aproximada del desvío del modelo con respecto a los datos reales. Estos hallazgos ponen de manifiesto la relevancia de exponer no solo la ecuación resultante, sino también indicadores que faciliten la valoración de la fiabilidad del ajuste.

La visualización gráfica de este análisis se expone en la Figura 5-7.



Figura 5 - 7. Representación de la regresión lineal de precipitación respecto al mes para la estación el dorado (Fuente: Elaboración propia).

El gráfico deja ver cómo se dispersan los datos y la línea que trazó el modelo. El bajo poder explicativo que marcó el R^2 se puede ver también a simple vista, fijándose en lo lejos que están los registros de la línea de regresión. Con este experimento pudimos confirmar que el proyecto es capaz de entender una petición de estadística, elegir la herramienta adecuada, hacer los cálculos con los datos climatológicos y, después, explicar los resultados con palabras claras.

5.2.3. Generación de reportes EDA

Se evaluó también la generación automática de informes de análisis exploratorio de datos. Para ello, se hizo la siguiente petición:

"Genérame un reporte en HTML de la estación el dorado."

Una vez procesada la consulta, el agente usó la herramienta pertinente y avisó al usuario que el informe se había creado sin problemas. Adicionalmente, le dio un enlace para acceder directamente al archivo HTML generado como se ve en la Figura 5-8.



Figura 5 - 8. Generación de un reporte EDA en formato HTML (Fuente: Captura tomada del proyecto).

A diferencia de lo que ocurre en una respuesta conversacional estándar, esta función te da la posibilidad de conseguir un documento con información más específica sobre los datos de una estación meteorológica concreta.

El informe que se generó para la estación 25161 se puede ver en la Figura 5-9.

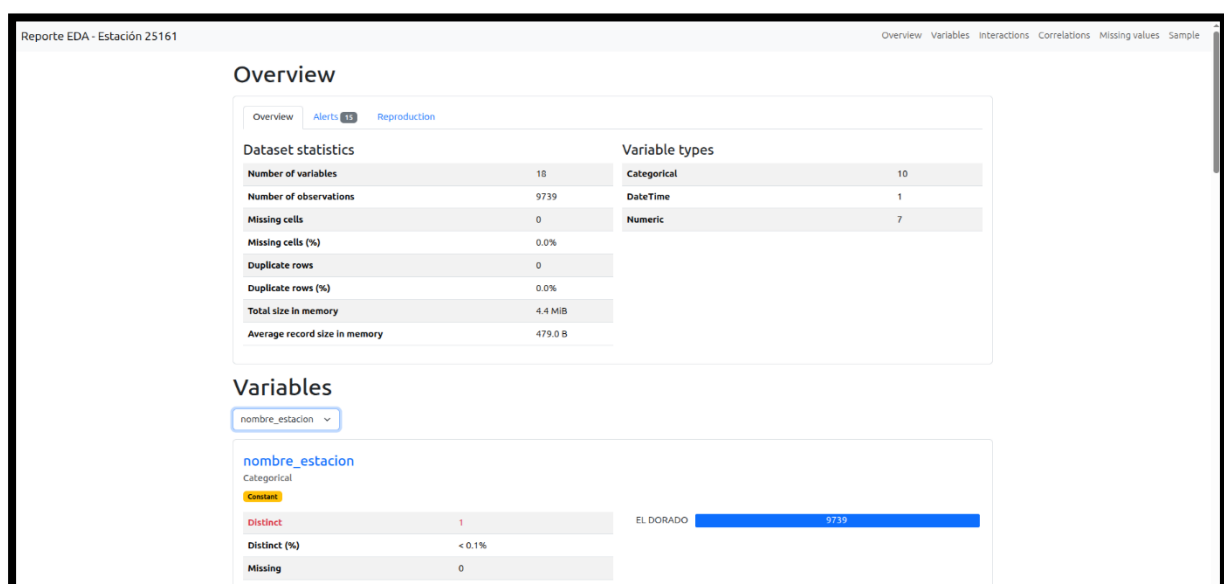


Figura 5 - 9. Reporte de análisis exploratorio de datos generado para la estación 25161 (El dorado)(Fuente: Captura tomada del proyecto).

En el reporte se consiguió un resumen general del conjunto de datos empleado para el análisis. Entre los hallazgos observados se cuentan:

- 18 variables.
- 9.739 observaciones.
- 0 celdas vacías.
- 0 filas duplicadas.
- 10 variables de tipo categórico.
- 1 variable de fecha y hora.
- 7 variables de tipo numérico.
- Un tamaño aproximado en memoria de 4.4 MiB.

Estos resultados dejan confirmar que el conjunto de datos empleado para esta estación está bien estructurado y sin valores ausentes en los registros revisados.

El reporte también deja indagar de manera individual las variables disponibles, así como consultar información referente a distribuciones, interacciones, correlaciones, valores faltantes y ejemplos de los registros.

La creación de este tipo de reportes complementa la interacción conversacional, dado que deja al usuario pasar de una consulta concreta a una exploración más a fondo de los datos sin necesidad de usar directamente herramientas de programación o bibliotecas de análisis.

5.2.4. Evaluación del flujo completo

Las pruebas que se hicieron también sirvieron para ver cómo se integraban las distintas partes del sistema. El proceso que vimos durante los experimentos se puede resumir así:

Petición del usuario → interpretación por el agente → elección de herramienta → lectura de los datos climáticos → realización del análisis → obtención de resultados → elaboración de la respuesta → muestra en la interfaz.

En las pruebas que llevamos a cabo, el agente pudo elegir distintas operaciones según lo que el usuario quería. Una pregunta sobre estaciones necesita una operación para buscar información, mientras que una petición de regresión necesita usar una herramienta estadística y una solicitud de reporte necesita crear un archivo HTML.

Esto prueba que el modelo de lenguaje no se usa para obtener directamente los números. Los análisis se hacen con herramientas que trabajan con los registros climáticos guardados en los datos, y el modelo de lenguaje se encarga de entender la petición y de presentar los resultados que se consiguen.

5.2.5. Resultados generales

Tras los experimentos que llevamos a cabo, pudimos comprobar que el proyecto es capaz de ejecutar el flujo principal que teníamos previsto. Este consiste en consultar información climática usando lenguaje natural, elegir automáticamente una herramienta de análisis, procesar los registros necesarios y mostrar los resultados de manera que el usuario pueda entenderlos fácilmente.

Los experimentos también permitieron ver que el sistema se adapta a distintos niveles de complejidad. Las preguntas pueden ir desde algo tan simple como pedir las estaciones disponibles hasta ejecutar procedimientos estadísticos complejos o generar informes completos de análisis exploratorio.

No obstante, los resultados obtenidos dependen de la información que se incluya en la pregunta. Para ciertos análisis, es indispensable indicar datos como la estación climática específica, la variable que queremos estudiar o el período de tiempo de interés. Si la solicitud no trae consigo suficiente información, hay que añadir los detalles que faltan para poder ejecutar el análisis correctamente.

Capítulo 6

6. Conclusión y trabajo futuro

En este capítulo se explica las conclusiones y los trabajos futuros del proyecto.

6.1. Conclusión

En resumen, el desarrollo de un sistema que pueda efectuar análisis hidrometeorológicos usando lenguaje natural se presenta como una alternativa para simplificar la consulta y el estudio de datos climatológicos históricos. El proyecto logró fusionar el procesamiento de datos, herramientas de análisis estadístico y un agente de inteligencia artificial en una interfaz web diseñada para facilitar la interacción del usuario con la información.

Durante la ejecución del proyecto, se procedió a la adquisición, limpieza, transformación y unificación de registros climatológicos provenientes de CONAGUA. Como resultado de esta labor, se compiló un conjunto de datos organizado en formato Apache Parquet, el cual sirve como la fuente primaria de información para las herramientas de análisis del sistema.

La puesta en marcha del agente a través de LangGraph posibilitó la coordinación de diversas herramientas especializadas para llevar a cabo operaciones como la filtración de registros, estadísticas descriptivas, análisis de tendencias, correlaciones, regresiones y la creación de informes de análisis exploratorio de datos (EDA). De este modo, el usuario tiene la capacidad de solicitar distintos tipos de análisis empleando lenguaje natural, sin necesidad de interactuar directamente con los archivos de datos ni de escribir código para su procesamiento.

La incorporación del modelo LLaMA 3.3 70B Versatile mediante Groq facilitó la interpretación de las peticiones del usuario y la presentación de los resultados de una forma clara. No obstante, los cálculos y los resultados estadísticos se obtienen a través de herramientas que operan directamente sobre el conjunto de datos climatológicos, evitando así depender exclusivamente del conocimiento que emana del modelo de lenguaje.

Los experimentos llevados a cabo permitieron validar el funcionamiento del flujo principal del proyecto. El sistema demostró ser capaz de consultar las estaciones disponibles, realizar análisis estadísticos sobre estaciones concretas y generar informes de análisis exploratorio en formato HTML. Estos resultados evidenciaron la cohesión entre la interfaz web, el agente de inteligencia artificial, las herramientas de análisis y el conjunto de datos previamente procesado.

Según los resultados obtenidos, consideramos que el proyecto cumple con el objetivo de ofrecer una herramienta que facilite la consulta y el análisis de información climatológica histórica a través de una interacción conversacional. El sistema permite acercar los procesos de análisis de datos a usuarios que no necesariamente tienen conocimientos de programación o de manejo directo de herramientas estadísticas.

Sin embargo, la implementación que se desarrolló es un prototipo y aún tiene margen de mejora. Entre estas mejoras se incluye la incorporación de más estaciones y variables climatológicas, el refuerzo de la validación de las respuestas que genera el sistema y la ampliación de las herramientas estadísticas que están disponibles.

6.2. Trabajo futuro

Como parte de la continuación del proyecto, se contemplan las siguientes direcciones para el trabajo futuro:

- Implementar una arquitectura multiagente, donde distintos agentes con especializaciones puedan asumir tareas concretas, tales como el análisis estadístico, la creación de informes, la obtención de información y la interpretación de hallazgos.
- Integrar un sistema RAG (Retrieval-Augmented Generation) que facilite la recuperación y el análisis de datos procedentes de documentos técnicos, informes, boletines y otras fuentes oficiales vinculadas a la información climatológica e hidrometeorológica.
- Extender las competencias analíticas del agente, añadiendo nuevos enfoques estadísticos y algoritmos para posibilitar análisis de mayor complejidad, así como la generación de informes con un nivel de detalle superior.
- Ampliar el área geográfica de operación del sistema, sumando estaciones climatológicas de otros estados de la República Mexicana, con el objetivo de ir extendiendo gradualmente la cobertura del proyecto a todo el país.
- Reforzar la evaluación del sistema, a través de un mayor volumen de pruebas con usuarios y consultas sobre climatología, lo que permitirá cuantificar aspectos como la exactitud de las herramientas seleccionadas, la calidad de las respuestas y la sencillez en el manejo de la plataforma.

Bibliografías

- Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y., & Demir, I. (2020). A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, 82(12), 2635-2670.
- World Meteorological Organization (WMO). (2023). *Guide to Climatological Practices*. Geneva, Switzerland: WMO.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Provost, F., & Fawcett, T. (2023). *Data Science for Business* (2nd ed.). O'Reilly Media.
- Raschka, S. (2024, 11 septiembre). *Build a Large Language Model (From Scratch)*. O'Reilly Online Learning, Pag. 2-7.
- Taylor, J. P. M. (2024, 16 mayo). *Prompt Engineering for Generative AI*. O'Reilly Online Learning, Pag. 20.
- Klesel, M., & Wittmann, H. F. (2025). Retrieval-augmented generation (rag) m. klesel, hf wittmann. *Business & Information Systems Engineering*, 67(4), Pag. 551-561.
- Sapkota, R., Roumeliotis, K. I., & Karkee, M. (2025). Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *Information Fusion*, 103599.
- Masterman, T., Besen, S., Sawtell, M., & Chao, A. (2024). The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*.
- Tahseen, A., Sumathi, J., Reddy, S., & Shravani, D. (2025). RAG-Based Query Engine using LLM and Vector DB for College Details. *Journal of Frontiers in Multidisciplinary Research*, 6(1), 86-91.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... Riedel, S. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.
- Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S., Konwinski, A., & Stoica, I. (2022). Data-centric AI: Perspectives and Challenges.
- Rajkumar, P., Li, X., & others. (2022). Evaluating the Text-to-SQL Capabilities of Large Language Models.
- Li, Y., Hui, B., Qu, Y., Yang, H., & others. (2023). Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation.
- Yulianto, A. A. (2019). Extract transform load (ETL) process in distributed database academic data warehouse. *APTİKOM Journal on Computer Science and Information Technologies*, 4(2), 61–68. <https://doi.org/10.11591/aptikom.j.csit.36>

Çetin, V., & Yıldız, O. (2022). A comprehensive review on data preprocessing techniques in data analysis. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 28(2), 299-312.

Galatro, D., & Dawe, S. (2023). Exploratory Data Analysis. In *Data Analytics for Process Engineers: Prediction, Control and Optimization* (pp. 13-57). Cham: Springer Nature Switzerland.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R* (2nd ed.). Springer.

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Linear regression. In *An introduction to statistical learning: With applications in python* (pp. 69-134). Cham: Springer international publishing.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R* (2nd ed.). Springer.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to Linear Regression Analysis* (6th ed.). Wiley.

Batini, C., & Scannapieco, M. (2020). *Data and Information Quality: Dimensions, Principles and Techniques*. Springer.

Gundersen, O. E., & Kjensmo, S. (2021). State of the Art: Reproducibility in Artificial Intelligence. *Proceedings of the AAAI Conference*.

Amazon Web Services. (s. f.). *¿Qué es una aplicación web?* Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/web-application/>

Kiesler, N., & Schiffner, D. (2023, September). What is a good api? a survey on the use and design of application programming interfaces. In *International Conference on Internet of Everything* (Pag. 45-55). Cham: Springer Nature Switzerland.

Lubanovic, B. (2023). *FastAPI*. " O'Reilly Media, Inc."

Lazuardy, M. F. S., & Anggraini, D. (2022). Modern front end web architectures with react.js and next.js. *Research Journal of Advanced Engineering and Science*, 7(1), Pag. 132-141.

George, A. S. (2025). The critical role of micro-segmentation in modern cybersecurity architectures: A comprehensive review. *Partners UniversalMultidisciplinary Research Journal*, 2(2), Pag. 24-34.

Vaughn, C., SHERMAN-MORRIS, K. A. T. H. L. E. E. N., GUTTER, B., & BROWN, M. (2023). A Change in the Weather: Understanding Public Usage of Weather Apps. *Journal of Operational Meteorology*, 11(11).

DePodwin, D., Grow, E., & Porter, J. (2023, January). Pioneering the Role of the Digital Meteorologist at AccuWeather-Blending the Best Forecasters and Communicators to Inform Users Through Digital Platforms. In *American Meteorological Society Meeting Abstracts* (Vol. 103, pp. 5B-5).

Yan, B., Li, K., Xu, M., Dong, Y., Zhang, Y., Ren, Z., & Cheng, X. (2025). On protecting the data privacy of Large Language Models (LLMs) and LLM agents: A literature review. *High-Confidence Computing*, 5(2), 100300.

Mavroudis, V. (2024). LangChain.

Arslan, M., Ghanem, H., Munawar, S., & Cruz, C. (2024). A Survey on RAG with LLMs. *Procedia computer science*, 246, 3781-3790.

Anexos

A. Manual De Usuario

A.1. Objetivo del manual

El propósito de este manual es explicar cómo funciona la interfaz web del proyecto y darte las indicaciones que necesitas para buscar y analizar información del clima.

El proyecto te deja trabajar con los datos usando un lenguaje normal y corriente. Así, no hace falta que sepas de programación, comandos especiales o herramientas estadísticas para usar las funciones principales del sistema.

Con la aplicación, el usuario puede ver qué estaciones meteorológicas hay disponibles, pedir estadísticas, hacer análisis de correlación, regresión y tendencias, y también crear informes de análisis exploratorio de datos en formato HTML.

A.2. Acceso al proyecto

Para usar el proyecto, el usuario tiene que ir a la dirección web que se le da para correr la aplicación.

En el ambiente local que usamos para desarrollarlo, es posible ver la interfaz en:

<http://localhost:3000>

Cuando la aplicación se cargue, verás la pantalla principal del proyecto, daremos al botón continuar con los valores predeterminados. Desde ahí, es posible empezar una nueva conversación o seguir con alguno de los chats que ya tienes guardados en el historial.

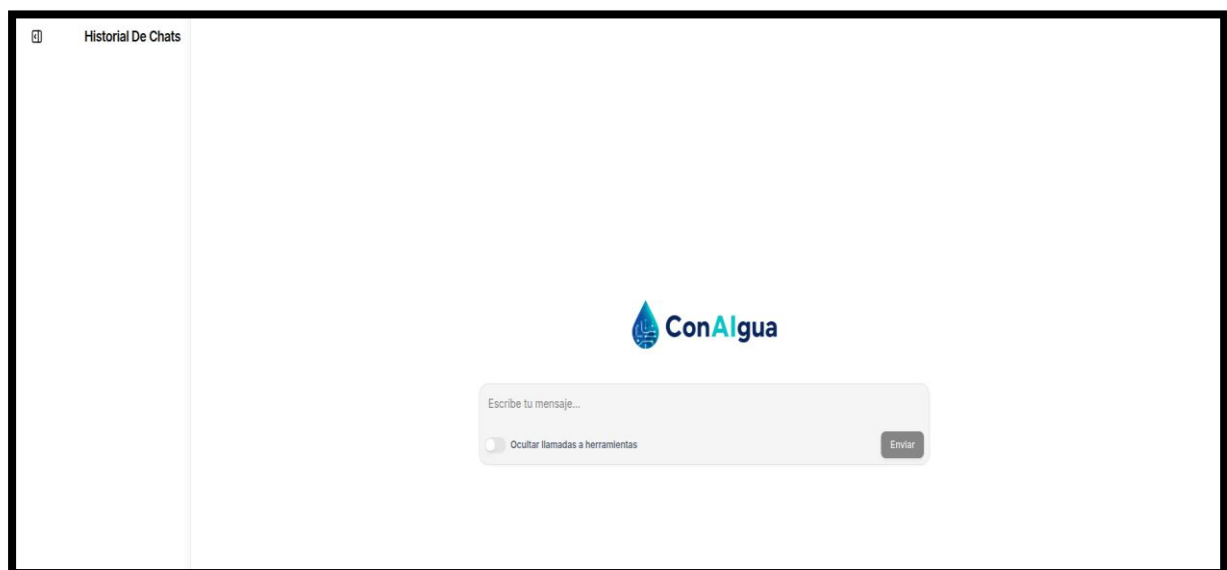


Figura A - 1. Pantalla principal de la aplicación (Fuente: Captura del proyecto).

A.3. Pantalla principal

La interfaz principal del proyecto está pensada para facilitar una interacción simple y enfocada en la conversación con el agente.

La pantalla se divide, básicamente, en dos áreas:

Historial de chats

Justo en la parte izquierda de la aplicación verás el historial de chats, que te muestra todas las conversaciones que has tenido al usar el sistema.



Figura A - 2. Sección historial de chats (Fuente: Captura del proyecto).

Área de conversación

La sección central representa el espacio primordial para interactuar con el agente. Aquí se exhiben tanto los mensajes redactados por el usuario como las respuestas que elabora el agente.

Justo debajo, hallarás el campo de texto que se usa para formular una nueva pregunta, junto con el botón para enviarla.

Adicionalmente, la interfaz cuenta con una función para disimular las invocaciones a herramientas, lo cual posibilita ver únicamente el diálogo y sus resultados finales cuando el usuario no precise revisar los procedimientos internos que emplea el agente.



Figura A - 3. Sección escribirle al agente (Fuente: Captura del proyecto).

A.4. Realizar una consulta

Para hacer una consulta, el usuario tiene que seleccionar el cuadro de texto que está en la parte de abajo de la pantalla y escribir ahí la pregunta o lo que sea que le quiera decir al agente.

Por ejemplo:

“¿Qué estaciones tienen disponibles?”

Cuando ya esté escrita la consulta, se presiona el botón de Enviar.

El agente revisa lo que pidió y se muestra la respuesta en el área donde conversamos.

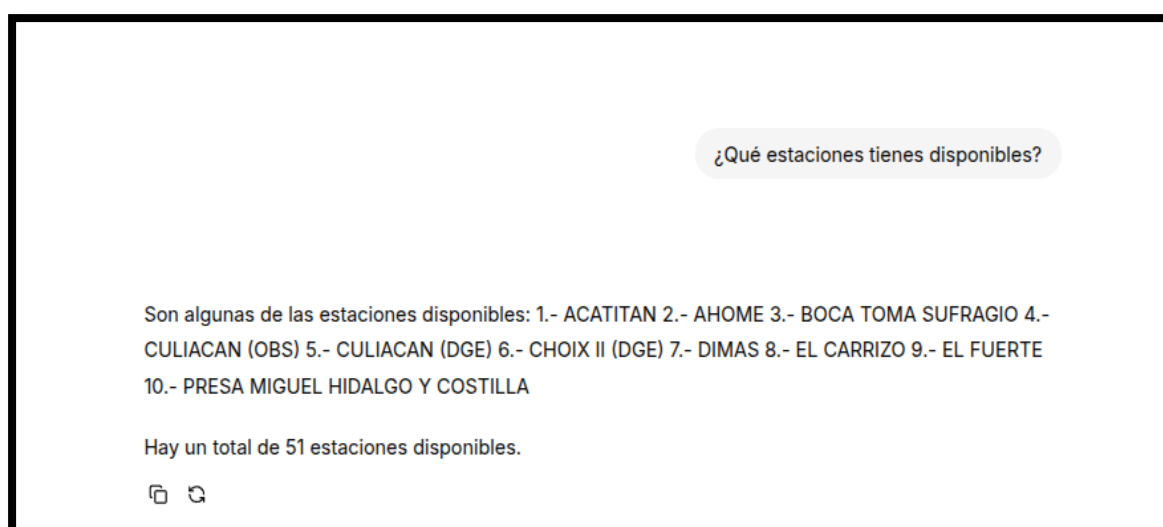


Figura A - 4. Resultado estaciones disponibles (Fuente: Captura del proyecto).

El sistema deja usar lenguaje natural, así que no hay que aprender ninguna sintaxis rara. Aun así, cuanto más detallada sea tu petición, más datos tendrá el agente para hacer bien el análisis.

A.5. Consultar estadísticas climatológicas

El agente te deja pedir estadísticas sobre las variables del clima que tenemos.

Las variables más importantes que se pueden preguntar son:

- Precipitación.
- Evaporación.
- Temperatura máxima.
- Temperatura mínima.
- El mes.
- El año.

Por ejemplo, si se requiere saber cómo se comporta la lluvia:

“Quiero saber el promedio de lluvia de la estación Acatitan.”

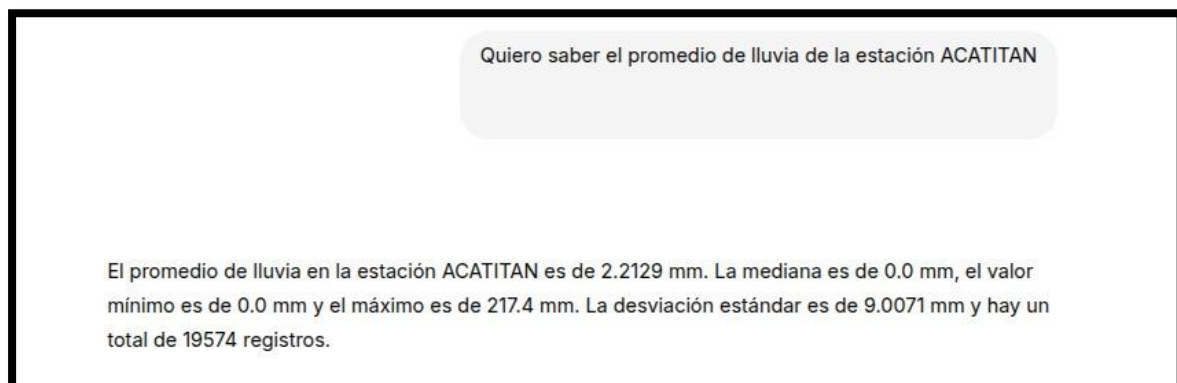


Figura A - 5. Resultado promedio de lluvia de la estación acatitan (Fuente: Captura del proyecto).

También se acotar la consulta a un tiempo específico:

“Quiero saber el promedio de lluvia de la estación Acatitan para el año 1978.”

Quiero saber el promedio de lluvia de la estación Acatitan para el año 1978.

El promedio de lluvia en la estación Acatitan para el año 1978 es de 1.5978 mm. La mediana es de 0.0 mm, el valor mínimo es de 0.0 mm y el máximo es de 64.4 mm. La desviación estándar es de 6.4012 mm y hay un total de 365 registros.

Figura A - 6. Resultado promedio de lluvia de la estación acatitan (Fuente: Captura del proyecto).

El agente se da cuenta de qué estación, qué variable y qué periodo quieres, hace los cálculos necesarios y luego te da la respuesta ahí mismo en la conversación.

A.6. Solicitar análisis de correlación

El usuario puede pedir un análisis de correlación entre dos variables del clima.

Por ejemplo:

“¿Cuál fue la correlación entre precipitación y temperatura máxima en la estación el dorado?”

“¿Cuál fue la correlación entre precipitación y temperatura máxima en la estación El dorado?”



La correlación entre precipitación y temperatura máxima en la estación El Dorado es muy baja según la correlación de Pearson, con un coeficiente de 0.0152 y un p-valor de 0.1337, lo que indica que no es estadísticamente significativa. Sin embargo, la correlación de Spearman muestra un coeficiente de 0.1183 con un p-valor de 0.0, lo que sugiere una correlación débil pero estadísticamente significativa. Hay un total de 9739 observaciones.

Figura A - 7. Resultado correlación entre precipitación y temperatura máxima en la estación el dorado (Fuente: Captura del proyecto).

El sistema procesa los registros correspondientes y presenta los resultados del análisis. Según la operación que se lleve a cabo, la respuesta podría incluir el coeficiente de correlación y una interpretación de cuán relacionadas están las variables analizadas.

A.7. Solicitar análisis de regresión

El agente cuenta con herramientas para llevar a cabo análisis de regresión. Una consulta de regresión lineal simple se puede efectuar de la siguiente manera:

“Calcula la regresión lineal de precipitación contra mes en la estación el dorado.”

Después de procesar la solicitud, el agente dará diferentes métricas asociadas al modelo, entre ellas:

- Pendiente.
- Intercepto.
- Coeficiente de determinación (R^2).
- Error cuadrático medio o RMSE.
- Intervalos de confianza.

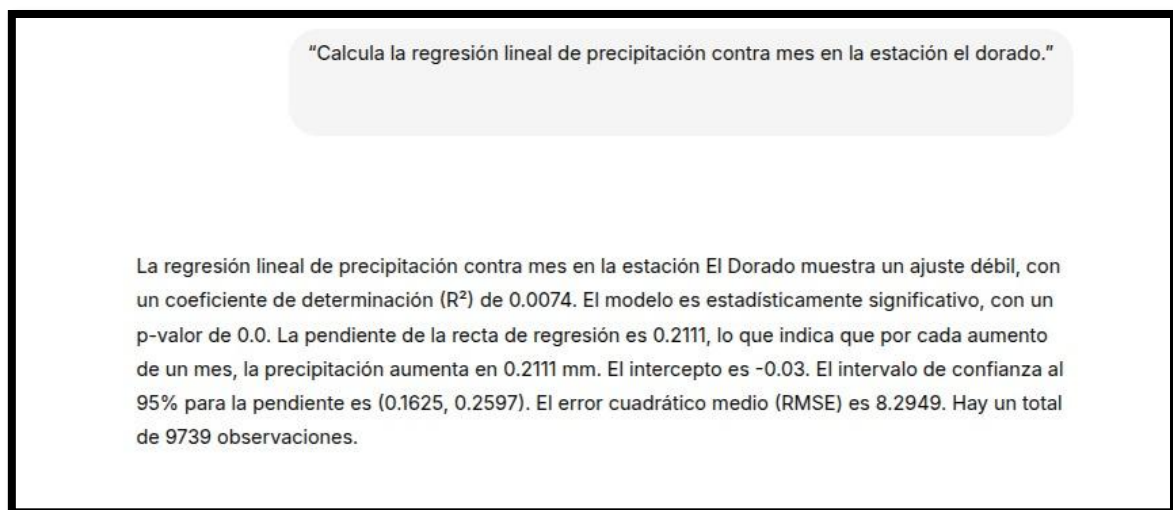


Figura A - 8. Resultado regresión lineal de precipitación contra mes en la estación el dorado (Fuente: Captura del proyecto).

Igualmente, se pueden pedir análisis que abarquen más de una variable.

Por ejemplo:

“Haz una regresión múltiple de precipitación contra mes y año en la estación Acatitan.”

"Haz una regresión múltiple de precipitación contra mes y año en la estación Acatitan."

La regresión múltiple de precipitación contra mes y año en la estación Acatitan muestra un ajuste débil, con un coeficiente de determinación (R^2) de 0.0118 y un R^2 ajustado de 0.0117. El intercepto es -8.7881. Los coeficientes para las variables independientes son 0.2824 para el mes y 0.0046 para el año. El error cuadrático medio (RMSE) es 8.9536. Hay un total de 19574 observaciones. Esto sugiere que tanto el mes como el año tienen un efecto pequeño en la precipitación, pero el modelo no explica gran parte de la variabilidad en la precipitación.



Figura A - 9. Regresión múltiple de precipitación contra mes y año en la estación Acatitan (Fuente: Captura del proyecto).

A.8. Generar un reporte EDA

El agente facilita la generación de informes de Análisis Exploratorio de Datos (EDA) para una estación climatológica específica.

Para pedir un informe, el usuario puede ingresar una consulta del tipo:

“Genérame un reporte en HTML de la estación el dorado.”

El agente localiza la estación que se pidió y pone en marcha la herramienta que sirve para hacer el reporte.

Una vez que termina todo el proceso, El agente presenta un mensaje que confirma que el archivo se ha generado y da acceso a él a través de un enlace para poder verlo.

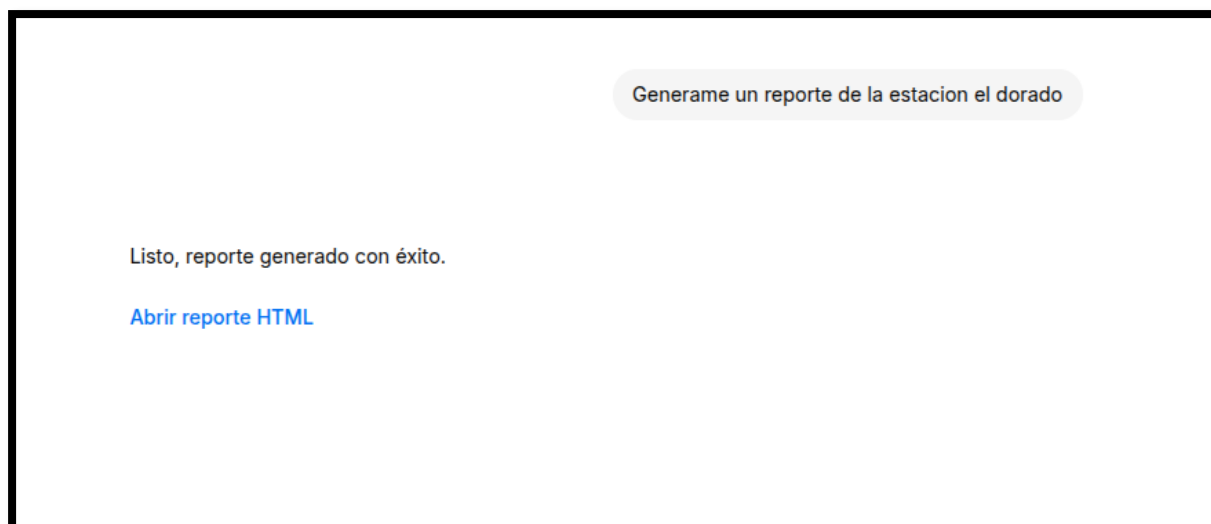


Figura A - 10. Enlace para el archivo HTML generado (Fuente: Captura del proyecto).

A.9. Visualización del reporte

Al abrir el reporte que se genera, el usuario tendrá acceso a información detallada sobre los registros del clima de la estación que haya escogido.

El reporte EDA puede incluir datos como:

- Número de registros analizados.
- Número y tipo de variables.
- Valores faltantes.
- Registros duplicados.
- Estadísticas descriptivas.
- Distribución de variables.
- Correlaciones.
- Interacciones entre variables.
- Muestras de los datos analizados.

Esta característica permite enriquecer las respuestas que da directamente el chatbot, añadiendo una visión más completa del conjunto de datos.

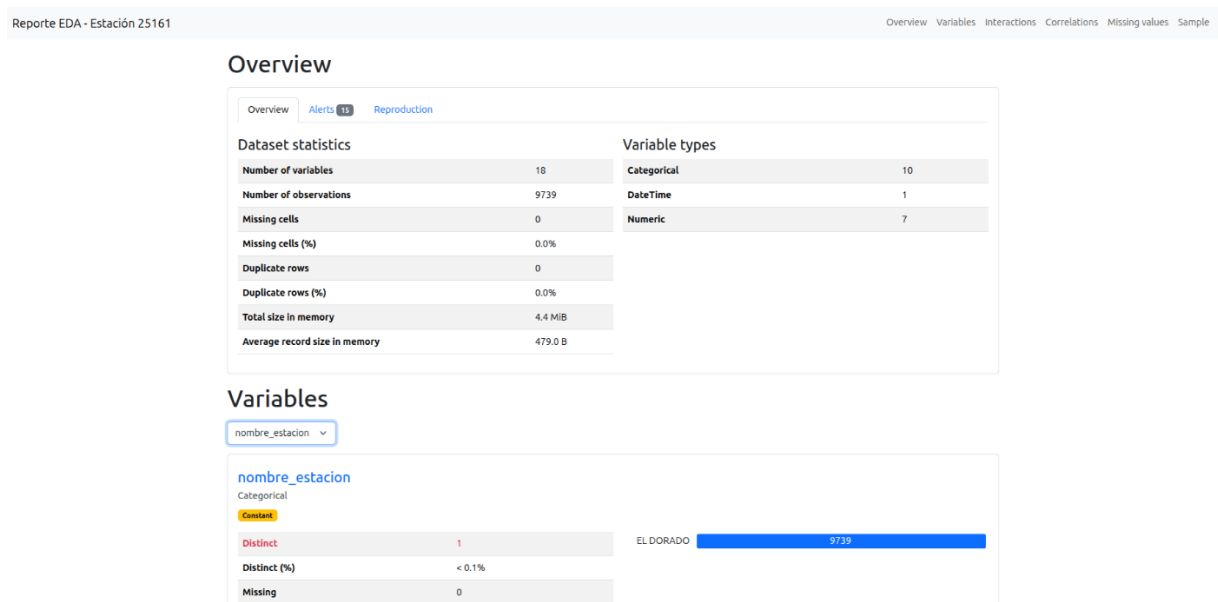


Figura A - 11. Visualización del reporte EDA generado por el agente (Fuente: Captura del proyecto).

A.10. Ocultar llamadas a herramientas

Al procesar ciertas peticiones, el agente podría valerse de varias herramientas internas para revisar o examinar datos.

La interfaz trae consigo una funcionalidad llamada Ocultar llamadas a herramientas, pensada para que dichos detalles técnicos no resten fluidez a la lectura del diálogo.

Si activas esta opción, verás, ante todo:

- La pregunta que hiciste.
- La respuesta definitiva del agente.
- Los hallazgos del análisis.

Esta funcionalidad no altera la respuesta a tu consulta; solo ajusta la información técnica que se presenta en pantalla.

B. Manual Tecnico

B.1. Objetivo

El presente manual técnico describe los requisitos, la configuración, la instalación, la ejecución y las tareas básicas de mantenimiento del sistema. Está dirigido a desarrolladores o personal técnico encargado de preparar el entorno, ejecutar los componentes del proyecto y verificar su funcionamiento.

El sistema está compuesto por un agente de inteligencia artificial desarrollado en Python, una interfaz web construida con React y Next.js, herramientas de análisis hidrometeorológico, un conjunto de datos procesado en formato Apache Parquet y un modelo de lenguaje externo accesible mediante Groq.

B.2. Componentes del sistema

- Interfaz web: aplicación desarrollada con React y Next.js para la interacción conversacional con el usuario.
- Agente: componente desarrollado en Python y orquestado mediante LangGraph.
- Integración con LangChain: soporte para el modelo de lenguaje y las herramientas del agente.
- Herramientas de análisis: estadísticas, correlaciones, regresiones, tendencias y generación de reportes.
- Dataset climatológico: registros procesados y almacenados en formato Apache Parquet.
- Modelo de lenguaje: LLaMA 3.3 70B Versatile mediante la API de Groq.
- Reportes EDA: archivos HTML y Markdown generados por estación.

NIVEL 3: Componentes
Descomposición del contenedor Agente ConAlgua

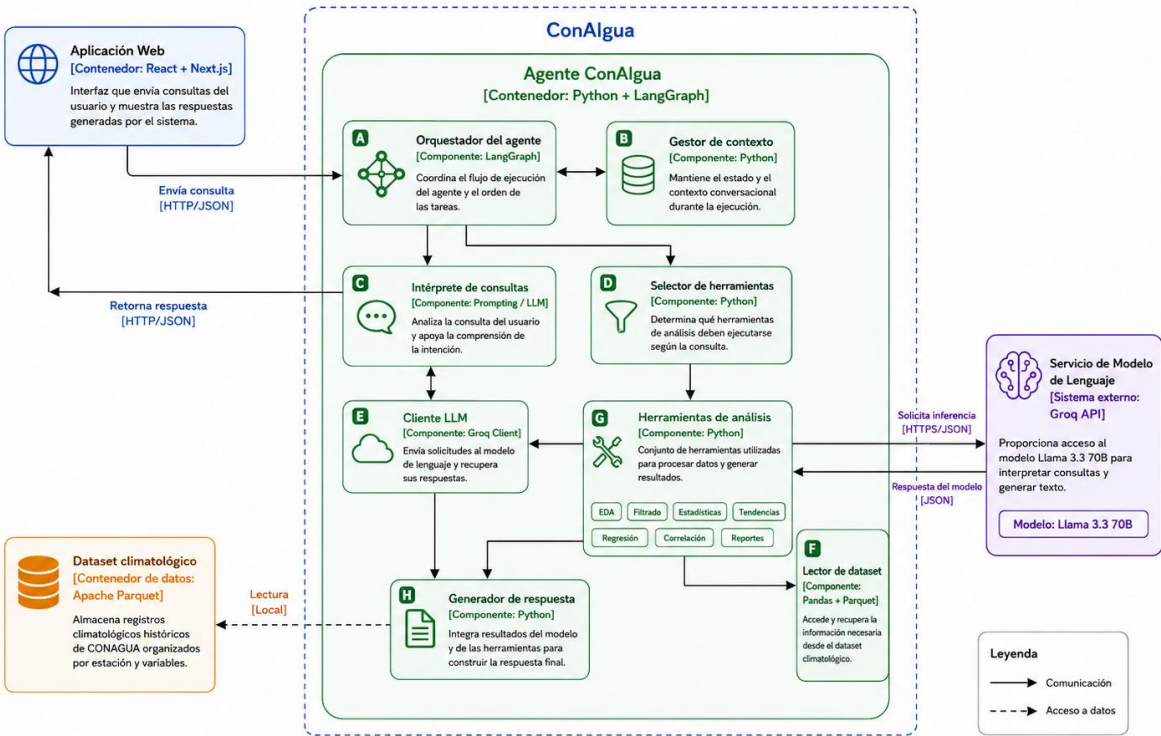


Figura B - 1. Componentes principales del agente (Fuente: Elaboración propia).

B.3. Requisitos técnicos

Componente	Requisito	Uso
Agente	Python 3.12	Ejecución del código y herramientas de análisis.
Agente	uv	Gestión del entorno y dependencias de Python.
Proyecto	Git	Obtención y actualización del código fuente.
Modelo LLM	API Key de Groq	Acceso al modelo de lenguaje.
Interfaz	Docker / Docker Compose	Ejecución reproducible del frontend.
Interfaz	Conexión a Internet	Descarga inicial de dependencias.

Tabla B - 1. Tabla requisitos técnicos.

B.4. Instalación del agente

B.4.1. Clonar el repositorio

Desde una terminal, se obtiene el código fuente del proyecto y se accede al directorio del agente:

```
git clone https://github.com/Chris6264/ConAIgua.git
cd ConAIgua/conaigua-agent
```

B.4.2. Crear el entorno virtual

Se recomienda utilizar un entorno virtual con Python 3.12 para aislar las dependencias del proyecto:

```
uv venv --python 3.12 .venv
```

Como alternativa, puede utilizarse el módulo venv incluido con Python:

```
python -m venv .venv
```

B.4.3. Activar el entorno virtual

En Windows PowerShell:

```
.venv\Scripts\Activate.ps1
```

En Linux o macOS:

```
source .venv/bin/activate
```

B.4.4. Instalar las dependencias

Con el entorno virtual activo, se instalan las dependencias del proyecto en modo editable:

```
uv pip install -e .
```

Si no se utiliza uv, puede emplearse pip:

```
pip install -e .
```

B.5. Configuración del agente

B.5.1. Variables de entorno

El repositorio incluye el archivo .env.example. Se debe crear una copia denominada .env antes de iniciar el agente.

Windows:

```
copy .env.example .env
```

Linux o macOS:

```
cp .env.example .env
```

Dentro del archivo `.env` se configura el proveedor y la clave de acceso. La clave real no debe publicarse ni almacenarse en documentación compartida:

```
LLM_PROVIDER="groq"  
GROQ_API_KEY="tu_api_key_aqui"
```

B.5.2. Archivo `config.yaml`

El archivo de configuración del agente se crea a partir de `config.example.yaml` y permite definir el proveedor, el modelo y los parámetros básicos de generación.

```
llm:  
  provider: groq  
  model: llama-3.3-70b-versatile  
  api_key_env: GROQ_API_KEY  
  temperature: 0  
  max_tokens: 1024
```

B.6. Preparación del dataset

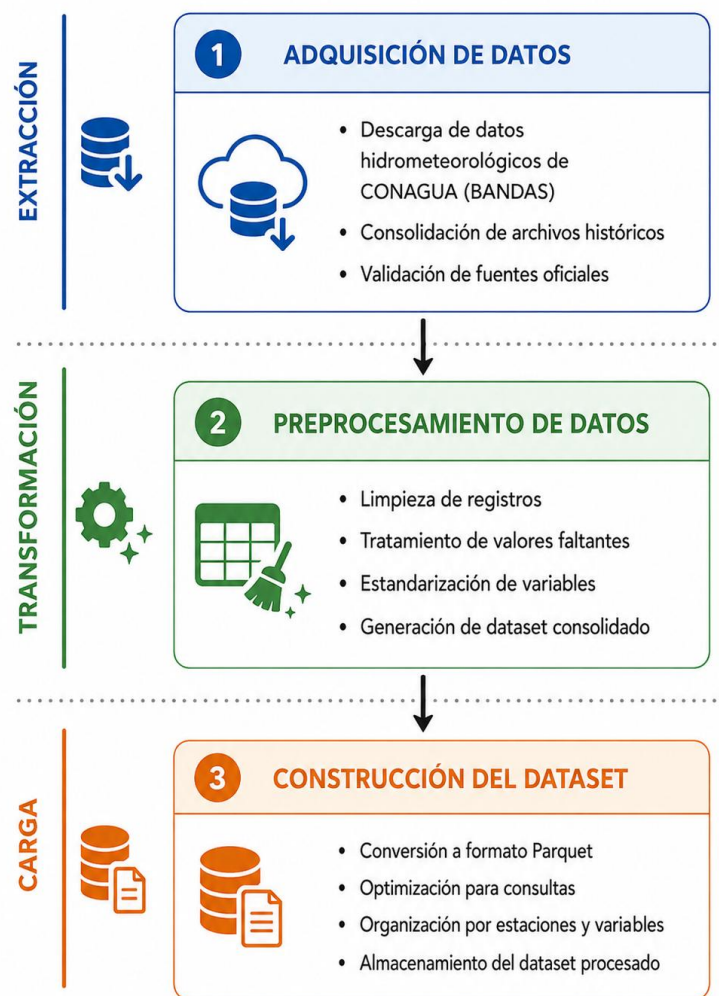
Los archivos históricos de las estaciones climatológicas deben colocarse en el directorio `data/raw/`. Posteriormente, el pipeline procesa los registros y genera el dataset consolidado utilizado por el agente.

```
python -m scripts.run_data_pipeline
```

El archivo resultante se almacena en:

```
data/processed/conAIgua_dataframe.parquet
```

Si el archivo Parquet ya se encuentra disponible, la generación puede omitirse.



Fuente: Elaboración propia.

Figura B - 2. Proceso ETL utilizado para preparar el dataset climatológico (Fuente: elaboración propia).

B.7. Ejecución del agente

El agente debe iniciarse desde la raíz del directorio conaigua-agent. El comando utilizado en el entorno del proyecto es:

```
PYTHONPATH="$PWD/src" uv run langgraph dev
```

La variable PYTHONPATH permite que Python localice correctamente el paquete conaigua dentro del directorio src. Para detener el proceso se utiliza Ctrl + C.

B.8. Ejecución y visualización de reportes EDA

Los reportes de análisis exploratorio se almacenan dentro de reports/eda/. Para visualizarlos desde un navegador se puede iniciar un servidor HTTP local:

```
cd reports/eda
python -m http.server 8088
```

Después de iniciar el servidor, los reportes quedan disponibles desde:

```
http://localhost:8088
```

Los reportes HTML se encuentran principalmente en `reports/eda/html/` y los archivos Markdown en `reports/eda/markdown/`. Para detener el servidor se utiliza `Ctrl + C`.

B.9. Generación anticipada de reportes

Si se desea generar los reportes de todas las estaciones antes de utilizar el agente, se ejecuta el pipeline de generación de reportes:

```
python -m scripts.run_reports_generator_pipeline
```

Los reportes generados se reutilizan posteriormente cuando el usuario los solicita, evitando regenerarlos innecesariamente.

B.10. Instalación y ejecución de la interfaz web

La interfaz web se encuentra en el directorio `conaigua-chat-ui` y fue desarrollada con React y Next.js. Docker Compose se utiliza para mantener un entorno reproducible y `pnpm` administra las dependencias del frontend.

B.10.1. Acceder al directorio de la interfaz

```
cd ConAIgua/conaigua-chat-ui
```

B.10.2. Instalar dependencias

Con Docker activo, se instalan las dependencias dentro del servicio web definido en Docker Compose:

```
docker compose run --rm web sh -c "corepack enable && pnpm install"
```

B.10.3. Iniciar la aplicación web

```
docker compose up
```

Una vez iniciado el contenedor, la interfaz puede abrirse desde el navegador en:

```
http://localhost:3000
```

Para detener la ejecución se utiliza Ctrl + C. Posteriormente, los contenedores pueden cerrarse con:

```
docker compose down
```

B.11. Orden recomendado de ejecución

- Verificar que data/processed/conAlgua_dataframe.parquet se encuentre disponible.
- Iniciar el agente con LangGraph.
- Iniciar el servidor HTTP de reportes EDA cuando se requiera visualizar archivos HTML.
- Iniciar la interfaz web mediante Docker Compose.
- Abrir <http://localhost:3000> en el navegador.
- Realizar las consultas desde la interfaz conversacional.

B.12. Capacidades técnicas del agente

Capacidad	Descripción
Estadísticas EDA	Media, mediana, mínimo, máximo, desviación estándar, outliers y estacionalidad.
Correlaciones	Pearson y Spearman con p-value y significancia estadística.
Regresiones	Lineal simple y múltiple con R^2 , RMSE, intervalos de confianza y diagnóstico.
Tendencias	Análisis de tendencia anual en rangos de años.
Reportes	Generación de HTML con ydata-profiling y reportes Markdown por estación.
Estaciones	Listado de las estaciones disponibles en el dataset.

Tabla B - 2. Tabla capacidades técnicas del agente.

B.13. Variables climatológicas disponibles

Variable	Descripción	Unidad
precip	Precipitación	mm
evap	Evaporación	mm
tmax	Temperatura máxima	°C
tmin	Temperatura mínima	°C
mes	Mes del año	1-12
año	Año del registro	-

Tabla B - 3. Tabla variables climatológicas disponibles.

B.14. Estructura principal del proyecto

La siguiente estructura resume los directorios y archivos principales del agente:

```
conaigua-agent/
├── config/
├── data/
│   ├── raw/
│   └── processed/
├── logs/
├── notebooks/
├── reports/
│   └── eda/
├── scripts/
├── src/
│   └── conaigua/
│       ├── core/
│       ├── data_pipeline/
│       ├── eda_agent/
│       ├── eda_engine/
│       ├── eda_reports_generator/
│       ├── orchestration/
│       └── utils/
├── tests/
├── .env.example
├── langgraph.json
├── pyproject.toml
├── README.md
└── uv.lock
```

Directorio	Función
config/	Configuración del modelo y del agente.
data/raw/	Archivos climatológicos originales.
data/processed/	Dataset procesado en formato Parquet.
logs/	Registros de eventos, métricas y errores.
reports/	Reportes EDA generados por el sistema.
scripts/	Pipelines y procesos auxiliares.
src/conaigua/	Código fuente principal del agente.
tests/	Pruebas automatizadas e integración.

Tabla B - 4. Tabla directorios del proyecto.

B.15. Pruebas automatizadas

Este proyecto incluye pruebas de la lectura y limpieza de datos, contratos internos, configuración del agente, herramientas, la integración del pipeline, el flujo extremo a extremo y todas las pruebas incluídas en el directorio tests/.

Para ejecutar este conjunto de pruebas se puede utilizar:

```
pytest
```

Para ejecutar un archivo específico y mostrar mayor detalle:

```
pytest tests/nombre_de_la_prueba.py -v
```

B.16. Logs y trazabilidad

El proyecto conserva los archivos de registro en el directorio logs/. Estos archivos permiten revisar errores, métricas, eventos generados durante la ejecución del pipeline y del agente.

```
logs/  
├─ errors.log  
├─ metrics.json  
├─ pipeline_events.jsonl  
└─ pipeline.log
```

El README también documenta el registro de interacciones del agente en formato JSON, lo que facilita la depuración y el seguimiento del comportamiento del sistema.

B.17. Consideraciones técnicas

- El agente requiere que el dataset procesado esté disponible antes de realizar análisis.
- La API Key debe almacenarse en variables de entorno y no debe publicarse en el repositorio.
- El agente debe ejecutarse desde la raíz de conaigua-agent para mantener las rutas esperadas.
- Docker debe encontrarse activo antes de ejecutar la interfaz web.
- El servidor HTTP de reportes debe estar activo cuando se quiera abrir un reporte HTML mediante el navegador.
- La memoria conversacional utiliza InMemorySaver; el contexto se mantiene durante la sesión y se reinicia al cerrar el agente.
- El agente está limitado al dominio de análisis hidrometeorológico definido.

B.18. Solución de problemas comunes

B.18.1. ModuleNotFoundError: No module named conaigua

Si Python no localiza el paquete conaigua, se debe confirmar que el comando se ejecute desde la raíz del proyecto utilizando la ruta src:

```
PYTHONPATH="$PWD/src" uv run langgraph dev
```

B.18.2. Docker indica “no such service”

Los comandos de Docker Compose deben utilizar el nombre del servicio definido en el archivo de composición. En la configuración actual del frontend, el servicio utilizado es web:

```
docker compose config --services  
docker compose run --rm web sh -c "corepack enable && pnpm install"
```

B.18.3. Docker no puede conectarse al motor

Si Docker muestra un error relacionado con dockerDesktopLinuxEngine o con el daemon, se debe comprobar que Docker Desktop se encuentre iniciado antes de volver a ejecutar Docker Compose.

B.18.4. El reporte HTML no puede abrirse

Se debe verificar que el servidor HTTP de reportes se encuentre activo y que se haya iniciado desde reports/eda/:

```
cd reports/eda  
python -m http.server 8088
```

B.19. Cierre del sistema

Para finalizar la ejecución, el agente y el servidor de reportes se detienen con Ctrl + C. La interfaz puede detenerse de la misma forma y posteriormente deben cerrarse los contenedores con:

```
docker compose down
```

Con estos pasos se finalizan los procesos utilizados por el agente y se liberan los recursos asociados a la ejecución local.