

UNIVERSIDAD NACIONAL DEL CENTRO DEL PERÚ



FACULTAD DE INGENIERÍA DE SISTEMAS MONOGRAFÍA

**CHATBOT AGRÍCOLA COMO HERRAMIENTA DE APOYO AL
AGRICULTOR PARA LA SIEMBRA Y COSECHA DE MAÍZ EN
EL DISTRITO DE HUALHUAS**

PRESENTADO POR:

Aranda Gomez, Zarella Andrea

Cerron Pizarro Sebastian

Huaman Rojas, Jhordan Armando

Macha Pariona, Angel Yoelver

Mallqui Meza, Jhamir Edu

Quispe Ortiz, Jhosep Frank

HUANCAYO – PERÚ

2025

AGRADECIMIENTOS

En primer lugar, agradezco a Dios, por brindarme la vida, la fortaleza y la sabiduría necesarias para culminar esta etapa importante de mi formación profesional. Su guía ha sido fundamental en los momentos de duda, y su presencia constante me ha permitido seguir adelante con esperanza y perseverancia.

Agradezco profundamente al Ingeniero Lozano, por su compromiso, orientación académica y disposición constante para brindarme apoyo durante el desarrollo de este proyecto. Su experiencia y consejos han sido clave para enriquecer el enfoque técnico de esta investigación, así como para fortalecer mis capacidades personales y profesionales.

ÍNDICE GENERAL

AGRADECIMIENTOS	2
ÍNDICE GENERAL	3
LISTA DE FIGURAS.....	5
LISTA DE TABLAS.....	5
RESUMEN.....	6
INTRODUCCIÓN	7
1. CAPÍTULO I: PLANTEAMIENTO DEL ESTUDIO	8
1.1. Planteamiento y formulación del problema.....	8
1.1.1. Enunciado del problema	8
1.1.2. Formulación del problema	8
1.2. Objetivos	8
1.2.1. Objetivo general	8
1.2.2. Objetivos específicos	8
1.3. Justificación e importancia	9
2. CAPÍTULO II: MARCO TEÓRICO	10
2.1. Antecedentes del problema	10
2.2. Bases teóricas.....	10
2.2.1. Inteligencia Artificial (IA)	10
2.2.2. Procesamiento de Lenguaje Natural (PLN).....	11
2.2.3. Chatbots: Definición y Tipología.....	11
2.2.4. Agricultura Inteligente y Chatbots	11
2.3. Definición de términos básicos	12
3. CAPÍTULO III: METODOLOGÍA.....	14
3.1. Diseño	14
3.1.1. Metodología del diseño	14
3.1.2. Lista de exigencias.....	15
3.1.3. Estructura de funciones	17
4. CAPÍTULO IV: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN.....	18
4.1. Identificación de requerimientos	18
4.1.1. Matriz morfológica	18
4.1.2. Requerimientos Funcionales y No Funcionales	19
4.2. Análisis de la solución.....	20
4.2.1. API para Clima (WeatherAPI)	21

4.2.2.	API para el Chatbot (Mistral-7B).....	21
4.2.3.	Base de datos MySQL	21
4.3.	Diseño	22
4.3.1.	Diseño de Interfaz	22
4.3.2.	Programación	24
4.3.2.1.	Arquitectura general.....	24
4.3.2.2.	Componentes clave del desarrollo.....	24
4.3.2.3.	Códificación clave.....	26
5.	CAPÍTULO V: CONSTRUCCIÓN	31
5.1.	Construcción	31
5.1.1.	Estructuración del chatbot	31
5.1.2.	Desarrollo del chatbot	31
5.1.3.	Prueba del chatbot.....	32
5.1.4.	Capacitación y difusión del chatbot	32
5.1.5.	Costos.....	32
5.2.	Pruebas y resultados	35
	CONCLUSIONES.....	36
	REFERENCIAS BIBLIOGRÁFICAS	37

LISTA DE FIGURAS

Figura 1. Impact Mapping del chatbot.....	14
Figura 2. Pantalla de Inicio	22
Figura 3. Panel de Usuario	23
Figura 4. Ventana del Chatbot	23
Figura 5. Sección de Material Didáctico	23
Figura 6. Panel de Administración.....	24
Figura 7. Resumen general de preguntas	36

LISTA DE TABLAS

Tabla 1. Matriz Morfológica	19
Tabla 2. Desglose de Requerimientos	20
Tabla 3. Selección de requerimientos	20
Tabla 4. Estimación de costos.....	35
Tabla 5. Reserva de Gestión	35

RESUMEN

La presente monografía desarrolla e implementa un **chatbot agrícola inteligente** como herramienta de apoyo técnico para agricultores del distrito de Hualhuas, con el objetivo de mejorar la toma de decisiones durante el proceso de cultivo de maíz. Esta propuesta surge como respuesta a las limitaciones detectadas en el contexto local: escasa asesoría técnica especializada, deficiente gestión del recurso hídrico y baja incorporación de tecnologías digitales en la actividad agrícola.

El sistema propuesto ofrece una plataforma accesible desde dispositivos móviles y computadoras, permitiendo a los agricultores realizar consultas en lenguaje natural sobre temas clave como siembra, fertilización, control de plagas, manejo del agua y condiciones climáticas. El chatbot está basado en un modelo de lenguaje natural (Mistral-7B) que, al integrar información agronómica validada y datos climáticos en tiempo real mediante APIs, proporciona respuestas precisas y adaptadas al entorno específico del agricultor.

El desarrollo técnico fue realizado bajo el marco metodológico SCRUM, utilizando Python, Flask, MySQL y servicios de terceros para la integración del clima y el motor conversacional. Se implementaron funcionalidades como panel de control, administración de usuarios, gestión de contenidos didácticos, seguridad en el almacenamiento de datos y seguimiento de interacciones.

Durante la fase de pruebas, el chatbot alcanzó una tasa de éxito del 84% en la respuesta a consultas reales, lo que demuestra su potencial como herramienta de transformación para el sector agrícola local. Asimismo, se diseñaron materiales de capacitación (guías, videos y tutoriales) para facilitar su adopción en comunidades rurales.

Los resultados obtenidos evidencian que el uso de inteligencia artificial aplicada al agro puede empoderar a los agricultores, optimizar los recursos, reducir la incertidumbre y fomentar una agricultura más eficiente, sostenible e inclusiva. Este proyecto representa un modelo replicable en otras zonas rurales con características similares, promoviendo la digitalización del agro en el Perú.

INTRODUCCIÓN

El sector agrícola en el Perú representa una de las principales actividades económicas y sociales, especialmente en zonas rurales como el distrito de Hualhuas, ubicado en la región central andina. A pesar de su importancia, los agricultores locales, en su mayoría pequeños productores, enfrentan múltiples limitaciones que obstaculizan la mejora de su productividad y sostenibilidad. Entre estas limitaciones destacan la escasa asesoría técnica, el uso inefficiente del recurso hídrico, la dependencia de prácticas tradicionales poco efectivas y, sobre todo, la reducida adopción de tecnologías digitales para la gestión agrícola. Esta situación, agravada por los efectos del cambio climático, repercute directamente en la eficiencia de los procesos de siembra y cosecha, generando pérdidas evitables y bajos rendimientos.

En este contexto, la presente monografía tiene como finalidad exponer el diseño, desarrollo e implementación de un **chatbot agrícola inteligente**, orientado a brindar asesoría técnica en tiempo real a los agricultores de Hualhuas, específicamente en el cultivo de maíz. Esta propuesta tecnológica busca democratizar el acceso a información agronómica confiable mediante una plataforma digital accesible, que integra inteligencia artificial, procesamiento de lenguaje natural, datos meteorológicos y contenidos educativos.

El chatbot desarrollado no solo responde consultas específicas del agricultor, sino que también considera variables como el tipo de suelo, método de riego, clima actual y fecha estimada de cosecha, adaptando sus respuestas al contexto particular de cada usuario. Asimismo, se implementaron módulos de formación técnica que permiten a los agricultores mejorar su alfabetización digital y adoptar prácticas más sostenibles.

Este documento describe el proceso integral del proyecto, desde la identificación del problema y la formulación de objetivos, hasta el análisis de requerimientos, el diseño funcional, la construcción técnica, las pruebas de validación y el análisis de resultados. El trabajo se fundamenta en metodologías ágiles de desarrollo y evidencia cómo la innovación tecnológica puede convertirse en una herramienta clave para transformar el agro peruano, promover la inclusión digital y contribuir a la seguridad alimentaria en comunidades vulnerables.

1. CAPÍTULO I: PLANTEAMIENTO DEL ESTUDIO

1.1. Planteamiento y formulación del problema

1.1.1. Enunciado del problema

En el distrito de Hualhuas, los agricultores dedicados al cultivo de maíz enfrentan múltiples limitaciones que afectan directamente la eficiencia y rentabilidad de su producción. Entre los principales problemas destacan la falta de asesoría técnica adecuada, el desconocimiento en el manejo eficiente del recurso hídrico y la carencia de información climática confiable y oportuna. Estas deficiencias provocan que las decisiones agrícolas se basen en prácticas tradicionales que no responden eficazmente a los desafíos actuales del cambio climático. Como consecuencia, se generan bajas productividades, pérdidas que podrían evitarse y conflictos entre agricultores por el uso inadecuado de los canales de riego. A esto se suma la escasa penetración tecnológica en el ámbito agrícola y la baja alfabetización digital de muchos productores, lo que dificulta la incorporación de herramientas digitales innovadoras que podrían optimizar sus procesos y resultados.

1.1.2. Formulación del problema

¿Cómo podría un chatbot agrícola contribuir a mejorar la toma de decisiones en los procesos de siembra y cosecha de maíz en el distrito de Hualhuas, frente a la falta de asesoría técnica, el manejo ineficiente del recurso hídrico y la limitada adopción tecnológica de los agricultores?

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un chatbot agrícola inteligente que brinde asesoría técnica personalizada y en tiempo real a los agricultores del distrito de Hualhuas para optimizar las prácticas de siembra, riego, fertilización, control de plagas y cosecha del maíz, integrando información climática local para mejorar la productividad y sostenibilidad del cultivo.

1.2.2. Objetivos específicos

- Diseñar y desarrollar una plataforma digital de chatbot accesible desde dispositivos móviles y web.

- Integrar una base de datos agronómica validada y un módulo de datos climáticos en tiempo real.
- Capacitar a un mínimo de 40 agricultores del distrito en el uso del chatbot.
- Implementar un piloto de prueba y evaluar la usabilidad, satisfacción y efectividad del chatbot en campo.
- Generar recomendaciones y ajustes basados en la retroalimentación de los usuarios para mejorar la herramienta.

1.3. Justificación e importancia

Este proyecto tiene una gran relevancia para la comunidad agrícola de Hualhuas, ya que busca modernizar y optimizar las prácticas agrícolas en una zona afectada por el cambio climático y limitaciones tecnológicas. Al proporcionar asesoría técnica confiable y personalizada, el chatbot empodera a los agricultores, reduce la dependencia de conocimientos empíricos y mejora la eficiencia en el uso de recursos, especialmente el agua. Además, contribuye a la seguridad alimentaria local y a la sostenibilidad ambiental mediante la promoción de prácticas agrícolas más responsables. Desde una perspectiva social, el proyecto impulsa la inclusión digital y fomenta la innovación en zonas rurales, creando un modelo replicable para otras regiones con problemáticas similares.

2. CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes del problema

A1. En el distrito de Hualhuas, los agricultores dedicados principalmente al cultivo de maíz enfrentan múltiples desafíos, como el manejo ineficiente del recurso hídrico, la carencia de asesoría técnica especializada y la escasa adopción de tecnologías digitales para la gestión agrícola. Esta situación se ve agravada por el cambio climático, que modifica los patrones meteorológicos, afectando directamente la productividad de los cultivos (Villanueva et al., 2022).

A2. A nivel internacional, la incorporación de herramientas digitales en la agricultura, especialmente los chatbots, ha demostrado ser una estrategia efectiva para apoyar a pequeños agricultores. Por ejemplo, en la India se ha utilizado el chatbot Plantix, que permite identificar enfermedades en los cultivos mediante el envío de fotos y ofrece tratamientos sugeridos (Sharma et al., 2019). Asimismo, en países como Tailandia y Filipinas, los asistentes virtuales agrícolas permiten consultar información climática, precios de mercado y técnicas de cultivo (Kumar et al., 2021).

A3. En Perú, diversas investigaciones sugieren la necesidad de promover la agricultura inteligente y la transformación digital del agro. Según la FAO (2021), la digitalización del sector agrícola puede mejorar la sostenibilidad, la productividad y la resiliencia de las comunidades rurales. Sin embargo, la adopción de estas tecnologías enfrenta barreras asociadas a la conectividad, el bajo nivel de alfabetización digital y la falta de contenidos en lengua local o adaptados al contexto específico de los agricultores.

2.2. Bases teóricas

2.2.1. Inteligencia Artificial (IA)

La inteligencia artificial es un área de la informática que busca desarrollar sistemas capaces de simular la inteligencia humana, incluyendo habilidades como el aprendizaje, la percepción, el razonamiento y la comprensión del lenguaje natural (Russell & Norvig, 2021). Las aplicaciones de la IA en la agricultura abarcan desde la predicción de cosechas hasta la automatización de procesos mediante drones, sensores y asistentes virtuales.

2.2.2. Procesamiento de Lenguaje Natural (PLN)

El Procesamiento de Lenguaje Natural (PLN) es una rama de la inteligencia artificial que permite a las máquinas comprender, interpretar y generar lenguaje humano. El PLN permite a los chatbots procesar preguntas en lenguaje común y generar respuestas coherentes y contextualizadas (Jurafsky & Martin, 2021).

Los modelos más avanzados de PLN utilizan redes neuronales profundas, como los modelos GPT (Generative Pre-trained Transformer), que potencian la capacidad conversacional de los chatbots. Estos modelos son entrenados con grandes cantidades de datos para comprender el lenguaje de manera contextual y adaptativa.

2.2.3. Chatbots: Definición y Tipología

Un chatbot es un programa informático diseñado para interactuar con los usuarios a través de texto o voz, simulando una conversación humana. De acuerdo con Pérez-Marín & Pascual-Nieto (2011), los chatbots pueden clasificarse en:

Basados en reglas: Funcionan mediante flujos predefinidos y respuestas fijas.

Basados en inteligencia artificial: Utilizan algoritmos de aprendizaje automático que permiten adaptar las respuestas según el contexto y la conversación previa.

En el contexto agrícola, los chatbots basados en IA son preferibles debido a su capacidad para adaptarse a preguntas imprevistas y brindar asesoría técnica diversa sobre riego, fertilización, plagas y condiciones climáticas.

2.2.4. Agricultura Inteligente y Chatbots

La agricultura inteligente es el conjunto de tecnologías que buscan optimizar el uso de recursos, mejorar la productividad y minimizar el impacto ambiental mediante el uso de sensores, big data, inteligencia artificial y sistemas de información geográfica. Dentro de esta tendencia, los chatbots representan una solución accesible y de bajo costo para que los agricultores accedan a información relevante en tiempo real (Wolfert et al., 2017).

En Perú, iniciativas como E-Agro han demostrado el potencial de combinar chatbots con IoT para la agricultura. El chatbot agrícola puede integrarse con

bases de datos locales de clima, suelos y enfermedades para brindar recomendaciones más precisas y adaptadas a la realidad del agricultor peruano.

2.3. Definición de términos básicos

Para una mejor comprensión del presente estudio, se definen a continuación los términos clave relacionados con el desarrollo del chatbot agrícola.

El chatbot es un programa informático diseñado para simular una conversación con un usuario humano a través de texto o voz. Su principal función es interactuar con las personas proporcionando información, respuestas o asistencia automatizada, lo que permite resolver dudas o guiar procesos específicos sin la necesidad de intervención humana directa.

La inteligencia artificial (IA) se refiere a la rama de la informática que estudia y desarrolla sistemas capaces de realizar tareas que, si fueran hechas por humanos, requerirían inteligencia. Entre estas habilidades se incluyen el razonamiento, la planificación, el aprendizaje, la comprensión del lenguaje y la percepción visual. La IA permite a las máquinas mejorar su desempeño a partir de la experiencia, haciendo posible que los sistemas sean cada vez más precisos y eficientes.

El procesamiento de lenguaje natural (PLN) es una subdisciplina de la inteligencia artificial que tiene como propósito la interacción entre computadoras y el lenguaje humano, permitiendo que los sistemas comprendan, interpreten y generen lenguaje de forma similar a cómo lo hacen las personas. Gracias al PLN, los chatbots pueden entender preguntas expresadas en lenguaje cotidiano y responder de manera coherente y contextualizada.

Por otro lado, la agricultura inteligente hace referencia al uso de tecnologías avanzadas, como sensores, big data, inteligencia artificial, sistemas de información geográfica e Internet de las cosas (IoT), para optimizar la producción agrícola. Esta modalidad busca hacer más eficiente el uso de los recursos naturales, incrementar la productividad y garantizar la sostenibilidad ambiental.

El machine learning o aprendizaje automático es una técnica dentro de la inteligencia artificial que permite a los sistemas aprender de los datos y mejorar su desempeño sin ser programados explícitamente para cada tarea. Una evolución de esta técnica es el deep learning o aprendizaje profundo, que utiliza redes neuronales artificiales de múltiples capas para procesar grandes volúmenes de datos y resolver problemas

complejos, como el reconocimiento de imágenes o la comprensión avanzada del lenguaje.

Estas definiciones permiten contextualizar el desarrollo de la solución tecnológica planteada en el proyecto, orientada a asistir a los agricultores de Hualhuas en sus procesos de siembra y cosecha mediante el uso de un chatbot especializado.

3. CAPÍTULO III: METODOLOGÍA

3.1. Diseño

3.1.1. Metodología del diseño

Para el desarrollo del chatbot agrícola inteligente destinado a los agricultores del distrito de Hualhuas, se optó por aplicar la metodología ágil SCRUM como marco principal de trabajo. Esta decisión se fundamenta en la necesidad de responder de manera flexible a los requerimientos del entorno agrícola, caracterizado por su complejidad, dinamismo y diversidad de actores.

SCRUM es una metodología ágil que permite gestionar proyectos tecnológicos complejos de forma iterativa e incremental, facilitando la entrega continua de funcionalidades útiles y permitiendo incorporar retroalimentación constante de los usuarios finales. Esto resulta especialmente útil en contextos como el agrícola rural, donde los procesos requieren validación directa en campo y ajustes frecuentes en base a la experiencia del usuario.

Como etapa inicial dentro de esta metodología, se utilizó la herramienta de Impact Mapping para alinear estratégicamente el propósito del proyecto con las funcionalidades técnicas que debía cumplir el chatbot. Esta técnica ayudó al equipo a visualizar de forma estructurada el impacto esperado del sistema, identificar a los actores involucrados y definir las acciones necesarias para alcanzar los objetivos generales del proyecto.

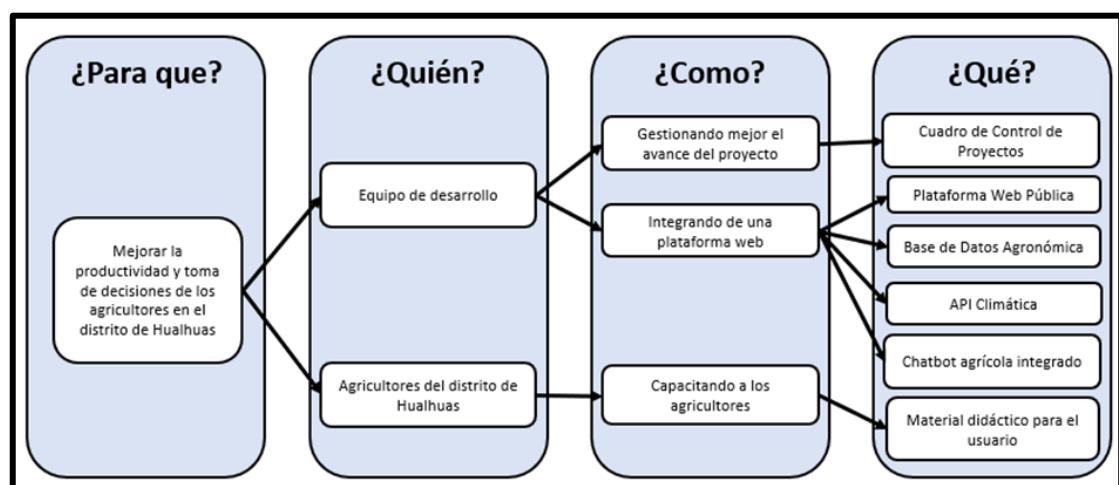


Figura 1. Impact Mapping del chatbot

3.1.2. Lista de exigencias

Con base en el propósito general del proyecto, que es mejorar la productividad y la toma de decisiones agrícolas en el distrito de Hualhuas, se establecieron una serie de exigencias estratégicas, operativas y técnicas que deben guiar el desarrollo del sistema. Estas exigencias han sido identificadas a partir del análisis realizado mediante la herramienta Impact Mapping, en el cual se definieron claramente los actores involucrados, los medios de intervención y los productos esperados.

Las exigencias aquí descritas constituyen las condiciones mínimas necesarias para que el chatbot agrícola integrado cumpla su función de forma eficiente, confiable y accesible para los agricultores del entorno rural.

a) Exigencias Estratégicas

Estas condiciones responden al impacto deseado del proyecto y a su alineación con las necesidades de los usuarios finales:

- Promover la toma de decisiones agrícolas informadas a través del acceso oportuno a información técnica confiable.
- Fomentar la inclusión digital y la adopción de nuevas tecnologías en comunidades rurales con bajo nivel de alfabetización digital.
- Empoderar a los agricultores mediante una herramienta interactiva que brinde asesoría técnica sin intermediarios.

b) Exigencias Operativas

Se refieren a aspectos clave que deben considerarse para el funcionamiento general del sistema en su entorno real de uso:

- El sistema debe ser accesible desde dispositivos móviles y computadoras, sin requerir instalaciones adicionales.
- El acceso debe ser posible con conectividad limitada, asegurando funcionalidades básicas aun en zonas rurales.
- La plataforma debe estar diseñada para ser comprendida por usuarios con experiencia tecnológica básica o nula.

- Debe incorporarse una estrategia de capacitación progresiva dirigida a los agricultores del distrito.

c) Exigencias Técnicas Generales

Estas exigencias establecen parámetros técnicos esenciales para la viabilidad del sistema y su rendimiento:

- Integración de un motor conversacional basado en lenguaje natural (chatbot) que permita la interacción fluida y contextual.
- Incorporación de información climática local en tiempo real mediante servicios externos de datos meteorológicos.
- Consolidación de una base de conocimientos agronómicos adaptada al cultivo de maíz y validada por fuentes confiables.
- Implementación de herramientas de monitoreo y control del proyecto que faciliten su gestión, evaluación y mejora continua.
- Disponibilidad de contenidos educativos (videos, guías, tutoriales) dentro de la plataforma, orientados al aprendizaje autónomo del usuario.

d) Exigencias de Calidad del Sistema

Finalmente, el sistema deberá garantizar niveles mínimos de calidad que aseguren una experiencia de uso satisfactoria:

- **Usabilidad:** La interfaz debe ser clara, intuitiva y accesible para todo tipo de usuario.
- **Seguridad:** Debe proteger los datos personales de los usuarios y garantizar la privacidad de las interacciones.
- **Eficiencia:** Las respuestas del chatbot deben generarse en menos de tres segundos por consulta.
- **Disponibilidad:** El sistema debe estar disponible las 24 horas del día y funcionar de forma estable en condiciones diversas.

3.1.3. Estructura de funciones

La estructura funcional del sistema fue definida a partir de la visualización estratégica proporcionada por el Impact Mapping, con el fin de alinear los esfuerzos técnicos con el propósito central del proyecto: mejorar la productividad y la toma de decisiones de los agricultores del distrito de Hualhuas. Este enfoque permitió identificar los elementos críticos que debe contener el sistema, así como la relación directa entre los actores involucrados, las acciones necesarias y los componentes que deben construirse para alcanzar el impacto esperado.

A continuación, se describen los bloques funcionales principales del sistema:

1. Plataforma Web Pública

Es el canal principal de acceso para los agricultores. Su función es proporcionar una interfaz amigable y comprensible que permita interactuar con el sistema desde cualquier dispositivo. A través de la plataforma, los usuarios podrán hacer consultas al chatbot, acceder a material educativo y gestionar su perfil. Esta plataforma será clave para garantizar la accesibilidad y facilitar el uso del sistema por parte de personas con poca experiencia tecnológica.

2. Chatbot Agrícola Integrado

Es el núcleo interactivo del sistema. Este componente permitirá a los agricultores hacer preguntas en lenguaje natural y recibir respuestas automáticas, contextualizadas y pertinentes sobre temas como siembra, riego, fertilización, plagas y clima. El chatbot será entrenado con contenido multiformato (documentos técnicos, videos, recursos audiovisuales) y estará conectado a la base de conocimientos agronómicos y climáticos del sistema.

3. Base de Datos Agronómica

Este módulo contiene la información técnica especializada necesaria para brindar asesoría confiable. Incluye conocimientos validados sobre el cultivo de maíz, como etapas de desarrollo, técnicas de manejo, prácticas sostenibles, enfermedades comunes y recomendaciones de fertilización. Además, almacenará los datos de los usuarios y el historial de sus interacciones con el sistema.

4. Integración Climática (API)

Se trata de un componente que enlaza el sistema con fuentes externas de datos meteorológicos. Esta función es fundamental para brindar recomendaciones adaptadas a las condiciones reales del clima en la zona de Hualhuas. Gracias a esta integración, el sistema podrá emitir alertas, sugerencias de riego o advertencias de fenómenos como heladas o lluvias intensas.

5. Cuadro de Control de Proyectos

Este componente será utilizado por el equipo de desarrollo para gestionar, monitorear y dar seguimiento al avance del proyecto. Permitirá visualizar tareas, cronogramas, hitos alcanzados y ajustes necesarios. Además, facilitará la coordinación entre los distintos roles del equipo SCRUM y la documentación de los sprints.

6. Módulo de Material Didáctico

Este bloque funcional está orientado a la capacitación y formación de los usuarios. Contendrá recursos educativos como tutoriales, guías prácticas y videos explicativos diseñados para fortalecer la alfabetización digital y fomentar la adopción del sistema. Todo el material estará disponible desde la plataforma web, en un formato comprensible y adaptado al contexto rural.

4. CAPÍTULO IV: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

4.1. Identificación de requerimientos

Para el desarrollo del chatbot agrícola que asistirá a los agricultores del distrito de Hualhuas, se han identificado una serie de requerimientos funcionales y no funcionales que deben cumplirse para garantizar el éxito del proyecto. Estos requerimientos se agrupan en diferentes áreas clave:

4.1.1. Matriz morfológica

A continuación, se presentan algunas de las principales características y opciones para el diseño del chatbot:

Característica	Opción A	Opción B	Opción C
Acceso al sistema	Plataforma web	Plataforma móvil	Híbrido (Web y Móvil)
Interfaz del usuario	Simple, accesible para novatos	Interactiva y amigable	Completa y detallada
Tecnología de backend	Python con Flask	Python con Flask	Python con Flask
Integración con APIs	API clima gratuita (WeatherAPI)	API clima gratuita (WeatherAPI)	API de pago para clima y chatbot (combined)
Base de datos	MySQL	MySQL	MySQL
Respuesta del Chatbot	IA (Mistral 7B-Instruct)	IA (Mistral 7B-Instruct)	IA (Mistral 7B-Instruct)

Tabla 1. Matriz Morfológica

4.1.2. Requerimientos Funcionales y No Funcionales

Entregables	Requerimientos específicos
Cuadro de control de proyectos	-Diseñar Interfaz de Cuadro de Control del proyecto -Diseñar tablas y relaciones del proyecto -Gestionar Información del proyecto
Plataforma Web	- Diseñar interfaz web accesible. -Creación de registro de usuario - Adaptación para dispositivos móviles y computadoras
Base de Datos Agronómica	-Diseñar tablas para datos de usuarios y administradores - Gestionar almacenamiento de datos del agricultor.
API Climática	- Integrar servicio externo de datos meteorológicos.
Chatbot agrícola	-Entrenamiento mediante instrucciones a un modelo ya trabajado en la nube -Diseñar flujos conversacionales para consultas agronómicas. - Registrar y almacenar interacciones de usuarios. - Integrar con plataforma web
Material didáctico para el usuario	- Diseñar y elaborar guías, tutoriales y videos. - Gestionar acceso a material desde la plataforma.

Tabla 2. Desglose de Requerimientos

Tipo	Requerimientos
<i>De Información</i>	<ul style="list-style-type: none"> - <i>Información agronómica validada para el cultivo de maíz.</i> - <i>Datos meteorológicos locales en tiempo real.</i> - <i>Historial de interacciones de usuarios.</i> - <i>Material didáctico y guías de uso.</i>
<i>Funcionales</i>	<ul style="list-style-type: none"> - <i>Diseñar Interfaz de Cuadro de Control del proyecto</i> - <i>Diseñar tablas y relaciones del proyecto</i> - <i>Gestionar Información del proyecto</i> - <i>Diseñar interfaz web accesible.</i> - <i>Creación de registro de usuario</i> - <i>Adaptación para dispositivos móviles y computadoras</i> - <i>Diseñar tablas para datos de usuarios y administradores</i> - <i>Gestionar almacenamiento de datos del agricultor.</i> - <i>Integrar servicio externo de datos meteorológicos.</i> - <i>Diseñar flujos conversacionales para consultas agronómicas.</i> - <i>Registrar y almacenar interacciones de usuarios.</i> - <i>Integrar con plataforma web</i> - <i>Diseñar y elaborar guías, tutoriales y videos.</i> - <i>Gestionar acceso a material desde la plataforma.</i>
<i>No Funcionales</i>	<ul style="list-style-type: none"> - <i>Eficiencia</i> - <i>Seguridad</i> - <i>Usabilidad</i> - <i>Disponibilidad</i>

Tabla 3. Selección de requerimientos

4.2. Análisis de la solución

En base a los requerimientos anteriores, la solución propuesta es un chatbot agrícola integrado con dos APIs: una gratuita para el clima y otra paga para las consultas del

chatbot. A continuación, se describe el análisis de las soluciones más adecuadas para abordar los problemas identificados en el planteamiento del estudio.

4.2.1. API para Clima (WeatherAPI)

La integración de la API de WeatherAPI permite obtener información climática en tiempo real para los agricultores. Con esta API, el sistema podrá proporcionar datos sobre temperatura, condiciones climáticas y ubicación, lo que ayudará a los agricultores a tomar decisiones informadas sobre el riego, la siembra y la cosecha, considerando las siguientes justificaciones:

- Gratis y fácil de implementar: La API gratuita cubre las necesidades básicas de pronóstico del clima y es fácilmente integrable en el sistema.
- Relevancia local: Al ofrecer datos específicos sobre la localidad, los agricultores pueden tomar decisiones adaptadas a su entorno inmediato, lo que mejora la precisión de las recomendaciones.

4.2.2. API para el Chatbot (Mistral-7B)

El uso de la API de Mistral-7B permite realizar consultas avanzadas sobre el cultivo de maíz, adaptadas a la realidad local. Este modelo de lenguaje utiliza el procesamiento de lenguaje natural (PLN) para interpretar las preguntas de los agricultores y proporcionar respuestas relevantes, teniendo en cuenta los datos proporcionados por el agricultor, como el tipo de suelo, el tipo de maíz y las condiciones climáticas, considerando las siguientes justificaciones:

- Alta capacidad de respuesta: El modelo Mistral-7B es avanzado y permite respuestas más complejas, adaptadas al contexto específico del agricultor, lo que resulta en un asesoramiento técnico más preciso.
- Costo razonable: Con un costo de \$2 por cada 80 respuestas, el costo de la API es razonable para las necesidades del proyecto, ofreciendo un balance entre costo y funcionalidad.

4.2.3. Base de datos MySQL

La base de datos en MySQL se utilizará para almacenar toda la información relevante sobre los agricultores y sus interacciones con el chatbot. Esto incluye datos como el tipo de suelo, el tamaño del terreno, el tipo de maíz sembrado, la fecha de cosecha y las interacciones previas con el sistema, considerando las siguientes justificaciones:

- Escalabilidad: MySQL permite escalar el sistema conforme crezca la base de usuarios sin perder rendimiento.
- Seguridad: Al almacenar las contraseñas de los usuarios de manera segura (utilizando script para el hashing), la base de datos garantiza la protección de la información sensible.

4.3. Diseño

El diseño del sistema se ha enfocado en la creación de una solución digital integrada, accesible y funcional para los agricultores del distrito de Hualhuas. A continuación, se detallan los aspectos clave del diseño.

4.3.1. Diseño de Interfaz

Se consideró la estructura lógica de la interfaz de usuario, la disposición de los elementos gráficos y la navegabilidad del sistema como parte del diseño estructural de la aplicación.

Estructura visual de la interfaz (Mockup básico):

Pantalla de Inicio: Logo del proyecto, botones de acceso/registro.



Figura 2. Pantalla de Inicio

Panel de Usuario: Acceso a funciones principales de customización para el usuario

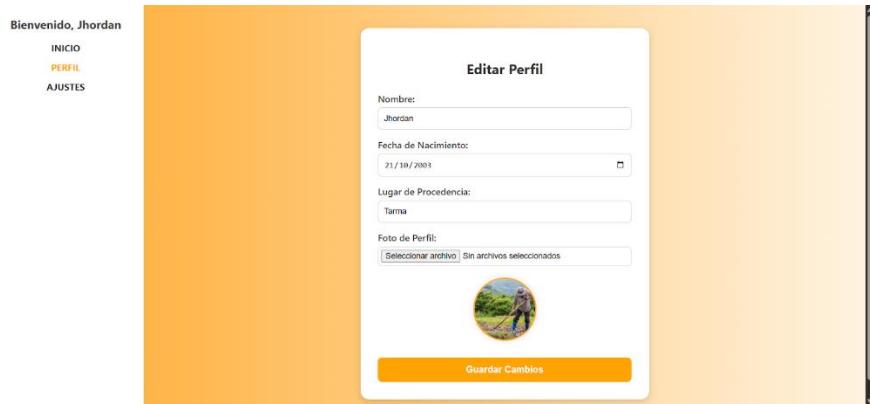


Figura 3. Panel de Usuario

Ventana del Chatbot: Área conversacional con el asistente virtual y acceso a herramientas rápidas como el pronóstico del clima y datos sobre el terreno del agricultor

Figura 4. Ventana del Chatbot

Sección de Material Didáctico: Guías visuales, videos, PDFs descargables.

Figura 5. Sección de Material Didáctico

Panel de Administración (para expertos agrónomos): Visualización de interacciones, gestión de contenidos.



Figura 6. Panel de Administración

4.3.2. Programación

La programación del sistema se desarrolló utilizando el lenguaje Python, apoyado en el microframework Flask, que permite una gestión ágil y modular de rutas, formularios y vistas para aplicaciones web. A continuación, se describen los principales aspectos del desarrollo del sistema.

4.3.2.1. Arquitectura general

El sistema sigue un enfoque cliente-servidor. El servidor ejecuta una aplicación Flask que gestiona:

- Autenticación y control de usuarios (administradores y agricultores).
- Conexión con la base de datos MySQL para almacenamiento estructurado.
- Consulta a dos APIs externas: una para obtener datos climáticos y otra para responder consultas agronómicas mediante un modelo de lenguaje.
- Renderización de interfaces web utilizando plantillas HTML para distintos perfiles de usuario.

4.3.2.2. Componentes clave del desarrollo

- Autenticación de usuarios y roles ya que Los usuarios pueden registrarse como “agricultores” o “administradores”, además se

emplea `bcrypt` para el hashing seguro de contraseñas y en caso de los administradores requieren una clave adicional para su validación, y la sesión de usuario se maneja con `Flask-Session` para mantener la seguridad y persistencia.

- Base de datos MySQL que se conecta mediante `mysql.connector` cuyas tablas principales son usuarios (Id, Username, Password, Rol, Datos personales, Foto); Datos Usuario(tipo de suelo, tamaño de terreno, tipo de maíz, riego, fecha de cosecha); PreguntasRespuestas(pregunta, respuesta, fecha, user_id); ModulosInformativos(material didáctico administrado por expertos)
- Integración con WeatherAPI para obtener datos en tiempo real sobre la temperatura actual, condición del clima (texto e ícono), Ubicación geográfica(ciudad, país) y esto permite contextualizar la recomendación agrícola del chatbot
- Integración con modelo de lenguaje (Mistral-7B) , se emplea la API de OpenRouter para enviar prompts personalizados y recibir respuestas generadas por el modelo mistralai/mistral-7b-instruct, El prompt incluye datos específicos del agricultor (tipo de suelo, método de riego, clima actual, etc.) y esto permite una respuesta personalizada y técnica, emulando la asesoría de un agrónomo real
- panel del agricultor, ya que luego de iniciar sesión el agricultor puede Ver el clima de su localidad, Ingresar o actualizar datos sobre su terreno y cultivo, Consultar al chatbot mediante una interfaz de texto, Visualizar su historial de preguntas y respuestas, Acceder a materiales educativos sobre buenas prácticas agrícolas
- En el panel del administrador, los administradores pueden ver estadísticas generales de uso del chatbot, visualizar preguntas más frecuentes o sin respuesta válida, agregar, modificar o eliminar módulos informativos, cambiar contraseñas o eliminar usuarios.
- Módulo de materiales informativos, en este módulo se puede ofrecer al agricultor: Guías descargables en PDF, videos o enlaces útiles, recomendaciones de temporada y además los administradores los gestionan desde su panel.

4.3.2.3. Códificación clave

```
from flask import Flask, render_template, request, redirect, url_for, session
import mysql.connector
import bcrypt
import requests
import os
app = Flask(__name__)
app.secret_key = 'clave_secreta_fis'
API_HF_URL = "https://openrouter.ai/api/v1/chat/completions"
API_HF_TOKEN = "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"
HEADERS_HF = {
    "Authorization": f"Bearer {API_HF_TOKEN}",
    "Content-Type": "application/json"
}
db_config = {
    'host': 'localhost',
    'user': 'root',
    'password': '-Jhordan21',
    'database': 'LoginDB'
}

def obtener_clima(ciudad):
    api_key = "4bb0d516246647c1818152450250306"
    url = f"http://api.weatherapi.com/v1/current.json?key={api_key}&q={ciudad}&lang=es"
    try:
        response = requests.get(url, timeout=10)
        data = response.json()
        return {
            'temp': data['current']['temp_c'],
            'condicion': data['current']['condition']['text'],
            'icono': data['current']['condition']['icon'],
            'ciudad': data['location']['name'],
            'pais': data['location']['country']
        }
    except:
        return {
            'temp': 'N/A',
            'condicion': 'Error al obtener clima',
            'icono': '',
            'ciudad': ciudad,
            'pais': ''
        }

def consultar_chatbot(pregunta, ciudad, clima, datos_usuario):
    tipo_suelo = datos_usuario.get('tipo_suelo', 'desconocido')
    tipo_maiz = datos_usuario.get('tipo_maiz', 'desconocido')
    fecha_cosecha = datos_usuario.get('fecha_cosecha', 'no registrada')
    tamano = datos_usuario.get('tamano_terreno', 'desconocido')
    riego = datos_usuario.get('riego_usado', 'desconocido')
    app.route('/registro', methods=['GET', 'POST'])
```

```

def registro():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        rol = request.form['rol']
        admin_key = request.form.get('admin_key', "")
        if rol == 'admin' and admin_key != '123456789':
            return redirect(url_for('fracaso'))

        hashed_pw = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
        try:
            conn = mysql.connector.connect(**db_config)
            cursor = conn.cursor()
            cursor.execute("INSERT INTO Usuarios (Username, Password, Rol) VALUES (%s, %s, %s)", (username, hashed_pw, rol))
            conn.commit()
            session['username'] = username
            cursor.execute("SELECT Id FROM Usuarios WHERE Username = %s", (username,))
            session['user_id'] = cursor.fetchone()[0]
            conn.close()

            if rol == 'admin':
                conn = mysql.connector.connect(**db_config)
                cursor = conn.cursor()
                cursor.execute("SELECT Username FROM Usuarios")
                usuarios = cursor.fetchall()
                conn.close()
                return render_template('admin.html', username=username, usuarios=usuarios)
            return redirect(url_for('inicio_registrado'))
        except mysql.connector.Error as e:
            return f"Error al registrar: {str(e)}"
        return render_template('registro.html')
@app.route('/inicio_registrado', methods=['GET', 'POST'])
def inicio_registrado():
    if 'username' not in session:
        return redirect(url_for('fracaso'))

    ciudad = request.form.get("ciudad", "Hualhuas")
    clima = obtener_clima(ciudad)
    condicion = clima['condicion'].lower()
    if "lluvia" in condicion:
        recomendacion = "Evita fertilizar hoy, el agua puede arrastrar los nutrientes. Revisa el drenaje del campo."
    elif "soleado" in condicion or "despejado" in condicion:
        recomendacion = "Buen día para fertilizar y revisar presencia de plagas en las hojas del maíz."
    elif "nublado" in condicion:
        recomendacion = "Condiciones estables, aprovecha para monitorear el crecimiento del cultivo."
    else:
        recomendacion = "Monitorea el clima antes de realizar labores intensas en el cultivo."

    user_id = session.get('user_id')
    datos_usuario = {}
    pregunta = ""
    respuesta = ""

```

```

preguntas_respuestas = []
foto_url = None

try:
    conn = mysql.connector.connect(**db_config)
    cursor = conn.cursor(dictionary=True)

    # [✓] Obtener foto de perfil del usuario
    cursor.execute("SELECT Foto FROM Usuarios WHERE Id = %s", (user_id,))
    result_foto = cursor.fetchone()
    foto_url = result_foto['Foto'] if result_foto and result_foto['Foto'] else None

    # [✓] Guardar o actualizar datos del campo si se envían
    if request.method == 'POST' and 'guardar_datos' in request.form:
        datos = {
            'tamano': request.form['tamano_terreno'],
            'fecha': request.form['fecha_cosecha'],
            'suelo': request.form['tipo_suelo'],
            'maiz': request.form['tipo_maiz'],
            'riego': request.form['riego_usado']
        }
        cursor.execute("SELECT * FROM DatosUsuario WHERE user_id = %s", (user_id,))
        if cursor.fetchone():
            cursor.execute("""
                UPDATE DatosUsuario
                SET tamano_terreno=%s, fecha_cosecha=%s, tipo_suelo=%s, tipo_maiz=%s,
                riego_usado=%s
                WHERE user_id = %s
            """, (datos['tamano'], datos['fecha'], datos['suelo'], datos['maiz'], datos['riego'], user_id))
        else:
            cursor.execute("""
                INSERT INTO DatosUsuario (user_id, tamano_terreno, fecha_cosecha, tipo_suelo,
                tipo_maiz, riego_usado)
                VALUES (%s, %s, %s, %s, %s)
            """, (user_id, datos['tamano'], datos['fecha'], datos['suelo'], datos['maiz'], datos['riego']))
        conn.commit()

    # [✓] Obtener datos del usuario
    cursor.execute("SELECT * FROM DatosUsuario WHERE user_id = %s", (user_id,))
    datos_usuario = cursor.fetchone()

    # [✓] Procesar pregunta del usuario si la envió
    pregunta = request.form.get('mensaje', '').strip()
    if pregunta:
        respuesta = consultar_chatbot(pregunta, ciudad, clima, datos_usuario)
        cursor.execute("""
            INSERT INTO PreguntasRespuestas (user_id, pregunta, respuesta)
            VALUES (%s, %s, %s)
        """, (user_id, pregunta, respuesta))
        conn.commit()

    # [✓] Recuperar historial de preguntas
    cursor.execute(""""

```

```

        SELECT pregunta, respuesta, fecha
        FROM PreguntasRespuestas
        WHERE user_id = %s
        ORDER BY fecha DESC
        LIMIT 10
    """", (user_id,))
preguntas_respuestas = cursor.fetchall()

    conn.close()
@app.route('/cambiar_contraseña', methods=['POST'])
def cambiar_contraseña():
    if 'username' not in session:
        return redirect(url_for('index'))
    usuario = request.form['usuario']
    nueva_contraseña = request.form['nueva_contraseña']
    hashed_pw = bcrypt.hashpw(nueva_contraseña.encode('utf-8'), bcrypt.gensalt())
    try:
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor()
        cursor.execute("UPDATE Usuarios SET Password = %s WHERE Username = %s", (hashed_pw, usuario))
        conn.commit()
        cursor.execute("SELECT Username FROM Usuarios")
        usuarios = cursor.fetchall()
        conn.close()
        return render_template('admin.html', username=session['username'], usuarios=usuarios)
    except mysql.connector.Error as e:
        return f"Error al cambiar contraseña: {str(e)}"

@app.route('/agregar_usuario', methods=['POST'])
def agregar_usuario():
    if 'username' not in session:
        return redirect(url_for('index'))
    username = request.form['username']
    password = request.form['password']
    rol = request.form['rol']
    admin_key = request.form.get('admin_key', '')
    if rol == 'admin' and admin_key != '123456789':
        return redirect(url_for('fracaso'))
    hashed_pw = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
    try:
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor()
        cursor.execute("INSERT INTO Usuarios (Username, Password, Rol) VALUES (%s, %s, %s)", (username, hashed_pw, rol))
        conn.commit()
        cursor.execute("SELECT Username FROM Usuarios")
        usuarios = cursor.fetchall()
        conn.close()
        return render_template('admin.html', username=session['username'], usuarios=usuarios)
    except mysql.connector.Error as e:
        return f"Error al agregar usuario: {str(e)}"

@app.route('/eliminar_usuario', methods=['POST'])

```

```

def eliminar_usuario():
    if 'username' not in session:
        return redirect(url_for('index'))
    username = request.form['usuario']
    try:
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor()
        cursor.execute("DELETE FROM Usuarios WHERE Username = %s", (username,))
        conn.commit()
        cursor.execute("SELECT Username FROM Usuarios")
        usuarios = cursor.fetchall()
        conn.close()
        return render_template('admin.html', username=session['username'], usuarios=usuarios)
    except mysql.connector.Error as e:
        return f"Error al eliminar usuario: {str(e)}"
@app.route('/admin_informate', methods=['GET', 'POST'])
def admin_informate():
    if 'username' not in session or session.get('user_id') is None:
        return redirect(url_for('index'))

    conn = mysql.connector.connect(**db_config)
    cursor = conn.cursor(dictionary=True)

    # Verificar si es administrador
    cursor.execute("SELECT Rol FROM Usuarios WHERE Id = %s", (session['user_id'],))
    rol = cursor.fetchone()['Rol']
    if rol != 'admin':
        conn.close()
        return redirect(url_for('index'))

    # Si se envía el formulario para agregar módulo
    if request.method == 'POST':
        titulo = request.form['titulo']
        descripcion = request.form['descripcion']
        imagen_url = request.form.get('imagen_url', '')
        documento_url = request.form.get('documento_url', '')
        color = request.form.get('color', 'verde')

        cursor.execute("""
            INSERT INTO ModulosInformativos (titulo, descripcion, imagen_url, documento_url, color)
            VALUES (%s, %s, %s, %s, %s)
        """, (titulo, descripcion, imagen_url, documento_url, color))
        conn.commit()

    # Obtener todos los módulos existentes
    cursor.execute("SELECT * FROM ModulosInformativos ORDER BY fecha_creacion DESC")
    modulos = cursor.fetchall()

    # También se necesita lista de usuarios para renderizar correctamente
    cursor.execute("SELECT Username FROM Usuarios")
    usuarios = cursor.fetchall()

    conn.close()

```

```
return render_template('admin.html',
    username=session['username'],
    usuarios=usuarios,
    modulos=modulos
)

if __name__ == '__main__':
    app.run(debug=True)
```

5. CAPÍTULO V: CONSTRUCCIÓN

5.1. Construcción

En la construcción del chatbot se desarrollaron las siguientes fases para su creación e implementación.

5.1.1. Estructuración del chatbot

Con el equipo de desarrollo del chatbot, se decidió las funciones del chatbot, propuestas para el diseño de las interfaces, el contenido que la base de datos almacenará y los posibles riesgos durante el proyecto. Todo a partir del problema planteado.

5.1.2. Desarrollo del chatbot

La primera fase del proyecto fue el desarrollar el chatbot para los agricultores. En esta fase se diseñó la interfaz de la página en la cual contiene el chatbot. Se utilizó Python y sus librerías para el desarrollo de las interfaces creando la interfaz para el ingreso de usuarios, la interfaz donde se realizan las preguntas, el clima del lugar, la modificación de cuentas, entre otros. También se diseñaron interfaces extras para los usuarios que son administradores. Luego se diseñó una base de datos para el chatbot, se utilizó MySQL para esta función. Después, agregaron APIs para el funcionamiento del chatbot, una se usó para que se guarden las preguntas y pueda ofrecer preguntas personalizadas para los agricultores; y otra para el funcionamiento del clima, permitiendo que nos muestre el clima del lugar seleccionado, en este mostrará el clima en la zona de Hualhuas. Finalmente se programaron las funciones que tendrá el chatbot, incluyendo también la seguridad de información para los usuarios.

5.1.3. Prueba del chatbot

Luego de haber desarrollado la interfaz del chatbot, sus funciones y conectarlo a una base de datos, se realizaron las pruebas requeridas para comprobar que el chatbot funciona correctamente. Para ello se crearon cuentas de prueba para comprobar si el ingreso de usuario estaba programado correctamente. Luego se verificó que el chat muestre el clima del lugar, después se hicieron preguntas al chatbot para verificar que los respondiera correctamente, y por último que los datos se guarden correctamente.

5.1.4. Capacitación y difusión del chatbot

Luego de comprobar que el chatbot esté funcionando correctamente, se planeó la forma de capacitar a los agricultores para que no tengan problemas al utilizar el chatbot. Por lo cual se optó por guías, tutoriales y videos.

5.1.5. Costos

Se registraron los siguientes costos

Nombre de tarea	Costo
GERENCIA DEL PROYECTO	S/ 8.550,00
Elaboración del Acta de Constitución	S/ 150,00
Identificación y Clasificación de Interesados	S/ 100,00
Análisis del Contexto Agrícola de Hualhuas	S/ 0,00
Aprobación por la Municipalidad y Equipo Técnico	S/ 0,00
Definición del Alcance y Objetivos	S/ 0,00
Elaboración del Cronograma del Proyecto	S/ 0,00
Estimación y Control del Presupuesto	S/ 2.320,00
Asignación de Recursos Humanos y Técnicos	S/ 4.100,00
Planificación de Comunicaciones Internas y Externas	S/ 0,00
Planificación de la Calidad del Producto	S/ 0,00
Planificación de las Adquisiciones	S/ 0,00

Aseguramiento de la Calidad de Desarrollo	S/ 0,00
Coordinación Técnica del Equipo Interdisciplinario	S/ 300,00
Reuniones con Comunidades Agrícolas Locales	S/ 0,00
Preparación de Infraestructura Tecnológica y Materiales	S/ 1.580,00
Supervisión de Calidad del Chatbot y Base de Datos	S/ 0,00
Documentación de Cambios y Retroalimentación	S/ 0,00
Cierre Formal de Actividades	S/ 0,00
Reunión de Evaluación con Patrocinador y Comunidad	S/ 0,00
EVALUACION	S/ -
Estudio de Viabilidad Técnica y de Adopción Social	S/ 0,00
Estimación de Costos del Proyecto y Beneficio Esperado	S/ 0,00
Clasificación y Validación de Requisitos Funcionales y No Funcionales	S/ 0,00
Análisis de Soluciones Similares	S/ 0,00
DISEÑO Y FABRICACIÓN	S/ 4.980,00
Diseño de Arquitectura Modular del Sistema	S/ 0,00
Diseño de la Interfaz Web	S/ 800,00
Modelado de la Base de Datos Agronómica	S/ 0,00
Diseño del Flujo Conversacional del Chatbot	S/ 0,00
Elaboración de Prototipo Interactivo para Pruebas Internas	S/ 0,00
Validación del Diseño con Usuarios Agrícolas y Expertos	S/ 400,00
Desarrollo del Motor Conversacional del Chatbot	S/ 2.980,00
Integración del Módulo Climático y Personalización	S/ 0,00
Programación de Reglas Técnicas para Recomendaciones	S/ 0,00
Conexión con la Base de Datos Agronómica Validada	S/ 0,00

Desarrollo de la Interfaz Web Accesible para Dispositivos	S/ 0,00
Pruebas de Unidad de Cada Módulo	S/ 800,00
Iteración de Ajustes según Retroalimentación	S/ 0,00
Preparación de Versión Preliminar para Pruebas Piloto	S/ 0,00
ADECUACIÓN	S/ 2.600,00
Evaluación del Entorno Tecnológico	S/ 0,00
Adquisición de Hosting, Dominio, Certificados de Seguridad	S/ 2.500,00
Creación de Manuales, Guías y Material de Soporte	S/ 100,00
Validación del Entorno de Ejecución con Comunidad	S/ 0,00
Aseguramiento de Acceso para Agricultores en Zona Piloto	S/ 0,00
DISTRIBUCIÓN	S/ -
Ubicación de Recursos Técnicos para la Capacitación	S/ 0,00
Definir Cronograma de Implementación Piloto	S/ 0,00
Establecer Canales de Comunicación con Usuarios	S/ 0,00
Determinar Métodos de Soporte y Capacitación	S/ 0,00
Coordinación con la Municipalidad para Lanzamiento	S/ 0,00
Reuniones con Asociaciones Agrarias y Tomeros	S/ 0,00
Entrega de Chatbot a Grupos de Agricultores	S/ 0,00
Levantamiento de Observaciones en el Campo	S/ 0,00
Campañas de Difusión Comunitaria	S/ 0,00
CIERRE	S/ -
Cierre Formal de Actividades	S/ 0,00
Revisión Técnica y Aprobación Final	S/ 0,00
Redacción del Informe Final del Proyecto	S/ 0,00

Firma del Acta de Cierre y Validación	S/ 0,00
TOTAL	S/ 16.130,00

Tabla 4. Estimación de costos

Reserva de Gestión

Aspectos	Costo
Intereses sobre Préstamos	S/ 450,00
Provisión para Inflación	S/ 300.00
Tasas de Cambio de Divisas	S/ 120,00
TOTAL	S/ 870,00

Tabla 5. Reserva de Gestión

Gráfico de costos:

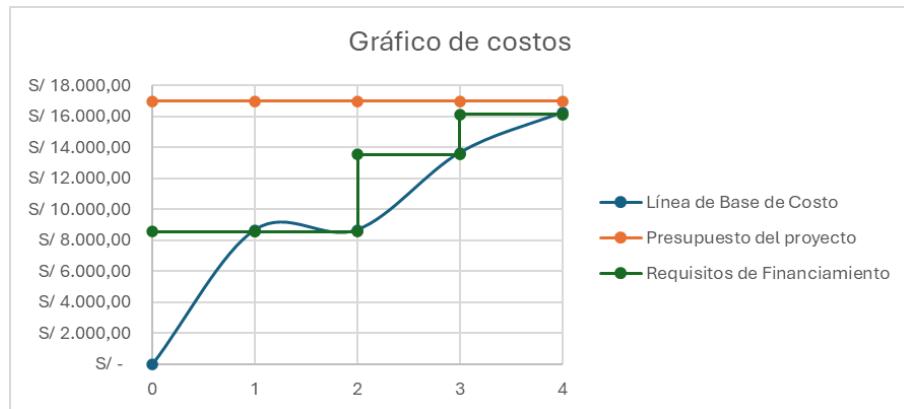


Gráfico 1. Gráfico de costos

5.2. Pruebas y resultados

Luego de realizar las preguntas se almacenaron los resultados en el usuario de administrador, en la cual nos muestra un resumen de las preguntas, donde se muestra el total de preguntas realizadas, las preguntas respondidas del chatbot y las que no pudo responder, como se muestra en la Figura 7. Resumen general de preguntas.

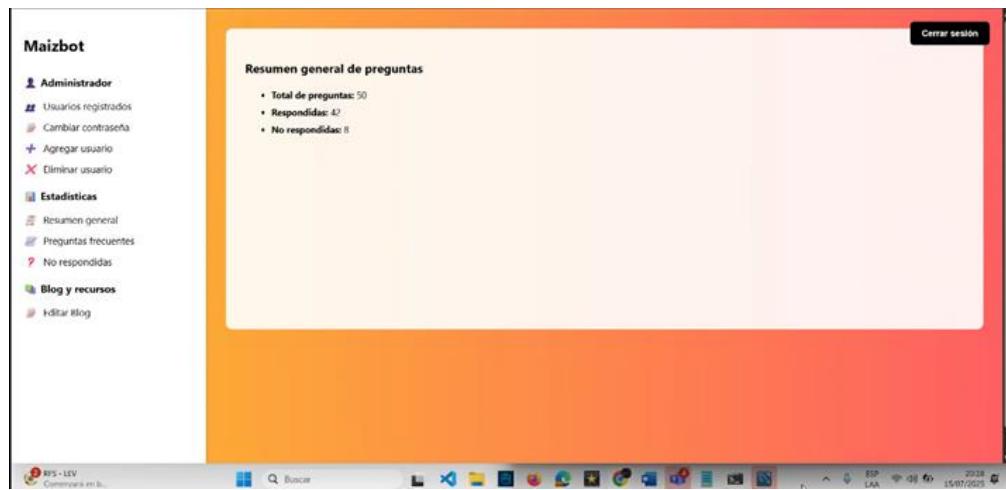


Figura 7. Resumen general de preguntas

A partir del resumen se puede concluir que, durante la evaluación del chatbot, se tiene un 84% de tasa de éxito al momento de contestar la pregunta. Este porcentaje se puede sacar a partir de la siguiente fórmula:

$$\text{Tasa de éxito} = \frac{\text{Respuestas correctas}}{\text{Total de preguntas}} \times 100 = \frac{42}{50} \times 100 = 84\%$$

Por lo que el chatbot mostró un rendimiento aceptable para ser el inicio, más el 16% de tasa de error muestra que tiene margen de mejora, por lo que se puede realizar las mejoras pertinentes para el futuro.

CONCLUSIONES

El uso de chatbot en la actualidad significó en la creación de ideas innovadoras para apoyar a las personas, teniendo muchas opciones para desarrollarse en muchas áreas. Durante el desarrollo de este proyecto, se realizó la implementación de esta herramienta innovadora para un área que no se desarrolló mucho en este aspecto, en la cual se integró aspectos importantes para el funcionamiento del chatbot, y facilitando uso para los agricultores, haciéndolo intuitivo. Durante los resultados se mostró que el chatbot respondió correctamente a 42 preguntas, por lo que mostró un gran potencial para el apoyo a los agricultores con el tema de siembra del maíz. Sin embargo, también mostró que tiene un margen de mejora, por lo que se tiene que mejorar para futuros proyectos.

REFERENCIAS BIBLIOGRÁFICAS

- FAO. (2021). La ciencia, la innovación y la transformación digital al servicio de la inocuidad alimentaria. <https://openknowledge.fao.org/items/315d5d54-a5b9-476c-8298-d054041cd1cb>
- Sandeep V. Gaikwad, Amol D. Vibhute, Karbhari V. Kale, Suresh C. Mehrotra,(2021) An innovative IoT based system for precision farming,Computers and Electronics in Agriculture.VOLUME 187,10629. <https://doi.org/10.1016/j.compag.2021.106291>.
- Suhaman,Tía Sari,Kamandanu Kamandanu,Dwy Aulianti,Muhammad Adhi,Utih Amartiwi(2023). Plant disease detection using Plantix: A mobile application. *Journal of Agricultural Technology*, 15(3), 635-645. [10.53889/gmpics.v2.173](https://doi.org/10.53889/gmpics.v2.173)
- Villanueva Gómez Keyler Liniker (2024). Técnicas de aplicación de fertilizantes en maíz (*Zea mays L.*) en el Valle de Santa, Ancash. *Revista Latinoamericana de Ciencias Agrarias - RLCA*;2(1).<https://doi.org/10.5281/zenodo.12728728>
- Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing* (3rd ed.). Stanford University.
- Pérez-Marín, D., & Pascual-Nieto, I. (2011). *Conversational agents and natural language interaction: Techniques and effective practices*. IGI Global.
https://www.researchgate.net/publication/235913109_Conversational_Agents_and_Natural_Language_Interaction_Techniques_and_Effective_Practices
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017). Big Data in Smart Farming – A review. *Agricultural Systems*, 153, 69-80. <https://doi.org/10.1016/j.agsy.2017.01.023>