



Documentación

SISTEMA DE PREDICCIÓN DE COSECHAS

Sección: D17

Materia: Ingeniería de software

Maestra: Graciela Lara López

Integrantes del equipo:

Sosa Godinez Luis Gustavo 219544281

Salazar Solis Gregorio 222966995

Ornelas Durán Natalia Alessandra 222966235

Cuevas Ruvalcaba Jessica Marie 222966022

Canul García Diego 220414537

Fecha de entrega:

12 de mayo del 2025

INTRODUCCIÓN.....	2
Breve descripción del proyecto.....	2
Importancia del proyecto / Justificación.....	2
Objetivo General.....	3
Objetivos Específicos.....	3
PLANTEAMIENTO DEL PROBLEMA.....	4
Definición del Problema.....	4
Contexto y Relevancia.....	4
Alcance del Proyecto.....	5
ANÁLISIS DE REQUISITOS.....	5
Requisitos Funcionales.....	5
Requisitos No Funcionales.....	6
Casos de Uso.....	6
Historias de Usuario.....	8
DISEÑO DEL SISTEMA.....	9
Diagrama de clases y entidad-relación (ERD).....	9
Arquitectura del software (Cliente-servidor, MVC, etc.) (Diagrama de componentes).....	9
Diseño de interfaces (Prototipos de UI).....	11
Flujo de datos / diagramas de secuencia.....	13
DESARROLLO Y TECNOLOGÍAS.....	13
Lenguajes de programación y frameworks utilizados.....	13
Base de datos y estructura.....	14
Herramientas y plataformas empleadas.....	14
PLANIFICACIÓN Y METODOLOGÍA.....	15
Metodología de desarrollo (Scrum).....	15
Cronograma del proyecto (Diagrama de Gantt).....	17
Roles y responsabilidades.....	18
Planeación de recursos y estimación de costos.....	19
IMPLEMENTACIÓN.....	20
Descripción del desarrollo del software.....	20
Capturas de pantalla de la aplicación en funcionamiento.....	21
Pruebas realizadas (unitarias, de integración, de usuario).....	22
RESULTADOS Y EVALUACIÓN.....	24
Comparación con los objetivos iniciales.....	24
Problemas encontrados y cómo se resolvieron.....	27
Posibles mejoras futuras.....	29
CONCLUSIONES Y RECOMENDACIONES.....	30
Reflexión sobre la experiencia del desarrollo.....	30
Lecciones aprendidas.....	31
BIBLIOGRAFÍAS Y ANEXOS.....	32

INTRODUCCIÓN

Breve descripción del proyecto

El presente proyecto tiene como finalidad el diseño, desarrollo e implementación de una plataforma tecnológica integral orientada a la predicción del rendimiento de cultivos mediante el uso de modelos estadísticos avanzados y técnicas modernas de análisis de datos. Esta solución se fundamenta en la recopilación, procesamiento e interpretación de grandes volúmenes de información agrícola y ambiental como datos meteorológicos, características del suelo, historial de producción, prácticas de cultivo, uso de insumos, entre otras, con el fin de generar predicciones precisas y accionables para los productores.

La plataforma emplea enfoques de ciencia de datos y aprendizaje automático para identificar patrones y correlaciones clave que impactan directamente en el desempeño de los cultivos. A través de algoritmos predictivos personalizados, se podrá anticipar el rendimiento esperado bajo distintas condiciones, permitiendo así a los agricultores tomar decisiones fundamentadas en datos reales y no solo en la experiencia o la intuición.

Entre los beneficios esperados se encuentran: la mejora en la planificación agrícola, la optimización del uso de recursos como agua, fertilizantes y energía, la reducción de pérdidas por eventos climáticos, el aumento de la productividad y la sostenibilidad de los sistemas agrícolas, y la minimización de riesgos económicos. Además, se busca facilitar el acceso a herramientas tecnológicas de vanguardia, haciendo que estén disponibles incluso para pequeños y medianos productores, quienes muchas veces enfrentan limitaciones tecnológicas y de información.

Este proyecto representa una apuesta por la innovación en el sector agrónomo, promoviendo una agricultura más inteligente, resiliente y adaptada a los desafíos del cambio climático y la creciente demanda de alimentos a nivel mundial. La plataforma no solo será una herramienta tecnológica, sino un aliado estratégico en la toma de decisiones del productor moderno.

Importancia del proyecto / Justificación

En el contexto actual de la agricultura, la toma de decisiones sigue estando, en muchos casos, fuertemente basada en la experiencia empírica, la intuición del productor y estimaciones generales que no siempre reflejan las condiciones reales del entorno. Esta dependencia de métodos tradicionales genera un alto nivel de incertidumbre en torno al rendimiento y la calidad de las cosechas, lo que puede traducirse en pérdidas económicas significativas, uso ineficiente de recursos y una menor capacidad de respuesta ante condiciones adversas como sequías, plagas o variaciones climáticas inesperadas.

Este proyecto surge como una respuesta innovadora a esa problemática, proponiendo el desarrollo de un sistema inteligente que aproveche el análisis de datos históricos y predicciones climáticas en tiempo real para generar proyecciones más precisas sobre el rendimiento agrícola. Al incorporar tecnologías de ciencia de datos y modelado predictivo, la solución permitirá identificar patrones y tendencias que no son fácilmente detectables a través de la observación tradicional.

La importancia de esta iniciativa radica en su potencial para transformar el proceso de planificación agrícola. Al brindar información basada en evidencia, los productores podrán anticiparse a posibles escenarios, ajustar sus estrategias de siembra, fertilización y riego, y maximizar la eficiencia del uso de insumos. Esto no solo contribuye a aumentar la rentabilidad de las explotaciones agrícolas, sino que también promueve una gestión más sostenible de los recursos naturales, lo que es especialmente relevante en un contexto de cambio climático y creciente presión sobre los sistemas alimentarios.

Además, el sistema contribuirá a reducir las brechas tecnológicas en el sector, haciendo que estén disponibles incluso para pequeños y medianos productores, quienes muchas veces enfrentan limitaciones y suelen ser los más vulnerables frente a la volatilidad del entorno productivo. En definitiva, se trata de una iniciativa con un fuerte impacto económico, social y ambiental, orientada a modernizar la agricultura y hacerla más resiliente y eficiente a largo plazo.

Objetivo General

Desarrollar un sistema inteligente de predicción del rendimiento y calidad de cultivos, integrando datos agrícolas, climáticos y ambientales, para optimizar decisiones y recursos en el sector agrícola, con un diseño accesible y fácil de usar para todo tipo de productores.

Objetivos Específicos

1. Recopilar y procesar datos agrícolas y ambientales, como históricos de producción, características del suelo, prácticas de cultivo y datos meteorológicos para integrarlos en una base de datos para análisis predictivo.
2. Desarrollar modelos predictivos basados en aprendizaje automático y análisis estadístico para identificar patrones y correlaciones entre variables climáticas y agronómicas, con el fin de predecir con alta precisión el rendimiento y la calidad de los cultivos bajo diferentes escenarios.
3. Diseñar e implementar una interfaz de usuario intuitiva y accesible, que facilite la visualización de predicciones, y reportes en un formato claro y accesible.

4. Integrar APIs de datos meteorológicos en tiempo real y otras fuentes de información ambiental para enriquecer la interfaz.
5. Generar reportes que incluyan la información del cultivo, la información del terreno y la información de los resultados de la predicción.

PLANTEAMIENTO DEL PROBLEMA

Definición del Problema

El rendimiento de los cultivos cambia cada año debido a diversos factores que están fuera del control de los agricultores como lo es la calidad de las semillas, las condiciones del suelo, las prácticas de siembra y, sobre todo, el clima. Todos estos factores son impredecibles provocando una incertidumbre que genera varios inconvenientes a los productores, se complica la planificación de las cosechas, aumenta el riesgo de pérdidas económicas y dificulta el uso eficiente de recursos como agua, fertilizantes y energía.

Contexto y Relevancia

La producción agrícola en México enfrenta actualmente un escenario de grandes retos y oportunidades. El país cuenta con una amplia diversidad climática y de suelos, lo que le permite producir una gran variedad de cultivos y ser un actor clave en el comercio agrícola internacional. Sin embargo, también enfrenta serios impedimentos de los cuales los que afectan de mayor manera son las limitaciones climáticas como sequías, lluvias intensas o cambios climáticos inesperados. Y las limitaciones tecnológicas, pues gran parte del campo mexicano está conformado por pequeños productores con acceso limitado a tecnología, financiamiento y capacitación provocando un uso excesivo de agua, o la aplicación ineficiente de fertilizantes y pesticidas. De tal manera que se ve afectada de forma directa su sostenibilidad y competitividad.

Por ello, ya es urgente incorporar herramientas digitales para transformar el sector agrícola. Esto ayudará a los agricultores a planificar mejor, reducir riesgos y optimizar recursos, beneficiando también a investigadores y empresas agrícolas que buscan mejorar la productividad y sostenibilidad del sector.

Alcance del Proyecto

El proyecto comenzará desarrollando la plataforma para un cultivo y una región específica en México, seleccionados por la disponibilidad de datos históricos. Una vez probada y ajustada, la plataforma se adaptará para incluir otros cultivos y regiones. Está diseñada para ser utilizada por agricultores de todos los niveles, desde pequeños productores hasta grandes empresas.

ANÁLISIS DE REQUISITOS

Requisitos Funcionales

1. El sistema permite la creación de cuentas a los agricultores, únicamente si el formulario de datos es llenado correctamente.
2. El sistema permite el inicio de sesión únicamente si ya posee una cuenta creada y sus credenciales coinciden con la base de datos.
3. La aplicación debe ser intuitiva y sencilla de usar, con iconografía representativa y comprensible para agricultores con poca experiencia técnica.
4. El sistema debe permitir la carga de datos históricos de clima (precipitación, temperatura), suelo y resultados previos de cultivos.
5. La predicción generada es según la ubicación establecida.
6. El usuario podrá generar reportes exportables (PDF).
7. Los PDF generados deberán contar con la información del cultivo, la información del terreno y la información de la predicción.
8. La base de datos se diseña bajo principios de integridad referencial, empleando claves foráneas para establecer relaciones entre las distintas entidades, garantizando un manejo seguro y estructurado de la información.
9. Existe un historial de predicciones generadas en la base de datos.
10. El historial de predicciones se encuentra en formato de tabla y es visible para el usuario.
11. En la opción de Gestión de terrenos, se puede crear y eliminar cualquier terreno ya ingresado, el cual también se eliminará de la base de datos.
12. Se muestran mediante messagebox, mensajes de éxito o error, cada que se efectúe algún cambio en el sistema, según sea el caso.
13. El sistema permite la navegación dentro de la aplicación sin necesidad de estar cerrando pantallas, ni saliendo de la sesión.
14. En el dashboard se muestra un Widget interactivo del clima en tiempo real.
15. Integrar API para el clima en tiempo real.

16. En el dashboard se muestra un Widget que cuenta con las noticias más relevantes en el sector agrícola.
17. Integrar API de noticias en tiempo real.
18. En el dashboard se muestra un Widget interactivo del progreso de cultivo.
19. El sistema permite al usuario contar con un perfil con toda su información, el cual es modificable para que sea personalizado.
20. Las predicciones se realizan en un tiempo menor a 1 minuto.
21. Las predicciones generadas tras llenar el formulario, se muestran en un messagebox de manera inmediata luego de presionar el botón.
22. Tras finalizar la cosecha, el usuario podrá agregar el rendimiento real obtenido.

Requisitos No Funcionales

1. El sistema permite la recuperación de cuenta en caso de haber olvidado la contraseña mediante el correo ingresado.
2. Se muestra mediante un icono de identificación en pantalla, el nombre del usuario que está dentro de la aplicación.
3. El tiempo de espera tras hacer un cambio no debe superar 1 segundo.
4. El sistema preguntará al usuario si está seguro de realizar un cambio, a manera de confirmación, cada que este desee hacer cualquier modificación.
5. La tabla mostrada en la opción de Gestión de Terrenos contendrá únicamente la información más relevante, no mostrará todos los datos que fueron solicitados para ingresar a la base de datos.
6. Cierre de pantalla tras presionar cualquier botón del dashboard, en otras palabras, no pantallas en cascada.
7. El sistema estará disponible todo el tiempo para garantizar que los agricultores puedan acceder en cualquier momento del día.
8. El sistema debe ser capaz de crecer sin problemas cuando haya más usuarios, más datos o se añadan nuevas regiones.
9. Existe adaptabilidad a los diversos tamaños de pantallas.
10. Tiene botones grandes, buen contraste y textos claros, de esta forma los usuarios no tendrán ningún problema para usar el sistema.
11. Los formularios son claros en el dato que están solicitando, por lo que son sencillos de llenar.

Casos de Uso

Caso 1: Creación de cuenta e inicio de sesión

Actor: Agricultor / Usuario

Descripción: El usuario crea una cuenta para acceder al sistema. Para ello se le piden los datos: usuario, contraseña, teléfono, estado, municipio y correo.

Precondición: El usuario no debe estar registrado.

Postcondición: El usuario queda registrado en la base de datos, ya goza de un perfil y puede iniciar sesión.

Caso 2: Inicio de sesión

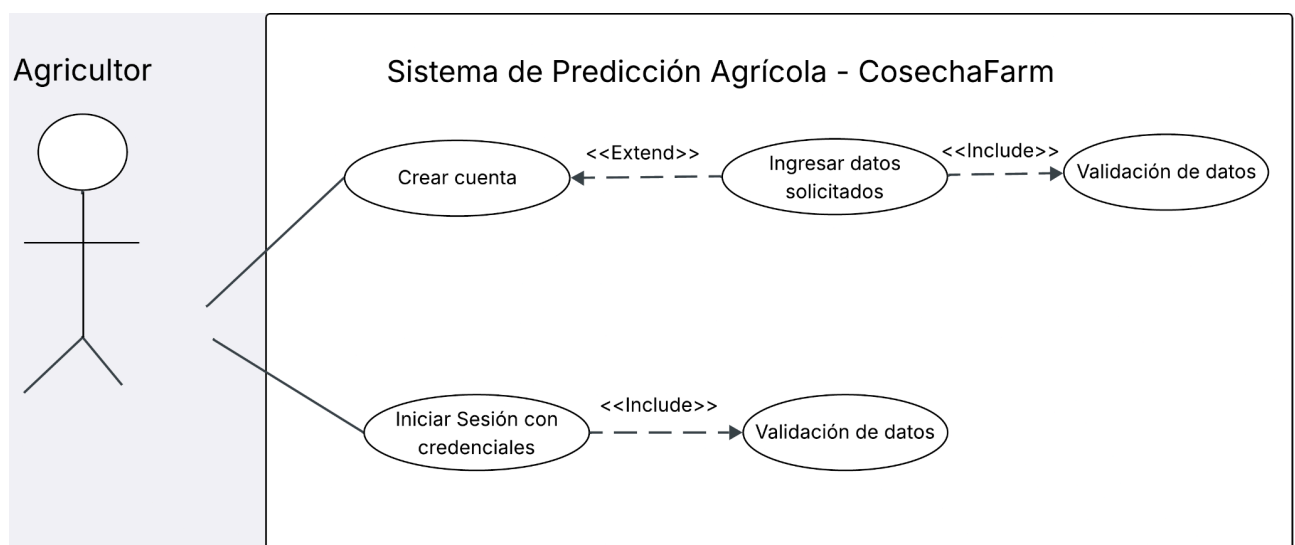
Actor: Agricultor / Usuario

Descripción: Se puede acceder al sistema utilizando sus credenciales de usuario y contraseña.

Precondición: El usuario debe estar registrado.

Postcondición: Accede a la plataforma y puede utilizar las funcionalidades.

Diagrama de caso de uso del caso 1 y caso 2:



Flujo del diagrama

1. El agricultor puede o crear su cuenta o iniciar sesión.
 - 1.1. 1.1Si toma la opción de iniciar sesión con sus credenciales, únicamente se hace una validación de datos
 - 1.2. Y en caso de que desee crear una cuenta, pasamos a otro proceso donde se tendrán que ingresar los datos solicitados
 - 1.2.1. Seguido de una validación de datos

Caso 3: Ingreso de datos

Actor: Agricultor / Usuario

Descripción: El usuario introduce las características solicitadas respecto a su cultivo (cultivo, tamaño del terreno, tipo de suelo, estado, municipio, precipitación, temperatura promedio).

Precondición: El usuario debe haber iniciado sesión.

Postcondición: Los datos del terreno quedan registrados para su análisis y se asocia con la predicción del rendimiento a generar.

Caso 4: Obtención de predicción

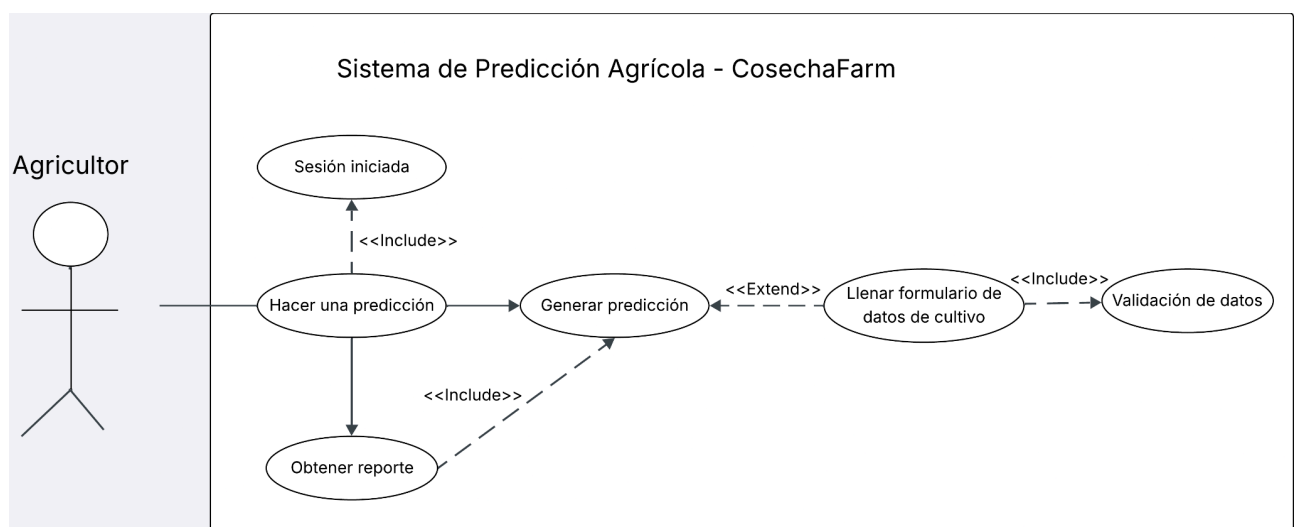
Actor: Agricultor / Usuario

Descripción: El sistema analiza los datos y genera una predicción del rendimiento incluyendo también recomendaciones.

Precondición: Datos ingresados y validados.

Postcondición: El usuario puede tomar decisiones basadas en la predicción la cual también podrá tener en un reporte PDF.

Diagrama de caso de uso del caso 3 y caso 4:



Flujo del diagrama

Para generar una predicción, es necesario haber iniciado sesión y haber completado previamente el formulario con los datos del cultivo, los cuales deben ser validados. Aunque la predicción puede generarse únicamente tras rellenar y validar dicho formulario, el acceso al reporte de la predicción generada requiere obligatoriamente haber completado el proceso de predicción.

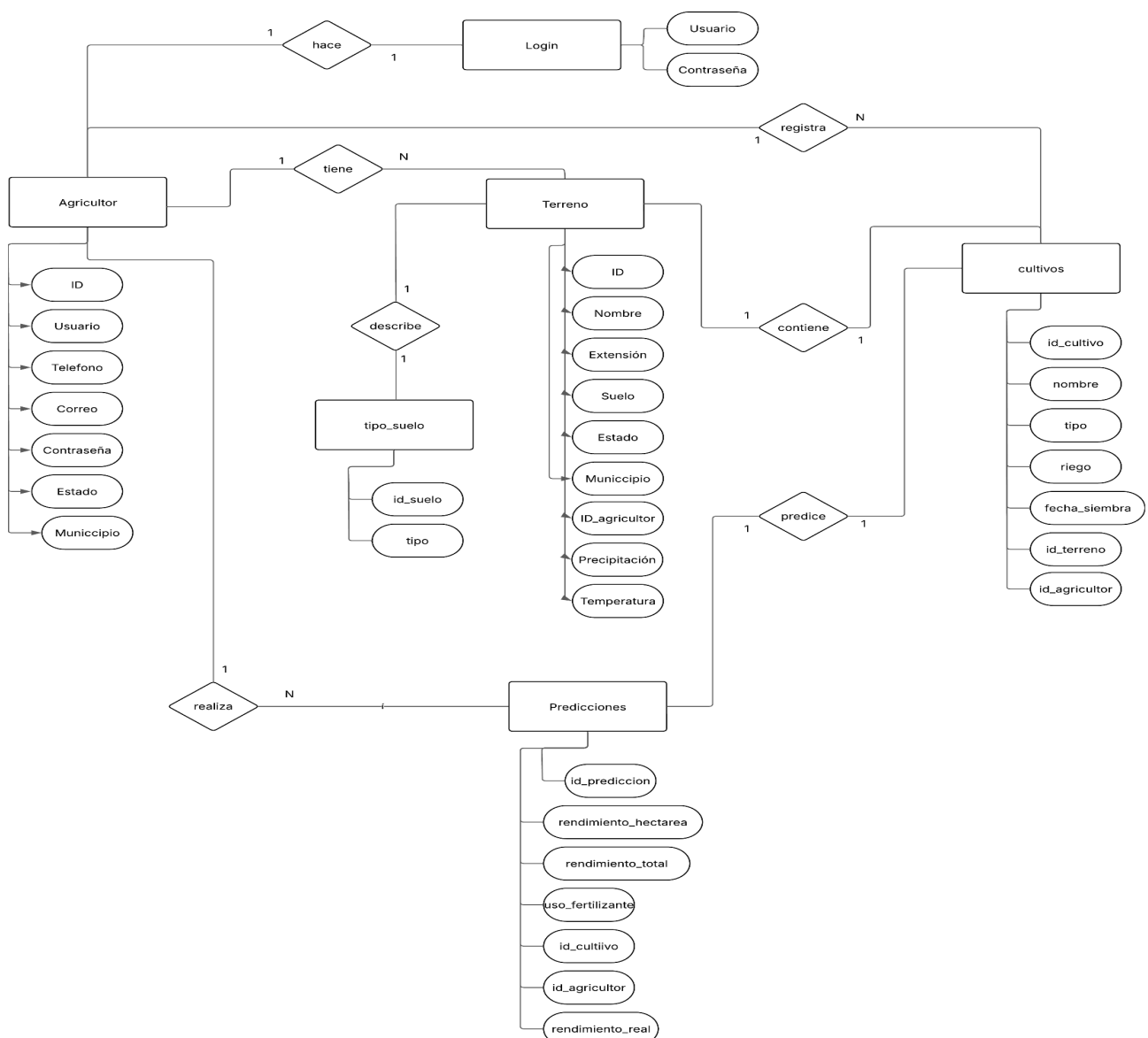
Historias de Usuario

- Como agricultor, quiero un sistema que me pueda calcular la cosecha final que tendré en una temporada.
- Como investigador, quiero analizar datos históricos de una región para estudiar el impacto del clima en un cultivo específico.

- Como agricultor, quiero poder tener los resultados de la predicción en un PDF para que pueda compartirlos con asesores, instituciones.
- Como agricultor indeciso, quiero comparar las predicciones de rendimiento de distintos cultivos en mi terreno, para decidir cuál sembrar según los resultados esperados.
- Como investigador quiero poder tener de alguna forma, un registro o historial de las predicciones que se han realizado.

DISEÑO DEL SISTEMA

Diagrama de clases y entidad-relación (ERD)

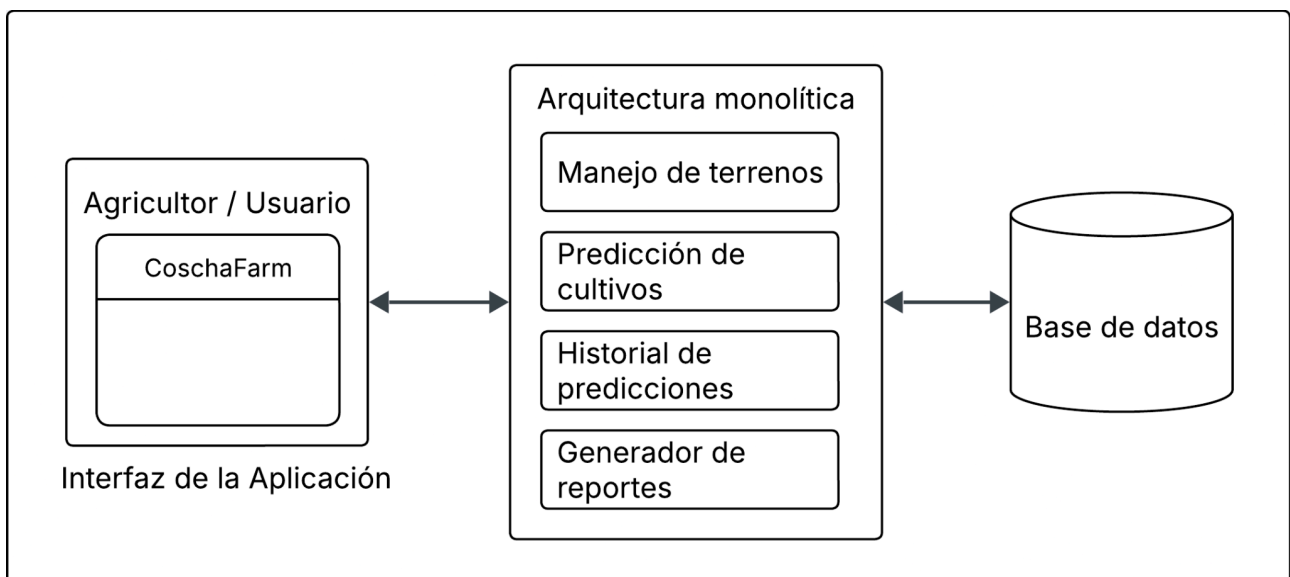


Arquitectura del software (Cliente-servidor, MVC, etc.) (Diagrama de componentes)

Se decidió utilizar una arquitectura de microservicios para el proyecto. Sin embargo, al encontrarse en una fase inicial como proyecto académico, es más conveniente implementarla inicialmente como una arquitectura monolítica. Aunque ambos enfoques comparten algunos principios, el diseño monolítico resulta más adecuado para proyectos más pequeños y simplificados, facilitando el desarrollo y la gestión en etapas tempranas.

Se plasman de esta manera, es decir como opciones en arquitectura monolítica, las acciones, que escalando el proyecto, vendrían siendo los microservicios. En este caso son:

- Manejo de terrenos
- Predicción de cultivos
- Historial de predicciones
- Generador de reportes



Diseño de interfaces (Prototipos de UI)

Imagen / Logo

Crear cuenta

Iniciar Sesión

Iniciar Sesión

Usuario

Contraseña

Ingresar

Volver

Crear Cuenta

Usuario

Telefono

Ubicación

Correo

MessageBox

Cuenta creada exitosamente

Cerrar sesión

Widget Clima

Widget Tips

Widget Adicional

Registro de datos

Monitoreo / Gestión

Impresión de predicción

Registro de Datos

Nombre cultivo

Fecha de siembra

Ubicación

Superficie sembrada

Generar Predicción

Predicción

Se estima una producción de
_(Cantidad producida)_kg
de_(cultivo)_ ...

Monitoreo / Gestión

Cultivo	Fecha de siembra	Resultado de predicción

[Volver](#)

Imprimir predicción

Historial de reportes generados

Cultivo	Fecha del reporte	Archivo

[Volver](#)

Reporte de predicción

Fecha del reporte:

Ubicación:

Agricultor responsable:

Se espera que el cultivo de _(cultivo)_ realizado el _(fecha de siembra)_ en un total de _(superficie sembrada)_, de como producción final de cosecha un total de _(resultado de la predicción)_.

Datos generales esperados del cultivo

Tipo de suelo:

Condiciones Climaticas:

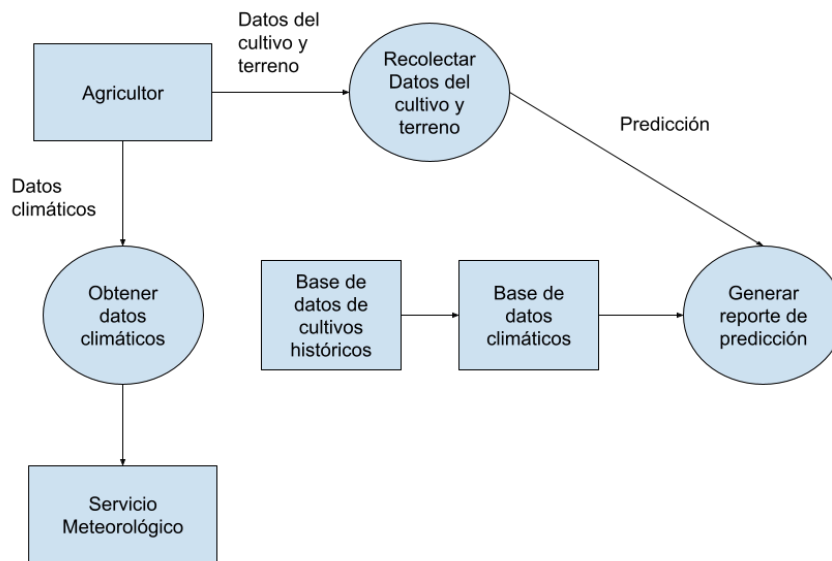
- Precipitación:

- Temperatura media:

Conclusiones y recomendaciones

- "Se espera una cosecha dentro del rango óptimo para esta variedad."
- "Se sugiere coordinar la logística de recolección y almacenamiento anticipadamente."
- "Se sugiere el uso del fertilizante X."

Flujo de datos / diagramas de secuencia



La entidad agricultor proporciona datos sobre el cultivo y el terreno, posteriormente recibe el reporte de predicción.

El Servicio Meteorológico es la fuente de datos climáticos actualizados.

En los procesos se recolecta la información que es proporcionada por el agricultor, como podría ser la ubicación, el tipo de cultivo, etc.

Seguido de esto se reciben los datos desde el servicio meteorológico, para calcular la predicción se integran los datos históricos y climáticos y de esta información se procesa el reporte de predicción, el cual se le entrega al agricultor

La base de datos de cultivos históricos contiene información previa del rendimiento de los cultivos por el tipo, zona, fecha, etc.

La base de datos climáticos almacena datos de clima pasados y actuales

DESARROLLO Y TECNOLOGÍAS

Lenguajes de programación y frameworks utilizados

Todo el código fue realizado en el lenguaje de programación Python. Python ofrece una amplia biblioteca estándar y cuenta con un ecosistema muy robusto de terceros, lo que permite abordar una gran cantidad de tareas como el análisis de datos y la automatización. Las librerías más relevantes utilizadas fueron pandas, numpy, psycpg2, json, joblib y sklearn

En específico, la parte de la IA, fue desarrollada mediante el framework de Machine Learning Scikit-Learn. Lo utilizamos debido a que se utiliza principalmente para análisis de datos, modelado predictivo y aprendizaje automático. Por ello mismo, es uno de los más populares debido a su simplicidad, eficiencia y amplia colección de algoritmos.

Base de datos y estructura

Base de datos SQL (PostgreSQL) en donde se implementaron las siguientes tablas:

- Agricultor, la cual contaba con las columnas de usuario, teléfono, correo electrónico, ubicación y contraseña.
- Cosecha, la cual contaba con las columnas del nombre del cultivo, fecha de siembra, ubicación y la superficie sembrada.
- Login, la cual contaba únicamente con 2 columnas, la de usuario y contraseña.
- Registro de datos, la cual contaba con las columnas cultivo, cantidad, fecha de siembra y ubicación

Herramientas y plataformas empleadas

Github

GitHub es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste fue utilizado ya que nos permite colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un historial de seguimiento detallado del progreso.

Monday

Es una plataforma de gestión del trabajo diseñada para facilitar la planificación, organización y seguimiento de tareas y proyectos. Se creó un tablero especial compartido

para el equipo en el que su interfaz visual e intuitiva permite coordinar activamente las actividades de forma colaborativa, adaptándose a los distintos flujos de trabajo y necesidades

Discord

En esta plataforma se creó un servidor especial que facilitó la comunicación durante el desarrollo del proyecto, mediante el uso de un chat de voz y la opción de compartir pantalla. También se permitió compartir avances organizados por ramas o módulos, además de contar con distintas herramientas para la gestión del proyecto y la resolución de dudas específicas.

PLANIFICACIÓN Y METODOLOGÍA

Metodología de desarrollo (Scrum)

Optamos por el uso de la metodología Scrum debido a que presenta un marco de trabajo ágil utilizado para gestionar proyectos de manera iterativa e incremental, centrándose en la entrega de valor al cliente de forma continua. Se basa en la autogestión del equipo, la colaboración y la adaptación constante a los cambios. Como roles principales nos encontramos con:

Product Owner (Dueño del Producto), el cual Representa al cliente o usuario final. Es responsable de definir y priorizar los elementos del *Product Backlog*, asegurarse que el equipo trabaje en lo que genera más valor para el negocio y tomar decisiones sobre el producto, acepta o rechaza el trabajo entregado.

Scrum Master, el cual actúa como facilitador del equipo Scrum, se asegura de que se sigan las prácticas correctamente, elimina impedimentos que afecten al equipo de desarrollo. y ayuda al equipo a mejorar continuamente fomentando la autoorganización.

Development Team (Equipo de Desarrollo), es un grupo de profesionales que trabajan juntos para entregar incrementos funcionales del producto. Son multifuncionales y tienen la responsabilidad de estimar, planificar y entregar el trabajo comprometido en cada Sprint.

En la implementación de este tipo de metodología para el proyecto, y al tener únicamente la idea principal del software a realizar, todo el equipo jugó todos los roles.

Backlog

El backlog es la lista priorizada de funcionalidades, características y tareas del proyecto. En este sistema de predicción agrícola, el backlog se resume en:

- Interfaz amigable para agricultores.
- Predicción de rendimiento de cultivos, basada en datos históricos y climáticos.
- Generación de reportes de predicción.
- Gestión de información de suelos y cultivos.
- Integración de datos meteorológicos y noticias en tiempo real.

Los Objetivos más críticos se colocan al inicio de la lista, de modo que el equipo los aborde primero (es el Scrum Master quien se encarga de esta tarea).

Para el caso de este proyecto la priorización de tareas se dio basándose en el valor para los agricultores y la urgencia del negocio.

Sprints

Un sprint es un ciclo corto y fijo de trabajo. En la práctica se suele definir sprints de 2 semanas (aunque pueden variar). Al final de cada sprint se debe entregar un incremento funcional del sistema: un prototipo operativo o un módulo probado. Por ejemplo, para el desarrollo de nuestro proyecto teníamos planeados los siguientes sprints:

- Sprint 1: Creación de la base de datos en etapas iniciales, es decir, tablas necesarias para dar de alta usuarios.
- Sprint 2: Crear la interfaz de usuario (dashboard) para el registro de los agricultores, solo abarca el crear cuenta e inicio de sesión.
- Sprint 3: Creación del modelo de IA para generar predicciones.
- Sprint 4: Pulir el dashboard para darle utilidad a widgets y botones.
- Sprint 5: Desarrollo e implementación de las tablas de gestión de terrenos, gestión de cultivos e historial de predicciones en la base de datos y su funcionalidad en la interfaz.
- Sprint 6: Pulir detalles en el modelo de IA para la predicción.
- Sprint 7: Obtención de reportes.
- Sprint 8: Pulir todo detalle de la interfaz.

Eventos Scrum

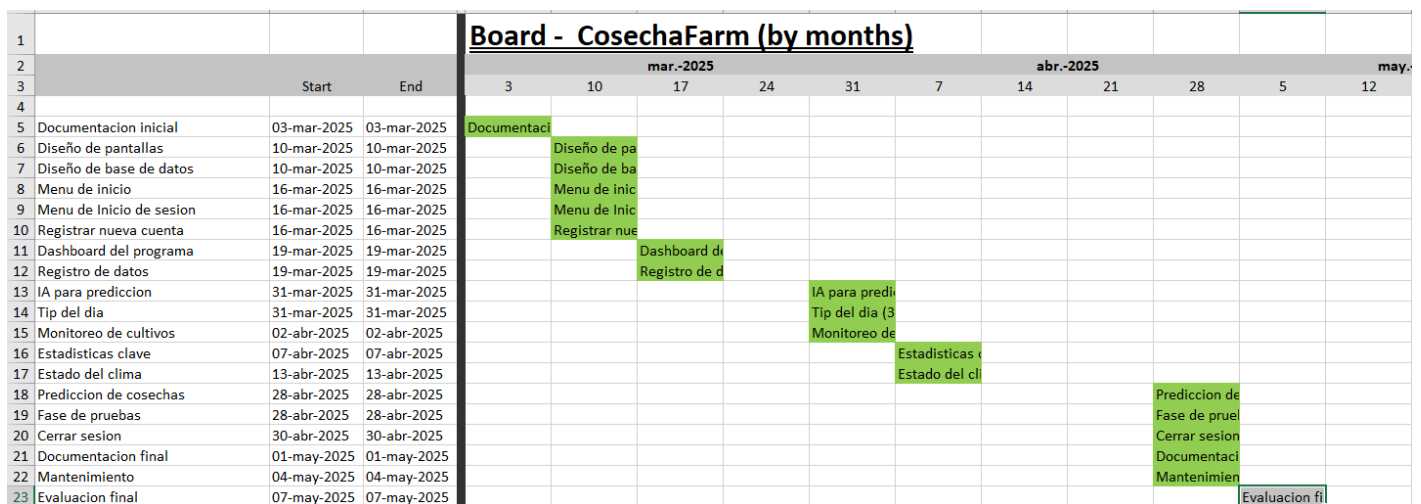
Planificación del sprint: Antes de iniciar cada sprint se realiza la planificación del sprint. En esta reunión el equipo define el objetivo del sprint y selecciona qué se incluirá.

Reunión regular: En los días hábiles el equipo se reúne para discutir los objetivos alcanzados y por realizar. Durante la reunión nos coordinamos para terminar los objetivos del día y los pasados.

Revisión del Sprint: El equipo presenta entre nosotros, todas las funcionalidades completadas durante el sprint. También lo aplicamos con los avances en la documentación.

Retrospectiva Sprint: En esta sesión el equipo reflexiona sobre el sprint reciente y al final de la retrospectiva se acuerdan acciones de mejora para aplicar en el siguiente sprint.

Cronograma del proyecto (Diagrama de Gantt)



En este Diagrama hemos marcado los tiempos que tomó cada actividad para su realización. Se toma en cuenta el tiempo que requiere hacer la investigación, el diseño, el desarrollo, las pruebas, la implementación, el mantenimiento, documentación, evaluación y el cierre.

A continuación se detalla la utilidad de cada punto:

- ➔ **Investigación y Análisis:** Identificar y recopilar datos relevantes para el desarrollo del proyecto (climáticos, de suelo, históricos de cultivos, etc.). Asi como de proyectos similares hechos anteriormente.
- ➔ **Diseño del Sistema:** Diseñar la estructura de la base de datos para almacenar los datos recopilados. Así como diseñar la interfaz de usuario mediante prototipos.
- ➔ **Desarrollo:** Selección de las herramientas que se utilizarán (lenguajes de programación, frameworks, etc.). Implementar la interfaz de usuario. Implementar la lógica del sistema de predicción.
- ➔ **Pruebas:** Realizar pruebas unitarias para cada componente del sistema (predicciones, carga de información en las bases de datos, funcionalidad correcta de la interfaz) y asegurar que todos los componentes del sistema funcionen

correctamente juntos (proyecto ya completo, sistema de predicción de cosechar), para proceder a realizar pruebas con usuarios finales para evaluar la usabilidad del sistema.

→ **Documentación y Cierre del Proyecto:** Documentar el código, la arquitectura y los procesos del sistema, en otras palabras, documentar el código y realizar la memoria del proyecto en la que se encuentra un informe final que resuma el proyecto, los resultados obtenidos.

Implementación - Mantenimiento y Soporte - Evaluación y Retroalimentación: Son puntos que no se llevarán a cabo dado que es un proyecto meramente académico, sin embargo, si en algún momento así se desea, pueden llevarse a cabo.

The screenshot shows the Monday work management interface for a project named 'CosechaFarm'. The interface includes a sidebar with navigation options like 'Inicio', 'Mi trabajo', and 'Espacios de trabajo'. The main area displays a table of tasks under the 'Completado' (Completed) status. The table has columns for 'Tarea' (Task), 'Responsa...' (Responsible), 'Estado' (Status), 'Fecha' (Date), and 'Vencimiento' (Due Date). The tasks listed include 'Documentación inicial', 'Diseño de pantallas', 'Diseño de base de datos', 'Menu de inicio', 'Menu de inicio de sesión', 'Registrar nueva cuenta', 'IA para predicción', 'Dashboard del programa', 'Cerrar sesión', 'Registro de datos', 'Tip del día', 'Monitoreo de cultivos', 'Predicción de cosechas', 'Estadísticas clave', 'Estado del clima', 'Fase de pruebas', and 'Mantenimiento'. Most tasks are marked as 'Listo' (Ready) with a green status indicator.

Tarea	Responsa...	Estado	Fecha	Vencimiento
Documentación inicial		Listo	mar. 3	mar-9
Diseño de pantallas		Listo	mar. 10	mar-12
Diseño de base de datos		Listo	mar. 10	mar-16
Menu de inicio		Listo	mar. 16	mar-19
Menu de inicio de sesión		Listo	mar. 16	mar-19
Registrar nueva cuenta		Listo	mar. 16	mar-19
IA para predicción		Listo	mar. 31	abr-27
Dashboard del programa		Listo	mar. 19	mar-23
Cerrar sesión		Listo	abr. 30	may-4
Registro de datos		Listo	mar. 19	mar-23
Tip del día		Listo	mar. 31	abr-2
Monitoreo de cultivos		Listo	abr. 2	abr-6
Predicción de cosechas		Listo	abr. 28	may-11
Estadísticas clave		Listo	abr. 7	abr-13
Estado del clima		Listo	abr. 13	abr-30
Fase de pruebas		Listo	abr. 28	may-4
Mantenimiento		Listo	may. 4	may-7

Roles y responsabilidades

Integrante	Rol	Responsabilidades
Canul García Diego	Redactor Técnico	Redacción y creación de la documentación
Cuevas Ruvalcaba Jessica Marie	Lider de proyecto Programador	Desarrollador de interfaces
		Funcionalidad entre código e interfaces
Ornelas Durán Natalia Alessandra	Redactor Técnico Programador Auxiliar	Creación y redacción de la documentación

		Desarrollador de interfaces
Salazar Solis Gregorio	Programador	Desarrollador de la IA predictiva
Sosa Godinez Luis Gustavo	Programador	Desarrollador de la Base de datos

Tareas realizadas a manera grupal

- Investigación y documentación respecto al tema
- Planteamiento de objetivos
- Diseño de interfaces
- Elaboración de conclusiones

Planeación de recursos y estimación de costos

Para estimar los costos de la producción del código se utiliza el modelo de Estimación COCOMO Básico. En este modelo el cálculo del esfuerzo se fundamenta en la relación entre el tamaño del proyecto, expresado en líneas de código y dos constantes empíricas que varían según la categoría del proyecto.

Se calcula que el tipo de proyecto es Orgánico: Pequeño, con requisitos estables y equipo experimentado.

La fórmula para calcular el esfuerzo en personas-mes es:

$$E=a*(KLOC)^b$$

Donde:

KLOC= Líneas de código = 2106

a y b son constantes que se definen según el proyecto. En este caso es Orgánico así que $a=2.4$ y $b=1.05$

$$E=2.4*(2106)^{1.05}$$

El resultado es de $E=5.24$ personas-mes

El tiempo de desarrollo es de $T=c*(E)^d$

Para proyectos Orgánicos $c=2.5$ y $d=0.38$

$$T=2.5*(5.24)^{0.38}$$

El resultado de este cálculo es de $T=4.69$ meses, redondeando el número es de 4.7 meses de desarrollo

Ahora se calcula el costo total, para ello se usa la siguiente fórmula

$$C=E*\text{Salario por personas-mes}$$

Dado que el salario de un programador varía según la región, se propondrá que el salario para este proyecto es de \$28,000 pesos mensuales

$$C=5.24*28,000=\$146,720 \text{ pesos}$$

Sin embargo por el tiempo de desarrollo de este proyecto fue de 3 meses por lo tanto, hay que ajustar la fórmula para este proyecto, es decir

$$C=N*T*Salario= 5*2*\$28,000=\$280,000 \text{ pesos}$$

Concepto	Estimación	Proyecto
Esfuerzo (E)	5.24 personas-mes	10 personas-mes
Tiempo (T)	4.69 meses	2 meses
Número de personas	~1 persona	5 personas
Costo	\$146,720 pesos	\$280,000 pesos

Esta fue la estimación de costos en total, debido a que no se utilizó ningún programa de pago o que necesitara licencia para su uso

IMPLEMENTACIÓN

Descripción del desarrollo del software

Para llevar a cabo el desarrollo del proyecto, trabajamos el software desde una perspectiva general en la que, a grandes rasgos, dividimos su estructura basándonos en tres módulos o componentes fundamentales: un motor de predicción basado en inteligencia artificial la cual utiliza los modelos linear regression (regresión lineal) y random forest (árboles de decisión), una base de datos relacional implementada en PostgreSQL y administrada mediante PgAdmin 4, y una interfaz gráfica de usuario desarrollada con Python y Tkinter. Esta arquitectura proporciona una herramienta eficiente y accesible para agricultores que buscan optimizar el rendimiento de sus cultivos y mejorar la toma de decisiones agronómicas.

El componente de inteligencia artificial constituye el núcleo analítico del sistema. Mediante algoritmos de aprendizaje supervisado, el software es capaz de predecir el rendimiento de los cultivos y determinar la necesidad de aplicación de fertilizantes, mejorando la planificación agrícola. Para la predicción del rendimiento por hectárea, se ha

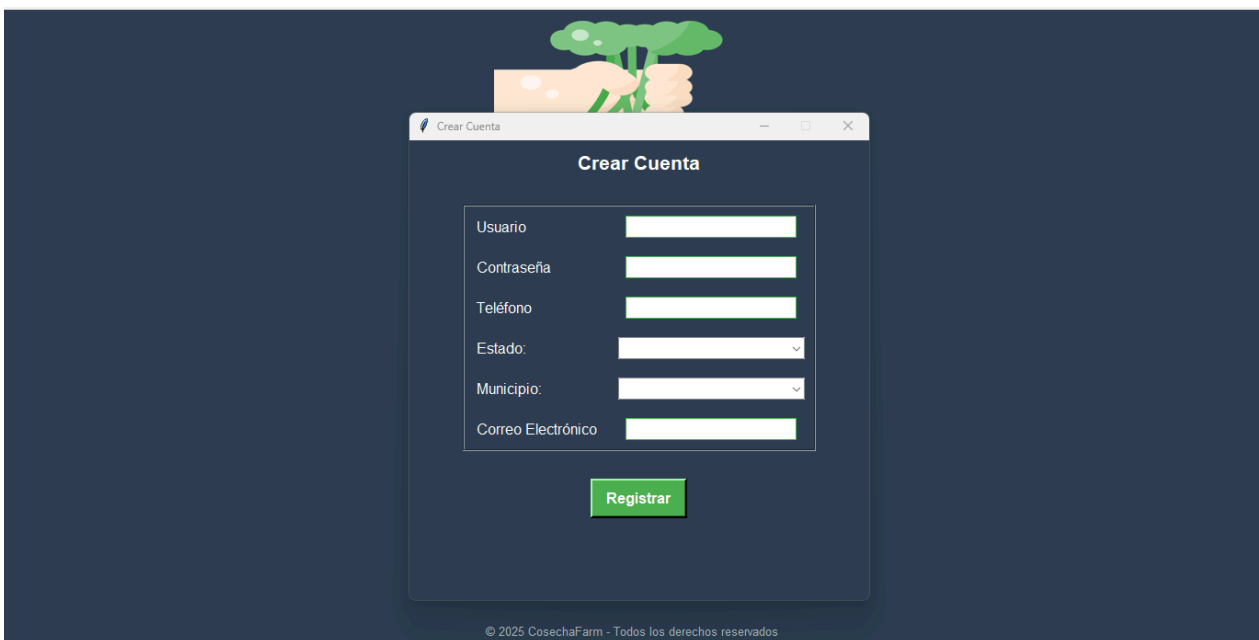
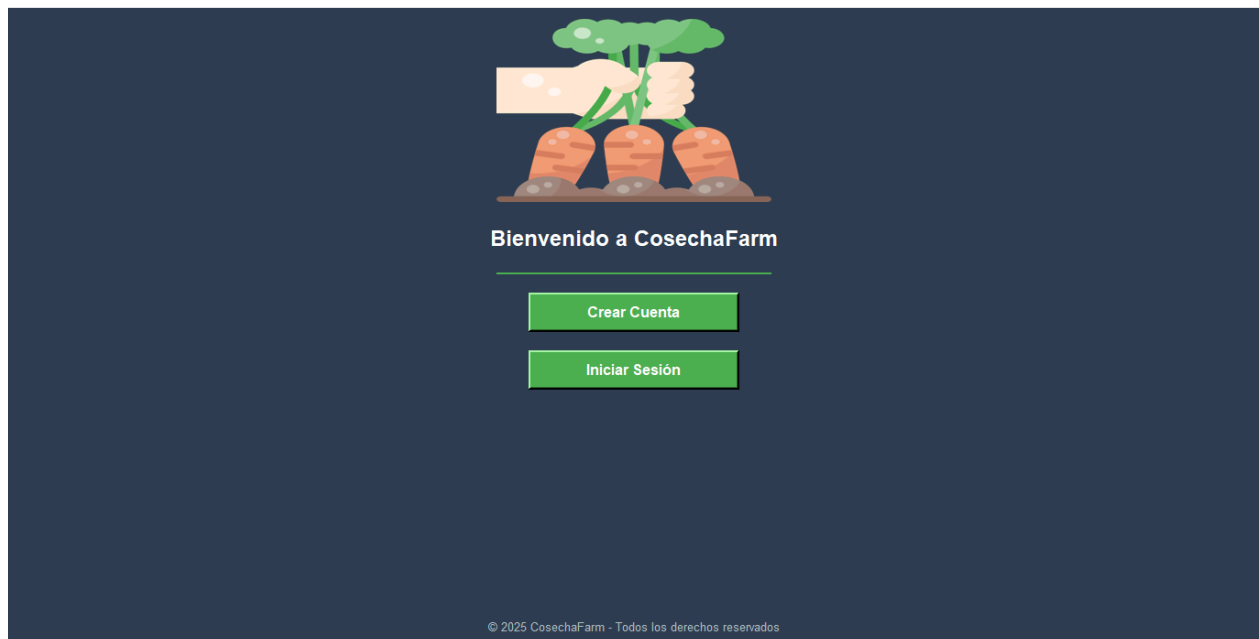
implementado un modelo de regresión lineal que considera principalmente la precipitación acumulada y otras variables climáticas. Por otro lado, el sistema incorpora un modelo de árboles de decisión, el cual clasifica la necesidad de fertilización mediante el análisis de factores como la temperatura promedio, los días estimados a la cosecha y el rendimiento esperado. Estos modelos han sido entrenados previamente y almacenados en archivos utilizando la biblioteca joblib. En el entorno de ejecución, se integran con la base de datos y procesan los datos de los cultivos registrados por el usuario, generando predicciones que se registran en la base de datos y pueden ser consultadas en reportes personalizados.

La base de datos relacional del sistema, denominada, ha sido desarrollada en PostgreSQL con el objetivo de organizar y almacenar información detallada de los usuarios, terrenos, cultivos y predicciones generadas. Su diseño estructural se compone de diversas tablas interrelacionadas que garantizan la integridad de los datos y optimizan el acceso a la información. Entre las principales entidades de la base de datos se encuentran la tabla "agricultor", que almacena la información de los usuarios registrados; la tabla "terrenos", que organiza los datos sobre las parcelas de cultivo, incluyendo su tipo de suelo y condiciones climáticas; la tabla "cultivos", que registra los productos agrícolas asociados a cada terreno; y la tabla "predicciones", donde se almacenan los resultados generados por los modelos de inteligencia artificial. La base de datos ha sido diseñada bajo principios de integridad referencial, empleando claves foráneas que establecen relaciones entre las distintas entidades, garantizando un manejo seguro y estructurado de la información.

La interfaz gráfica de usuario ha sido desarrollada utilizando el entorno Tkinter, con el propósito de ofrecer una herramienta intuitiva y accesible. El sistema permite a los agricultores gestionar sus terrenos y cultivos de manera eficiente, registrar datos agronómicos y consultar predicciones generadas por la inteligencia artificial. La interfaz está estructurada en distintos módulos funcionales, que incluyen la gestión de usuarios mediante inicio de sesión y registro, la administración de terrenos y cultivos, la ejecución de predicciones con un solo clic, el ingreso de rendimiento real de cosecha y la generación de reportes personalizados en formato PDF. Adicionalmente, se ha integrado la carga dinámica de municipios desde un archivo JSON externo, optimizando la selección geográfica de los terrenos registrados en el sistema. La presentación visual de la interfaz se ha diseñado con estilos personalizados para mejorar la legibilidad y la experiencia del usuario.

El sistema en conjunto establece un flujo de trabajo eficiente que permite la captura de información por parte del usuario, el almacenamiento de los datos en la base de datos, el procesamiento de predicciones mediante inteligencia artificial y la presentación de resultados en la interfaz gráfica. Esta integración optimiza la toma de decisiones agronómicas al proporcionar datos precisos y estructurados, asegurando que los agricultores puedan planificar sus cultivos de manera informada.

Capturas de pantalla de la aplicación en funcionamiento



Iniciar Sesión

Usuario

Contraseña

Iniciar Sesión

Volver

Cerrar Sesión

Bienvenido a tu Dashboard de CosechaFarm

Usuario : GSS

Clima en Guadalupe Victoria

21.79°C | Muy nuboso

Humedad: 27%

Noticias Agrícolas

Todos los ojos en China: Boric pone a prueba la diplomacia en complejo viaje a Beijing

Wen. df. el | 2025-05-10

Las relaciones de precios entre la soja y el maíz condiciona los escenarios futuros

La Nación | 2025-05-10

Estadísticas Clave

Producción Total: 1200 kg

Cultivos Activos: 15

Progreso de la Temporada:

75% completado

Selecciona una opción para continuar

Predicción de Cosechas

Gestión de Terrenos

Registro de Cultivos

Predicciones Realizadas

© 2025 CosechaFarm - Todos los derechos reservados

Gestión de Terrenos

Crear Terreno

Volver

ID	Nombre	Extensión (ha)	Tipo de Suelo	Estado	Municipio
3	TERRENO 3	24	duro	Durango	Guadalupe Victoria
6	Terreno norte	23	Limoso	Durango	Guadalupe Victoria
5	Terreno sur	54	Duro	Durango	Cuencame

Eliminar Terreno

Actualizar

Gestión de Terrenos

Crear Terreno

Registrar Terreno

Crear terreno

Nombre del terreno

Extensión del terreno en Hectáreas

Tipo de suelo

Estado:

Municipio:

Precipitación (en mm)

Temperatura promedio en °C

Registrar

ID	Nombre
3	TERRENO 3
6	Terreno norte
5	Terreno sur

Eliminar Terreno

Estado	Municipio
Guadalupe Victoria	
Guadalupe Victoria	
Cuencame	

Actualizar

Registro de cultivos

Crear cultivo

Volver

Registro de cultivos

Registrar Cultivo

Crear cultivo

Nombre del Cultivo

Tipo o Variedad de cultivo

Fecha de siembra (YYYY-MM-DD)

Uso de riego (1-si 0-no)

ID de terreno a sembrar

Guardar Datos

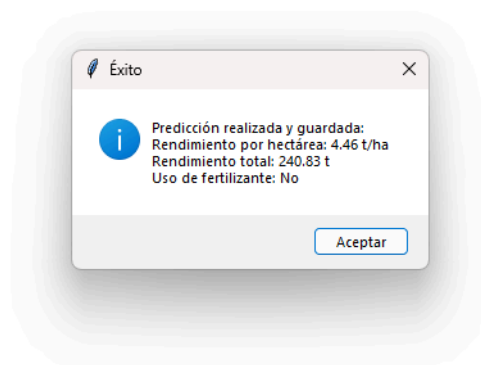
Predicción de Cosechas

ID	Nombre	Tipo	Fecha Siembra	Riego	ID Terreno
1	Cultivo 1	Maz amarillo	2025-06-11	True	3
3	Cultivo 2	Frijol negro	2025-06-06	True	3
4	Cultivo 2	Maz amarillo	2025-06-11	False	5
5	Frijo 2	Frijol rojo	2025-05-12	True	6

Realizar Predicción

Volver

4	Cultivo 2	Maz amarillo	2025-06-11	False	5
2		Frijol rojo	2025-05-12	True	



Realizar Predicción

Predicciones Realizadas

ID	Cultivo	Fecha	Rend. Hectárea (t/ha)	Rend. Total (t)	Fertilizante	Rend. Real (t)
4	Cultivo 2	2025-05-11 22:45:50.499417	4.459859242222674	240.83239908002437	False	None
3	Frijo 2	2025-05-11 19:54:45.557126	4.409937242059644	101.4285565673718	True	None
2	Cultivo 1	2025-05-11 18:06:53.654730	2.023665634266782	48.567975222402765	False	46.00
1	Cultivo 2	2025-05-11 18:06:35.052096	4.459859242222674	240.83239908002437	False	None

Registrar Rendimiento Real

Generar Reporte

Volver

Predicciones Realizadas

ID	Cultivo	Fecha	Rend. Hectárea (t/ha)	Rend. Total (t)	Fertilizante	Rend. Real (t)
4	Cultivo 2	2025-05-11 22:45:50.499417	4.459859242222674	240.83239908002437	False	None
		2025-05-11 19:54:45.557126	4.409937242059644	101.4285565673718	True	None
		2025-05-11 18:06:53.654730	2.023665634266782	48.567975222402765	False	46.00
		2025-05-11 18:06:35.052096	4.459859242222674	240.83239908002437	False	None

Registrar Rendimiento...

Rendimiento Real (t):

Guardar

Registrar Rendimiento Real

Predicciones Realizadas

ID	Cultivo	Fecha	Rend. Hectárea (t/ha)	Rend. Total (t)	Fertilizante	Rend. Real (t)
4	Cultivo 2	2025-05-11 22:45:50.499417	4.459859242222674	240.83239908002437	False	235.00
3	Frijo 2	2025-05-11 19:54:45.557126	4.409937242059644	101.4285565673718	True	None
2	Cultivo 1	2025-05-11 18:06:53.654730	2.023665634266782	48.567975222402765	False	46.00
1	Cultivo 2	2025-05-11 18:06:35.052096	4.459859242222674	240.83239908002437	False	None

Éxito

Reporte generado: reporte_prediccion_4.pdf

Aceptar

Registrar Rendimiento Real

Generar Reporte

Volver

Reporte de Predicción	
Información del Cultivo	
Nombre	Cultivo 2
Tipo	Majz amarillo
Fecha de Siembra	2025-06-11
Riego	No
Información del Terreno	
Nombre	Terreno sur
Extensión (ha)	54
Tipo de Suelo	Duro
Estado	Durango
Municipio	Cuencame
Información de la Predicción	
Fecha	2025-05-11 22:45:50
Rendimiento por Hectárea (t/ha)	4.46
Rendimiento Total (t)	240.83
Uso de Fertilizante	No
Rendimiento Real (t)	235.00

Pruebas realizadas (unitarias, de integración, de usuario)

Pruebas Unitarias

Las pruebas unitarias se enfocaron en evaluar funciones individuales del sistema, verificando su correcto comportamiento ante entradas esperadas y no esperadas. Por ejemplo, se validó la función `validar_numeros` con entradas como "123" y "abc", esperando como resultado `True` y `False` respectivamente. De igual forma, se probó la función `validar_fecha` con fechas en formato correcto e incorrecto, como "2024-10-01" (válido) y "01-10-2024" (inválido), asegurando que el sistema pueda identificar correctamente los formatos esperados.

La función `cargar_datos_json` fue evaluada para garantizar que el archivo JSON correspondiente exista, esté correctamente formateado y que se carguen adecuadamente los datos de estados y municipios. Asimismo, se revisó la función `obtener_datos_terreno` utilizando tanto identificadores válidos como inválidos, observando si se retornaban los datos correctos o, en su defecto, valores nulos (`None`) cuando no se encontraba el terreno.

Para estas pruebas se recurrió principalmente a la ejecución manual desde consola, utilizando la impresión de mensajes mediante la función `print`, y a los cuadros de diálogo `messagebox` para corroborar el manejo adecuado de errores y excepciones.

Funciones que tras quedar correctamente implementadas no son visibles ante el usuario final, y la única manera de percibirlas es tras cometer un error en alguna validación. Estas se llevaron a cabo con la finalidad de elevar la calidad final del proyecto.

Pruebas de Integración

En las pruebas de integración se comprobó el correcto funcionamiento entre módulos interdependientes del sistema. Por ejemplo, durante el proceso de creación de cuenta, se validó que los datos se insertaron correctamente tanto en la tabla `login` como en la tabla `agricultor`, permitiendo posteriormente el inicio de sesión con las credenciales recién creadas.

En el registro de terrenos, se verificó que el sistema pudiera recuperar el ID del agricultor de forma dinámica y asociar correctamente los datos del terreno. Se observó que al finalizar el proceso, la información quedaba reflejada correctamente en la tabla visual (`TreeView`), confirmando que la base de datos y la interfaz gráfica estaban correctamente integradas.

Para la funcionalidad de predicción de cosechas, se comprobó la secuencia completa: selección de cultivo, obtención de datos del terreno y del tipo de cultivo, carga de los

modelos entrenados y cálculo de la predicción, finalizando con el almacenamiento de los resultados en la base de datos.

También se realizaron pruebas relacionadas con la integración con servicios externos y con fines totalmente estéticos, como OpenWeatherMap y NewsAPI, evaluando la correcta conexión, autenticación y visualización de los datos climáticos y noticiosos en la interfaz del usuario.

Pruebas de Usuario

Se llevaron a cabo pruebas funcionales desde la perspectiva del usuario final, ya sea mediante simulación manual o interacción directa de terceros. El flujo principal probado incluyó: ingreso a la pantalla de inicio, creación de cuenta, inicio de sesión, navegación por el dashboard, registro de terrenos y cultivos, realización de predicciones y finalmente la obtención de reportes en formato PDF.

Durante estas pruebas se observó que la interfaz gráfica era intuitiva y amigable, con botones bien distribuidos, colores contrastantes y mensajes informativos claros. Se validaron escenarios de error comunes, como campos vacíos, credenciales incorrectas y entradas no válidas, confirmando que el sistema reaccionaba adecuadamente mediante mensajes de advertencia o error.

A partir de estas pruebas se implementaron diversas mejoras: se fortaleció la validación de campos obligatorios, se adaptó la interfaz a resoluciones estándar para asegurar su correcta visualización en la mayoría de los dispositivos, y se utilizó el método `grab_set()` para implementar ventanas modales que evitan conflictos por múltiples formularios abiertos simultáneamente.

Herramientas utilizadas para la realización de las pruebas

Las pruebas se realizaron sobre un entorno desarrollado con Python, utilizando la librería Tkinter para la interfaz gráfica y Psycopg2 para la comunicación con una base de datos PostgreSQL.

Para la funcionalidad de predicción se empleó la biblioteca Scikit-learn, mientras que la validación de uso se completó mediante la simulación de distintos perfiles de usuario interactuando con la aplicación en diferentes escenarios.

RESULTADOS Y EVALUACIÓN

Comparación con los objetivos iniciales

A lo largo del desarrollo del proyecto, se establecieron objetivos generales y específicos que guiaron la construcción del sistema. A continuación, se presenta un análisis comparativo entre dichos objetivos y los logros alcanzados en la versión final del sistema:

Objetivo General	
Objetivo planteado	Resultado Obtenido
Desarrollar un sistema inteligente de predicción del rendimiento y calidad de cultivos, integrando datos agrícolas, climáticos y ambientales, para optimizar decisiones y recursos en el sector agrícola, con un diseño accesible y fácil de usar para todo tipo de productores.	Este objetivo se cumplió de manera satisfactoria. Se desarrolló un sistema funcional que permite realizar predicciones del rendimiento de cultivos utilizando modelos de aprendizaje automático. La integración de datos agrícolas (como tipo de cultivo, uso de riego y datos de terreno) junto con información climática (temperatura y precipitación) y ambiental (a través de APIs como OpenWeatherMap) permitió generar estimaciones útiles para los agricultores. La interfaz prioriza una experiencia intuitiva y clara para usuarios sin conocimientos técnicos, lo que responde a la intención de ser accesible para productores de distintos perfiles.
Objetivos específicos	
1. Recopilar y procesar datos agrícolas y ambientales, como históricos de producción, características del suelo, prácticas de cultivo y datos meteorológicos	Este objetivo se logró mediante la implementación de formularios para el registro de cultivos, terrenos, tipo de suelo y otros datos relevantes. Toda esta información es almacenada en una base

para integrarlos en una base de datos para análisis predictivo.	de datos PostgreSQL, lo cual permite su posterior uso en los modelos predictivos. Además, se estableció una relación entre cultivos, agricultores y terrenos, permitiendo un análisis más completo.
2. Desarrollar modelos predictivos basados en aprendizaje automático y análisis estadístico para identificar patrones y correlaciones entre variables climáticas y agronómicas, con el fin de predecir con alta precisión el rendimiento y la calidad de los cultivos bajo diferentes escenarios.	Se cumplieron los criterios fundamentales de este objetivo. Se entrenaron y utilizaron modelos de regresión lineal y Random Forest, los cuales fueron integrados exitosamente al sistema mediante la biblioteca joblib. Estos modelos permiten estimar el rendimiento por hectárea y determinar si se requiere fertilización, basados en variables clave como lluvia, temperatura y días estimados para la cosecha según la variedad del cultivo.

3. Diseñar e implementar una interfaz de usuario intuitiva y accesible, que facilite la visualización de predicciones, y reportes en un formato claro y accesible.	Cumplimiento: El sistema cuenta con una interfaz gráfica construida en Tkinter, la cual ofrece una navegación clara por medio de botones, etiquetas y paneles organizados por secciones. Se han incorporado tablas (TreeView), reportes visuales y mensajes informativos para facilitar la comprensión de los resultados, incluso por parte de usuarios sin formación técnica.
4. Integrar APIs de datos meteorológicos en tiempo real y otras fuentes de información ambiental para enriquecer la	Se integró exitosamente la API de OpenWeatherMap, permitiendo consultar y mostrar información climática en tiempo real con base en la ubicación del agricultor. También se integró NewsAPI para ofrecer noticias relacionadas con el

interfaz.	sector agrícola, enriqueciendo así la experiencia del usuario con información relevante y contextual.
5. Generar reportes que incluyan la información del cultivo, la información del terreno y la información de los resultados de la predicción.	El sistema permite generar reportes en PDF que incluyen los datos del cultivo, el terreno relacionado y los resultados de la predicción. Esta funcionalidad facilita el análisis posterior por parte del usuario y cumple con los requisitos de claridad y utilidad establecidos inicialmente.

Evaluación de Requerimientos Funcionales

ID	Descripción	Prioridad	Cumplimiento	Escala (1-5)	Observaciones
RF 01	Creación de cuentas si se llena el formulario	Alta	Si	5	Se crean las cuentas con éxito
RF 02	Inicio de sesion solo si coinciden las credenciales	Alta	Si	5	Se validan las credenciales
RF 03	Intuitiva y sencilla de usar	Baja	Parcial	3	Hace falta una numeración para saber donde empezar y acabar
RF 04	Carga de datos históricos de clima, suelo y cultivos	Alta	Si	5	Los datos se cargan correctamente
RF 05	Predicción según la región establecida	Alta	Si	5	La predicción funciona

RF 06	Generar reportes en PDF	Media	Si	5	Se generan y guardan los reportes correctamente
RF 07	PDF con información del cultivo, terreno y predicción	Media	Si	5	El PDF muestra la información como se debe
RF 08	La base de datos se diseña con integridad referencial	Alta	Si	4	La base de datos fue diseñada según las especificaciones
RF 09	Opción para ver el historial de predicciones	Alta	Si	5	Se puede observar el historial
RF 10	Historial de predicciones visible para el usuario	Media	Si	5	Los usuarios pueden ver sus propias predicciones
RF 11	Crear y eliminar terrenos, si se elimina se borra de la base de datos	Alta	Si	5	Se pueden crear y borrar terrenos sin problema
RF 12	Una messagebox confirma si hubo un error o si se efectuó un cambio en el sistema	Media	Si	5	La messagebox funciona en todo el sistema
RF 13	Se permite navegar por el sistema dentro de la aplicación	Baja	Si	5	Se navega por el sistema sin ningún problema
RF 14	Un Widget interactivo del clima en tiempo real	Media	Si	5	El menu tiene el Widget y se lee correctamente
RF 15	Integrar API para el clima en tiempo real	Alta	Si	5	El API funciona bien
RF 16	Un Widget con las	Media	Parcial	5	Se muestran

	noticias más relevantes en el sector agrícola				las noticias y al darle click puedes ir al enlace
RF 17	Integrar API de noticias en tiempo real	Media	Parcial	5	La API se integró correctamente
RF 18	Widget interactivo para el progreso del cultivo	Media	Parcial	4	Muestra la cantidad de cultivos que hay
RF 19	El perfil del usuario es modificable y visualizable	Media	No	1	Aún no se implementa
RF 20	Predicciones generadas en menos de 1 minuto	Baja	Si	5	El reporte se genera en menos de un minuto
RF 21	Las predicciones generadas tras llenar el formulario, se muestran en un messagebox inmediatamente	Alta	Si	5	La predicción se genera sin problema y muestra el messagebox inmediatamente
RF 22	El usuario podrá agregar el rendimiento real obtenido	Media	No	1	Aún no se implementa

Problemas encontrados y cómo se resolvieron

Problemas encontrados y cómo se resolvieron

Durante el desarrollo del sistema, surgieron diversos retos técnicos y organizativos que fueron superados mediante investigación, pruebas y trabajo colaborativo. A continuación, se describen los principales problemas encontrados y las estrategias implementadas para solucionarlos:

Problema 1 : Conexión a la base de datos PostgreSQL desde la interfaz gráfica

Durante las primeras etapas del desarrollo, se presentaron errores de conexión entre la aplicación en Tkinter y la base de datos PostgreSQL, especialmente relacionados con el puerto y las credenciales.

Solución: Se revisó detalladamente el archivo de configuración y se utilizó la librería psycopg2. Se implementó una clase exclusiva para la gestión de la conexión, encapsulando el manejo de errores con try-except. También se habilitó el puerto correcto (5433) en el servidor local y se ajustaron los permisos en pgAdmin para permitir conexiones desde la aplicación.

Problema 2: Fallos en la carga de modelos predictivos entrenados

Al integrar los modelos de aprendizaje automático con joblib, surgieron errores de incompatibilidad por diferencias en versiones de scikit-learn entre el entorno de entrenamiento y el de ejecución.

Solución: Se verificaron las versiones instaladas de las bibliotecas utilizadas y se estandarizó el entorno mediante la creación de un archivo requirements.txt. Además, se entrenaron de nueva cuenta los modelos utilizando la misma versión de scikit-learn que se empleó en el entorno del proyecto para evitar errores de deserialización.

Problema 3: Error en el uso de datos numéricos provenientes de entradas de texto

En múltiples secciones del sistema, los usuarios ingresaban datos numéricos como extensión del terreno, precipitaciones o temperatura. Estos datos eran tratados inicialmente como texto, lo que causaba errores al realizar operaciones matemáticas o al alimentar modelos predictivos.

Solución: Se incorporaron validaciones específicas para campos numéricos utilizando funciones como validatecommand en Tkinter, además de convertir los datos explícitamente a float o int antes de ser procesados por los modelos.

Problema 4: Mala visualización de la interfaz en diferentes resoluciones de pantalla

Durante las pruebas de usuario, se detectó que algunos componentes de la interfaz gráfica no se adaptaban correctamente en pantallas con resoluciones más pequeñas o mayores a la estándar.

Solución: Se ajustaron los tamaños de ventana y se centraron dinámicamente las pantallas modales. También se usaron contenedores con propiedades `expand` y `fill="both"` para mejorar la distribución de elementos en pantalla. Se probó el sistema en distintas resoluciones para asegurar una experiencia consistente.

Problema 5: Fallas al cargar imágenes o archivos externos (JSON, modelos, íconos)

En varias ocasiones, los archivos necesarios como imágenes, archivos JSON o los modelos .pkl no se encontraban al ejecutar el programa, especialmente si se cambiaba la ubicación del proyecto.

Solución: Se implementó el uso de rutas absolutas utilizando `os.path.join` y `file` para garantizar que el programa siempre localizara los archivos necesarios, sin depender de la carpeta desde la que se ejecutará. También se agregaron mensajes de error más informativos para facilitar el diagnóstico.

Problema 6: Dificultad para obtener datos climáticos reales desde la API

Inicialmente, las consultas a la API de OpenWeatherMap fallaban debido a la falta de una clave API válida y a errores en la construcción de la URL.

Solución: Se gestionó el registro en la plataforma OpenWeatherMap para obtener una API key gratuita. Se construyó la URL correctamente y se manejaron los posibles errores de red y formato de respuesta con bloques `try-except`. Además, se validó el campo de ciudad para evitar solicitudes vacías o mal escritas.

Problema: Redundancia de ventanas emergentes (Toplevel) en la interfaz

Durante la navegación del sistema, los usuarios podían abrir varias veces la misma ventana (por ejemplo, "Crear cuenta" o "Registrar terreno"), lo que generaba confusión y sobrecarga visual.

Solución: Se controló la creación de ventanas mediante una verificación previa: si una ventana ya estaba abierta, esta se traía al frente en lugar de abrir una nueva. Esto se logró iterando sobre los widgets hijos del root y revisando si ya existía una ventana con el mismo título.

Posibles mejoras futuras

El proyecto presenta diversas oportunidades de mejora para aumentar su escalabilidad, precisión y funcionalidad. Entre las más importantes se incluyen:

- Integración con sensores IoT: Conectar sensores de humedad, temperatura y pH del suelo para recopilar datos en tiempo real y ajustar las predicciones según el entorno.
- Recomendaciones inteligentes: Ampliar la IA para sugerir cultivos óptimos según ubicación, clima, tipo de suelo y temporada, usando aprendizaje automático.
- Análisis económico: Calcular costos estimados y ganancias esperadas para facilitar decisiones comerciales más informadas.
- Enciclopedia agrícola: Incluir una base de datos educativa con información sobre cultivos, cosechas y plagas para apoyo a los agricultores.
- App móvil complementaria: Crear una versión móvil para acceso fácil desde el campo, incluso sin conexión constante.
- Modo colaborativo: Permitir la colaboración anónima entre usuarios para mejorar los modelos de predicción sin necesidad de registro.
- Alertas automatizadas: Enviar notificaciones por SMS o correo sobre eventos críticos como lluvias, heladas o requerimientos de fertilización.
- Visualización avanzada: Añadir gráficos, mapas de calor, vistas satelitales y dashboards personalizables para mejorar la experiencia del usuario.
- Multilenguaje y accesibilidad: Ofrecer soporte en varios idiomas y facilitar el uso para personas con discapacidades.
- Red de colaboración: Crear una plataforma para que agricultores, investigadores y productores puedan comunicarse y colaborar en proyectos.

CONCLUSIONES Y RECOMENDACIONES

Reflexión sobre la experiencia del desarrollo

El desarrollo del sistema representó una experiencia formativa significativa, no solo desde el punto de vista técnico, sino también en términos de crecimiento personal y profesional. A lo largo del proyecto, fue necesario combinar conocimientos adquiridos en diversas materias como programación orientada a objetos, bases de datos, análisis de datos, interfaces gráficas y aprendizaje automático. Esta integración de saberes permitió comprender de forma más completa cómo convergen distintas áreas para crear una solución funcional y útil.

Uno de los aspectos más enriquecedores fue enfrentarse a problemas reales y tener que idear soluciones prácticas bajo restricciones de tiempo, recursos y conocimientos. En

lugar de trabajar con ejemplos controlados de laboratorio, se abordaron situaciones similares a las que se presentan en el desarrollo de software profesional: manejo de errores, conexión con servicios externos, validación de datos introducidos por el usuario, entrenamiento de modelos predictivos y diseño de interfaces amigables. Esto permitió fortalecer la capacidad de análisis, adaptación y resolución de problemas.

Otro elemento destacado fue la importancia de la organización del código y la documentación del proyecto. A medida que el sistema crecía en complejidad, resultó evidente la necesidad de modularizar correctamente, mantener buenas prácticas de nomenclatura y realizar pruebas constantes para asegurar la estabilidad. Esta experiencia consolidó habilidades relacionadas con el trabajo disciplinado, la planificación de tareas y la autoevaluación.

Además, se aprendió a valorar el enfoque centrado en el usuario. Desde el diseño de pantallas hasta la generación de reportes, fue necesario pensar desde la perspectiva del productor agrícola, alguien que probablemente no tiene formación técnica, pero necesita una herramienta confiable, clara y fácil de utilizar. Esto implicó un ejercicio de empatía y simplificación, elementos claves para cualquier desarrollador de soluciones tecnológicas.

Lecciones aprendidas

Para todos los miembros del equipo, este proyecto representó un punto de inflexión, ya que fue la primera vez que tuvimos que trabajar de manera verdaderamente estructurada y metódica en el tiempo que llevamos en la carrera. A diferencia de proyectos anteriores, que solíamos desarrollar de forma más informal, sin evidencias claras, sin una planificación detallada y sin un esquema de responsabilidades bien definido, esta experiencia nos obligó a adaptarnos a un enfoque mucho más disciplinado.

El proceso nos llevó a comprender la importancia de documentar cada paso, coordinar esfuerzos y anticipar posibles obstáculos para alcanzar un objetivo común. Sin embargo, a pesar del esfuerzo, el tiempo resultó ser uno de nuestros mayores desafíos, impidiendo que muchas de las ideas y mejoras que inicialmente planeamos se concretaran en el producto final. Esta limitación nos hizo reflexionar como individuos y como equipo, llevándonos a extraer como tal, tres lecciones fundamentales que resumen nuestro aprendizaje y que consideramos cruciales para enfrentar futuros proyectos con mayor éxito.

La comunicación como pilar fundamental en la realización de un proyecto: Desarrollar este proyecto resultó más complicado de lo que realmente debería haber sido, no por falta de habilidad o compromiso, sino por desafíos en la comunicación dentro del equipo. Aunque las relaciones personales entre los miembros eran positivas, esto no siempre se tradujo en una coordinación efectiva. La capacidad para expresar ideas con claridad, manejar diferencias de opinión y alinear expectativas sobre el nivel de

compromiso fueron factores que impactaron el proceso. Estas barreras comunicativas no solo generaron malentendidos, sino que también dificultaron la toma de decisiones rápidas y precisas, afectando directamente el resultado final del proyecto.

Importancia de una correcta documentación: Uno de los desafíos más inesperados que enfrentamos fue la falta de documentación detallada sobre proyectos similares. Aunque existen muchos software de este tipo en el mercado, acceder a la información técnica y los procesos que los respaldan es sorprendentemente difícil. Es sencillo analizar el producto final, pero comprender a profundidad cada etapa necesaria para llegar a ese resultado, especialmente al desglosar ideas complejas para que sean accesibles a estudiantes, fue más complicado de lo que esperábamos. Esto nos hizo valorar la importancia de una documentación precisa y confiable para futuros proyectos, ya que dedicar tiempo a este aspecto es fundamental para lograr resultados de calidad y facilitar el aprendizaje para quienes continúen con este trabajo.

La importancia de planificar: Aunque a menudo subestimamos la planificación de tareas, considerándola innecesaria o incluso una pérdida de tiempo, es, después de la comunicación dentro del equipo, uno de los factores más importantes para el éxito de cualquier proyecto. En esta fase es donde se definen las bases para un desarrollo organizado y eficiente. Sin una planificación adecuada, es fácil desviarse en tareas secundarias, subestimar el tiempo necesario para funciones críticas o incluso pasar por alto aspectos esenciales del proyecto. Además, una buena planificación permite anticipar riesgos, optimizar el uso de recursos y coordinar mejor al equipo, lo que resulta en un producto final más completo, de mayor calidad y alineado con las expectativas del cliente.

BIBLIOGRAFÍAS Y ANEXOS

- Innovatiana. (2023, noviembre 22). *15 conjuntos de datos para revolucionar la agricultura con IA*. <https://es.innovatiana.com/post/ai-datasets-for-agriculture>
- Alliance Bioversity International y CIAT. (2023, agosto 9). *Inteligencia artificial: ¿cómo podría transformar la agricultura?* <https://alliancebioversityciat.org/es/stories/inteligencia-artificial-como-podria-transformar-agricultura>
- Blasstech. (s.f.). *Machine learning en la agricultura: Transformando el futuro del agro con inteligencia artificial*. <https://blasstech.cl/machine-learning-en-la-agricultura-transformando-el-futuro-del-agro-con-inteligencia-artificial/>
- EOS Data Analytics. (s.f.). *Software de agricultura de precisión y más soluciones digitales*. <https://eos.com/es/products/crop-monitoring/key-functions/>

- Red Hat. (s.f.). *Alta disponibilidad (HA): ¿Qué es y cómo funciona?*
<https://www.redhat.com/es/topics/linux/what-is-high-availability>
- Autor desconocido. (s.f.). *Inteligencia artificial aplicada a agricultura de precisión: Control de hongos en tomate* [PDF]. Scribd.
<https://ro.scribd.com/document/675076180/Inteligencia-artificial-aplicada-a-agricultura-precision-control-hongos-tomate>
- Evolbit. (s.f.). *¿Software agrícola? Te guiamos.*
<https://blog.evolbit.net/software-agricola-te-guiamos/>
- Mountain Goat Software. (s.f.). *The Scrum Team and Scrum of Scrums.*
<https://www.mountaingoatsoftware.com/agile/scrum/roles/team>
- Atlassian. (s.f.). *Product backlog: Definition & prioritization tips.*
<https://www.atlassian.com/agile/scrum/backlogs>
- Atlassian. (s.f.). *Scrum sprints: Everything you need to know.*
<https://www.atlassian.com/agile/scrum/sprints>
- Atlassian. (s.f.). *Sprint to success with Jira Scrum boards.*
<https://www.atlassian.com/software/jira/features/scrum-boards>
- Atlassian. (s.f.). *What is sprint planning? [+How to get started].*
<https://www.atlassian.com/agile/scrum/sprint-planning>
- Mountain Goat Software. (s.f.). *What is a product owner in Scrum?*
<https://www.mountaingoatsoftware.com/agile/scrum/roles/product-owner>
- Creately. (s.f.). *Caso de uso: Iniciar sesión.*
<https://creately.com/diagram/example/in6bj7z31/caso-de-uso-iniciar-sesion-classic>
- Visual Paradigm. (s.f.). *Diagrama de caso de uso.*
<https://online.visual-paradigm.com/es/community/share/untitled-15d20y75kw>
- Boardmix. (s.f.). *Ejemplos de casos de uso.*
<https://boardmix.com/es/articles/use-case-examples/>
-
- Schwaber, K., & Sutherland, J. (2020). *Guía Scrum: La guía definitiva para Scrum: Las reglas del juego.* Scrum.org.
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish.pdf>