



Gestión de proyecto para el sistema de predicción de producción de árboles frutales con inteligencia artificial

Alumno: Alvaro Marchant Concha
Correo: Alvaro.marchant@usm.cl

Profesor Guía: Ricardo Cahe
Profesor Correferente: Rodrigo Pinochet



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título Tesis de Postgrado

Título del trabajo: Gestión de proyecto para el sistema de predicción de producción de árboles frutales con inteligencia artificial

Nombre del candidato(a): Álvaro Nicolas Marchant Concha

Carrera / Grado: Ingeniería en Informática

Campus: Sede Viña del Mar **Departamento:** Electrónica e Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Ricardo Cahé C, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.


El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

6 meses 12 meses 2 años 3 años 5 años 10 años


Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 25/02/2026 Firma: 

Estudiante o Candidato(a):

Fecha: 25/02/2026 Firma: 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.



Tabla de contenido

1. Introducción	4
1.1 Contexto y justificación del tema	4
1.2 Especificación de requisitos ágiles mediante Historias de Usuario.....	4
1.3 Relevancia de las Metodologías Ágiles en proyectos de software.....	7
1.4 Planteamiento del problema	7
1.5 Objetivo General y Específicos de la Tesina.....	8
2. Metodología y plan de trabajo.....	8
2.1 Enfoque para la implementación de Metodologías Ágiles en el proyecto	8
2.2 Herramientas de gestión empleadas.....	9
2.3 Plan de personal	9
2.4 Planificación temporal del proyecto	9
3. Marco Conceptual	10
3.1 Principios y Valores del Manifiesto Ágil	10
3.2 Descripción de Metodologías Ágiles.....	11
4. Planificación de Sprints.....	13
4.1 Definición de Sprints y ciclos de desarrollo	13
4.2 Estimación de tareas y asignación de recursos.....	16
5. Ejecución de reuniones ágiles	17
5.1 Reuniones diarias	17
5.2 Sprint Planning	17
5.3 Sprint Review	18
5.4 Sprint Retrospective	18
6. Gestión de cambios y mejora continua	19
6.1 Identificación y registro de cambios	19
6.2 Aplicación de mejoras basadas en feedback	22
6.3 Control de cambios e impacto	22
7. Evaluación del desempeño ágil.....	23
7.1 Concepto y cálculo de velocity.....	23
7.2 Análisis de velocity y tendencias.....	23



8. Desafíos y soluciones implementadas.....	24
8.1 Desafíos técnicos	24
8.2 Desafío de tiempo.....	25
8.3 Tabla de riesgos	25
9. Estimación de costos	26
9.1 Costos de recursos humanos.....	26
9.2 Costos de herramientas, infraestructura y otros.....	27
10. Conclusiones	27
10.1 Cumplimiento de objetivos.....	27
10.2 Validación de metodologías ágiles	28
10.3 Análisis de éxito	29
10.4 Lecciones aprendidas	29
10.5 Reflexión final.....	29
Referencias	31



Agradecimientos: A mis amigos y miembros del proyecto Matías Aguayo, Bastián Berríos, Sebastián Lagos y Javier Lucero, ya que sin ellos no se hubiera podido llevar a cabo el proyecto de forma tan amena y entretenida.

A mi familia, en especial a mi madre, por apoyarme en todo lo que podía y soportarme siempre.

A mis amigos del colegio, por siempre estar ahí para despejarse jugando o saliendo.

A mis amigas y amigos que conocí este año, ya que, a pesar de conocerme durante poco tiempo, igualmente me han ayudado de muchas maneras.

Y finalmente a mi gato Tomy, que dejó este mundo en este último tramo del año, gracias por ser un gato regalón y traer felicidad a nuestra familia.

Resumen

El presente trabajo tiene como objetivo documentar la gestión de proyecto para el desarrollo de AgroPredict, un sistema predictivo agrícola web orientado a gerentes y encargados de empresas agrícolas. El proyecto busca reducir la incertidumbre en la toma de decisiones agrícolas mediante la predicción del rendimiento de cultivos frutales basándose en variables como región, hectáreas de terreno, época del año y tipo de árbol frutal. Para ello se implementará una metodología ágil Scrum que permita entregas incrementales y adaptación a cambios en los requerimientos. El sistema fue desarrollado en colaboración con la empresa Neering, utilizando tecnologías como Django y modelos de machine learning para generar predicciones de producción a corto, mediano y largo plazo, presentadas mediante gráficos y visualizaciones intuitivas.

Palabras Clave: Gestión de proyectos, Metodología ágil, Sistema predictivo, Agricultura, Scrum



1. Introducción

1.1 Contexto y justificación del tema

El sector agrícola en Chile enfrenta desafíos significativos relacionados con la incertidumbre en la producción, especialmente ante los cambios climáticos y la necesidad de optimizar la planificación de recursos. Las empresas agrícolas requieren herramientas que les permitan tomar decisiones informadas sobre la producción esperada, la asignación de recursos como agua, tiempo y logística, y la planificación estratégica a largo plazo.

En este contexto, el desarrollo de AgroPredict surge como una respuesta tecnológica innovadora que busca proporcionar predicciones precisas sobre el rendimiento de cultivos frutales. El proyecto representa una oportunidad de aplicar metodologías ágiles de gestión de proyectos en un entorno real de colaboración empresa-universidad, específicamente entre los estudiantes de ingeniería en informática y la empresa Neering.

1.2 Especificación de requisitos ágiles mediante Historias de Usuario

Como parte integral de la implementación de la metodología Scrum en el desarrollo de AgroPredict, previamente al comienzo del ciclo de trabajo, el equipo definió detalladamente las historias de usuario con criterios de aceptación específicos. Esta especificación responde tanto a los principios fundamentales del desarrollo ágil como a la necesidad de establecer un marco claro y medible para la evaluación del éxito del proyecto.

Las historias de usuario constituyen una herramienta fundamental en el marco Scrum para capturar los requisitos funcionales desde la perspectiva del usuario final. En el contexto de AgroPredict, estas historias representan las necesidades específicas de gerentes y encargados de empresas agrícolas que buscan optimizar sus procesos de toma de decisiones mediante predicciones precisas sobre el rendimiento de cultivos frutales.

Se presentan siete historias de usuario principales identificadas para el desarrollo de AgroPredict, cada una con sus respectivos criterios de aceptación:

Historia de Usuario 1: Ingreso de datos de cultivo

Como gerente de empresa agrícola, quiero ingresar los datos básicos de mi cultivo (región, hectáreas, época del año, tipo de árbol frutal), para obtener predicciones personalizadas sobre la producción esperada.

Criterios de Aceptación:

- DADO que estoy en la pantalla de ingreso de datos, CUANDO selecciono una región de la lista desplegable, ENTONCES el sistema debe mostrar solo las opciones válidas para Chile
- DADO que ingreso la cantidad de hectáreas, CUANDO introduzco un valor numérico válido, ENTONCES el sistema debe aceptar y validar el dato
- DADO que selecciono la época del año, CUANDO escojo un mes específico, ENTONCES el sistema debe considerar las variables climáticas estacionales correspondientes
- DADO que selecciono el tipo de árbol frutal, CUANDO elijo una opción del catálogo disponible, ENTONCES el sistema debe cargar los parámetros específicos de ese cultivo



- DADO que he completado todos los campos requeridos, CUANDO presiono "Generar Predicción", ENTONCES el sistema debe calcular la proyección usando el modelo de IA

Historia de Usuario 2: Visualización de predicciones

Como encargado de planificación agrícola, quiero visualizar las predicciones de producción en gráficos y tablas interactivos, para analizar fácilmente los resultados proyectados.

Criterios de Aceptación:

- DADO que se ha generado una predicción, CUANDO accedo a la pantalla de resultados, ENTONCES debo ver un gráfico de barras con la producción esperada por periodo
- DADO que visualizo el gráfico principal, CUANDO hago pongo el ratón sobre una barra, ENTONCES debe mostrar el valor exacto de producción estimada
- DADO que estoy en la vista de resultados, CUANDO selecciono la pestaña de tabla, ENTONCES debo ver los datos numéricos organizados por período (corto, mediano, largo plazo)
- DADO que tengo múltiples predicciones, CUANDO activo la comparación, ENTONCES el sistema debe mostrar gráficos superpuestos de diferentes escenarios
- DADO que visualizo los resultados, CUANDO hago clic en "Exportar", ENTONCES debo poder descargar los datos en formato PDF o Excel

Historia de Usuario 3: Gestión de sesiones de usuario

Como usuario del sistema, quiero tener una cuenta personal con login seguro, para acceder a mis predicciones guardadas y mantener un historial personalizado.

Criterios de Aceptación:

- DADO que soy un usuario registrado, CUANDO ingreso mis credenciales correctas, ENTONCES el sistema debe permitirme acceder al dashboard principal
- DADO que mis credenciales son incorrectas, CUANDO intento iniciar sesión, ENTONCES el sistema debe mostrar un mensaje de error específico
- DADO que he iniciado sesión exitosamente, CUANDO navegue por el sistema, ENTONCES mi sesión debe mantenerse activa por el tiempo establecido
- DADO que no tengo actividad por un período determinado, CUANDO expire la sesión, ENTONCES el sistema debe cerrar la sesión automáticamente y redirigirme al login
- DADO que deseo cerrar sesión, CUANDO hago clic en "Cerrar Sesión", ENTONCES el sistema debe limpiar mi información de sesión y redirigirme a la página principal

Historia de Usuario 4: Historial de predicciones

Como gerente agrícola, quiero acceder al historial de todas mis predicciones anteriores, para comparar resultados y tomar decisiones informadas basadas en datos históricos.

Criterios de Aceptación:

- DADO que tengo predicciones guardadas, CUANDO accedo a la sección "Historial", ENTONCES debo ver una lista cronológica de todas mis predicciones anteriores



- DADO que visualizo el historial, CUANDO hago clic en una predicción específica, ENTONCES debo poder ver todos los detalles de esa proyección anterior
- DADO que tengo múltiples predicciones, CUANDO uso los filtros de búsqueda, ENTONCES debo poder filtrar por fecha, tipo de cultivo o región
- DADO que selecciono múltiples predicciones del historial, CUANDO activo la función de comparación, ENTONCES el sistema debe mostrar un análisis comparativo visual
- DADO que quiero eliminar predicciones antiguas, CUANDO selecciono elementos del historial y confirmo la eliminación, ENTONCES el sistema debe remover permanentemente esos registros

Historia de Usuario 5: Integración de datos climáticos en tiempo real

Como usuario del sistema, quiero que las predicciones incorporen datos climáticos actuales de AccuWeather, para obtener proyecciones más precisas basadas en condiciones meteorológicas reales.

Criterios de Aceptación:

- DADO que ingreso los datos de mi cultivo, CUANDO el sistema procesa la información, ENTONCES debe consultar automáticamente la API de AccuWeather para la región seleccionada
- DADO que se obtienen datos climáticos, CUANDO el sistema los integra al modelo, ENTONCES debe mostrar un indicador de que se están usando datos meteorológicos actuales
- DADO que la API de AccuWeather no está disponible, CUANDO se intenta generar una predicción, ENTONCES el sistema debe mostrar una advertencia al usuario
- DADO que las condiciones climáticas son extremas, CUANDO se genera la predicción, ENTONCES el sistema debe mostrar alertas sobre posibles impactos en la producción

Historia de Usuario 6: Dashboard principal de monitoreo

Como gerente de operaciones agrícolas, quiero tener un dashboard central que muestre un resumen de mis cultivos y predicciones activas, para monitorear el estado general de mis proyecciones de producción.

Criterios de Aceptación:

- DADO que accedo al dashboard principal, CUANDO la página carga, ENTONCES debo ver un resumen de mis predicciones más recientes en formato de tarjetas
- DADO que tengo múltiples cultivos registrados, CUANDO visualizo el dashboard, ENTONCES debe mostrar indicadores de rendimiento por tipo de cultivo
- DADO que hay predicciones con alertas, CUANDO accedo al dashboard, ENTONCES debo ver notificaciones destacadas sobre cultivos que requieren atención
- DADO que quiero ver detalles específicos, CUANDO hago clic en cualquier tarjeta del dashboard, ENTONCES debo navegar a la vista detallada de esa predicción



- DADO que el dashboard contiene información actualizada, CUANDO realizo cambios en otras secciones, ENTONCES el dashboard debe reflejar automáticamente las actualizaciones

Historia de Usuario 7: Análisis comparativo de cultivos

Como planificador agrícola, quiero comparar diferentes escenarios de cultivo (distintas regiones, épocas o tipos de árbol), para optimizar mis decisiones de inversión y planificación.

Criterios de Aceptación:

- DADO que tengo múltiples predicciones generadas, CUANDO accedo a la herramienta de comparación, ENTONCES debo poder seleccionar hasta 4 predicciones para comparar simultáneamente
- DADO que he seleccionado predicciones para comparar, CUANDO genero el análisis, ENTONCES el sistema debe mostrar gráficos lado a lado con los resultados comparativos
- DADO que visualizo la comparación, CUANDO reviso los resultados, ENTONCES debo ver métricas clave como ROI estimado, riesgo relativo y periodo de recuperación
- DADO que quiero entender las diferencias, CUANDO hago clic en "Análisis Detallado", ENTONCES el sistema debe mostrar factores específicos que influyen en cada predicción

1.3 Relevancia de las Metodologías Ágiles en proyectos de software

Las metodologías ágiles han demostrado ser fundamentales en el desarrollo de proyectos de software por diversas razones:

- Flexibilidad y Adaptación: Permiten adaptarse rápidamente a cambios en los requisitos, lo que es crucial en un entorno tecnológico en constante evolución donde las necesidades del cliente pueden variar.
- Colaboración Continua: Fomentan la comunicación constante entre equipos de desarrollo y partes interesadas, asegurando que todos estén alineados y que el producto final cumpla con las expectativas.
- Entrega Incremental: Promueven la entrega de incrementos funcionales del software en ciclos cortos (sprints), permitiendo a los clientes recibir valor desde etapas tempranas del proyecto.
- Mejora Continua: Las retrospectivas al final de cada ciclo permiten al equipo reflexionar sobre su proceso y hacer ajustes, llevando a un aprendizaje constante y mejora en la calidad del trabajo.

1.4 Planteamiento del problema

El principal problema que se busca abordar es la falta de herramientas predictivas precisas y accesibles para la toma de decisiones en el sector agrícola chileno. Los gerentes y encargados de empresas agrícolas carecen de sistemas que les permitan:

- Predecir con precisión el rendimiento esperado de sus cultivos frutales.
- Optimizar la planificación de recursos basándose en proyecciones confiables.
- Tomar decisiones estratégicas informadas sobre la producción a largo plazo.
- Visualizar datos complejos de manera intuitiva y comprensible.



1.5 Objetivo General y Específicos de la Tesina

Objetivo General:

Gestionar el desarrollo de un sistema predictivo agrícola mediante la implementación de metodologías ágiles, específicamente Scrum, maximizando la eficiencia del equipo y la alineación con las necesidades del cliente para entregar un producto mínimo viable funcional.

Objetivos Específicos:

- Implementar la metodología Scrum en el contexto del desarrollo de AgroPredict, adaptando sus ceremonias y artefactos a las particularidades del proyecto y las restricciones académicas.
- Optimizar la gestión de recursos y tiempos mediante la planificación estratégica de sprints, la definición clara de roles y responsabilidades, y el establecimiento de procesos eficientes de comunicación con el cliente.
- Facilitar la colaboración efectiva entre el equipo de desarrollo y la empresa Neering, estableciendo canales de comunicación claros y procesos de retroalimentación continua.
- Evaluar el impacto de las metodologías ágiles en la calidad de los entregables, la satisfacción del cliente y la eficiencia del desarrollo a través de métricas específicas y análisis de resultados.
- Documentar las lecciones aprendidas y mejores prácticas identificadas durante la gestión del proyecto, proporcionando un marco de referencia para futuros proyectos similares.

2. Metodología y plan de trabajo

2.1 Enfoque para la implementación de Metodologías Ágiles en el proyecto

Para el desarrollo de AgroPredict, se ha seleccionado la metodología Scrum en su última versión oficial (año 2020) como marco principal de trabajo, complementada con elementos de Kanban para la visualización del flujo de tareas. Esta elección se fundamenta en:

Justificación de Scrum:

- Estructura iterativa: Los sprints de 3-4 semanas se alinean perfectamente con los hitos académicos establecidos por la universidad.
- Responsabilidades definidas: La clara definición de Scrum Master, Product Owner y desarrolladores facilita la organización del equipo estudiantil.
- Ceremonias adaptables: Las reuniones pueden ajustarse al horario académico y la disponibilidad del equipo.
- Enfoque en el valor: Permite priorizar funcionalidades críticas para entregar valor temprano al cliente.

Justificación de usar la última versión (2020):

- Product Goal como compromiso: En otras versiones de scrum se le daba más valor al sprint como tal, en esta versión se introduce el concepto de Product Goal como un objetivo a largo plazo, lo que se alinea perfectamente con el enfoque en entregar valor claro a Neering.



- Estructura de equipo más clara: Elimina la jerarquía de “equipo desarrollador y otros equipos”, quedando solamente un “equipo scrum” más unido.
- Equipo autogestionado: El equipo decide quién, cómo y qué trabajar. Se les da más autonomía y responsabilidad sobre el alcance del trabajo seleccionado.

Adaptaciones Específicas:

- Sprints alineados con hitos universitarios: Tres sprints principales coincidentes con las entregas académicas.
- Integración con cliente: Reuniones semanales con Neering para feedback y validación continua.

2.2 Herramientas de gestión empleadas

Gestión de proyectos:

- ClickUp: Para la gestión del backlog y visualización de tareas en formato Kanban.

Comunicación y colaboración:

- Discord/Microsoft Teams: Para reuniones diarias y comunicación asíncrona del equipo.
- Correo electrónico: Para comunicaciones formales con Neering.
- Figma: Para colaboración en diseño de interfaces con el equipo de UX de Neering.

Control de versiones y desarrollo:

- Git/GitHub: Para control de versiones y colaboración en código.
- AWS/Railway: Infraestructura para despliegue.

2.3 Plan de personal

Estructura del equipo (5 integrantes):

- Scrum Máster: Responsable de facilitar el proceso Scrum, eliminar impedimentos y asegurar el cumplimiento de las ceremonias ágiles.
- Diseñador UI/UX y Product Owner: Encargado del diseño de interfaces y experiencia de usuario, trabajando en colaboración con el equipo de diseño de Neering, además de ser el encargado del enlace entre el equipo y el cliente.
- Arquitecto de Software: Responsable del diseño y arquitectura técnica del sistema, definición de tecnologías y estructura del proyecto.
- Ingeniero de Datos: Encargado de la recopilación, procesamiento y modelado de datos agrícolas, desarrollo del sistema de predicción.
- Desarrollador Full-Stack: Desarrollador con rol flexible de apoyo en frontend y backend según las necesidades del sprint.

2.4 Planificación temporal del proyecto

Sprint 1 (25 de agosto - 29 de agosto)

- Objetivo: MVP funcional básico
- Duración: 1 semana



- Entregables: Login, predicción mock, prototipo visual validado

Sprint 2 (1 de septiembre - 26 de septiembre)

- Objetivo: MVP mejorado con feedback aplicado
- Duración: 4 semanas
- Entregables: Modelo predictivo mejorado, visualizaciones, dashboard inicial, flujo de ingreso de datos

Sprint 3 (29 de septiembre - 24 de octubre)

- Objetivo: MVP final para presentación
- Duración: 4 semanas
- Entregables: Sistema completo, pitch preparado, stand diseñado

3. Marco Conceptual

3.1 Principios y Valores del Manifiesto Ágil

El manifiesto ágil fue creado a principios del año 2001, reuniendo a personas para debatir el futuro del desarrollo de software, de aquí nacieron los 4 valores y 12 principios que constituyen la base filosófica de todas las acciones y adaptaciones ágiles que se implementan en AgroPredict:

- Satisfacción al cliente: En AgroPredict se implementó mediante entregas incrementales en cada sprint, comenzando con un MVP básico en el Sprint 1 que ya proporcionaba valor a Neering con una página inicial, permitiendo feedback temprano y ajustes continuos
- Cambios en los requerimientos: El equipo adaptó los requerimientos según el feedback de Neering, como la integración de la API de AccuWeather que no estaba inicialmente planificada
- Entregas frecuentes: AgroPredict implementó sprints de 3-4 semanas con entregas funcionales al final de cada uno: MVP básico (Sprint 1), MVP mejorado con retroalimentación (Sprint 2), y sistema completo (Sprint 3)
- Colaboración diaria: Se establecieron reuniones semanales entre el equipo Treering y Neering, además de comunicación diaria vía Discord con el equipo
- Colaboradores motivados: Cada miembro del equipo recibió roles específicos según sus fortalezas, con autonomía para ejecutar sus tareas
- Comunicación cara a cara: Se realizaron reuniones presenciales en las oficinas de Neering
- Medición de avances: El progreso de AgroPredict se midió por funcionalidades desplegadas y demostrables tales como el sistema de login, flujo de ingreso de datos, modelo predictivo funcionando, y visualizaciones interactivas
- Promover ritmo sostenido: Se estableció un ritmo de trabajo sostenible considerando las responsabilidades académicas del equipo, con reuniones diarias cortas, y sprints alineados con el calendario universitario
- Atención a la excelencia: El equipo está enfocado principalmente en la funcionalidad del proyecto, sin dejar de lado cosas importantes como la experiencia de usuario



- Simplicidad esencial: El equipo se enfocó en funcionalidades esenciales para el MVP: ingreso de datos, predicción básica y visualización simple, evitando características complejas que no agregaran valor inmediato al usuario final
- Equipos autoorganizados: El equipo Treering tomó decisiones técnicas de forma autónoma, como la elección de tecnologías (Django, React)
- Mejora continua: Al final de cada sprint se realizaron retrospectivas para identificar mejoras: optimización de comunicación con Neering, mejor distribución de tareas, y ajustes en la planificación para el siguiente sprint

3.2 Descripción de Metodologías Ágiles

Scrum - Metodología principal:

Scrum es un marco de trabajo para desarrollo ágil de software que se ha expandido a otras industrias. Se adopta una estrategia de desarrollo incremental en lugar de la planificación y ejecución completa del producto desde el inicio. Se solapan las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada y se trabaja con ciclos de ejecución muy cortos (sprints) entre una y cuatro semanas para entregar valor de forma rápida y continua.

En AgroPredict, Scrum se implementó como el marco principal de trabajo, adaptándose a las restricciones académicas del equipo y a la colaboración con Neering como empresa patrocinadora.

Sprint:

Un sprint es un período breve de tiempo fijo (entre una y cuatro semanas) en el que un equipo Scrum trabaja para completar una cantidad específica de trabajo establecida. Los sprints constituyen el corazón de las metodologías Scrum y ágil. Cada sprint tiene una meta específica que debe alcanzarse. Durante el sprint, el alcance no puede ser modificado para proteger al equipo.

Product Backlog:

El Product Backlog es una lista emergente, ordenada y priorizada de todo lo que se conoce que es necesario para el producto. Es la única fuente de requisitos para cualquier cambio a realizarse en el producto y constituye el plan de trabajo maestro del proyecto.

Sprint Backlog:

El Sprint Backlog es el conjunto de elementos del Product Backlog seleccionados para el sprint, más un plan de trabajo para entregar el incremento del producto y lograr el objetivo del sprint. Es un pronóstico del equipo de desarrollo sobre qué funcionalidad estará disponible en el próximo incremento.

Sprint Planning:

El Sprint Planning es una reunión especial donde el equipo Scrum decide qué elementos del Product Backlog abordará y cómo los llevará a cabo durante el próximo sprint. El objetivo es crear un Product Backlog del Sprint que incluya las tareas necesarias, un plan de ejecución y un objetivo de sprint claro y por escrito.

Sprint Review:



La Sprint Review es un evento fundamental dentro del marco de trabajo Scrum que se desarrolla al cierre de cada ciclo de iteración, justo antes de realizar la retrospectiva del equipo. Durante esta sesión, el equipo trabaja juntamente con los actores clave de la organización (representantes de diversos departamentos, clientes finales, usuarios y otros interesados) para examinar el incremento de producto generado y en algunos casos, realizar ajustes en la lista priorizada de funcionalidades pendientes.

Sprint Retrospective:

La retrospectiva de sprint es una sesión de trabajo que lleva a cabo el equipo de desarrollo durante la metodología ágil, realizándose una vez que finaliza cada ciclo iterativo. Su propósito es evaluar la manera en que transcurrió el flujo de trabajo y elaborar estrategias de mejora aplicables al próximo sprint. El enfoque principal radica en examinar qué aspectos resultaron exitosos, cuáles presentaron dificultades, y determinar iniciativas concretas para potenciar la interacción entre integrantes, optimizar los procedimientos implementados y mejorar los recursos tecnológicos disponibles, manteniendo distancia del análisis sobre los logros específicos del producto.

Kanban - Metodología Complementaria

Kanban es una metodología ágil visual para gestionar el trabajo conforme avanza a través de un proceso. Visualiza tanto el proceso como el trabajo real que pasa por él, con el objetivo de identificar cuellos de botella y optimizar el flujo. Se utiliza un tablero visual (ver figura 1) con columnas que representan las diferentes etapas del proceso (Pendiente, En curso, Completado).

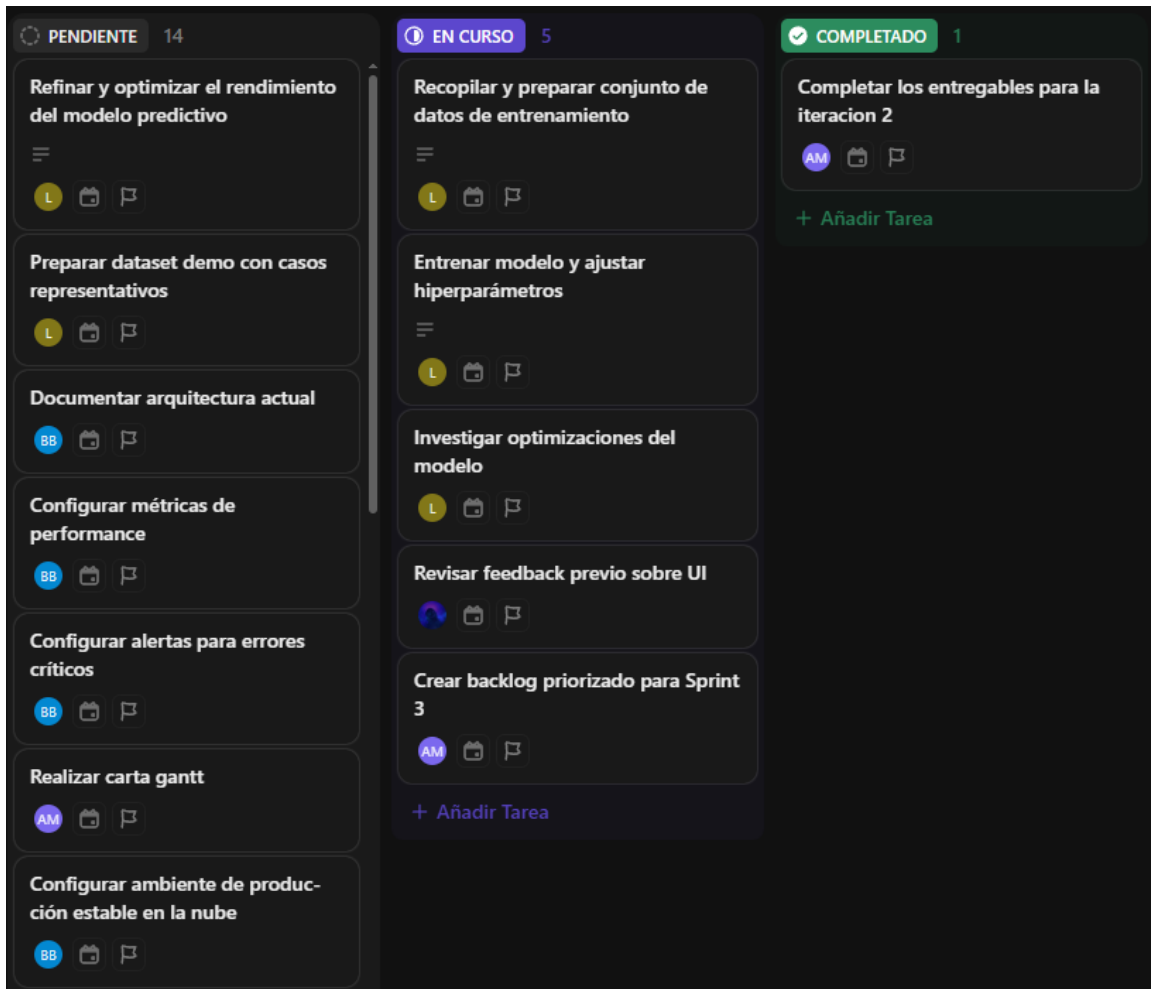


Figura 1. Ejemplo de tablero Kanban

Fuente: ClickUp

4. Planificación de Sprints

4.1 Definición de Sprints y ciclos de desarrollo

El proyecto AgroPredict fue estructurado principalmente en tres sprints, el primero más corto de una semana fue orientado a establecer bases técnicas iniciales y validar el enfoque con el cliente. El sprint 2 y sprint 3 fueron ambos de cuatro semanas, permitiendo desarrollo significativo de funcionalidades principales, con objetivos definidos alineados con la entrega incremental. La duración de cuatro semanas fue seleccionada por ser un período suficientemente largo para permitir desarrollo significativo de funcionalidades.

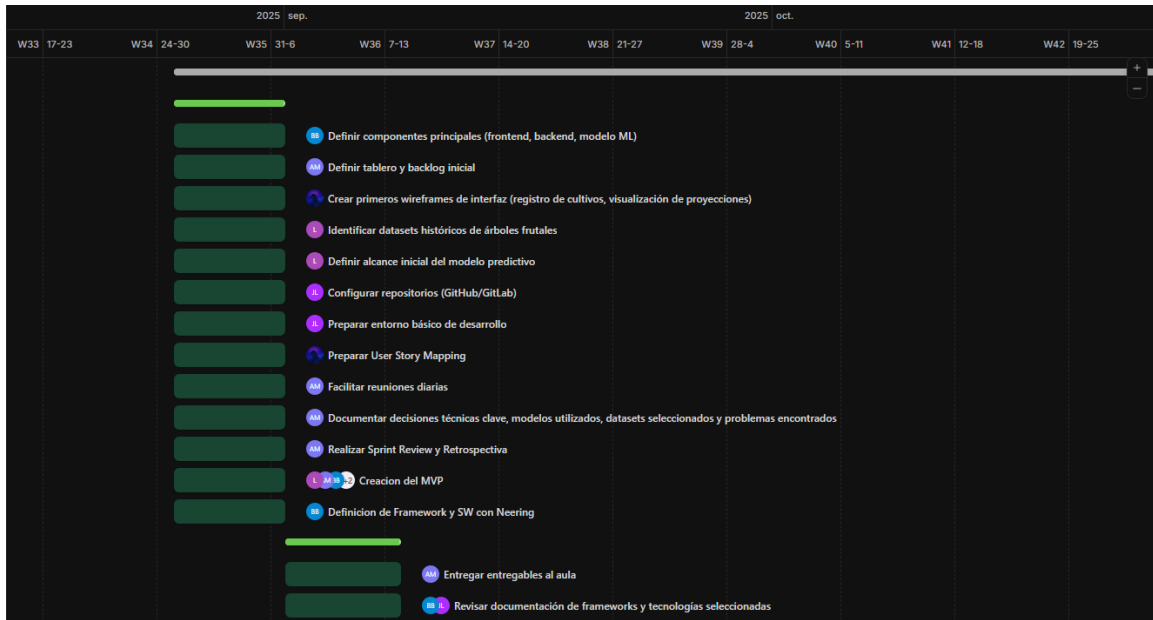


Figura 2. Carta Gantt sprint 1

Fuente: ClickUp

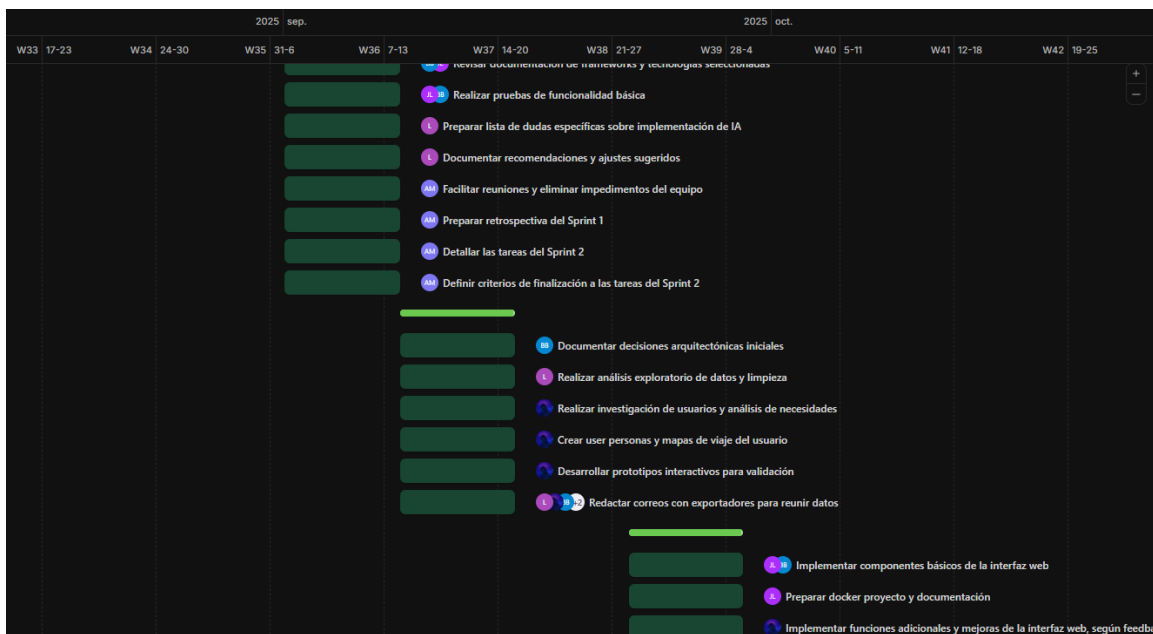


Figura 3. Carta Gantt sprint 2

Fuente: ClickUp

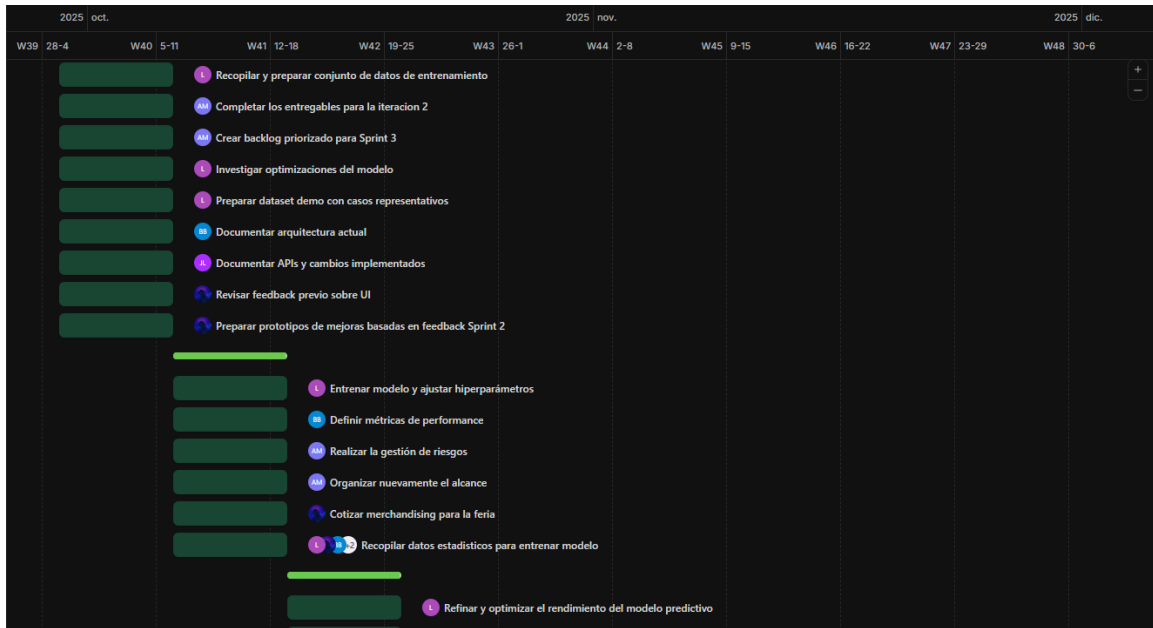


Figura 4. Carta Gantt sprint 3

Fuente: ClickUp

A continuación, se muestran las principales vistas que fueron presentadas en cada uno de los sprints.

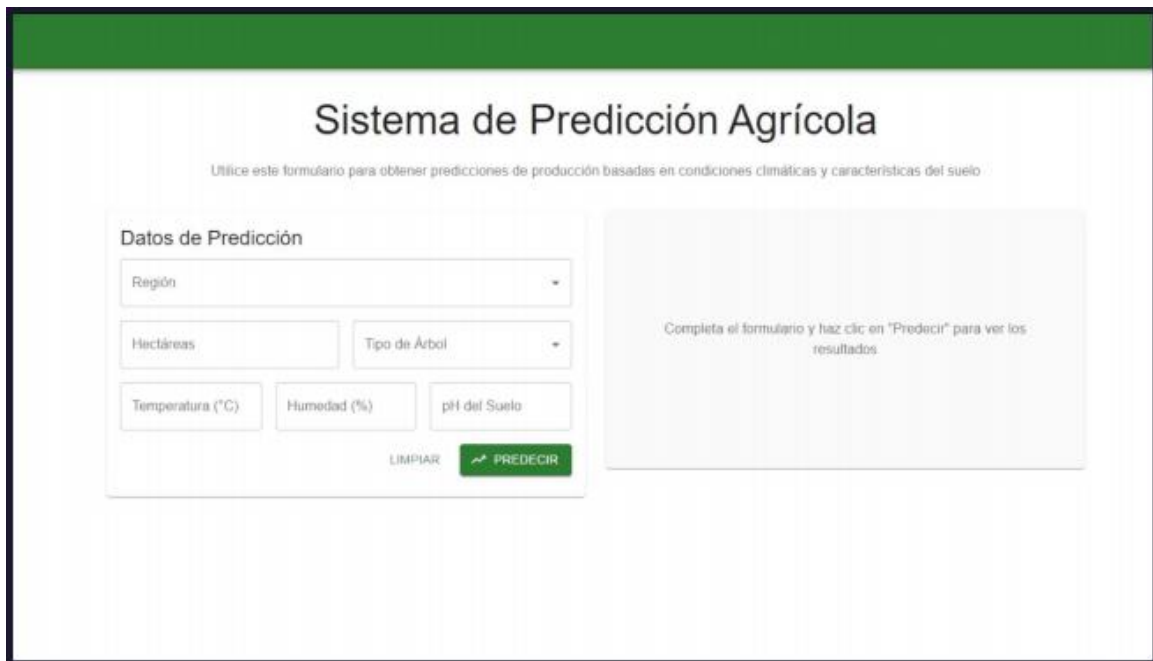


Figura 5. Vista principal presentada en sprint 1

Fuente: AgroPredict



Figura 6. Vista Dashboard presentada en sprint 2

Fuente: AgroPredict

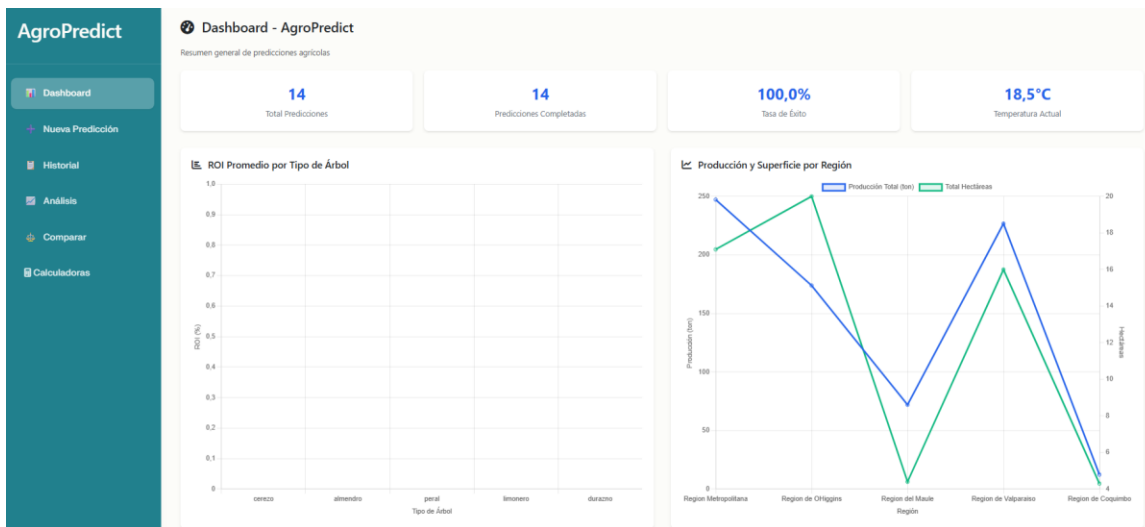


Figura 7. Vista Dashboard presentada en sprint 3

Fuente: AgroPredict

4.2 Estimación de tareas y asignación de recursos

El equipo asignó puntos de esfuerzo a sus tareas basados en sus conocimientos, incluyendo tiempo de investigación y tiempo de holgura.

Rol	Sprint 1	Sprint 2	Sprint 3
Diseñador UI/UX & PO	7h	23h	30h
Arquitecto de software	7h	26h	25h
Ingeniero de datos	6h	30h	35h
Desarrollador full-stack	6h	26h	25h
Scrum master	4h	15h	15h
Total	30h	120h	120h

Tabla 1. Distribución de recursos por Sprint

Fuente: Elaboración propia



Análisis de Distribución:

- Sprint 1: Fase exploratoria con carga equilibrada (6h por persona en promedio)
- Sprint 2: Mayor demanda debido a migración tecnológica y exploración de modelos (24h por persona en promedio)
- Sprint 3: Carga consistente con énfasis en Ingeniería de Datos (herramientas) e UI/UX (pulido final)
- Total: 270 horas de esfuerzo del equipo en las 9 semanas de proyecto

5. Ejecución de reuniones ágiles

5.1 Reuniones diarias

Se realizaron reuniones todas las semanas de lunes a viernes a las 21:00 horas con una duración variable entre 5 y 15 minutos con una estructura tipo: ¿Qué hice hoy? ¿Tengo algún impedimento?

Estas reuniones durante los tres sprints del proyecto ayudaron a resolver diversos impedimentos técnicos que permitieron avances significativos. En el sprint 1, se detectaron tres impedimentos relacionados con la configuración del entorno de desarrollo, los cuales fueron abordados de manera sistemática. El sprint 2 marcó un punto de inflexión importante al evidenciar la necesidad de una migración tecnológica antes de lo inicialmente planificado; esta identificación temprana permitió que el equipo realizara una adaptación proactiva sin comprometer los objetivos del proyecto. Finalmente, en el sprint 3, la curva de aprendizaje y las decisiones tomadas en reuniones anteriores se reflejaron en una mejora sustancial.

5.2 Sprint Planning

Actividades:

- Revisión del Product Backlog priorizado
- Selección de historias de usuario para el sprint
- Descomposición en tareas técnicas
- Compromiso del equipo sobre la meta del sprint

Sprint Planning 1:

- Seleccionadas 3 historias (HU-001, HU-002, HU-003)
- Meta: "Entregar MVP funcional con autenticación e ingreso de datos básico"

Sprint Planning 2:

- Seleccionadas 3 historias (HU-004, HU-006, mejoras de Sprint 1)
- Meta: "Mejorar arquitectura y agregar integración climática"

Sprint Planning 3

- Seleccionadas 3 historias (HU-005, HU-007, refinamiento general)
- Meta: "Pulir producto final y preparar para feria de software"



5.3 Sprint Review

El sprint Review se realizó al final de cada sprint con todo el equipo, junto con el patrocinador y profesores guía en las instalaciones del distrito v21.

Estructura:

- Demostración de funcionalidades completadas
- Feedback del cliente
- Discusión de cambios necesarios

Sprint Review 1 - Resultados:

- Opinión del patrocinador: satisfecho con dirección general del proyecto
- Feedback: Mejorar visualización de datos y claridad en formularios
- Decisión: Incorporar mejoras en próximo sprint

Sprint Review 2 - Resultados:

- Datos climáticos en tiempo real funcionando correctamente
- Opinión de profesores: Dudas sobre robustez del modelo de predicción
- Identificación de restricción temporal: Necesidad de adaptar enfoque técnico

Sprint Review 3 - Resultados:

- Producto final presentado exitosamente
- Cliente aprobó el sistema para demostración en feria de software

5.4 Sprint Retrospective

El sprint retrospective se realizó al final de cada sprint con todo el equipo, posterior al sprint review, con el formato: Qué salió bien, qué no salió bien, qué mejorar

Sprint Retrospective 1:

- Qué salió bien: Buena comunicación del equipo, reuniones diarias efectivas, herramientas de colaboración funcionando
- Qué no salió bien: Estimaciones iniciales fueron optimistas, falta de documentación temprana
- Mejoras para próximo sprint: Documentar decisiones arquitectónicas en tiempo real, ser más conservador en estimaciones

Sprint Retrospective 2:

- Qué salió bien: Adaptación rápida a cambios, migración a Django salió mejor de lo esperado, integración de API con problemas menores
- Qué no salió bien: Falta de datos para hacer un modelo robusto
- Mejoras para próximo sprint: Implementar herramientas útiles para los agricultores y directivos

Sprint Retrospective 3:



- Qué salió bien: Enfoque en pulir el producto, trabajo colaborativo intenso, la adaptación del modelo funcionó
- Qué no salió bien: Presión de tiempo al final fue alta, testing manual podría haber sido más sistemático
- Mejoras futuras: Iniciar con suite de pruebas automatizadas desde el inicio

6. Gestión de cambios y mejora continua

6.1 Identificación y registro de cambios

La gestión sistemática de cambios fue un pilar fundamental de la ejecución ágil del proyecto AgroPredict. Durante los tres sprints, se identificaron y registraron formalmente tres cambios significativos que desviaban del plan original. Cada cambio fue documentado considerando su descripción, razón fundamental, impacto en cronograma y recursos, nivel de aprobación requerido, y resultado final de su implementación.

El primer cambio crítico ocurrió durante el Sprint 2, cuando el equipo de arquitectura identificó limitaciones técnicas en Laravel que impedían la escalabilidad e integración efectiva con librerías de machine learning en Python. Específicamente, la comunicación entre el backend y los modelos de predicción en Python. Tras analizar las alternativas técnicas, el equipo propuso migrar completamente a Django, un framework de Python que permitiría una integración más natural de los modelos de machine learning. Esta decisión se aprobó en consenso durante una reunión especial con todo el equipo. La migración requirió aproximadamente 8 horas adicionales de esfuerzo, pero compensó con una arquitectura significativamente más robusta y mejor posicionada para mantenimiento futuro. El resultado fue una arquitectura más limpia, mejor documentada, y con mejor rendimiento en la integración de servicios externos.

El segundo cambio crítico, y probablemente el más significativo del proyecto, ocurrió en la transición entre el Sprint 2 y Sprint 3, cuando el equipo se reunió con los profesores guía para evaluar la viabilidad de completar el modelo robusto de predicción dentro de las restricciones de tiempo disponibles. Tras análisis detallado, se determinó que la complejidad técnica del modelo robusto originalmente planificado, que requería integración avanzada con datos y modelado complejo de variables agroclimáticas, no era viable en el tiempo restante. La decisión tomada fue pragmática y estratégica: implementar un modelo de predicción menos robusto, pero completamente funcional, complementado con herramientas adicionales de valor como diversas calculadoras usadas en el rubro agrícola (ver figura 8), análisis de predicción (ver figura 9), un apartado para comparar predicciones (ver figura 10) y un chatbot para consultas rápidas (ver figura 11). Esta adaptación fue aprobada por el equipo completo, el cliente Neering, y los profesores guía. El resultado fue mantener la funcionalidad principal del sistema, entregar a tiempo para la feria de software, y maximizar el valor entregado bajo condiciones de incertidumbre técnica.

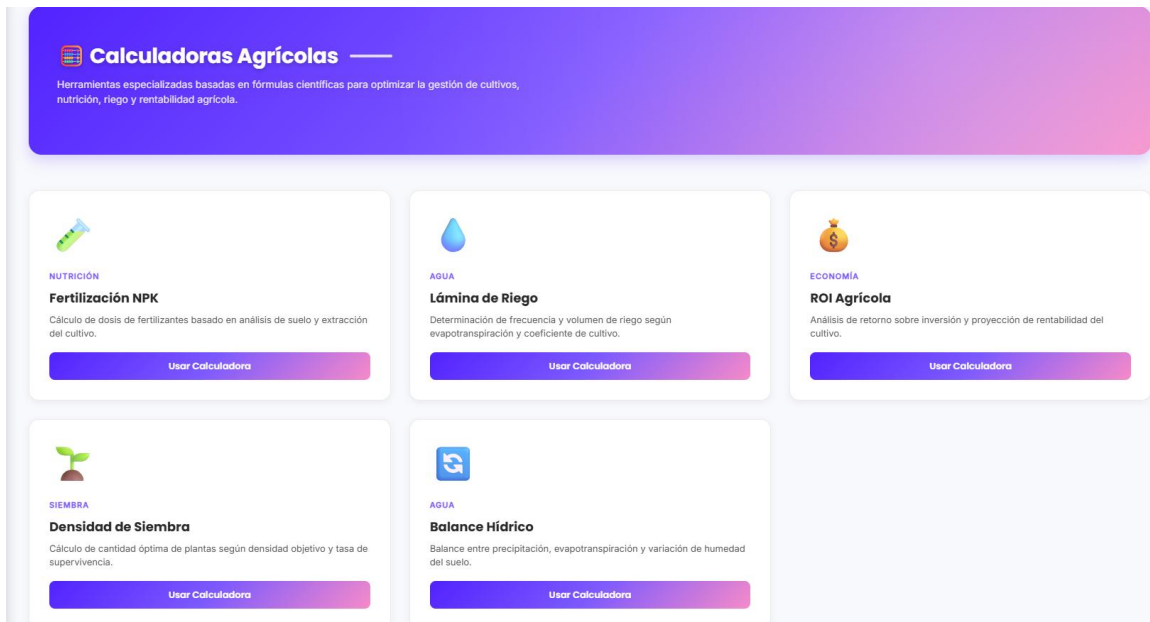


Figura 8. Vista de calculadoras

Fuente: AgroPredict

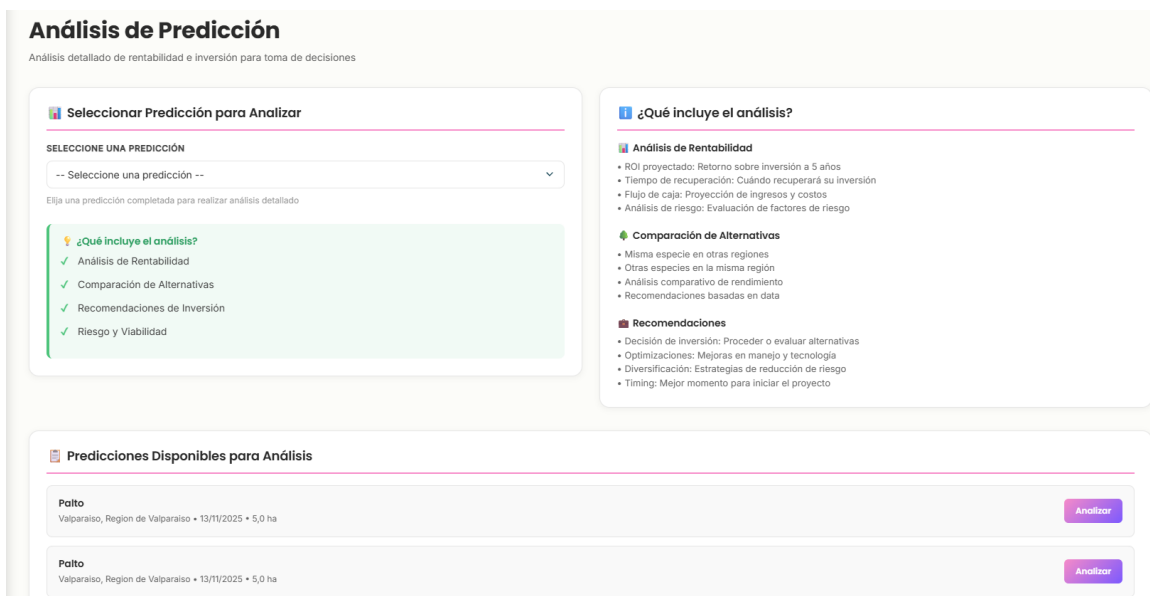


Figura 9. Vista de análisis de predicción

Fuente: AgroPredict

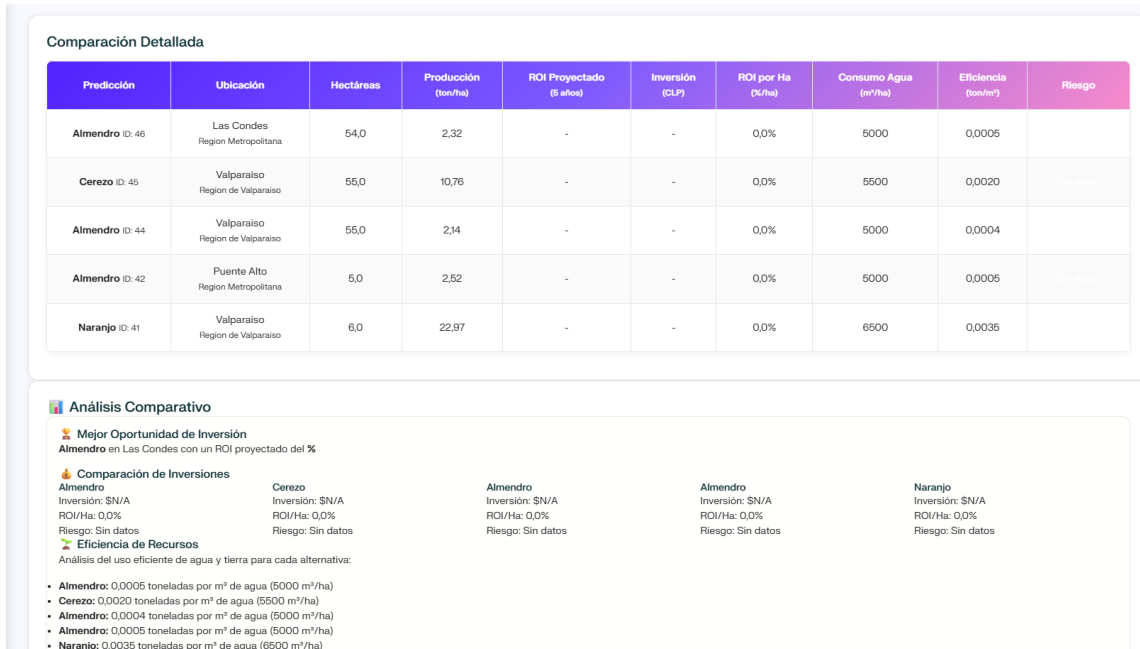


Figura 10. Vista de comparación de predicciones

Fuente: AgroPredict

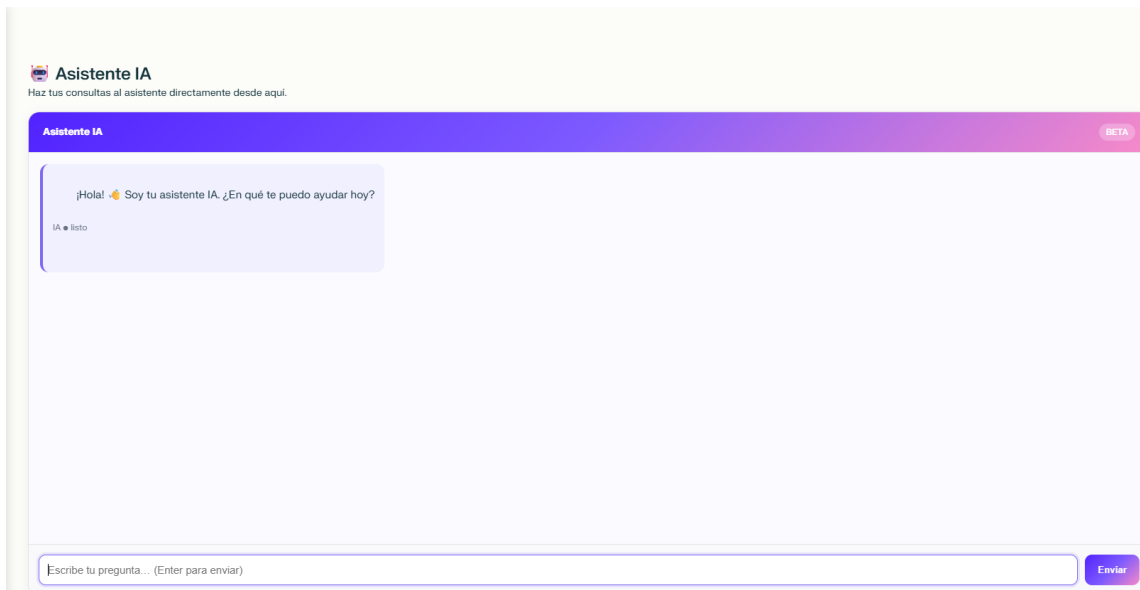


Figura 11. Vista de chatbot con ia

Fuente: AgroPredict

El tercer cambio se incorporó como mejora durante el Sprint 2, cuando el cliente solicitó explícitamente la integración de una API de datos meteorológicos para enriquecer las predicciones con información climática en tiempo real. Esta solicitud surgió durante la revisión de sprint cuando Neering reconoció que las predicciones agrícolas sin contexto climático tenían valor limitado. El equipo evaluó que esta funcionalidad podía implementarse en aproximadamente 6 horas sin



comprometer el cronograma del sprint. El resultado fue una funcionalidad altamente valorada por el cliente que aumentó significativamente el atractivo del producto final.

6.2 Aplicación de mejoras basadas en feedback

El proceso de mejora continua fue fundamental para la evolución del producto AgroPredict a través de los tres sprints. Después de cada Sprint Review, el equipo realizaba una sesión de análisis detallado del feedback recibido del cliente Neering, identificando patrones de solicitudes y priorizando las mejoras para la siguiente iteración.

Tras el Sprint Review 1, el cliente proporcionó feedback constructivo respecto a la interfaz de usuario y la claridad en la presentación de datos. Específicamente, los formularios de ingreso de datos agrícolas presentaban campos que no eran suficientemente intuitivos, y la visualización de resultados de predicciones carecía de contexto explicativo. En respuesta a este feedback, el diseñador UI/UX realizó un rediseño completo de los formularios durante el Sprint 2, mejorando etiquetas, añadiendo descripciones contextuales para cada campo, e implementando validación visual en tiempo real. Adicionalmente, se agregaron gráficas complementarias y cuadros informativos que explicaban qué representaba cada métrica de predicción. Estas mejoras fueron validadas nuevamente con el cliente en el Sprint Review 2, y resultaron en un aumento notable en la percepción de usabilidad del sistema.

Entre el Sprint 2 y Sprint 3, el feedback del cliente se enfocó en la experiencia general del dashboard principal. El cliente mencionó que le gustaría visualizar de forma más clara el resumen de sus cultivos activos. El equipo respondió reorganizando los elementos del dashboard para enfatizar la información más relevante, implementando tarjetas visuales mejoradas para cada cultivo.

6.3 Control de cambios e impacto

Cada cambio identificado fue evaluado considerando el impacto para asegurar que las decisiones fueran tomadas con información completa y consenso del equipo. El marco de evaluación consideraba el valor agregado para el cliente, la complejidad técnica asociada, la disponibilidad de tiempo en el cronograma actual, el riesgo técnico inherente a la implementación, y la prioridad estratégica del cambio en relación con los objetivos del proyecto.

La migración tecnológica de Laravel a Django fue evaluada como un cambio de alto valor para el cliente porque mejoraba significativamente la robustez y mantenibilidad del sistema, con alta complejidad técnica media. Además, dado que se identificó tempranamente durante el Sprint 2, había tiempo suficiente para completarla sin comprometer el cronograma general. El riesgo técnico fue mitigado mediante una planificación cuidadosa, y la experiencia del Arquitecto de Software junto con el desarrollador backend. La decisión fue aprobada con prioridad alta, y los resultados demostraron que fue la decisión correcta.

La adaptación del modelo de predicción fue evaluada como crítica en valor para el cliente, porque mantener la viabilidad del proyecto y entregar a tiempo era más importante que implementar un modelo perfecto que nunca se completaría. La complejidad técnica seguía siendo alta, pero el tiempo disponible era el factor crítico limitante. El riesgo técnico fue alto si se intentaba el modelo robusto original, pero bajo si se implementaba un modelo funcional complementado. La decisión fue aprobada con prioridad máxima, reconociendo que era una decisión pragmática alineada con principios ágiles de "entregar valor funcionando sobre funcionalidad perfecta".



7. Evaluación del desempeño ágil

7.1 Concepto y cálculo de velocity

El velocity es una métrica que mide la cantidad de trabajo completada por un equipo Scrum en un sprint, expresada en "puntos de historia". Estos puntos representan la complejidad relativa de tareas independientemente de tiempo invertido. El velocity es herramienta predictiva que permite estimar trabajo futuro y mejorar planificación.

Cálculo Utilizado en AgroPredict:

- Planning Poker adaptado: Estimación en horas directas de trabajo
- Fórmula Velocity: Suma de horas completadas en tareas con 100% de criterios de aceptación cumplidos

Sprints	Horas planificadas	Horas completadas	Velocity	% Cumplimiento
Sprint 1	30h	28h	28h	93%
Sprint 2	120h	115h	115h	96%
Sprint 3	120h	118h	118h	98%
Total	270h	261h	Promedio:87h/sprint	Promedio: 97%

Tabla 2. Velocity real por sprint

Fuente: Elaboración propia

7.2 Análisis de velocity y tendencias

El velocity mostró tendencia de mejora consistente de 93% a 98%. En el sprint 1, siendo exploración inicial, completó 28 de 30 horas planificadas. Para el sprint 2, a pesar de migración tecnológica no planeada, completó 115 de 120 horas (96%), demostrando capacidad de adaptación. Finalmente, para el sprint 3, se alcanzó un 98% (118 de 120), indicando madurez, ya que, el equipo conocía la arquitectura, las dinámicas de trabajo y herramientas.

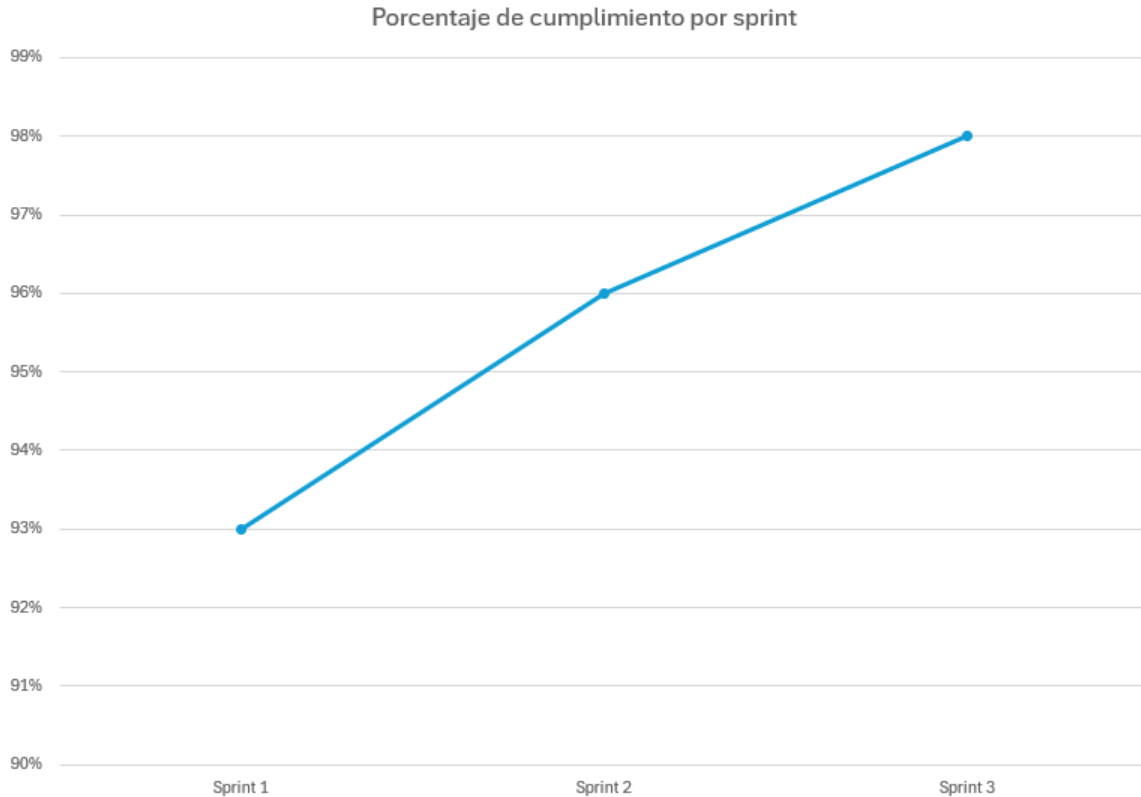


Figura 12. Gráfico de porcentaje de cumplimiento de horas por sprint

Fuente: Elaboración propia

Como parte de la retroalimentación obtenida tras la entrega final del proyecto, durante la 33ª feria de software se logró una gran aceptación por parte del público y del patrocinador. Además, se recibieron recomendaciones e ideas para implementar en el futuro, lo que motiva al equipo a seguir trabajando en el desarrollo del proyecto.

8. Desafíos y soluciones implementadas

8.1 Desafíos técnicos

Desafío técnico #1: Falta de datos robustos para modelo complejo

- Problema: El modelo de predicción robusto requería gran cantidad de datos que tenían disponibilidad limitada.
- Síntomas: Exploración extensa sin resultado, incertidumbre en precisión, presión de tiempo creciente.
- Solución: Pivote estratégico hacia herramientas complementarias prácticas (calculadoras, chatbot, análisis comparativo) + modelo funcional básico + integración de datos ERA5 para contexto histórico.
- Resultado: Entrega exitosa a tiempo, máximo valor para usuario final, validación de pragmatismo ágil.



Desafío técnico #2: Migración tecnológica no planeada

- Problema: Laravel presentaba limitaciones para integración natural con librerías ML en Python.
- Síntomas: Problemas de rendimiento, dificultad de integración, falta de documentación.
- Solución: Migración a Django, planificación cuidadosa, testing exhaustivo.
- Resultado: Arquitectura más robusta, mejor preparada para escalabilidad futura.

8.2 Desafío de tiempo

Desafío tiempo #1: Coordinación entre especialidades

- Problema: Equipo de datos y backend trabajaban con diferentes velocidades.
- Síntomas: Retrasos en integración, incertidumbre sobre formatos de datos, tiempo muerto.
- Solución: Documentación clara de interfaces, revisiones de código compartidas.
- Resultado: Integración ERA5 completada eficientemente en Sprint 3.

Desafío tiempo #2: Presión de Feria de Software

- Problema: Restricción temporal para entrega (13 de noviembre).
- Síntomas: Necesidad de priorizar, reducción de alcance, presión en equipo.
- Solución: Definición clara de MVP, reducción de alcance técnico manteniendo valor, gestión de carga de trabajo.
- Resultado: Entrega exitosa sin comprometer calidad.

8.3 Tabla de riesgos

Riesgo	Probabilidad	Impacto	Nivel de Riesgo	Estado	Mitigación
Falta de datos robusto	Alta	Alto	CRÍTICO	Mitigado	Herramientas complementarias
Migración Django	Media	Medio	MEDIO	Mitigado	Planificación temprana
Falta coordinación inter-equipos	Media	Medio	MEDIO	Mitigado	Revisiones de código compartidas
Presión temporal	Alta	Medio	MEDIO	Mitigado	Priorización clara

Tabla 3. Principales riesgos del proyecto

Fuente: Elaboración propia

9. Estimación de costos

9.1 Costos de recursos humanos

Los valores por hora fueron establecidos basándose en datos oficiales del Ministerio de Educación de Chile (MINEDUC), en la plataforma "Mi Futuro", que proporciona estadísticas de empleabilidad e ingresos promedio para Ingeniería en Computación e Informática y otras carreras en Chile (ver figura 13).

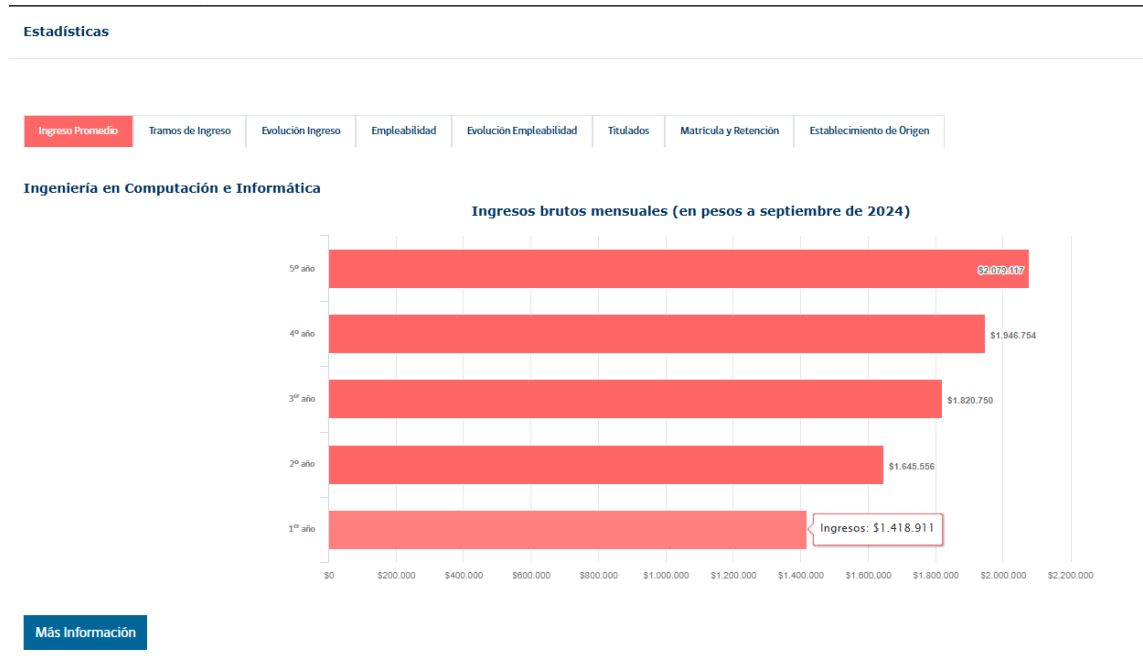


Figura 13. Ingresos brutos mensuales ingeniería en computación e informática

Fuente: mifuturo.cl

Metodología de Estimación:

- Ingreso bruto mensual al primer año de titulación: \$1.418.911 CLP
- Fuente: Promedio de cohortes titulados 2020, 2021 y 2022 de Ingeniería en Computación e Informática
- Tipo de cambio: \$980 CLP/USD (noviembre 2025)
- Conversión: $\$1.418.911 / 980 = \$1,448$ USD mensuales

Cálculo del valor hora profesional:

- Horas mensuales estándar: 21 días \times 8 horas = 168 horas/mes
- Valor hora profesional al primer año: $\$1,448 / 168 = \8.62 USD/hora

Dado que todos los miembros del equipo son estudiantes de mismo nivel académico, se utiliza valor hora uniforme de \$8.62 USD para todos los roles.



Rol	Horas totales	Costo/hora	Costo total
Diseñador UI/UX	60h	\$8.62 USD/hora	\$517 USD
Arquitecto de software	58h	\$8.62 USD/hora	\$500 USD
Ingeniero de datos	71h	\$8.62 USD/hora	\$612 USD
Desarrollador Full-Stack	57h	\$8.62 USD/hora	\$491 USD
Scrum master	24h	\$8.62 USD/hora	\$207 USD
Total Recursos Humanos	270h	\$8.62 USD/hora	\$2327 USD

Tabla 4. Costos por rol

Fuente: Elaboración propia

Sprint	Horas totales	Costo/hora	Costo total
Sprint 1	30h	\$8.62 USD/hora	\$259 USD
Sprint 2	120h	\$8.62 USD/hora	\$1034 USD
Sprint 3	120h	\$8.62 USD/hora	\$1034 USD
Total Recursos Humanos	270h	\$8.62 USD/hora	\$2327 USD

Tabla 5. Costos por sprint

Fuente: Elaboración propia

9.2 Costos de herramientas, infraestructura y otros

Categoría	Concepto	Costo
Herramientas	ClickUp, Discord, GitHub	Gratuito
Herramientas	Figma (2 meses)	\$10 USD
Infraestructura	Railway (1 mes)	\$5 USD
Infraestructura	API openai	\$5 USD
Total otros		\$20 USD

Tabla 6. Costos de herramientas e infraestructura

Fuente: Elaboración propia

Costo total del proyecto: \$2327 (RR.HH.) + \$20 (Otros) = \$2347 USD

10. Conclusiones

10.1 Cumplimiento de objetivos

El proyecto AgroPredict alcanzó exitosamente su objetivo general: gestionar el desarrollo de un sistema predictivo agrícola mediante metodologías ágiles Scrum, entregando producto de valor funcional bajo condiciones de incertidumbre técnica y temporal. Se implementó un ciclo Scrum completo con tres sprints diferenciados, cada uno con objetivos claros y ceremonias ejecutadas consistentemente. El equipo completó el 97% de las horas planificadas (261 de 270 horas), validando que la planificación inicial fue realista y la ejecución fue disciplinada.

Los objetivos específicos fueron cumplidos integralmente. Primero, Scrum fue implementado correctamente con todas las ceremonias: Las reuniones diarias ayudaron a identificar impedimentos tempranamente, la Sprint Planning estableció metas claras para cada ciclo, las Sprint Reviews



permitieron verificar la validación frecuente con el patrocinador y profesores, finalmente con las Sprint Retrospectives facilitaron mejora continua. El equipo pasó de 93% de cumplimiento en Sprint 1 a 98% en Sprint 3, demostrando curva de madurez consistente. En segundo lugar, los recursos fueron optimizados efectivamente con 270 horas de trabajo distribuidas estratégicamente según las necesidades de cada sprint: Sprint 1 fue exploración, Sprint 2 fue construcción con migración no planeada, Sprint 3 fue refinamiento con herramientas complementarias. También, la comunicación fue efectiva y cliente participó activamente en revisiones validando dirección del proyecto. Además, el desempeño fue evaluado sistemáticamente con métricas claras: velocity promedio de 87 horas/sprint indicó capacidad predecible y satisfacción equipo mejoró progresivamente. Finalmente, las lecciones aprendidas fueron documentadas: el valor de herramientas complementarias sobre perfección técnica y cómo la comunicación diaria acelera resolución de impedimentos. El proyecto concluye con un producto completo, cliente satisfecho, equipo desarrollado, y metodología validada.

10.2 Validación de metodologías ágiles

El proyecto AgroPredict proporciona evidencia concreta de que los principios fundamentales del Manifiesto Ágil no son meros ideales teóricos, sino directrices operacionales que generan resultados tangibles en contextos reales con restricciones. La validación empírica se manifiesta en cuatro dimensiones.

Primero, "Responder al cambio sobre seguir un plan": Cuando el equipo identificó en Sprint 2 que el modelo robusto original no era viable (restricción de datos + tiempo), ejecutó un pivote estratégico hacia herramientas complementarias. Este cambio no fue considerado fracaso de planificación, sino adaptación inteligente. La decisión fue tomada en conjunto con profesores guía, validada mediante Sprint Reviews, e implementada sin comprometer la entrega final a tiempo. Resultado: producto completo en fecha objetivo con 5 herramientas en lugar de 1 incompleta.

Segundo, "Colaboración con el cliente sobre negociación de contratos": El patrocinador participó en cada Sprint Review aportando feedback que fue incorporado directamente. Tras sprint 1, el cliente señaló falta de claridad en formularios; sprint 2 respondió con rediseño completo. Tras Sprint 2, el cliente sugirió herramientas prácticas; sprint 3 implementó calculadoras y chatbot. Esta colaboración continua resultó en producto que verdaderamente resuelve necesidades reales, no requisitos especificados sin validación.

Tercero, "Individuos e interacciones sobre procesos y herramientas": Reuniones diarias de 5-15 minutos fueron más efectivos que documentación pesada. La comunicación diaria permitió identificar el problema de migración de framework el primer día en que surgió, no después de semanas de retraso. Las reuniones sincrónicas entre equipos de datos y backend resolvieron desafíos de integración en horas, no días.

Cuarto, "Software funcionando sobre documentación extensiva": El equipo priorizó entregar funcionalidad que se pueda probar en cada sprint sobre documentación completa anticipada. El MVP fue validado con cliente en Sprint 1, no después de construcción completa. Las herramientas complementarias fueron desarrolladas respondiendo a feedback real en Sprint 3, no basadas en especificaciones iniciales que podrían haber sido irrelevantes.



10.3 Análisis de éxito

El éxito del proyecto AgroPredict resulta de la interacción sinérgica de factores críticos que se reforzaron mutuamente:

- Planificación inicial clara flexible: El Sprint 1 fue deliberadamente corto (1 semana) y explorador, reconociendo incertidumbre técnica. Sprints 2-3 fueron más extensos (4 semanas) para construcción, pero manteniendo capacidad de adaptación. Esta estructura permitió identificar limitaciones antes de invertir demasiado tiempo.
- Decisión pragmática alineada con principios ágiles: En lugar de perseguir modelo perfecto que no se completaría, el equipo propuso herramientas complementarias que entregaran valor hoy. Cliente y profesores validaron que era una decisión correcta.
- Comunicación diaria efectiva: Los 5-15 minutos diarios de las reuniones diarias fueron una inversión de tiempo que rindió dividendos enormes. Los impedimentos fueron identificados en horas, no días. La migración a Django fue coordinada sin que diferentes especialidades desalinearan, esto resultando en un proyecto nunca "roto".
- Cliente comprometido en cada revisión: Neering no fue un stakeholder remoto revisando al final, sino participante activo en cada Sprint Review. Este compromiso permitió feedback temprano, validación de dirección, y ajustes incrementales. La satisfacción final fue alta porque cliente vio su voz reflejada en el producto incrementalmente, no sorprendido al final.

10.4 Lecciones aprendidas

El proyecto generó cuatro lecciones aprendidas de alto valor y son aplicables a futuras iniciativas de software ágil:

La primera lección fue que pivotes tempranos en incertidumbre resultan en mejor valor que perseguir objetivos inviables. En contextos donde la complejidad técnica es alta (ML, datos espaciales), es mejor detectar tempranamente si una aproximación original no funcionará, y entonces darle una vuelta, que invertir 150 horas en un camino incorrecto.

La segunda lección fue que herramientas complementarias de valor práctico pueden ser más valiosas que una herramienta "perfecta" incompleta. Cliente prefirió 5 herramientas funcionales (modelo + calculadoras + chatbot + análisis + históricos) a 1 herramienta incompleta e incierta. Un MVP que funciona hoy es mejor que uno que no llega nunca.

La tercera lección fue que comunicación frecuente (diaria entre especialidades) acelera integración. Cuando Datos y Backend se reunían varias veces por semana, avanzaban. Cuando era asincrónico, se atrasaban. Sincronía es más importante que los individuos sean perfectos.

La cuarta lección fue que el pragmatismo ágil (entregar valor hoy) resulta mejor que optimización perfecta. Invertir \$2,349 en un producto funcional entregado que genera valor en primer año es mejor que \$0 gastado en modelo perfecto que nunca se entrega.

10.5 Reflexión final

El proyecto AgroPredict, representa un caso de estudio exitoso de cómo metodologías ágiles transforman incertidumbre técnica en valor entregable.



Pero más allá de métricas, el proyecto demuestra que equipos pequeños de estudiantes en formación pueden gestionar efectivamente proyectos de software complejos cuando adoptan disciplina ágil. No fue el modelo "perfecto" que se aspiraba originalmente, fue mejor: fue pragmático, entregable, valorado por cliente, y comercializable. La decisión de pivotar en sprint 2 cuando el modelo robusto se mostró inviable fue audaz pero correcta, validando que agilidad no significa carecer de dirección, sino mantener dirección mientras se adaptan tácticas.

El impacto académico es que demostramos metodologías ágiles no en teoría sino en práctica, bajo restricciones reales (tiempo, datos, equipo en formación). El impacto profesional es que cada miembro del equipo desarrolló capacidades invaluable: gestión de cambios, comunicación técnica, decisiones pragmáticas bajo incertidumbre. El impacto comercial es que existe un producto viable que puede generar ingresos y resolver un problema real del sector agrícola chileno.

Para finalizar, se puede decir que el éxito de AgroPredict no fue accidental, sino resultado de aplicación disciplinada de metodologías ágiles combinada con pragmatismo, comunicación, y adaptabilidad. Estos son exactamente los atributos que la industria de software busca en profesionales y equipos. Esta tesina no es solo documento de proyecto, es evidencia que estudiantes de la USM están preparados para contribuir valor real al sector tecnológico.



Referencias

<https://www.scrum.org/>

<https://www.scrumio.com/scrumsprint-review/>

<https://www.linkedin.com/advice/1/youre-facing-scope-creep-your-agile-project-atsgf?lang=es&originalSubdomain=es>

<https://asana.com/es/resources/what-is-scrum>

<https://www.linkedin.com/pulse/alcance-cambios-e-incertidumbre-c%C3%B3mo-los-trata-scrum-rosales-garc%C3%ADa/>

<https://www.scrummanager.com/blog/2011/04/gestion-de-cambios-en-scrum/>

<https://www.mifuturo.cl/buscador-de-estadisticas-por-carrera/>