

**Departamento de Ciencias de la
Computación (DCCO)**

Carrera de Ingeniería de Software

Análisis y Diseño de Software

Perfil del Proyecto

Presentado por: Panchez Darwin, Proaño José,
Robalino Cristian,
Grupo 4.

Tutor académico: Ing. Jenny A Ruiz R

Ciudad: Sangolquí

Fecha: 11/05/2025

Índice

Pág.

| | |
|--|-----------|
| 1. Introducción | 5 |
| 2. Planteamiento del trabajo | 6 |
| 2.1 Formulación del problema | 6 |
| 2.2 Justificación | 6 |
| 3. Sistema de Objetivos | 6 |
| 3.1. Objetivo General | 6 |
| 3.2. Objetivos Específicos | 7 |
| 4. Alcance | 7 |
| 5. Marco Teórico | 8 |
| 5.1 Metodología (Marco de trabajo 5W+2H) | 11 |
| 6. Ideas a Defender | 12 |
| 7. Resultados Esperados | 13 |
| 8. Viabilidad | 13 |
| 8.1 Humana | 14 |
| 8.1.1 Tutor Empresarial | 14 |
| 8.1.2 Tutor Académico | 14 |
| 8.1.3 Estudiantes | 15 |
| 8.2 Tecnológica | 15 |
| 8.2.1 Hardware | 15 |
| 8.2.2 Software | 16 |
| 9. Conclusiones y recomendaciones | 17 |
| 10. Planificación para el Cronograma: | 17 |
| 11. Referencias | 18 |

1. Introducción



En los criaderos de conejos tradicionales, gran parte de las labores diarias, como el control de jaulas, la alimentación, el monitoreo sanitario y el seguimiento reproductivo, se realizan de forma manual. Este enfoque, aunque funcional, resulta altamente demandante en tiempo y esfuerzo, además de ser propenso a errores humanos. Como consecuencia, se dificulta la trazabilidad individual de los animales y se limita la capacidad de analizar datos productivos en tiempo real, lo que impacta negativamente en la eficiencia de la producción. Diversos estudios en el campo de la cunicultura han

resaltado la importancia de implementar tecnologías que optimicen la gestión del criadero. Si bien existen avances en la tecnificación de la producción, muchos procesos rutinarios aún dependen del registro manual, como el consumo diario de alimento, el control de tratamientos sanitarios y la planificación de la reproducción. Esta brecha tecnológica representa una oportunidad para desarrollar soluciones innovadoras que atiendan a las necesidades específicas del sector. En este contexto, surge el sistema Hoptolt, una propuesta orientada a digitalizar y centralizar la gestión operativa del criadero. El objetivo principal de este sistema es facilitar el trabajo del productor, mejorar la organización de los datos, y permitir una toma de decisiones más informada y oportuna. La creación de este sistema responde a una necesidad real del sector y se plantea como una herramienta clave para modernizar la producción cunícola.

2. Planteamiento del trabajo

2.1 Formulación del problema

Actualmente, los productores de conejos registran la información clave (número de jaulas, animales asignados, vacunas aplicadas, datos reproductivos) en papel o planillas independientes. Este enfoque no solo es propenso a errores e inconsistencias, sino que también dificulta la consulta y análisis oportuno de la información.

En respuesta a estas limitaciones, el proyecto propone el desarrollo de Hoptolt, un sistema digital integrado que permita automatizar y centralizar las labores del criadero, incluyendo el manejo de jaulas, la programación de alimentación, y el monitoreo continuo de la salud y reproducción de los conejos. Con esta solución, se busca no solo reducir la carga operativa diaria, sino también mejorar la precisión en los registros, optimizar la toma de decisiones y aumentar la productividad general del criadero.

2.2 Justificación

La propuesta del sistema Hoptolt representa una contribución relevante tanto a nivel práctico como investigativo. Su implementación busca optimizar la gestión del criadero mediante un control más preciso de recursos, salud y reproducción, alineándose con estudios que demuestran cómo la digitalización en el sector agropecuario mejora la producción.

Desde el ámbito científico, la aplicación de tecnologías informáticas en la cunicultura es todavía primeriza, lo que convierte a Hoptolt en una propuesta innovadora. Este proyecto no solo atiende una necesidad concreta del sector, sino que también aporta una base para futuras investigaciones sobre transformación digital en áreas ganaderas con bajo nivel de tecnificación.

3. Sistema de Objetivos

3.1. Objetivo General

Desarrollar el sistema Hoptolt para la gestión automatizada de criaderos de conejos, empleando tecnologías web modernas, con el fin de agilizar el control de jaulas, la programación de alimentación, el registro de salud y la monitorización reproductiva, donde el sistema debe integrar estas funciones en una plataforma amigable para mejorar la trazabilidad de los animales y optimizar la eficiencia operativa del criadero.

3.2. Objetivos Específicos

- Optimizar la gestión de los criaderos de conejos a través de una organización más eficiente de los datos y procesos operativos, esto incluye la automatización del registro, consulta y actualización de información sobre las jaulas, razas y animales, lo que sustituye los métodos manuales y dispersos
- Diseñar la arquitectura del sistema para la gestión integral del criadero de conejos, considerando módulos que aborden aspectos como alimentación, vacunación y desparasitación, asegurando escalabilidad, modularidad y facilidad de mantenimiento.
- Seleccionar e implementar un patrón de diseño adecuado que permita estructurar de manera eficiente los componentes del sistema, facilitando la reutilización de código, la extensibilidad del software y la interacción coherente entre los módulos funcionales.

4. Alcance

El sistema Hoptolt incluirá funcionalidades para gestionar todas las fases clave de la crianza de conejos en un criadero, proporcionando una plataforma centralizada que optimiza la administración y operación diaria. Como parte del desarrollo del sistema, se llevará a cabo un proceso completo que incluirá la elicitación de requerimientos, la elaboración de casos de uso, el análisis y diseño del sistema (en el que se diseña

la arquitectura y se implementan patrones de diseño), el desarrollo, y la implementación de pruebas para validar el correcto funcionamiento.

Las funcionalidades específicas incluyen:

Gestión de Jaulas:

- Registro, consulta, edición y eliminación de datos de jaulas.
- Funcionalidad para asignar jaulas a los conejos, lo que permitirá un control detallado de los espacios disponibles y utilizados.

Gestión de Razas:

- Registro, consulta, edición y eliminación de razas de conejos.
- La posibilidad de filtrar conejos por raza para facilitar la organización y el seguimiento.

Gestión de Datos de Conejos:

- Registro, consulta, edición y eliminación de datos de los conejos.
- Filtrado por razas, permitiendo un manejo más eficiente de los grupos de animales.

Control de Alimentación:

- Control del esquema alimentario de los conejos, teniendo en cuenta factores como la edad y peso para optimizar la dieta.

Control de Vacunación y Desparasitación:

- Registro de datos relacionados con la vacunación y desparasitación.

Control de Crecimiento:

- Monitoreo del crecimiento de los conejos, actualizando los datos a tiempo real, permitiendo la identificación temprana de desviaciones que puedan afectar la producción.

Gestión de Reproducción y Parto:

- Registro de datos relacionados con la monta, cálculo de la fecha de parto y aviso anticipado al criador sobre partos próximos.
- Eliminación de registros de partos.

Gestión de Reportes:

- Generación de reportes detallados sobre alimentación, vacunación y desparasitación, lo que facilitará la toma de decisiones en tiempo real.

Además, el sistema incluye una funcionalidad de inicio de sesión que asegura un control de acceso para los usuarios, garantizando la seguridad y privacidad de la información. Estas funcionalidades reemplazarán los métodos tradicionales de

gestión manual, mejorando la eficiencia, la trazabilidad y la organización del criadero.

5. Marco Teórico

Para la construcción de Hoptolt, se emplearán diversas herramientas y conceptos fundamentales que aseguran un desarrollo robusto, escalable y centrado en el usuario. A continuación, se detallan las principales tecnologías y metodologías que guiarán este proceso:

- **Diagramas de Casos de Uso (UML):** Los diagramas de casos de uso, parte del Lenguaje de Modelado Unificado (UML), son herramientas visuales que representan las interacciones entre los usuarios (actores) y el sistema. Estos diagramas ayudan a identificar y documentar los requisitos funcionales del sistema desde la perspectiva del usuario, facilitando la comunicación entre los diferentes stakeholders y asegurando que el desarrollo se alinee con las necesidades del usuario final.
- **Historias de Usuario:** Las historias de usuario son una técnica utilizada dentro de metodologías ágiles para capturar requisitos desde el punto de vista del usuario final. Cada historia describe una funcionalidad específica que aporta valor, en un lenguaje sencillo y comprensible. Esta herramienta permite priorizar las tareas de desarrollo, enfocándose en la experiencia del usuario y asegurando que el sistema evolucione según sus necesidades reales.
- **Principios de diseño:** Se refieren a una directriz o regla general que ayuda a los desarrolladores a crear sistemas más eficientes, fáciles de mantener y escalables. Estos principios proporcionan enfoques fundamentales sobre cómo estructurar el código y organizar los componentes de una aplicación. Los principios de diseño buscan mejorar la calidad del software y garantizar que el sistema sea flexible, robusto y comprensible.
- **Patrones de diseño:** Son una solución reutilizable y probada para un problema común que ocurre en el desarrollo de software. Los patrones de diseño son enfoques que resuelven problemas específicos de manera estructurada y eficiente. Un patrón de diseño no es una implementación exacta, sino una guía para abordar ciertos problemas en la programación.

Los patrones ayudan a los desarrolladores a resolver problemas recurrentes y a crear software más robusto y flexible, siguiendo un esquema probado.

- **Express.js (Framework para Node.js):** Express.js es un framework minimalista y flexible para Node.js, ampliamente utilizado para desarrollar aplicaciones web y APIs. Aunque no impone una estructura específica, su simplicidad permite a los desarrolladores implementar patrones arquitectónicos como el Modelo-Vista-Controlador (MVC) de manera eficiente. Express.js se destaca por su sistema de enrutamiento robusto y su capacidad para manejar middleware, lo que facilita la creación de aplicaciones web y móviles de alto rendimiento.

- **React (Biblioteca de JavaScript para Interfaces de Usuario):**

React es una librería de código abierto desarrollada por Facebook para construir interfaces de usuario declarativas y basadas en componentes. En Hoptolt se empleará React para crear vistas dinámicas y reutilizables, facilitando la actualización de datos en pantalla (por ejemplo, listas de jaulas o reportes en tiempo real) sin recargar toda la página.

- **MongoDB (Sistema de Gestión de Bases de Datos NoSQL):**

MongoDB es un sistema de gestión de bases de datos NoSQL de código abierto, ampliamente utilizado por su flexibilidad y escalabilidad. A diferencia de las bases de datos relacionales, MongoDB almacena los datos en documentos BSON (una extensión de JSON), lo que permite una estructura de datos más dinámica y adaptable. Es ideal para aplicaciones que requieren manejo eficiente de grandes volúmenes de datos no estructurados o semi-estructurados. Ofrece características como replicación, particionamiento horizontal (sharding) y alta disponibilidad, lo que lo convierte en una opción robusta para desarrollos modernos y en tiempo real.

- **Figma (Herramienta de Diseño y Prototipado de Interfaces):** Figma es una plataforma de diseño colaborativa basada en la nube que permite a los equipos crear, prototipar y compartir interfaces de usuario de manera eficiente. Sus funcionalidades de edición en tiempo real y comentarios integrados facilitan la colaboración entre diseñadores y desarrolladores,

asegurando que las interfaces sean intuitivas y alineadas con las necesidades del usuario final. Además, Figma ha incorporado recientemente herramientas impulsadas por inteligencia artificial que automatizan la creación de prototipos y mejoran la eficiencia del proceso de diseño.

- **GitHub (Plataforma de Control de Versiones y Colaboración):** GitHub es una plataforma ampliamente utilizada para el control de versiones y la colaboración en proyectos de software. Permite a los equipos gestionar el código fuente mediante sistemas de control de versiones como Git, facilitando el seguimiento de cambios, la revisión de código y la gestión de incidencias. Las funcionalidades de GitHub, como los pull requests y las acciones automatizadas, mejoran la eficiencia del desarrollo y aseguran una integración continua en el flujo de trabajo del equipo.
- **Docker (Contenedores para Entornos Consistentes):** Docker es una plataforma que permite empaquetar aplicaciones y sus dependencias en contenedores, asegurando que se ejecuten de manera consistente en diferentes entornos. Esta tecnología facilita la implementación y escalado de aplicaciones, reduciendo los problemas relacionados con las diferencias entre entornos de desarrollo y producción. Además, Docker mejora la eficiencia en la gestión de recursos y simplifica la integración de nuevas funcionalidades en el sistema.
- **Visual Studio Code (Entorno de Desarrollo Integrado - IDE):** Visual Studio Code (VS Code) es un editor de código fuente ligero y extensible desarrollado por Microsoft. Ofrece soporte para múltiples lenguajes de programación y cuenta con una amplia gama de extensiones que mejoran la productividad del desarrollador. Entre sus características destacan la integración con sistemas de control de versiones, herramientas de depuración y funcionalidades de autocompletado, lo que lo convierte en una herramienta versátil para el desarrollo de aplicaciones web.

La integración de estas herramientas y metodologías proporcionará una base sólida para el desarrollo de Hoptolt, asegurando una aplicación eficiente, escalable y centrada en el usuario. La combinación de un framework robusto como Express.js,

una base de datos confiable como MongoDB, herramientas de diseño colaborativas como Figma y prácticas de desarrollo modernas garantizarán el éxito del proyecto.

5.1 Metodología (Marco de trabajo 5W+2H)

| ¿QUÉ? | ¿CÓMO? | ¿QUIÉN? | ¿CUÁNDO? | ¿DÓNDE? | ¿POR QUÉ? | ¿CUÁNTO? |
|---|--|---|---|--|---|---|
| Desarrollar Hoptolt, un sistema web para la gestión automatizada de criaderos de conejos. | Mediante métodos ágiles, específicamente iteraciones Scrum. Durante el análisis se utilizarán historias de usuario para definir y priorizar funcionalidades. El diseño contempla principios y patrones de diseño. Se utilizará prototipado en Figma y se realizará una validación continua con el cliente. | Equipo de estudiantes de Ingeniería, bajo la tutoría académica y guía de un tutor empresarial (productor de conejos). | Fase 1 (meses 1): análisis y diseño. Fase 2 (meses 2): desarrollo iterativo. Fase 3 (mes 1): pruebas finales y entrega la cual se debe realizar el 26 de agosto del 2025. | Desarrollo en el laboratorio de computación de la universidad ESPE para la implementación en Galpón de crianza de conejos (Machachi) | Para automatizar procesos manuales y mejorar la eficiencia del criadero, aportando innovación al sector cunícola. | El proyecto Hoptolt tiene un costo de \$1470.90, debido a los valores totales de las máquinas utilizadas para el proyecto y el software, como el sistema operativo, que ocupan. |

Tabla 1 Marco de trabajo 5W+2H

6. Ideas a Defender

Las ideas que defender en este proyecto son las siguientes:

- **Reducción de la carga operativa diaria:** Automatización de tareas rutinarias como registro de alimentación, seguimiento de nacimientos y crecimiento de los conejos.

- **Precisión y centralización de los registros:** Almacenar toda la información en el sistema minimizando errores y duplicados. El criador puede consultar datos históricos de cada conejo al instante.
- **Gestión eficiente de jaulas y espacio:** Asignación de conejos a jaulas de forma ordenada según raza y edad, evitando la sobrepoblación.
- **Control optimizado de la alimentación:** El sistema registra las raciones suministradas a cada conejo o grupo, permitiendo ajustar dietas y evitar derroches de comida, permitiendo al criador asegurar una nutrición adecuada y reducir gastos en alimento.
- **Seguimiento de la salud:** El sistema digital centraliza y gestiona los historiales de vacunación y desparasitación de cada conejo. Esto permite al criador tener acceso inmediato a la información sobre el estado de salud de cada animal
- **Planificación y seguimiento de la reproducción:** El módulo de reproducción registra apareamientos, fechas de parto y tamaño de la camada, facilitando organizar el ciclo reproductivo y prever nacimientos
- **Registros completos de cada conejo:** Cada conejo tiene un perfil único con datos relevantes. Tener toda esa información accesible en un solo lugar ayuda al criador a obtener la información necesaria de una forma eficiente.
- **Generación de reportes claros para la toma de decisiones:**El sistema permitirá generar reportes detallados sobre la alimentación, vacunación y desparasitación de los conejos, proporcionando al criador información valiosa para la toma de decisiones..
- **Optimización de recursos y reducción de costos:** Al tener control total de la alimentación, la salud y la reproducción, el criador puede reducir desperdicios de insumos. La eficiencia del sistema contribuye a bajar los costos operativos del criadero.

7. Resultados Esperados

Al finalizar el desarrollo e implementación del sistema Hoptolt, se espera alcanzar los siguientes resultados, los cuales funcionarán como indicadores clave de calidad, eficiencia y utilidad práctica para el criador:

- Reducción significativa del uso de registros en papel, mediante la digitalización de la información relacionada con jaulas, animales, alimentación, salud y reproducción.
- Centralización y organización eficiente de los datos del criadero, permitiendo el acceso rápido a la información y mejorando la trazabilidad individual de cada conejo.
- Mejora en la precisión y consistencia de los registros, reduciendo errores humanos y la duplicación de datos.
- Aumento en la capacidad de toma de decisiones informadas, a partir de reportes generados automáticamente por el sistema, basados en datos actualizados en tiempo real.
- Incremento general en la productividad del criadero, al contar con un sistema que mejora la eficiencia operativa y reduce pérdidas.

8. Viabilidad

| Cantidad | Descripción | Valor Unitario (USD) | | Valor Total (USD) |
|----------|---|----------------------|--|-------------------|
| | Equipo en casa | | | |
| 1 | Laptop HP AMD Ryzen 7 5700U Radeon Graphics / 16gb RAM / 500gb SSD | 700 | | 700 |
| 1 | Laptop HP Intel i7-1255U Intel iris Xe Graphics / 12gb RAM / 500gb SSD | 500 | | 500 |
| 1 | Laptop Lenovo i5-5200U Intel HD 5500 Graphics / 4gb RAM / 500gb Sata | 150 | | 150 |
| | Software | | | |
| 1 | Sistema operativo Windows 11 | 120.90 | | 120.90 |
| 1 | Visual Studio Code Entorno de desarrollo (gratuito) | 0 | | 0 |
| 1 | MongoDB (open-source) | 0 | | 0 |
| 1 | Node.js / Express Framework backend (open-source) | 0 | | 0 |
| 1 | Docker Contenedores (open-source) | 0 | | 0 |
| 1 | React Framework Frondend (open-source) | | | |
| 1 | GitHub Control de versiones (plan gratuito) | 0 | | 0 |

| | | | | |
|---|---|-------|--|---------|
| 1 | Figma Prototipado de interfaz (plan gratuito) | 0 | | 0 |
| | Infraestructura | | | |
| 1 | Dominio Vercel (gratuito) | 0 | | 0 |
| | | TOTAL | | 1470.90 |

Tabla 2 Presupuesto del proyecto

8.1 Humana

8.1.1 Tutor Empresarial

La señora María Jácome, productora del criadero, cumple un rol fundamental como tutor empresarial en el desarrollo del proyecto Hoptolt. Su participación permite al equipo de desarrollo entender con precisión las dinámicas reales del entorno productivo. Entre sus principales contribuciones se encuentran la definición de los requisitos funcionales del sistema con base en sus necesidades operativas, la provisión de datos reales del criadero para pruebas y validaciones.

8.1.2 Tutor Académico

La Ingeniera Jenny Ruiz, docente de la asignatura de Análisis y Diseño de Software en la Universidad de las Fuerzas Armadas ESPE, desempeña un papel fundamental en el proyecto en desarrollo. Contribuye brindando asesoramiento para la correcta aplicación de metodologías de análisis, diseño y modelado de software.

Además, guía al equipo en la toma de decisiones técnicas, en la estructuración de los requerimientos y en la validación de las soluciones implementadas, fortaleciendo el enfoque académico y profesional del proyecto.

8.1.3 Estudiantes

- Panchez Darwin
- Proaño Jose
- Robalino Cristian

Estudiantes de la carrera de Ingeniería de Software de la ESPE y programadores novatos que se encargarán de todo lo correspondiente al análisis, diseño, desarrollo y pruebas del software del sistema.

8.2 Tecnológica

8.2.1 Hardware

Computadora personal 1 (HP Pavilion Laptop 15-eh1xxx):

| | Requisitos mínimos | Disponibilidad |
|----------------------|--|----------------|
| Procesador | AMD Ryzen 7 5700U with Radeon Graphics | Alta |
| Memoria RAM | 16 GB de RAM | Alta |
| Almacenamiento | 500 GB de espacio de almacenamiento | Alta |
| Tipo de Sistema | Sistema operativo de 64 bits, procesador basado en x64 | Alta |
| GPU | Integrada | Media |
| Procesadores Lógicos | 16 | Media |

Tabla 3 Requisitos de Hardware

Computadora personal 2 (HP 15 dyxxx) :

| | Requisitos mínimos | Disponibilidad |
|----------------|--|----------------|
| CPU | Intel i7-1255U Procesador 10 núcleos @ 1.7 GHz | Alta |
| Memoria RAM | 12 GB de RAM | Alta |
| Almacenamiento | 500 GB de espacio de almacenamiento | Alta |
| Conectividad | LAN 5 Mbps | Alta |

| | | |
|-----|-----------|-------|
| GPU | Integrada | Media |
|-----|-----------|-------|

Tabla 4 Requisitos de Hardware

Computadora personal 3 (G40-80 Laptop Lenovo - Type 80E4):

| | Requisitos mínimos | Disponibilidad |
|----------------------|--|----------------|
| CPU | Intel® Core™ i5-5200U - Procesador 2 núcleos @ 2.2 GHz | Alta |
| Memoria RAM | 4 GB de RAM | Alta |
| Almacenamiento | 500 GB de espacio de almacenamiento | Alta |
| Conectividad | LAN 5 Mbps | Alta |
| GPU | Intel HD 5500 | Media |
| Procesadores Lógicos | 4 | Media |

Tabla 5 Requisitos de Hardware

8.2.2 Software

| | Requisitos mínimos | Disponibilidad |
|-------------------|--------------------|----------------|
| Sistema Operativo | Windows 10/11 | Alta |
| IDE | Visual Studio Code | Alta |

| | | |
|-----------------------------|--------------------------------------|------|
| Lenguaje de programación | JavaScript | Alta |
| Framework Backend | Node.js \geq 12.x + Express.js 4.x | Alta |
| Biblioteca Frontend | React 17+ | Alta |
| Gestor de Base de Datos | MongoDB | Alta |
| Control de versiones | Git + GitHub | Alta |
| Documentación | Microsoft Word / Google Docs | Alta |
| Diagrama UML | Lucidchart | Alta |
| Comunicación y colaboración | Discord y WhatsApp | Alta |
| Pruebas Unitarias | JUnit (Java) / TestNG / Jest (JS) | Alta |
| Contenedores | Docker Engine | Alta |

Tabla 6 Requisitos de Software

9. Conclusiones y recomendaciones

10. Planificación para el Cronograma:

| CRONOGRAMA GRUPO 4 | | | | | |
|--------------------|---|--|-----------------|------------------|-----------|
| # | Nombre de la tarea | Asignado a | Fecha de inicio | Duración (HORAS) | Fecha fin |
| 1 | Control de calidad de requisitos funcionales con Matriz IREB | Panchez Darwin, Proaño José, Robalino | 29/4/2025 | 2 | 4/5/2025 |
| 2 | Elaborar la Matriz Ccheck List | Panchez Darwin | 29/4/2025 | 0,3 | 29/4/2025 |
| 3 | Redactar el Modelo Lista de Comprobacion | Proaño José | 29/4/2025 | 0,6 | 29/4/2025 |
| 4 | Plantear las Recomendaciones | Robalino Cristian | 29/4/2025 | 0,1 | 29/4/2025 |
| 5 | Realizar el Informe Matriz IREB | Robalino Cristian | 4/5/2025 | 1 | 4/5/2025 |
| 6 | Realizar Perfil del Proyecto | Panchez Darwin, Proaño José, Robalino Cristian | 4/5/2025 | 3 | 12/5/2025 |
| 7 | Redactar introducción del perfil de proyecto | Robalino Cristian | 4/5/2025 | 0,5 | 4/5/2025 |
| 8 | Plantear del problema Perfil del proyecto | Robalino Cristian | 6/5/2025 | 0,3 | 6/5/2025 |
| 9 | Desarrollar el Sistema de Objetivos | Robalino Cristian | 11/5/2025 | 0,3 | 12/5/2025 |
| 10 | Delimitar el Alcance | Proaño José | 11/5/2025 | 0,3 | 12/5/2025 |
| 11 | Sustentar el Marco Teórico | Proaño José | 11/5/2025 | 0,4 | 12/5/2025 |
| 12 | Proyectar las Ideas a Defender | Panchez Darwin | 11/5/2025 | 0,3 | 12/5/2025 |
| 13 | Evaluar los Resultados Esperados | Panchez Darwin | 11/5/2025 | 0,3 | 12/5/2025 |
| 14 | Realizar la Viabilidad | Panchez Darwin | 11/5/2025 | 0,4 | 12/5/2025 |
| 15 | Realizar FODA | Panchez Darwin, Proaño José, Robalino Cristian | 6/5/2025 | 1,5 | 6/5/2025 |
| 16 | Redactar los Antecedentes | Panchez Darwin | 6/5/2025 | 0,2 | 6/5/2025 |
| 17 | Plantear el Desarrollo | Proaño José | 6/5/2025 | 0,25 | 6/5/2025 |
| 18 | Realizar las Preguntas | Robalino Cristian | 6/5/2025 | 0,8 | 6/5/2025 |
| 19 | Desarrollar la Matriz con el Contexto de la Organización con las partes interesadas | Robalino Cristian | 6/5/2025 | 0,25 | 6/5/2025 |
| 20 | Elaborar Preguntas para la entrevista | Panchez Darwin, Proaño José, Robalino | 6/5/2025 | 1,5 | 7/5/2025 |
| 21 | Redactar la introducción | Robalino Cristian | 6/5/2025 | 1 | 7/5/2025 |
| 22 | Realizar el Objetivo de la Entrevista | Panchez Darwin | 6/5/2025 | 2 | 7/5/2025 |
| 23 | Plantear el Proceso Preliminar de Realización de Entrevista | Proaño José | 6/5/2025 | 2 | 7/5/2025 |
| 24 | Redacción de la justificación del Perfil de Proyecto | Proaño José | 7/5/2025 | 0,7 | 7/5/2025 |
| 25 | Planificar el cronograma para el perfil del proyecto | Panchez Darwin, Proaño José, Robalino Cristian | 8/5/2025 | 0,5 | 12/5/2025 |
| 26 | Realizar historias de usuario | Panchez Darwin, Proaño José, Robalino Cristian | 11/5/2025 | 4 | 11/5/2025 |
| 27 | Desarrollar Matriz de Marco de Trabajo de HU | Panchez Darwin, Proaño José, Robalino Cristian | 11/5/2025 | 1 | 11/5/2025 |
| 28 | Defensa del perfil de Proyecto | Panchez Darwin, Proaño José, Robalino Cristian | 13/5/2025 | 2 | 13/5/2025 |
| 29 | Ejecución de la entrevista | Panchez Darwin, Proaño José, Robalino Cristian | 14/5/2025 | 0,16 | 14/5/2025 |

Tabla 5 Cronograma del proyecto.

11. Referencias

- Blumetto, O. (2022). *Nueva tecnología para la producción de conejos de carne*. **Revista del Plan Agropecuario**, 42(2), 42–45. Departamento de Cunicultura, INIA Las Brujas.
- Weatherbed, J. (2025, May 7). Figma's big AI update takes on Adobe, WordPress, and Canva. The Verge.
<https://www.theverge.com/news/662678/figma-buzz-draw-make-sites-announce-cement-availability>

- Equipo Editorial de IONOS. (2022, 2 de marzo). *¿Qué es GitHub? – Control de versiones de un vistazo*. IONOS España. Recuperado de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/github/>
- Amazon Web Services. (s.f.). *¿Qué es Docker?* AWS. Recuperado de <https://aws.amazon.com/es/docker/>
- GeeksforGeeks. (2025, January 3). Use Case Diagram Unified Modeling Language (UML). GeeksforGeeks. <https://www.geeksforgeeks.org/use-case-diagram>
- Comparing Sails and Express: Which Node.js framework is right for your project? (2023, February 7). <https://blog.sailscasts.com/comparing-sails-and-express>
- S-O-L-I-D *Los cinco principios del diseño orientado a objetos*. (2024, September 19). SW Hosting. <https://www.swhosting.com/es/blog/s-o-l-i-d-los-cinco-principios-del-diseno-orientado-a-objetos>
- *Patrones de diseño / Design patterns*. (n.d.). Refactoring.Guru <https://refactoring.guru/es/design-patterns>

Anexos.

Anexo I. Crono

https://docs.google.com/spreadsheets/d/1ONq1Nv4Y4vGD0rp8fLpsDBuEfe_nBIUamzyziZhYv3c/edit?usp=sharing

Anexo II. MTZ Historia de Usuario

https://docs.google.com/spreadsheets/d/1JvLXrPWb9aQ_ev87RSJYYg_hZ57Xh45RhKapl9FWMHg/edit?usp=sharing

Anexo III. Video Perfil de Proyecto

<https://youtu.be/lbUqsU6YgnI>