



# Sistema de monitoreo y gestión remota del clima en invernaderos

Autor:

Lic. Martín Anibal Lacheski

Director:

Mg. Lic. Leopoldo Alfredo Zimperz (FIUBA)

Codirectora:

Dra. Lic. Nancy Beatriz Ganz (FCEQyN-UNaM)

*Esta planificación fue realizada en el curso de Gestión de proyectos entre el 20 de agosto de 2024 y el 8 de octubre de 2024.*

## Índice

1. Descripción técnica-conceptual del proyecto a realizar . . . . .	5
2. Identificación y análisis de los interesados . . . . .	6
3. Propósito del proyecto . . . . .	6
4. Alcance del proyecto . . . . .	6
5. Supuestos del proyecto. . . . .	7
6. Requerimientos . . . . .	8
7. Historias de usuarios ( <i>Product backlog</i> ). . . . .	10
8. Entregables principales del proyecto . . . . .	12
9. Desglose del trabajo en tareas . . . . .	12
10. Diagrama de Activity On Node. . . . .	15
11. Diagrama de Gantt . . . . .	15
12. Presupuesto detallado del proyecto . . . . .	18
13. Gestión de riesgos . . . . .	18
14. Gestión de la calidad . . . . .	19
15. Procesos de cierre . . . . .	20

## Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	20 de agosto de 2024
1	Se completa hasta el punto 5 inclusive	2 de septiembre de 2024
2	Se completa hasta el punto 9 inclusive	8 de septiembre de 2024

## Acta de constitución del proyecto

Buenos Aires, 20 de agosto de 2024

Por medio de la presente se acuerda con el Lic. Martín Anibal Lacheski que su Trabajo Final de la Carrera de Especialización en Internet de las Cosas se titulará “Sistema de monitoreo y gestión remota del clima en invernaderos” y consistirá en desarrollar un prototipo preliminar de un sistema basado en sensores y actuadores, junto con un servidor en la nube y una aplicación web, que permita el monitoreo y control remoto del clima en invernaderos.

El trabajo tendrá un presupuesto preliminar estimado de 615 horas y un costo estimado de \$ **XXX**, con fecha de inicio el 20 de agosto de 2024 y fecha de presentación pública el 23 de junio de 2025.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg  
Director posgrado FIUBA

Ing. Ftal. Facundo Ramos Hentz  
FCF-UNaM

Mg. Lic. Leopoldo Alfredo Zimperz  
Director del Trabajo Final

Dra. Lic. Nancy Beatriz Ganz  
Codirectora del Trabajo Final

## 1. Descripción técnica-conceptual del proyecto a realizar

La agricultura enfrenta desafíos crecientes en la optimización de la productividad y la eficiencia, especialmente en regiones con condiciones climáticas adversas y variables. Los sistemas de cultivo tradicionales suelen ser ineficientes en la gestión de recursos esenciales como agua, nutrientes y energía, en gran parte debido a la falta de monitoreo en tiempo real, lo que afecta negativamente tanto la calidad como el rendimiento de los cultivos. Además, los agricultores se enfrentan a altos costos operativos y a un impacto negativo en la sostenibilidad ambiental debido a prácticas no optimizadas. Ante estas dificultades, los cultivos hidropónicos han surgido como una solución mejorada, permitiendo una utilización más eficiente de los recursos.

El presente trabajo se desarrollará en los invernaderos de la unidad hidropónica de la Facultad de Ciencias Forestales (FCF) de la Universidad Nacional de Misiones (UNaM). En la actualidad, estos invernaderos utilizan un sistema de control con temporizadores que operan sin tener en cuenta las variables ambientales, lo que implica que la intervención humana debe ser continua y las mediciones de temperatura, potencial de Hidrógeno (pH) y conductividad eléctrica (CE) de la solución nutritiva se realiza de forma manual. Esta falta de monitoreo en tiempo real afecta negativamente la calidad y el rendimiento de los cultivos. Además, esto conlleva a altos costos operativos y un impacto negativo en la sostenibilidad ambiental debido a prácticas no optimizadas.

La propuesta, como puede observarse en la figura 1, consiste en desarrollar un conjunto de sensores y actuadores basados en el microcontrolador ESP32. Estos dispositivos se conectan a un servidor IoT en la nube a través de Wi-Fi y envían los datos mediante el protocolo MQTT (*Message Queue Telemetry Transport*). Esta configuración permitirá monitorear y gestionar el sistema de manera remota desde una aplicación web del tipo SPA (*Single Page Application*).

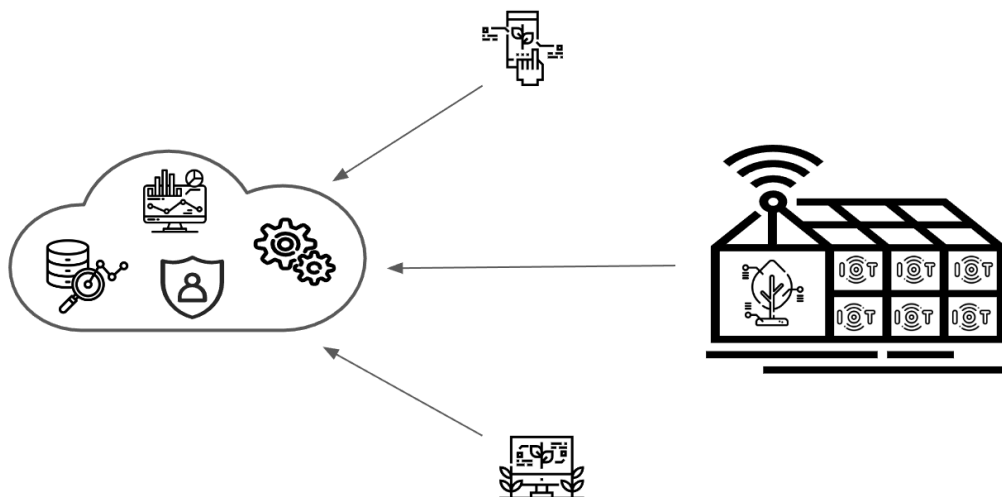


Figura 1. Diagrama en bloques del sistema.

Esta solución proporcionará un control más preciso sobre los cultivos, mejorará la eficiencia en la gestión del clima y optimizará el uso de los recursos. Esto se traducirá en una mayor productividad, una reducción de los costos operativos y una gestión más sostenible de los cultivos.

## 2. Identificación y análisis de los interesados

En el cuadro 1, se pueden identificar los participantes interesados del proyecto.

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Ing. Ftal. Facundo Ramos Hentz	FCF-UNaM	Director Proyecto
Responsable	Lic. Martín Anibal Lacheski	FIUBA	Alumno
Orientadores	Mg. Lic. Leopoldo Alfredo Zimperz	FIUBA	Director
	Dra. Lic. Nancy Beatriz Ganz	FCEQyN-UNaM	Codirectora
Opositores	Empresas que ya ofrecen soluciones similares en el mercado.		
Usuario final	Organizaciones interesadas en la automatización de cultivos en invernaderos.		

Cuadro 1. Identificación de los interesados

- Cliente: El señor Ing. Ftal. Facundo Ramos Hentz es especialista en cultivos y gestión forestal, con amplia experiencia en cultivos hidropónicos. Va a colaborar con la definición de los requerimientos y el seguimiento del proyecto.
- Orientadores:
  - El Director Mg. Lic. Leopoldo Alfredo Zimperz es experto en la temática y guiará con la implementación de los protocolos y herramientas del proyecto.
  - La Codirectora Dra. Lic. Nancy Beatriz Ganz, experta en Sistemas de Información, ayudará con el seguimiento metodológico, para garantizar una gestión rigurosa y efectiva del desarrollo del trabajo.

## 3. Propósito del proyecto

Diseñar y desarrollar un prototipo preliminar de un conjunto de sensores y actuadores conectados a través de WiFi, un servidor IoT en la nube y una aplicación web, que permita la monitorización y el control remoto del clima en invernaderos.

Este trabajo busca proporcionar a la institución un mayor control sobre las condiciones ambientales, optimizar el uso de los recursos, reducir los costos operativos y contribuir a una mayor sostenibilidad ambiental. Además, el sistema estará diseñado para apoyar la investigación académica, con el objetivo de que estudiantes, docentes e investigadores puedan utilizar los datos para desarrollar proyectos, trabajos finales y estudios específicos.

## 4. Alcance del proyecto

El presente trabajo incluye:

- Diseño e implementación de la conexión WiFi en los nodos.
- Desarrollo del firmware de los nodos.
  - Programación del firmware de los siguientes nodos sensores para la recopilación y transmisión de datos.

- Nodo sensor de temperatura ambiente, humedad relativa, presión atmosférica y nivel de luminosidad en el invernadero.
- Nodo sensor de dióxido de carbono ( $CO_2$ ).
- Nodo sensor de pH, CE y Total de Sólidos Disueltos (TDS).
- Nodo sensor de nivel y temperatura de la solución nutritiva.
- Nodo sensor de consumo de agua, nutrientes y energía eléctrica.
- Programación del firmware del nodo actuador de cuatro salidas para gestionar el control de dispositivos y reportar su estado.
- Implementación del broker MQTT en el servidor IoT.
- Implementación de cifrados de conexión mediante TLS (*Transport Layer Security*).
- Configuración del nodo central para enviar los datos recolectados al servidor IoT mediante el protocolo MQTT.
- Diseño e implementación de una base de datos para almacenar los datos recolectados por los sensores y permitir su consulta y análisis.
- Diseño y desarrollo de una API (*Application Programming Interface*) REST (*Representational State Transfer*) que permita la comunicación con el sistema utilizando HTTP (*Hypertext Transfer Protocol*), MQTT y WebSockets.
- Desarrollo de una aplicación Web del tipo SPA.
- Entrega del código del sistema, que incluye todos los componentes desarrollados (sensores, actuadores, servidor y aplicación web), guías de instalación, configuración y operación.

El presente trabajo no incluye:

- Armado de PCB, se va a utilizar gabinetes estancos, protoboard y cables para conectar los microcontroladores con los sensores y actuadores.
- Desarrollo de una aplicación móvil compatible con iOS y Android.

## 5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Se contará con los materiales necesarios para la implementación de los nodos de sensores y actuadores.
- Se contará con el apoyo financiero de la institución para comprar los componentes necesarios para el desarrollo del prototipo.
- Se dispondrá del conocimiento necesario para el desarrollo de los firmwares de los microcontroladores.
- Se dispondrá del conocimiento necesario para la implementación del protocolo de cifrado TLS en los microcontroladores y servidor IoT.

- Se dispondrá del conocimiento necesario para el diseño y desarrollo del frontend, backend, API y base de datos.
- El cliente tendrá disponibilidad para atender las consultas.
- Los directores tendrán disponibilidad para atender las consultas.
- El responsable tendrá tiempo para desarrollar el proyecto.
- Se podrá terminar el proyecto en el tiempo estipulado.

## 6. Requerimientos

### 1. Requerimientos de los nodos sensores:

- 1.1. Cada nodo deberá contar con un microcontrolador basado en ESP32.
- 1.2. Cada nodo deberá conectarse a una red Wi-Fi.
- 1.3. Cada nodo deberá tener un identificador único dentro del sistema.
- 1.4. Cada nodo deberá permitir configurar de manera remota el tiempo para enviar los datos recolectados.
- 1.5. Cada nodo, según corresponda, deberá enviar al servidor IoT:
  - 1) la temperatura del ambiente, humedad relativa, presión atmosférica y nivel de luminosidad,
  - 2) el nivel de dióxido de carbono,
  - 3) el nivel de pH, CE y TDS,
  - 4) el nivel y temperatura de la solución nutritiva,
  - 5) el consumo de agua, nutrientes y energía eléctrica.

### 2. Requerimientos de los nodos actuadores:

- 2.1. Deberá contar con un microcontrolador basado en ESP32.
- 2.2. Cada nodo deberá conectarse a una red Wi-Fi.
- 2.3. Cada nodo deberá tener un identificador único dentro del sistema.
- 2.4. Cada nodo deberá contar con cuatro canales.
- 2.5. Cada nodo deberá permitir configurar de manera remota los parámetros del actuador y cada canal.
- 2.6. Cada nodo deberá enviar al servidor IoT el estado de cada canal.
- 2.7. Cada nodo deberá permitir activar de manera remota cada canal.

### 3. Requerimientos asociados al broker MQTT:

- 3.1. Deberá contar con un broker MQTT que soporte conexiones seguras mediante TLS.
- 3.2. Deberá gestionar las suscripciones y publicaciones de los datos enviados desde los nodos y permitir la comunicación bidireccional para el envío de comandos.
- 3.3. Deberá implementar QoS (*Quality of Service*) para garantizar la entrega de los mensajes.

### 4. Requerimientos asociados al Frontend:

- 4.1. La interfaz deberá ser intuitiva y accesible desde dispositivos móviles y de escritorio.



- 4.2. Deberá permitir el acceso al sistema a usuarios debidamente autenticados.
- 4.3. Deberá permitir realizar el CRUD (*Create, Read, Update and Delete*) de ambientes.
- 4.4. Deberá permitir realizar el CRUD de sensores asociados a un determinado ambiente.
- 4.5. Deberá permitir realizar el CRUD de actuadores asociados a un determinado ambiente.
- 4.6. Deberá permitir visualizar en tiempo real los datos recibidos de los sensores y actuadores.
- 4.7. Deberá permitir enviar comandos y configuraciones a los sensores y actuadores.
- 4.8. Deberá permitir visualizar los datos históricos de los sensores y actuadores.
- 4.9. Deberá permitir visualizar, en tiempo real, un tablero con los datos de los sensores y actuadores.
5. Requerimientos asociados al Backend y la API:
  - 5.1. Deberá permitir conexiones seguras mediante TLS.
  - 5.2. Deberá poder implementar JWT (*JSON Web Token*) para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario.
  - 5.3. Deberá soportar los métodos HTTP para realizar operaciones CRUD y visualizar los reportes, WebSockets para la visualización en tiempo real de los datos y MQTT para la interacción con los sensores y actuadores.
  - 5.4. Deberá poder persistir la información de los usuarios.
  - 5.5. Deberá poder persistir la información de los ambientes.
  - 5.6. Deberá poder persistir la información de los sensores.
  - 5.7. Deberá poder persistir la información de los actuadores.
  - 5.8. Deberá poder persistir la información histórica de las mediciones de los sensores.
  - 5.9. Deberá poder persistir la información histórica de los estados de los actuadores.
  - 5.10. Deberá poder persistir la información histórica de los parámetros enviados a los sensores.
  - 5.11. Deberá poder persistir la información histórica de los parámetros enviados a los actuadores.
  - 5.12. Deberá permitir la autenticación y autorización de usuarios autorizados para acceder a los recursos.
  - 5.13. Deberá incluir los endpoints para permitir realizar el CRUD de usuarios para el acceso seguro al sistema.
  - 5.14. Deberá incluir los endpoints para permitir realizar el CRUD de ambientes.
  - 5.15. Deberá incluir los endpoints para permitir realizar el CRUD de sensores asociados a un determinado ambiente, con la configuración de los parámetros específicos.
  - 5.16. Deberá incluir los endpoints para permitir realizar el CRUD de actuadores asociados a un determinado ambiente, con la configuración de los parámetros específicos por canal.
  - 5.17. Deberá incluir el endpoint para registrar las mediciones de los sensores.
  - 5.18. Deberá incluir el endpoint para registrar los estados de los actuadores.
  - 5.19. Deberá incluir el endpoint para obtener el histórico de las mediciones de los sensores.
  - 5.20. Deberá incluir el endpoint para obtener el histórico de los estados de los actuadores.
  - 5.21. Deberá incluir un endpoint para el envío de parámetros a los sensores.

5.22. Deberá incluir un endpoint para el envío de parámetros a los actuadores.

6. Requerimientos de documentación:

- 6.1. Se entregará el código del sistema, que incluye todos los componentes desarrollados (sensores, actuadores, broker MQTT, Frontend, Backend y API).
- 6.2. Se entregarán las guías y diagramas de instalación, configuración y operación.
- 6.3. Se desarrollará un informe de avance al finalizar el Taller de Trabajo Final A.
- 6.4. Se desarrollará la memoria del proyecto al finalizar el Taller de Trabajo Final B.

## 7. Historias de usuarios (*Product backlog*)

Para definir las historias de usuario y estimar su dificultad, complejidad y riesgo, es necesario primero descomponer los requerimientos en historias de usuario específicas. A cada historia se le asignan puntos de acuerdo con la serie de Fibonacci (1, 2, 3, 5, 8, 13, 21, 34, etc.), que representan los niveles de dificultad, complejidad y riesgo.

El puntaje total de cada historia de usuario se calcula con la suma de los valores asignados a estos tres criterios, y luego se asigna el número de la secuencia de Fibonacci más cercano que sea mayor o igual a la suma obtenida.

Para asignar ponderación a los (*history points*) se han asignado los siguientes pesos:

### Dificultad del trabajo a realizar

- **Baja:** Peso  $\rightarrow$  1.
- **Media:** Peso  $\rightarrow$  3.
- **Alta:** Peso  $\rightarrow$  5.

### Complejidad del trabajo a realizar

- **Baja:** Peso  $\rightarrow$  1.
- **Media:** Peso  $\rightarrow$  5.
- **Alta:** Peso  $\rightarrow$  13.

### Riesgo del trabajo a realizar

- **Baja:** Peso  $\rightarrow$  1.
- **Media:** Peso  $\rightarrow$  3.
- **Alta:** Peso  $\rightarrow$  5.

## Historia de Usuario 1: Desarrollador

- “Como desarrollador quiero integrar los sensores y actuadores con el sistema para que se comuniquen correctamente a través de la red y asegurar que los datos se transmitan de manera eficiente y segura.”
  - **Dificultad:** Alta (5) → Porque implica múltiples integraciones y la configuración de varios dispositivos para que funcionen en conjunto de manera sincronizada.
  - **Complejidad:** Alta (13) → Porque la integración de todas las tecnologías podría presentar desafíos técnicos.
  - **Riesgo:** Alta (5) → Porque existe incertidumbre sobre posibles problemas de comunicación entre dispositivos y ajustes de la configuración.

*Story points* = 34.

(5 + 13 + 5 = 23 → 34). Es el siguiente valor en la serie de Fibonacci.

## Historia de Usuario 2: Cliente

- “Como cliente, quiero que el sistema permita la gestión completa de ambientes, sensores y actuadores a través de una interfaz intuitiva, con capacidades para visualizar en tiempo real, ajustar las configuraciones de manera remota y visualizar reportes históricos.”
  - **Dificultad:** Media (3) → Porque implica el desarrollo y personalización de funcionalidades estándar que son comunes en sistemas de monitoreo y control.
  - **Complejidad:** Alta (13) → Si bien existen soluciones de este tipo y se podría consultar a profesionales expertos en la materia, integrar todas las funcionalidades en una interfaz coherente y eficiente podría ser complejo.
  - **Riesgo:** Media (3) → Porque la integración de múltiples módulos y la gestión de datos en tiempo real presenta riesgos moderados que podrían afectar la estabilidad del sistema.

*Story points* = 21.

(3 + 13 + 3 = 19 → 21). Es el siguiente valor en la serie de Fibonacci.

## Historia de Usuario 3: Usuario final

- “Como usuario final, quiero poder visualizar un panel de control que sea fácil de usar, responsivo desde dispositivos móviles y de escritorio, que me permita gestionar y visualizar datos de sensores y actuadores en tiempo real, además de consultar reportes históricos.”
  - **Dificultad:** Media (3) → Aunque existen herramientas para el desarrollo, la integración de funcionalidades avanzadas y la necesidad de una interfaz de usuario adaptativa para diferentes dispositivos pueden añadir dificultad.
  - **Complejidad:** Alta (13) → El diseño y la implementación de una interfaz responsiva que soporte múltiples funcionalidades (monitoreo en tiempo real, datos históricos, gestión remota) pueden requerir un desarrollo más complejo.
  - **Riesgo:** Media (3) → Si bien existen las tecnologías para desarrollar el proyecto, la implementación exitosa de todas las funcionalidades sin problemas técnicos y la integración con otras partes del sistema pueden presentar riesgos.

*Story points* = 21.

$(3 + 13 + 3 = 19 \rightarrow 21)$ . Es el siguiente valor en la serie de Fibonacci.

## 8. Entregables principales del proyecto

Los entregables del proyecto son:

- Repositorio con:
  - Código fuente del firmware de los microcontroladores.
  - Código fuente de Frontend.
  - Código fuente de Backend y la API.
- Diagramas y guías de instalación.
- Guías de configuración y operación.
- Informe de avance.
- Memoria del trabajo final.
- Video con demostración del sistema.

## 9. Desglose del trabajo en tareas

1. Planificación del proyecto (50 h)
  - 1.1. Elaboración del documento de planificación del proyecto (35 h)
  - 1.2. Diseño de la arquitectura del proyecto (15 h)
2. Investigación preliminar (80 h)
  - 2.1. Investigación de los protocolos MQTT, HTTP y WebSockets (12 h)
  - 2.2. Investigación de certificado TLS (8 h)
  - 2.3. Investigación y elección de los microcontroladores a utilizar (5 h)
  - 2.4. Investigación y elección de los sensores y actuadores a utilizar (10 h)
  - 2.5. Investigación y elección de la base de datos a utilizar (10 h)
  - 2.6. Investigación y elección del framework de backend a utilizar (10 h)
  - 2.7. Investigación y elección del framework de frontend a utilizar (10 h)
  - 2.8. Investigación y elección de la plataforma del servidor IoT a utilizar (10 h)
  - 2.9. Instalación y puesta a punto del entorno de desarrollo (5 h)
3. Prototipado del proyecto (30 h)
  - 3.1. Configuración y conexionado de sensores y actuadores (15 h)
  - 3.2. Integración de componentes y microcontroladores (15 h)
4. Desarrollo del firmware de los microcontroladores (127 h)

- 4.1. Configuración inicial de los microcontroladores (2 h)
- 4.2. Implementación y configuración de los certificados TLS (10 h)
- 4.3. Implementación del firmware del nodo sensor de temperatura ambiente, humedad relativa, presión atmosférica y nivel de luminosidad (20 h)
- 4.4. Implementación del firmware del nodo sensor de nivel de dióxido de carbono (15 h)
- 4.5. Implementación del firmware del nodo sensor de pH, CE y TDS (20 h)
- 4.6. Implementación del firmware del nodo sensor de nivel y temperatura de la solución nutritiva (20 h)
- 4.7. Implementación del firmware del nodo sensor de consumo de agua, nutrientes y energía eléctrica (20 h)
- 4.8. Implementación del firmware del nodo actuador (20 h)
5. Desarrollo del backend (90 h)
  - 5.1. Configuración inicial del entorno (10 h)
  - 5.2. Implementación y configuración del broker MQTT (10 h)
  - 5.3. Implementación y configuración de los certificados TLS (10 h)
  - 5.4. Implementación y configuración de JWT (10 h)
  - 5.5. Implementación y configuración de la base de datos (10 h)
  - 5.6. Implementación y configuración de WebSockets (10 h)
  - 5.7. Implementación de los endpoints de usuarios (4 h)
  - 5.8. Pruebas de los endpoints de usuarios (1 h)
  - 5.9. Implementación de los endpoints de ambientes (2 h)
  - 5.10. Pruebas de los endpoints de ambientes (1 h)
  - 5.11. Implementación de los endpoints de sensores (4 h)
  - 5.12. Pruebas de los endpoints de sensores (1 h)
  - 5.13. Implementación de los endpoints de actuadores (4 h)
  - 5.14. Pruebas de los endpoints de actuadores (1 h)
  - 5.15. Implementación del endpoint de mediciones de sensores (2 h)
  - 5.16. Pruebas del endpoint de mediciones de sensores (1 h)
  - 5.17. Implementación del endpoint de estados de actuadores (2 h)
  - 5.18. Pruebas del endpoint de estados de actuadores (1 h)
  - 5.19. Implementación del endpoint de parámetros enviados a los sensores (2 h)
  - 5.20. Pruebas del endpoint de parámetros enviados a los sensores (1 h)
  - 5.21. Implementación del endpoint de parámetros enviados a los actuadores (2 h)
  - 5.22. Pruebas del endpoint de parámetros enviados a los actuadores (1 h)
6. Desarrollo del frontend (93 h)
  - 6.1. Configuración inicial del entorno (10 h)
  - 6.2. Implementación y configuración de los certificados TLS (10 h)
  - 6.3. Implementación y configuración de JWT (10 h)
  - 6.4. Implementación y configuración de WebSockets (5 h)
  - 6.5. Implementación de la interfaz de login de usuario (4 h)

- 6.6. Pruebas de la interfaz de login de usuario (1 h)
- 6.7. Implementación de las interfaces de manejo de usuarios (4 h)
- 6.8. Pruebas de las interfaces de manejo de usuarios (1 h)
- 6.9. Implementación de las interfaces de manejo de ambientes (3 h)
- 6.10. Pruebas de las interfaces de manejo de ambientes (1 h)
- 6.11. Implementación de las interfaces de manejo de sensores (5 h)
- 6.12. Pruebas de las interfaces de manejo de sensores (1 h)
- 6.13. Implementación de las interfaces de manejo de actuadores (5 h)
- 6.14. Pruebas de las interfaces de manejo de actuadores (1 h)
- 6.15. Implementación de la interfaz de mediciones de sensores (8 h)
- 6.16. Pruebas de la interfaz de mediciones de sensores (1 h)
- 6.17. Implementación de la interfaz de estados de actuadores (8 h)
- 6.18. Pruebas de la interfaz de estados de actuadores (1 h)
- 6.19. Implementación de la interfaz de parámetros enviados a los sensores (3 h)
- 6.20. Pruebas de la interfaz de parámetros enviados a los sensores (1 h)
- 6.21. Implementación de la interfaz de parámetros enviados a los actuadores (3 h)
- 6.22. Pruebas de la interfaz de parámetros enviados a los actuadores (1 h)
- 6.23. Implementación de la interfaz principal del sistema (5 h)
- 6.24. Pruebas de la interfaz principal del sistema (1 h)
- 7. Pruebas y validación (48 h)
  - 7.1. Pruebas de los microcontroladores
    - 1) Pruebas de los firmwares de los microcontroladores (5 h)
    - 2) Pruebas de conexión y comunicación de los microcontroladores (5 h)
    - 3) Pruebas de seguridad (5 h)
    - 4) Pruebas de sensores y actuadores (5 h)
  - 7.2. Pruebas del backend
    - 1) Pruebas de MQTT y TLS (3 h)
    - 2) Pruebas de validación de almacenamiento de los datos (3 h)
    - 3) Pruebas de manejo de los datos en tiempo real y endpoints (3 h)
    - 4) Pruebas de seguridad (5 h)
  - 7.3. Pruebas del frontend
    - 1) Pruebas de interfaz y usabilidad (3 h)
    - 2) Pruebas de visualización de los reportes y gráficos (3 h)
    - 3) Pruebas de visualización de los datos en tiempo real (3 h)
    - 4) Pruebas de seguridad (5 h)
- 8. Documentación del proyecto (97 h)
  - 8.1. Informe de avance (15 h)
  - 8.2. Video con demostración del sistema (2 h)
  - 8.3. Estructura de la memoria del proyecto (10 h)
  - 8.4. Escritura de la memoria del proyecto (40 h)
  - 8.5. Revisión de la memoria del proyecto (10 h)
  - 8.6. Presentación final (20 h)

Cantidad total de horas: 615.

## 10. Diagrama de Activity On Node

Armado el AoN a partir del WBS definido en la etapa anterior.

Una herramienta simple para desarrollar los diagramas es el Draw.io (<https://app.diagrams.net/>). Draw.io

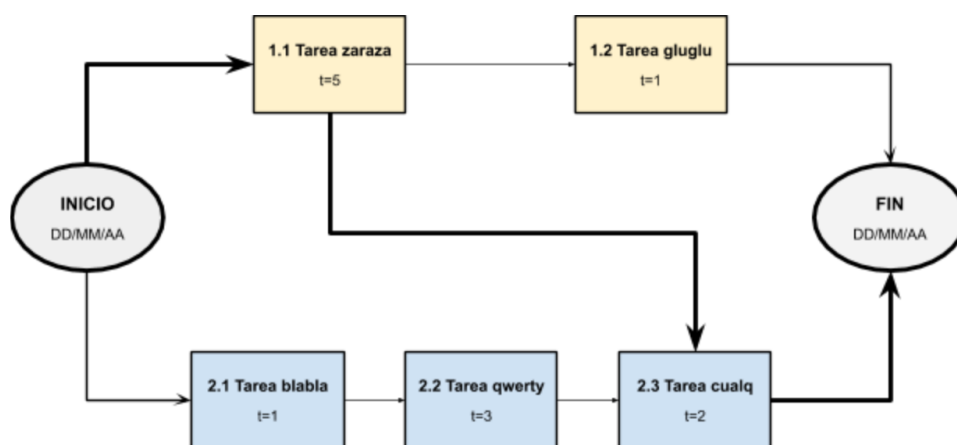


Figura 2. Diagrama de *Activity on Node*.

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semi críticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color.

## 11. Diagrama de Gantt

Existen muchos programas y recursos *online* para hacer diagramas de Gantt, entre los cuales destacamos:

- Planner
- GanttProject
- Trello + *plugins*. En el siguiente link hay un tutorial oficial:  
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- Creately, herramienta online colaborativa.  
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>
- Se puede hacer en latex con el paquete *pgfgantt*  
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Pegar acá una captura de pantalla del diagrama de Gantt, cuidando que la letra sea suficientemente grande como para ser legible. Si el diagrama queda demasiado ancho, se puede pegar primero la “tabla” del Gantt y luego pegar la parte del diagrama de barras del diagrama de Gantt.

Configurar el software para que en la parte de la tabla muestre los códigos del EDT (WBS).  
Configurar el software para que al lado de cada barra muestre el nombre de cada tarea.  
Revisar que la fecha de finalización coincida con lo indicado en el Acta Constitutiva.

En la figura 3, se muestra un ejemplo de diagrama de gantt realizado con el paquete de *pgfgantt*. En la plantilla pueden ver el código que lo genera y usarlo de base para construir el propio.

Las fechas pueden ser calculadas utilizando alguna de las herramientas antes citadas. Sin embargo, el siguiente ejemplo fue elaborado utilizando esta hoja de cálculo.

Es importante destacar que el ancho del diagrama estará dado por la longitud del texto utilizado para las tareas (Ejemplo: tarea 1, tarea 2, etcétera) y el valor *x unit*. Para mejorar la apariencia del diagrama, es necesario ajustar este valor y, quizás, acortar los nombres de las tareas.

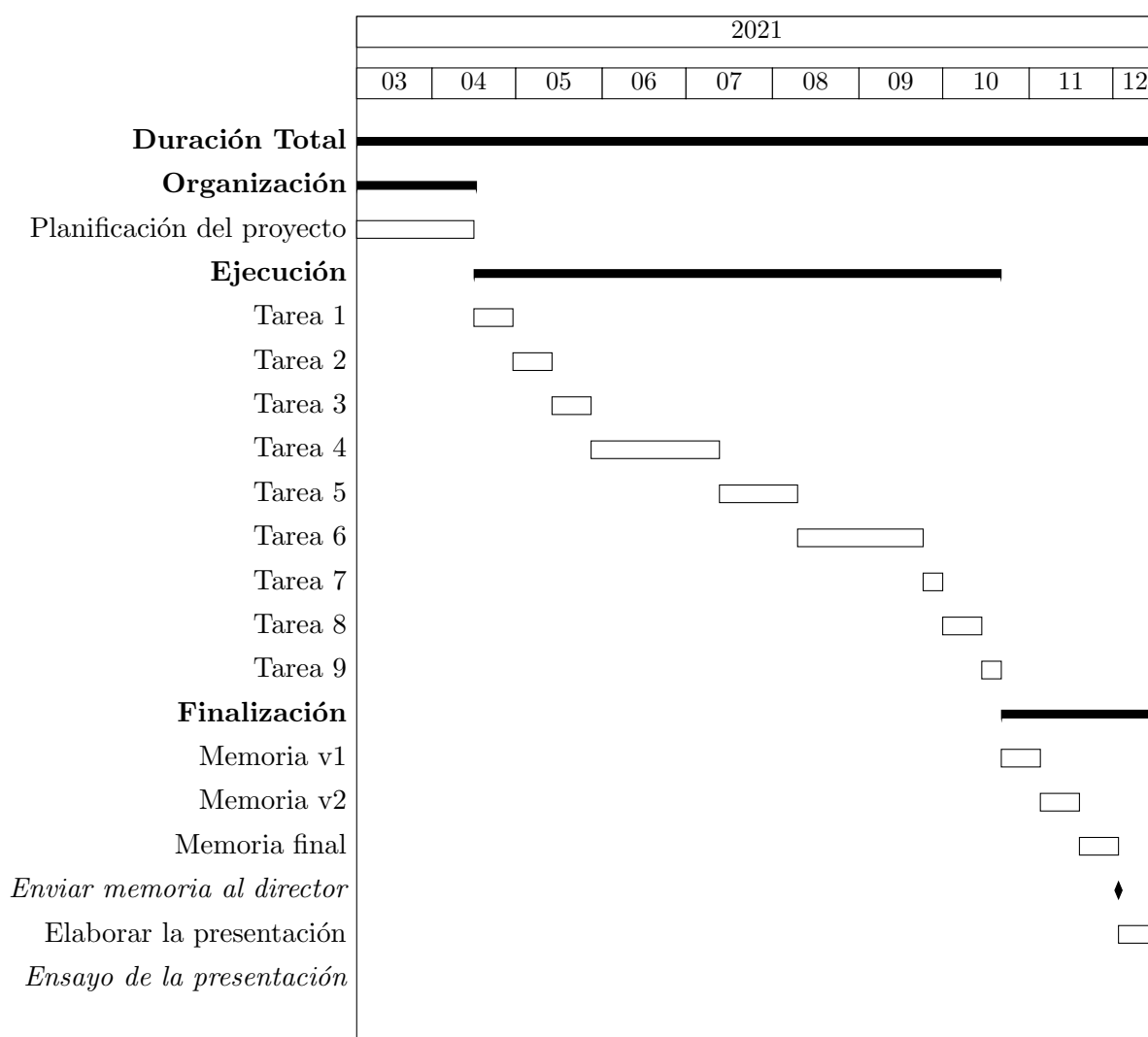


Figura 3. Diagrama de gantt de ejemplo



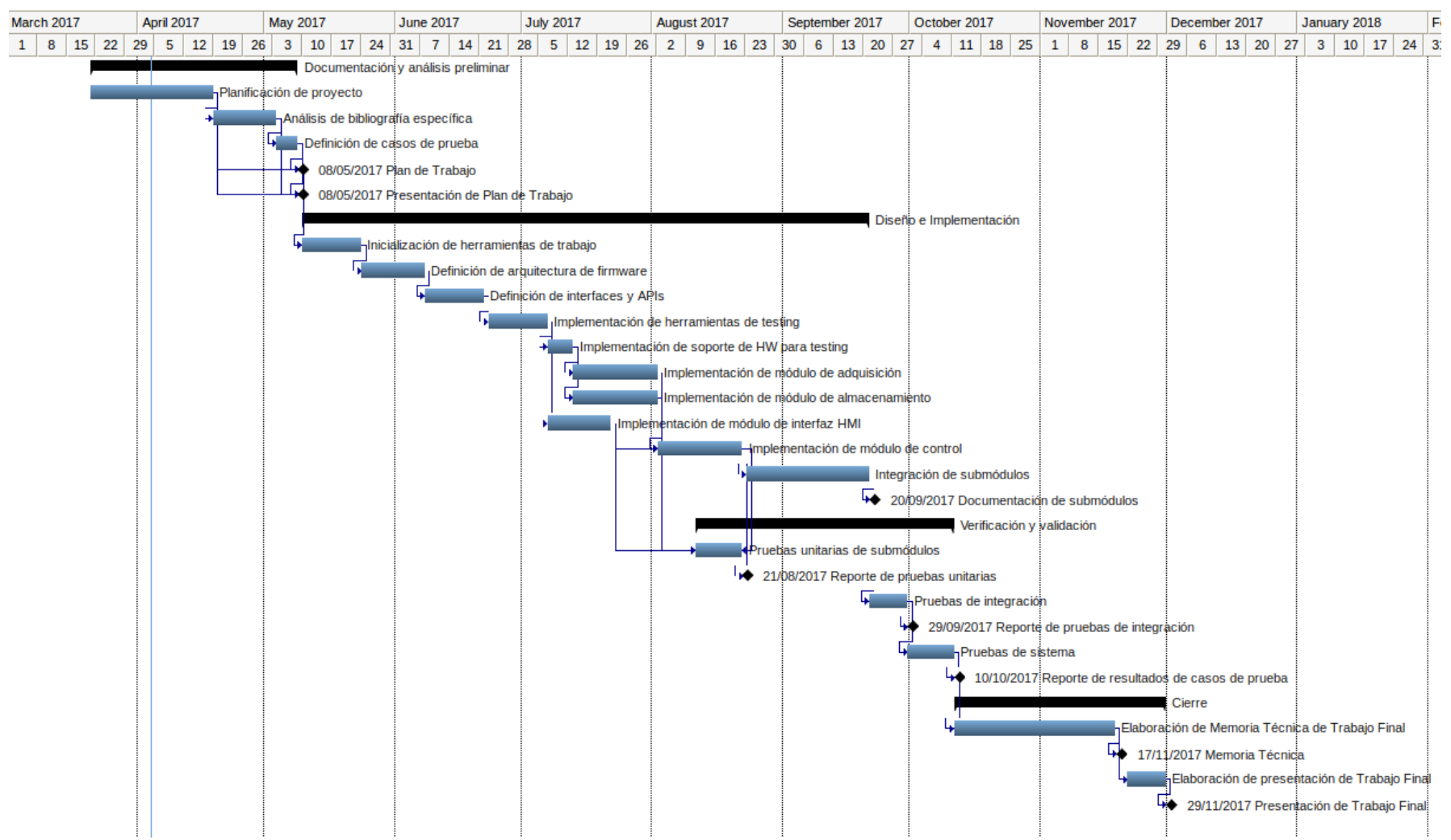


Figura 4. Ejemplo de diagrama de Gantt (apaisado).

## 12. Presupuesto detallado del proyecto

Si el proyecto es complejo entonces separarlo en partes:

- Un total global, indicando el subtotal acumulado por cada una de las áreas.
- El desglose detallado del subtotal de cada una de las áreas.

**IMPORTANTE:** No olvidarse de considerar los **COSTOS INDIRECTOS**.

Incluir la aclaración de si se emplea como moneda el peso argentino (ARS) o si se usa moneda extranjera (USD, EUR, etc). Si es en moneda extranjera se debe indicar la tasa de conversión respecto a la moneda local en una fecha dada.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
SUBTOTAL			
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
SUBTOTAL			
TOTAL			

## 13. Gestión de riesgos

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

Riesgo 1: detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2:

- Severidad (S): X.  
Justificación...

- Ocurriencia (O): Y.  
Justificación...

Riesgo 3:

- Severidad (S): X.  
Justificación...
- Ocurriencia (O): Y.  
Justificación...

b) Tabla de gestión de riesgos: (El RPN se calcula como  $RPN=S \times O$ )

Riesgo	S	O	RPN	S*	O*	RPN*

Criterio adoptado:

Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (\*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).  
Nueva asignación de S y O, con su respectiva justificación:

- Severidad (S\*): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O\*): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

## 14. Gestión de la calidad

Elija al menos diez requerimientos que a su criterio sean los más importantes/críticos/que aportan más valor y para cada uno de ellos indique las acciones de verificación y validación que permitan asegurar su cumplimiento.

- Req #1: copiar acá el requerimiento con su correspondiente número.

- Verificación para confirmar si se cumplió con lo requerido antes de mostrar el sistema al cliente. Detallar.
- Validación con el cliente para confirmar que está de acuerdo en que se cumplió con lo requerido. Detallar.

Tener en cuenta que en este contexto se pueden mencionar simulaciones, cálculos, revisión de hojas de datos, consulta con expertos, mediciones, etc.

Las acciones de verificación suelen considerar al entregable como “caja blanca”, es decir se conoce en profundidad su funcionamiento interno.

En cambio, las acciones de validación suelen considerar al entregable como “caja negra”, es decir, que no se conocen los detalles de su funcionamiento interno.

## 15. Procesos de cierre

Establecer las pautas de trabajo para realizar una reunión final de evaluación del proyecto, tal que contemple las siguientes actividades:

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original:
  - Indicar quién se ocupará de hacer esto y cuál será el procedimiento a aplicar.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, los problemas que surgieron y cómo se solucionaron:
  - Indicar quién se ocupará de hacer esto y cuál será el procedimiento para dejar registro.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores:
  - Indicar esto y quién financiará los gastos correspondientes.