

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



Sistema de monitoreo y control de un acuario “Salvando a Nemo”

Autor(es):

- Dylan Flores
- Cristian Gutiérrez
- Joaquin Jelves
- Cristobal Hernández

Asignatura: Proyecto 2

Académico: Diego Aracena

Arica, 28 OCTUBRE 2025

Historial de cambios

| Fecha | Versión | Descripción | Autores |
|------------|---------|---|---|
| 30/09/2025 | 1.0 | Versión preliminar del formato | Dylan flores Cristobal Hernandez Joaquin Jelves Cristian Gutierrez |
| 07/10/2025 | 1.1 | Se estructuró el formato inicial del informe | Dylan flores Cristobal Hernandez Joaquin Jelves Cristian Gutierrez |
| 14/10/2025 | 1.2 | Panorama General y Organización del Proyecto | Dylan flores Cristobal Hernandez Joaquin Jelves Cristian Gutierrez |
| 21/10/2025 | 1.3 | Estimación de Costos y Planificación de Riesgos | Dylan flores Cristobal Hernandez Joaquin Jelves Cristian Gutierrez |



Tabla de contenido.

| | |
|---|-----------|
| Historial de cambios..... | 2 |
| 1. Panorama General..... | 4 |
| 1.1. Resumen del Proyecto..... | 4 |
| 1.1.1. Propósito..... | 5 |
| 1.1.2. Alcance..... | 5 |
| 1.1.3. Objetivo General..... | 5 |
| 1.1.4. Objetivos Específicos..... | 5 |
| 1.1.5. Suposiciones y restricciones..... | 6 |
| 1.1.6. Entregables del Proyecto..... | 6 |
| 2. Organización del proyecto..... | 7 |
| 2.1. Personal y entidades internas..... | 7 |
| 2.2. Roles y responsabilidades..... | 7 |
| 2.3. Mecanismos de comunicación y organización..... | 8 |
| 3. Planificación de los procesos de gestión..... | 9 |
| 3.1. Planificación inicial del proyecto..... | 9 |
| 3.1.1. Planificación de estimaciones..... | 9 |
| 3.1.2. Planificación de Recursos Humanos..... | 10 |
| 3.1.3. Costos totales..... | 11 |
| 3.2. Distribución de tiempos..... | 12 |
| 3.2.1. Carta Gantt..... | 12 |
| 3.2.2. Asignación de tiempo..... | 12 |
| 3.3. Planificación de la gestión de riesgos..... | 13 |
| 4. Conclusión..... | 14 |
| 5. Referencias..... | 15 |



Índice de figuras y tablas

1. Panorama General

1.1. Resumen del Proyecto

El proyecto se enfoca en resolver las significativas dificultades inherentes al cuidado y mantenimiento de acuarios, una tarea que a menudo se convierte en un desafío considerable, no solo para los aficionados principiantes, sino también para negocios que dependen de ecosistemas acuáticos saludables, como tiendas de mascotas o establecimientos de restauración. La estabilidad del entorno acuático es absolutamente vital para el bienestar y la supervivencia de los peces y otras especies, pero la realidad actual implica que las cruciales tareas de supervisión, así como el ajuste de parámetros clave —tales como la temperatura, el nivel de pH, la intensidad de la luz, y la concentración de nitratos o la conductividad—, se realizan de manera tediosa, manual y con una alta dependencia del conocimiento técnico del cuidador. La problemática fundamental que identifica este proyecto es la notable carencia de un sistema que sea a la vez accesible, completamente automatizado y suficientemente confiable para garantizar un entorno acuático estable de forma continua, simplificando radicalmente las exigencias del mantenimiento.

Justificar
ambos
lados

La solución propuesta para este problema es el diseño y el desarrollo de un prototipo de sistema de monitoreo y automatización que se apalanca en el poder del Internet de las Cosas (IoT). Este sistema innovador, que hemos denominado "Salvando a Nemo", utiliza una Raspberry Pi 4 como su núcleo de procesamiento y servidor central. La Raspberry Pi 4 no solo es una plataforma potente y de bajo consumo, sino que es esencial para centralizar la recopilación y el procesamiento de todos los datos críticos. Este sistema integra una variedad de sensores especializados de alta precisión diseñados para medir en tiempo real las condiciones vitales del acuario, detectando inmediatamente cualquier variación en parámetros cruciales como son la temperatura del agua, el nivel de acidez o pH, y la intensidad de la luz.

Los datos recogidos por esta red de sensores son procesados continuamente por la Raspberry Pi, la cual no solo registra los valores, sino que también ejecuta la lógica de control necesaria. Esta capacidad de procesamiento en el borde es esencial para la funcionalidad de "Salvando a Nemo", permitiendo tomar decisiones automatizadas y rápidas. Por ejemplo, si la temperatura cae por debajo de un umbral preestablecido, el sistema puede activar automáticamente un calentador conectado, o si el pH se desvía peligrosamente, puede enviar una alerta inmediata al usuario. Además, el proyecto contempla la implementación de una interfaz de usuario accesible, probablemente a través de una aplicación móvil o un *dashboard* web, donde los propietarios podrán visualizar los datos históricos y en tiempo real de su acuario desde cualquier ubicación, recibiendo notificaciones y alertas instantáneas sobre cualquier anomalía.

1.1.1. Propósito

El propósito de este proyecto es brindar una solución tecnológica que facilite el cuidado y mantenimiento de acuarios, tanto para personas con escaso conocimiento en este ámbito como para establecimientos que requieran un sistema automatizado de monitoreo, tales como tiendas de mascotas, restaurantes u otros negocios. El sistema busca garantizar el bienestar de los peces mediante la regulación y supervisión automática de los parámetros vitales del agua, simplificando las tareas de mantenimiento y promoviendo un ecosistema acuático estable mediante su monitoreo.

1.1.2. Alcance

Nuestro proyecto está diseñado para tener un alcance variado, desde entusiastas de la marina hasta pequeños negocios. **Comenzaremos** con el uso personal, ofreciendo soluciones para acuarios domésticos, como guías sencillas, monitoreo del agua y automatización. Luego, **nos expandiremos** a tiendas de mascotas (pet shops) para mejorar la exhibición y el cuidado de sus peces, y también a restaurantes que deseen acuarios atractivos en su ambiente. **Ofreceremos** desde el diseño y la instalación hasta el mantenimiento. El objetivo es que todos puedan acceder a las últimas innovaciones en acuariofilia.

1.1.3. Objetivo General

Diseñar y hacer funcional un prototipo de sistema de monitoreo y automatización para acuarios, usando una Raspberry Pi 4, que regule solo y autónomamente los parámetros vitales del agua (temperatura, el pH y el nivel de agua y luz) y que permite la supervisión de esos datos en remoto por una interfaz que usa la gente, todo esto con el fin de garantizar un ecosistema acuático que es estable y que simplificará la labor de mantenimiento para el aficionados que están comenzando.

1.1.4. Objetivos Específicos

- Seleccionar y caracterizar los componente de hardware, incluso los sensores para la mediciones de temperatura (el DS18B20), el PH y nivel de el agua (el interruptor de flotador), y los actuador para el controlar el calefactor, la bomba del agua y el alimentador automático que da comida.
- Diseñar y ensamblar el circuito electrónico que interconecta de una forma segura a la Raspberry Pi 4 con todos los sensores y los módulos de control (los relés y servomotor), para asegurar la integridad de todos los componentes de bajo voltaje.
- Desarrollar el software para el control en el lenguaje Python para la Raspberry Pi 4, que implementa la lógica que es necesario para:
 - Leer y analizar la interpretación de los datos de los sensores.
 - Comparar las mediciones con umbrales que son de seguridad definida antes.
 - Activar los actuadores que son lo correcto de manera automática.
- Construir e integrar la estructura físico del prototipo, utilizando la maqueta 3D para guiar y organizar todos los componentes de hardware de una manera que funciona y que es seguro para el entornos acuático.

- Implementar una interfaz de usuario (el panel web) que se pueda acceder desde la red local o el internet, que dará permiso al usuario para visualizar el estado de los parámetros del acuario en tiempo real, recibirán notificaciones y control manual los actuadores.
- Hacer pruebas completas del sistema en un entorno de acuarios para validar la precisión de las medidas que se hacen, la fiabilidad de las acciones que son automáticas y el correcto funcionamiento de la interfaz que es remota, documentando los resultados obtenidos.

1.1.5. Suposiciones y restricciones

- Suposiciones:
 - Se asume que todos los equipos actuadores (calefactores...) funcionarán correctamente y habrá un plan de mantenimiento regular.
 - Se asume que el sistema de monitoreo podrá alertar a los usuarios de cambios críticos en tiempo real.
- Restricciones:
 - Puede haber limitaciones en el conocimiento del usuario sobre el manejo y el mantenimiento adecuado del sistema de control y monitoreo.
 - El sistema puede ser susceptible a fallos tecnológicos o cortes de energía, lo que puede afectar el monitoreo continuo.

1.1.6. Entregables del Proyecto

Los entregables del proyecto son:

1. Maqueta del sistema
2. Presentaciones
3. Informes
4. Redmine UTA (wiki, bitácoras y carta Gantt)
5. Video explicativo de la maqueta del proyecto en VR
6. Manual de usuario
7. Poster
8. Sistema de "Salvando a Nemo"

2. Organización del proyecto

2.1. Personal y entidades internas.

Para **organizarnos definimos** algunos roles para mejor organización:

Jefe de grupo: Persona encargada de la coordinación general del proyecto. Define los objetivos, asigna tareas, supervisa el cronograma, y es el principal punto de contacto para la toma de decisiones. Es el líder.

Programador: Responsable de escribir, probar y mantener el código que se ejecutará en Raspberry Pi 4. Esto incluye la configuración del sistema operativo de la Raspberry Pi, la programación de la lógica principal del proyecto, y la integración de librerías para interactuar con sensores o periféricos.

Documentador: Encargado de crear y mantener toda la documentación escrita del proyecto. Esto abarca desde el diseño inicial, las bitácoras, informes, presentaciones, administración de Redmine (diseño Wiki, subir archivos y carta Gantt) , hasta el manual de usuario.

Técnico de hardware: Responsable del montaje físico, la conexión de componentes (sensores, actuadores) a los pines de la Raspberry Pi, la gestión de la alimentación, y la identificación/solución de problemas físicos o de interfaz electrónica. Se asegura de que la configuración física sea robusta y funcional.

2.2. Roles y responsabilidades

Los roles de cada integrantes son:

| Rol | Responsable |
|---------------------|--|
| Jefe de Grupo | Cristian Gutierrez |
| Programador | Dylan Flores - Cristobal Hernández |
| Documentador | Cristian Gutierrez - Cristobal Hernández |
| Técnico de hardware | Joaquin Jelves |

Tabla 1: Roles y Responsables

2.3. Mecanismos de comunicación y organización

WhatsApp: Se usa para la organización y principal fuente de comunicación. Aquí se ponen de acuerdo con los horarios de las reuniones y se comparten cosas importantes al instante, como enlaces, archivos o referencias.

Redmine: Funciona como la oficina principal del proyecto. Aquí se guarda toda la documentación, se lleva el control de las tareas con el calendario y el "plan de trabajo" (Carta Gantt), y se presenta la información general del proyecto (como una enciclopedia interna o *wiki*).

Google Drive: Es el almacén de archivos del equipo. Ayuda a organizar todos los documentos, enlaces, videos y referencias del proyecto, haciendo que sea fácil acceder a ellos y trabajar en ellos al mismo tiempo.

GitHub: Control de versiones del código del proyecto. Es el mecanismo principal para que el Programador gestione, colabore y fusione cambios en el código de forma segura, evitando sobrescribir el trabajo de otros.

3. Planificación de los procesos de gestión

3.1. Planificación inicial del proyecto

En la parte de hardware, se utilizarán los siguientes productos:

- Computadores
- Smartphone
- Raspberry Pi 4
- Kit GrovePi o Gpio
- Tarjeta SD 32 Gb
- Sensores
- Accionadores

Mientras que para la parte de software:

- Visual Studio Code
- Unity Hub
- Blender
- Canva
- Meta Link
- Meta Link Developer

oh!! lenguaje de programación te olvidaron

3.1.1. Planificación de estimaciones

- Costos de Hardware:

| Producto | Cantidad | Costo Unidad | Costo Total |
|----------------------------|----------|----------------------|------------------|
| Computador(personal) | 4 | \$0 | \$0 |
| Raspberry Pi 4 | 1 | \$110.000 | \$110.000 |
| Kit GrovePi | 1 | \$132.000 | \$132.000 |
| Pantalla LCD | 1 | (Incluida en el Kit) | - |
| Sensor Ultrasónico | 1 | (Incluida en el Kit) | - |
| Sensor de Luz | 1 | (Incluida en el Kit) | - |
| Luz Led | 1 | (Incluida en el Kit) | - |
| Tarjeta SD 32 GB | 1 | \$8.000 | \$8.000 |
| Sensor Temperatura DS18B20 | 1 | \$ 1.990 | \$ 1.990 |
| Sensor de Ph | 1 | \$ 24.990 | \$ 24.990 |
| Calefactor | 1 | \$6.990 | \$6.990 |
| Monitor LG | 1 | \$119.990 | \$119.990 |
| Total | | | \$403.960 |

tienen costos

Tabla 2: Costos Hardware

- Costos de Software:

| Producto | Costo | Costo Total |
|---------------------|-------|-------------|
| Visual Studio Code | \$0 | \$0 |
| Canva | \$0 | \$0 |
| Documentos Google | \$0 | \$0 |
| Unity Hub | \$0 | \$0 |
| Blender | \$0 | \$0 |
| Meta Link | \$0 | \$0 |
| Meta Link Developer | \$0 | \$0 |
| Raspberry PI OS | \$0 | \$0 |
| Total | | \$0 |

leng.pgm

Tabla 3: Costos Software

3.1.2. Planificación de Recursos Humanos

- Costos de Recursos Humanos:

| Rol | Cantidad por rol | Costo/Hora | Horas Mensuales Totales | Costo Total |
|--------------------------|------------------|------------|-------------------------|---------------------|
| Jefe de Grupo | 1 | \$17.000 | 48 | \$816.000 |
| Programador | 2 | \$9.500 | 48 | \$672.000 |
| Documentador | 2 | \$7.000 | 48 | \$768.000 |
| Técnico de Hardware | 1 | \$8.000 | 48 | \$384.000 |
| Total por 1 mes | | | | \$2.640.000 |
| Total por 4 meses | | | | \$10.560.000 |

Tabla 4: Costos Recursos Humanos

son caritos, en el postular no hay engaño

3.1.3. Costos totales

- Costo Totales

| Elemento | Costo Total |
|------------------|---------------------|
| Hardware | \$403.960 |
| Software | \$0 |
| Recursos Humanos | \$10.560.000 |
| Total | \$10.963.960 |




Tabla 5: Costos Totales

3.2. Distribución de tiempos

3.2.1. Carta Gantt

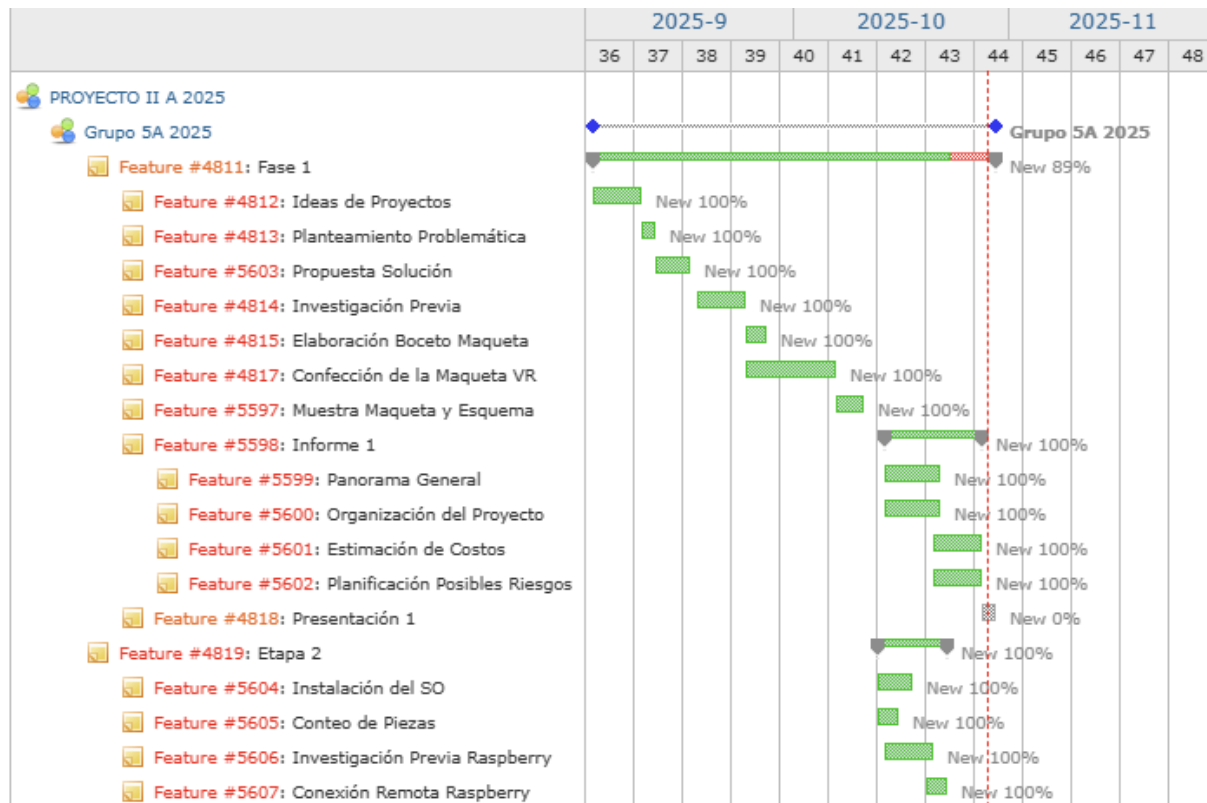


Ilustración 1: Carta Gantt

Bien a medias, se debe estimar todo los procesos

3.2.2. Asignación de tiempo

El proyecto está planificado para desarrollarse a lo largo de 15 semanas, con una distribución de trabajo semanal que incluye 6 horas de trabajo en clases y 6 horas de trabajo autónomo, lo que da un total estimado de 180 horas de trabajo. Estas 15 semanas se han dividido en 3 fases, las cuales se encuentran en la siguiente tabla.

| Fase | Semanas Transcurridas | Fecha |
|------|---|--|
| 1 | Desde la 1° semana hasta la 8° semana | Del 02 de septiembre hasta el 28 de octubre |
| 2 | Desde la 9° semana hasta la 12° semana | Del 29 de octubre hasta el 25 de noviembre |
| 3 | Desde la 13° semana hasta la 15° semana | Del 26 de noviembre hasta el 16 de diciembre |

Tabla 6: Asignación de tiempos

3.3. Planificación de la gestión de riesgos

Se elabora una tabla para establecer y analizar los posibles riesgos del proyecto y las medidas a tomar para solucionarlos, según su nivel de impacto. Los niveles de impacto son:

1. Catastrófico
2. Crítico
3. Marginal
4. Despreciable

| Riesgos | Probabilidad de Ocurrencia | Nivel de Impacto | Acción Remedial |
|---|----------------------------|------------------|---|
| Pérdida de calibración de sensores | 60% | 2 | Programar una alerta en la interfaz de usuario que recuerde "Calibrar Sonda de pH" cada 15 días. |
| Corrupción de la Tarjeta SD de la Raspberry Pi | 50% | 1 | Programar copias de seguridad (backups) automáticas del código a un repositorio Git. |
| Fallo de la red Wi-Fi | 50% | 3 | Diseñar el sistema para que sea autónomo. El control vital (calefactor, nivel) debe seguir funcionando localmente en la Raspberry Pi aunque no tenga internet. |
| Indisponibilidad de un miembro clave | 50% | 3 | Fomentar la documentación interna y el trabajo en pares. Usar un repositorio centralizado (como GitHub) donde todo el código está comentado, para que otro programador pueda tomar el relevo si es necesario. |
| Consumo excesivo de recursos de la Raspberry Pi | 20% | 3 | Monitorear el uso de CPU y memoria de la Raspberry Pi durante las pruebas. |
| Desgaste acelerado por corrosión | 10% | 2 | Aislar. Asegurarse de que toda la electrónica (Raspberry Pi, relés) esté en una caja cerrada y lejos de las salpicaduras. |

Tabla 7: Gestión de Riesgos

4. Conclusión

El desarrollo del proyecto “Salvando a Nemo” permitió aplicar conocimientos teóricos y prácticos en el diseño e implementación de un sistema de monitoreo automatizado basado en tecnologías IoT. A través del uso de una Raspberry Pi 4 y diversos sensores especializados, se logró crear una solución capaz de medir y controlar parámetros críticos del agua, como la temperatura, el pH y el nivel, garantizando así un ambiente estable y saludable para las especies acuáticas.

Durante el proceso se fortalecieron habilidades de trabajo en equipo, planificación, programación, electrónica y gestión de riesgos, logrando una integración efectiva entre el hardware y el software. Asimismo, se diseñó una interfaz accesible que facilita la supervisión remota y el control del acuario, aportando una experiencia tecnológica intuitiva y confiable para el usuario final.

En conjunto, este proyecto demuestra que la automatización puede simplificar tareas tradicionalmente complejas, ofreciendo una herramienta innovadora y económica que mejora el bienestar de los peces y la eficiencia en el mantenimiento de acuarios. A futuro, se podrían incorporar nuevas funcionalidades, como el control mediante inteligencia artificial o la conexión con aplicaciones móviles avanzadas, para continuar evolucionando el sistema hacia un producto de uso doméstico y comercial completamente inteligente.

Obs:

Informe muy desordenado, en contenido está bien y detallado..

Faltantes, las figuras se numeran y se referencian del texto, las tablas también. Hay ausencia de figuras de lo trabajado en XR

Se debe escribir en tercera persona

Falta definir el lenguaje de programación

se debe rehacer la carta gantt, los costos de personal y recursos

5. Referencias

- [1] Aquaforest. (2023, Junio 19). Water parameters in tank - How to use water tests? Aquaforest.
<https://aquaforest.eu/en/knowledge-base/testing-your-aquarium-water-guide-to-essential-water-parameter/> ✓
- [2] Flores Mollo, S., & Aracena Pizarro, D. (2018, Agosto 06). Sistema de monitoreo remoto de acuicultura en estanques para la crianza de camarones. Ingeniare. Revista chilena de ingeniería.
https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052018000500055&lng=es&nrm=iso&tlng=es
- [3] Guía Definitiva Acuario para Principiantes 2023 - AQ, Aquarium Solutions. (n.d.). AQ-arium.
<https://www.aq-arium.com/aqmarine/guia-definitiva-acuario-para-principiantes-2023/>
- [4] Kit Módulo Sensor De PH + Sonda De Electrodo BNC Arduino - Envío Gratis A Todo Chile. (2020, November 9). MechatronicStore.
<https://www.mechatronicstore.cl/kit-modulo-sensor-de-ph-sonda-de-electrodo-bnc-arduino/>
- [5] Raspberry Pi Documentation. (n.d.). Raspberry Pi.
<https://www.raspberrypi.com/documentation/>
- [6] Seeed Studio. (2023, Marzo 13). Grove Ecosystem Introduction. Seeed Studio Wiki.
https://wiki.seeedstudio.com/Grove_System/ ✓
- [7] Sensor De Temperatura Ds18b20 Tipo Termocupla - Envío Gratis A Todo Chile. (n.d.). MechatronicStore.
<https://www.mechatronicstore.cl/sensor-de-temperatura-ds18b20-tipo-termocupla-arduino-pi/>
- [8] Sueldo de Documentador en Chile. (n.d.). Indeed.
<https://cl.indeed.com/career/archivista-y-capturista/salaries>
- [9] Sueldo de Jefe de proyecto en Chile. (n.d.). Indeed Chile.
<https://cl.indeed.com/career/jefe-de-proyecto/salaries> ✓
- [10] Sueldo de Programador en Chile. (n.d.). Indeed Chile.
<https://cl.indeed.com/career/programador-web/salaries>
- [11] Sueldo de Técnico Electrónico en Chile. (n.d.). Indeed.
<https://cl.indeed.com/career/t%C3%A9cnico-electronico/salaries> ✓