

UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE INGENIERÍA  
MAESTRÍA EN SISTEMAS EMBEBIDOS



MEMORIA DEL TRABAJO FINAL

**Sistema de control para estación  
autónoma de monitoreo de ruido  
ambiente submarino**

**Autor:**  
**Patricio Bos**

Director:  
Dr. Ing. Ariel Lutenberg

Jurados:  
Dr. Ing Pablo Gómez (FIUBA)  
Ing. Juan Manuel Cruz (FIUBA, UTN-FRBA)  
Mg. Lic. Igor Prario (FIUBA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires, entre enero  
de 2017 y diciembre de 2018.*



## *Resumen*

La presente memoria describe el diseño e implementación de un sistema embebido para el control de una estación de monitoreo de ruido ambiente submarino. El sistema integra y almacena información de distintos sensores y provee una interfaz para la configuración de la estación. A su vez, permite operar con distintos perfiles de consumo de energía. El conocimiento del nivel de ruido submarino es importante para distintas disciplinas dentro de la oceanografía acústica, estudios de impacto ambiental y diversas aplicaciones que utilicen sistemas SONAR dentro del ámbito científico, civil y militar, entre otras.

Este documento incluye la información de trazabilidad de requerimientos. Se detalla la arquitectura multicore utilizada, junto con los mecanismos de comunicación y sincronización implementados. Asimismo, se describen las técnicas de ingeniería de software aplicadas y el modelo de ramas adoptado para el desarrollo con control de versiones. Finalmente, se incluye la documentación de testing y los resultados obtenidos.



## *Agradecimientos*

Agradecimientos personales. **[OPCIONAL]**

No olvidarse de agradecer al tutor.

No vale poner anti-agradecimientos (este trabajo fue posible a pesar de...)



# Índice general

<b>Resumen</b>	<b>III</b>
<b>1. Introducción General</b>	<b>1</b>
1.1. Descripción técnica-conceptual del proyecto . . . . .	1
1.2. Motivación . . . . .	3
1.3. Objetivos y alcance . . . . .	3
<b>2. Introducción Específica</b>	<b>5</b>
2.1. Requerimientos . . . . .	5
2.2. Planificación . . . . .	6
2.3. Metodologías . . . . .	8
2.3.1. Control de versiones . . . . .	8
2.3.2. Programación concurrente con Protothreads . . . . .	9
<b>3. Diseño e Implementación</b>	<b>11</b>
3.1. Análisis del software . . . . .	11
3.2. Módulo de adquisición . . . . .	12
3.2.1. Sensor de temperatura . . . . .	12
<b>4. Ensayos y Resultados</b>	<b>13</b>
4.1. Pruebas funcionales del hardware . . . . .	13
<b>5. Conclusiones</b>	<b>15</b>
5.1. Conclusiones generales . . . . .	15
5.2. Próximos pasos . . . . .	15
<b>Bibliografía</b>	<b>17</b>





# Índice de figuras

1.1. Diagrama en bloques del sistema. . . . .	2
1.2. Diagrama en bloques del sistema. Se diferencian por color los distintos sub-módulos funcionales y se indica mediante línea de puntos los componentes incluidos en el alcance. . . . .	4
2.1. Diagrama <i>Activity on Node</i> . . . . .	7
2.2. Esquema del flujo de trabajo entre repositorios . . . . .	8



# Índice de cuadros

2.1. Diagrama <i>Etapas del proyecto</i> . . . . .	6
--	---



# Capítulo 1

## Introducción General

### 1.1. Descripción técnica-conceptual del proyecto

La acústica submarina estudia la propagación del sonido en el agua y la interacción de las ondas mecánicas que constituyen el sonido con el agua, los elementos dispersores presentes y con las interfaces aire-agua y agua-lecho marino. Debido a que sufre menor atenuación que otras formas de radiación, el sonido es ampliamente empleado por el hombre en su exploración de los océanos. Las frecuencias típicas utilizadas se encuentran en el rango comprendido entre  $\sim 10$  Hz y 1 MHz, dependiendo de la aplicación. Los sistemas que utilizan la propagación del sonido bajo el agua con diversos fines se conocen como sistemas SONAR (*SOund Navigation And Ranging*).

La propagación del sonido en el agua depende de diversos factores. La dirección de propagación está determinada principalmente por el gradiente vertical de velocidades del sonido, que a su vez depende fundamentalmente de la temperatura y la salinidad del agua. El perfil de velocidades del sonido puede causar zonas de baja intensidad del sonido, llamadas “zonas de sombra”, y regiones de alta intensidad llamadas “cáusticas”. Estas zonas pueden hallarse con el método de trazado de rayos (Tappert, 1977).

El sonido en el agua puede propagarse a grandes distancias, en el orden de miles de kilómetros, debido a la presencia de un canal especial que actúa como guía de onda para el sonido, conocido como SOFAR (*SOund Fixing And Ranging*) que se produce, bajo ciertas condiciones, a la profundidad donde el gradiente de velocidades del sonido alcanza un mínimo (Medwin y Clay, 1997).

Los distintos fenómenos que afectan al sonido submarino pueden ser conveniente y lógicamente agrupados en un pequeño número de parámetros conocidos como parámetros SONAR que se pueden relacionar entre sí mediante las ecuaciones SONAR (Urick, 1975). Estas ecuaciones exhiben las relaciones de trabajo que agrupan los efectos del medio de propagación, el blanco y el equipamiento utilizado y constituyen las herramientas básicas para los profesionales que trabajen en aplicaciones de acústica submarina.

En el campo de la acústica submarina resulta muy relevante el conocimiento del parámetro SONAR Nivel de Ruido en el mar (NL: *Noise Level*), que incluye el Ruido Ambiente propiamente dicho (NL<sub>a</sub>) y el Ruido Propio (NL<sub>p</sub>) asociado al sistema de medición. En el caso de escucha pasiva (estudios de impacto ambiental sobre mamíferos marinos a bajas frecuencias o detección subacuática efectuada desde vehículos submarinos), el NL está dominado por el NL<sub>a</sub>.

Conceptualmente, el Nivel de Ruido en el mar está asociado al ruido de “fondo” (*background*) remanente en ausencia de toda otra fuente identificable. Es el nivel de energía acústica mínimo que debe tener una señal para ser detectada.

Cabe destacar que el ruido subacuático puede clasificarse esencialmente en tres tipos:

- Ambiente: comúnmente denominado ruido de fondo, se mide omnidireccionalmente y es originado principalmente por ruido en la superficie marina (debido al viento, oleaje o lluvia), ruido de origen biológico (producido por peces, mamíferos e invertebrados), ruido sísmico o geoacústico natural, ruido de tráfico marítimo (originado por tráfico marítimo distante).
- Ruido Radiado: originado por una fuente específica tal como un buque en particular, plataformas de explotación de petróleo o gas, instalaciones de exploración y perforación, instalaciones de generación eléctrica, etc.
- Ruido Propio: generado por el propio sistema de electrónico de medición de ruido y por la plataforma donde se encuentre instalado.

El objetivo de este proyecto consiste en diseñar e implementar un sistema embebido para controlar una estación autónoma para la medición in-situ del ruido ambiente submarino para ser instalada en regiones de interés en el mar Argentino y con la capacidad de transmitir datos a una estación receptora en tierra. Este desarrollo permitirá disponer de series temporales de ruido ambiente submarino durante períodos lo suficientemente largos como para analizar los resultados mediante modelos teóricos y/o empíricos que contribuyan a incrementar el conocimiento de dicho parámetro, especialmente a nivel local.

Se presenta un diagrama en bloques del sistema en la figura 1.1.

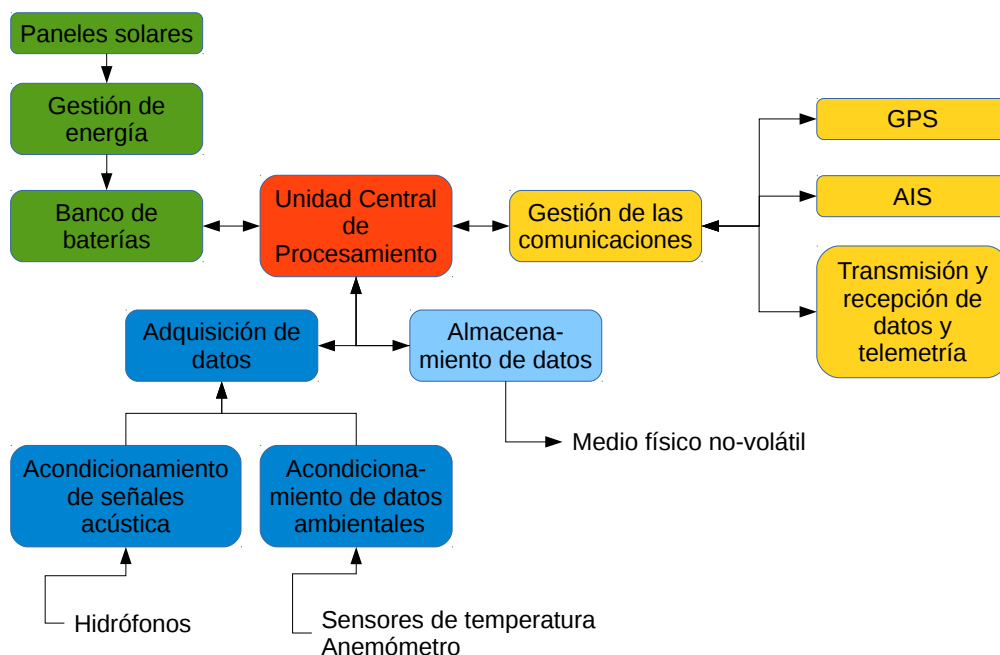


FIGURA 1.1: Diagrama en bloques del sistema. Se diferencian por color los subsistemas funcionales: Energía, Unidad central de procesamiento, comunicaciones, adquisición y almacenamiento.

Para alcanzar el objetivo general se dispone de un paquete tecnológico compuesto principalmente por un equipo de trabajo multidisciplinario con conocimientos teóricos de los fenómenos físicos subyacentes a la propagación del sonido en el medio submarino, acceso a bibliografía especializada en acústica submarina y conocimientos de ingeniería en el campo de los sistemas embebidos.

## 1.2. Motivación

El conocimiento de valores de NL es fundamental en aplicaciones tales como oceanografía acústica, predicción SONAR, exploración geofísica, comunicación subacuática e ingeniería offshore, entre otras.

Por otra parte, existen muy pocas normas a nivel internacional para la estandarización de la medición in-situ del ruido ambiente subacuático. Si bien en acústica aérea sí existen estándares nacionales e internacionales muy aceptados, éstos no pueden extrapolarse fácilmente a la acústica subacuática dadas las diferentes características físicas del fluido en el cual se propaga el sonido, respectivamente.

Actualmente, en el ámbito de la comunidad científica internacional existe una creciente necesidad de medición y monitoreo del ruido subacuático. El interés está parcialmente motivado por un marco regulatorio internacional en lo concerniente al impacto ambiental del ruido subacuático de origen antrópico y principalmente para la evaluación de los efectos sobre la vida marina.

## 1.3. Objetivos y alcance

En particular, para el trabajo final de la Maestría en Sistemas Embebidos, se realizará una primera iteración sobre el ciclo de diseño centrada en la programación del sistema embebido que constituye la unidad central de procesamiento de la boya. Se propone desarrollar sobre la plataforma CIAA-NXP, un firmware multi-core de control que utilice ambos procesadores del microcontrolador LPC4337 y sea capaz de cumplir las siguientes funciones:

- Adquirir datos ambientales de temperatura y velocidad de viento.
- Controlar el sistema mediante una interfaz serie.
- Almacenar los datos en una memoria no volátil.

En la primera iteración, se contempla la posibilidad de simular algún elemento del sistema según sea necesario para avanzar rápidamente en el diseño del firmware de control y las funciones mencionadas.

A los fines prácticos de cumplir los requerimientos de tiempo del trabajo final de maestría, quedarán excluidos del diseño:

- La transmisión de datos en tiempo real a una estación receptora en tierra.
- Consideraciones mecánicas del proyecto.
- La gestión de energía.
- La gestión y control del señalamiento reglamentario marítimo.

- La adquisición de señales acústicas.

Según un estudio preliminar, para el registro de señales acústicas resulta necesario una placa de adquisición A/D con características muy específicas en cuanto a frecuencia de muestreo, bits de resolución y figura de ruido, del tipo NI USB-6356<sup>1</sup> o equivalente. Este tipo de placas poseen *drivers* propietarios cerrados que, en principio, no fue posible utilizar con la CIAA-NXP. Por este motivo, la adquisición de señales acústicas también queda excluida del alcance en esta primera iteración.

Se muestra en la figura 1.2 un diagrama en bloques reducido con los componentes del sistema incluidos en la presente memoria.

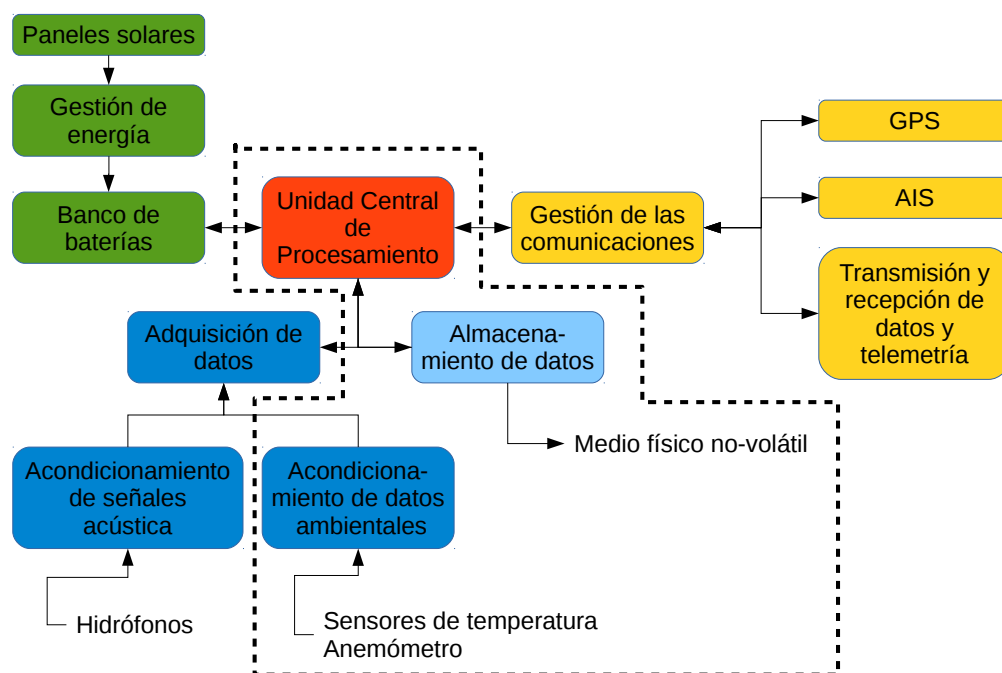


FIGURA 1.2: Diagrama en bloques del sistema. Se diferencian por color los distintos sub-módulos funcionales y se indica mediante línea de puntos los componentes incluidos en el alcance.

<sup>1</sup><http://www.ni.com/pdf/manuals/374452c.pdf>



## Capítulo 2

# Introducción Específica

### 2.1. Requerimientos

A continuación se enumeran los requerimientos del proyecto.

1. Requerimientos de documentación:

- 1.1 Se debe generar un Memoria Técnica con la documentación de ingeniería detallada.
- 1.2 Se debe generar un documento de casos de prueba.

2. Requerimientos funcionales del sistema:

- 2.1 El sistema debe adquirir datos de un array de sensores de temperatura a intervalos regulares con un período de adquisición seleccionable.
- 2.2 El sistema debe adquirir datos de un anemómetro a intervalos regulares con un período de adquisición seleccionable.
- 2.3 El sistema debe almacenar los datos de temperatura y velocidad de viento adquiridas junto con una marca de tiempo identificatoria en un medio físico no volátil.
- 2.4 El sistema debe poder operar con dos perfiles de consumo de energía, uno maximizando el desempeño y otro minimizando el consumo de energía.
- 2.5 El sistema debe contar con una interfaz serie tipo HMI cableada que permita interactuar y realizar operaciones de configuración y mantenimiento.

3. Requerimientos de verificación:

- 3.1 Se debe generar una matriz de trazabilidad entre la Memoria Técnica y los requerimientos.
- 3.2 Se debe generar una matriz de trazabilidad entre las pruebas de integración y los requerimientos.

4. Requerimientos de validación:

- 4.1 Se debe generar una matriz de trazabilidad entre el documento de casos de prueba y los requerimientos.

## 2.2. Planificación

La planificación completa del proyecto puede encontrarse publicada en la web del Laboratorio de Sistemas Embebidos de FIUBA [[insertar referencia](#)].

A los fines de facilitar la comprensión del trabajo realizado, se detallan en la tabla 2.1 las etapas del proyecto junto con la cantidad de horas destinadas y los hitos a alcanzar en cada una de ellas. Puede observarse que el proyecto insume 600 horas de trabajo en total.

Etapas	Horas	Hitos
Documentación y análisis preliminar	100	Plan de trabajo
		Presentación de plan de trabajo
Diseño e implementación	340	Documentación de submódulos
Verificación y validación	60	Reporte de pruebas unitarias
		Reporte de pruebas de integración
		Reporte de resultados de casos de prueba
Proceso de cierre	100	Memoria Técnica
		Presentación de Trabajo Final

TABLA 2.1: Etapas principales del proyecto con el detalle de las horas planificadas y los hitos a alcanzar en cada una de ellas.

Para las etapas planificadas en la tabla 2.1, se realizó un desglose de tareas que puede verse esquemáticamente en el diagrama de *Activity on Node* que se ilustra en la figura 2.1. Se utiliza el mismo código de colores para identificar las diferentes etapas del proyecto y las tareas planificadas que las componen. En el diagrama, los tiempos de duración de las tareas está expresado en horas. Asimismo, las tareas poseen un código único que será usado para realizar la trazabilidad de los requerimientos.

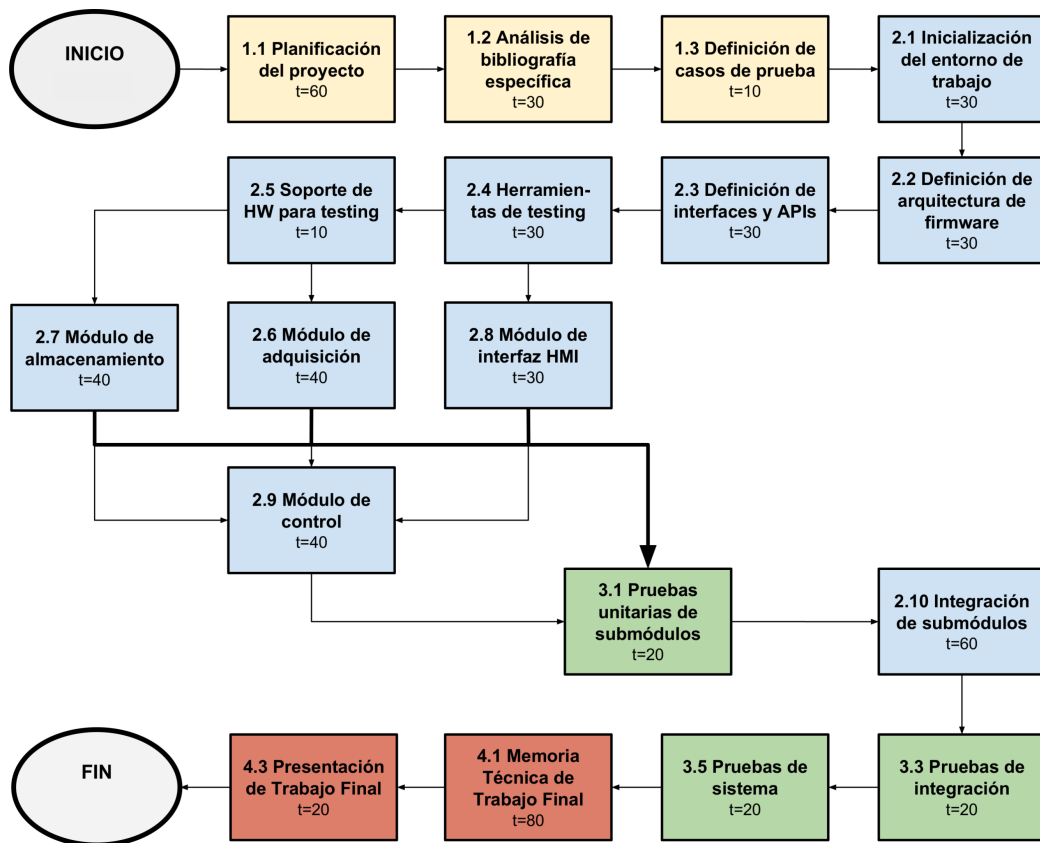


FIGURA 2.1: Diagrama *Activity on Node*. En colores pueden distinguirse las diferentes etapas del proyecto: en amarillo la etapa de documentación y análisis preliminar; en azul la etapa de diseño e implementación; en verde la etapa de verificación y validación y en rojo el proceso de cierre. El tiempo  $t$  está expresado en horas.

## 2.3. Metodologías

En esta sección se describen los aspectos metodológicos relevantes que se aplicaron durante el desarrollo del trabajo.

### 2.3.1. Control de versiones

Se adoptó un modelo de desarrollo creado por Vincent Driessen llamado “A successful Git branching model”<sup>1</sup>. El modelo está basado en la herramienta de control de versiones *git* y consiste en un conjunto de procedimientos para ordenar y sistematizar el flujo de trabajo. Este modelo propone utilizar un repositorio considerado a los fines prácticos “central” (en git todos los repositorios son idénticos) llamado *origin*. Todos los desarrolladores trabajan contra este repositorio central con las operaciones típicas de *push* y *pop*.

Adicionalmente, puede haber intercambios entre los repositorios de los distintos desarrolladores que formen un mismo equipo de trabajo. Estos intercambios pueden visualizarse en la figura 2.2, donde se esquematizan por un lado los posibles flujos de trabajo entre el repositorio *origin* y los distintos desarrolladores y por el otro, entre los repositorios propios de cada desarrollador.

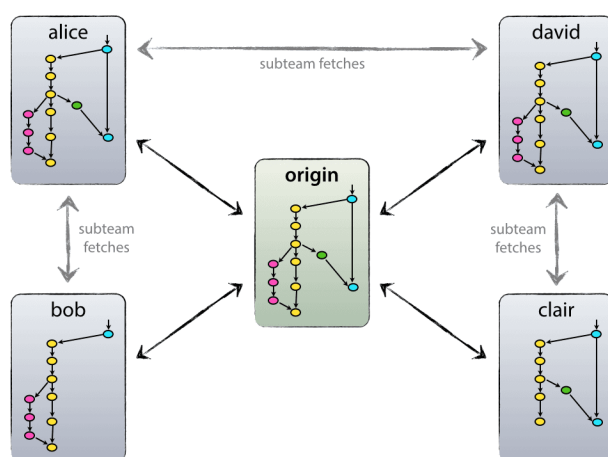


FIGURA 2.2: Esquema del flujo de trabajo entre repositorios<sup>2</sup>.

Para la elaboración de este trabajo, donde la codificación recayó principalmente sobre una sola persona, no fueron habituales las operaciones contra un repositorio distinto de *origin* implementado en *github*. Sin embargo, se considera la experiencia de apropiación de la metodología de trabajo muy valiosa para la formación profesional ya que el autor de este trabajo no había tenido oportunidad de trabajar tan extensa y sistemáticamente con control de versiones previamente.

En cuanto a la estrategia de uso de ramas, siguiendo el modelo adoptado, se dispuso de dos ramas principales llamadas *master* y *develop*. En *origin/master* sólo se

<sup>1</sup><https://nvie.com/posts/a-successful-git-branching-model/>

<sup>2</sup>Imagen tomada de <https://nvie.com/img/centr-decentr@2x.png>.

incluyen *commits* con versiones estables con capacidad de ser puestas en producción, es decir sobre el prototipo de manera que éste pueda operar satisfactoriamente. En *origin/develop* contiene los últimos cambios que integran las diferentes características ya logradas del código.

### 2.3.2. Programación concurrente con Protothreads

Los Protothreads son una abstracción de programación creada por Adam Dunkel<sup>3</sup> para implementar mecanismos de programación concurrente conocidos como multi-tarea cooperativa en sistemas embebidos con recursos limitados. Funcionan como hilos de ejecución sin *stack* o co-rutinas y proveen mecanismos para bloquear la ejecución de una tarea sin que se produzca un cambio de contexto. Esto permite un control de flujo secuencial sin máquinas de estado complejas o soporte multi-hilo completo en arquitecturas basadas en eventos (Dunkels y col., 2006) (Dunkels, Schmidt y Voigt, 2005).

En el presente trabajo, se hace uso de protothreads en la codificación del protocolo de comunicación 1-wire que se describe en la subsección 3.2.1.

---

<sup>3</sup><http://dunkels.com/adam/pt/>



## Capítulo 3

# Diseño e Implementación

### 3.1. Análisis del software

La idea de esta sección es resaltar los problemas encontrados, los criterios utilizados y la justificación de las decisiones que se hayan tomado.

Se puede agregar código o pseudocódigo dentro de un entorno `lstlisting` con el siguiente código:

```
\begin{lstlisting}[caption= "un epígrafe descriptivo"]
```

las líneas de código irían aquí...

```
\end{lstlisting}
```

A modo de ejemplo:

```
1 #define MAX_SENSOR_NUMBER 3
2 #define MAX_ALARM_NUMBER 6
3 #define MAX_ACTUATOR_NUMBER 6
4
5 uint32_t sensorValue[MAX_SENSOR_NUMBER];
6 FunctionalState alarmControl[MAX_ALARM_NUMBER]; //ENABLE or DISABLE
7 state_t alarmState[MAX_ALARM_NUMBER]; //ON or OFF
8 state_t actuatorState[MAX_ACTUATOR_NUMBER]; //ON or OFF
9
10 void vControl() {
11     initGlobalVariables();
12
13     period = 500 ms;
14
15     while(1) {
16         ticks = xTaskGetTickCount();
17
18         updateSensors();
19
20         updateAlarms();
21
22         controlActuators();
23
24         vTaskDelayUntil(&ticks, period);
25     }
26 }
27
28 }
```

ALGORITMO 3.1: Pseudocódigo del lazo principal de control.

## **3.2. Módulo de adquisición**

### **3.2.1. Sensor de temperatura**



## Capítulo 4

# Ensayos y Resultados

### 4.1. Pruebas funcionales del hardware

La idea de esta sección es explicar cómo se hicieron los ensayos, qué resultados se obtuvieron y analizarlos.



## Capítulo 5

# Conclusiones

### 5.1. Conclusiones generales

La idea de esta sección es resaltar cuáles son los principales aportes del trabajo realizado y cómo se podría continuar. Debe ser especialmente breve y concisa. Es buena idea usar un listado para enumerar los logros obtenidos.

### 5.2. Próximos pasos

Acá se indica cómo se podría continuar el trabajo más adelante.



# Bibliografía

- Dunkels, Adam, Oliver Schmidt y Thiemo Voigt (2005). «Using Protothreads for Sensor Node Programming». En: *Proceedings of the REALWSN'05 Workshop on Real-World Wireless Sensor Networks*. Stockholm, Sweden. URL: <http://dunkels.com/adam/dunkels05using.pdf>.
- Dunkels, Adam y col. (2006). «Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems». En: *Proceedings of the Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*. Boulder, Colorado, USA. URL: <http://dunkels.com/adam/dunkels06protothreads.pdf>.
- Medwin, H. y C.S. Clay (1997). *Fundamentals of Acoustical Oceanography*. Applications of Modern Acoustics. Elsevier Science. ISBN: 9780080532165. URL: <https://books.google.com.ar/books?id=kymUKicld2cC>.
- Tappert, Fred D (1977). «The parabolic approximation method». En: *Wave propagation and underwater acoustics*. Springer, págs. 224-287.
- Urick, R.J. (1975). *Principles of Underwater Sound*. McGraw-Hill. ISBN: 9780070660861. URL: <https://books.google.com.ar/books?id=zAJRAAAAMAAJ>.