

Solución IoT para robot de exploración ambiental de datos críticos con almacenamiento en blockchain

Esp. Ing. Gonzalo Carreño

Carrera de Maestría en Internet de las Cosas

Director: Esp. Ing. Sergio Alberino (UTN-FRBA)

Jurados:

Jurado 1 (pertencia)
Jurado 2 (pertencia)
Jurado 3 (pertencia)

Ciudad de [lugar], [mes] de [año]

Resumen

El presente trabajo describe la implementación de un proyecto personal en el que se desarrolla una solución de IoT (Internet de las Cosas) para un caso de uso de robot de exploración ambiental en el que, debido a la criticidad de los datos, sea necesario garantizar la inmutabilidad y transparencia de su almacenamiento.

El sistema cuenta con una arquitectura robusta y flexible en la que se integra el dispositivo robótico de exploración ambiental, desarrollado en el marco de la Carrera de Especialización de Sistemas Embebidos [1], con un sistema *backend* desplegado en la nube pública, y una red blockchain a fin de poder asegurar la inmutabilidad y transparencia de las lecturas ambientales.

Para su implementación se utilizaron conceptos y herramientas tales como el desarrollo de sistemas embebidos en ESP32 y FreeRTOS, mensajería MQTT, almacenamiento y procesamiento BigData, la implementación de soluciones *cloud*, entre otros.

Agradecimientos

Esta sección es para agradecimientos personales y es totalmente **OPCIONAL**.

Índice general

Resumen	I
1. Introducción general	1
1.1. Estado del arte	1
1.2. Motivación del trabajo	1
1.3. Alcance y objetivos	1
1.4. Requerimientos del producto	2
2. Introducción específica	5
2.1. Estilo y convenciones	5
2.1.1. Uso de mayúscula inicial para los título de secciones	5
2.1.2. Este es el título de una subsección	5
2.1.3. Figuras	6
2.1.4. Tablas	7
2.1.5. Ecuaciones	8
3. Diseño e implementación	11
3.1. Análisis del software	11
4. Ensayos y resultados	13
4.1. Pruebas funcionales del hardware	13
5. Conclusiones	15
5.1. Conclusiones generales	15
5.2. Próximos pasos	15
Bibliografía	17

Índice de figuras

2.1. Ilustración del cuadrado azul que se eligió para el diseño del logo.	6
2.2. Imagen tomada de la página oficial del procesador ¹	7
2.3. ¿Por qué de pronto aparece esta figura?	7
2.4. Tres gráficos simples.	7

Índice de tablas

2.1. caption corto	8
--	---

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción general

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

1.1. Estado del arte

L^AT_EX no es WYSIWYG (What You See is What You Get), a diferencia de los procesadores de texto como Microsoft Word o Pages de Apple o incluso LibreOffice en el mundo open-source. En lugar de ello, un documento escrito para L^AT_EX es en realidad un archivo de texto simple o llano que *no contiene formato*. Nosotros le decimos a L^AT_EX cómo deseamos que se aplique el formato en el documento final escribiendo comandos simples entre el texto, por ejemplo, si quiero usar texto en itálicas para dar énfasis, escribo `\it{texto}` y pongo el texto que quiero en itálicas entre medio de las llaves. Esto significa que L^AT_EX es un lenguaje del tipo «mark-up», muy parecido a HTML.

1.2. Motivación del trabajo

L^AT_EX no es WYSIWYG (What You See is What You Get), a diferencia de los procesadores de texto como Microsoft Word o Pages de Apple o incluso LibreOffice en el mundo open-source. En lugar de ello, un documento escrito para L^AT_EX es en realidad un archivo de texto simple o llano que *no contiene formato*. Nosotros le decimos a L^AT_EX cómo deseamos que se aplique el formato en el documento final escribiendo comandos simples entre el texto, por ejemplo, si quiero usar texto en itálicas para dar énfasis, escribo `\it{texto}` y pongo el texto que quiero en itálicas entre medio de las llaves. Esto significa que L^AT_EX es un lenguaje del tipo «mark-up», muy parecido a HTML.

1.3. Alcance y objetivos

A continuación se detallan las funcionalidades incluidas en el alcance del trabajo.

- La publicación del endpoint MQTT para la recepción de los datos enviados por el robot.
- La adaptación del sistema embebido del robot de exploración ambiental para la conexión segura con el backend vía MQTT.

- La arquitectura e implementación de los sistemas backend y el modelo de datos necesario para el almacenamiento de las mediciones enviadas por el robot.
- La arquitectura, implementación y despliegue de la dApp y *smart contracts* necesarios para el almacenamiento de las mediciones en una red Blockchain (a definir).
- La definición de métricas agregadas de valor y posterior arquitectura e implementación de los sistemas analíticos para procesar de forma *batch* y/o *real-time* (dependiendo de las métricas a definir) utilizando herramientas de procesamiento paralelo basadas en big data.
- La implementación de la interfaz gráfica para poder visualizar los datos enviados y analíticas calculadas.

1.4. Requerimientos del producto

A continuación, se listan los requerimientos del producto:

1. Requerimientos funcionales

- a) El robot de exploración ambiental debe poder enviar a la plataforma datos de mediciones de parámetros ambientales, incluyendo los datos de fecha, hora, localización geográfica (que puede ser implementada como un mock inicialmente) y la categorización si es o no un valor crítico.
- b) El robot de exploración ambiental debe incorporar lógica para categorizar los valores medidos de cada parámetro ambiental como valores críticos si:
 - 1) Representan un máximo o mínimo global sensado hasta el momento.
 - 2) Representan un máximo o mínimo local durante el último día.
- c) La plataforma debe poder recibir y almacenar las mediciones de parámetros ambientales enviadas por el robot.
- d) Los datos considerados críticos deben ser almacenados en un sistema inmutable.
- e) La plataforma debe poder procesar las mediciones de parámetros ambientales enviadas por el robot para generar métricas de valor para el usuario de negocio.
- f) La plataforma debe brindar dos *front-end* con interfaz web:
 - 1) El *front-end* para el usuario de negocio.
 - 2) El *front-end* para el usuario administrador.
- g) El *front-end* para el usuario de negocio debe proveer métricas para visualizar:
 - 1) Las lecturas históricas almacenadas.

- 2) Agregaciones (máximo, mínimo, promedio, etc) de cada parámetro ambiental agrupado por frecuencias (ventanas de tiempo) y coordenadas geográficas.
 - 3) Las referencias a los datos persistidos en blockchain.
 - h) El *front-end* para el usuario de administración debe permitir:
 - 1) Acceder a los diferentes recursos utilizados por la herramienta (topics MQTT, *smart contracts*, *buckets*, etc).
 - 2) Resetear valores y estado.
2. Requerimientos no funcionales
- a) La plataforma debe contar con al menos un *back-end* de procesamiento y acceso a datos operacionales para la lógica de negocio.
 - b) La plataforma debe contar con al menos un *back-end* de acceso, procesamiento, almacenamiento de datos analíticos para la generación de métricas.
 - c) El envío de los valores ambientales censados al *back-end* debe ser mediante MQTT.
 - d) Las lecturas ambientales categorizadas como críticas deben ser almacenadas en blockchain para garantizar fiabilidad e inmutabilidad.
 - e) La gestión de datos almacenados en blockchain debe ser implementada mediante *smart contracts* desplegados en la red.
 - f) La interacción con los *smart contracts* debe realizarse desde una dApp.
 - g) Los sistemas de transferencia y almacenamiento de datos utilizados deben contar con seguridad, permitiendo encriptación, autenticación y autorización.
3. Requerimientos de documentación
- a) Video demostrativo.
 - b) Documentación de arquitectura técnica del diseño del sistema.
 - c) Manual de usuario.
 - d) Memoria final.
4. Requerimiento de testing
- a) Se debe incluir tests de unitarios de componentes.
 - b) Se debe incluir tests funcionales (*smoke test*) del producto general.
5. Requerimientos opcionales
- a) De infraestructura y despliegue:
 - 1) Se permite realizar el despliegue de la dApp en un IPFS (preferentemente) o en la nube.

- 2) Se permite agregado hardware al robot para la captura de datos adicionales.
- 3) Se permite agregar automatización para la creación de la infraestructura como código.

b) De datos:

- 1) Se permite almacenar cualquier otro dato adicional sensado o derivado.
- 2) Se permite agregar cualquier implementación de gobierno de datos.
- 3) Se permite almacenar cualquier otra métrica o gráfico de explotación de datos adicional.

Capítulo 2

Introducción específica

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

2.1. Estilo y convenciones

2.1.1. Uso de mayúscula inicial para los título de secciones

Si en el texto se hace alusión a diferentes partes del trabajo referirse a ellas como capítulo, sección o subsección según corresponda. Por ejemplo: “En el capítulo 1 se explica tal cosa”, o “En la sección 2.1 se presenta lo que sea”, o “En la subsección 2.1.2 se discute otra cosa”.

Cuando se quiere poner una lista tabulada, se hace así:

- Este es el primer elemento de la lista.
- Este es el segundo elemento de la lista.

Notar el uso de las mayúsculas y el punto al final de cada elemento.

Si se desea poner una lista numerada el formato es este:

1. Este es el primer elemento de la lista.
2. Este es el segundo elemento de la lista.

Notar el uso de las mayúsculas y el punto al final de cada elemento.

2.1.2. Este es el título de una subsección

Se recomienda no utilizar **texto en negritas** en ningún párrafo, ni tampoco texto subrayado. En cambio sí se debe utilizar *texto en itálicas* para palabras en un idioma extranjero, al menos la primera vez que aparecen en el texto. En el caso de palabras que estamos inventando se deben utilizar “comillas”, así como también para citas textuales. Por ejemplo, un *digital filter* es una especie de “selector” que permite separar ciertos componentes armónicos en particular.

La escritura debe ser impersonal. Por ejemplo, no utilizar “el diseño del firmware lo hice de acuerdo con tal principio”, sino “el firmware fue diseñado utilizando tal principio”.

El trabajo es algo que al momento de escribir la memoria se supone que ya está concluido, entonces todo lo que se refiera a hacer el trabajo se narra en tiempo pasado, porque es algo que ya ocurrió. Por ejemplo, "se diseñó el firmware empleando la técnica de test driven development".

En cambio, la memoria es algo que está vivo cada vez que el lector la lee. Por eso transcurre siempre en tiempo presente, como por ejemplo:

"En el presente capítulo se da una visión global sobre las distintas pruebas realizadas y los resultados obtenidos. Se explica el modo en que fueron llevados a cabo los test unitarios y las pruebas del sistema".

Se recomienda no utilizar una sección de glosario sino colocar la descripción de las abreviaturas como parte del mismo cuerpo del texto. Por ejemplo, RTOS (*Real Time Operating System*, Sistema Operativo de Tiempo Real) o en caso de considerarlo apropiado mediante notas a pie de página.

Si se desea indicar alguna página web utilizar el siguiente formato de referencias bibliográficas, dónde las referencias se detallan en la sección de bibliografía de la memoria, utilizado el formato establecido por IEEE en [2]. Por ejemplo, "el presente trabajo se basa en la plataforma EDU-CIAA-NXP [3], la cual...".

2.1.3. Figuras

Al insertar figuras en la memoria se deben considerar determinadas pautas. Para empezar, usar siempre tipografía claramente legible. Luego, tener claro que **es incorrecto** escribir por ejemplo esto: "El diseño elegido es un cuadrado, como se ve en la siguiente figura:"



La forma correcta de utilizar una figura es con referencias cruzadas, por ejemplo: "Se eligió utilizar un cuadrado azul para el logo, como puede observarse en la figura 2.1".



FIGURA 2.1. Ilustración del cuadrado azul que se eligió para el diseño del logo.

El texto de las figuras debe estar siempre en español, excepto que se decida reproducir una figura original tomada de alguna referencia. En ese caso la referencia de la cual se tomó la figura debe ser indicada en el epígrafe de la figura e incluida como una nota al pie, como se ilustra en la figura 2.2.

FIGURA 2.2. Imagen tomada de la página oficial del procesador¹.

La figura y el epígrafe deben conformar una unidad cuyo significado principal pueda ser comprendido por el lector sin necesidad de leer el cuerpo central de la memoria. Para eso es necesario que el epígrafe sea todo lo detallado que corresponda y si en la figura se utilizan abreviaturas entonces aclarar su significado en el epígrafe o en la misma figura.



FIGURA 2.3. ¿Por qué de pronto aparece esta figura?

Nunca colocar una figura en el documento antes de hacer la primera referencia a ella, como se ilustra con la figura 2.3, porque sino el lector no comprenderá por qué de pronto aparece la figura en el documento, lo que distraerá su atención.

Otra posibilidad es utilizar el entorno *subfigure* para incluir más de una figura, como se puede ver en la figura 2.4. Notar que se pueden referenciar también las figuras internas individualmente de esta manera: 2.4a, 2.4b y 2.4c.



(A) Un caption.



(B) Otro.



(C) Y otro más.

FIGURA 2.4. Tres gráficos simples.

El código para generar las imágenes se encuentra disponible para su reutilización en el archivo **Chapter2.tex**.

2.1.4. Tablas

Para las tablas utilizar el mismo formato que para las figuras, sólo que el epígrafe se debe colocar arriba de la tabla, como se ilustra en la tabla 2.1. Observar que sólo algunas filas van con líneas visibles y notar el uso de las negritas para los encabezados. La referencia se logra utilizando el comando `\ref{<label>}` donde label debe estar definida dentro del entorno de la tabla.

¹Imagen tomada de <https://goo.gl/images/i7C70w>

```

\begin{table}[h]
\centering
\caption[caption corto]{caption largo más descriptivo}
\begin{tabular}{l c c}
\toprule
\textbf{Especie} & \textbf{Tamaño} & \textbf{Valor}\\
\midrule
Amphiprion Ocellaris & 10 cm & \$ 6.000 \\
Hepatus Blue Tang & 15 cm & \$ 7.000 \\
Zebrasoma Xanthurus & 12 cm & \$ 6.800 \\
\bottomrule
\hline
\end{tabular}
\label{tab:peces}
\end{table}

```

TABLA 2.1. caption largo más descriptivo.

Especie	Tamaño	Valor
Amphiprion Ocellaris	10 cm	\$ 6.000
Hepatus Blue Tang	15 cm	\$ 7.000
Zebrasoma Xanthurus	12 cm	\$ 6.800

En cada capítulo se debe reiniciar el número de conteo de las figuras y las tablas, por ejemplo, figura 2.1 o tabla 2.1, pero no se debe reiniciar el conteo en cada sección. Por suerte la plantilla se encarga de esto por nosotros.

2.1.5. Ecuaciones

Al insertar ecuaciones en la memoria dentro de un entorno *equation*, éstas se numeran en forma automática y se pueden referir al igual que como se hace con las figuras y tablas, por ejemplo ver la ecuación 2.1.

$$ds^2 = c^2 dt^2 \left(\frac{d\sigma^2}{1 - k\sigma^2} + \sigma^2 [d\theta^2 + \sin^2 \theta d\phi^2] \right) \quad (2.1)$$

Es importante tener presente que si bien las ecuaciones pueden ser referidas por su número, también es correcto utilizar los dos puntos, como por ejemplo “la expresión matemática que describe este comportamiento es la siguiente:”

$$\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r})\Psi = -i\hbar \frac{\partial \Psi}{\partial t} \quad (2.2)$$

Para generar la ecuación 2.1 se utilizó el siguiente código:

```

\begin{equation}
\label{eq:metric}
ds^2 = c^2 dt^2 \left( \frac{d\sigma^2}{1-k\sigma^2} + \right.
\sigma^2 \left[ d\theta^2 + \sin^2 \theta d\phi^2 \right] \left. \right)
\end{equation}

```

Y para la ecuación [2.2](#):

```
\begin{equation}
\label{eq:schrodinger}
\frac{\hbar^2}{2m}\nabla^2\Psi + V(\mathbf{r})\Psi =
-i\hbar \frac{\partial\Psi}{\partial t}
\end{equation}
```


Capítulo 3

Diseño e implementación

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

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
12     initGlobalVariables();
13
14     period = 500 ms;
15
16     while(1) {
17
18         ticks = xTaskGetTickCount();
19
20         updateSensors();
21
22         updateAlarms();
23
24         controlActuators();
25
26         vTaskDelayUntil(&ticks, period);
27     }
```

28 }

CÓDIGO 3.1. Pseudocódigo del lazo principal de control.

Capítulo 4

Ensayos y resultados

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

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

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

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.

En esta sección no se deben incluir ni tablas ni gráficos.

Algunas preguntas que pueden servir para completar este capítulo:

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se pudo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

5.2. Próximos pasos

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

Bibliografía

- [1] Esp. Ing. Gonzalo Carreño. *UBA/CESE - Robot de exploración ambiental*. URL: https://github.com/kronleuchter85/cese_proyecto_especializacion/blob/main/docs/TTFB_Memoria_Gonzalo_Carreno.pdf.
- [2] IEEE. *IEEE Citation Reference*. 1.^a ed. IEEE Publications, 2016. URL: <http://www.ieee.org/documents/ieeecitationref.pdf> (visitado 26-09-2016).
- [3] Proyecto CIAA. *Computadora Industrial Abierta Argentina*. Visitado el 2016-06-25. 2014. URL: <http://proyecto-ciaa.com.ar/devwiki/doku.php?id=start>.