

Índice general

Resumen	I
1. Introducción general	1
1.1. Descripción del sistema	1
1.2. Motivación	2
1.3. Estado del arte	2
1.4. Objetivos y alcance	3
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

Índice general

Resumen	I
1. Introducción general	1
1.1. Descripción del sistema	1
1.2. Motivación	2
1.3. Estado del arte	2
1.4. Objetivos y alcance	3
2. Introducción específica	5
2.1. Protocolos de comunicación	5
2.1.1. Protocolo HTTP	5
2.1.2. Protocolo MQTT	5
2.1.3. Eclipse Mosquitto	6
3. Diseño e implementación	7
3.1. Análisis del software	7
4. Ensayos y resultados	9
4.1. Pruebas funcionales del hardware	9
5. Conclusiones	11
5.1. Conclusiones generales	11
5.2. Próximos pasos	11
Bibliografía	13

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 figuras

2.1.	Arquitectura MQTT publish/subscribe.	6
------	--------------------------------------	---

Índice de tablas

1.1. caption corto 3

2.1. caption corto 8

Índice de tablas

1.1. caption corto 3

Capítulo 1

Introducción general

El presente capítulo aborda cuestiones relativas a las etapas en los procesos de rectificación de motopartes en la empresa Arancibia Rectificaciones y las problemáticas de administración que motivaron la implementación del sistema.

1.1. Descripción del sistema

En el taller de rectificaciones de motopartes se realizan diferentes tipos de trabajos relacionados a la tornería de piezas pertenecientes a los motores de motocicletas. Estos trabajos pasan por distintas etapas o estados en los cuales se realizan procesos específicos como encamisado de cilindro, cambio de biela, centrado de cigüeñal, rectificación de cilindro, rectificación de tapa de cilindro, entre otros.

Las etapas en general que atraviesa un repuesto desde que ingresa hasta que es retirado de la empresa son:

1. Ingreso de la pieza o repuesto a la empresa.
2. Registro de datos del cliente y generación de orden de trabajo.
3. Puesta en espera del repuesto.
4. Trabajo de mano de obra correspondiente.
5. Finalización del trabajo de mano de obra.
6. Entrega del repuesto al cliente.

Un cliente de la empresa se presenta con una pieza para ser reparada, el personal de atención le informa precio del servicio, fecha de entrega, entre otros datos (punto 1). Luego de aceptadas las condiciones por el cliente, se registran sus datos y se genera una orden de trabajo (punto 2), posteriormente se ubica la pieza en el sector de trabajos en espera (punto 3). Una vez que un empleado de taller de la empresa está libre recoge el repuesto para efectuar la mano de obra necesaria (punto 4), en esta etapa se realizan distintos tipos de tareas hasta que se termina todo el proceso y se coloca el repuesto en el sector de finalizados a la espera de ser retirado por el cliente (punto 5). Cuando el cliente pasa a retirar su pieza, se registran los datos correspondientes y se hace la entrega finalizando así todas las etapas del servicio.

Capítulo 1

Introducción general

El presente capítulo aborda cuestiones relativas a las etapas en los procesos de rectificación de motopartes en la empresa Arancibia Rectificaciones y las problemáticas de administración que motivaron la implementación del sistema.

1.1. Descripción del sistema

En el taller de rectificaciones de motopartes se realizan diferentes tipos de trabajos relacionados a la tornería [1] de piezas pertenecientes a los motores de motocicletas. Estos trabajos pasan por distintas etapas o estados en los cuales se realizan procesos específicos como encamisado de cilindro [2], cambio de biela [3], balanceo de cigüeñal [4], rectificación de cilindro [5], rectificación de tapa de cilindro [6], entre otros.

Las etapas en general que atraviesa un repuesto desde que ingresa hasta que es retirado de la empresa son:

1. Ingreso de la pieza o repuesto a la empresa.
2. Registro de datos del cliente y generación de orden de trabajo.
3. Puesta en espera del repuesto.
4. Trabajo de mano de obra correspondiente.
5. Finalización del trabajo de mano de obra.
6. Entrega del repuesto al cliente.

Un cliente de la empresa se presenta con una pieza para ser reparada, el personal de atención le informa el precio del servicio, fecha de entrega, entre otros datos (punto 1). Luego de aceptadas las condiciones por el cliente, se registran sus datos y se genera una orden de trabajo (punto 2), posteriormente se ubica la pieza en el sector de trabajos en espera (punto 3). Una vez que un empleado de taller de la empresa está libre toma el repuesto para efectuar la mano de obra necesaria (punto 4), en esta etapa se realizan distintos tipos de tareas hasta que se termina todo el proceso y se coloca el repuesto en el sector de finalizados a la espera de ser retirado por el cliente (punto 5). Cuando el cliente pasa a retirar su pieza, se registran los datos correspondientes y se hace la entrega finalizando así todas las etapas del servicio.

1.2. Motivación

Este trabajo surgió de la necesidad de desarrollar un sistema que permita **visualizar** en qué etapa se encuentra un repuesto en particular en la empresa, esto **permite** conocer el estado general de trabajos, informar a los clientes, tomar **decisiones** administrativas o técnicas, realizar reportes, etc.

No es necesario conocer el trabajo específico que se está efectuando en cada **etapa** sino que basta con saber **el estado actual y en ciertos casos los estados anteriores**.

Cuando un cliente se comunica con la empresa para saber si puede pasar a retirar la pieza, el personal de atención tiene que consultar a los empleados de taller el estado en el que se encuentra el trabajo, estos deben dejar de hacer sus tareas por un momento para buscar y responder la consulta, lo cual interrumpe el proceso, genera demoras y consume **tiempo**, **además** mientras esto **sucede** el cliente debe **quedar esperando generalmente** varios minutos.

Por otro lado, resulta complicado cuando personal de la empresa desea **obtener** información **como** cantidad de trabajos en cada sector, tiempos promedio **de proceso**, trabajos para ser retirados, cantidad de servicios efectuados en un lapso de tiempo determinado, etc., ya que la manera de obtener estos datos es **realizando** conteos manuales lo cual resulta **improductivo** y demanda demasiado tiempo por lo que nunca se realizan estos informes.

Ante este escenario es evidente la necesidad de contar con un sistema informático que posibilite registrar las etapas **necesarias** y generar la información necesaria para cuando esta sea requerida.

1.3. Estado del arte

En el mercado argentino actualmente se ofrecen diferentes soluciones para resolver problemas relacionados al control de productos ya sea de stock, logística, trazabilidad, transporte, distribución, entre otros. Estas soluciones **están** basadas en su mayoría en tecnología de lectura de código de barras o de ingresos manuales de datos mediante teclado. Además existen algunas soluciones de empresas **extranjeras**, mas orientadas al sector industrial, basadas en tecnología **RFID**, en su mayoría por banda **UHF**.

No se encontraron soluciones en el mercado para las necesidades específicas que se plantean en este trabajo. Una de las **problemáticas** que plantea el escenario para el cual se desarrolla este **sistema** es el contexto en el que se realizan los servicios. Las piezas que se reparan están sometidas constantemente a aceites, residuos **grasos**, polvo, etc. Este escenario hace que se descarte el uso de tecnología por código de **barras** ya que cualquier lectura a un código sería dificultada **por lo mencionado**, quedando como mejor opción la utilización de tecnología RFID.

Las soluciones RFID encontradas están planteadas para otro tipo de rubros o industrias, usan generalmente banda UHF **las cuales son efectivas pero demasiado** costosas para una **empresa** chica o mediana. **Por otra parte la oferta de servicios de banda HF son menos costosas pero están implementadas en materiales adhesivos que no se podrían utilizar en el escenario actual por el material de las** motopartes (metal y aluminio).

1.2. Motivación

Este trabajo surgió de la necesidad de desarrollar un sistema que permita **visualizar** en qué etapa se encuentra un repuesto en particular en la empresa, esto **permite** conocer el estado general de **los** trabajos, informar a los clientes, tomar **decisiones** administrativas o técnicas, realizar reportes, etc.

No es necesario conocer el trabajo específico que se está efectuando en cada **momento**, sino que basta con saber **la etapa actual del proceso**.

Cuando un cliente se comunica con la empresa para saber si puede pasar a retirar la pieza, el personal de atención tiene que consultar a los empleados de taller el estado en el que se encuentra el trabajo, estos deben dejar de hacer sus tareas por un momento para buscar y responder la consulta, lo cual interrumpe el proceso, genera demoras y consume **tiempo**. **Además**, mientras esto **sucede**, el cliente debe **esperar** varios minutos.

Por otro lado, resulta complicado cuando **el** personal de la empresa desea **obtener** información **como**: cantidad de trabajos en cada sector, tiempos promedio **de proceso**, trabajos para ser retirados, cantidad de servicios efectuados en un lapso de tiempo determinado, etc., ya que la manera de obtener estos datos es **realizando** conteos manuales lo cual resulta **improductivo** y demanda demasiado tiempo por lo que nunca se realizan estos informes.

Ante este escenario es evidente la necesidad de contar con un sistema informático que posibilite registrar las etapas **del proceso** y generar la información necesaria para cuando esta sea requerida.

1.3. Estado del arte

En el mercado argentino actualmente se ofrecen diferentes soluciones para resolver problemas relacionados al control de productos ya sea de stock, logística, trazabilidad, transporte, distribución, entre otros. Estas soluciones **están** basadas en su mayoría en tecnología de lectura de código de barras o de ingresos manuales de datos mediante teclado. Además existen algunas soluciones de empresas **extranjeras**, más orientadas al sector industrial, basadas en tecnología **RFID** [7], en su mayoría por banda **UHF** [8].

No se encontraron soluciones en el mercado para las necesidades específicas que se plantean en este trabajo. Una de las **problemáticas** que plantea el escenario para el cual se desarrolla este **sistema**, es el contexto en el que se realizan los servicios. Las piezas que se reparan están sometidas constantemente a aceites, residuos grasos, polvo, etc. Este escenario hace que se descarte el uso de **la tecnología de lectura de código de barras**, ya que cualquier lectura a un código sería dificultada **por lo mencionado**, quedando como mejor opción la utilización de tecnología RFID.

Las soluciones RFID encontradas están planteadas para otro tipo de rubros o industrias, usan generalmente banda UHF **y son demasiado** costosas para una **em-**presa chica o mediana.

Únicamente se encontró una empresa en Argentina que ofrece servicios algo similares a los que se plantean en este trabajo, Telectrónica S.A. [9]. A continuación se detallan algunas características.

Únicamente se encontró una empresa en Argentina que ofrece servicios algo similares a los que se plantean en este trabajo. A continuación se detallan algunas características.

TABLA 1.1. servicios ofrecidos por Telectrónica

Característica	Telectrónica
Tecnología RFID	si
Rubro motores	no
Costo accesible a empresa pequeña	no
Hardware económico	no

La principal característica que imposibilita acceder a este tipo de servicios con empresas argentinas o **extranjer**as es el alto costo de desarrollo e implementación, debido a que están enfocadas en industrias o empresas grandes que pueden afrontar inversiones de gran escala.

1.4. Objetivos y alcance

El objetivo de este trabajo fue desarrollar un sistema que permita registrar los estados por los que va pasando un repuesto en el taller de tornería de la **empres**a, y poder visualizar esos estados en una plataforma web o móvil.

En primer **lugar** se realizó el abordaje de requerimientos de la empresa y se comenzó con la planificación del proyecto. Se continuó con el diseño de la arquitectura tecnológica que se emplearía para el sistema, tanto a nivel de herramientas de desarrollo de software como el hardware a utilizar.

Además, se tuvo en cuenta que los trabajadores de la empresa no debían detener sus tareas para realizar ingresos en teclados ya que esto generaría una interrupción en el flujo de trabajo y el registro de datos en el sistema sería incomodo. Fue por esta razón, principalmente, que se pensó en una tecnología que permita enviar datos a un servidor sin necesidad de manipulación de teclados o dispositivos similares. La tecnología que cumple con este requerimiento es la RFID, la que abordaremos en el siguiente capítulo.

Una vez determinado el diseño y la planificación se comenzaron las investigaciones necesarias, las cuales requirieron una parte importante del tiempo total del trabajo.

El alcance del trabajo se acotó a lo siguiente:

- Desarrollo frontend: aplicación web compatible con móvil.
- Desarrollo backend: API Rest.
- Desarrollo de base de datos.
- Desarrollo e implementación en dispositivos de hardware IoT.
- Desarrollo e implementación de la infraestructura total del sistema, servidor basado en contenedores para servicio web, API Rest, **broker** mqtt y base de datos.

TABLA 1.1. servicios ofrecidos por Telectrónica

Característica	Telectrónica
Tecnología RFID	si
Rubro motores	no
Costo accesible a empresa pequeña	no
Hardware económico	no

La principal característica que imposibilita acceder a este tipo de servicios con empresas argentinas o **extranjer**as es el alto costo de desarrollo e implementación, debido a que están enfocadas en industrias o empresas grandes que pueden afrontar inversiones de gran escala.

1.4. Objetivos y alcance

El objetivo de este trabajo fue desarrollar un sistema que permita registrar los estados por los que va pasando un repuesto en el taller de tornería de la **empres**a y poder visualizar esos estados en una plataforma web o móvil.

En primer **lugar**, se realizó el abordaje de requerimientos de la empresa y se comenzó con la planificación del proyecto. Se continuó con el diseño de la arquitectura tecnológica que se emplearía para el sistema, tanto a nivel de herramientas de desarrollo de software como el hardware a utilizar.

Además, se tuvo en cuenta que los trabajadores de la empresa no debían detener sus tareas para realizar ingresos en teclados ya que esto generaría una interrupción en el flujo de trabajo y el registro de datos en el sistema sería incomodo. Fue por esta razón, principalmente, que se pensó en una tecnología que permita enviar datos a un servidor sin necesidad de manipulación de teclados o dispositivos similares. La tecnología que cumple con este requerimiento es la RFID, la que abordaremos en el siguiente capítulo.

Una vez determinado el diseño y la planificación se comenzaron las investigaciones necesarias, las cuales requirieron una parte importante del tiempo total del trabajo.

El alcance del trabajo se acotó a lo siguiente:

- Desarrollo frontend: aplicación web compatible con móvil.
- Desarrollo backend: API Rest.
- Desarrollo de base de datos.
- Desarrollo e implementación en dispositivos de hardware IoT.
- Desarrollo e implementación de la infraestructura total del sistema, servidor basado en contenedores para servicio web, API Rest, **broker** mqtt y base de datos.
- Implementaciones particulares como gabinetes, soportes para tags RFID, entre otros.

- Implementaciones particulares como gabinetes, soportes para tags RFID, entre otros.

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”.

Capítulo 2

Introducción específica

En este capítulo se describen las herramientas, tecnologías y hardware que se utilizó para el desarrollo del sistema.

2.1. Protocolos de comunicación

2.1.1. Protocolo HTTP

HTTP [10], por sus siglas en inglés: *Hypertext Transfer Protocol*, es un protocolo de tipo cliente-servidor [11], mediante el cual se establece una comunicación enviando peticiones y obteniendo respuestas.

Las características principales de este protocolo son:

- Basado en arquitectura cliente-servidor.
- Además de hipertexto (HTML [12]) se puede utilizar para transmitir otro tipo de documentos como imágenes o videos.
- Es un protocolo de capa de aplicación del modelo OSI [13].
- Se transmite principalmente sobre el protocolo TCP[14].

HTTP define un conjunto de métodos de petición, cada uno indica una acción a ejecutar en el servidor. Los más utilizados son:

- GET: se utiliza para recuperar datos.
- POST: sirve principalmente para cargar nuevos datos.
- PATCH: este método aplica modificaciones parciales a los datos existentes.
- PUT: permite reemplazar completamente un registro.
- DELETE: elimina datos específicos.

2.1.2. Protocolo MQTT

MQTT [15] son las siglas de *Message Queuing Telemetry Transport*. Se trata de un protocolo de mensajería liviano para usar en casos donde existen recursos limitados de ancho de banda.

Se transmite sobre protocolo TCP en la arquitectura publish/subscribe [16].

Los roles que intervienen en un protocolo MQTT son los siguientes:

- Publicadores: son los que envían los datos.

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, donde las referencias se detallan en la sección de bibliografía de la memoria, utilizando el formato establecido por IEEE en [IEEE:citation]. Por ejemplo, "el presente trabajo se basa en la plataforma EDU-CIAA-NXP [CIAA], 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

- Suscriptores: son los que consumen los datos.
- Broker: Transmite los mensajes publicados a los suscriptores.

Un cliente puede ser publicador, suscriptor o ambos. El broker es el punto central de la comunicación ya que sin este los mensajes nunca llegarían a destino.

En la figura 2.1 se puede apreciar un ejemplo de comunicación en la arquitectura MQTT.

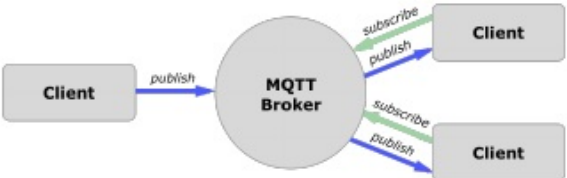


FIGURA 2.1. Arquitectura MQTT publish/subscribe.

La estructura de mensajes en este protocolo se dividen en dos: *topics* los cuales son de tipo jerárquicos, utilizando la barra (/) como separador, y *payload* en donde se incluye el mensaje que se quiere transmitir. Por ejemplo: topic: "nodo-s/procesos/guardar", payload:"mensaje de ejemplo". Siguiendo este ejemplo un cliente podría suscribirse a ese topic o a una jerarquía más alta y recibir todos los mensajes de los topics que comiencen con nodo/procesos.

2.1.3. Eclipse Mosquitto

Eclipse Mosquitto [17] es un broker MQTT OpenSource liviano y adecuado para utilizar en todo tipo de dispositivos sobre todo aquellos que cuenten con baja potencia como microcontroladores.



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

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.

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

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

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
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.

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.

```
\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.

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

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:”

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

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 + \right.
\sin^2\theta d\phi^2 \left. \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 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 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
17         ticks = xTaskGetTickCount();
18
19         updateSensors();
20
21         updateAlarms();
22
23         controlActuators();
24
25         vTaskDelayUntil(&ticks, period);
26     }
27 }
28 }
```

CÓDIGO 3.1. Pseudocódigo del lazo principal de control.

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.

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.

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.

Bibliografía

- [1] Lautaro Regis. «Tornaría». En: *Esc. de Ed. Tec. Nuestra Señora de la Guarda* (2018).
- [2] MecanicaFácil. *Rectificado del Bloque del Motor*. http://www.mecanicafacil.info/Rectificado_del_Bloque_del_Motor.html. Sep. de 2012. (Visitado 06-09-2022).
- [3] Gilda Liliana Ballivian Rosado. «Reparación del Motor». En: *Instituto de Educación Superior Tecnológico Público* (2016).
- [4] Robert Bosch. «Manual de la técnica automóvil». En: Editorial Reverte S.A., 1999, págs. 387-389.
- [5] William H. Crouse. *Mecánica de la Motocicleta*. Marcombo, 1992, pág. 321.
- [6] Ediciones Necochea. *Tapas de Cilindros*. Ediciones Necochea, 2021.
- [7] Dipole. ¿Qué es RFID? <https://www.dipolerfid.es/blog-rfid/que-es-rfid>. (Visitado 06-09-2022).
- [8] Enciclopedia EcuRed. *Sistemas radioeléctricos*. <https://www.ecured.cu/UHF>. (Visitado 06-09-2022).
- [9] Electrónica. *Web oficial*. <https://telectronica.com/soluciones-rfid/>. (Visitado 06-09-2022).
- [10] Mozilla. *Generalidades del protocolo HTTP*. <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>. (Visitado 06-09-2022).
- [11] Oscar Blancarte. *Arquitectura cliente-servidor*. <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor>. (Visitado 06-09-2022).
- [12] Mozilla. *HTML: Lenguaje de etiquetas de hipertexto*. <https://developer.mozilla.org/es/docs/Web/HTML>. (Visitado 06-09-2022).
- [13] Cloudflare. ¿Qué es el modelo OSI? <https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/>. (Visitado 06-09-2022).
- [14] Rfc. *Specification of internet transmission control program*. <https://www.rfc-editor.org/rfc/rfc675>. (Visitado 06-09-2022).
- [15] Paessler. ¿Qué es MQTT? <https://www.paessler.com/es/it-explained/mqtt>. (Visitado 06-09-2022).
- [16] Amazon. *Pub/Sub Messaging*. <https://aws.amazon.com/es/pub-sub-messaging/>. (Visitado 06-09-2022).
- [17] Eclipse. *Eclipse Mosquitto™ An open source MQTT broker*. <https://mosquitto.org/>. (Visitado 06-09-2022).