

v

Índice general

Resumen	I
1. Introducción general	1
1.1. Planteo básico	1
1.1.1. Botones antipánico	1
1.2. Motivación	2
1.3. Objetivos y alcance	3
1.4. Soluciones comerciales similares	4
1.4.1. LK109	4
1.4.2. TK102B	5
2. Introducción específica	7
2.1. Características de las soluciones existentes	7
2.1.1. Características del hardware	7
2.1.2. Características de los sistemas web	8
2.2. Componentes de hardware utilizados	9
2.2.1. Microcontrolador ESP32	9
2.2.2. Módulo GPS NEO-6M	10
2.2.3. Módulo GSM SIM800L	11
2.2.4. Batería 18650 y chip cargador TP4056	11
2.3. Componentes de software utilizados	12
2.3.1. Django REST framework	12
2.3.2. Angular	12
2.3.3. PostgreSQL	13
2.3.4. Heroku	13
2.4. Herramientas y servicios de software utilizados	14
2.4.1. Docker	14
2.4.2. Visual Studio Code	14
2.4.3. PyGPSClient	14
2.4.4. Moserial	15
2.4.5. Twilio	15
3. Diseño e implementación	17
3.1. Análisis del software	17
4. Ensayos y resultados	19
4.1. Pruebas funcionales del hardware	19
5. Conclusiones	21
5.1. Conclusiones generales	21
5.2. Próximos pasos	21
Bibliografía	23

v

Índice general

Resumen	I
1. Introducción general	1
1.1. Planteo básico	1
1.1.1. Botones antipánico	1
1.2. Motivación	2
1.3. Objetivos y alcance	3
1.4. Soluciones comerciales similares	4
1.4.1. LK109	4
1.4.2. TK102B	4
1.5. Personalizando la plantilla, el archivo <code>memoria.tex</code>	6
1.6. El código del archivo <code>memoria.tex</code> explicado	6
1.7. Bibliografía	7
2. Introducción específica	9
2.1. Estilo y convenciones	9
2.1.1. Uso de mayúscula inicial para los título de secciones	9
2.1.2. Este es el título de una subsección	9
2.1.3. Figuras	10
2.1.4. Tablas	11
2.1.5. Ecuaciones	12
3. Diseño e implementación	15
3.1. Análisis del software	15
4. Ensayos y resultados	17
4.1. Pruebas funcionales del hardware	17
5. Conclusiones	19
5.1. Conclusiones generales	19
5.2. Próximos pasos	19
Bibliografía	21

Índice de figuras

1.1. Diagrama en bloques del dispositivo y el sistema web	1
1.2. Imagen del botón LK109	5
1.3. Imagen del botón TK102B	6
2.1. Microcontrolador ESP32-S	9
2.2. Módulo GPS NEO-6M con una antena externa	10
2.3. Módulo SIM800L	11
2.4. Batería 18650	11
2.5. Circuito Mp4056	11
2.6. Aplicación PyGPSClient	14
2.7. Aplicación Moserial	15

Índice de figuras

1.1. Diagrama en bloques del dispositivo y el sistema web	1
1.2. Imagen del botón LK109	5
1.3. Imagen del botón TK102B	6
2.1. Ilustración del cuadrado azul que se eligió para el diseño del logo.	10
2.2. Imagen tomada de la página oficial del procesador ¹	11
2.3. ¿Por qué de pronto aparece esta figura?	11
2.4. Tres gráficos simples	11

Índice de tablas

1.1. Resumen de objetivos	4
1.2. Tabla comparativa	6
2.1. Tabla comparativa	9
2.2. Tabla comparativa	10
2.3. Tabla comparativa	12
2.4. Tabla comparativa	13

Índice de tablas

1.1. Resumen de objetivos	4
1.2. Tabla comparativa	5
2.1. caption corto	12

Estos dispositivos generalmente son brindados por las autoridades, o también pueden ser brindados por empresas de seguridad privadas. Sus beneficiarios suelen ser:

- Personas en situación de violencia de género.
- Personas en situación de violencia intrafamiliar.
- Establecimientos donde pueden ocurrir situaciones de violencia y es necesario dar rápidamente aviso a las autoridades.
- Agente o sereno cuyo recorrido debe ser registrado.
- Entre otros.

Entre sus funcionalidades, suelen poseer:

- Conectividad a la red de telefonía móvil (*Global System for Mobile communications* o GSM).
- Acceso a datos móviles (*General Packet Radio Service* o GPRS).
- **Conectividad** a sistemas de geoposicionamiento satelital (*Global Navigation Satellite System* o GNSS).
- Botón de disparo de alerta.
- Reporte periódico de ubicación por mensaje de texto (SMS) o vía web.

Un detalle interesante respecto a estos dispositivos es que generalmente se comercializan en dos formatos:

- Dispositivo personal: dispositivo embebido que puede ser llevado como colgante, en la muñeca o en el bolsillo. Tiene una autonomía limitada.
- Dispositivo para móviles: también conocido como AVL o *Automatic Vehicle Location*; además de contar con antenas externas para mayor precisión, cuenta con cables para conexión a la batería de los vehículos, permitiendo tener una mayor autonomía.

En relación a este trabajo, se planteó el desarrollo **del** prototipo de un botón geolocalizador y una plataforma web a modo de prueba de concepto, para la recepción de alertas.

1.2. Motivación

El autor del trabajo ha trabajado anteriormente con este tipo de soluciones, principalmente en la adquisición, configuración, implementación y soporte de botones antipánico disponibles en el mercado. Se detectó y se llegó a la conclusión, respecto a los dispositivos, de que estos presentan los siguientes inconvenientes:

- Especificaciones de hardware obsoletas.
- Documentación escasa, desactualizada o incluso incorrecta en algunos casos.
- Poca fiabilidad de dispositivo (dificultad para configurarlos, mala señal en ambientes *indoor*, desconfianza al momento de utilizarlo).
- Inconsistencias entre dispositivos idénticos.

Estos dispositivos generalmente son brindados por las autoridades, o también pueden ser brindados por empresas de seguridad privadas. Sus beneficiarios suelen ser:

- Personas en situación de violencia de género.
- Personas en situación de violencia intrafamiliar.
- Establecimientos donde pueden ocurrir situaciones de violencia y es necesario dar rápidamente aviso a las autoridades.
- Agente o sereno cuyo recorrido debe ser registrado.
- Entre otros.

Entre sus funcionalidades, suelen poseer:

- Conectividad a la red de telefonía móvil (*Global System for Mobile communications* o GSM).
- Acceso a datos móviles (*General Packet Radio Service* o GPRS).
- **conectividad** a sistemas de geoposicionamiento satelital (*Global Navigation Satellite System* o GNSS).
- Botón de disparo de alerta.
- Reporte periódico de ubicación por mensaje de texto (SMS) o vía web.

Un detalle interesante respecto a estos dispositivos es que generalmente se comercializan en dos formatos:

- Dispositivo personal: dispositivo embebido que puede ser llevado como colgante, en la muñeca o en el bolsillo. Tiene una autonomía limitada.
- Dispositivo para móviles: también conocido como AVL o *Automatic Vehicle Location*; además de contar con antenas externas para mayor precisión, cuenta con cables para conexión a la batería de los vehículos, permitiendo tener una mayor autonomía.

En relación a este trabajo, se planteó el desarrollo **un** prototipo de un botón geolocalizador y una plataforma web a modo de prueba de concepto, para la recepción de alertas.

1.2. Motivación

El autor del trabajo ha trabajado anteriormente con este tipo de soluciones, principalmente en la adquisición, configuración, implementación y soporte de botones antipánico disponibles en el mercado. Se detectó y se llegó a la conclusión, respecto a los dispositivos, de que estos presentan los siguientes inconvenientes:

- Especificaciones de hardware obsoletas.
- Documentación escasa, desactualizada o incluso incorrecta en algunos casos.
- Poca fiabilidad de dispositivo (dificultad para configurarlos, mala señal en ambientes *indoor*, desconfianza al momento de utilizarlo).
- Inconsistencias entre dispositivos idénticos.

- Poca duración de batería (desde aproximadamente 24 horas hasta 3 o 4 días como máximo, cuando lo deseable para el beneficiario final del botón es de al menos 5 o 6 días).
- Precio exageradamente elevado y muchas dificultades para adquirirlos mediante proveedores, más en períodos de alta inflación.
- Nulas características de seguridad.

Se investigaron y revisaron diferentes alternativas en el mercado, así como también se trabajó con dispositivos utilizados por sistemas de la **competencia**, llegando a conclusiones similares: no se encuentran dispositivos aceptables cuyo costo sea accesible.

Todas estas cuestiones nombradas anteriormente son las que motivaron la **formulación** del proyecto, siendo necesario disponer de mínimamente un **dispositivo** embebido con **un conjunto de** funcionalidades **definidas**, y un sistema web que permita recepcionar la información del botón.

Otras motivaciones menores del trabajo, pero igualmente muy relacionadas a la especialización, fueron:

- Profundizar en el uso de frameworks para *Single Web Applications* o SPAs.
- Poder aplicar los conocimientos trabajados en relación a tecnologías *cloud* y desarrollo de APIs web.

1.3. Objetivos y alcance

En relación a los objetivos de alto nivel del trabajo, estos fueron:

- Investigación y determinación de un conjunto de módulos de hardware que sean adecuados para la construcción del prototipo.
- Diseño y **construcción** un prototipo de botón antipánico que permita validar las decisiones tecnológicas.
- Tener fiabilidad superior sobre ciertas características, en comparación con algunos de los dispositivos disponibles en el **mercado**.
- Desarrollo de un sistema web a modo de prueba de concepto para el uso del dispositivo.

Respecto a los objetivos por componente, estos fueron:

No se incluyeron dentro del alcance del trabajo, las siguientes características:

- Aplicación web productiva para gestionar las activaciones del botón antipánico. Por productiva se refiere a que pueda ser ofrecida al mercado o utilizable por usuarios encargados del monitoreo.
- Desarrollo de un dispositivo listo para reemplazar dispositivos existentes, es decir que no sea un prototipo.
- Desarrollo de un contenedor físico para el dispositivo.

Además, durante el desarrollo del trabajo, se fueron detectando diferentes cuestiones técnicas a resolver, que modificaron algunos objetivos planteados inicialmente.

- Poca duración de batería (desde aproximadamente 24 horas hasta 3 o 4 días como máximo, cuando lo deseable para el beneficiario final del botón es de al menos 5 o 6 días).
- Precio exageradamente elevado y muchas dificultades para adquirirlos mediante proveedores, más en períodos de alta inflación.
- Nulas características de seguridad.

Se investigaron y revisaron diferentes alternativas en el mercado, así como también se trabajó con dispositivos utilizados por sistemas de la **competencia** llegando a conclusiones similares: no se encuentran dispositivos aceptables cuyo costo sea accesible.

Todas estas cuestiones nombradas anteriormente son las que motivaron la **formulación** del proyecto, siendo necesario disponer de mínimamente un **dispositivo** embebido con funcionalidades **muy básicas**, y un sistema web **básico** que permita recepcionar la información del botón.

Otras motivaciones menores del trabajo, pero igualmente muy relacionadas a la especialización, fueron:

- Profundizar en el uso de frameworks para *Single Web Applications* o SPAs.
- Poder aplicar los conocimientos trabajados en relación a tecnologías *cloud* y desarrollo de APIs web.

1.3. Objetivos y alcance

En relación a los objetivos de alto nivel del trabajo, estos fueron:

- Investigación y determinación de un conjunto de módulos de hardware que sean adecuados para la construcción del prototipo.
- Diseño y **contrucción** un prototipo **básico** de botón antipánico que permita validar las decisiones tecnológicas.
- Tener fiabilidad superior sobre ciertas características, en comparación con algunos de los dispositivos disponibles en el **mercado**.
- Desarrollo de un sistema web a modo de prueba de concepto para el uso del dispositivo.

Respecto a los objetivos por componente, estos fueron:

No se incluyeron dentro del alcance del trabajo, las siguientes características:

- Aplicación web productiva para gestionar las activaciones del botón antipánico. Por productiva se refiere a que pueda ser ofrecida al mercado o utilizable por usuarios encargados del monitoreo.
- Desarrollo de un dispositivo listo para reemplazar dispositivos existentes, es decir que no sea un prototipo.
- Desarrollo de un contenedor físico para el dispositivo.

Además, durante el desarrollo del trabajo, se fueron detectando diferentes cuestiones técnicas a resolver, que modificaron algunos objetivos planteados inicialmente.

TABLA 1.1. Resumen de objetivos del trabajo

Componente	Objetivo
Prototipo	Duración de la batería
Prototipo	Calidad de señal de GSM en ambientes cerrados y abiertos
Prototipo	Calidad de señal de GNSS en ambientes cerrados y abiertos
Prototipo	Tiempo de actualización de la ubicación (<i>time-to-first-fix</i>)
Prototipo	Envío de alertas mediante pulsador
Sistema web	Recepción de SMS de alertas
Sistema web	Alta de dispositivos
Sistema web	Visualización de alertas en tiempo real
Sistema web	Despliegue en un entorno <i>cloud</i>

1.4. Soluciones comerciales similares

Retomando los inconvenientes de soluciones actuales, se presentan a continuación algunos modelos conocidos y utilizados en el mercado local. Estos dos modelos sufren los inconvenientes nombrados en las secciones anteriores: el motivo de fondo es que existen muchos clones de estos modelos, resultando muy difícil determinar si se está trabajando con un dispositivo original o un clon con posibles problemas [CLONES:1].

1.4.1. LK109

LK109 es el nombre de un dispositivo geolocalizador que trabaja con redes GSM y GPRS, y usa *Global Positioning System* (GPS) como sistema de GNSS. Existe también una versión que trabaja con redes 3G, cuyo costo es superior. Entre sus características más interesantes se encuentran [LK109MANUAL:1]:

- Indicadores de estado: batería, señal de GPS y señal de GSM.
- Posibilidad de reportar la ubicación mediante TCP o SMS.
- Reporte de ubicación en tiempo real, con una periodicidad configurable.
- Posibilidad de monitorear de forma remota el entorno del dispositivo mediante un parlante.
- Autonomía de 240 horas o 10 días.

TABLA 1.1. Resumen de objetivos del trabajo

Componente	Objetivo
Prototipo	Duración de la batería
Prototipo	Calidad de señal de GSM en ambientes cerrados y abiertos
Prototipo	Calidad de señal de GNSS en ambientes cerrados y abiertos
Prototipo	Tiempo de actualización de la ubicación (<i>time-to-first-fix</i>)
Prototipo	Envío de alertas mediante pulsador
Sistema web	Recepción de SMS de alertas
Sistema web	Alta de dispositivos
Sistema web	Visualización de alertas en tiempo real
Sistema web	Despliegue en un entorno <i>cloud</i>

1.4. Soluciones comerciales similares

Retomando los inconvenientes de soluciones actuales, se presentan a continuación algunos modelos conocidos y utilizados en el mercado local. Estos dos modelos sufren los inconvenientes nombrados en las secciones anteriores: el motivo de fondo es que existen muchos clones de estos modelos, resultando muy difícil determinar si se está trabajando con un dispositivo original o un clon con posibles problemas [2].

1.4.1. LK109

LK109 es el nombre de un dispositivo geolocalizador que trabaja con redes GSM y GPRS, y usa *Global Positioning System* (GPS) como sistema de GNSS. Existe también una versión que trabaja con redes 3G, cuyo costo es superior. Entre sus características más interesantes se encuentran [3]:

- Indicadores de estado: batería, señal de GPS y señal de GSM.
- Posibilidad de reportar la ubicación mediante TCP o SMS.
- Reporte de ubicación en tiempo real, con una periodicidad configurable.
- Posibilidad de monitorear de forma remota el entorno del dispositivo mediante un parlante.
- Autonomía de 240 horas o 10 días.

1.4.2. TK102B

TK102B es el nombre de otro dispositivo geolocalizador que trabaja con redes GSM y GPRS, y usa *Global Positioning System* (GPS) como sistema de GNSS. Entre sus características más relevantes se encuentran [4]:

- Indicadores de estado: batería, señal de GPS y señal de GSM.
- Posibilidad de reportar la ubicación mediante TCP o SMS, incluyendo la dirección aproximada, mediante geodecodificación inversa.
- Reporte de ubicación en tiempo real, con una periodicidad configurable.
- Posibilidad de monitorear de forma remota el entorno del dispositivo mediante un parlante.



FIGURA 1.2. Imagen del botón LK109

1.4.2. TK102B

TK102B es el nombre de otro dispositivo geolocalizador que trabaja con redes GSM y GPRS, y usa *Global Positioning System* (GPS) como sistema de GNSS. Entre sus características más relevantes se encuentran [TK102BMANUAL:1]:

- Indicadores de estado: batería, señal de GPS y señal de GSM.
- Posibilidad de reportar la ubicación mediante TCP o SMS, incluyendo la dirección aproximada, mediante geodecodificación inversa.
- Reporte de ubicación en tiempo real, con una periodicidad configurable.
- Posibilidad de monitorear de forma remota el entorno del dispositivo mediante un parlante.
- Autonomía de 80 horas o 3 días aproximadamente.



FIGURA 1.2. Imagen del botón LK109

- Autonomía de 80 horas o 3 días aproximadamente.

TABLA 1.2. Tabla comparativa de los dos modelos

Característica	LK109	TK102B
Conectividad	GSM, GPRS, SMS	GSM, GPRS, SMS
Reporte periódico	Si	Si
Autonomía (horas)	240	80
Indicadores de estado	Si	Si
Escucha remota	Si	Si
Protocolo de comunicación	h02	gps103

Respecto a al ítem de autonomía, no se ha podido comprobar en la práctica que la duración sea la indicada (siempre es mucho menor, entre 12 y 24 horas), a excepción del uso mediante *sleep* o ahorro de batería, que no tiene utilidad para las aplicaciones de seguimiento de una persona. Esto es debido a que en el modo de ahorro de batería, la ubicación se actualiza posterior al envío de una alerta y no antes, resultando inútil la ubicación reportada.

En relación al ítem de escucha remota, es decir la posibilidad de escuchar lo que ocurre alrededor del dispositivo, no se ha podido verificar para varios dispositivos; de todas maneras no resulta, al menos para los casos de uso reales, una funcionalidad atractiva. Las funcionalidades que se deben cumplir y validar a rajatabla son:

- Precisión al momento de obtener la ubicación.
- Autonomía considerable de la batería.
- Rapidez para obtener señal de GSM y GNSS.



FIGURA 1.3. Imagen del botón TK102B

TABLA 1.2. Tabla comparativa de los dos modelos

Característica	LK109	TK102B
Conectividad	GSM, GPRS, SMS	GSM, GPRS, SMS
Reporte periódico	Si	Si
Autonomía (horas)	240	80
Indicadores de estado	Si	Si
Escucha remota	Si	Si
Protocolo de comunicación	h02	gps103

Respecto al ítem de autonomía, no se ha podido comprobar en la práctica que la duración sea la indicada (siempre es mucho menor, entre 12 y 24 horas), a excepción del uso mediante *sleep* o ahorro de batería, que no tiene utilidad para las aplicaciones de seguimiento de una persona. Esto es debido a que en el modo de ahorro de batería, la ubicación se actualiza posterior al envío de una alerta y no antes, resultando inútil la ubicación reportada.

En relación al ítem de escucha remota, es decir la posibilidad de escuchar lo que ocurre alrededor del dispositivo, no se ha podido verificar para varios dispositivos; de todas maneras no resulta, al menos para los casos de uso reales, una funcionalidad atractiva. Las funcionalidades que se deben cumplir y validar a rajatabla son:

- Precisión al momento de obtener la ubicación.
- Autonomía considerable de la batería.
- Rapidez para obtener señal de GSM y GNSS.

Por este motivo, es que no se intentó entre los objetivos iniciales ir más allá de las características iniciales.



FIGURA 1.3. Imagen del botón TK102B

Por este motivo, es que no se intentó entre los objetivos iniciales ir más allá de las características básicas.

1.5. Personalizando la plantilla, el archivo `memoria.tex`

Para personalizar la plantilla se debe incorporar la información propia en los distintos archivos `.tex`.

Primero abrir `memoria.tex` con TexMaker (o el editor de su preferencia). Se debe ubicar dentro del archivo el bloque de código titulado *INFORMACIÓN DE LA PORTADA* donde se deben incorporar los primeros datos personales con los que se construirá automáticamente la portada.

1.6. El código del archivo `memoria.tex` explicado

El archivo `memoria.tex` contiene la estructura del documento y es el archivo de mayor jerarquía de la memoria. Podría ser equiparable a la función `main()` de un programa en C, o mejor dicho al archivo fuente `.c` donde se encuentra definida la función `main()`.

La estructura básica de cualquier documento de \LaTeX comienza con la definición de clase del documento, es seguida por un preámbulo donde se pueden agregar funcionalidades con el uso de paquetes (equiparables a bibliotecas de C), y finalmente, termina con el cuerpo del documento, donde irá el contenido de la memoria.

```
\documentclass{article} <- Definición de clase
\usepackage{listings} <- Preambulo

\begin{document} <- Comienzo del contenido propio
    Hello world!
\end{document}
```


Capítulo 2

Introducción específica

En este capítulo se hace una presentación de cuestiones técnicas particulares del trabajo, una descripción de alto nivel de las tecnologías involucradas en el desarrollo del trabajo, las herramientas que se utilizaron y servicios externos que formaron parte de la elaboración del trabajo, así también como los componentes de hardware empleados.

2.1. Características de las soluciones existentes

Es necesario, en primer lugar, hacer una introducción de las características que deben tener las soluciones de este tipo, junto con sus limitaciones y particularidades técnicas. Cualquier desarrollo de este tipo, debe considerarlas siguientes características para los módulos de hardware y software.

2.1.1. Características del hardware

Los botones antipánico y dispositivos geolocalizadores, suelen ser utilizados en zonas donde la cobertura de telefonía móvil es muy baja. Es decir, se cuenta con conectividad limitada para llamadas y mensajes, y prácticamente no hay cobertura para datos móviles, o es de baja tasa de transferencia. La tecnología más presente es la de GPRS o 2G[NPERF:1], ya que es la red de mayor alcance, pero también la de menor ancho de banda. Esto impone una restricción respecto a la conectividad, por este motivo la mayoría de los dispositivos tienen las siguientes características respecto al método de comunicación:

- Comunicación vía SMS, TCP o UDP para configuración y envío de datos.
- Comunicación vía protocolos personalizados¹ para reducir el *overhead* de datos.
- Mecanismos para almacenar datos en caso de perder conectividad.

Debido a que dentro del alcance del trabajo se apunta a configuración y envío de mensajes de alerta, se consideró innecesario, para estas funciones, requerir otro medio adicional. No se considera apropiado usar un método de comunicación que pueda no ser fiable, como lo son los datos móviles en zonas donde la conectividad es baja.

¹Existe una cantidad muy diversa de protocolos, desde protocolos binarios hasta protocolos en formato texto; puede consultarse el [listado de protocolos](#) que incorpora la plataforma *open source* Traccar al respecto.

El archivo `memoria.tex` se encuentra densamente comentado para explicar qué páginas, secciones y elementos de formato está creando el código \LaTeX en cada línea. El código está dividido en bloques con nombres en mayúsculas para que resulte evidente qué es lo que hace esa porción de código en particular. Inicialmente puede parecer que hay mucho código \LaTeX , pero es principalmente código para dar formato a la memoria por lo que no requiere intervención del usuario de la plantilla. Si se deben personalizar con su información los bloques indicados como:

- Informacion de la memoria
- Resumen
- Agradecimientos
- Dedicatoria

El índice de contenidos, las listas de figura de tablas se generan en forma automática y no requieren intervención ni edición manual por parte del usuario de la plantilla.

En la parte final del documento se encuentran los capítulos y los apéndices. Por defecto se incluyen los 5 capítulos propuestos que se encuentran en la carpeta `/Chapters`. Cada capítulo se debe escribir en un archivo `.tex` separado y se debe poner en la carpeta `Chapters` con el nombre `Chapter1`, `Chapter2`, etc... El código para incluir capítulos desde archivos externos se muestra a continuación.

```
\include{Chapters/Chapter1}
\include{Chapters/Chapter2}
\include{Chapters/Chapter3}
\include{Chapters/Chapter4}
\include{Chapters/Chapter5}
```

Los apéndices también deben escribirse en archivos `.tex` separados, que se deben ubicar dentro de la carpeta `Appendices`. Los apéndices vienen comentados por defecto con el carácter `%` y para incluirlos simplemente se debe eliminar dicho carácter.

Finalmente, se encuentra el código para incluir la bibliografía en el documento final. Este código tampoco debe modificarse. La metodología para trabajar las referencias bibliográficas se desarrolla en la sección 1.7.

1.7. Bibliografía

Las opciones de formato de la bibliografía se controlan a través del paquete de latex `biblatex` que se incluye en la memoria en el archivo `memoria.tex`. Estas opciones determinan cómo se generan las citas bibliográficas en el cuerpo del documento y cómo se genera la bibliografía al final de la memoria.

En el preámbulo se puede encontrar el código que incluye el paquete `biblatex`, que no requiere ninguna modificación del usuario de la plantilla, y que contiene las siguientes opciones:

```
\usepackage[backend=bibtex,
             natbib=true,
             style=numeric,
```

Por otra parte, otro punto importante respecto a estos dispositivos, es su capacidad de obtener señal aceptable tanto de GSM como GNSS en ambientes *indoor* y *outdoor*. Prácticamente todos los botones antipánico poseen antenas internas y sin alimentación activa, por lo que su capacidad para adquirir un nivel de señal aceptable es limitada; esto introduce varios puntos a favor y en contra:

- No generan un consumo de energía adicional.
- Su tiempo de *startup*, es decir, hasta que adquieren un nivel de señal operativo aceptable, es elevado si están en ambientes cerrados.
- Su precisión suele ser baja en ambientes cerrados, lo cual puede ser un problema en situaciones de emergencia.
- El tiempo de actualización de la posición de la persona puede ser elevado si la antena no logra captar una buena señal.

2.1.2. Características de los sistemas web

Un sistema web, al requerir estar en comunicación con un dispositivo y mostrar su información en tiempo real, debe contar con las siguientes características, idealmente:

- No requerir ningún software adicional para su acceso y uso más allá de un navegador web y conexión a Internet.
- Poder recibir mensajes de texto, es decir, tener un número telefónico virtual asociado.
- Enviar actualizaciones *push* al navegador del usuario en tiempo real.

Se presentan algunos desafíos respecto al segundo y tercer punto. Para la recepción de SMS, es necesario tener integrada una central de telefonía virtual o implementar un *Short Message Service Center* o SMSC, un servicio específico para el ruteo de mensajes de texto desde y hacia la web. Respecto al tercer punto, resulta obligatoria la implementación de algún mecanismo de actualización en tiempo real desde el servidor hacia el cliente, el navegador web, y no al revés, como puede ser *WebSocket*.

En algunos casos, también debería permitir comunicación bidireccional con los dispositivos, para que un operador pueda configurarlos *on-demand*. De todas maneras, debido a que está fuera del alcance del trabajo, no se planteó el desarrollo de esta característica.

```
    sorting=none}
{bibtex}
```

En el archivo **reference.bib** se encuentran las referencias bibliográficas que se pueden citar en el documento. Para incorporar una nueva cita al documento lo primero es agregarla en este archivo con todos los campos necesario. Todas las entradas bibliográficas comienzan con @ y una palabra que define el formato de la entrada. Para cada formato existen campos obligatorios que deben completarse. No importa el orden en que las entradas estén definidas en el archivo .bib. Tampoco es importante el orden en que estén definidos los campos de una entrada bibliográfica. A continuación se muestran algunos ejemplos:

```
@ARTICLE{ARTICLE:1,
  AUTHOR="John Doe",
  TITLE="Title",
  JOURNAL="Journal",
  YEAR="2017",
}

@BOOK{BOOK:1,
  AUTHOR="John Doe",
  TITLE="The Book without Title",
  PUBLISHER="Dummy Publisher",
  YEAR="2100",
}

@INBOOK{BOOK:2,
  AUTHOR="John Doe",
  TITLE="The Book without Title",
  PUBLISHER="Dummy Publisher",
  YEAR="2100",
  PAGES="100-200",
}
```

Se debe notar que los nombres *ARTICLE:1*, *BOOK:1*, *BOOK:2* y *WEBSITE:1* son nombres de fantasía que le sirve al autor del documento para identificar la entrada. En este sentido, se podrían reemplazar por cualquier otro nombre. Tampoco es necesario poner : seguido de un número, en los ejemplos sólo se incluye como un posible estilo para identificar las entradas.

La entradas se citan en el documento con el comando:

```
\cite{nombre_de_la_entrada}
```

Y cuando se usan, se muestran así: [5], [6], [7], [8]. Notar cómo se conforma la sección Bibliografía al final del documento.

Finalmente y como se mencionó en la subsección ??, para actualizar las referencias bibliográficas tanto en la sección bibliografía como las citas en el cuerpo del documento, se deben ejecutar las herramientas de compilación PDFLaTeX, BibTeX, PDFLaTeX, PDFLaTeX, en ese orden. Este procedimiento debería resolver cualquier mensaje Citation xxxxx on page x undefined".

2.2. Componentes de hardware utilizados

2.2.1. Microcontrolador ESP32



FIGURA 2.1. Microcontrolador ESP32-S

Se utilizó un SoC (*System on a Chip*) de la familia ESP32 de Espressif debido a que resulta una opción ideal para el desarrollo de sistemas embebidos y aplicaciones de Internet de las Cosas. Entre las características que resultaron más atractivas para su elección fueron[ESP32:1]:

- Bajo costo y disponibilidad en el mercado local.
- Prestaciones muy interesantes: Wi-Fi, Bluetooth, modos de ahorro de energía, varios periféricos, entre otros.
- Gran soporte de la comunidad y disponibilidad de bibliotecas, junto a su documentación.
- Ya utilizado en la carrera de especialización.

Se presenta a continuación algunas alternativas comparadas²

TABLA 2.1. Tabla comparativa de placas de desarrollo

Característica	ESP32	A9G	LoPy 4	TTGO T-Call
Disponibilidad	Baja	No	No	Baja
Comunidad	Muy grande	Grande	Poca	Si
Costo	Muy bajo	Moderado	Muy elevado	Muy elevado
Curva aprendizaje	Muy baja	Alta	Baja	Baja

²La comparación debe ser interpretada linealmente ya que hay placas como Ai Thinker A9G y TTGO T-Call V1.3 que poseen ya módulos integrados, pero todos fueron analizados para el trabajo.

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”.

2.2.2. Módulo GPS NEO-6M



FIGURA 2.2. Módulo GPS NEO-6M con una antena externa

Para la implementación del módulo de posicionamiento, se decantó por el uso del módulo de GPS NEO-6M desarrollado por la compañía u-blox. Este módulo GPS posee un rango operativo de entre 2.7V y 3.6V, siendo muy similar al microcontrolador elegido anteriormente. Tiene modos de ahorro de energía configurables e interfaces de comunicación UART, SPI y PC[NEO6M:1]. Se analizaron otras alternativas, encontrando mucha similitudes, pero encontrando mayor documentación y casos de uso por parte de la comunidad para el NEO-6M.

Se muestra a continuación una tabla comparativa respecto a otro módulo similar:

TABLA 2.2. Tabla comparativa de placas de desarrollo

Característica	Quectel L86	NEO-6M
Disponibilidad	Moderada	Baja
Costo	Moderado	Elevado
Sistemas GNSS	GPS	GPS, GLONASS, Galileo

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 [9]. Por ejemplo, "el presente trabajo se basa en la plataforma EDU-CIAA-NXP [10], 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.

2.2.3. Módulo GSM SIM800L



FIGURA 2.3. Módulo SIM800L

Para el desarrollo de la conectividad a las redes de telefonía móvil, se utilizó el módulo SIM800L. SIM800L es un módulo de GSM desarrollado por la compañía Simcom. Funciona con un rango de voltaje entre 3.4V y 4.4V, y permite el uso del módulo mediante comandos AT³. Posee mecanismos para optimizar el consumo de energía pero tiene picos elevados de consumo durante la transmisión. Respecto a la compatibilidad de generaciones de tecnologías de telefonía móvil, solo soporta hasta 2G[SIM800L:1]. De todas maneras, para los fines del trabajo, resultó suficiente.

2.2.4. Batería 18650 y chip cargador TP4056



FIGURA 2.4. Batería 18650

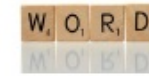


FIGURA 2.5. Circuito Mp4056

Para la alimentación del microcontrolador y sus módulos, se decidió utilizar una batería recargable del tipo 18650⁴ de Li-ion de 2600 mAh. Este tipo de baterías tiene la ventaja de trabajar con un rango de voltaje apropiado para microcontroladores y módulos, pudiendo además tener una elevada corriente máxima de descarga (hasta 2 A aproximadamente).

³Un comando AT consiste en una cadena de caracteres corta utilizada dentro de la industria desde la década de 1980 hasta el día de hoy para la comunicación con módulos, módems, entre otros dispositivos[AT:1].

⁴El nombre 18650 es debido a las medidas de la batería: 18 indican su diámetro, 18 mm, mientras que el número 650 indica su longitud, que es 65 mm.

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>

Debido a que estas baterías requieren un circuito integrado para la carga y el control ante sobrecargas, se utilizó un módulo Tp4056 que permite la carga de la batería con una entrada micro USB de 5V, lo cual posibilita la carga con un cargador común de teléfono celular, por ejemplo.

2.3. Componentes de software utilizados

2.3.1. Django REST framework

Django REST framework es un conjunto de bibliotecas y herramientas del lenguaje Python para el desarrollo de aplicaciones web y APIs REST. Es el framework *de facto* para el desarrollo web en Python. Entre sus características más interesantes, se encuentran [DJANGO:1]

- Soporte incluido para serialización de datos.
- Código modular y reusable mediante *views* y *viewsets*⁵ para definir interfaces REST sobre objetos.
- Mapeo de bases de datos relacionados a objetos mediante un *Object-relational mapping* (ORM) propio.
- Documentación automática para las API por defecto.
- Estable desde hace más de una década.
- Combina la rapidez para el desarrollo y flexibilidad de un lenguaje de tipo dinámico junto a un conjunto de características y estructuras para escalar el desarrollo y que siga siendo mantenible.
- Soporte para múltiples mecanismos de autenticación ya incluidos.
- Bajo uso de recursos.

Se eligió Django para el desarrollo de la API de la aplicación porque se deseó trabajar con un lenguaje más dinámico, pero con un framework que posea ya una estructura para el desarrollo. Esto es debido a que la flexibilidad es un arma de doble filo para la evolución de los sistemas.

TABLA 2.3. Tabla comparativa de frameworks web

Característica	Django	Express
Lenguaje	Python	Javascript
Curva de aprendizaje	Moderada	Muy baja
Arquitectura	Definida	Flexible
Enfoque	Estructurado	Libre/Flexible
Performance	Muy alta	Extremadamente alta
Baterías incluidas ⁶	Sí	No

2.3.2. Angular

Angular es un framework de código abierto desarrollado originalmente por Google para la construcción de aplicaciones web del tipo SPA [ANGULAR:1]. Utiliza

⁵En Django, las diferentes vistas o *views* se agrupan en *viewsets*

```
\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{\sigma^2}{1-k\sigma^2} + \right.
\sigma^2 \left[ d\theta^2 + \sin^2 \theta d\phi^2 \right] \left. \right)
\end{equation}
```

el lenguaje Typescript que es un superset de Javascript, dotándolo de tipado estático y características no presentes en Javascript. Entre sus características más potentes se pueden destacar[ANGULAR:2]

- Framework completo con una estructura definida.
- Arquitectura orientadas en componentes.
- *Binding* de variables bidireccional.
- Disponibilidad de muchas bibliotecas y herramientas.

Para el desarrollo de las interfaces de usuario, se planteó el uso de *Angular Material*, que es una biblioteca de componentes de interfaces disponible para Angular. Su principal ventaja es la disponibilidad de componentes estándar ya armados y diseño común[ANGULAR:3].

Para la incorporación de mapas en la aplicación, se definió el uso de Leaflet, una biblioteca para el desarrollo de mapas en aplicaciones web. Su incorporación a Angular puede hacerse mediante el paquete *ngx-leaflet*[LEAFLET:1] junto con el proveedor gratuito y *open source* de mapas OpenStreetMap.

2.3.3. PostgreSQL

PostgreSQL es un *relational database management system* o RDBMS de código abierto que hace uso de SQL y es famoso por ser *ACID-compliant*, su gran amplitud de tipos de datos, robustez y extensibilidad[POSTGRES:1]. Es utilizado ampliamente por la industria del software[POSTGRES:2] y se encuentra disponible como una opción por los mayores proveedores de *cloud* de la actualidad. Prácticamente todos los frameworks web permiten conectarse a este motor de bases de datos.

2.3.4. Heroku

Heroku es un proveedor de infraestructura y servicios en la nube y *Platform as a Service* (PaaS) focalizado en el rápido despliegue y mantenimiento de las aplicaciones web. La principal ventaja de Heroku es que apunta a abstraer al usuario de detalles técnicos y la infraestructura que da soporte a los servicios usados.

Se presenta una breve tabla comparativa frente a otros proveedores, como Amazon Web Services (AWS) y Google Cloud Platform (GCP).

TABLA 2.4. Tabla comparativa de proveedores *cloud*

Característica	Heroku	AWS	GCP
Complejidad	Baja	Muy alta	Media-Alta
Costo	Medio	Bajo	Bajo-Medio
Flexibilidad	Baja-Media	Muy alta	Alta
Servicios	Poca oferta	Mucha oferta	Mucha oferta
Regiones	Limitadas	Amplias	Amplias

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}
```

2.4. Herramientas y servicios de software utilizados

2.4.1. Docker

Docker es una plataforma *open source* para la contenerización de aplicaciones, con el objetivo de[DOCKER:1]:

- Generar versiones de sistemas idénticas entre diferentes ambientes, es decir que un sistema sea portable.
- Aislar los recursos del *host* de los sistemas que corren en él.
- Reducir los tiempos de despliegue, mantenimiento y soporte de aplicaciones.

Docker permite generar imágenes de sistemas siguiendo un conjunto de instrucciones definidos mediante un archivo *Dockerfile*, habilitando generar estas mismas imágenes en otros ambientes.

2.4.2. Visual Studio Code

Visual Studio Code es un editor de texto e IDE (Integrated development environment o Entorno de desarrollo integrado) de código abierto desarrollado originalmente por Microsoft[VSCODE:1]. Su flexibilidad y soporte para una cantidad considerable de extensiones para gran parte de los lenguajes y frameworks de la industria lo posicionó en la última década como uno de los IDE más utilizados.

Permite desarrollar aplicaciones con Python, Javascript/Typescript, incorporando incluso frameworks para sistemas embebidos como ESP-IDF o Platform.IO desarrollo embebidos como es el framework ESP-IDF o PlatformIO⁷. La mayoría de proveedores de servicios en la nube han desarrollado extensiones para agilizar el despliegue de aplicaciones en sus servicios.

2.4.3. PyGPSClient

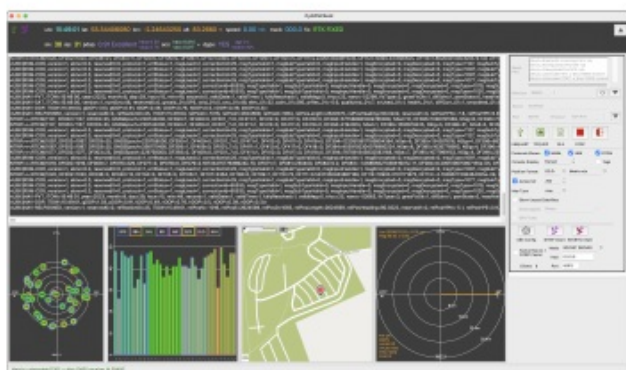


FIGURA 2.6. Aplicación PyGPSClient

⁷Ver [Installation · PlatformIO](#) o [Getting started with ESP-IDF VS Code - ESP32](#).

PyGPSClient es una herramienta de escritorio disponible desarrollada en Python para el *debugging* y análisis de varios módulos GNSS de u-blox. Su principal ventaja frente a la herramienta estándar para las pruebas de estos módulos, u-center, es que está disponible para cualquier sistema que soporte Python, en comparación de la anterior que solo está disponible para Windows[PYGPSCIENT:1].

2.4.4. Moserial

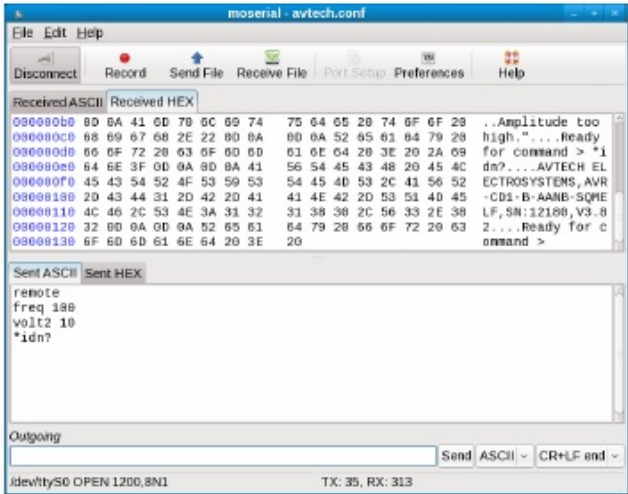


FIGURA 2.7. Aplicación Moserial

Moserial es una aplicación de escritorio orientada al *testing* de dispositivos embebidos, consolas y equipos electrónicos que permiten comunicación vía *Universal Asynchronous Receiver-Transmitter* o UART. Se encuentra disponible para el entorno gráfico GNOME, popular en sistemas GNU/Linux. Su uso resultó de importancia para la comunicación con el módulo de GSM, SIM800L.

2.4.5. Twilio

Twilio es un *Communications Platform as a Service* o CPaaS que ofrece diferentes servicios y herramientas para el desarrollo de sistemas que requieran funcionalidades relacionadas a telefonía IP, envío y recepción de mensajes de texto, accediendo a dicho servicios mediante varios mecanismos ofrecidos por la plataforma, como APIs web[TWILIO:1]. Dentro del trabajo, su principal uso fue para la adquisición de un número de teléfono virtual para la recepción de mensajes de texto provenientes de un dispositivo embebido.

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 epigrafe 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 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 epigrafe 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 }
```

CÓDIGO 3.1. Pseudocódigo del lazo principal de control.

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 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.

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 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.

Bibliografía

- [1] Programa de las Naciones Unidas para el Desarrollo. *Sinopsis: Seguridad Ciudadana*. <https://www.undp.org/es/publications/sinopsis-seguridad-ciudadana>. Abr. de 2014. (Visitado 15-04-2014).
- [2] Traccar. *Chinese Clones*. <https://www.traccar.org/clones/>.
- [3] TKSTAR LK109 USER MANUAL. <https://www.manualslib.com/manual/1035746/Tkstar-Lk109.html#manual>.
- [4] TK102B Manual.pdf. <https://www.munstergps.ie/wp-content/uploads/TK102B-Manual.pdf>.
- [5] John Doe. «Title». En: *Journal* (2017).
- [6] John Doe. *The Book without Title*. Dummy Publisher, 2100.
- [7] John Doe. «The Book without Title». En: Dummy Publisher, 2100, págs. 100-200.
- [8] Intel. *Example Website*. <http://example.com>. Dic. de 1988. (Visitado 26-11-2012).
- [9] IEEE. *IEEE Citation Reference*. 1.ª ed. IEEE Publications, 2016. URL: <http://www.ieee.org/documents/ieeecitationref.pdf> (visitado 26-09-2016).
- [10] Proyecto CIAA. *Computadora Industrial Abierta Argentina*. Visitado el 2016-06-25. 2014. URL: <http://proyecto-ciaa.com.ar/devwiki/doku.php?id=start>.