

# Módulo de sistema de sensores para rastreo de animales

Allison Aldoradin, Fortunato Alvarado, Miguel Belén, Yanpierss Figueroa, Diego Palma, Luis Velasquez

Universidad de Ingeniería y Tecnología – UTEC, Lima, Perú

allison.al doradin, fortunato.alvarado,miguel.belen, yanpierss.figueroa, diego.palma, luis.velasquez  
@utec.edu.pe

**Abstract**—El presente informe plantea el desarrollo de un módulo de sistema de sensores para el rastreo de pingüinos. Este módulo de rastreo consta de sensores de bajo costo, un microcontrolador, y diversos componentes electrónicos. Entre los sensores, tenemos al reloj de tiempo real RTC ZS-042, el cual nos sirve para conocer la fecha y hora en la cual se realiza cada toma de datos de los sensores; el GPS GY-GPS6MY2, que nos entrega las coordenadas de latitud y longitud para conocer la ubicación del pingüino en tierra; el sensor de temperatura LM35, que mide la temperatura del pingüino tanto en la tierra como durante la inmersión; la unidad de medición inercial (IMU) MPU6050 de 6 grados de libertad, la cual nos brinda los valores de aceleración y velocidad angular, que nos permitirán conocer el comportamiento del pingüino y la profundidad a la cual se sumergen. En adición a todo ello, se hace uso de una tarjeta micro SD y un módulo adaptador de micro SD, en la cual se almacenan todos los parámetros mencionados. Por otro lado, el módulo presenta un pulsador de inicialización y de seguridad que sirven para iniciar el registro de datos y la extracción segura de la tarjeta micro SD.

## I. INTRODUCCIÓN

En la planta de licuefacción de gas natural de Pampa Melchorita se construyó un rompeolas con el objetivo de atenuar la fuerza de las olas y facilitar la carga de gas natural licuado (GNL) en los buques metaneros[1]. Si bien esa es su principal función, hoy en día la estructura ha permitido la creación de un nuevo ecosistema que alberga a 108 especies de plantas y animales marinos[2]. Dentro de ellas, se destaca una colonia de pingüinos de Humboldt (Fig. 1), especie que en los últimos años sus poblaciones se han reducido drásticamente a consecuencia de la escasez de lugares de nidificación, reducción en la oferta de alimentos, depredación por parte del hombre y, sobre todo, el impacto de fenómenos climáticos. ¿De dónde vinieron?, ¿por qué se establecieron? y ¿cómo usan el área?, son algunas interrogantes que este hecho ha generado, y de ser respondidas aportarán gran información para futuras tomas de decisiones relacionadas al rompeolas y dicha especie marina.

En este sentido, es necesario realizar un rastreo y monitoreo de los pingüinos de Humboldt para conocer cómo utilizan el área, la profundidad máxima de inmersión en el mar para pescar, su comportamiento en a lo largo del día y la temperatura de su entorno. Dicha información puede ser recolectada

a través de tecnologías de rastreo, sin embargo, actualmente en el mercado estas son costosas (1200–1500), difíciles de adquirir y muchas veces, con parámetros limitados. Por dichas razones, se busca desarrollar un módulo de bajo costo que permita el rastreo de esta especie y, para ello, debe cumplir con las siguientes consideraciones: no debe superar el 3% de su peso (140 g.), debe ser de dimensiones coherentes a su tamaño, la duración de batería suficiente para almacenar los datos requeridos y que la geometría del dispositivo no debe afectar las actividades rutinarias del pingüino.

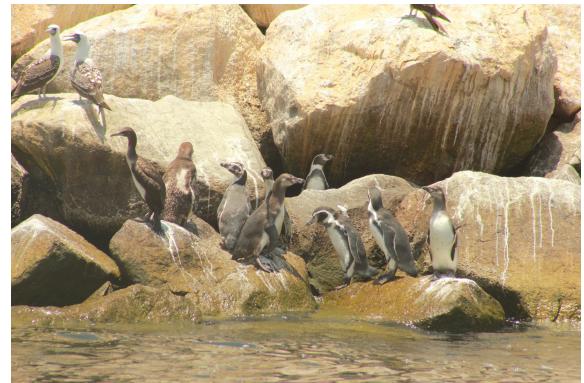


Fig. 1. Pingüinos de Humboldt en la Pampa Melchorita

## II. METODOLOGÍA

En primer lugar, se realizó un diagrama de bloques (Fig. 2) que representa el funcionamiento interno del módulo. En él, se muestra los sensores utilizados, los pulsadores, el microcontrolador ATmega328P y la tarjeta micro SD. Además, se presenta la fase de post-procesamiento, en la cual se procesarán los datos brutos recolectados para una mejor visualización de la información obtenida.

A partir del diagrama de bloques, se dividió el desarrollo del módulo en tres etapas: Diseño electrónico, diseño de software y diseño mecánico. A continuación, se presenta y se explica detalladamente en qué consiste cada una de estas etapas:

### A. Diseño Electrónico

El módulo requiere una tarjeta de circuito impresa para el montaje de los sensores y componentes electrónicos. Por esta razón, se diseña la tarjeta en base a las consideraciones

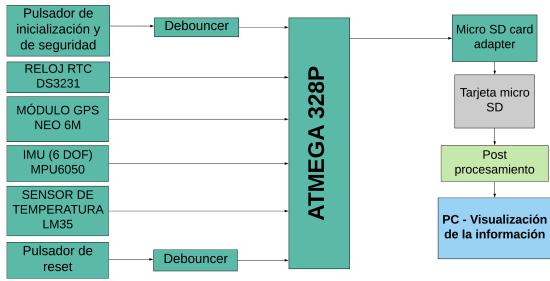


Fig. 2. Diagrama de bloques del módulo de sistema de sensores de rastreo de animales.

anteriormente expuestas. El diseño de la tarjeta se realizó con el uso del software EAGLE versión 8.6.3 e implicó el desarrollo del esquemático y la ubicación de los componentes.

Por un lado, se realizó un esquemático (Fig. 3) del circuito eléctrico, donde se muestra los cables y conexiones necesarios. Asimismo, se definió los pines correspondientes a cada sensor y su respectiva función; las entradas y salidas, donde se consideró la facilidad que brindan los pines del Atmega 328P para su futura programación. Por otro lado, la ubicación de los sensores y componentes en la tarjeta electrónica se diseñó con el fin de conseguir las mínimas dimensiones posibles, tanto en ancho, largo y altura. Por esta razón, en la etapa de ruteo, la tarjeta de circuito impreso fue de doble capa (Fig. 4) y además, se utilizaron componentes de tecnología de montaje superficial (SMT).

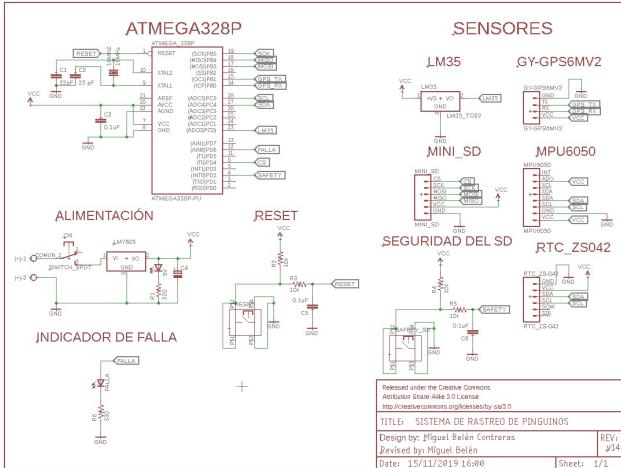


Fig. 3. Diagrama esquemático del módulo.

Esta etapa incluye también la edición del footprint de los componentes electrónicos y la creación de componentes electrónicos que no se encuentran en alguna librería de EAGLE o de distribuidores de componentes. Por ejemplo, en el caso del pulsador solo se encontró las dimensiones del componente y por ende, se diseñó el dispositivo, encapsulado y su símbolo. Asimismo, esta etapa necesita la constante

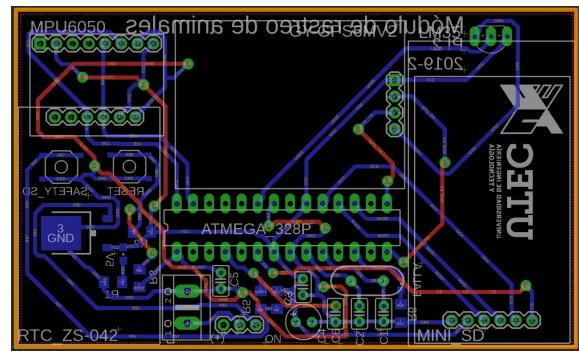


Fig. 4. Ubicación y ruteado de los sensores y componentes electrónicos.

verificación de cada componente para que no existan fallas posteriores en el montaje de los elementos. También, se considera la serigrafía apropiada en la tarjeta, lo cual incluye: la ubicación adecuada de los nombres de los componentes para la soldadura de los componentes, indicaciones para la fuente de alimentación, el uso de los leds de encendido y de falla, las indicaciones para ejecutar correctamente los pulsadores de inicialización y reset, los límites de los componentes, etc.

#### B. Diseño del Software

El diseño del software comprende dos sub-etapas: la lógica de la programación del módulo y el post-procesamiento de los datos.

- Lógica de la programación del módulo:** En esta etapa se establece los puntos de entrada y salida de los datos de acuerdo al esquemático. En este caso, los puntos de entrada son: el pulsador, el IMU, el módulo GPS, módulo RTC y el sensor de temperatura; y el punto de salida es el módulo memoria SD [4]. La lógica de la programación comienza con la inicialización de los módulos de sensores para hacer una revisión de estado de los mismos, antes de entrar al programa principal, adicionalmente se configura la forma de trabajo de la entrada del pulsador, este es el concepto de “SETUP”. En esta parte, se da los valores de offset del IMU debido a que estos se eliminan en el momento de desenergizarlos. Asimismo, se incluye una función que verifica el estado de los sensores denominado “check-estatus”, si la función no retorna ningún error en esa parte se procede al programa principal.

En el programa principal, se desarrolló un diagrama de flujo (Fig. 5) para entender la lógica de funcionamiento. El programa principal solo se iniciará con la función denominada iniciación, en esta función toma el estatus del pulsador, es decir, detecta la activación del pulsador, cuando este sea detectada dará paso al módulo de sensores. Cada sensor tiene una función propia donde se utilizó las librerías asociadas a ellos para tomar los datos respectivos de cada uno. Una vez que los datos fueron recolectados, estos son guardados en la memoria SD a través de una comunicación tipo interface serial en un formato de fecha,hora,temperatura,IMU[0,6], GPS[0,2].En

principio, la frecuencia de muestreo fue de aproximado de 1 Hz, pero por temas de post procesamiento se aumentó a 12 Hz.

Debido a que la recolección y guardado de datos es de forma continua, se estableció un retiro seguro de la memoria SD que consiste en parar el “muestreo y guardado”, es decir, reiniciar el sistema las estado inicial y esto hará que el módulo SD no se encuentre operando al momento de retirar la memoria SD y así evitar pérdidas de data o errores en ella. La data guardada se encontraba en principio en formato txt pero por temas de post procesamiento se pasó a un formato CSV. Cabe mencionar que el programa fue desarrollado en la plataforma de Arduino debido a las librerías implementadas de cada sensor y los tipos de comunicación utilizados.

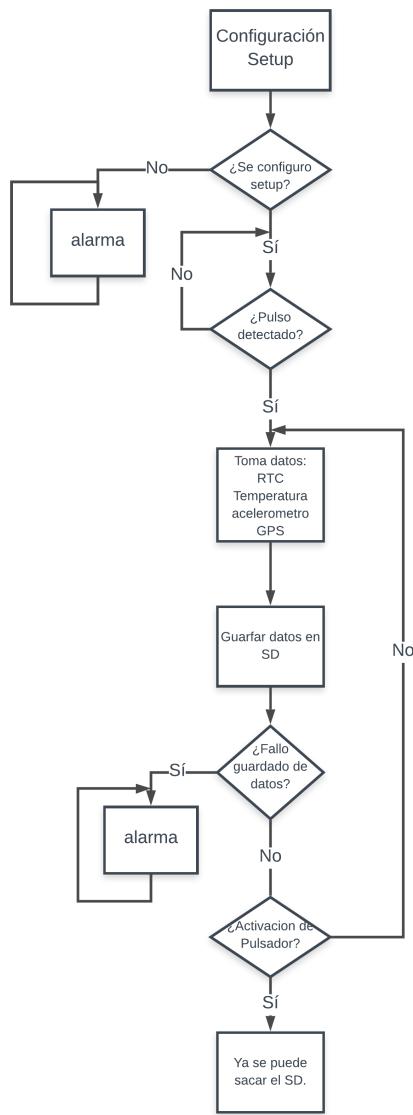


Fig. 5. Diagrama de flujo de la lógica del módulo.

- **Post-procesamiento de los datos:** Esta etapa se desarrollará en el lenguaje de programación Python y consistirá en el análisis de la data obtenida por los sensores del módulo. A continuación, se muestra qué valores se obtendrán a partir de los sensores utilizados y cómo estos serán procesados:

- La fecha y hora: Estos valores son obtenidas por El RTC ZS-042, y serán de gran utilidad para conocer en qué momento se realizó la toma de datos y, posteriormente, realizar gráficas en función del tiempo.
- Las coordenadas de latitud y longitud: Estas coordenadas son brindadas por el GPS [3], y a partir de dichos valores se graficará un mapa, en el cual se mostrará la ubicación del dispositivo en tierra.
- La temperatura: Estos datos, obtenidos por el sensor LM35, junto a los valores obtenidos por el RTC, nos permitirán realizar una gráfica de temperatura vs tiempo, la cual nos ayudará a visualizar el rango de temperaturas a las cuáles el pingüino está sometido.
- La aceleración y velocidad angular: Estos valores serán conocidos gracias al IMU MPU-6050. Por un lado, de acuerdo con YODA (2001), la aceleración en los ejes X, Y y Z nos permitirá describir el comportamiento del pingüino, tal como se muestra en Fig. 6. Por otro lado, para determinar la profundidad de inmersión del pingüino se utilizará las aceleraciones y velocidades angulares en los 3 ejes, junto a un método matemático (Fig. 7), el cual consta de dos etapas. En la primera etapa, se realiza una integración de las velocidades angulares con respecto al tiempo, de lo cual obtenemos los valores del roll, pitch y yaw; y a partir de ello, construimos una matriz de rotación (MR) que nos permite obtener los valores de las aceleraciones en el sistema de referencia convencional X, Y y Z. En la segundo etapa, se busca obtener la aceleración dinámica en el eje Z, para finalmente realizar dos integraciones con respecto al tiempo y obtener la profundidad de inmersión. Para lograr ello, multiplicamos MR con el vector de aceleraciones y restamos el factor de la gravedad al valor del eje Z.

### C. Diseño mecánico

El placa electrónica donde se encontrará implementados todos los sensores estará protegida por un case, el cual se encontrará unido a la superficie exterior del pingüino. En este sentido, se busca minimizar la fuerza de arrastre y las alteraciones en el nado del pingüino a través de la optimización del diseño mecánico. Durante este proceso de optimización, se buscó que la base presente una pequeña curvatura con el fin de mejorar el acople a la espalda del pingüino. Asimismo, se

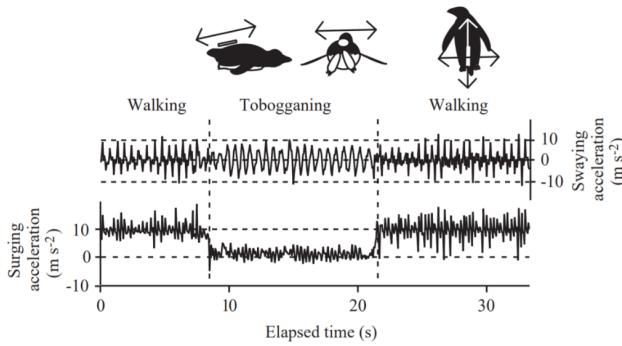


Fig. 6. Comportamiento del pingüino a partir de la gráfica de aceleración vs tiempo.

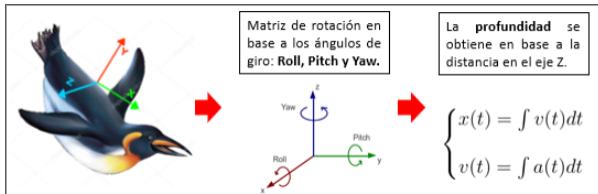


Fig. 7. Cálculo de la profundidad a partir de la aceleración en el eje Z.

prioriza que el case tenga una mayor medida en largo que ancho. Finalmente, se toma en cuenta las dimensiones de la placa electrónica y la batería que se iba a utilizar, la cual básicamente delimita la altura del case. Para el desarrollo del diseño mecánico, se utilizó el software Fusion 360 como se muestra en la Fig. 8.

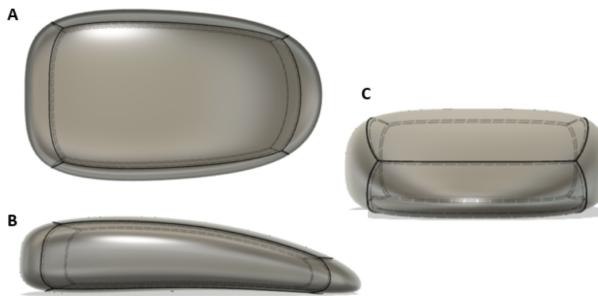


Fig. 8. Diseño final del encapsulado

### III. RESULTADOS

A partir de la metodología descrita anteriormente, se construyó el dispositivo final cuyas dimensiones son: 8.9 cm de base, 5.3 de ancho y 2.4 cm altura. El peso resultante del módulo es de 126 gramos, posee un tiempo de operación de 3 horas y un costo de fabricación de 120 soles. La Fig. 9. muestra el resultado del diseño de la tarjeta de circuito impreso con todos los módulos planteados.

Además, se realizaron pruebas para validar el correcto funcionamiento de los sensores y los datos que estos nos brindan. En el caso del sensor de GPS, se realizó una prueba en los alrededores de UTEC para obtener las coordenadas



Fig. 9. Circuito impreso con los sensores implementados

de localización. Luego, se procesaron estos datos, usando la librería Folium de Python, para graficarlos en un mapa, lo cual se muestra en Fig. 10.

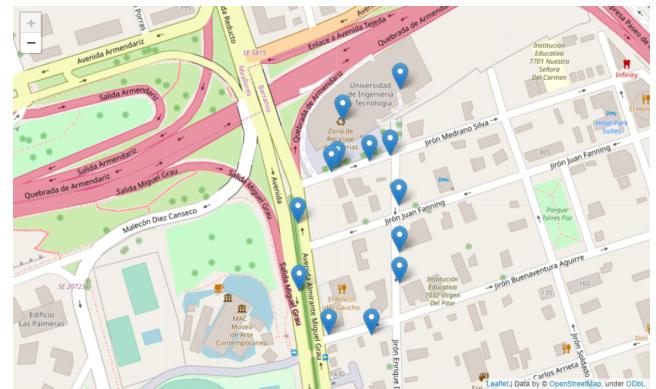


Fig. 10. Mapa con las localizaciones obtenidas por el GPS

Con respecto a los datos obtenidos por los sensores de temperatura y el módulo RTC, se realizó una gráfica Temperatura vs tiempo, la cual nos permite determinar el valor de la temperatura a cualquier hora. Esto se puede observar en la Fig. 11.

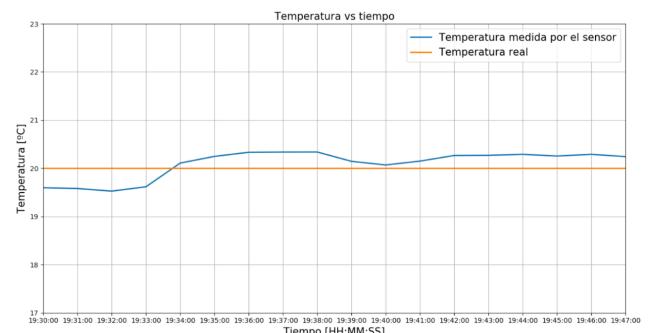


Fig. 11. Gráfica de Temperatura vs Tiempo

Con respecto a los valores obtenidos por el IMU - MPU 6050, se realizaron gráficas de aceleración en los ejes X, Y y

Z. La Fig. 12. es una prueba de recorrido, y en dicho gráfico se puede observar que la aceleración en Z se mantiene en aproximadamente  $9.8 \text{ m/s}^2$ , el cual corresponde al valor de la gravedad; en los ejes Y y Z se observan pequeños desplazamientos que corresponden, según el paper, a una persona caminando, lo cual corresponde a la prueba realizada. Por otro lado, para validar el código para determinar la profundidad, se realizó una pequeña prueba que consistía en medir la altura de una persona. Es decir, se tomó como base la cabeza de una persona y se llevó el dispositivo hasta el suelo, y se graficaron los datos en el tiempo (Fig. 13).

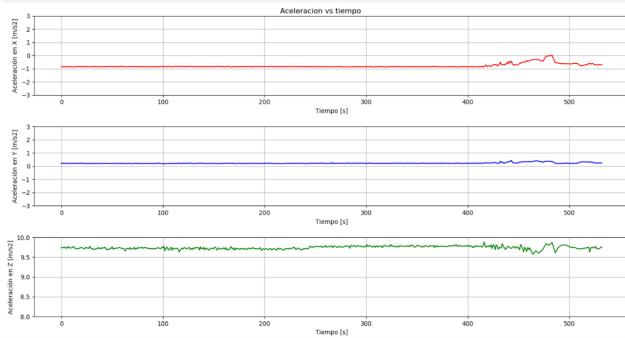


Fig. 12. Gráfica de Aceleración vs Tiempo



Fig. 13. Gráfica de Profundidad vs Tiempo

Cabe mencionar, que se realizaron pruebas similares a la anterior, pero con objetos más grandes, como la universidad UTEC. En dicha prueba, pudimos observar que el error en la profundidad aumentó enormemente (300%), por lo que, para realizar un mejor uso de esta técnica se recomienda utilizar filtros digitales en la programación, como el filtro Kalman, filtro pasa bajas, etc. Esto, con el fin de reducir el error en dicho parámetro.

Con respecto al diseño mecánico, este fue impreso en 3D con el material PLA (Fig. 14), el grosor de la capa de impresión fue 1.6 mm. Las dimensiones finales del case fueron  $12 \times 6 \times 2.5 \text{ cm}$ , las cuales pueden reducirse aún más pues cuando se colocó la placa electrónica, había un espacio vacío de 1.5 cm. Asimismo, la altura se puede disminuir utilizando otro tipo de batería que sea más plana para así reducir la

turbulencia causada por la parte trasera de el dispositivo. En adición a ello, para lograr que el case sea impermeable se recomienda recubrir con resina epoxi.

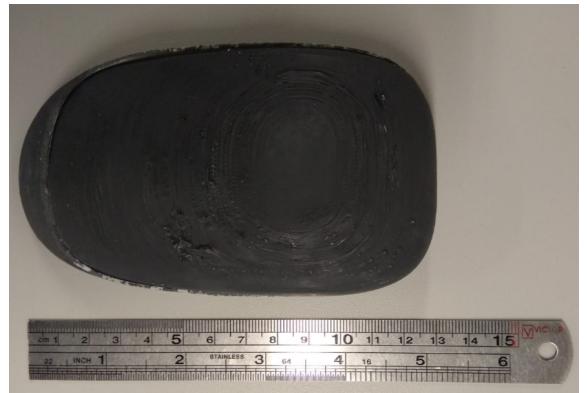


Fig. 14. Imagen del case impreso en 3D

#### IV. CONCLUSIONES

En conclusión, se logró desarrollar un módulo de sistema de sensores que cumple con las especificaciones requeridas en cuanto al peso final, el cual fue de 126 gramos y tiene un tiempo de operación de 3h. Además, se diseñó la lógica para la recolección de datos en el micro SD, el postprocesamiento de los datos y se obtuvo la información necesaria de acuerdo a las especificaciones propuestas.

Con respecto a las pruebas realizadas, se obtuvieron las gráficas esperadas y mencionadas en la metodología. Sin embargo, aún existe un error muy significativo en los datos de profundidad, el cual es debido a la acumulación del error en cada integración. Para reducir dicho error, se recomienda utilizar un IMU de 9 grados y filtros adicionales en el post-procesamiento o en el microcontrolador, como el filtro Kalman.

#### REFERENCES

- [1] OSIGNERMIN, “Cultura del gas natural”. Disponible en <http://srvgart07.osinerg.gob.pe/webdgn/contenido/cult003.html>.
- [2] Peru LNG, “La riqueza del rompeolas de Melchorita”. Disponible en [https://perulng.com/wp-content/uploads/2016/05/Infografia\\_Pingu%C3%adnino.pdf](https://perulng.com/wp-content/uploads/2016/05/Infografia_Pingu%C3%ADnino.pdf).
- [3] Santos, S. (2018, 4 abril). Guide to NEO-6M GPS Module with Arduino. Recuperado 23 septiembre, 2019, de <https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/>
- [4] ElectroPeak. (2019, 16 septiembre). SD Card Module with Arduino: How to Read/Write Data. Recuperado 25 septiembre, 2019, de <https://create.arduino.cc/projecthub/electropeak/sd-card-module-with-arduino-how-to-read-write-data-37f390>