

**Universidad Tecnológica Nacional**

**Proyecto Final**

---

**Lector de mano portátil para caravanas de  
identificación electrónica**

---

*Autores:*

- Velázquez, Miguel Oscar

*Director:*

- Ing. Maggiolo, Gustavo

*Proyecto final presentado para cumplimentar los requisitos académicos  
para acceder al título de Ingeniero en Electrónica*

*en la*

**Facultad Regional Paraná**

Noviembre de 2019



## **Declaración de autoría:**

Declaro que el Proyecto Final “Lector de mano portátil para caravanas de identificación electrónica” y el trabajo realizado son propios.

Declaro:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero en Electrónica, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

- 
- 
- 

Fecha:



## Agradecimientos:

Quiero agradecer en primer lugar a mi madre por todo el esfuerzo que realizo para que pueda lograr este cometido. También agradecerles a mis amigos y compañeros de facultad, que gracias a ellos estoy logrando este título universitario. A la facultad por brindarme todas las herramientas para poder realizar la carrera de ingeniero y por supuesto a los ingenieros docentes que fueron formándome a lo largo del cursado.



Universidad Tecnológica Nacional

*Abstract*

Facultad Regional Paraná

Ingeniero en Electrónica

**Lector de mano portátil para caravanas de identificación  
electrónica**

Velázquez, Miguel Oscar

**Abstract:**

This document details the different design and programming instances of a portable handheld reader for electronic identification caravans, which is accompanied by an executable program made in Microsoft Visual Studio 2019 and with its corresponding user manual.

For the reading of caravans, which work with RFID (Radio Frequency IDentification), a reader module compatible with the ISO standards established for animal identification (ISO 11784 and ISO 11785) was used. Communication between electronic scale and reader is done through a Bluetooth module (HC-05).

In the caravan reading test, the reader has a range of approximately 5 centimeters distance between the caravan and the transmitting antenna, in bluetooth communication you reach up to 4 meters between electronic balance and reader stick. All the information collected is stored on a microSD card of the instrument, which, after connecting the reader stick to the PC, is extracted using the designed software. The visualization of said data can be seen on the screen as well as average weighing, weight evolution and the possibility of knowing the value, in pesos, of the batch or lots that were read.

**Keywords:**

RFID (Radio Frequency IDentification), Electronic Caravans, Walking Stick, XML (Extensible Markup Language), C#, Modules, IDE (integrated development environment).

**Resumen:**

En el presente documento se detalla las diferentes instancias de diseño y programación de un lector de mano portátil para caravanas de identificación electrónica, el cual va acompañado de un programa ejecutable realizado en *Microsoft Visual Studio 2019* y con su correspondiente manual de usuario.

Para la lectura de caravanas, las cuales trabajan con RFID (Radio Frequency IDentification), se utilizó un módulo lector compatible con las normas ISO establecidas para la identificación animal (ISO 11784 e ISO 11785). La comunicación entre balanza electrónica y lector se realiza a través de un módulo Bluetooth (HC-05).

En la prueba de lectura de caravanas el lector tiene un alcance de aproximadamente 5 centímetros de distancia entre la caravana y la antena emisora, en la comunicación bluetooth se llega hasta los 4 metros de distancia entre balanza electrónica y bastón lector. Toda la información recaudada se almacena en una tarjeta microSD propia del instrumento que luego, conectando el bastón lector a la pc, se extrae mediante el software diseñado. La visualización de dichos datos se aprecia por pantalla como así también promedios de pesajes, evolución de pesos y la posibilidad de conocer el valor, en pesos, del o los lotes que se tomó lectura.

**Palabras Clave:**

RFID (Identificación por radiofrecuencia), Caravanas Electrónicas, Bastón Lector, XML (Lenguaje de Marcado Extensible), C#, Módulos, IDE (entorno de desarrollo integrado).



## *Reconocimientos:*

Quiero agradecer a mi docente director el Ing. Gustavo Maggiolo por brindarme siempre la mejor opción de encarar el proyecto, métodos y soluciones cuando había dudas en la programación del mismo.

A mi amigo e Ing. Nicolás Wagner por su gran colaboración en el proyecto, a mi amigo Diego Heit por el mismo motivo, a mis compañeros y amigos de facultad Gastón Aquino, Matías González y Julio Fállico por colaborar en distintas partes del proyecto, aconsejando cual sería la mejor forma de resolver el problema. Muchas Gracias.

# Índice:

Capítulo 1: Introducción.....	18
Capítulo 2: Desarrollo .....	19
2.1 Diagrama de Bloques.....	19
2.2 Caravana Electrónica .....	19
2.2.1 Half Duplex .....	21
2.2.2 Full Duplex.....	21
2.3 Balanza Electrónica.....	22
2.4 Bastón Lector .....	24
2.4.1 Modulo Bluetooth HC-05 .....	25
2.4.2 Teclado Matricial 4x4 .....	30
2.4.3 Pantalla LCD Nokia 5110.....	31
2.4.4 Buzzer Pasivo.....	33
2.4.5 Pulsador.....	34
2.4.6 Reloj de tiempo real “RTC” (DS3231).....	35
2.4.7 Modulo adaptador de memoria microSD .....	37
2.4.8 Lector RFIDRW-E-TTL .....	39
2.4.9 Fuente de alimentación.....	43
2.4.10 Arduino ATmega 2560.....	45
2.4.11 Puerto Serie .....	50
2.5 Software de pc .....	51
2.5.1 Información .....	52
2.5.2 Conexión.....	52
2.5.3 Base de datos.....	54
2.5.4 Rendimiento.....	57
2.6 Problemas y soluciones en Bastón lector y software .....	59
2.7 Diseño Completo.....	61
Capítulo 3: Resultados.....	69
3.1 Resultados obtenidos.....	69
Capítulo 4: Análisis de Costos .....	70
4.1 Análisis de costos:.....	70
Capítulo 5: Discusión y Conclusión.....	72
5.1 Conclusión y mejoras a futuro .....	72
Capítulo 6: Literatura Citada. ....	73

## Lista de Figuras:

Imagen 1: Colocación de caravanas.....	20
Imagen 2: Caravana Electrónica.....	20
Imagen 3: Caravana Electrónica HDX .....	21
Imagen 4: Caravana Electrónica FDX.....	22
Imagen 5: Logo de la aplicación .....	23
Imagen 6: Serial Bluetooth Terminal 1.28 .....	24
Imagen 7: Conexión HC-05 a Placa arduino.....	27
Imagen 8: Comandos AT .....	28
Imagen 9: Serial Monitor.....	29
Imagen 10: Teclado Matricial 4x4.....	31
Imagen 11: Display Nokia 5110.....	32
Imagen 12: Conexión Display Nokia 5110 .....	33
Imagen 13: Buzzer pasivo .....	34
Imagen 14: Circuito Antirrebote .....	35
Imagen 15: Modulo RTC.....	36
Imagen 16: Conexión RTC .....	37
Imagen 17: Modulo MicroSD .....	38
Imagen 18: Conexión Modulo MicroSD .....	39
Imagen 19: Modulo RFID.....	40
Imagen 20: Conexión antena RFID .....	41
Imagen 21:RFIDCOIL-49A.....	43
Imagen 22: Fuente de alimentación.....	45
Imagen 23: Inicio IDE.....	46
Imagen 24: Sketch del IDE .....	47
Imagen 25: Documento XML .....	49
Imagen 26: Cable USB .....	50
Imagen 27: Visual Studio 2019 .....	51
Imagen 28: Pestaña de información .....	52
Imagen 29: Pestaña de Conexión.....	53
Imagen 30: Pestaña de base de datos .....	55
Imagen 31: Pestaña de Rendimiento.....	57
Imagen 32: Histograma.....	58
Imagen 33: Logotipo Notepad++.....	60
Imagen 34: Diseño en SketchUp .....	61

Imagen 35: Prototipo Lector de mano portátil .....	62
Imagen 36: Compartimiento de batería.....	63
Imagen 37: Menú principal.....	63
Imagen 38: Selección del Corral .....	64
Imagen 39: Lectura de caravana .....	64
Imagen 40: Comunicación con la balanza .....	65
Imagen 41: Habilitación del pulsador. ....	65
Imagen 42: Lectura Correcta. ....	66
Imagen 43: Datos almacenados en memoria. ....	67
Imagen 44: Datos del RTC. ....	67
Imagen 45: Descarga del documento XML.....	68

## Lista de Diagramas:

Diagrama 1: Bloques Principales .....	19
Diagrama 2: Módulos del Bastón Lector .....	24
Diagrama 3: Código de lectura y almacenamiento de datos.....	48
Diagrama 4: Conexión y actualización de la base de datos.....	54
Diagrama 5: Consulta a la base de datos XML.....	56
Diagrama 6: Histograma de rendimiento .....	59

# Lista de Tablas

Tabla 1: Gastos en fabricación y diseño .....70

## **Lista de Abreviaciones**

RFID: Identificación por radiofrecuencia.

IDE: Entorno de desarrollo integrado.

XML: Lenguaje de Marcado Extensible.

Tx: Transmisión de datos por puerto serie.

Rx: Recepción de datos por puerto serie.

I2C: Inter-Integrated Circuit.

SPI: Serial Peripheral Interface.

## **Dedicado a:**

Mi familia, especialmente a mi madre.





## Capítulo 1: Introducción

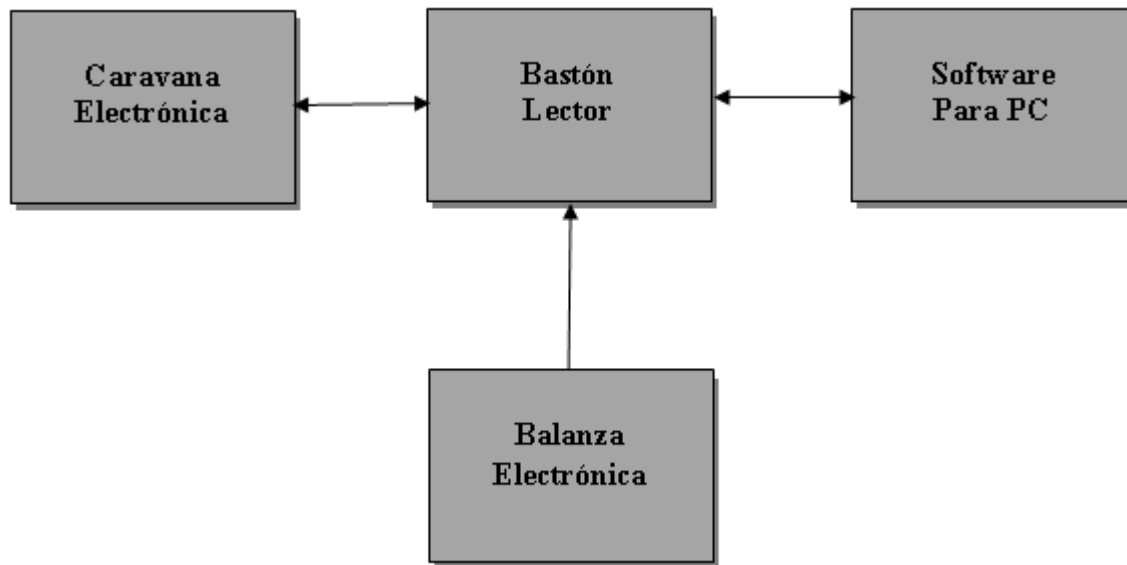
El constante avance tecnológico en el sector agrícola del campo argentino es extraordinario, maquinarias de última tecnología, avances en la genética de las semillas y demás, no tan así en el sector ganadero. Si bien la tecnología en la parte ganadera va en alza es muy cara para los productores de zonas con no tanto poder adquisitivo, me refiero con esto a pequeños productores ganaderos. Al ser nuestro país un gran exportador de carne vacuna, cada vez se ve más en zonas rurales corrales para engorde de animales. Muchos de estos propietarios no presentan tecnología alguna para la trazabilidad del producto. El Servicio nacional de sanidad y calidad agroalimentaria (SENASA) define como trazabilidad a *“la relación ininterrumpida del animal desde su nacimiento hasta los productos derivados de la faena de ese animal, comercializados y puestos a disposición del consumidor. Es la posibilidad de reencontrar esos datos, los antecedentes, la locación de una entidad, mediante identificaciones registradas”*.

Debido a lo mencionado anteriormente mi proyecto está relacionado con la parte de engorde a corral de bovinos (Feedlot) desarrollando un bastón lector de caravanas electrónicas. La caravana (también llamado transpondedor) contiene un único código de identificación que el bastón lector leerá como también la comunicación con una balanza electrónica que informará el peso de dicho bovino en el momento de lectura de la misma.

Este dispositivo tiene la particularidad de ser más económico y versátil de los que están en el mercado, los cuales dependiendo de sus características son de precios muy elevados, inaccesibles para los pequeños y medianos productores ganaderos. El instrumento contendrá en su memoria interna los datos (número de caravana, peso, fecha y demás) que fueron tomados en el momento que pasan por la balanza. Esta información luego será descargada en la computadora del propietario que contendrá toda la información, del o los lotes, desde la primera lectura de caravanas. En el software de pc se podrá apreciar la evolución del engorde de animales teniendo así un mayor control de producción tanto del lote como de cada animal, muchos lectores en el mercado presentan un software simple comparado con el que se mencionó. El objetivo del proyecto es mejorar la eficiencia, la precisión y el control en el manejo de hacienda y por supuesto incrementar las ganancias.

## Capítulo 2: Desarrollo

### 2.1 Diagrama de Bloques



*Diagrama 1: Bloques Principales*

El diagrama presenta cuatro bloques bien definidos que son las partes principales del proyecto, el primer bloque hace referencias a las caravanas electrónicas que el segundo bloque (ubicado en el centro) deberá leer. El bloque ubicado en la parte inferior corresponde a la balanza electrónica que enviara un dato (peso del animal) al instrumento y por último el bloque de software de pc que es el encargado de extraer los datos almacenados en la memoria del bastón lector y mostrar por pantalla toda la información recaudada como así también su base de datos con antiguas lecturas. Las flechas indican los sentidos de comunicación entre ellos.

A continuación, serán explicados con más detalles cada bloque del sistema.

### 2.2 Caravana Electrónica

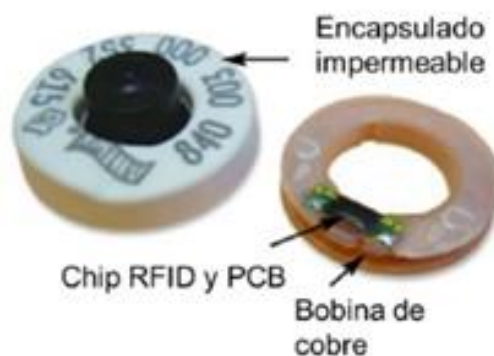
La función principal del proyecto es leer caravanas electrónicas del tipo botón (Imagen 2), las cuales cuentan con un código único de identificación del animal. Están compuestas por dos piezas (macho y hembra) que se encastran, debido a esto es a veces imposible retirarlas de la oreja del animal. Vienen caravanas reutilizables las cuales si se puede retirar de la oreja del animal cuando se necesite. Para la colocación de las mismas se utiliza una pinza como se muestra en la siguiente imagen.



*Imagen 1: Colocación de caravanas*

Internamente contienen un chip de silicio (donde tiene grabada la información), un capacitor y una bobina como indica la Imagen 2.

La información son dígitos que corresponden al código país (982, en el caso de Argentina y 999 para muestra), y los siguientes al código de identificación animal (054000000356).



*Imagen 2: Caravana Electrónica*

Estas caravanas cumplen con las siguientes normas:

- La norma ISO 11784 (Estructura de código) especifica la información que debe ser contenida en la caravana.
- La norma ISO 11785 (Conceptos Técnicos) define cómo se deben activar las caravanas y cómo la información almacenada en los mismos debe ser transferida a los lectores.

Por normativa la frecuencia a la cual se comunica el lector y las caravanas es a 134,2 KHz. Otro dato a tener en cuenta que el tiempo de vida de estos dispositivos es entre 10 a 12 años aproximadamente.

Existen dos grupos de sistemas de identificación electrónica de animales por radiofrecuencia:

- Half duplex (HDX) o de media duplicidad
- Full duplex (FDX) o de completa duplicidad.

### 2.2.1 Half Duplex

En este tipo de tecnología la caravana inicia la señal de respuesta al lector solo después recibir la señal enviada por el lector y solo después de que el condensador de almacenamiento se haya cargado completamente. La señal de respuesta solo se envía una vez, ya que la caravana tiene vació su condensador de almacenamiento después de haber enviado todo su contenido de datos.



*Imagen 3: Caravana Electrónica HDX*

### 2.2.2 Full Duplex

A diferencia de la anterior tecnología mencionada cuando el lector envía la señal de activación, el retorno de información por parte de la caravana es inmediato. No tiene que almacenar energía para poder enviar el dato que presenta en su chip.



*\*Imagen 4: Caravana Electrónica FDX*

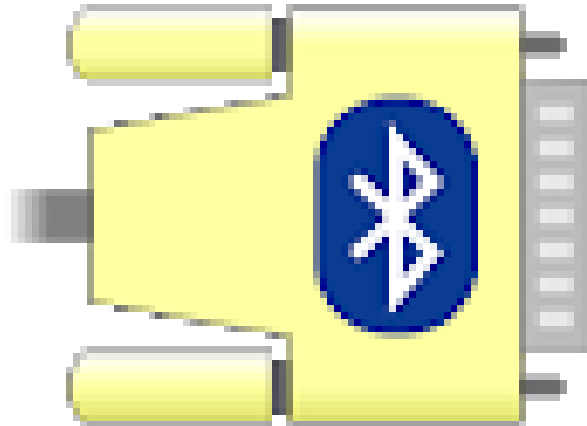
### **2.3 Balanza Electrónica**

La comunicación del lector con la balanza electrónica se realiza mediante bluetooth, utilice un módulo HC-05, al cual ingreso a sus comandos AT para configurar su nombre, contraseña y lo más importante la configuración de trabajo como esclavo o maestro. En mi caso va a trabajar como esclavo, lo que quiero decir con esto es que otro dispositivo (balanza electrónica) se va a comunicar para enviar la información del peso del animal. El dispositivo será descrito más detalladamente en el bloque correspondiente a Bastón Lector.

La balanza electrónica es simulada mediante una aplicación descargada al celular de *Play Store*, la aplicación se llama *Serial Bluetooth Terminal 1.28* (Imagen 5).

Todas las funciones de esta aplicación son gratuitas, se especifica en la información del producto.

La aplicación presenta dos tipos de lectura, bluetooth classic y Low Energy (LE, de baja energía). Como el módulo HC-05 presenta una versión 4.0 se debe seleccionar el tipo bluetooth LE para la comunicación entre ambos dispositivos.



*Imagen 5: Logo de la aplicación*

Para un fácil manejo de envío de datos se configuró ciertas teclas de la aplicación con valores de pesos (en kilogramos) ya definidos (M1, M2, M3,..., M6). Simulando así una balanza electrónica.

Se debe verificar que en el celular esta activada la opción de bluetooth, de todas formas la aplicación les indicara que deben activarlo.

En la siguiente imagen se puede ver los distintos botones con sus correspondientes pesos guardados, cuando se presiona uno de ellos automáticamente envía al dispositivo vinculado, la vinculación se da en la imagen 6 (segunda imagen, ubicada a la derecha) donde se seleccionó la opción "BLUETOOTH LE" y al presionar "SCAN" busca el dispositivo, se ingresa la clave (si la tiene) y queda establecida la vinculación.

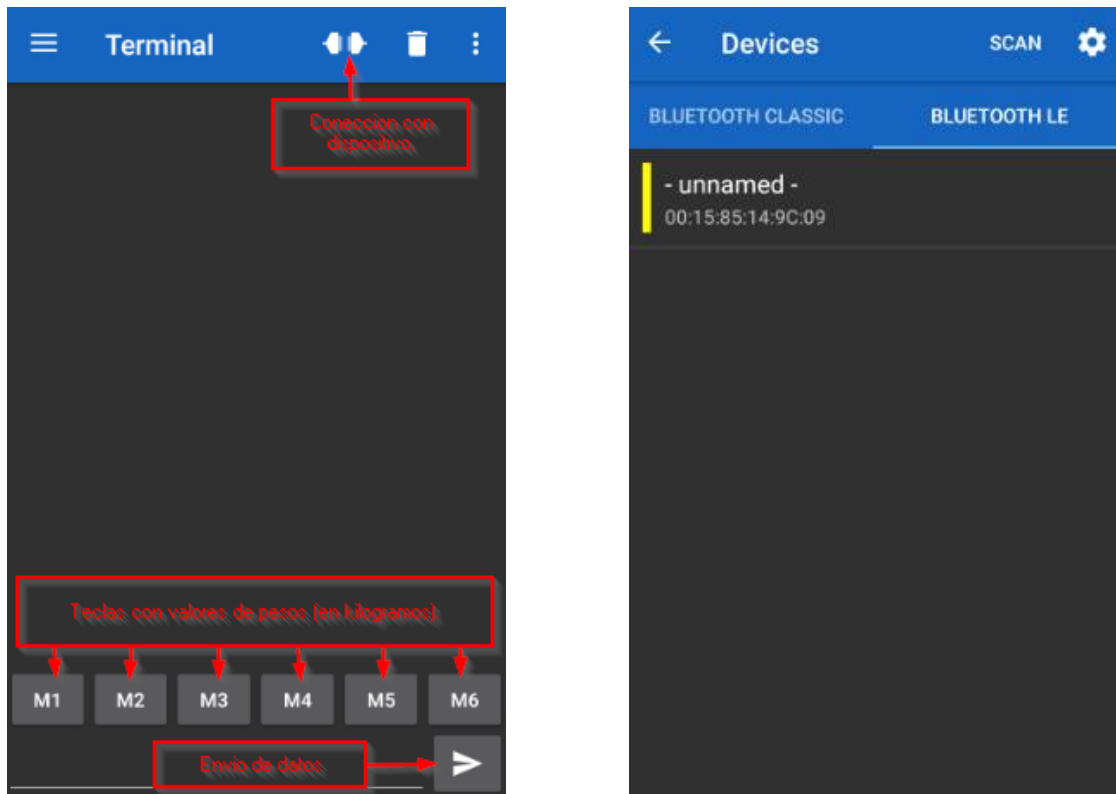


Imagen 6: Serial Bluetooth Terminal 1.28

2.4 Bastón Lector

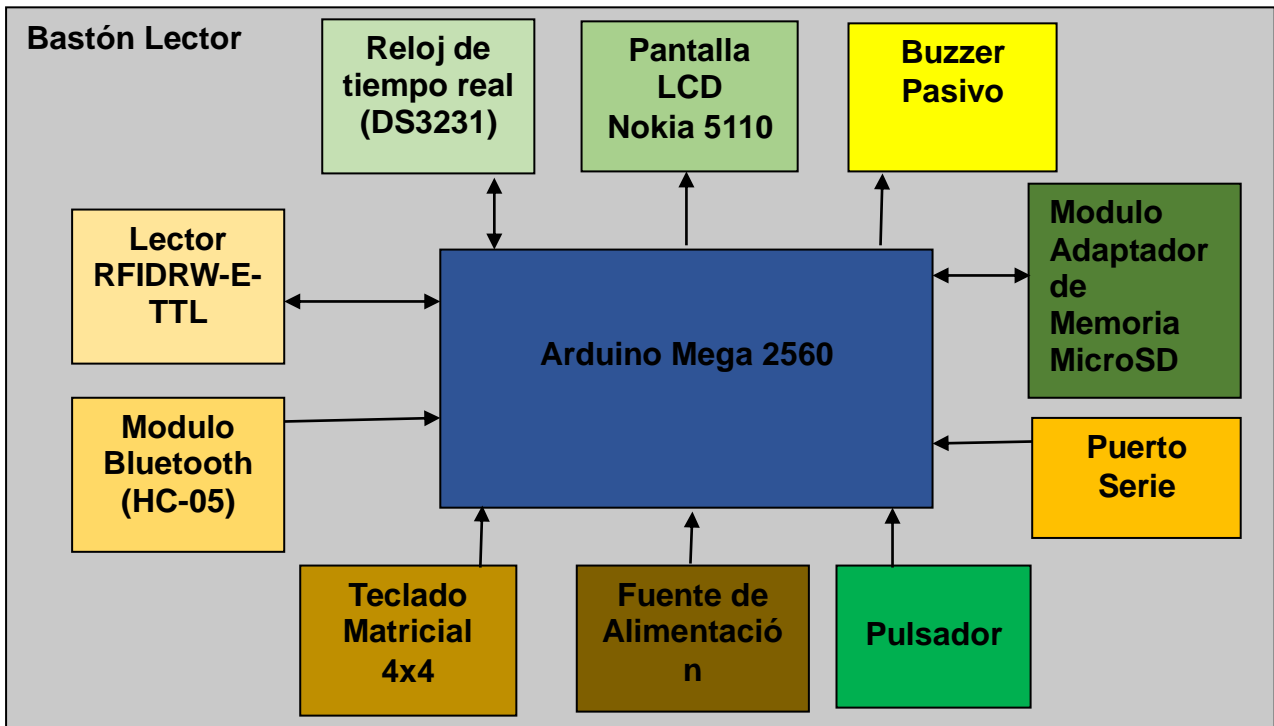


Diagrama 2: Módulos del Bastón Lector



El diagrama anterior nos muestra como está compuesto el bastón lector internamente, cada bloque corresponde a un módulo que se conecta a la placa arduino (bloque central), que es la unidad de control del dispositivo, las flechas indican el sentido de comunicación entre ellos.

El bloque central es el encargado de todas las operaciones de control. Elegí para ello la placa arduino, que contiene el chip microcontrolador ATMEL ATmega2560, debido a la cantidad de puertos series que presenta para la comunicación con los dispositivos periféricos conectados a él. Trabajé con tres de los cuatros puertos series que dispone, uno de ellos el puerto serie 1 se utiliza para conectarse al ordenador y mediante al software del bastón lector descargar los datos almacenados en la memoria microSD a la computadora. Los siguientes puertos series son utilizados para el módulo RFID (puerto serie 2) y el módulo bluetooth HC-05 (puerto serie 3).

Además la placa arduino mega 2560 contiene un gran número de entradas digitales y analógicas, necesarias para la conexión de los distintos dispositivos usados en este proyecto y como estas entradas operan con 5 volts lógicos al igual que la mayoría de las partes conectadas a él resulta muy conveniente. También presenta una fuente de 5 volts la cual se utiliza para alimentar estos periféricos acoplados a la placa, la tensión de alimentación de la mayoría de estos periféricos corresponde a la tensión mencionada.

Esta unidad de control presenta su propio entorno de desarrollo integrado (IDE), es una aplicación multiplataforma para Windows, Linux, etc. El lenguaje de programación usado es C++.

En las siguientes páginas se detallara cada bloque correspondiente al bastón lector, componentes utilizados, formas de conexión, pruebas realizadas y características propias de cada componente, y por supuesto que llevaron a la elección de los mismos para la construcción del proyecto.

### **2.4.1 Modulo Bluetooth HC-05**

Este módulo tiene como objetivo la comunicación con la balanza electrónica (simulada a través de la aplicación Serial Bluetooth Terminal 1.28), recibe por parte de la balanza un dato correspondiente al peso del animal.

Este dispositivo es un módulo serie bluetooth (Maestro / Esclavo) tiene las siguientes características técnicas obtenidas de su hoja de datos importantes a tener en cuenta para la conexión con la placa arduino.

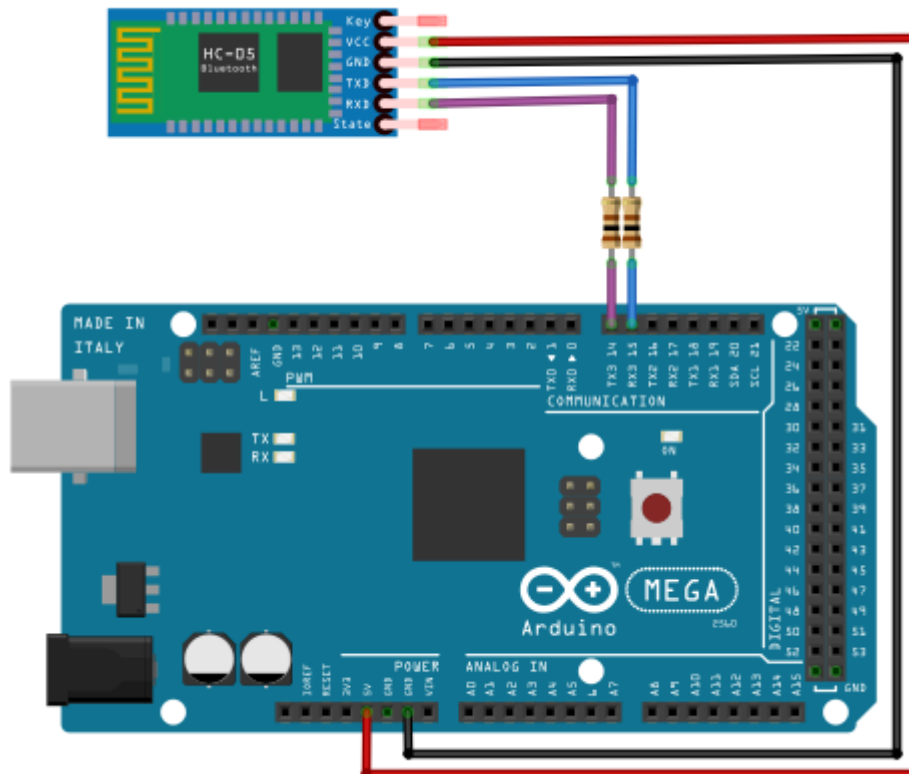
Características de Hardware:

- Compatible con arduino
- Voltaje de alimentación: 3.6 V a 6 V
- Funcionamiento de bajo consumo
- Antena PCB integrada
- Interfaz UART con velocidad de modulación en baudios programable
- Alcance de 5 a 10 metros.
- Niveles lógicos de 5 volts.
- Potencia de emisión:  $\leq 4$  dBm, Clase 2
- Consumo de corriente: 50 mA
- Frecuencia: 2.4 GHz, banda ISM
- Dimensiones totales: 1.7 cm x 4 cm aprox.

Características de Software:

- Velocidad en baudios (Modo comandos AT):38400, Bits de datos: 8, Bit de parada: 1, Paridad: Sin paridad.
- Tasa de velocidad de modulación en baudios soportadas: 9600, 19200, 38400, 57600,115200, 230400, 460800.
- Auto-conexión del dispositivo con la última configuración por defecto.
- Reconexión automática en 30 minutos cuando se desconecta a causa de pérdida de conexión por salir del rango de alcance.
- Seguridad: Autenticación y encriptación (contraseña por defecto: 1234)

La conexión del dispositivo a la placa arduino se muestra en la imagen 7, conectando el pin de Tx del HC-05 al pin Rx del arduino y Rx del HC-05 al pin Tx del arduino, en serie una resistencia de 100 ohm como protección por eventuales caídas de tensión que pueden dañar los puertos serie del arduino o del módulo. Cabe señalar que por el pin Tx se realiza el envío de información y por el pin Rx la recepción. Los pines Key y State no es necesario su conexión.



*Imagen 7: Conexión HC-05 a Placa arduino*

Una vez aclarado lo del párrafo anterior sobre Tx y Rx, paso a explicar la configuración del HC-05 que se realiza mediante el IDE de arduino configurando los comandos AT mediante el puerto serie 2 del Mega 2560.

La lista de comandos se aprecia en la siguiente imagen junto con los posibles errores que puede devolver el HC-05 cuando se está estableciendo su condición de trabajo:

## AT COMMAND LISTING

	COMMAND	FUNCTION
1	AT	Test UART Connection
2	AT+RESET	Reset Device
3	AT+VERSION	Query firmware version
4	AT+ORGL	Restore settings to Factory Defaults
5	AT+ADDR	Query Device Bluetooth Address
6	AT+NAME	Query/Set Device Name
7	AT+RNAME	Query Remote Bluetooth Device's Name
8	AT+ROLE	Query/Set Device Role
9	AT+CLASS	Query/Set Class of Device CoD
10	AT+IAC	Query/Set Inquire Access Code
11	AT+INQM	Query/Set Inquire Access Mode
12	AT+PSWD	Query/Set Pairing Passkey
13	AT+UART	Query/Set UART parameter
14	AT+CMODE	Query/Set Connection Mode
15	AT+BIND	Query/Set Binding Bluetooth Address
16	AT+POLAR	Query/Set LED Output Polarity
17	AT+PIO	Set/Reset a User I/O pin
18	AT+MPIO	Set/Reset multiple User I/O pin
19	AT+MPIO?	Query User I/O pin
20	AT+IPSCAN	Query/Set Scanning Parameters
21	AT+SNIFF	Query/Set SNIFF Energy Savings Parameters
22	AT+SENM	Query/Set Security & Encryption Modes
23	AT+RMSAD	Delete Authenticated Device from List
24	AT+FSAD	Find Device from Authenticated Device List
25	AT+ADCN	Query Total Number of Device from Authenticated Device List
26	AT+MRAD	Query Most Recently Used Authenticated Device
27	AT+STATE	Query Current Status of the Device
28	AT+INIT	Initialize SPP Profile
29	AT+INQ	Query Nearby Discoverable Devices
30	AT+INQC	Cancel Search for Discoverable Devices
31	AT+PAIR	Device Pairing
32	AT+LINK	Connect to a Remote Device
33	AT+DISC	Disconnect from a Remote Device
34	AT+ENSNIFF	Enter Energy Saving mode
35	AT+EXSNIFF	Exit Energy Saving mode

## ERROR CODES

ERROR CODE	VERBOSE
0	Command Error/Invalid Command
1	Results in default value
2	PSKEY write error
3	Device name is too long (>32 characters)
4	No device name specified (0 length)
5	Bluetooth address NAP is too long
6	Bluetooth address UAP is too long
7	Bluetooth address LAP is too long
8	PIO map not specified (0 length)
9	Invalid PIO port Number entered
A	Device Class not specified (0 length)
B	Device Class too long
C	Inquire Access Code not Specified (0 length)
D	Inquire Access Code too long
E	Invalid Inquire Access Code entered
F	Pairing Password not specified (0 length)
10	Pairing Password too long (> 16 characters)
11	Invalid Role entered
12	Invalid Baud Rate entered
13	Invalid Stop Bit entered
14	Invalid Parity Bit entered
15	No device in the Pairing List
16	SPP not initialized
17	SPP already initialized
18	Invalid Inquiry Mode
19	Inquiry Timeout occurred
1A	Invalid/zero length address entered
1B	Invalid Security Mode entered
1C	Invalid Encryption Mode entered

*Imagen 8: Comandos AT*

Abrimos el monitor serie del IDE (Imagen 9) y escribimos los comandos que aparecen en la imagen 8 los que utilizo para la configuración son los siguientes y la forma que se escriben para la configuración:

- AT+ROLE x (x=0, modo esclavo; x=1 modo maestro).
- AT+NAME x (x, nombre que desea para el modulo).
- AT+PSWD x (en x escribe la contraseña,"1234" es por defecto).
- AT+BAUD x (x=6 corresponde a 9600 baudios que es la velocidad de comunicación que utilizo y no aparece en la lista de la imagen anterior).

Este dispositivo, para detectar un comando, tiene que leer el "final de línea" y el "retorno de carro" ("\r\n"), por tanto el Serial Monitor del IDE tiene que estar en esa opción.

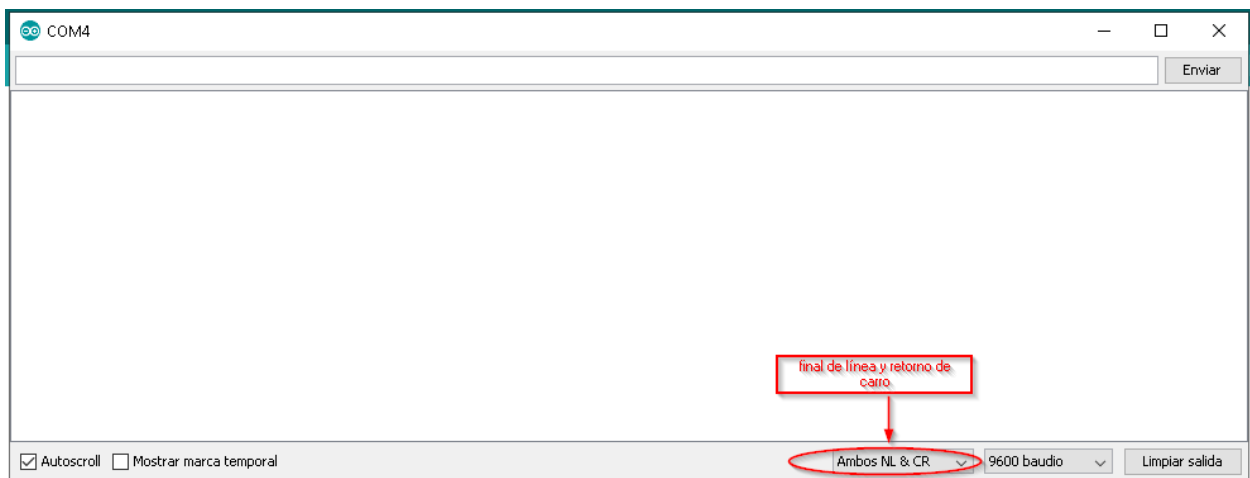


Imagen 9: Serial Monitor

El HC-05 presenta un led rojo en su parte inferior derecha el cual indica de forma visual si uno se encuentra en modo de configuración (parpadeo rápido y continuo) o no (parpadeo lento y continuo). Para salir de los comandos AT se puede realizar de dos maneras, una es enviando el comando AT+RESET lo cual resetea el modulo y queda guardado lo que se configuro, la otra manera es desconectando el cable de VCC y volviéndolo a conectar. Probé de las dos maneras y funciona.

Una vez realizado todos los pasos anteriores, el modulo está listo para conectarse a la balanza electrónica o en mi caso al software simulador de la misma. Al estar en modo esclavo solo recibe información, desde el programa *Serial Bluetooth Terminal 1.28*, como se explicó en el capítulo anterior ingreso al HC-05 mediante su código y si se produjo la vinculación de ambos dispositivos, por pantalla se informa y también mediante el led rojo el cual permanece encendido de forma constante.

### 2.4.2 Teclado Matricial 4x4

Para moverse en el menú desarrollado y que se visualiza en la pantalla Nokia 5110 , la cual se desarrollara más adelante en el informe, utilice un teclado matricial del cual solo se ocupan 4 pulsadores de los 16 que proporciona, dejando los demás para futuras mejoras en el código.

En la pantalla se indica que tecla presionar de acuerdo a donde quiera ingresar en el menú. Se utilizó la librería "Keypad.h" (open source) correspondiente al teclado para el desarrollo del programa.

Algunas de sus características principales extraídas de su hoja de datos:

- Diseño ultrafino
- Respaldo adhesivo
- Interfaz fácil para cualquier microcontrolador.
- Excelente relación precio / rendimiento
- Clasificación máxima: 24 V CC, 30 mA
- Interfaz: acceso de 8 pines a la matriz 4x4



*Imagen 10: Teclado Matricial 4x4*

En el código arduino se declaran las filas (4) y columnas (4), con esto quiero decir que se debe declarar los pines de filas y columnas que se conectan a la placa, en una variable del tipo “string” (matriz de datos de una sola fila).

### **2.4.3 Pantalla LCD Nokia 5110**

Elegí esta pantalla para mi diseño debido a su bajo costo comparado con otras pantallas (táctiles, a color, etc.) como también la cantidad de datos que se pueden visualizar en ella su fácil programación debido a que presenta librerías de libre uso (open source) para su programación, tal es el caso de “Adafruit\_GFX.h” para realizar gráficas y “Adafruit\_PCD8544.h” para el controlador PCD8544 LCD CMOS, diseñado para manejar una pantalla gráfica de 48 filas y 84 columnas, y algo muy importante es su bajo consumo de energía.

La biblioteca nos indicará a que pin de la placa arduino debemos conectar los pines de la pantalla, y como podemos configurarlos.

Especificando un poco más sus características, es una pantalla monocromática de 84x48 píxeles individuales. Son reducidas en tamaño de alrededor de 1,5 pulgadas, vienen con

una luz de fondo incorporada (Backlight) la cual se puede ir variando su luminosidad, en el proyecto se colocó un potenciómetro para su variación.

El protocolo de control para este display es por medio de SPI (Serial Peripheral Interface), lo cual hace más fácil su comunicación entre este y la placa arduino, en el sketch del IDE se debe incorporar la librería "SPI.h" para hacer posible la conexión y traspaso de información entre ambos.

El bus SPI (Interfaz periférica serie) presenta una arquitectura del tipo maestro-esclavo, en este caso el maestro es el microcontrolador ATmega 2560 y el display es el esclavo, es un bus síncrono, arduino proporciona una señal de reloj para mantener la sincronización entre ellos. Tiene una alta velocidad de transmisión.



*Imagen 11: Display Nokia 5110*

A continuación se detallan cada pin de la placa extraídos de su hoja de datos:

- Vcc (1): Alimentación 5V.
- Gnd (2): Ground/ masa.
- CE (3): Chip enable, el pin de habilitación el cual permite la entrada de datos. La señal activa es LOW, se aplica en estado bajo.
- RST (4): Reset, esta señal restablecerá el dispositivo y deberá aplicarse para inicializar correctamente el chip. La señal activa es LOW, se aplica en estado bajo.



- D/C (5): Data/Comand, entrada para seleccionar comandos o entrada de datos.
- Din (6): SERIAL DATA LINE, Entrada para datos en línea.
- Clk (7): Entrada de reloj.
- BL (8): Luz de fondo (Backlight) de 2.7V-3.2V.

El esquema de montaje es muy sencillo y se observa en la imagen, se alimenta el display a 5 volts, a masa, y se coloca un potenciómetro de 5 kohm al pin 8, los demás pines van conectados de acuerdo como se configuro en el programa y pueden llevar o no una resistencia de 1 kohm en serie, ya que algunos modelos no son tolerables a tensiones de 5 volts en las líneas de datos. En mi caso coloque esas resistencias como protección.

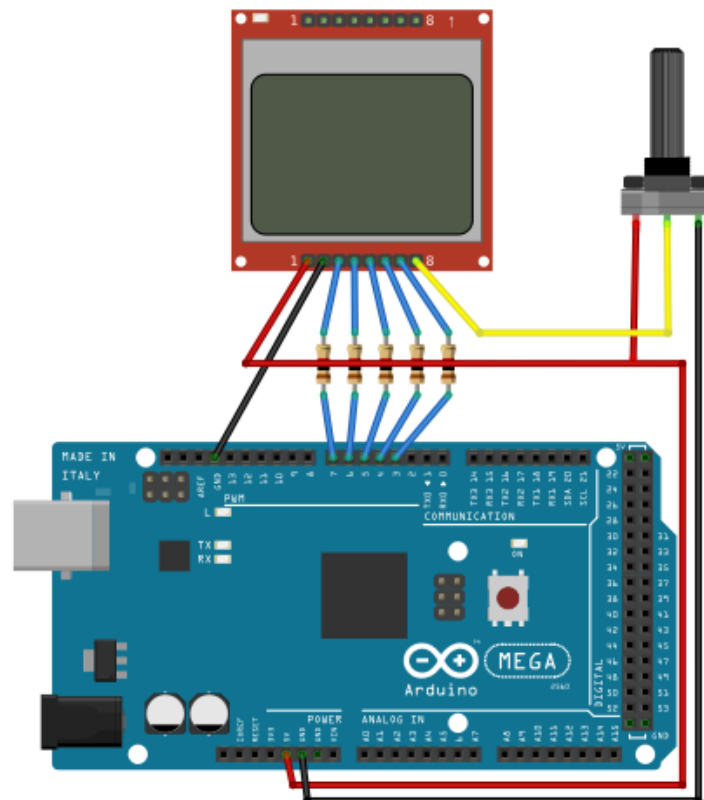


Imagen 12: Conexión Display Nokia 5110

#### 2.4.4 Buzzer Pasivo

El buzzer pasivo es un dispositivo electrónico que convierte una señal eléctrica (5 volts) en una onda de sonido, estos son muy distintos a lo buzzer activos que presentan en su arquitectura un oscilador interno que puede resonar a la frecuencia que deseamos. El

buzzer utilizado es el indicado en la imagen 13 y tiene como finalidad en el bastón lector informar la lectura correcta de caravanas electrónicas o no, la forma de identificar si es correcta o no la lectura es a través de su pitido siendo corto en la primera y prolongado en la segunda. Presenta 3 pines uno a VCC (5 v) otro a GND y el último es la señal de 5 volts que activa el sonido (I/O). El dispositivo es muy económico debido a ello opte por él.



*Imagen 13: Buzzer pasivo*

#### **2.4.5 Pulsador**

El circuito de habilitación de lectura de caravana está dado por un pulsador con su correspondiente circuito antirrebote, es abierto en reposo de tecla roja y soporta una potencia de 0,5A /250V. La habilitación es por bajo, significa que la entrada al pin de la placa arduino siempre va a estar alto (5 volts) y cuando se presiona el dispositivo toda la corriente va a masa, en el pin de la placa va a tomar como un cero lógico.

Al referirme al antirrebote estoy diciendo que es un circuito que se adiciona al pulsador para evitar fluctuaciones en la tensión que el arduino puede interpretar como un pulso en alto o en bajo, en otras palabras interpreta que fue pulsado varias veces.

Este circuito consiste en una resistencia conectada al pin del pulsador por el cual circulan 5 volts que a su vez van a la entrada digital del pin del arduino. Entre las patas del pulsador se coloca en paralelo un condensador y un diodo de conmutación en inversa y





*Imagen 15: Modulo RTC*

La comunicación entre el dispositivo y la placa arduino se realiza a través del bus I2C (Inter-Integrated Circuit), en los pines de transferencia de datos 20(SDA) y el pin de sincronización o señal de reloj 21(SCL) como se indica en la imagen 15.

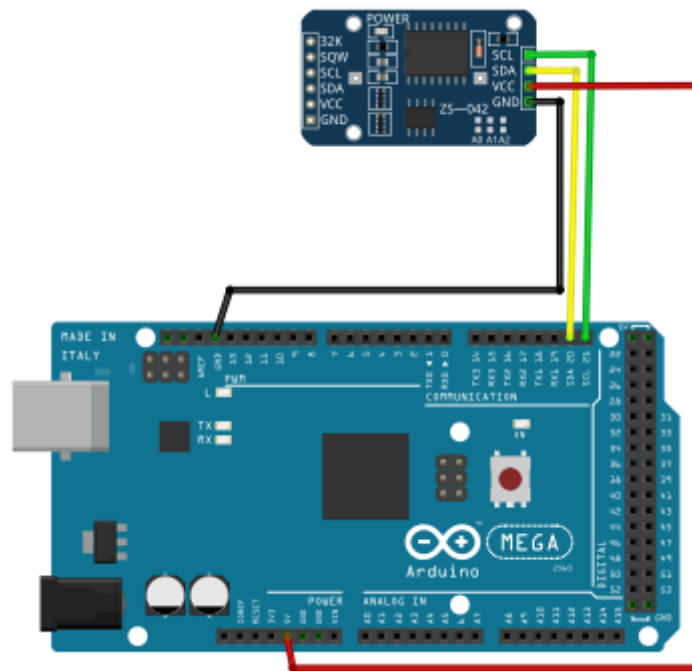
Al igual que el bus SPI el bus I2C tiene una arquitectura de tipo maestro-esclavo, pero este tipo de comunicación solo dispone de dos cables y es síncrono. El dispositivo maestro inicia la comunicación con el o los esclavos, y puede mandar o recibir datos de los esclavos. Pero el esclavo, en este caso el DS3231, no puede establecer comunicación con el arduino.

Es recomendable utilizar resistencias de Pull-UP de las líneas (SDA y SCL) a 5 volts, cuando se conectan más dispositivos al bus pero como conecto solo el RTC no es necesario.

Al realizar el programa en el IDE se debe incluir la librería "Sodaq\_DS3231.h" propia del módulo y de libre uso. Mediante las funciones de dicha librería se configura el año, mes, fecha, día, hora, minuto y segundo. Una vez configurados estos valores quedan guardados y no se modifican si se retira la alimentación Vcc.

La fecha al final del mes se ajusta automáticamente para meses con menos de 31 días, incluidas las correcciones por año bisiesto. El reloj funciona en formato de 24 horas o de 12 horas con un indicador AM / PM, fue configurado para trabajar en el formato de 24 horas.

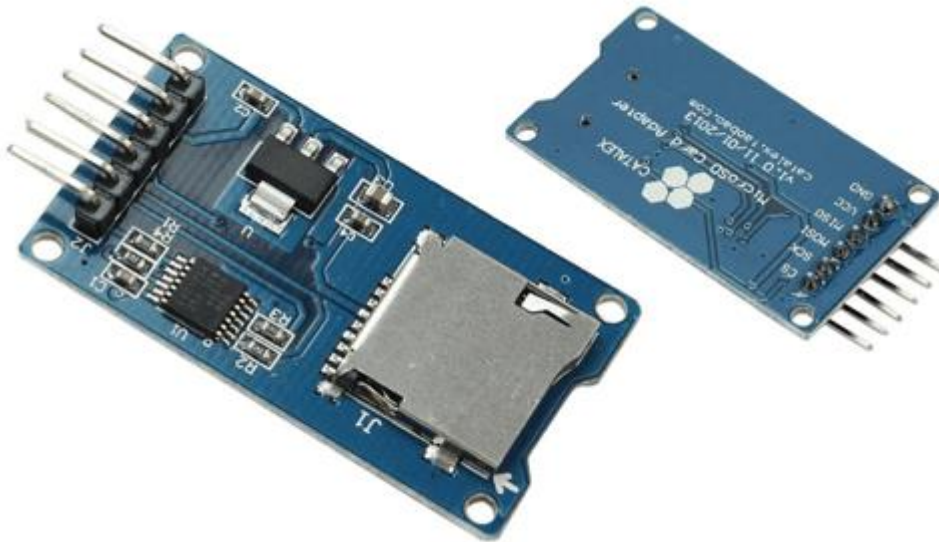
Dentro del menú del bastón lector se puede acceder a la información que proporciona el RTC, será detallado en el capítulo 2.2.



*Imagen 16: Conexión RTC*

### 2.4.7 Modulo adaptador de memoria microSD

Toda la información recaudada en la lectura de caravana de los distintos corrales leídos, es almacenada en una memoria microSD (Standard Capacity) con una capacidad de almacenamiento de 4 gigabytes, puede almacenar los datos de más de 4 mil animales. La tarjeta de memoria esta insertada en un módulo adaptador, quien presenta un circuito de conversión de nivel que permite interactuar con niveles TTL de tensión (de 0 a 5 volts). La interfaz de comunicación es una interfaz SPI al igual que anteriores dispositivos mencionados anteriormente y trabaja con la misma librería en el IDE ("SPI.h"). Esta librería ya viene incluida en el Sketch del IDE por lo que no es necesario descargarla.

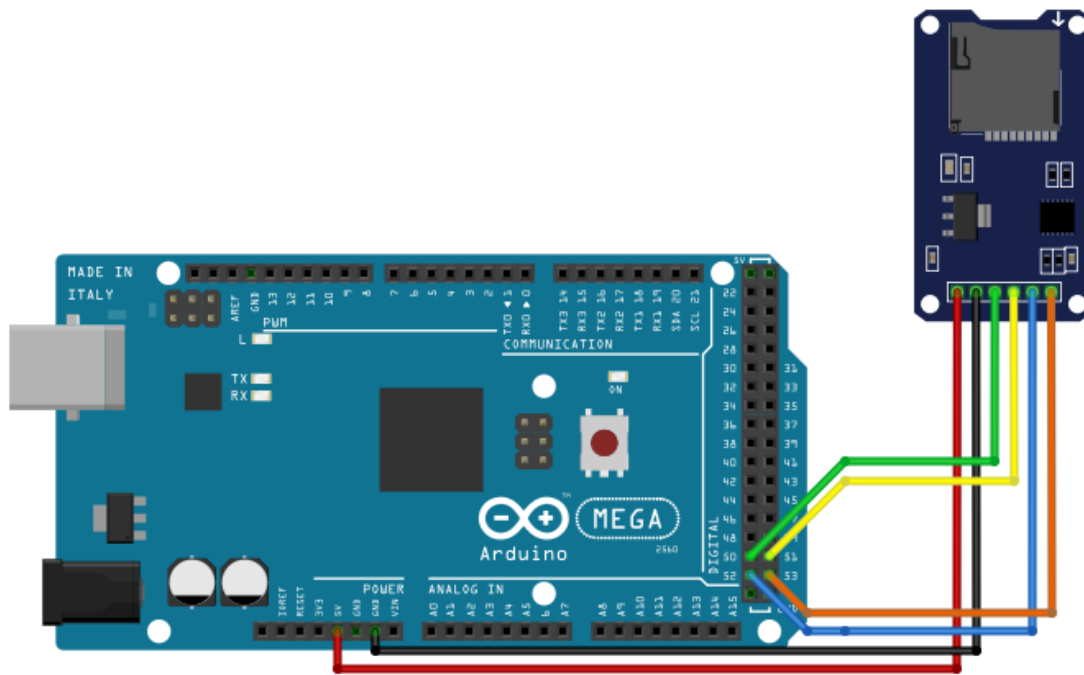


*Imagen 17: Modulo MicroSD*

El aparato presenta un total de 6 pines en su interfaz de control:

- VCC: Fuente de alimentación, a 5 volts.
- GND: tierra.
- MISO: (Master Input Slave Output) Es la señal de entrada a nuestro dispositivo, por aquí se reciben los datos desde el arduino. Corresponde al pin 50 del ATmega 2560 (cable verde).
- MOSI: (Master Output Slave Input) Transmisión de datos hacia el arduino. Corresponde al pin 51 del ATmega 2560(cable amarillo).
- SCK: Señal de reloj del bus. Esta señal rige la velocidad a la que se transmite cada bit. Corresponde al pin 52 del ATmega 2560(cable azul).
- CS: Chip Select o Slave Select, habilita el integrado hacia el que se envían los datos. Corresponde al pin 53 del ATmega 2560(cable anaranjado).

Las conexiones al módulo varían de acuerdo al modelo de placa arduino que se está utilizando para mi caso como trabajo con el ATmega 2560 las conexiones se indican a continuación en la imagen.



*Imagen 18: Conexión Módulo MicroSD*

En la tarjeta microSD serán guardados en formato XML los datos tomados de cada animal, contendrá nombre del corral, número de caravana, peso, fecha, hora y estado. El menú del bastón lector tendrá una opción para descargar estos datos a la computadora una vez conectados entre ellos y cuando se haya descargado toda la información los datos serán eliminados de la misma. Esto que menciono será explicado con detalles más adelante en el informe en la parte de explicación de código en el Sketch del IDE de arduino.

#### **2.4.8 Lector RFIDRW-E-TTL**

En este apartado explicare el dispositivo vital del proyecto el cual es un módulo lector de RFID (Radio Frequency Identification) o identificación por radiofrecuencia. La función principal de este tipo de tecnología es transmitir un número de serie único (ID) mediante ondas de radio.

El dispositivo es capaz de leer los dos sistemas de identificación electrónica de animales por radiofrecuencia (Full Duplex y Half Duplex) protocolos descritos por las normas ISO 11784 e ISO 11785. Cuenta con salidas y entradas de puertos series en niveles TTL (0-5

volts y de 0-3,3volts) lo cual hace una conexión directa con la placa arduino sin necesidad de conectar entre ambos un regulador de tensión.

Lee y escribe una gran variedad de tarjetas RFID como lo son:

- EM4100
- Serie Atmel T55xx (Lectura/Escritura)
- TIRIS de 64 bits
- EM4205 / EM4305
- Lectura y escritura de caravanas electrónicas de protocolos Full Duplex y Half Duplex

Esta son algunas de la gran variedad de tarjetas RFID que puede leer y escribir, extraídas de su hoja de datos.

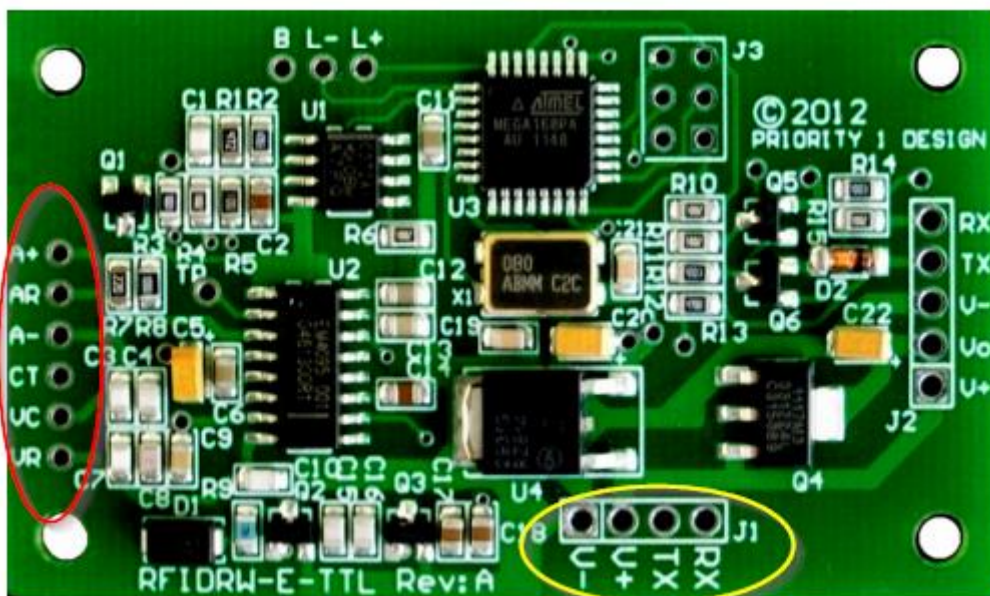


Imagen 19: Modulo RFID

En la imagen 19 se aprecia el dispositivo lector de caravanas electrónicas, presenta una gran variedad de entradas y salidas. De todas los pines de entrada/salida solo usare los que están remarcados con un círculo de color rojo y amarillo.

El círculo amarillo corresponde a "J1" y por ellos se alimenta al módulo y se realiza la comunicación serie con la unidad de control. El círculo rojo es la conexión con la antena emisora/receptora.



J1:

- V+: tensión de alimentación (5V).
- V-: Tierra.
- RX: Recepción de datos.
- TX: Transmisión de datos.

La antena del módulo es un inductor simple en forma de bobina que ya viene incluida, fue adquirido en Melbourne, Australia ya que en nuestro país no se fabrica, el costo del producto y envío será detallado más adelante en el capítulo 4.

La función de la bobina acoplada a un condensador (internamente), esto se llama circuito sintonizado, es de producir una frecuencia de resonancia. El inductor es un RFIDCOIL-49A estándar (Imagen 21) y la forma de conectarlo (círculo rojo de la imagen 19) es la que se muestra a continuación.

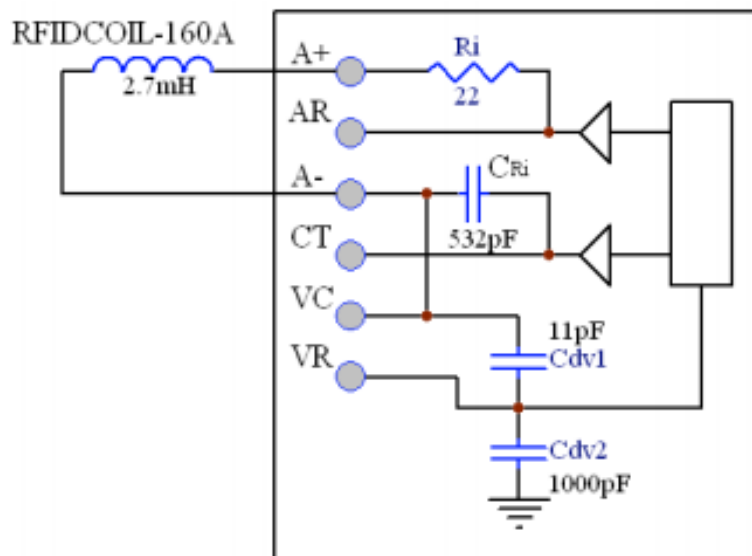


Imagen 20: Conexión antena RFID

Esta es la conexión para trabajar con la frecuencia de resonancia de 134,2 KHz provista por las normas ISO para la lectura de chip de animales. En su hoja de datos especifica el agregado de un condensador entre los pines CT y VC en caso que necesite trabajar con otro tipo de frecuencia, y aplicando la fórmula de resonancia puede encontrar su valor, no entrare más en detalles debido a que el proyecto solo necesita trabajar a la frecuencia

citada. También cabe mencionar que presenta salida para buzzer (B) y led (L-, L+) que indican la lectura de caravanas.

El principio de funcionamiento del dispositivo RFIDRW-E-TTL es que genera un campo magnético a través de la bobina externa, de 134kHz, este campo cuando está dentro del rango de lectura de la caravana, entra en resonancia con lo que alimenta su electrónica interna. Una vez alimentada la caravana, puede modular el campo magnético incidente que es detectado por el lector y de esta manera, los transpondedores pueden transmitir sus datos al lector.

Para poder realizar lo explicado anteriormente el lector necesita de una habilitación por parte de la placa arduino, estos se comunican por puerto serie (puerto serie 2 del ATmega 2560), el arduino envía el comando `"RAT <crn>"`, <crn> corresponde a un retorno de carry, por el puerto serie (Rx). El modulo lector reconoce este comando y si está dentro del rango del transpondedor lee sus datos, la respuesta es una trama que contiene los valores de identificación de la caravana con un retorno de carry.

En caso de ocurrir algún tipo de error en la lectura informa a la placa arduino enviando una trama con los valores correspondiente al error y por supuesto al final de la trama el `"<crn>"`. En la información proporcionada por el fabricante especifica 5 tipos de errores como los es `"?1<crn>"` que hace referencia a que no se encontró ningún dispositivos dentro del rango de lectura, estos errores son tomados en cuenta en la realización del código y cuando se detectan se informa por pantalla que se vuelva a presionar para una nueva toma de datos. En la practica el rango de lectura de las tarjetas es de aproximadamente 5 centímetros de distancia.

Se conecta el pin RX del lector al pin TX del arduino (TX3, pin 16) y el pin TX del lector al pin RX del arduino (RX3, pin 17), al igual que al módulo bluetooth se coloca una resistencia de 100 ohm en serie en cada pin para prevenir desniveles de tensión que dañen el puerto del arduino o el del lector.



*Imagen 21:RFIDCOIL-49A*

#### **2.4.9 Fuente de alimentación**

Para la fuente de alimentación opte por una pila alcalina de 9 volts, también se puede utilizar pilas recargables de níquel-cadmio. Esta batería alimenta por separado a la placa arduino y al modulo lector RFIDRW-E-TTL, donde en su hoja de datos especifica que trabaja en el rango de tensión de entre 5,5 a 15 volts.

Se tiene que tomar en cuenta el consumo de corriente de cada módulo para luego realizar el cálculo de cantidad de horas que tendrá la batería de uso continuo antes que se agote. El consumo de corriente de cada módulo es extraído de sus hojas de datos y son los siguientes:

- Placa Arduino Mega 2560: 93 mA.
- MicroSD card: 80 mA.
- RTC DS3231: 300 uA.
- Modulo Bluetooth HC-05: 40 mA.
- Display Nokia 5110: 20 mA.
- Módulo RFIDRW-E-TTL: 38 mA.
- Buzzer: 30 mA.

La suma de corriente de los dispositivos usado para el proyecto bastón lector da un total de aproximadamente 300 mA. A esa corriente se la multiplica por la tensión de operación de cada módulo, la mayoría opera con 5 volts, eso nos da un valor de 1,505 Watts.

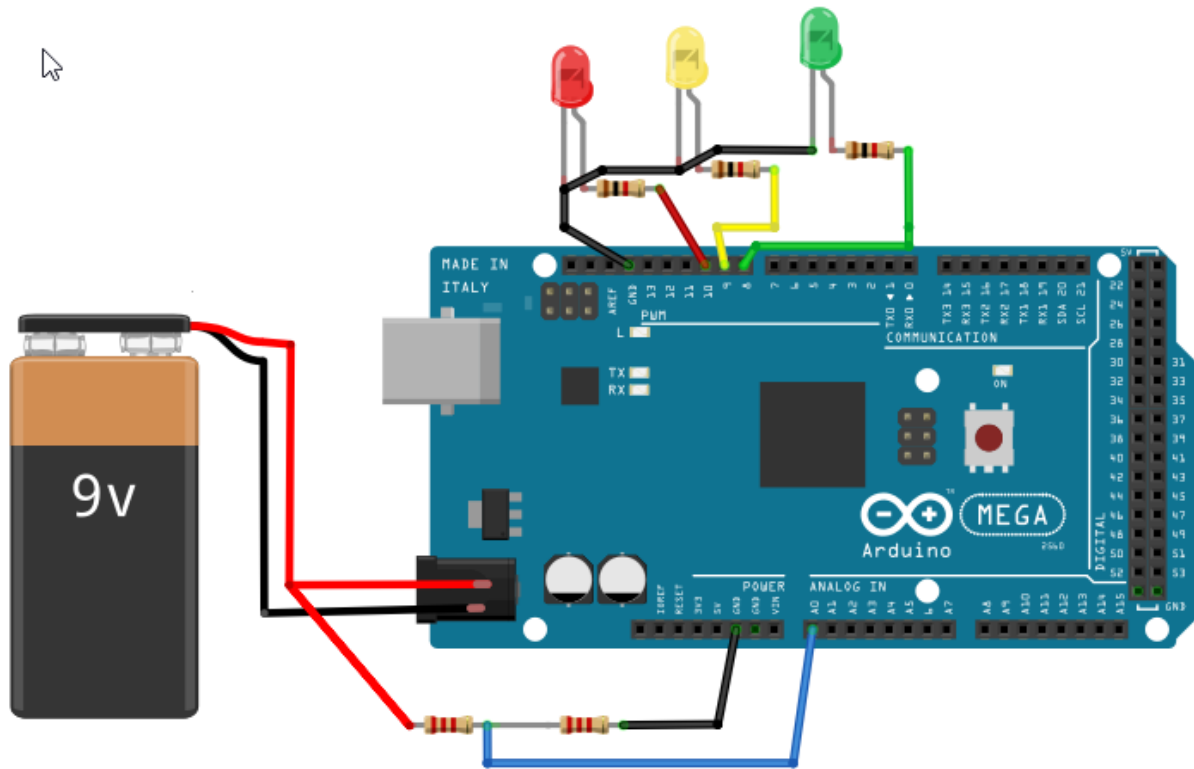
Si a la potencia obtenida se la divide por la tensión que entrega la pila nos da la corriente total que consume el instrumento completo.

Consumo de corriente = 1,505 Watts / 9 volts = 0.167 Amperes

A ese valor de corriente se lo divide por la corriente que entrega la pila alcalina (2,5 Amperes) y nos da un valor que es la autonomía de trabajo (en horas) que tendrá el bastón lector antes de agotarse por completo la carga de dicha pila, da un valor de aproximadamente 14 horas de uso continuo.

En la parte frontal del lector se colocó tres led de distintos colores para informar el estado de la pila. La lectura se realiza mediante software, para ello realice un circuito divisor resistivo de tensión que reduzca la tensión de la pila a una tensión de 5 volt apta para las entradas digitales de la placa arduino de lo contrario dañaría dichas entradas. Use resistencias de 2,2Kohm. La entrada analógica A0 es la encargada del censado de tensión de la fuente y las salidas digitales 8,9 y 10 para informar por medio visual el estado de carga de la fuente (imagen 22).

- Verde: pila cargada 9V.
- Amarillo: pila al 50 %.
- Rojo: pila descargada.



*Imagen 22: Fuente de alimentación*

Es aconsejable cambiar la pila cuando indique el led rojo de baja batería debido a que puede producir lecturas erróneas y mal funcionamiento del producto.

#### **2.4.10 Arduino ATmega 2560**

La unidad de control del bastón lector es una placa arduino Mega 2560, placa que se ha ido mostrando en los distintos conexiones de aparatos como bluetooth, RTC, etc. Arduino es una plataforma de desarrollo la cual está basada en una placa electrónica (placa de circuito impreso) de hardware libre, contiene un microcontrolador re-programable (en este caso el ATmega 2560) y una serie de pines hembra, los que permiten establecer conexiones entre el microcontrolador y los diferentes módulos que componen el dispositivo lector de caravanas electrónicas, además presenta una fuente regulada a 5 volts que permite alimentarlos (también una fuente de 3,3 volts). Presenta una gran variedad de ventajas respecto a otras placas arduino como lo son arduino uno, nano, etc. Entre ellas están la cantidad de pines digitales y analógicos, la cantidad de puertos serie que presenta (4), pines especiales para el bus I2C, entre otras características como la cantidad de memoria flash (256K) que posee para desarrollar un

firmware extenso en lenguaje C++ y por supuesto el costo razonable que tienen este tipo de placas y fácil de conseguir en cualquier casa de electrónica o por internet.

Para la programación de la placa arduino se necesita descargar el IDE (Integrated Development Environment) , este entorno de desarrollo integrado es un software con herramientas que nos permiten escribir, editar, depurar y grabar nuestro código al microcontrolador a través del puerto serie. La pantalla donde se realiza el código se llama "Sketch". El Sketch presenta un monitor serie el cual permite realizar comunicaciones entre la pc y la placa, viendo en pantalla lo que está sucediendo.

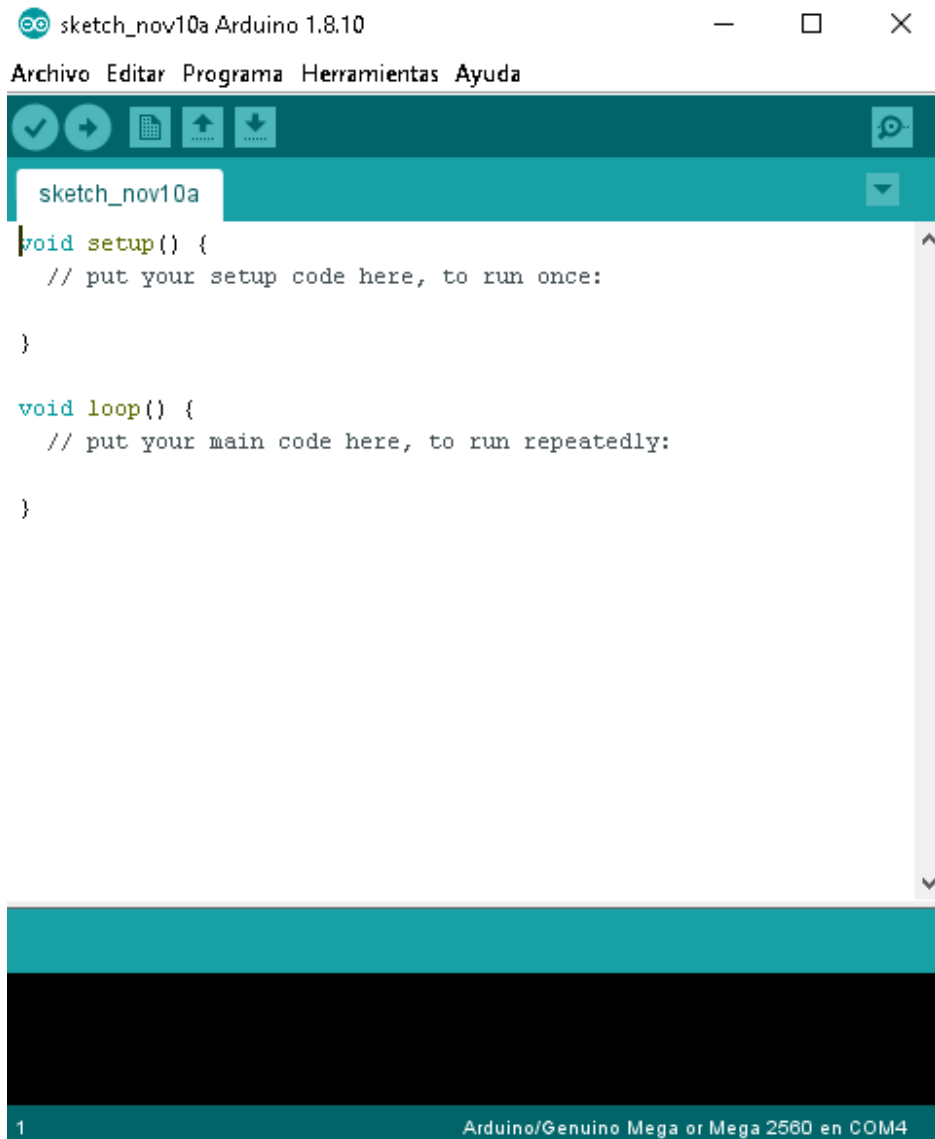
En el desarrollo del código del dispositivo se instaló una gran variedad de librerías de los distintos módulos, facilitando así la programación y comunicación con los mismos, librerías que el IDE no trae incluidas. El software se puede descargar de la página de arduino, gratuito y de libre uso ([www.arduino.cc](http://www.arduino.cc)). También existe otro entorno de programación como lo es el Atmel Studio (en distintas versiones) pero en mi caso me resulta más fácil trabajar con el IDE.

El lenguaje de programación C++. Una vez descargado e instalado el software aparecerá una pestaña como en la imagen siguiente.



*Imagen 23: Inicio IDE*

Una vez inicializado el programa tendremos el Sketch listo para empezar con el desarrollo del código.



*Imagen 24: Sketch del IDE*

En el “setup” va las inicializaciones de los distintos puertos series, es muy importante en este paso definir la misma velocidad de comunicación entre el arduino y los dispositivos ya que los caracteres transferidos no serán reconocidos correctamente., declaración de pines de entrada o salida en estado bajo (LOW) o alto (HIGH), etc.

En el “loop” se desarrolla el programa principal que va a ejecutar el microcontrolador.

El código empleado en el microcontrolador para el manejo de los distintos periféricos conectados a él, que en conjunto arman el bastón lector es explicado en el siguiente diagrama de bloque.

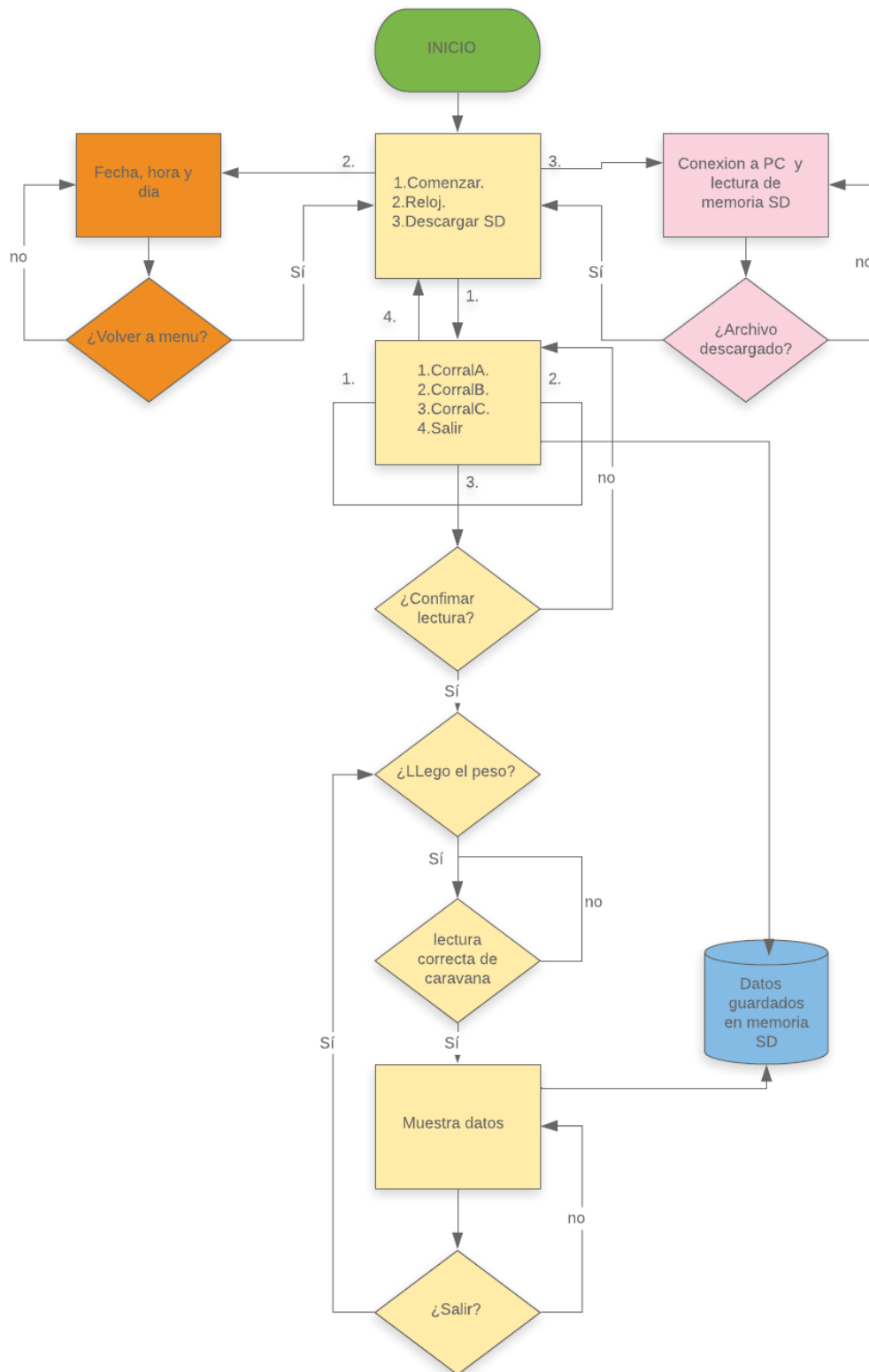


Diagrama 3: Código de lectura y almacenamiento de datos



Es importante señalar que en el almacenamiento de información en la tarjeta de memoria microSD, el archivo que los contendrá deberá ser XML (Extensible Markup Language) Lenguaje de Marcado Extensible. Este tipo de documento tiene un encabezado el cual especifica que lo que está a continuación tiene formato XML. Es un lenguaje que permite jerarquizar y estructurar la información y describir los contenidos dentro del propio documento. Al referirme con jerarquizar estoy diciendo que presenta un tipo de herencia dentro del documento, luego del encabezado sigue un nodo raíz o nodo padre donde en el arranca y finaliza los datos que queremos contener dentro de él. Luego siguen nodos hijos que pueden contener cualquier flujo de datos basados en textos.

Siempre el inicio de un nodo comienza con "<nombre>" y finaliza con "</nombre>".

La siguiente imagen muestra el documento XML guardado en la memoria microSD con la lectura de tres caravanas electrónicas, cada una en un diferente corral.

```
<?xml version="1.0" encoding="utf-8" ?>
<animales>
  <lote lote="corralA">
    <id id="000000356">
      <peso>234</peso>
      <fecha>2019/11/5</fecha>
      <info>10:57:5</info>
      <estado></estado>
    </id>
  </lote>
  <lote lote="corralB">
    <id id="000000386">
      <peso>236,6</peso>
      <fecha>2019/11/5</fecha>
      <info>10:58:5</info>
      <estado></estado>
    </id>
  </lote>
  <lote lote="corralC">
    <id id="000000379">
      <peso>234,8</peso>
      <fecha>2019/11/5</fecha>
      <info>10:58:5</info>
      <estado></estado>
    </id>
  </lote>
</animales>
```

El diagrama muestra un documento XML con las siguientes anotaciones:

- Encabezado:** Señala a la línea de declaración de XML: `<?xml version="1.0" encoding="utf-8" ?>`
- Nodo raíz o padre:** Señala al elemento raíz: `<animales>`
- Nodo Hijo:** Señala a los tres elementos de lote: `<lote lote="corralA">`, `<lote lote="corralB">` y `<lote lote="corralC">`
- Finalización Nodo raíz:** Señala al elemento de cierre del nodo raíz: `</animales>`

Imagen 25: Documento XML

Como se puede ver en el documento XML, presenta un nodo raíz “animales” y sus hijos que corresponden a los 3 corrales que se configuro. Este nodo raíz no puede estar incluido en ninguna otra parte del archivo solo al principio y final.

Dentro de cada corral contiene el nodo “id” en ese nodo se guarda el número de identificación de la caravana leída y dentro de él va almacenado sus datos como lo es el peso en kilogramos que es recibido desde la balanza electrónica, la fecha y hora que se tomó la lectura y el estado.

En el nodo estado va ser almacenado un dato dependiendo de qué tecla que indica el menú se apretó. Si presiono la tecla borrar se guarda “<estado>borrar</estado>”, significa que el animal por algún motivo abandono su corral (ventas, muerte, etc.), de lo contrario se guarda un estado vacío “<estado></estado>”.

#### **2.4.11 Puerto Serie**

Para él envío de datos entre ambos dispositivos (PC y arduino) se conecta un cable USB 2.0 del tipo A/B (plug macho tipo “A” USB y plug macho tipo “B” R3). El cable tiene un encapsulado plástico para protección de color azul transparente, su tiempo de vida útil es de 8 años aproximadamente y es de 48 cm de largo.



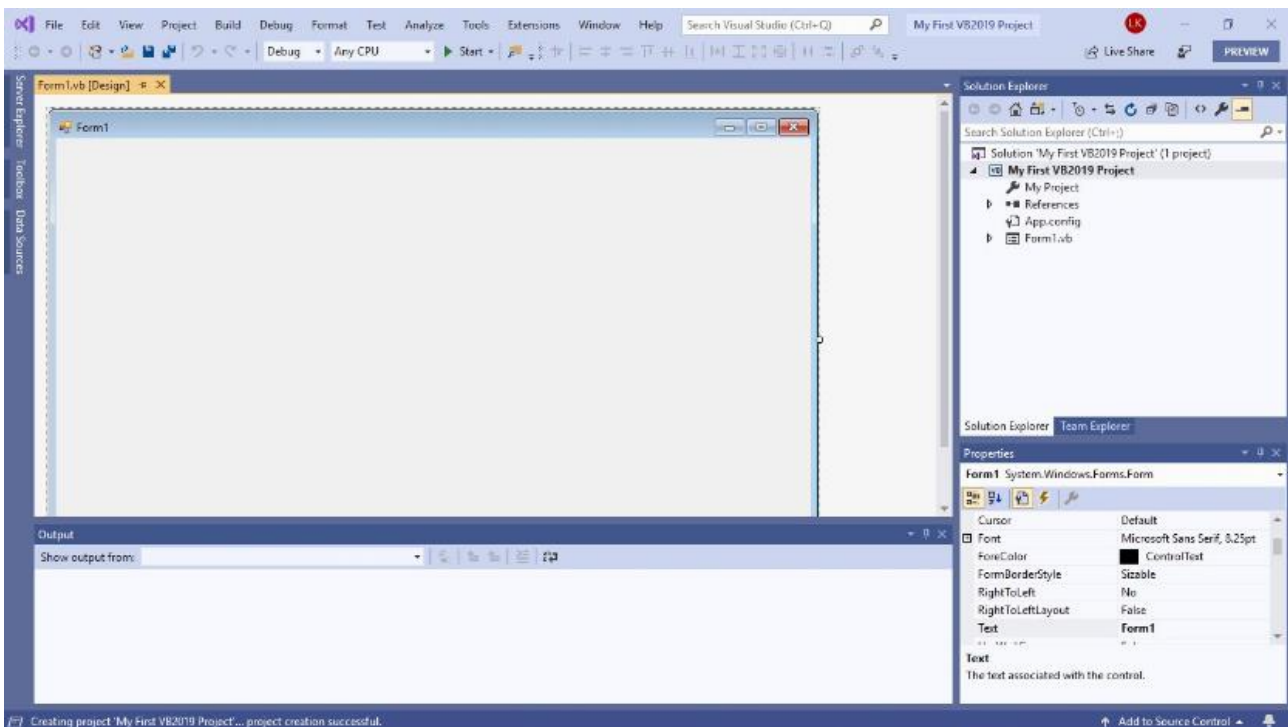
*Imagen 26: Cable USB*

## 2.5 Software de pc

El Instrumento lector RFID va acompañado por una aplicación de consola para el almacenamiento y visualización de datos tomados desde el lector de caravanas. El programa se desarrolló en “Visual Studio 2019” de Microsoft y se programó con el lenguaje “C Sharp” (C#). El programa es portable y va junto con el lector de mano.

Visual Studio es un programa gratuito y de código abierto, es un editor de código fuente. Compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un idioma dado.

El lenguaje de programación como se menciona es “C#”, es un lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, su sintaxis básica deriva de C y C++ y presenta claras diferencias, aunque estaba fuertemente basado en C++, la compatibilidad ofrecida se perdió. De tal forma que, un programa escrito en C o C++ no es válido en C#.



*Imagen 27: Visual Studio 2019*

En la imagen representa la interfaz gráfica del programa (Form1) en donde vamos a ir colocando las diferentes herramientas que necesitaremos como botones, tablas, histogramas, etc. Luego de organizar en la interfaz las distintas herramientas con que vamos a trabajar, se procede a la programación de cada una de ellas.

El programa realizado para la visualización de los datos de cada corral se explica a continuación en las 4 pestañas de la interfaz gráfica del programa.

### 2.5.1 Información

En la pestaña de información se muestra al operador los pasos a seguir en las distintas pestañas del software.



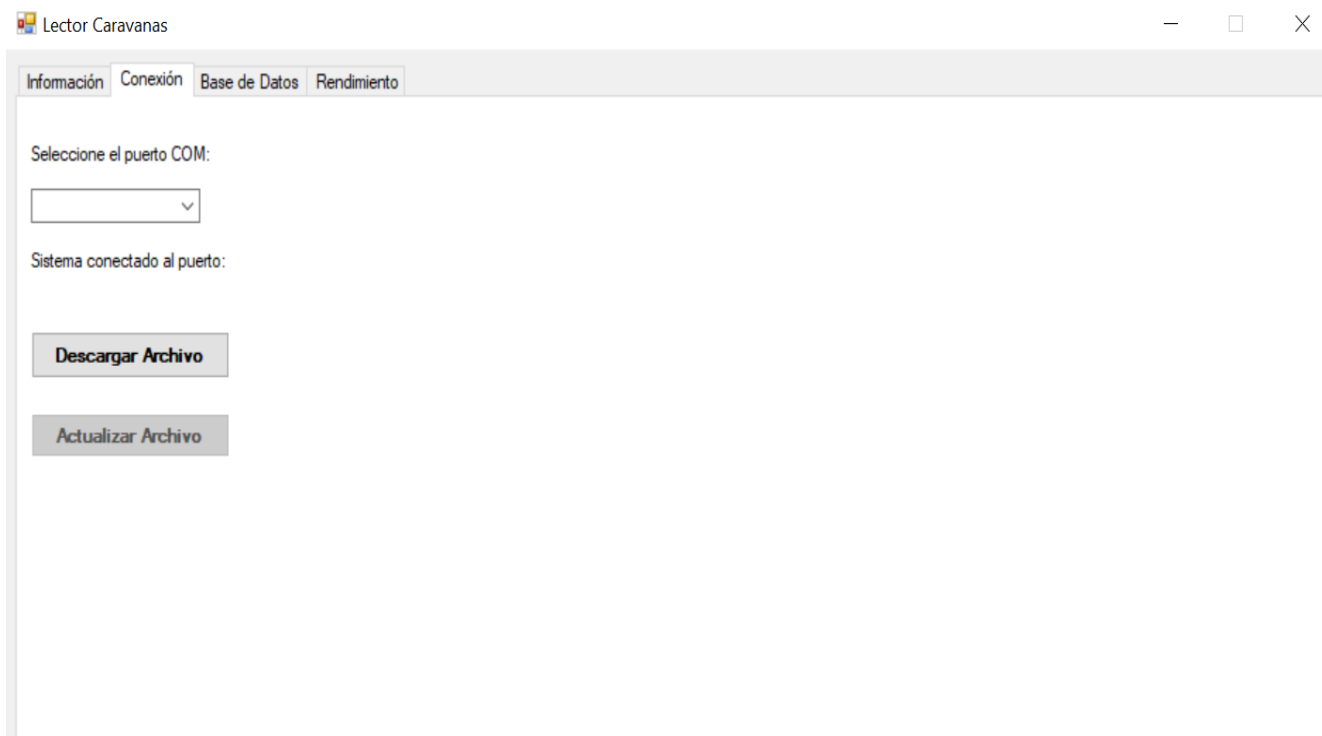
*Imagen 28: Pestaña de información*

### 2.5.2 Conexión

En la conexión se refiere a conectar el bastón lector de caravanas electrónicas a la pc que contiene el programa mencionado. Una vez conectado aparecerá en la ventana de la parte superior izquierda el o los puertos conectados a la pc deberá seleccionar el puerto correspondiente al bastón lector.

Cuando se seleccionó el puerto el sistema indica que se conectó y se procede a la descarga de información, la cual esta almacenada en la tarjeta microSD del bastón, presionando el botón "Descargar Archivo". Internamente se realizó una trama de comunicación entre el programa y el dispositivo para saber cuándo se inicia y termina el

traspaso de datos. Cuando se descargó los datos de los animales se habilita el botón “Actualizar Archivo” donde se agregan los nuevos valores a la base de datos del sistema. El programa informara con una línea de texto cuando se descargó por completo el documento (derecha del botón “Descargar Archivo”) y también informara cuando se haya actualizado el documento XML del programa (derecha del botón “Actualizar Archivo”), donde estarán guardadas mediciones anteriores.



*Imagen 29: Pestaña de Conexión*

En siguiente diagrama en bloque da una idea general del código realizado para la ventana de conexión. En el bloque de “Actualización de base de datos”, modifica el archivo XML del programa con los nuevos valores agregados y también si desde el documento proveniente de la memoria microSD llega un dato con la opción “borrar” elimina por completo el dato de su documento XML principal.

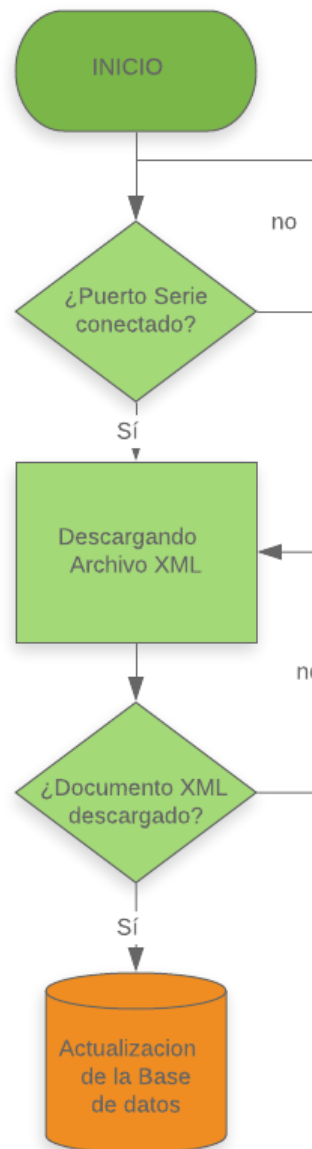


Diagrama 4: Conexión y actualización de la base de datos

### 2.5.3 Base de datos

La primera pestaña se despliega un menú y se selecciona el corral o si desea una búsqueda por caravana, si es por esta última tendrá que escribir el número de caravana que desea buscar. Si se encuentra dicha caravana se muestra sus datos en la grilla de no ser así saldrá un cartel de error indicando que no se encuentra lo que busca.

En el caso de seleccionar el corral, en la grilla, aparecerán todos los animales (con sus datos: corral, número de identificación, peso, fecha y hora que se tomó la medida). En el lado derecho de la grilla, solo para la opción de selección de corral, se puede ver

promedio y peso total del corral como también la opción de calcular el valor total en pesos que tiene en ese corral. En la parte inferior izquierda tiene la opción de eliminar caravanas. La forma es seleccionar la cantidad de caravanas que desea eliminar (por algún motivo como venta, muerte, etc.) y presionar el botón “Eliminar”. Al borrarlas se eliminan por completo de la base de datos, esto significa que también se borran los datos de lecturas anteriores guardados en la base de datos. Es importante que después de cada nueva descarga de información, luego de seleccionar el corral y mostrar por grilla, calcule el precio. De esta manera estará guardando en documentos de texto el promedio de kilo por animal, la cantidad de ellos y la fecha, que se visualizaran en la ventana de rendimiento.

Windows title bar: Lector Caravanas

Navigation tabs: Información, Conexión, Base de Datos, Rendimiento

Form fields:

- Seleccionar Corral:
- Seleccionar Caravana:
- Buscar:

Corral	ID:	Peso:	Fecha:	Hora:

Labels and buttons on the right side:

- Peso total del lote:
- Promedio de peso por animal:
- Ingrese precio por kilo de animal:
- Calcular:
- Valor total del lote:

Buttons and text at the bottom left:

- Eliminar:
- Caravanas eliminadas:

Imagen 30: Pestaña de base de datos

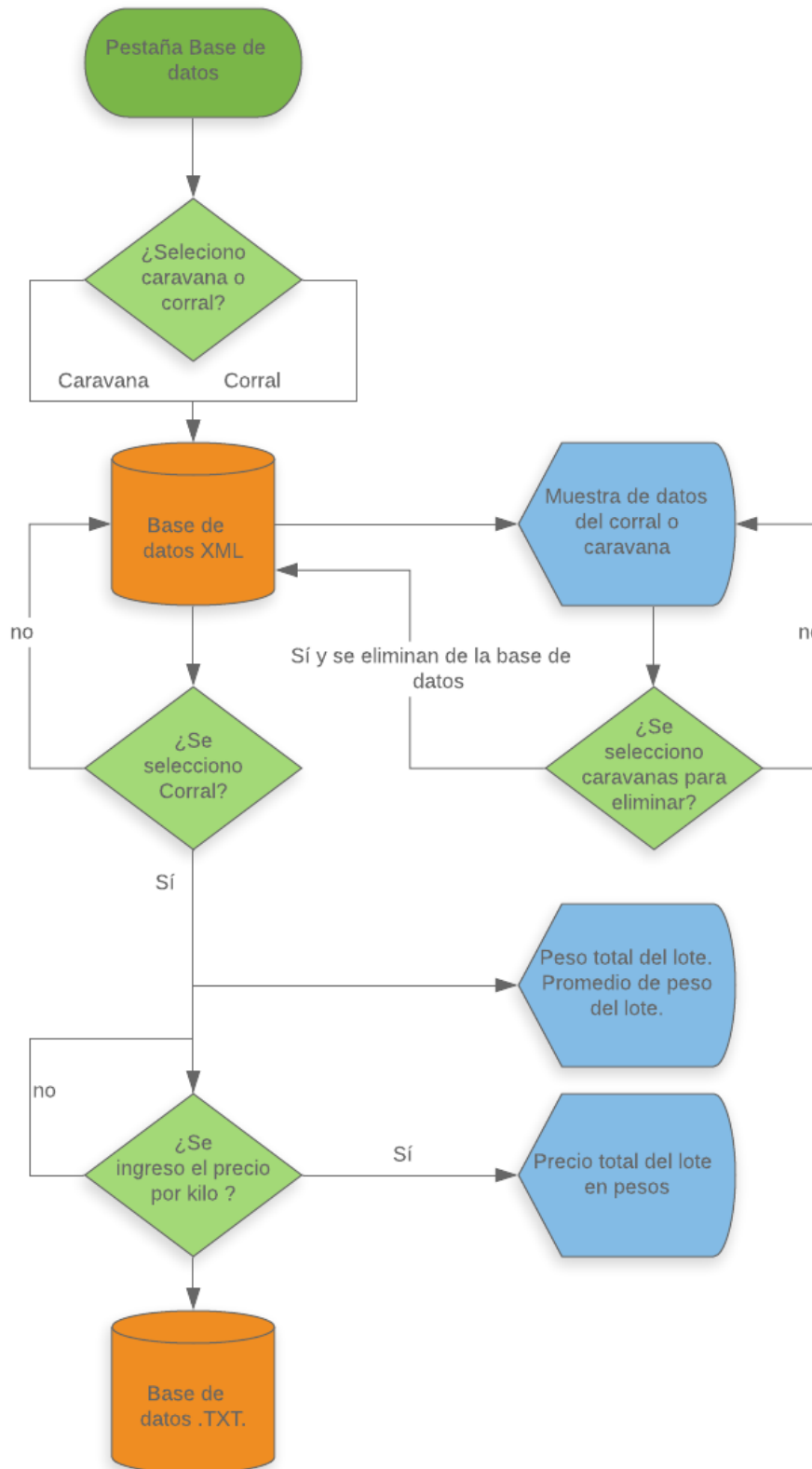


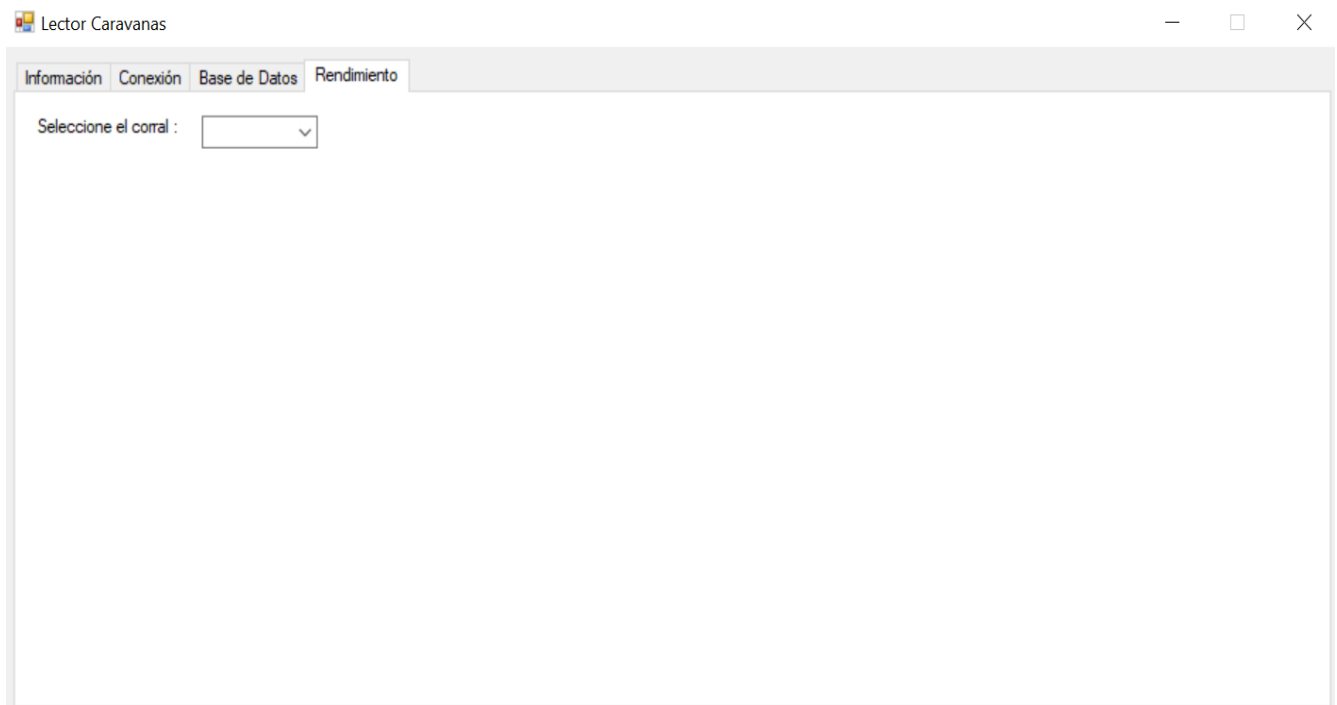
Diagrama 5: Consulta a la base de datos XML



### 2.5.4 Rendimiento

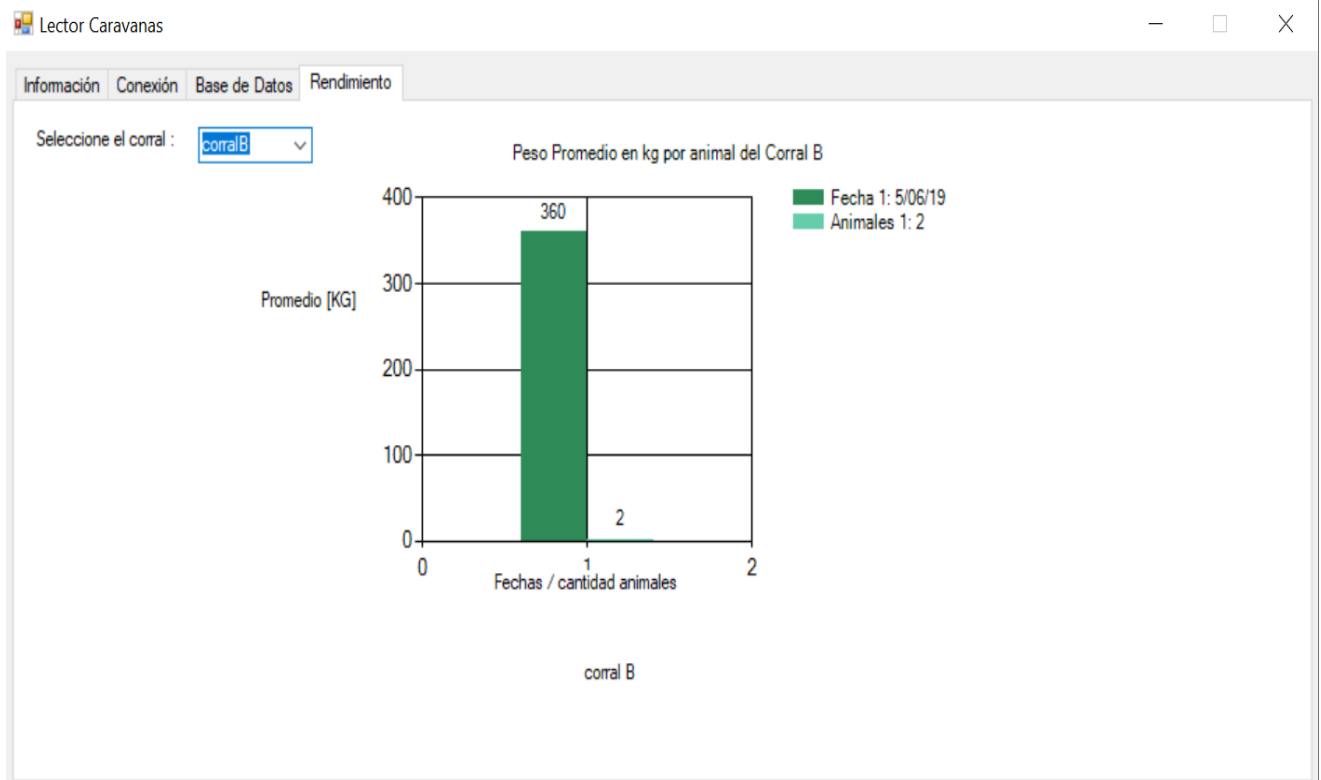
En la base de datos del ejecutable se almacena la información de las caravanas de los distintos corrales, hasta que no se elimine la caravana (ID, numero identificador) los datos de la misma se van acumulando de acuerdo a la fecha que se tomó su lectura.

Para tener un control de la evolución del peso del animal se muestra en la pestaña, mediante un histograma, el promedio de kilo por animal con su fecha correspondiente y también la cantidad de animales que tenía ese corral en esa toma de datos (que puede variar por X motivos). En las siguientes imágenes se puede apreciar lo mencionado anteriormente.



*Imagen 31: Pestaña de Rendimiento*

En el menú de “seleccione corral” aparecerán los tres corrales que están cargados en el software. Al seleccionar uno de ellos aparecerá en el histograma el historial de lecturas con: el promedio de kilos por animal y la cantidad de animales que había en esa fecha que se tomó la lectura (imagen 32).



*Imagen 32: Histograma*

Dentro del código realizado en Visual Studio 2019, en la parte que corresponde al rendimiento donde luego se muestra dichos valores en un histograma como el visto anteriormente. Estos datos se van guardando dentro de tres archivos de texto distintos de acuerdo al corral, para que el almacenamiento sea válido se debe calcular el precio que tendrá el lote después de cada actualización de la base de datos (cada vez que se extrae datos del bastón lector), como se ve en la parte inferior del diagrama 5.

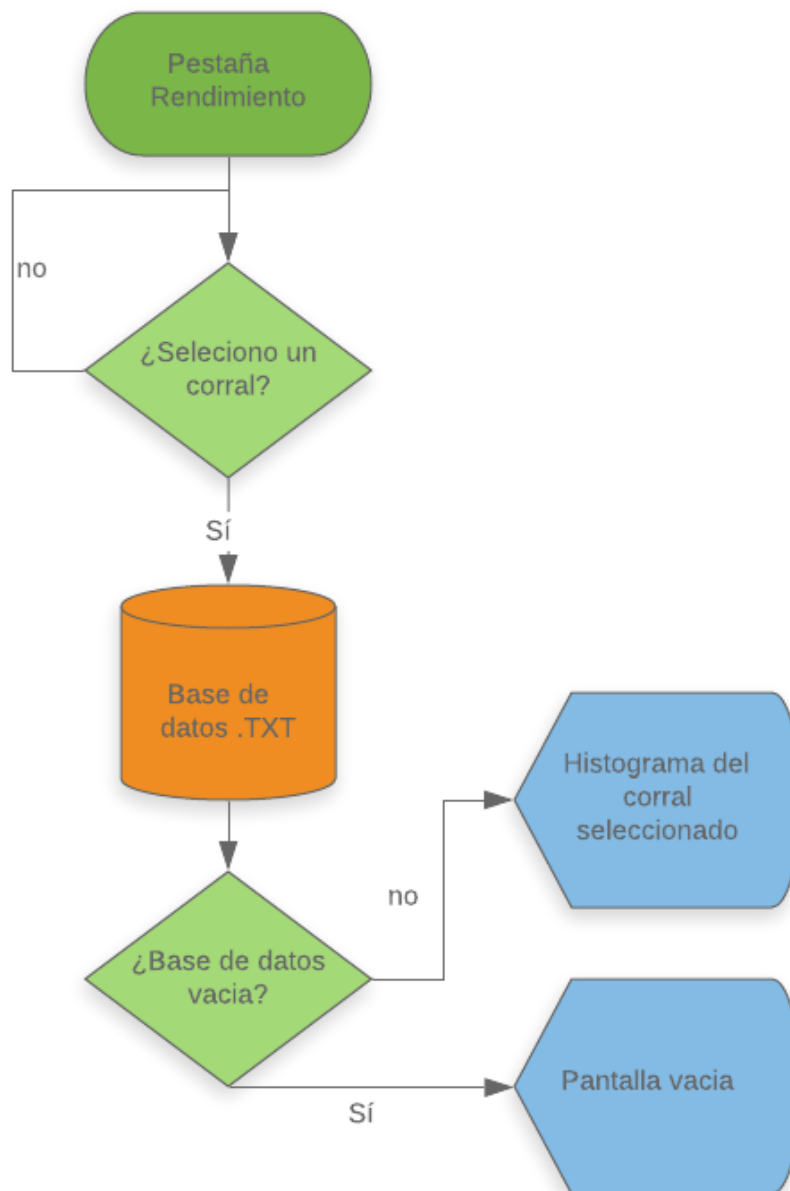


Diagrama 6: Histograma de rendimiento

## 2.6 Problemas y soluciones en Bastón lector y software

Al referirme al Lector de mano portátil de caravanas electrónica, su proceso de programación y armado ocurrieron una gran variedad de inconvenientes uno de ellos y el más importante fue en la comunicación bluetooth y RFID, en el primer código se utilizó interrupciones internas y de puerto serie para la comunicación de ellos con la unidad de control. Ocurriendo fallas en la transmisión bluetooth, los datos provenientes de la

aplicación de celular no eran transferidos al arduino, se solucionó creando funciones tanto para bluetooth como para RF. El inconveniente es que se necesita que llegue primero el dato de la balanza para habilitar la lectura de las caravanas (Modulo RFID).

El siguiente problema es que el documento XML creado en la tarjeta de memoria microSD tiene que presentar un encabezado, seguido del nodo raíz y luego de los siguientes nodos que contienen los datos leídos, esto es necesario para que el software reconozca que es un archivo del tipo XML y pueda actualizar el archivo principal con los nuevos valores.

El encabezado y nodo raíz son incluidos en la memoria cuando se ingresa al menú de selección de corral. Una vez que el operario tome las lecturas y vuelva al menú principal el documento se cierra con el nodo raíz (explicado en capítulos anteriores). Si por alguna razón vuelve a entrar al menú de selección de corral se vuelve a crear (encabezado y nodo padre, etc.) y esto se va almacenando en el mismo documento produciendo un error cuando se desea descargar los datos al programa. Una mejora a futuro sería resolver este problema, se puede modificar el código grabado en el microcontrolador o también el código realizado en el software.

En el caso que se siguió los pasos correctamente y salió del menú (sin volver a ingresar) el documento XML queda guardado de forma correcta y el programa no tendrá problema alguno en reconocerlo y actualizar su base de datos.

En el programa desarrollado en Visual Studio el principal problema fue crear el archivo XML por software, la solución a esto fue realizar el documento XML mediante un programa llamado "Notepad++", el cual es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación, crear la estructura jerárquica (imagen 25) que va a tener mi archivo y luego guardarlo dentro de la carpeta del software.



*Imagen 33: Logotipo Notepad++*

Cada vez que necesite actualizar, borrar, mostrar datos, el programa va a la dirección donde está guardado el archivo y realiza los cambios. Fue la solución más sencilla de realizar el archivo que por funciones dentro del Visual Studio.

## 2.7 Diseño Completo

Para la realización del gabinete del lector de mano portátil utilice un programa de diseño gráfico y modelado en tres dimensiones (3D) llamado "SketchUp", realizado a escala. Una vez realizado el diseño fue enviado para su construcción a una impresora 3D.

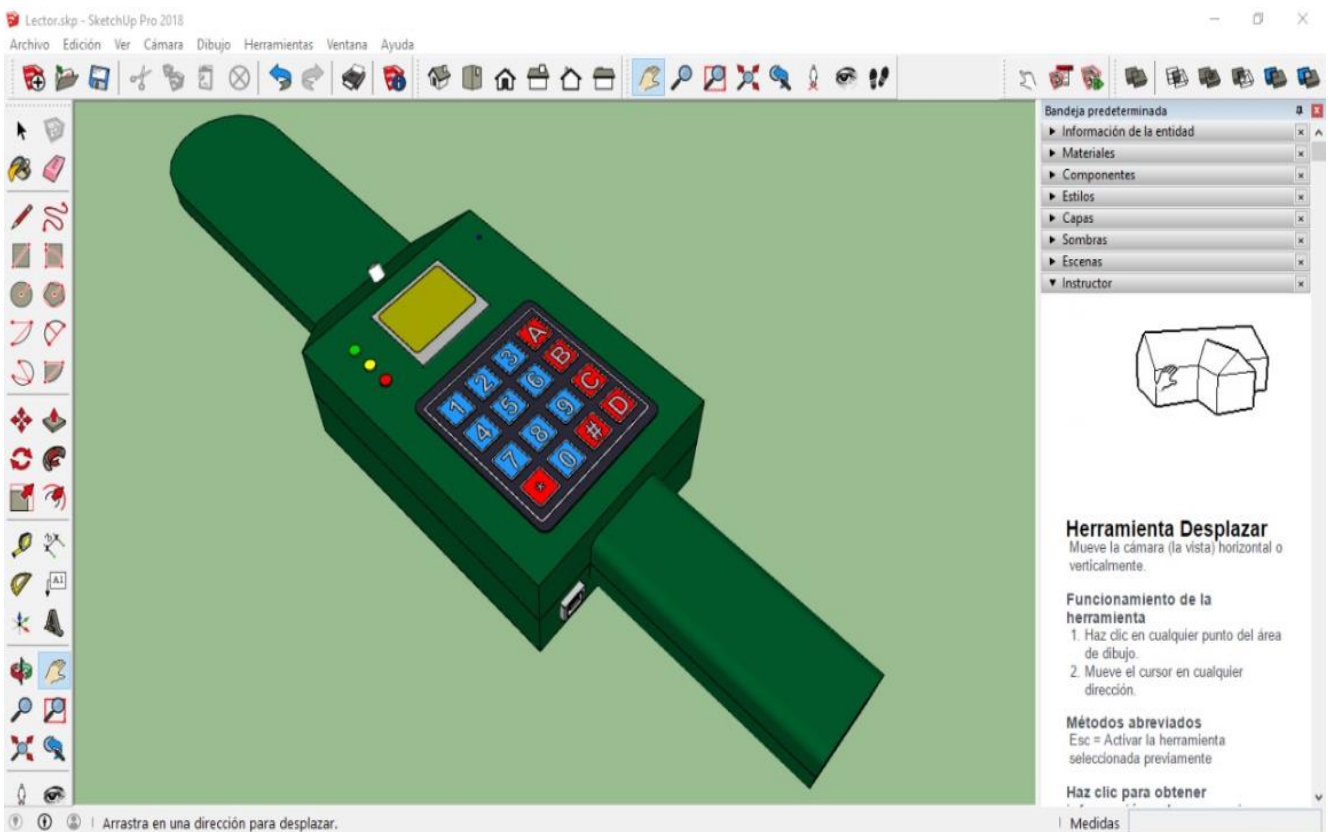


Imagen 34: Diseño en SketchUp

Una vez ensamblado y conectados todos los componentes, el dispositivo es el que se ve en la siguiente fotografía donde:

1. Ubicación antena RFID.
2. Potenciómetro regulador de intensidad de luz de la pantalla.
3. Led indicador de conexión Bluetooth.
4. Display LCD Nokia 5110, visualización de menú.

5. Led indicadores de estado de carga de la batería.
6. Teclado
7. Pulsador
8. Interruptor de encendido.
9. Compartimiento de batería.

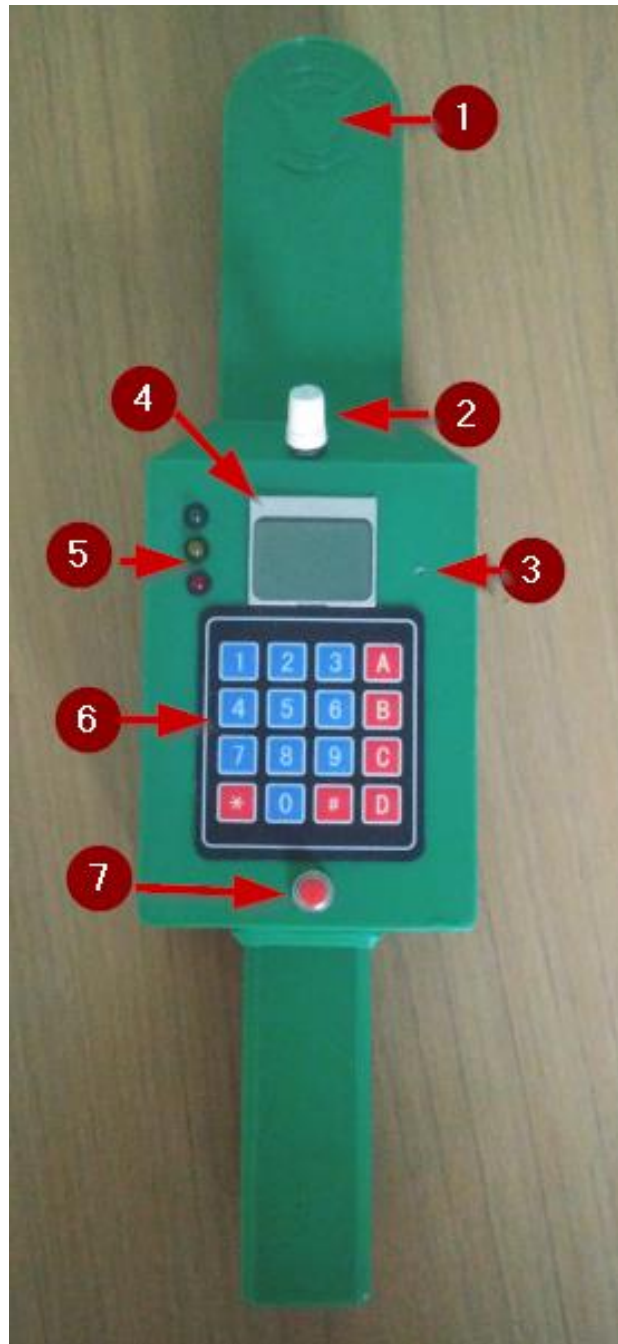


Imagen 35: Prototipo Lector de mano portátil



*Imagen 36: Compartimiento de batería.*

Las siguientes fotografías indican las distintas etapas del menú del bastón lector dependiendo del botón que se presionó. A la derecha del menú, en número, se indica que numero pulsar.



*Imagen 37: Menú principal.*

El menú principal contiene 3 opciones, al presionar "1" entrara a un submenú de elección de corral, del cual tomara lectura de los animales que contiene (corral A, B o C). Los submenú presentan una tecla de retroceso (tecla 4).



*Imagen 38: Selección del Corral*

El ejemplo que voy a desarrollar corresponde a la elección del corral A.



*Imagen 39: Lectura de caravana*





*Imagen 40: Comunicación con la balanza*

Al llegar al menú “Esperando Balanza” significa que el animal está dentro de la misma y hasta que el peso se estabilice y sea enviado al dispositivo no está habilitado la opción de presionar el pulsador para tomar el número de identificación de la caravana. Al llegar el valor del peso se indica “Balanza OK” y permite la lectura.



*Imagen 41: Habilitación del pulsador.*

Con la caravana a no más de 5 centímetros de la parte frontal donde se encuentra la antena transmisora y presionando el pulsador, tendremos el siguiente resultado.

1. Número de identificación de la caravana.
2. Pesos del animal, en kilogramos.
3. Borrar o no el dato en el XML del software, si el animal por algún motivo no va a estar más en el corral al cual pertenece.



*Imagen 42: Lectura Correcta.*

Luego de presionar “2”, en el caso de no eliminación de caravana, los datos se guardan en la memoria microSD.



*Imagen 43: Datos almacenados en memoria.*

Al volver al menú principal y seleccionando la opción de reloj, veremos los datos extraídos del módulo RTC por pantalla.



*Imagen 44: Datos del RTC.*

Cuando se ingresa a la opción número "3" del menú principal, primeramente el aparato tiene que estar conectado a la pc por medio de su cable USB, informa por pantalla que se está descargando el archivo una vez finalizada la descarga vuelve al menú principal y está listo para comenzar nuevas lecturas.



*Imagen 45: Descarga del documento XML.*

El producto final es un prototipo de bastón lector para caravanas electrónicas, debo aclarar que el dispositivo deberá pasar por ciertas pruebas realizadas por el INTI (Instituto Nacional de Tecnología Industrial) donde se resolverá si es apto para el mercado hacia el que fue producido, certificando por parte de la entidad que el prototipo brinda ciertas garantías sobre su correcto funcionamiento.

Otro requisito necesario es el de inscribirse en el registro de fabricantes de dispositivos de identificación animal como proveedores de dispositivos de identificación electrónica (RFID) de baja frecuencia (caravana electrónica tipo botón) y/o lectores de los mismos, para la identificación de bovinos, bubalinos y cérvidos, esto lo dicta el Servicio Nacional de Sanidad y Calidad Agroalimentaria (Senasa) en su Resolución N° 678-E/201.

## Capítulo 3: Resultados

### 3.1 Resultados obtenidos

El producto se realizó, principalmente, como una opción más económica a las que presenta el mercado de dispositivos de identificación animal, refiriéndome al instrumento, en cuanto al software que viene de la mano con el bastón lector busque realizar mejoras tanto en la información que le proporcionan al productor (de ganado en este caso) como también en la facilidad de manejo del programa.

Mucho de los programas de los distintos lectores RFID que existe en el mercado solo informan el número de caravana, en algunos el peso y la fecha que se leyó. Mi producto informa, no solo lo que se mencionó, sino que también da la posibilidad de conocer cuánto dinero tiene en el corral en el momento que se descargan los datos de la nueva medición de caravanas, ver el rendimiento del avance (o no) de los kilos del lote que se toma medición para tener una idea de si se tiene que mejorar, cambiar o aumentar la ración diaria de alimentos, ya que está pensado para engorde de animales a corral, de esta manera tener un control más sofisticado sobre la producción.

Para conocer la clasificación del lector están las siguientes categorías:

- **Categoría 1:** Estos transceptores RFID solamente incorporan la funcionalidad básica de lectura según el ISO 11784/85 y carecen de: baterías recargables, conectividad con otros dispositivos o interfaz de usuario ininteligible.
- **Categoría 2:** Implica aquellos transceptores RFID ISO 11784/85 que poseen una interfaz de usuario basada en botones o teclados sencillos, conectividad con otros dispositivos ya sea mediante USB, Bluetooth o RS232, asimismo, poseen una batería recargable.
- **Categoría 3:** Son aquellos lectores que tienen capacidad de recolectar datos de trazabilidad. Asimismo tienen mejoras en la conectividad con otros dispositivos ya que además de interfaz USB, Bluetooth o RS232, pueden poseer conectividad Wi-Fi o GPRS.
- **Categoría 4:** Son aquellos dispositivos portátiles que incorporan un sistema operativo y son la evolución de la categoría 3.

El lector que desarrolle es de categoría 2.

## Capítulo 4: Análisis de Costos

### 4.1 Análisis de costos:

La siguiente tabla indica el costo de realización del producto (1 unidad):

<b><i>Materiales:</i></b>	<b><i>Cantidad:</i></b>	<b><i>Precio Unitario [\$]:</i></b>	<b><i>Precio Total [\$]:</i></b>
Placa Arduino Mega2560	1	1171,00	1171,00
Modulo Bluetooth HC-05	1	482,88	482,88
Módulo RFID	1	31 [US\$]	1851,50
Buzzer pasivo	1	77,95	77,95
Reloj de tiempo real DS3231	1	202,92	202,92
Modulo memoria microSD	1	92,08	92,08
Memoria MicroSD de 4GB	1	194,97	194,97
Display Nokia 5110	1	315,00	315,00
Teclado Matricial 4x4	1	86,39	86,39
Pila 9V	1	95,00	95,00
Gastos varios	1	4100,00	4100,00
<b>TOTAL</b>			<b>8669,69</b>

*Tabla 1: Gastos en fabricación y diseño*

El proyecto tardó en su realización un periodo de 6 meses en la cual trabajé aproximadamente 8 horas diarias de lunes a viernes, esto da un tiempo de trabajo aproximado de 1216 horas. Al consultar al Colegio de Ingenieros Especialistas de Entre Ríos (CIEER) en su página web, en la parte de aranceles y de acuerdo con la RESOLUCIÓN N° 1289/2016 en su artículo 5, la cual establece un monto de 160 pesos por hora de trabajo, da un valor de 194.560 pesos (ciento noventa y cuatro mil quinientos sesenta).

El producto ya terminado tendrá un costo de 20.000 pesos (veinte mil). Es nos da una ganancia de 11330 pesos (once mil trescientos treinta) por producto. Para saldar el trabajo realizado necesitare que se vendan 18 unidades de lectores de mano portátil para caravanas electrónicas, superando ese monto lo demás es ganancia.

## Capítulo 5: Discusión y Conclusión.

### 5.1 Conclusión y mejoras a futuro

Lo que me propuse se cumplió, que fue realizar un producto que tenga las mismas prestaciones de los que están en el mercado y mejorar el software de aplicación que viene junto con los lectores comerciales. El dispositivo detecta las caravanas electrónicas si bien el rango de lectura (5 cm) es muy reducido, es fiable por debajo de él. Es un aspecto a mejorar aumentar la potencia de emisión del campo electromagnético para tomar lecturas de caravanas a un rango de 30 a 40 cm que es la distancia que la mayoría de los instrumentos comerciales trabajan.

El programa realizado para tener una base de datos de las lecturas es muy completo, se puede seguir mejorándolo como por ejemplo, tener otro archivo conjunto al principal que sea de caravanas eliminadas, cuestión de tener los datos de los animales que terminaron el proceso de engorde a corral y fueron vendidos. Presentar la evolución de peso de cada animal en un histograma, ya que el programa realizado muestra la evolución del promedio de peso por animal en el corral.

El producto compite con lectores que arrancan en los 65.000 (sesenta y cinco mil) pesos hasta bastones lectores que superan los cien mil pesos. Siento los más baratos de igual categoría que mi producto. De igual manera para lanzarlo al mercado se necesita la aprobación del INTI y del SENASA, un requisito es que el gabinete tenga protección con el polvo y agua el tipo de protección es IP6, donde IP (*internacional protección*) y 6 significa el nivel máximo que es donde no tiene que entrar ningún tipo de polvo.

Otra mejora a futuro sería reemplazar las pila alcalina de 9 volts por pilas de litio recargables con su correspondiente modulo cargador y de protección, dentro del compartimento de la batería, más que nada para no estar cambiando de batería cada vez que se agoten, simplemente conectaríamos el cargador y estaría en funcionamiento nuevamente el lector portátil.

Una nueva mejora sería la parte del teclado, colocando un teclado más "rustico" debido al trabajo que se realiza. Se puede incrementar la categoría del instrumento agregándole un módulo para poseer conectividad Wi-Fi o GPRS, lo cual incrementaría también el precio del bastón lector.



**Capítulo 6: Literatura Citada.**

1. Alldatasheet. (12 de Noviembre de 2019). *Alldatasheet*. Obtenido de <https://www.alldatasheet.com>
2. Arduino. (12 de Noviembre de 2019). *Arduino* . Obtenido de <https://www.arduino.cc/>
3. Microsoft. (12 de Noviembre de 2019). *Microsoft Visual Studio 2019*. Obtenido de <https://visualstudio.microsoft.com>