|  |  |  |
| --- | --- | --- |
|  | LOGO FACET 2011.jpg | **Facultad de Ciencias Exactas y Tecnología**  Universidad Nacional de Tucumán| |

UNIVERSIDAD NACIONAL DE TUCUMAN

FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGIA

Departamento de Eléctrica,

Electrónica y Computación



TRABAJO DE GRADUACION

INFORME FINAL

FEEDLOT AUTOMATIZADO

Alumnos:

Raimondo, Matias Sebastian

Robles, Lucas Matias

Ruiz Holgado, Matias

Carrera:

Ingeniería en Computación

Tutor:

Ing. Carlos Sueldo

Fecha de presentación

19/10/2017

**ÍNDICE DE CONTENIDOS**

**CAPITULO 1** – INTRODUCCIÓN

1. INTRODUCCIÓN. 5
2. OBJETIVO DEL PROYECTO. 5
3. OBJETIVOS ESPECÍFICOS DEL PROYECTO 5

**CAPITULO 2** – REQUISITOS DE DISEÑO

1. DESCRIPCIÓN DEL FEEDLOT . 6
2. LAS ESTADISTICAS EN FEEDLOT. 8
3. SOFTWARE Y HARDWARE UTILIZADOS. 10

**CAPITULO 3** – HARDWARE

1. ARDUINO2650. 12
2. SENSORES. 14
3. ACTUADORES. 18

**CAPITULO 4** – DESCRIPCIÓN DE LA AUTOMATIZACIÓN

1. PROGRAMACION DEL ARDUINO. 21
2. ALMACENAMIENTO DE DATOS Y LECTURA EN SOFTWARE. 22

**CAPITULO 5** – SISTEMA DE SOFTWARE

1. IDEA DE PRODUCTO. 25
   1. IDENTIFICACIÓN DE USUARIOS PARTICIPANTES. 25
   2. ESPECIFICACIÓN DE REQUISITOS. 25
   3. ESTUDIO DE VIABILIDAD. 29
2. ESPECIFICACIÓN C. 29

2.1- Modelo E-R 29

2.2- Capturas de pantalla software 29

2.3- GUIONES Y ESCENARIOS. 29

**CAPITULO 6** – CONCLUSIONES E INVESTIGACIONES FUTURAS

1. CONCLUSIÓN.
2. INVESTIGACIONES FUTURAS.

**BIBLIOGRAFÍA Y RECURSOS WEB**

ANEXOS

**ANEXO A** – CONCEPTOS TEÓRICOS PRÁCTICOS INVOLUCRADOS

**ANEXO B** – DISEÑO DE PLAQUETAS

**ANEXO C** – HOJA DE DATOS

**ANEXO D** – CODIGO FUENTE

**ÍNDICE DE ILUSTRACIONES**

**INTRODUCCIÓN**

1. **INTRODUCCIÓN**

El Objetivo de este proyecto es automatizar el rastreo individual de bovinos durante el proceso de engorde en un FEEDLOT( Engorde de ganado en Corral), así también la presentación de estadísticas a nivel individual y global.

Para lograr dicho objetivo se diseñará un dispositivo que permita recolectar información de una manera accesible para el usuario y se desarrollará un software que procese la información y la presente de una manera amigable al usuario

La recolección de información contempla las siguientes variables: Ingreso, vacunación, peso y ubicación en corral del ganado.

El Ingeniero Zootecnista Pedro Perez quien dirige el proceso de FEEDLOT del corral XX comentó la deficiencia que presenta el actual sistema que posee el corral XX, ya que no poseen una forma fehaciente que les permita ubicar a cada cabeza de ganado de manera individual en las diferentes etapas del proceso, de manera contemporánea una resolución dictada por el AFIP para poder llevar registros de la cantidad de cabezas de ganado a nivel nacional impulsaron la creación de este proyecto.

Utilizando los conocimientos obtenidos durante el cursado de la carrera se buscará poder cubrir estas necesidades y desarrollar el dispositivo que se adapte al ambiente donde se desarrolla la actividad y pueda recolectar la información necesaria para poder llevar un rastreo eficiente del ganado.

1. **OBJETIVO DEL PROYECTO**

Diseño y creación del dispositivo que tenga todos los periféricos necesarios para poder brindarle al usuario la posibilidad de recolectar datos con un dispositivo de fácil manejo enfocado en el diseño y las prestaciones; el desarrollo de un sistema de software que complemente la recolección de datos procesando la información permitiendo al usuario visualizar el seguimiento individual y las estadísticas requeridas. brindándole una interfaz amigable y útil.

1. **OBJETIVOS ESPECÍFICOS**

* Investigación sobre proceso de FEEDLOT.
* Análisis ambientales del corral para contemplar diseño del dispositivo.
* Elección del microprocesador y tecnología de desarrollo.
* Análisis de Sensores y Actuadores.
* Elección la tecnología para el desarrollo del Sistema.

**REQUISITOS DE DISEÑO**

1. **DESCRIPCIÓN DE LOS FEEDLOT**

El engorde en corrales se ha instalado en la región como una alternativa más de producción de carne con diversos objetivos. En algunos casos es convertir granos a carne si económicamente la conversión es rentable, y en otros, aunque puede ser poco atractivo desde el punto de vista de la conversión, interesa desde el conjunto del sistema de producción para liberar potreros, eliminar cultivos forrajeros anuales de las rotaciones, incrementar la carga animal del campo, asegurar la terminación y la salida o la edad a faena, etc.

Para plantear el engorde a corral es necesario entender desde el principio que seremos nosotros y no "el pasto" quien define la composición de la dieta del animal, cuánto va a comer y cómo lo hará. Es posible realizar encierres simples, con pocos insumos y lograr engordes aceptables. Las primeras definiciones pasan por la categoría a encerrar y la cantidad de animales. Luego resolveremos aspectos relacionados con la ubicación en el campo, la dieta o alimento y su suministro. Estos dos últimos son frecuentemente los condicionantes del planteo en términos

físicos y económicos. Finalmente, es importante tener en cuenta los aspectos del diseño que puedan afectar la calidad del proceso de producción, entendiéndose por ello el diseño de las instalaciones y el manejo de los efluentes emergentes, factores éstos que pueden condicionar la gestión productiva en el mediano y largo plazo.

1. **SOFTWARE Y HARDWARE UTILIZADOS**

SOFTWARE

Para poder trabajar en equipo se decidió utilizar **GITHUB** como repositorio del software, dicho software se encuentra en el siguiente repositorio:

* **Definir Repo**

De acuerdo a las necesidades del programa a desarrollar se eligió el lenguaje **Java** ya que posee virtudes muy importantes:

* Posee características de performance necesarias para el desarrollo del sistema.
* Ya se posee conocimiento del lenguaje, eliminando la necesidad de aprender un lenguaje, sus buenas practicas y metodologías de desarrollo.

La existencia de Frameworks que facilitan el desarrollo en dicho lenguaje tales como **Spring**, **Hibernate** asi también gestores de proyectos que permiten asegurar una estructura solida y de facil mantenimiento como Maven o Gradle.

* Licencia Pública General de GNU, esta licencia es la más usada en el mundo del software y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

Al desarrollar el sistema con el lenguaje **JAVA** se abre la posibilidad de utilizar frameworks que permiten la disminución del tiempo de desarrollo así como la mejora en la calidad del código.

Para el desarrollo del mismo hemos elegido como motor de persistencia **MySQL**.

* **MySQL** es un sistema que permite la gestión de base de datos relacional, multihilo y multiusuario.
* Soporta gran cantidad de datos.
* Posee Licencia Pública General de GNU ya que nos garantiza la libertad de usar y estudiar el software.

La inversión de control en desarrollo de software permite un código poco acoplado y mas mantenible, se utiliza el framework **SpringBoot** para la implementación de inyección de dependencias, este framework permite utilizar fácilmente un ORM como **Hibernate** como implementación de JPA y **Flyway** para la migración en base de datos.

Como gestor de proyectos se usa Maven ya que permite descargar dependencias, definir roles para ambientes de desarrollo y fácil integración de flyway.

Para poder desarrollar en Java, se puede elegir un sencillo bloc de notas pero para desarrollar de mejor manera se utilizará un IDE que facilita el desarrollo.

Luego de analizar las necesidades del software a desarrollar, elegimos **NetBeans** 8.2 desarrollado por la empresa [Sun Microsystems](http://es.wikipedia.org/wiki/Sun_Microsystems)/[Oracle Corporation](http://es.wikipedia.org/wiki/Oracle_Corporation), y es un [entorno de desarrollo integrado](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) [libre](http://es.wikipedia.org/wiki/Software_libre) y gratuito, sin restricciones de uso.

Versionado de herramientas de desarrollo

Java 1.7

Maven 2.3

Spring 1.4

Hibernate 5.1

Flyway 4.0.3

HARDWARE- SIMULACION

Para diseñar el dispositivo se llevo a cabo el siguiente análisis:

1. Identificación de hardware necesario
2. Diseño de bloques
3. Placa de desarrollo.
4. Terminal de desarrollo para software de dispositivo y aplicación.
5. Módulos:

* Módulo de RFID
* Módulo RTC
* Display LCD 16x2
* Adaptador I2C
* Módulo SD
* Teclado Matricial
* Gabinete

1. Alimentación

* Fuente de 12 V
* Pilas AA
* Pila 3V

**HARDWARE**

**1- Identificación de Hardware necesario**

Para poder cumplir las funciones necesarias para la recolección de datos se reconocieron las siguientes necesidades:

* Procesamiento de datos.
* Visualización de opciones y funciones.
* Lectura de ID
* Selección manual de opciones e ingreso de datos
* Almacenamiento autónomo de fecha y hora
* Almacenamiento de datos
* Alimentación autónoma del dispositivo

**PORQUE SE DECIDIO POR ARDUINO**

**Procesamiento de datos: Se decidió utilizar ARDUINO debido a que cubre debidamente las necesidades del proyecto, su interfaz sencilla, al soporte oficial que posee, la documentación existente en internet, fácil acceso a los módulos y a placas de desarrollo.**

**Visualización de opciones y funciones:** existe una gran variedad de display, pero se eligió un LCD display 16x2; esta pantalla permite visualizar completamente las opciones existentes y posee bajo consumo, asi como tambien presenta una interfaz amigable para el usuario al que está destinado el dispositivo.

**Lectura de ID**: se implementará tecnología RFID para identificación a distancia, ya que los tags pasivos permiten solamente tener que alimentar al dispositivo que utiliza el usuario y no cada uno de los usados por los animales, así también como su reutilización.

**Selección manual de opciones e ingreso de datos:** mediante un teclado matricial que incluirá números y teclas de opciones, su robustez se adapta al ambiente donde dicho dispositivo será utilizado y el usuario ya se encuentra familiarizado con el mismo.

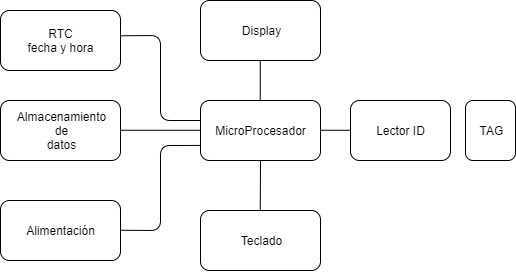
**Almacenamiento autónomo de fecha y hora:** se realizará en un módulo RTC para que al apagar el dispositivo se siga llevando registro de este dato y se deje al usuario la menor cantidad de responsabilidades.

**Almacenamiento de datos:** La utilización de memorias SD es muy común hoy en dia ya que son un medio confiable y de fácil acceso, se utilizará un módulo SD y una tarjeta para la persistencia de datos.

**Alimentación autónoma del dispositivo:** Para la autonomía del dispositivo utilizaremos **LLENAR CON DATOS DE ALIMENTACION**

**2- Diseño de bloques**

En la **Figurax** se definió un diagrama de bloques del dispositivo necesario.



**3- Placa de desarrollo**

La placa de desarrollo permite una conexión directa de la computadora, herramienta de desarrollo, con un Microprocesador, permite realizar el debug de la programación que dicho Microprocesador está ejecutando y posee consola para poder visualizar la transferencia que está ejecutando el microprocesador.

Existen varias opciones a elegir según el Microprocesador que se desee programar, si se elige un AtMega, las placas Arduino presentan una interfaz de desarrollo tales como Arduino MEGA, Arduino UNO, etc, con un IDE correspondiente. En el caso de elegir un Microprocesador FREESCALE dicho fabricante ofrece placas FRDM para desarrollo libre y un IDE apropiado. Si se desea utilizar un Microprocesador de MICROCHIP mas conocidos como PIC, se ofrece también una placa programadora y un IDE correspondiente.

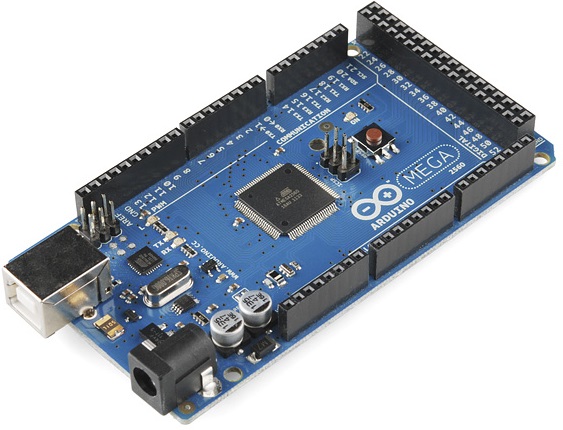
Las posibles aplicaciones que se pueden brindar dichos dispositivos exceden al proyecto ya que la aplicación que se busca es puntual con un número definido de periféricos, los Microprocesador pueden administrar sensores y actuadores, pueden utilizar memorias y display, presentan mucha potencia y capacidad de procesamiento y estas características hacen que estos elementos sean fundamentales para el desarrollo de este proyecto y la necesidad de comunicarse con ellos crean una necesidad de adquirir una placa de desarrollo.

Para el desarrollo completo de este proyecto se necesita poder realizar las siguientes acciones:

* Determinar fecha y hora de las acciones realizadas
* Presentar información al usuario de manera visual y auditiva.
* Tener la capacidad de ingresar datos manualmente y poder elegir la acción a realizar.
* Tener la capacidad de almacenar datos.
* Poder ser utilizado sin ninguna conexión a una fuente de energía.

La placa de desarrollo Arduino MEGA, **FiguraX,** permite una conexión rápida para el inicio de desarrollo, potenciado por un Microprocesador ATmega2560, presenta 54 I/O digitales las cuales 15 que pueden ser utilizadas como salidas PWM, 16 entradas analógicas, un crystal de 16MHz, conexiones tipo I2C, SPI, 4 UARTs, , y un ambiente de desarrollo estable y fácil de usar.

Posee también librerías oficiales de acceso público para el manejo de periféricos y una comunidad amplia para la resolución de posibles inconvenientes.



Toda la documentación oficial, ayuda, tutoriales y dispositivos disponibles para conectar con Arduino se encuentran en su pagina oficial:

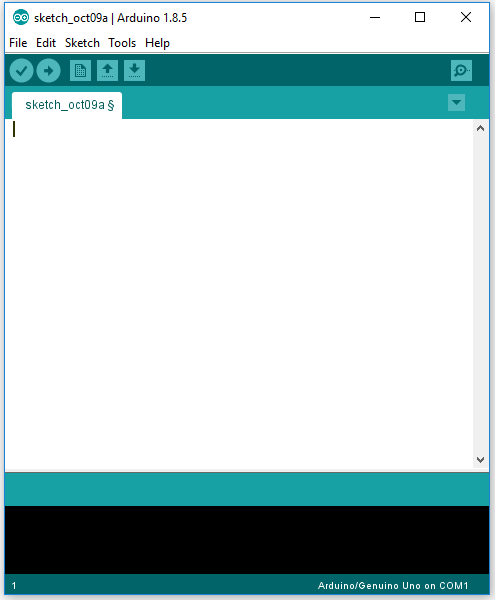
<https://www.arduino.cc/>

**4- Terminal de desarrollo para software de dispositivo y aplicación.**

Arduino posee un ambiente de desarrollo especifico para sus placa, posee una interfaz sencilla, es de acceso gratuito y puede ser descargado desde su pagina oficial:

[**https://www.arduino.cc/en/Main/Software**](https://www.arduino.cc/en/Main/Software)

Actualmente se encuentra en su versión 1.8.5, la **FIGURAX** muestra la interfaz del IDE

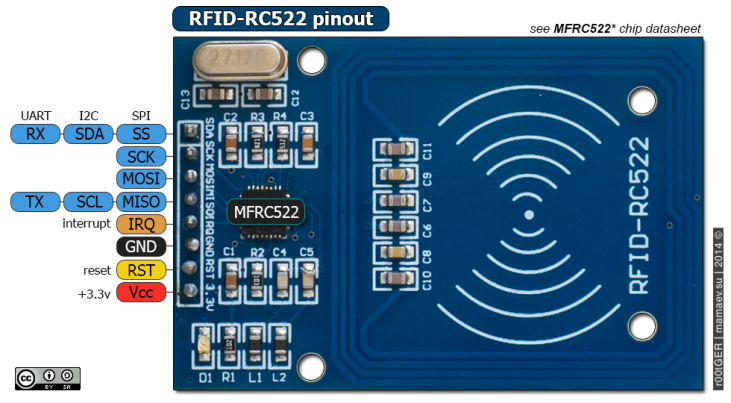


**5-** **Módulos:**

Para los periféricos necesarios para recopilar datos y para su almacenamiento, se pueden utilizar módulos que se conectan directamente a la placa de desarrollo, esto permite realizar el desarrollo y pruebas completamente de un dispositivo sin la necesidad de diseñar y soldar componentes, es un ahorro importante de tiempo y gastos de desarrollo. De acuerdo al diagrama de bloques, **FIGURAX**, se utilizarán los siguientes módulos:

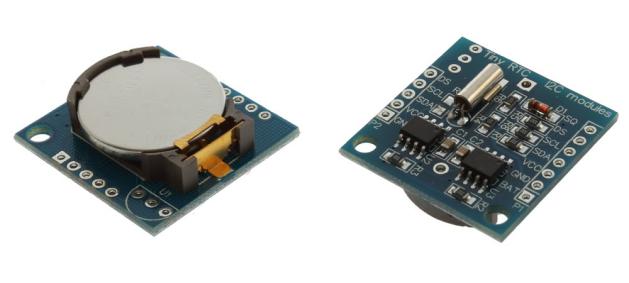
* Módulo de RFID

Los bovinos necesitan ser diferenciados el uno del otro, se debe asignar una identidad la cual debe ser leída para recolectar la información del mismo, para poder asignarle una identificación el bovino poseerá un TAG el cual es un dispositivo pasivo que almacena un identificador univoco en un microchip que al ser alimentado emite ese identificador mediante una antena que permite ser leído por un lector RFID, **LLENAR INFO** , el módulo RFID que alimenta dicho TAG obtiene el identificador y dicho ID queda a disponibilidad del microprocesador para su utilización. **LLENAR INFO COMO FUNCIONA EL RFID, AGREGAR RANGO DE ALCANCE**

****

* Módulo RTC

Este módulo posee un chip DS1302 con conexiones I2C que puede ser alimentado por una fuente de alimentación (Vcc) o por batería (Vbat) manteniendo dato de fecha y hora aunque el dispositivo se encuentre apagado, gracias a la batería, una vez configurada inicialmente se actualizará automáticamente. Posee un crystal de 32.768 KHz. Se alimenta. La alimentación debe mantenerse por encima de 3,3 V, si cae de dicha tensión, pasará a alimentarse de la batería Vbat para poder mantener fecha y hora.

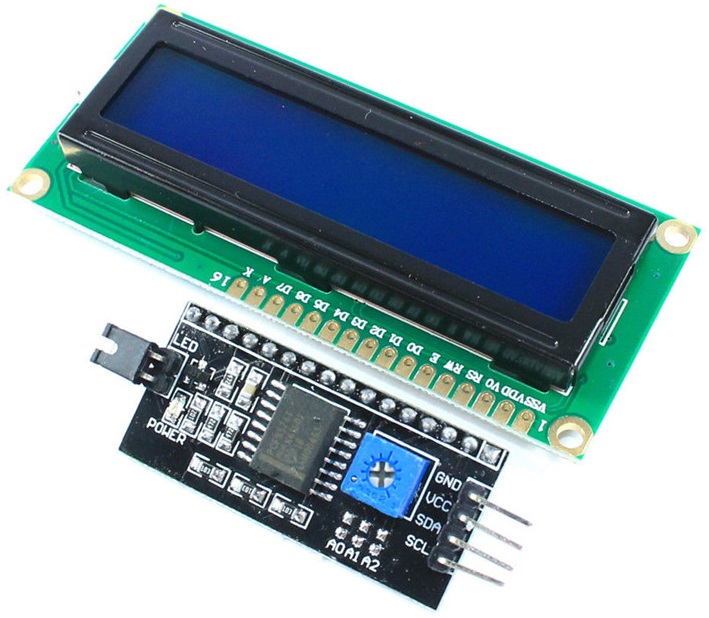


* Display LCD 16x2

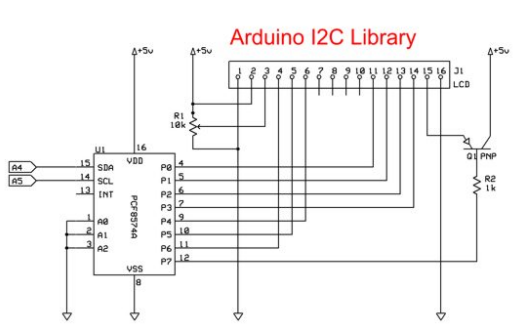
Las condiciones donde será utilizado el dispositivo y los usuarios que manejarán el mismo, requieren de una pantalla que simplemente muestre la información necesaria, en una época de pantallas táctiles se podría pensar que es la opción mas recomendable pero ya que las personas que trabajan en los feedlots no utilizan esa tecnología, lo mas recomendable es usar un display sencillo y fácil de entender y no integrar el teclado junto con la pantalla. Un display 16x2 permite mostrar toda la información que

* Adaptador I2C

Para poder disminuir la cantidad de pines necesarios para la comunicación se utiliza un adaptador I2C, el cual reduce la cantidad pines necesarios de 6 a 2 para comunicación, en la **figuraX** vemos tanto el display como el adaptador.

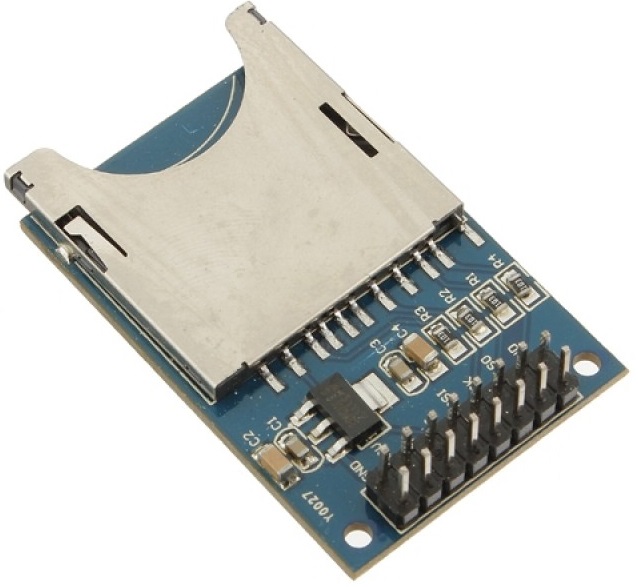


En la **figuraX** podemos ver la conexión del adaptador para transformar la comunicación paralela del display en comunicación serie I2C



* Módulo SD

Una vez recolectado el dato es necesario almacenarlo temporalmente hasta poder transferirlo al software, la memoria volátil del dispositivo no es una opción valida ya que es necesario mantener el dispositivo encendido, en el feedlot visitado, losmn corrales se encuentran a una distancia considerable de la oficina donde se se encuentra la computadora que posee el software. Se ah decidido utilizar una memoria SD extraible para almacenar los datos y posteriormente utilizar esta misma memoria para transferir los datos a la computadora,



* Teclado Matricial
* Gabinete

Debido al uso que se le dará al dispositivo consideramos que un gabinete de la forma de la **FIGURAX** es la mas adecuada para el usuario, ya que posee una forma cómoda para su agarre y para la distribución de los periféricos.



Tomando en cuenta las necesidades de nuestro proyecto elegimos el PLC **ZELIO SR2 B201BD** que cumple con todas las especificaciones.

Posee las siguientes características:

* Alimentación: 24V
* Número de E/S: 20
* Entradas analógicas: 6 de 0-10V
* Entradas digitales: 6
* Salidas a Relé: 8

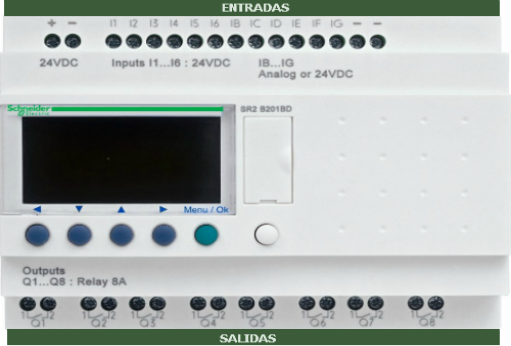


Ilustración 3 - PLC ZELIO SR2 B201BD

1. **SENSORES**

Un **sensor** es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación(temperatura, intensidad lumínica, etc.), y transformarlas en variables eléctricas.

Para el desarrollo del proyecto se utiliza diferentes sensores para medir magnitudes físicas.

**SENSOR DE TEMPERATURA**

Teniendo en cuenta que la temperatura del invernadero varía entre los 0 y 50, decidimos acondicionar el sensor de tal manera que cuando haya 50 grados a la salida del circuito se obtengan 10 voltios. Para ello, se utilizo un operacional en modo amplificador no inversor con una ganancia de 6.66.

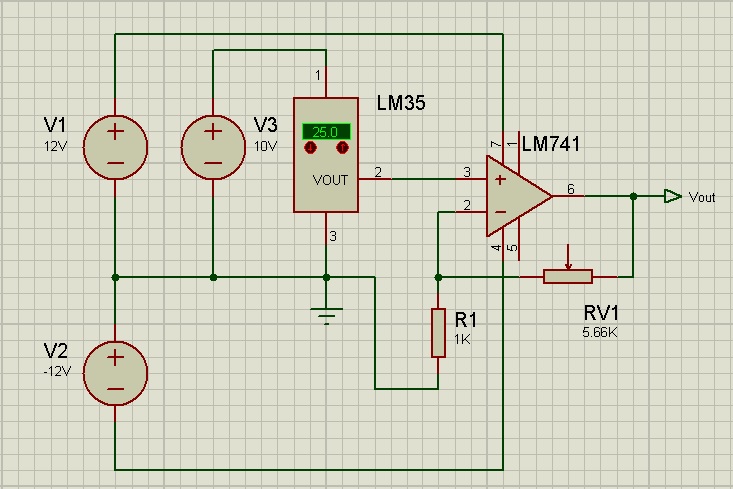


Ilustración 4 - SENSOR DE TEMPERATURA

Características del LM35:

Está calibrado directamente en grados Celsius.

La tensión de salida es proporcional a la temperatura.

Tiene una precisión garantizada de 0.5°C a 25°C.

Su rango de medición abarca desde -55°C hasta 150°C.

Opera entre 4 y 30 voltios de alimentación.

Baja impedancia de salida.

Baja corriente de alimentación (60uA).

Bajo costo.

Como el Lm35 trabaja en el orden de los mili volts y el PLC trabaja de 0v a 10v necesitamos amplificar la señal, por lo que se necesito utilizar un amplificar operación.

Amplificador no inversor:

**Vout** = **Vin** \* (**R1+R2/R1**)

Donde Vin es la salida del LM35

Vout es la salida que va al PLC.

R1=1k R2=5.66k **Ganancia**=6.66

Al tener esta salida se obtiene una salida de 10V cuando el sensor llega a registrar los 50 ºC y 0,5 V cuando es 0 ºC.

La conversión no es lineal sino en forma de

Alimentación:

Para alimentar el Lm35 se usa una fuente de 10v.

Para alimentar el Amplificador LM741CN se usa una fuente partida de +12V -12V.

**SENSOR DE LUMINOSIDAD**

Un LDR (Light Dependent Resistor) es una resistencia que varía su valor en función de la luz recibida, cuanta más luz recibe, menor es su resistencia.

Un LDR está fabricado con un semiconductor de alta resistencia como puede ser el sulfuro de cadmio. Si la luz que incide en el dispositivo es de alta frecuencia, los fotones son absorbidos por la elasticidad del semiconductor dando a los electrones la suficiente energía para saltar la banda de conducción. El electrón libre que resulta (y su hueco asociado) conduce electricidad, de tal modo que disminuye la resistencia.

El rango de resistencia que da un LDR desde la total oscuridad hasta la plena luz, varía de un modelo a otro, pero en general oscilan entre unos 50Ω a 1000Ω cuando están completamente iluminadas y entre 50KΩ y varios MΩ cuando está completamente a oscuras.

De acuerdo la necesidad lumínica que se necesita en el proyecto se utiliza un circuito divisor de tensiones donde la salida va desde 1 V a 10 V.

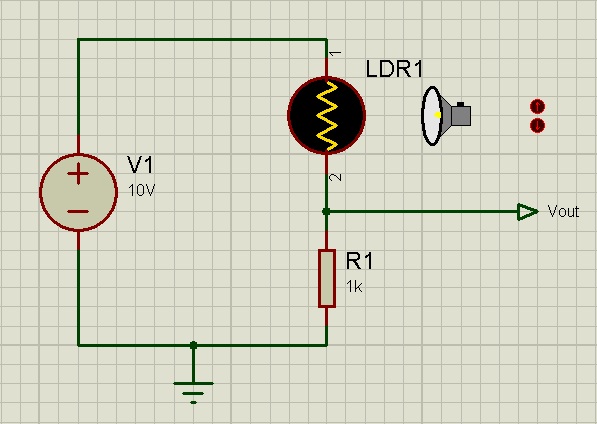


Ilustración 5 - SENSOR DE LUMINOSIDAD

Al utilizar la LDR de esta manera se obtiene la tensión máxima cuando esté completamente iluminado, ya que se comportará prácticamente como un circuito abierto, con una resistencia de 50Ω o 100Ω.

Funcionamiento:

* 0 Lux(en la oscuridad): Vout = 1V
* 20.000 Lux(Luz intermedia): Vout=6,3V
* 40.000 Lux (Plena Luz): Vout = 10V

**SENSOR DE HUMEDAD**

De acuerdo el porcentaje de humedad relativa necesaria para el invernadero el sensor de humedad elegido es el **HIH-4030/31** el cual nos permite conocer el porcentaje de humedad dependiendo del voltaje de salida.

Salida: 0.958V a 0% de humedad, hasta 4.028 Voltios a 100% de humedad relativa.

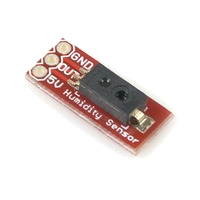


Ilustración 6 - Sensor de Humedad

Características de funcionamiento:

|  |  |  |  |
| --- | --- | --- | --- |
| Voltaje de salida | HR a 25°C | HR a 40°C | Variacion |
| 1.1115V | 5% | 5,2% | 4,0% |
| 1.265 V | 10% | 10,3% | 3,3% |
| 1.4185V | 15% | 15,5% | 3,3% |
| 1.572 V | 20% | 20,7% | 3,3% |
| 1.7255 V | 25% | 25,8% | 3,3% |
| 1.879V | 30% | 30,8% | 2,7% |
| 2.0325V | 35% | 36,1% | 3,3% |
| 2.186V | 40% | 41,3% | 3,3% |
| 2.3395 V | 45% | 46,5% | 3,3% |
| 2.493V | 50% | 51,6% | 3,3% |
| 2.6465V | 55% | 56,8% | 3,3% |
| 2.8V | 60% | 62,0% | 3,3% |
| 2.9535V | 65% | 67,1% | 3,3% |
| 3.107V | 70% | 72,3% | 3,3% |
| 3.2605V | 75% | 77,5% | 3,3% |
| 3.414V | 80% | 82,6% | 3,3% |
| 3.5675V | 85% | 87,8% | 3,3% |
| 3.5675V | 90% | 93,0% | 3,3% |
| 3.8745V | 95% | 98,1% | 3,3% |
| 4.028V | 100% | 103,3% | 1,6% |

1. **ACTUADORES**

Un **actuador** es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.

Los actuadores que cumplen con las necesidades para la automatización de nuestro invernadero capilla son:

* Bombillas Rojas
* Motor de Apertura y cierre de Portón
* Ventilador
* Calefactor diesel con encendido eléctrico
* Humidificador

**BOMBILLAS ROJAS**

Cuando el sensor de iluminación detecte que los lux’s necesarios para el crecimiento del tomate son mínimos o que no se cumplió la condición de 16 horas de luz constante, se encenderá las bombillas.

Se utilizará varias líneas en serie de bombillas rojos para la iluminación, ya que estas bombillas dan el 75% menos de luz que las blancos pero la radiación infrarroja y el calor que emiten son iguales al de los bombillos blancos.

En la siguiente grafica podemos observar de acuerdo la distancia los lux’s que emite un bombilla Roja.

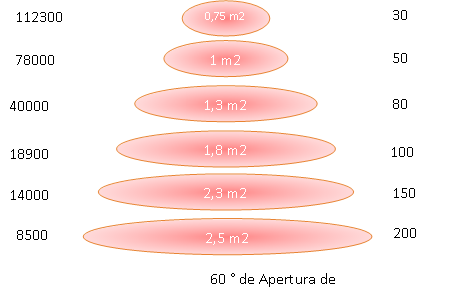


Ilustración 7 - Bombillas Rojas

**MOTOR DE APERTURA Y CIERRE DE PORTÓN**

El siguiente motor cumple con las características necesarias para la apertura de los portones ya que se toma que la estructura del mismo no superara los 350 KG.

Características:

* Motor de 1/5hp Monofásico 0,148 Kw 1740rpm (60hz)
* Reductora con corona de nylon
* Relación de Transmisión motor-reductora 25:1
* Piñón de salida Z18
* Capacidad de arrastre 350kg
* Velocidad de 15 m/min (60hz)
* Tiempo de apertura 11,5 seg (3 metros)
* Consumo x maniobra 0,0009 Kw/h

Ilustración 8 - Motor de Apertura y Cierre de Portón

**VENTILADOR**

Los ventiladores poseen la función de desplazaran aire de un lugar a otro, esta acción ayuda a mezclar el aire a lo largo de un invernadero para crear un ambiente más fresco y así bajar la temperatura.

Las características generales que debe poseer el ventilador para nuestro invernadero serán:

* Tamaño: 40 pulgadas
* Voltaje: 380v/220v
* Energía: 0.75kw
* Flujo de aire: 45000m3/h
* Velocidad giratoria: 560rpm

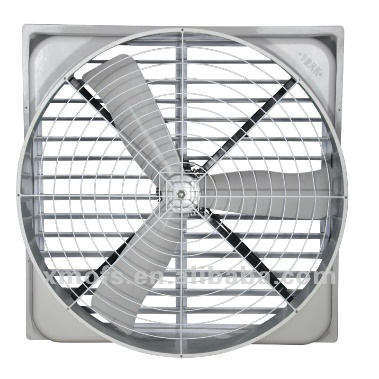


Ilustración 9 - Ventilador

**CALEFACTOR DIESEL CON ENCENDIDO ELÉCTRICO.**

Elegimos el calefactor diesel con encendido eléctrico ya que cumple con las necesidades de salida del PLC utilizado, y es de bajo costo de funcionamiento ya que los calefactores eléctricos poseen un alto consumo de energía por lo que generaría mucho costo.

Para calcular cuántos calefactores diesel necesitamos o de cuantas calorías debería ser nuestro calefactor, la resolvemos con la siguiente ecuación:

Alto(m) x Largo(m)x Ancho(m)x50= Calorías necesarias



Ilustración 10 - Calefactor Diesel con Encendido Eléctrico

**HUMIDIFICADOR**

Básicamente existen tres tipos de humidificadores de humedad relativa:

* Fríos (ultrasónicos)
* Calientes (electrodos)
* Calientes (evaporación

Para nuestro proyecto elegimos los humidificadores ultrasónicos, ya que producen una nebulización del agua a través de vibraciones de muy alta frecuencia, no afectan a la temperatura, son extremadamente seguros y silenciosos, con caudal regulable y muy bajo consumo.

Ilustración 11 - Humificador

**DESCRIPCIÓN DE LA AUTOMATIZACIÓN**

1. **PROGRAMACION DEL PLC**



Ilustración 12 - Programación del PLC

Para la programación del PLC se utilizo el leguaje BDF.

El BDF (Diagrama de bloques de funciones): Es un lenguaje gráfico amplio en posibilidades, que permite funciones SFC-Grafcet en la aplicación permitiendo modelar con maquina de estado finito.

El GRAFCET (GRAphe Functionel de Croissant Etape Transition) es un grafo o diagrama funcional normalizado, que permite hacer un modelo del proceso a automatizar, contemplando entradas, acciones a realizar, y los procesos intermedios que provocan estas acciones.

Para la programación del PLC se tuvo en cuenta las siguientes requisitos de entrada:

1. Tiempo de Histéresis de las variables
2. Comparación de lectura
3. Numero de sensores
4. Tiempo de Histéresis de las Variables:

Debido a un cambio brusco de alguna variable a medir (temperatura, humedad, luminosidad) por desperfecto del sensor o algún agente externo, definimos en la programación un timer que se acciona para comparar el valor medido con un promedio de 6 valores a medir al fin del timer, si el promedio no es próximo o igual al valor que detecto el cambio, no accionara al actuador, de lo contrario será accionado el actuador correspondiente.

1. Comparación de Lectura

Para poder tener una lectura homogénea de las variables se dispondrá varios sensores, permitiendo así poder tener un promedio de la medición solicitada y descartar lecturas erróneas que puede generar un solo sensor.

1. Numero de Sensores

Definimos que el número de sensores a colocar sea siempre impar, ya que esto nos permitirá tener una mejor medición si llegara a fallar algún sensor, por ejemplo si disponemos 3 sensores de temperatura y uno posee una falla tendremos la ventaja de comparar la temperatura con 2, descartar el que falla y detectar el sensor que dañado con mayor facilidad.

1. **COMUNICACIÓN DEL PLC CON LA PC**

La comunicación establecida entre el PLC y la Pc es la comunicación serie.

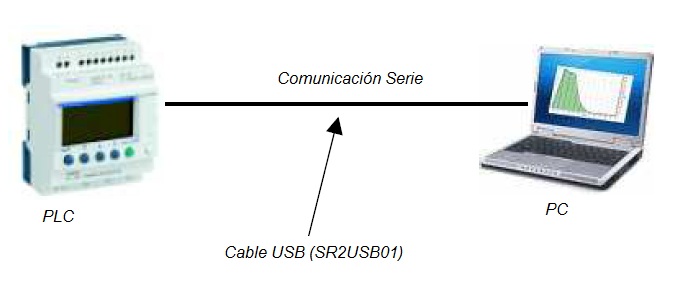


Ilustración 13 - Comunicación del PLC con la PC

**Parámetros**:

* + Bits por segundo: **115200**
  + Bits de Datos: **7**
  + Paridad: **Par**
  + Bits de parada: **1**
  + Control de Flujo: **Ninguno**

**TRAMA ESCRITURA**:

Delimitador de inicio: ":"

Dirección del esclavo: 0x01

Comando de escritura: 0x10

Datos: 0x00 00 FF XX

(XX es un número entre 0x00 y 0x17, corresponde a la dirección del dato a ser escrito menos 1)

Número de bytes: 0xnn

(Este es el número de datos a escribir, Cada valor se compone de dos bytes)

Datos a escribirse: 0xd1H D1L D2H ... dnnL

(Estos son bytes 0xnn para escribir)

Suma de comprobación (CheckSum): 0xcc

(es el complemento a 2 de toda la suma )

**La respuesta:**

· Delimitador de inicio: ":"

· Dirección del esclavo: 0x01

· Comando de escritura: 0x10

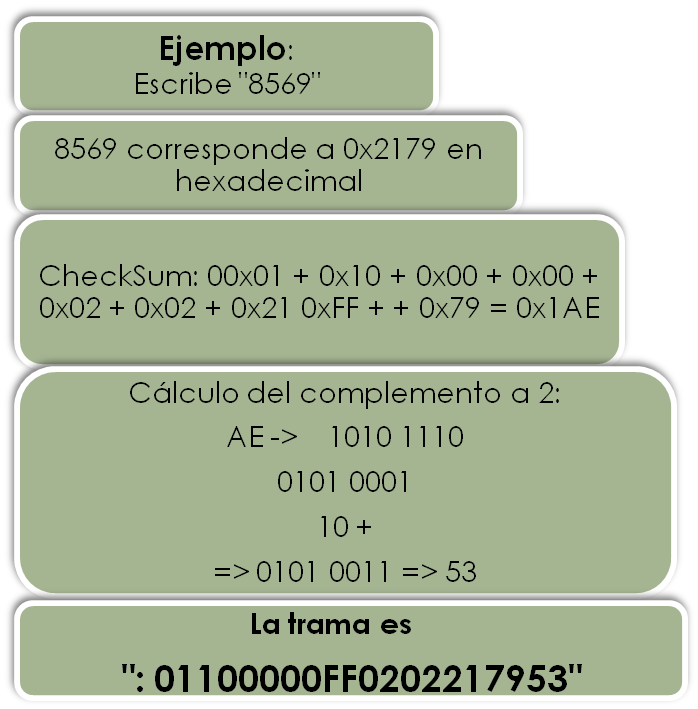
· Datos: 0x00 00 FF xx

· Número de bytes: 0xnn

· Suma de comprobación: 0xcc

Ejemplo:

Escribir un dato en el PLC. Todos están números están en hexadecimal.



**TRAMA LECTURA**:

· Delimitador de inicio: ":"

· Dirección del esclavo: 0x01

· Leer comando: 0x03

· Datos: 0x00 00 FF XX

(XX es un número comprendido entre 0x00 y 0x2F corresponde a la dirección de datos para ser leído menos 1) 19hexa = 25 decimal

· Número de bytes: 0xnn

(Este es el número de datos a leer.

10hex=16decimal  
· Suma de comprobación (CheckSum): 0xcc

(Es el complemento a 2 de la toda la suma)

**La respuesta del controlador:**

· Delimitador de inicio: ":"

· Dirección del esclavo: 0x01

· Leer comando: 0x03

· Número de bytes: 0xnn

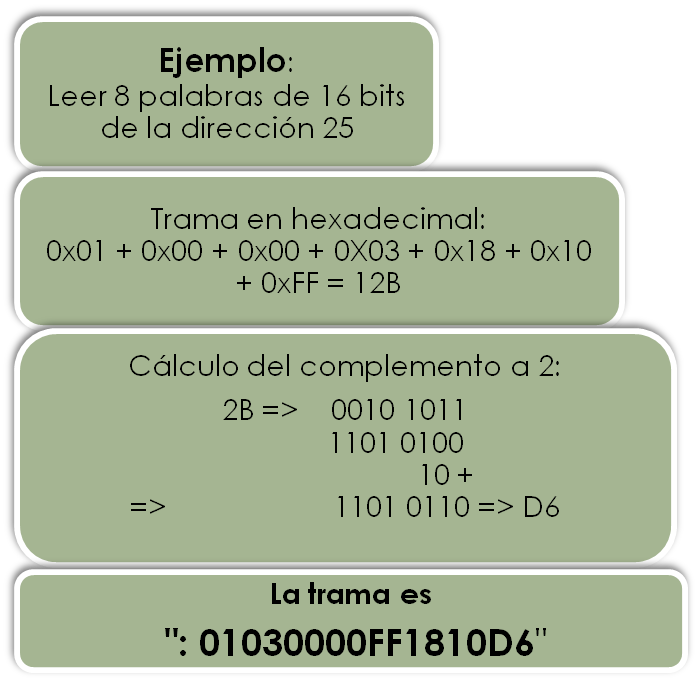
· Lectura de datos: 0xd1H D1L D2H ... dnnL

(Hay 0xnn bytes a leer)

· Suma de comprobación: 0xcc

(Es el complemento a 2 de la toda la suma)

Ejemplo para Leer los datos que envía el PLC.



Al mandar esta trama se obtiene los siguientes datos en el puerto de la PC.



Ilustración 14 - Datos Recibidos del PLC

Como se puede observar en el grafico a dependiendo que salida se active en el PLC nos envía los datos.

De esta manera cada vez que obtenemos una trama ya se sabe que está sucediendo en el PLC y de esta manera se puede utilizar dicha información.

**SISTEMA DE SOFTWARE**

1. **IDEA DE PRODUCTO**
   1. **IDENTIFICACIÓN DE USUARIOS PARTICIPANTES**

Se identificaron los siguientes usuarios:

* Usuario: es la persona que controla el Software “FeedLot”.
  1. **ESPECIFICACIÓN DE REQUISITOS**

**1.2.1- INTRODUCCIÓN**

*PROPÓSITO*

Esta especificación tiene como objetivo analizar y documentar en forma clara, precisa, completa y verificable las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar.

Para ello, se identificaran los requisitos que ha de satisfacer el nuevo sistema mediante el estudio de los problemas de las unidades afectadas y sus necesidades actuales.

El documento va dirigido tanto al equipo de desarrollo, como a los posibles usuarios finales.

El sistema será exclusivo para los productores de FeedLot, orientado a la gestión integral de eventos productivos y sanitarios, que permite administrar sistemas de trazabilidad animal, preparar e imprimir documentación para el Senasa, AFIP o para el usuario.

*ALCANCE DEL SISTEMA*

El nombre dispuesto para la herramienta a desarrollar es “FeedLot”

*DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS*

**Definiciones:**

No se han definido.

**Acrónimos:**

No se han definido.

**Abreviaturas:**

IEEE: Institute of Electrical and Electronics Engineers

ERS: Documento de Especificación de Requisitos de Software

JDBC: Java Database Conectivity

DBMS: Database Management System

DCE: Data Communication Equipment

*REFERENCIAS*

\* IEEE Recommended Practice for Software Requirements Specification. IEEE/ANSI std. 830, 1998.

*VISIÓN GENERAL DEL DOCUMENTO*

Este documento es una Especificación de Requisitos Software (ERS) que se ha estructurado basándose en las directrices dadas por el estándar IEEE/ANSI 830-1998, “IEEE Recommended Practice for Software Requirements Specifications”.

Este documento consta de tres secciones. Esta sección corresponde a la Introducción y proporciona una visión general de la ERS. En la siguiente sección se da una descripción general del sistema, con el fin de conocer las principales funciones que el mismo debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. Por último, en la sección de requisitos específicos se definen detalladamente los requisitos que debe satisfacer el sistema.

**1.2.2- DESCRIPCIÓN GENERAL**

Esta sección nos presenta una descripción general del sistema con el fin de conocer las funciones que debe soportar, los datos asociados, las restricciones impuestas y cualquier otro factor que pueda influir en la construcción del mismo.

*PERSPECTIVA DEL PRODUCTO*

Esta herramienta es un módulo de software que no posee dependencias con otros productos de software de mayor jerarquía para su funcionamiento. Se presupone que el sistema operativo sobre el cual se ejecutará la herramienta “FeedLot” es Windows 7 o superior, de otro modo la herramienta desarrollada no podrá funcionar. El DBMS que se utilizará es MySQL, y se debe a que es libre, robusto, de código abierto y de alta performance.

**Interfaz de Usuario:**

Las interfaces del sistema deben ser intuitivas, fáciles de usar y amigables, de manera que con poca capacitación los usuarios sean capaces de usar el sistema. Estas interfaces tendrán similitudes con interfaces de sistemas ya usadas para facilitar su adaptación a las mismas.

**Interfaz de Hardware**

La computadora destinada a ejecutar “FeedLot” debe contener un Procesador de 1GHz, 1GB de memoria RAM, 16GB de espacio en disco duro   
Tarjeta gráfica con soporte para DirectX 9 y 128MB de memoria, similar o superior. Debe contar con una impresora configurada e instalada, del tipo y marca que se prefiera.

**Interfaz de Software**

El sistema operativo debe ser Windows 7 o superior ya que es capaz de correr el sistema sin problemas.

*FUNCIONES DEL PRODUCTO*

A grandes rasgos, las funciones que debe llevar a cabo “FeedLot” se pueden describir de la siguiente manera:

* Recepción de animales (importar datos): Este proceso permitirá importar de forma masiva la información de ingresos, bajas, pesadas, diagnósticos de enfermedades, cambio de corrales, tratamientos, alimentación, etc.; de los animales del .
* Alimentación: El usuario podrá crear y administrar las raciones que se utilizan en el FeedLot.
* Sanidad: El usuario cargara individualmente o masivamente los tratamientos y enfermedades diagnosticadas en los animales del establecimiento.
* Estadísticas: Se permitirá al usuario ver historiales de sanidad, alimentación, origen, ubicación, etc.
* Reportes: El usuario podrá exportar o imprimir informes requeridos o evaluaciones de trazabilidad.

*CARACTERÍSTICAS DE LOS USUARIOS*

Los objetivos de esta tarea son identificar y calificar a los responsables de cada una de las unidades y a los principales usuarios implicados. En la organización se identificó un tipo de usuario:

**Usuario**: Puede ser cualquier persona capacitada ya que es un entorno amigable y de fácil comprensión.

*RESTRICCIONES*

Como restricción de lenguaje, se elige optar por un lenguaje de programación y un DBMS de tipo relacional que posean licencia de software libre para desarrollar y emplear “FeedLot”.

*SUPOSICIONES Y DEPENDENCIAS*

Se asume que todos los requisitos pedidos se encuentran en este documento y que son estables.

*REQUISITOS FUTUROS*

Proporcionar al usuario la posibilidad de acceder remotamente al software.

El ciclo de vida elegido para desarrollar el sistema es el de prototipado evolutivo, debiendo orientarse hacia el desarrollo de un sistema flexible que me permita incorporar nuevas funciones de manera sencilla, rápida y eficiente.

**1.2.3- REQUISITOS ESPECÍFICOS**

*INTERFACES EXTERNAS*

**Interfaz con el Usuario:**

Se utilizaron interfaces del tipo ventana debido a que son las más conocidas por parte de los usuarios.

**Interfaz de Hardware:**

Ratón, teclado estándar, impresora.

**Interfaz de Software:**

No se ha definido.

*REQUISITOS FUNCIONALES*

Las funciones que debe realizar el sistema son las que se detallan a continuación:

* Ingreso a la aplicación: El sistema debela tener un nombre por medio del cual permita su ingreso “FeedLot”
* Administración de Categorías: El usuario podrá gestionar las categorías de los animales del establecimiento, como ser terneros, novillos, etc.
* Administración de Proveedores: El usuario podrá gestionar los datos de los proveedores de ganado vacuno, como ser: datos personales y adicionales.
* Administración de Corrales: El usuario podrá gestionar los corrales dándole un nombre y asignándole una etapa.
* Administración de Raciones: El usuario podrá gestionar las raciones que se utilizaran en el FeedLot y registrar la alimentación.
* Administración de Vacunas: El usuario podrá gestionar las vacunas, dosis, etc. Para los seguimientos y tratamientos de los animales.
* Administración de Animales: El usuario podrá gestionar los ingresos y movimientos de los animales, como ser identificador, proveedor, corral, peso, etc.
* Importación: Proceso por el cual un usuario podrá realizar una carga masiva de los animales censados desde un archivo de texto.

*REQUISITOS DE RENDIMIENTO*

El tiempo de respuesta de la aplicación a cada función solicitada por el usuario no debe ser superior a los 10 seg. El tiempo de respuesta a los listados dependerá de la tecnología de impresión.

Se espera almacenar miles de registros en la base de datos.

*RESTRICCIONES DE DISEÑO*

**Lenguaje de Programación y Base de Datos:**

Como restricción de lenguaje, se elige optar por un lenguaje de programación y un DBMS que posean licencia de software libre para desarrollar FEEDLOT.

El DBMS debe ser relacional y se accederán a los mismos usando la tecnología que proveen los drivers JDBC para manejo de base de datos mediante un entorno de desarrollo Java.

**Política de Seguridad:**

La seguridad de los datos será establecida por el sistema gestor de base de datos que se emplee.

**Política de Borrado:**

No se ha definido.

**Política de Respaldo:**

Una copia completa de seguridad cada vez que el usuario lo requiera.

* 1. **ESTUDIO DE VIABILIDAD**

VIABILIDAD TÉCNICA:

Tanto el software como el Hardware tienen viabilidad.

VIABILIDAD OPERATIVA:

No se violan ninguna disposición legal.

1. **ESPECIFICACIÓN C**

**2.2- GUIONES Y ESCENARIOS**

**CONCLUSIÓN**

Este proyecto impulsado por una disposición nacional permitió conocer un mercado poco explotado por la informática, con muchas necesidades que se deben cubrir para poder mejorar su producción y reducir tiempos. Las personas involucradas en la actividad ganadera tienen desconocimiento de las posibilidades que les ofrece hoy en día la tecnología y queda en manos del Ingeniero acercarse a estas actividades y mejorar los procesos.

Se logró integrar varios módulos obteniendo un dispositivo final que cubría todas las necesidades del proyecto, con un resultante superior al que el usuario había solicitado en un inicio, de esta manera se pudo entregar mas información, de mejor forma y mas útil para el usuario final. La experiencia del usuario con el software resultó positiva y abrió puertas a futuras mejoras en el proceso y en el mismo software y hardware, la posibilidad de capacitaciones al personal da lugar a pensar mas allá de la simple necesidad y buscar una automatización del proceso casi completa.

**MEJORAS E INVESTIGACIONES FUTURAS**

Siendo este campo de estudio muy amplio, planteamos las mejoras que se podrían realizar:

1- Al mejorar la infraestructura de conexión de internet se puede pensar en un sistema de recolección de datos centralizado, permitiendo una muestra regional y no solamente de un criadero en particular.

2- Con capacitación se pueden implementar mejores tecnologías como pantallas touch o tecnologías bluetooth para transmisión de datos.

3- Si se logra mejorar la tecnología de pesaje a basculas electrónicas, se elimina la necesidad de un usuario colocando manualmente el peso del bovino y mejorando la fiabilidad del dato.

4- Mejora en la identificación, existen identificadores como el bolo ruminal y el chip subcutaneo, ya que la identificación por caravana se pueden salir.

5- Identificación de datos que pueden ser relevantes para el usuario que permitan mejorar el proceso.

6- Utilizacion de comunicación vía USB para la carga de datos en el sistema. Aunque el uso de tarjeta de memoria no representa una complicación para el usuario, la conexión del dispositivo con la computadora vía USB mejoraría la experiencia.