



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA



# Detección Automática de Mastitis Vacuna

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE  
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Fabian Vique, Henry Marichal

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS  
PARA LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO ELECTRICISTA.

## TUTOR

Dr. Ing. Leonardo Steinfeld . . . . . Universidad de la República

## TRIBUNAL

Mag. Ing. Julián Oreggioni . . . . . Universidad de la República

Dr. Ing. Nicolás Pérez . . . . . Universidad de la República

Dr. Ing. Pablo Cancela . . . . . Universidad de la República

Dr. Ing. Leonardo Steinfeld . . . . . Universidad de la República

Montevideo  
jueves 2 agosto, 2018

*Detección Automática de Mastitis Vacuna*, Fabian Vique, Henry Marichal.

Esta tesis fue preparada en L<sup>A</sup>T<sub>E</sub>X usando la clase iietesis (v1.1).

Contiene un total de 157 páginas.

Compilada el jueves 2 agosto, 2018.

<http://iie.fing.edu.uy/>

La ilustración del pueblo, es la verdadera locomotora del progreso.

JOSÉ PEDRO VARELA

Esta página ha sido intencionalmente dejada en blanco.

# Agradecimientos

En estas modestas líneas queremos expresar nuestro enorme agradecimiento a todas las personas que nos apoyaron en el transcurso de este proyecto.

Comenzando por nuestras familias que nos apoyaron durante nuestra formación hasta estos últimos tiempos. Y a nuestros amigos por acompañarnos siempre.

Agradecemos a nuestro tutor Leo que durante el proyecto nos guió y estuvo siempre que lo necesitamos. También a otros integrantes del IIE como Nicolás Pérez por aconsejarnos con componentes del circuito y prestarnos el analizador de impedancias, Mauricio González por ayudarnos con la fabricación del PCB y prestarnos la estación de soldar, Francisco Veirano por enseñarnos a utilizar la impresora 3D y Santiago Eizaguirre por prestarnos la termocupla.

Agradecemos Santiago de Izaguirre por dejarnos probar el dispositivo en su tambo, brindar datos de las vacas e informarnos sobre mastitis. También agradecemos a los ordeñadores por colaborar con la prueba de el dispositivo. Y a los tamberos de Florida que fueron entrevistados en busca de necesidades.

Por último agradecemos a personas de otras Facultades como la Dra. Sorondo de Facultad de Veterinaria por informarnos sobre mastitis. Y a el Qco. Mauricio Musso de Facultad de Química por brindarnos soluciones patrón para las pruebas de calibración y modelado de la celda.

Esta página ha sido intencionalmente dejada en blanco.

# Resumen

La rentabilidad de un tambo está directamente determinada por la cantidad y calidad de leche que produce. Por este motivo los tamberos están continuamente buscando su mejora de las mismas. La mastitis afecta directamente estos dos factores. Por ambos motivos es de interés tratar rápidamente el animal.

Este trabajo busca desarrollar un sistema accesible económicamente y automático que permita detectar mastitis en el tambo durante el ordeño. Para ello se desarrolla un dispositivo que se integra al organo de una ordeñadora estándar.

Luego de una cuidadosa revisión del estado del arte, el método de detección seleccionado para el dispositivo es la medición de conductividad eléctrica durante el ordeño de la vaca, utilizando una unidad de medición de conductividad por organo y una unidad central que tiene dos funciones, por un lado identifica las vacas con su número de caravana y por otro realiza el procesamiento de las medidas de conductividad tomadas por cada unidad de medición. En base a esta información se utilizan algoritmos de reconocimiento de patrones para estimar si tiene o no mastitis. El resultado se indica en leds en la unidad de medición.

El sistema propuesto tiene una precisión de detección de vacas con mastitis del 65 %, y un recall de 64 %. Estos resultados son prometedores ya que utilizando un sistema con un precio modesto (4000 USD para 20 organos) es posible dar alertas en el tambo.

Esta página ha sido intencionalmente dejada en blanco.



# Prefacio

La motivación del presente proyecto surge de aplicar los conocimientos adquiridos en la carrera a una problemática de los productores lecheros. Para esto se realizaron encuestas a los mismos. De las mismas, nos enteramos de la necesidad de los tamberos por una herramienta autónoma capaz de detectar mastitis vacuna en tiempo real. De esta manera pueden tratar rápidamente las vacas con mastitis y por ende mejorar la calidad de leche que producen.

Actualmente en Uruguay no existe una herramienta que satisfaga completamente esta necesidad, ya sea por motivos de escasa practicidad, tiempo de respuesta o relación costo-beneficio. Por lo que se busca desarrollar un dispositivo de detección completamente automático y con buena relación costo-beneficio.

A lo largo del proyecto se presentaron diversos obstáculos. Desde el abandono de un integrante que se mitiga recortando el alcance, hasta el requerimiento de robustez de la unidad de medición debido a los químicos que debe soportar durante el lavado de la ordeñadora. Este último obstáculo produjo un atraso de un mes en la planificación.

Fabián Vique, Henry Marichal

Esta página ha sido intencionalmente dejada en blanco.

# Tabla de contenidos

<b>Agradecimientos</b>	<b>III</b>
<b>Resumen</b>	<b>V</b>
<b>Prefacio</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Antecedentes . . . . .	2
1.3. Objetivo General . . . . .	6
1.4. Alcance . . . . .	6
1.5. Requerimientos Conceptuales . . . . .	7
1.5.1. Sistema . . . . .	7
1.5.2. Unidad de Medición (UM) . . . . .	7
1.5.3. Lectura de caravanas y Procesamiento de datos (LecPro) . . . . .	8
<b>2. Descripción General del Sistema</b>	<b>9</b>
2.1. Descripción del Tambo . . . . .	9
2.2. Diagrama de bloques . . . . .	10
<b>3. Hardware de UM</b>	<b>13</b>
3.1. Diagrama de bloques . . . . .	13
3.2. Transductores . . . . .	14
3.2.1. Temperatura . . . . .	15
3.2.2. Conductividad . . . . .	17
3.3. Microcontrolador . . . . .	18
3.4. Bluetooth . . . . .	18
3.5. Alimentación . . . . .	19
3.6. Implementación . . . . .	19
<b>4. Transductor de conductividad</b>	<b>27</b>
4.1. Introducción . . . . .	27
4.2. Modelado y diseño de celdas . . . . .	27
4.3. Diseño mecánico . . . . .	35
4.4. Diseño electrónico . . . . .	38
4.4.1. Diseño . . . . .	38

## Tabla de contenidos

4.4.2. Prueba . . . . .	49
4.5. Integración de la pieza en el organo . . . . .	51
4.6. Calibración . . . . .	57
<b>5. Software de UM</b>	<b>65</b>
5.1. Diseño de software . . . . .	65
5.2. Descripción de Módulos . . . . .	65
<b>6. Hardware y Software de LecPro</b>	<b>71</b>
6.1. Hardware de LecPro . . . . .	71
6.2. Diseño de Software . . . . .	72
6.2.1. Diagrama de Módulos . . . . .	72
6.3. Implementación de Software . . . . .	74
6.4. Clasificación . . . . .	87
<b>7. Prueba de Campo</b>	<b>95</b>
7.1. Instalación del detector en el tambo . . . . .	95
7.2. Problemas Surgidos . . . . .	96
<b>8. Conclusiones y trabajo a futuro</b>	<b>99</b>
<b>Apéndices</b>	<b>104</b>
<b>A. Plan Original de Proyecto</b>	<b>105</b>
<b>B. Transductor de Conductividad</b>	<b>127</b>
B.1. Prueba de embudo . . . . .	127
B.2. Circuito . . . . .	128
B.2.1. Oscilador . . . . .	128
B.2.2. Filtro pasa banda MFB . . . . .	129
B.2.3. Detector de Picos . . . . .	130
<b>C. Conclusiones</b>	<b>135</b>
C.1. Costos materiales de UM y LecPro . . . . .	135
C.2. Medición de volumen de leche . . . . .	135
<b>Referencias</b>	<b>137</b>
<b>Índice de tablas</b>	<b>139</b>
<b>Índice de figuras</b>	<b>141</b>

# Capítulo 1

## Introducción

### 1.1. Motivación

La rentabilidad de un tambo está directamente determinada por la cantidad y calidad de leche que produce. Por este motivo los tamberos están continuamente buscando la mejora de los mismos. Se entiende por leche de calidad [11] a la proveniente del ordeño de vacas sanas, bien alimentadas, libre de olores, sedimentos y sustancias extrañas. Además, debe reunir las siguientes características:

- Cantidad y calidad apropiada de los componentes sólidos (grasa, proteína, lactosa y minerales).
- Mínimo de carga microbiana.
- Libre de bacterias causantes de enfermedades (brucelosis, tuberculosis, patógenos de mastitis) y toxinas (sustancias tóxicas) producidas por bacterias u hongos.
- Libre de residuos químicos.
- Mínimo de células somáticas (leucocitos y células epiteliales).

Con el objetivo de mantener y conquistar nuevos mercados, las empresas lácteas han implementado penalizaciones para los tambos que no logren los niveles promedio permitidos de calidad de leche.

Mastitis bovina es el factor que provoca el mayor aumento de células somáticas. Esta enfermedad consiste en la inflamación de la glándula mamaria, la cuál provoca cambios en la composición bioquímica de la leche y en el tejido de la glándula. Por lo tanto, la mastitis bovina afecta a la calidad de leche producida ya que como se menciona anteriormente la leche de calidad posee un recuento de células somáticas bajo. Además, esta enfermedad provoca disminución de la cantidad de leche que produce el animal [14]. Dado que la mastitis afecta directamente a los dos índices (calidad y cantidad) que determinan la remuneración económica que recibe el productor, es clave tratar los animales infectados lo antes posible.

Existen tres tipos de mastitis [7]:

## Capítulo 1. Introducción

- **Mastitis subclínica:** Es la más difícil de detectar debido a que no se observa a simple vista cambios en la leche, ni en la ubre de la vaca. Además, es de larga duración y más frecuente que los otros tipos.
- **Mastitis clínica:** Se observa inflamación de la ubre afectada y el animal siente dolor al tocarlo. La leche se observa alterada con la presencia de descamaciones, coágulos, suero descolorido y a veces sangre.
- **Mastitis aguda:** Pone en riesgo la vida del animal. Se observan signos generalizados como fiebre y pérdida de apetito.

Por lo tanto, la producción lechera se beneficiaría enormemente si contara con un método eficiente y económico de detección temprana de la mastitis bovina.

### 1.2. Antecedentes

Los métodos de diagnóstico de mastitis pueden ser directos o indirectos. El método directo es el análisis bacteriológico, dirigido a detectar al microorganismo causante de la infección. Por otro lado, los métodos indirectos reflejan alteraciones en la composición de la leche debidas a la inflamación, midiendo distintos aspectos de la secuencia de cambios, que tienen lugar cuando la glándula esta infectada por un organismo patógeno. Los métodos indirectos, más utilizados en Uruguay son [6]:

- **Prueba de Wisconsin (RCS):** Consiste en enviar muestras a analizar en laboratorios para realizar el Recuento de las Células Somáticas(RCS). Para esto, es necesario que una persona tome las muestras de leche a cada vaca de manera manual y las envíe al laboratorio. Todo el proceso lleva aproximadamente una semana. Debido a esto, se considera que es una prueba lenta, laboriosa y costosa.
- **California Mastitis Test (CMT):** Es el método más utilizado para detectar mastitis subclínica en Uruguay. Esta prueba se realiza en el establecimiento tomando manualmente una muestra de leche por cuarto por vaca. Se requiere de un veterinario para realizarla dado que se necesita de alguien con experiencia y formación para hacerlo. La misma no proporciona un resultado numérico, si no más bien una indicación de si el recuento de células somáticas es elevado o bajo.
- **Conductividad Eléctrica (CE):** Esta prueba se basa en el aumento de conductividad eléctrica de la leche debido a su mayor contenido electrolítico especialmente a iones de sodio y de cloro. Actualmente existen productos manuales (medidores portátiles) y automáticos (equipos de ordeño computarizados) que detectan mastitis midiendo la conductividad de la leche. En la Figura 1.1 se observa un medidor manual en funcionamiento.



Figura 1.1: Dispositivo para determinación de la conductividad eléctrica de la leche. Imagen extraída de [6]

Se encuentran diversas publicaciones que comparan entre sí estos métodos. Cepero et al. [9] realizan pruebas comparativas entre CMT y CE. Para esto toman muestras de leche de vacas aparentemente sanas para la detección de mastitis subclínica. Se toma una muestra por cuarto<sup>1</sup>. Las muestras son analizadas con instrumental de laboratorio. Como conclusión se obtiene que hay una correlación entre ambos métodos sin embargo CE tiene como ventaja que es una técnica rápida.

Posteriormente, De la Cruz et al. [19] realizan una comparación de los tres métodos. Para esto, se toma un única muestra de leche por vaca. Esta es almacenada en un frasco de 100 ml para luego ser analizada en el laboratorio. Se concluye que estadísticamente existe correlación entre los tres métodos pero no es suficiente como para predecir RCS mediante regresión lineal a partir de CE. Se destaca que la CE tiene dificultad para discriminar los animales con problemas de mastitis subclínica, especialmente en los estados primarios de la infección. Además, no se encuentra correlación entre la mastitis y el número de partos y los meses de lactancia de los animales analizados.

Más recientemente, da Costa et al. [10] también comparan los tres métodos. Para medir CE se emplea un medidor manual. Además, se mide la conductividad en cada cuarto de la ubre ya que consideran que existe una correlación entre la mastitis y la diferencia de CE entre cuartos. Se toma una muestra de leche de cada cuarto de la ubre al inicio del ordeño. Concluyen que la medición de CE con un medidor portable presenta una baja correlación con RCS y CMT. La diferencia de resultados con [9] puede deberse a que en esta publicación se utiliza instrumental de laboratorio y no se analizan vacas con mastitis clínica.

Whyte et al. [13] investigan cuatro métodos de detección. Se encuentra una relación entre el ácido láctico y algunas proteínas de fase aguda con la mastitis. Además, se presenta a RCS como el método directo más adecuado para detectar mastitis. Se menciona CE pero aclaran que hay muchos factores que inciden sobre la misma, por lo que una variación de la misma no necesariamente debe estar rela-

<sup>1</sup>Cuarto de una ubre: una de las cuatro cuartos de la ubre

## Capítulo 1. Introducción

cionada con una infección. Sin embargo, puede ser utilizada para detectar mastitis si la CE de la leche de todos los cuartos son medidos durante el ordeño, y estos datos son comparados con los datos previos (obtenidos durante ordeños anteriores) de conductividad por cuarto de la ubre de la vaca.

Por otro lado, Elizalde et al. [14] comparan entre medir CE por cuarto o medir CE de la mezcla. Se obtiene mejor desempeño cuando se mide CE de cada cuarto. La CE de la mezcla de la leche se muestra ineficiente para discriminar los casos de mastitis. Se aclara que aunque el porcentaje de detección de CE (midiendo por cuarto) no es muy elevado, la técnica resulta eficaz para identificar precozmente animales con mastitis ya que es una técnica diagnóstica de bajo costo y resultado inmediato.

De estas publicaciones se puede concluir que la medición de la CE de la leche de cada cuarto puede utilizarse como un método de detección de mastitis. A continuación se presentan de manera breve publicaciones que hacen foco en la construcción del sensor de CE para detectar mastitis de manera automática.

Ferrero et al. [16] proponen un circuito de bajo costo basado en la medición de corriente alterna de la CE de la leche. Se utiliza un circuito acondicionador para minimizar los efectos parásitos y para que la salida analógica del sensor sea proporcional a la CE. Además, se diseña una celda para la medición de CE de la leche, compuesta por dos electrodos y un cable coaxial. Se mide la temperatura de la leche para compensar sus efectos sobre la CE. Esta compensación es realizada por software. La salida analógica del sensor es adquirida por un microcontrolador cada un segundo. Se considera que una vaca tiene mastitis si la varianza de CE de la serie temporal de todo el ordeño de la vaca y la varianza entre cuartos de la misma es superior a cierto umbral.

Ramírez et al. [22] presentan un circuito electrónico para medir CE en líquidos donde la compensación de temperatura se realiza por hardware. La salida analógica es proporcional a la CE.

Seguidamente se presentan las publicaciones que se enfocan en el procesamiento de los datos de CE.

Biggadike et al. [18] proponen utilizar el historial de EC por cuarto de la ubre de los últimos 14 ordeños. Se establece una alerta de un posible caso de mastitis si se supera cierto umbral determinado por la media del historial de EC. Los resultados que obtienen son sensibles al umbral utilizado. Concluyen que el método no es lo suficientemente preciso para ser utilizado como el único criterio para detectar mastitis.

Posteriormente, Norberg et al. [15] investigan varias características extraídas de CE. Se mide la CE de cada cuarto de la ubre de la vaca a intervalos de 2 segundos. Se calcula la media entre las 20 máximas medidas de CE válidas ( $X_{20}$ ) y la variación de todas las medidas de CE válidas dentro de un ordeño ( $\sigma_{EC}^2$ ) para cada cuarto de la ubre (8 indicadores, 2 por cuarto). A partir de estos se definen las siguientes cuatro características:

- $Max\_X_{20}$ : máximo  $X_{20}$  del ordeño.
- $Max\_σ_{EC}^2$ : máximo  $σ_{EC}^2$  del ordeño.



- $IQR\_X_{20}$ : cociente entre el máximo y mínimo  $X_{20}$  del ordeño.
- $IQR\_σ_{EC}^2$ : cociente entre el máximo y mínimo  $σ_{EC}^2$  del ordeño.

Se menciona que todas estas características se incrementan significativamente cuando las vacas están clínicamente o subclínicamente infectadas. Se utiliza una simple prueba de umbral y un análisis de función discriminante para validar la habilidad de estas características de distinguir entre vacas en diferentes grupos de salud. Las características que reflejan el nivel (en vez de la variación), y en particular el IQR, tuvieron un mejor desempeño para clasificar vacas correctamente. Usando esta característica (IQR), 80,6 % de los casos clínicos y 45 % de los casos subclínicos fueron correctamente clasificados.

Por otro lado, Cavero et al. [12] estudian el potencial de detectar mastitis en un sistema automático de ordeño usando información de la evolución temporal (días) de la CE de la mezcla de leche. Se mide una muestra de CE por vaca/ordeño. Como método patrón se utiliza el RCS donde se utiliza un límite variable. Primero se fija este límite en 400.000 células/ml y después en 100.000 células/ml. Si se está por encima del límite la muestra se etiqueta como infectada. Se estima el próximo valor de CE utilizando tres métodos distintos que se basan en el historial de muestras. Se da alerta para mastitis cuando la desviación relativa entre el valor medido y el valor estimado excede un umbral dado. Se obtiene una sensibilidad y especificidad por encima del 70 % pero una alta tasa de error (alrededor del 70 %)².

Más recientemente, Lien et al. [20] tienen como objetivo desarrollar un detector online de mastitis subclínica midiendo la CE de la leche por cuarto. Para etiquetar las muestras³ se realizan análisis de RCS. Si se obtiene un resultado menor a 200,000 células/ml, se etiqueta a la muestra como sana. Se define la media móvil (1.1) y el ratio móvil (1.2) :

$$CE_{t1-t2} = \sum_{t1}^{t2} CE_t / (t_2 - t_1) \tag{1.1}$$

$$CER_{t1-t2} = \sum_{t1}^{t2} CER_t / (t_2 - t_1) \quad \text{donde} \quad CER_t = \max(CE_t) / \min(CE_t)^4 \tag{1.2}$$

Se extraen características a partir de las series temporales de CE utilizando diversas combinaciones de 1.1 y 1.2. En cuanto a los resultados que presentan, obtienen una especificidad de 83,7 % para los cuartos sanos y una sensibilidad del 46,2 % para los cuartos infectados⁵. Se concluye que si bien los resultados obtenidos son más bajos que los que obtienen los detectores comerciales (RCS), este sistema permite dar

---

²En el capítulo Hardware y Software de LecPro, se define especificidad , sensibilidad y tasa de error.

³Una muestra es la serie de CE de una vaca

⁴Se tiene cuatro medidas de CE por unidad de tiempo (una por cuarto).

⁵En capítulo clasificación se definen los indicadores acierto de predicción para cuartos infectados y no infectados.

## Capítulo 1. Introducción

alertas en el tambo. Además, es barato y no utiliza el historial (series temporales de otros ordeños de la misma vaca) de CE de cada vaca.

En conclusión, existen esfuerzos para obtener un método de detección de mastitis utilizando la CE y diversas técnicas para determinar si una vaca tiene mastitis o no. Sin embargo, algunos tienen un costo operativo significativo ya que se requiere medir manualmente la CE de cada vaca (medidor portátil) mientras que hay proveedores de máquinas de ordeño que integran la funcionalidad de medir la CE pero solo para su producto (sin aclarar si miden la CE por cuarto o de la mezcla). Por lo tanto, es de interés desarrollar un sistema que se pueda integrar a las máquinas de ordeño de los productores lecheros sin que deban realizarse grandes cambios.

### 1.3. Objetivo General

Crear un dispositivo que detecte mastitis en cada vaca de manera automática. Esto permitirá al tambero tomar acción inmediata para controlar la mastitis, afectando de la menor forma posible su ingreso económico.

### 1.4. Alcance

Teniendo en cuenta el objetivo general del proyecto y los antecedentes se define lo siguiente<sup>6</sup>:

- Se realiza un dispositivo que detecte mastitis midiendo la CE de cada cuarto de la ubre de vaca de forma automática sin utilizar el historial de muestras.
- En lo que a software refiere, en lo posible, se utilizan librerías ya implementadas.
- No se implementará seguridad de datos.
- Se realizan las protecciones mecánicas necesarias para que soporte el funcionamiento en el tambo.
- Se realiza cableado para la alimentación de la red eléctrica.
- Se fabrica un prototipo para un solo organo que se prueba en la sala de ordeño, aunque el diseño del sistema contempla todos los organos de la sala.
- La identificación de las vacas al ingresar al tambo se realiza de forma manual<sup>7</sup>. Se requiere la identificación porque para el procesamiento de los datos puede ser pertinente utilizar otras características además de las extraídas a partir de la CE.

---

<sup>6</sup>El alcance y los requerimientos funcionales fueron recortados principalmente por el abandono de un integrante del equipo. Se adjunto plan original del proyecto en el Apéndice A.

<sup>7</sup>Las vacas cuentan con un identificador numérico único que se coloca en su oreja denominado caravana.

## 1.5. Requerimientos Conceptuales

### 1.5.1. Sistema

El sistema estará compuesto por una Unidad de Medición (UM) en cada organo y por una Unidad de Lectura de caravanas y Procesamiento de datos (LecPro).

### 1.5.2. Unidad de Medición (UM)

#### **Ubicación**

- Debe estar ubicada en el organo de cada puesto de la máquina de ordeño.

#### **Requerimientos mecánicos**

- Debe soportar las patadas de las vacas y golpes que se provocan al salirse el organo de la vaca.
- Debe ser impermeable .

#### **Requerimientos de Interfaces**

- Interfaz inalámbrica con el LecPro.
- Leds para identificar los resultados de la detección al usuario.

#### **Requerimientos Regulatorios**

- No contaminar la leche.
- La comunicación con el LecPro debe ser en una banda ISM.

#### **Requerimientos funcionales**

- El dispositivo debe muestrear la conductividad de la leche durante todo el ordeño
- Enviar datos al LecPro para el procesamiento.
- Debe realizar la acción correspondiente a un comando por parte del LecPro de mastitis o no mastitis en menos de un segundo.
- Debe medir CE en un rango de 1.0 mS/cm hasta 12.0 mS/cm.
- Este dispositivo debe poder integrarse a la ordeñadora <sup>8</sup> sin alterar su funcionamiento normal.

---

<sup>8</sup>Sistema completo de Ordeño

## Capítulo 1. Introducción

### 1.5.3. Lectura de caravanas y Procesamiento de datos (LecPro)

#### **Ubicación**

- Dentro del rango de alcance permitido por la tecnología inalámbrica para comunicarse con la UM.

#### **Requerimientos mecánicos**

- Debe ser impermeable.

#### **Requerimientos de interfaces**

- Comunicación inalámbrica con la UM.
- Comunicación para extracción de datos.

#### **Requerimientos regulatorios**

- La comunicación con el UM debe ser en una banda ISM.

#### **Requerimientos funcionales**

- Identificar vaca.
- Recibir medidas realizadas por la UM.
- Realizar procesamiento de datos.
- Enviar resultado a la UM en un tiempo menor a dos segundos después que se detecta fin de ordeño de vaca.

## Capítulo 2

# Descripción General del Sistema

### 2.1. Descripción del Tambo

Se presenta una breve descripción del funcionamiento del tambo y sus partes principales.

En la Figura 2.1 se observa un dibujo de la sala de ordeño del tambo. Las vacas ingresan a dicha sala por los corrales laterales (1). Cada vaca tiene acceso a un comedero (2), en el cual tiene ración para su alimentación. En muchos establecimientos, es el ordeñador (3) quien se encarga de ordeñar y racionar los comederos.

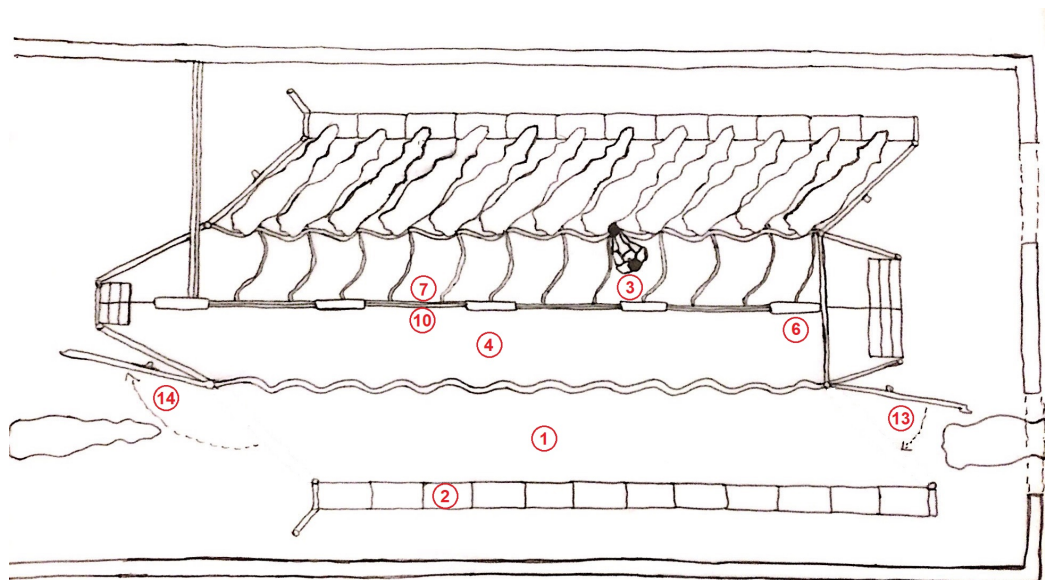


Figura 2.1: Sala de ordeño

Durante la mayor parte del tiempo de ordeño, el ordeñador se encuentra en la fosa (4). Esta se ubica en el centro de la sala de ordeño, a un nivel de aproxima-

## Capítulo 2. Descripción General del Sistema

damente un metro por debajo de los corrales donde se encuentran las vacas. Este desnivel facilita al ordeñador la colocación del organo (5) en la ubre de la vaca, tal cual se puede observar en la Figura 2.2. La fosa cuenta con luminarias (6).

Por el centro de la fosa, a una altura lo suficientemente elevada para no molestar al ordeñador, se ubica el caño central de leche (7) (ver Figura 2.1). Por este transita el flujo de leche de todos los organos hacia el tanque de frío. En dicho tanque se almacena la leche hasta que el camión cisterna la retira.

En la Figura 2.2 se aprecia el caño (8) por donde circula la leche desde el organo al caño central (7). Los caños de vacío (9) que contienen aire a baja presión para el sistema de extracción de leche de la ubre van conectados desde el organo a los caños principales de vacío (10). Por otro lado, se tiene un dispositivo (11) que retira el organo de la vaca automáticamente cuando se detiene el flujo de leche. Los cables de alimentación de este dispositivo vienen dentro del ducto rectangular (12).

A continuación se explica el funcionamiento normal de ordeño en un tambo. Primero, las vacas ingresan por ambas puertas laterales a la sala (ver Figura 2.1). La sala puede contener por turno el doble de vacas que de organos. Una vez que se completan ambos corrales de la sala, se colocan los organos en las ubres de las vacas de un lateral (supongamos en el lateral de abajo en la Figura 2.1). A medida que van terminándose de ordeñar las vacas, se van colocando los organos en las vacas del lateral de arriba.

Cuando todas las vacas del lateral de abajo terminan de ordeñarse, se abre la portera de salida (13). Las vacas salen de la sala. Se cierra la portera de salida, y se abre la de entrada (14) para que ingresen otras vacas. Se cierra la portera de entrada. Y nuevamente, a medida que las vacas del lateral de arriba van terminando de ordeñarse, se van colocando los organos en las vacas del lateral de abajo. Este proceso se repite, hasta que se ordeñan todas las vacas.

### 2.2. Diagrama de bloques

En la Figura 2.3 se observa un diagrama de bloques del sistema. El mismo está compuesto por el LecPro y las UM. La cantidad de UM está determinada por la cantidad de organos que tiene el tambo (una por organo).

En la Figura 2.4 se observa el diagrama del tambo con el sistema implementado. Como se mencionó en el alcance, solo se fabrica una UM. Debido a la distancia que se tiene al acceso de la red eléctrica es necesario utilizar un alargue para alimentar con energía eléctrica el sistema. Para poder extraer los datos del LecPro a una computadora se utiliza un cable Ethernet. En el futuro, se podrá hacer de manera inalámbrica por ejemplo Wifi o utilizando la red celular, dependiendo de las posibilidades del tambo en particular.

## 2.2. Diagrama de bloques

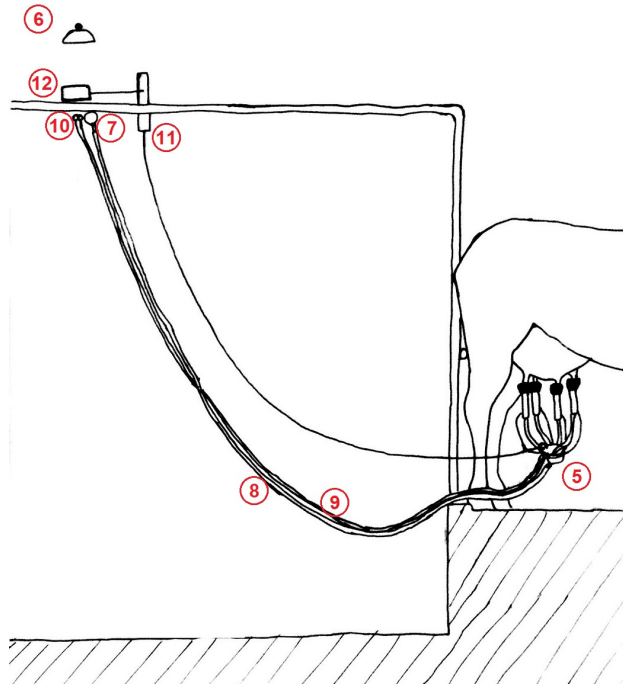


Figura 2.2: Fosa. Se observa un organo en funcionamiento. En el tambo en el cual se prueba el sistema se tienen 23.

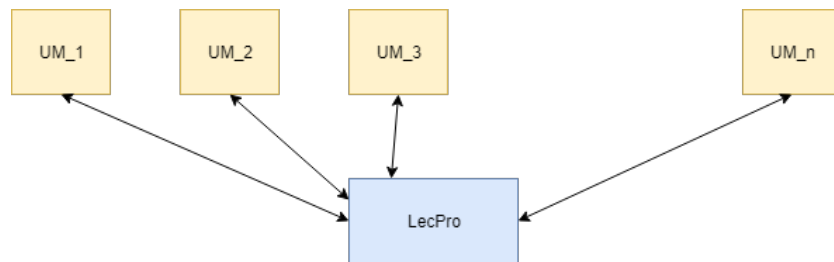


Figura 2.3: Diagrama de bloques del sistema completo

## Capítulo 2. Descripción General del Sistema

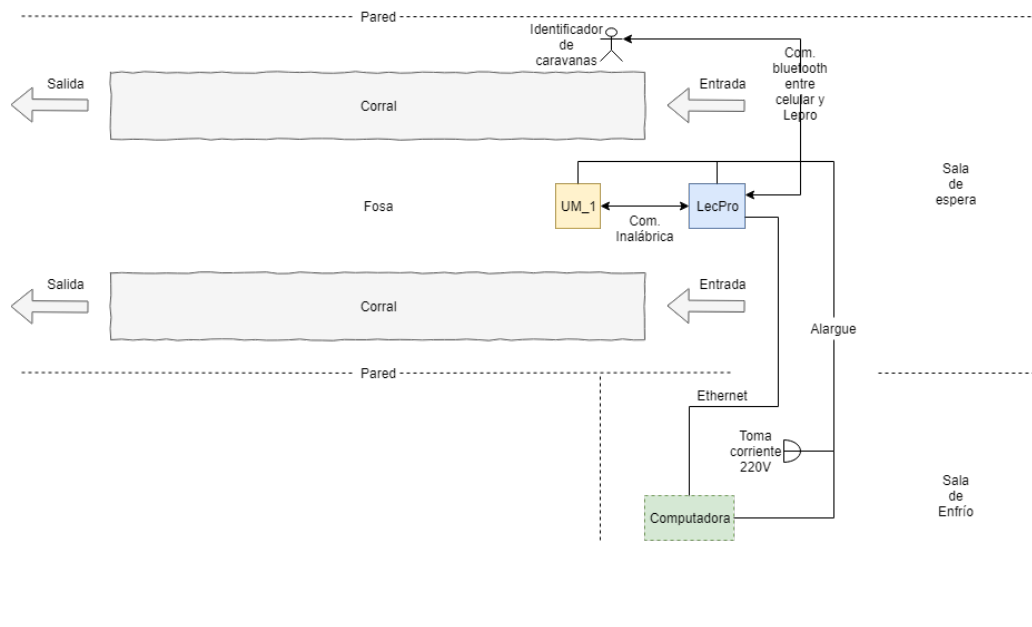


Figura 2.4: Diagrama del tambo con el sistema implementado



# Capítulo 3

## Hardware de UM

En este capítulo se explica cuáles son los componentes que conforman la UM. Como se mencionó anteriormente, la función principal de la misma es muestrear la conductividad eléctrica de la leche de cada cuarto y enviar estas muestras al LecPro. Para el muestreo de la CE, se diseña un transductor de conductividad. Por otra parte, se utiliza un transductor de temperatura para adquirir la temperatura de la mezcla de leche ya que es necesario realizar una compensación de las muestras de CE. Esto se debe a que la CE depende linealmente de la temperatura<sup>1</sup>. Esta compensación es realizada por software. Las muestras de CE y temperatura son enviadas al LecPro inalámbricamente utilizando la tecnología Bluetooth 2.0.

### 3.1. Diagrama de bloques

En la Figura 3.1 se observa el diagrama de bloques de hardware de la UM. El transductor de conductividad diseñado contiene cuatro celdas que son recintos donde cae la leche de cada cuarto para ser realizada la medida de conductividad. El circuito de medición es el encargado de realizar las medidas de conductividad, temperatura y convertir esos valores a un voltaje DC para que los pueda leer el microcontrolador.

El microcontrolador controla el circuito de medición. Es quién decide cuándo y cuál medida tomar. También decide cuándo se envían los datos por Bluetooth al LecPro.

Las celdas y el transductor de temperatura se ubican en el interior del organo. En cambio el circuito de medición, el microcontrolador, el módulo de hardware Bluetooth y la alimentación se ubican en una zona segura (aferrando al ducto rectangular que contiene los cables del dispositivo que saca el organo) lejos del organo. Esto se debe a que no es posible colocar estos componentes junto al organo ya que recibe patadas de las vacas y a que debe ser fácilmente maniobrable por el ordeñador. Además, no es posible colocarlos en el interior del mismo ya que el espacio es muy reducido.

---

<sup>1</sup>En la sección 4.2 se explica en detalle.

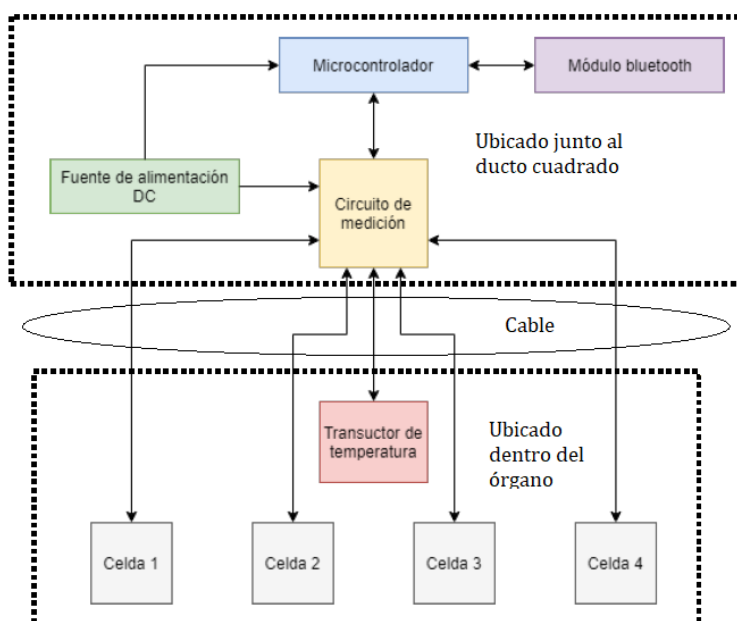


Figura 3.1: Diagrama del sistema completo en el tambo

Por lo tanto se necesita un cable para conectar el circuito con los componentes que están en el órgano (celdas y el transductor de temperatura).

El cable se aferra a los caños de vacío que van desde órgano hasta los caños principales de vacío. El largo de los caños es de  $2,5m$ . Por lo tanto con un cable de  $3m$  es suficiente.

Se necesita que el cable tenga 8 conductores como mínimo (4 para un borne de cada celda, uno común para un borne de todas las celdas y 3 para el transductor de temperatura, tal como se verá más adelante). Por lo que se selecciona un cable que cuenta con 8 conductores de  $0,25mm^2$  más un conductor de tierra y una malla de aluminio que protege los conductores de campos electromagnéticos externos.

La alimentación se encarga de alimentar el circuito y el microcontrolador. Más adelante se detalla el diseño de acuerdo a los requerimientos de los bloques.

## 3.2. Transductores

Para la medición de conductividad y temperatura se necesitan transductores. En el trabajo de Ferrero et al. [16] se utiliza un transductor de temperatura LM35 que tiene una precisión de  $0,5^{\circ}C$ . Entonces se opta por utilizar el mismo ya que esta precisión es suficiente por que la dependencia de la conductividad con la temperatura es baja ( $2,4\% ^2$  por grado celcius). Se diseña un transductor similar al diseñado en [16] para la medición de conductividad. A diferencia la geometría de los electrodos que se utiliza es rectangular en vez de cilíndrica para facilitar la construcción.

<sup>2</sup>Se calcula en la sección 4.2.

### 3.2.1. Temperatura

En la Figura 3.2 se muestra una foto del transductor LM35. Es un circuito integrado que necesita ser alimentado en el rango 4-30V. La salida es una tensión analógica proporcional a la temperatura. La misma se incrementa en 10mV por grado celsius (la salida es 0 mV cuando la temperatura es 0 °C). Además su rango de medición va de -55°C a 150°C y como se menciona anteriormente tiene una precisión de 0.5°C a 25 °C [5].

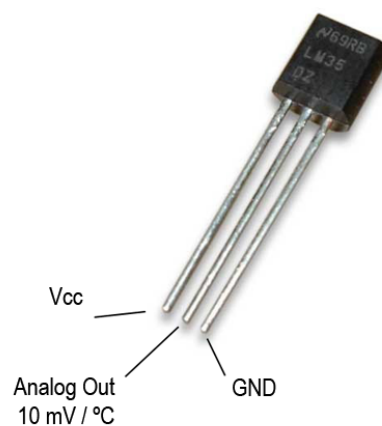


Figura 3.2: Transductor LM35. Imagen extraída de <https://www.internetdelascosas.cl/wp-content/uploads/2012/05/arduino-LM35-sensor-pines-278x300.png>

#### Caracterización del transductor de temperatura

Si bien el LM35 según sus hojas de datos está calibrado, se verifica la exactitud y linealidad comparándolo con una termocupla.

La termocupla se conecta a un multímetro para observar en el display la temperatura. La termocupla es Extech tipo K, el multímetro es Extech MN35. El sistema termocupla-multímetro tiene una exactitud de 0,5°C.

Se sumerge ambos en agua variando la temperatura entre 27 y 36 °C. Este rango comprende la temperatura a la que sale la leche de la vaca (alrededor de 33 °C). En la Tabla 3.1 se presentan los resultados.

### Capítulo 3. Hardware de UM

Tabla 3.1: Comparación de LM35 con termocupla

Termocupla ( $^{\circ}C$ )	LM35 ( $^{\circ}C$ )	Diferencia
36.2	36.0	0.2
35.1	34.9	0.2
34.0	34.2	0.2
32.7	32.9	0.2
32.1	32.0	0.1
31.0	30.9	0.1
30.0	29.9	0.1
29.0	29.0	0.0
28.0	28.0	0.0
27.0	26.9	0.1

La diferencia máxima entre medidas es de  $0.2^{\circ}C$ , lo que está comprendido dentro del margen de incertidumbre de la termocupla.

#### Amplificador de LM35

Para tener menor error de cuantización de la entrada analógica del microcontrolador se agrega una etapa amplificadora de ganancia del orden de  $10V/V$  al LM35. Esta etapa se realiza con un amplificador operacional en configuración no inversora como se muestra en la Figura 3.3. Donde  $V_t$  es la salida del LM35 y  $V_{A0}$  es una entrada analógica de el microcontrolador.

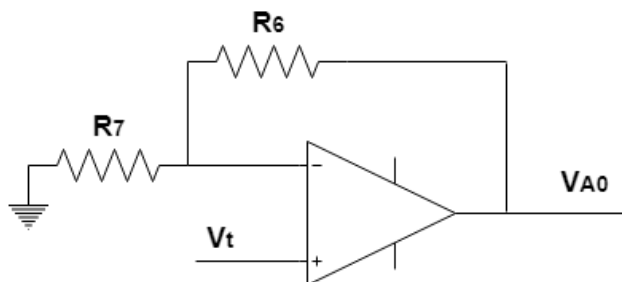


Figura 3.3: Amplificador de LM35

La temperatura a la que la leche sale de la vaca ronda los  $33^{\circ}C$ , entonces  $V_t = 330mV$ . Se elije  $R_6 = 100k\Omega$  y  $R_7 = 12k\Omega$  por lo tanto la ganancia es  $9,33V/V$ . Entonces a  $33^{\circ}C$  se tiene  $V_{A0} = 3,08V$ .

Tabla 3.2: Componentes seleccionados

$R_6(\Omega)$	$R_7(\Omega)$
100k	12k



### 3.3. Microcontrolador

Debido a la disponibilidad de variedad de librerías, se decide utilizar Arduino. El Arduino UNO cuenta con el microcontrolador ATmega328p. Se necesitan sólo dos entradas analógicas y tres pines digitales por lo tanto con este micro es suficiente ya que cuenta con 6 entradas analógicas y 14 pines digitales.

Arduino se puede alimentar entre 7V y 12V.

### 3.4. Bluetooth

Se selecciona el módulo bluetooth HC-05 ya que es muy utilizado con Arduino y tiene entre 5 y 10m de alcance, por lo que es suficiente para este prototipo.

El HC-05 funciona con lógica de 3.3V. Por lo que es necesario adaptar la tensión. La solución sencilla y operacional que se encuentra es usar un divisor para bajar de 5 V a 3.3 V para la señal Rx (salida del Arduino y entrada del HC-05) y para Tx (entrada del Arduino y salida del HC-05) conectarlo directamente ya que una tensión de 3.3V es suficiente para que el Arduino lo tome como nivel alto.

En la Figura 3.5 se muestra el conexionado. El HC-05 se puede alimentar con 5V por lo tanto se conecta directo a la Arduino.

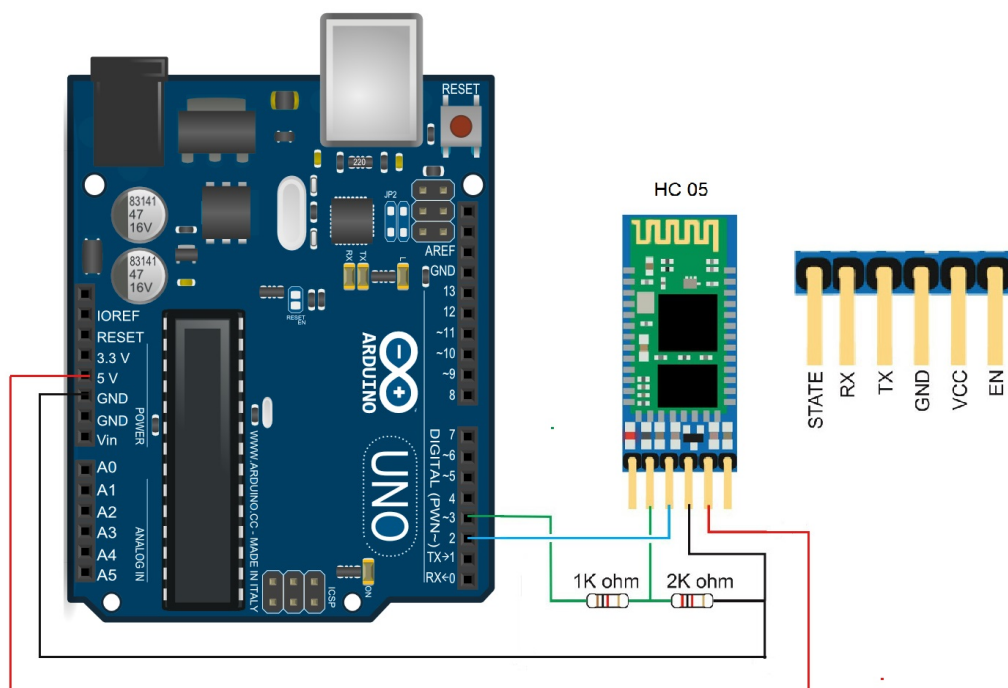


Figura 3.5: Conexionado del HC-05

## 3.5. Alimentación

### Alimentación

Para la alimentación se utiliza una fuente de computadora de 200W de potencia <sup>3</sup>. Esta fuente entrega voltajes de  $\pm 5V$  y  $\pm 12V$ .

Como 12V es lo máximo que se puede alimentar el Arduino, para no arriesgar se decide bajar la tensión. Se utiliza un regulador L7808 que presenta una salida de 8V si se le introduce una tensión de entrada en el rango de 11,5V a 21,5V. Se utiliza los 12 V disponibles. Este regulador es capaz de entregar sin disipador hasta 1A. El Arduino consume activo en el orden de centenas de miliamperes por lo que este regulador es suficiente.

Se conectaron los condensadores recomendados en la hoja de datos [4], tal cual muestra en la Figura 3.6.

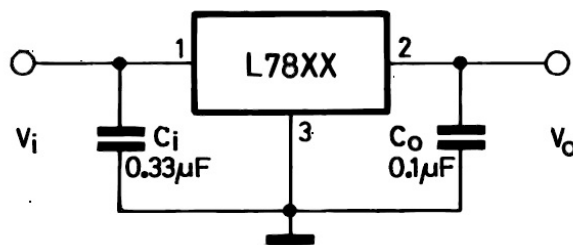


Figura 3.6: Conexión L7808 extraído de [4].

### Prueba

En la Tabla 3.3 se presentan los valores esperados y medidos de la alimentación.

Tabla 3.3: Alimentación

<b>Esperado (V)</b>	+5,0	-5,0	+12,0	-12,0	+8,0
<b>Medido (V)</b>	+4,93	-5,23	+11,5	-9,8	+7,8

Como se observa el de -12V dio un poco diferente (-9,8V), esta tensión es solo para alimentar los amplificadores, se los probó con esta tensión y no se visualizó cambio alguno a alimentarlo con -12V.

## 3.6. Implementación

### PCB

En la Figura 3.7 se muestra el layout del PCB de los circuitos diseñados. El diseño de dichos circuitos es detallado en la subsección 3.2.1 y la sección 4.4. Se utilizó la herramienta Eagle para el diseño de esquemáticos y layout.

<sup>3</sup>Marca: DELL. Modelo: NPS-200PB-73M

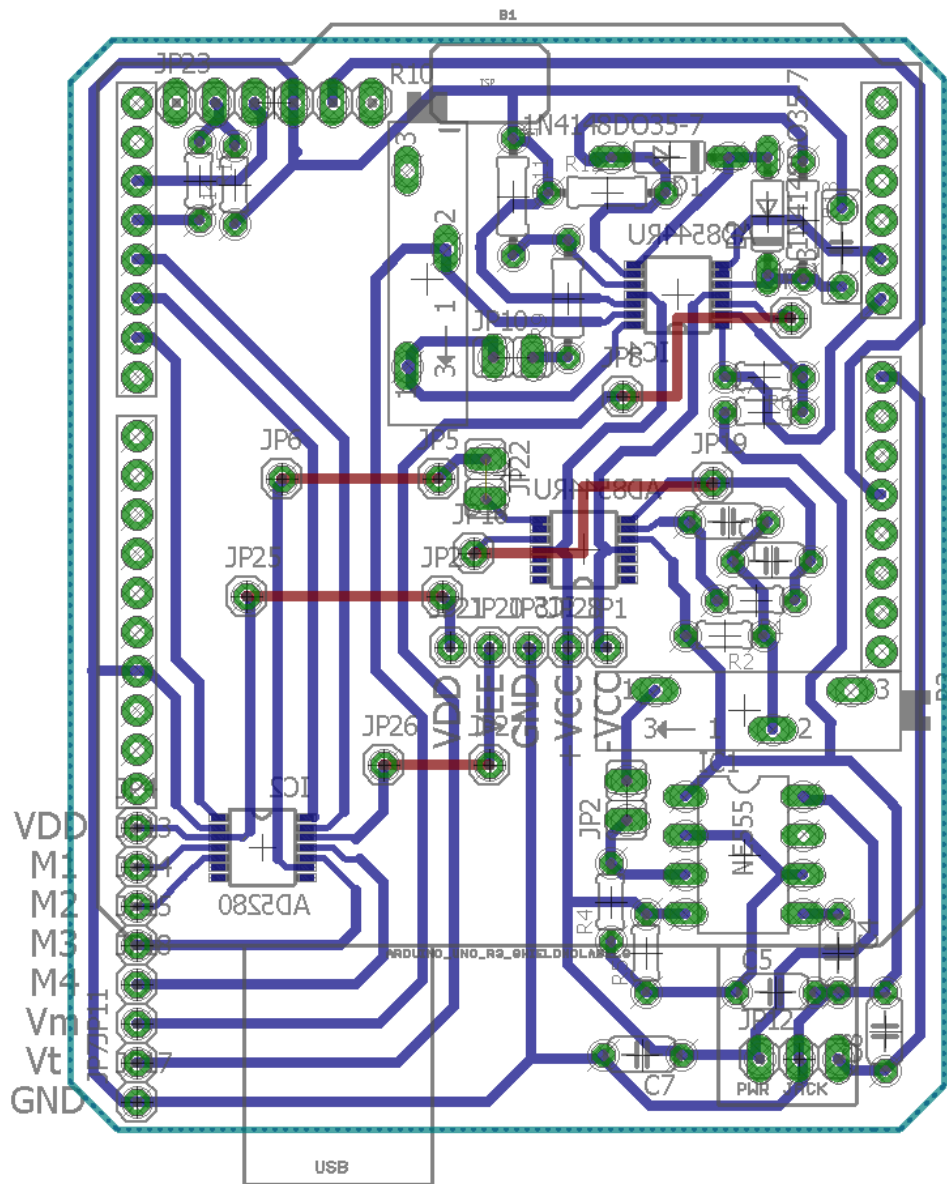


Figura 3.7: Diseño PCB

A continuación se listan brevemente las consideraciones tenidas en cuenta para el diseño del layout y algunas características del PCB:

- Se utiliza plano de tierra.
- Se implementa en una sola capa.
- Como se utiliza una sola capa es necesario colocar 5 cables puente.



### 3.6. Implementación

- Se disponen dos hileras de pines para incrustar la placa en el Arduino.
- Se hace el conexionado del regulador lineal para alimentar el Arduino.
- Se colocan pines para conectar la alimentación
  - VDD: +5V
  - VEE: -5V
  - +VCC: +12V
  - -VCC: -12V
  - GND: Tierra
- Se colocan pines para conectar el cable que va a al organo
  - VDD: +5V
  - M1...M4: Bornes de cada celda.
  - Vm: Borne común de todas las celdas.
  - Vt: Salida del LM35.
  - GND: Tierra
- Se hace el conexionado para el módulo HC-05.
- Se utilizaron tres componentes de montaje superficial (1 mux y 2 encapsulados de 4 operacionales) el resto es through-hole.
- Para facilitar la prueba de las diferentes etapas, estas se interconectaron utilizando jumpers (JP2 conecta el LM555 con el filtro, JP22 conecta el seguidor que está después del filtro con el mux, JP10 conecta el amplificador de conductividad con el detector de picos).

En las Figuras 3.8 y 3.9 se aprecia el PCB armado. Se realiza con la prototipadora LPKF ProtoMat S63 del Instituto de Ing. Eléctica.

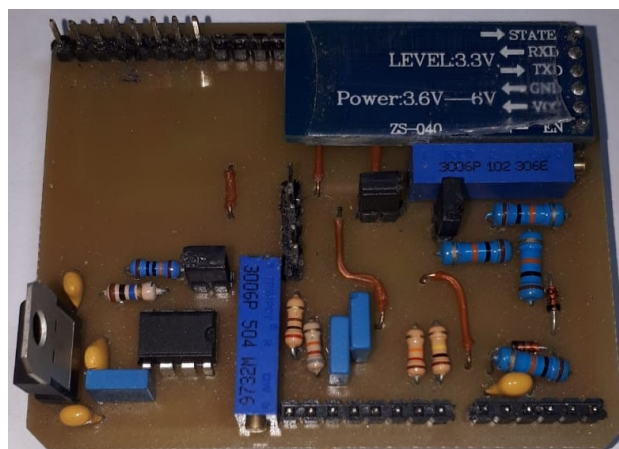


Figura 3.8: Vista superior del PCB armado

## Capítulo 3. Hardware de UM

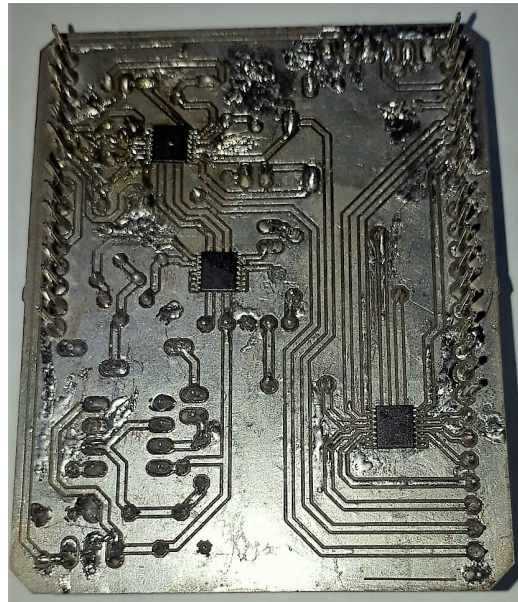


Figura 3.9: Vista inferior del PCB armado

### Cable

Para la conexión de el cable con las celdas y el LM35 se utiliza un puerto que se detalla en el capítulo 4. En 3.10 se muestra un diagrama de cómo se sueldan los cables en el puerto, el diagrama representa el puerto macho visto desde el lado que se suelda.

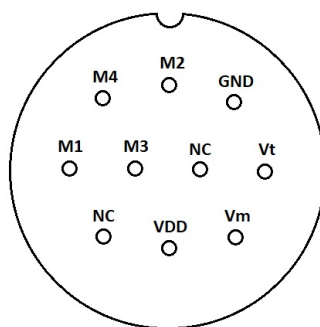


Figura 3.10: Diagrama de conexionado del puerto macho.

En la Tabla 3.4 se muestra cómo se conectan los conductores del cable que va para el organo, referido a sus colores.

Tabla 3.4: Colores de conductores

M1	M2	M3	M4	Vm	Vt	VDD	GND
Blanco	Rojo	Verde	Azul	Naranja	Marrón	Amarillo	Negro

## Protecciones

Luego de terminado el ordeño general, se lava con agua a presión la sala de ordeño por lo que la electrónica debe estar protegida contra agua. Se decide colocar el circuito con el Arduino y la fuente de alimentación dentro de una caja de registro eléctrico con grado de protección IP65<sup>4</sup>. La misma tiene una tapa que se cierra con cuatro tornillos y cuenta con una junta de goma para que no ingrese agua como se puede ver la Figura 3.11.

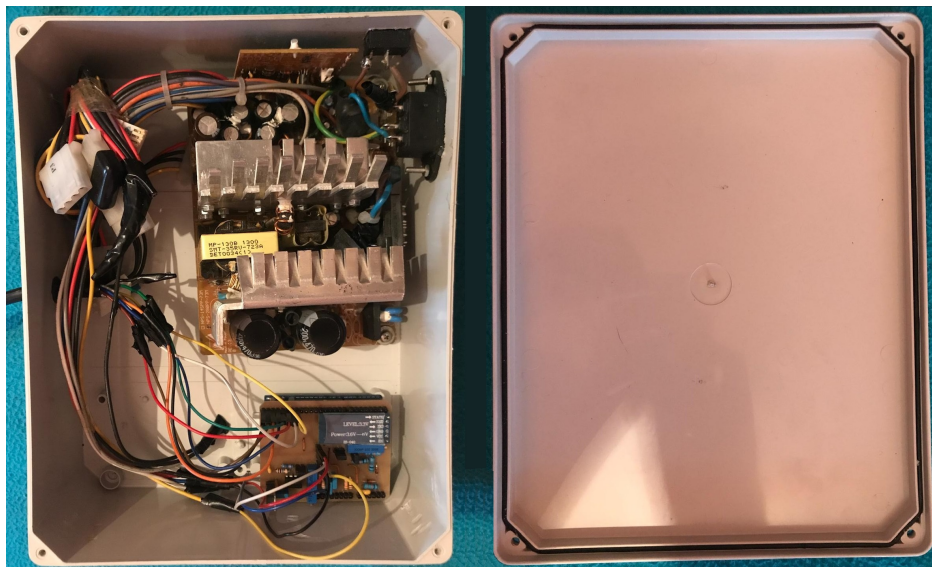


Figura 3.11: Caja de registro.

El Arduino se la fija a la caja con tornillos como se puede ver en la Figura 3.12

<sup>4</sup> Este grado de protección indica que: El polvo no debe entrar bajo ninguna circunstancia. No debe entrar el agua arrojada a chorro (desde cualquier ángulo) por medio de una boquilla de 6,3 mm de diámetro, a un promedio de 12,5 litros por minuto y a una presión de 30  $kN/m^2$  durante un tiempo que no sea menor a 3 minutos y a una distancia no menor de 3 metros.

### Capítulo 3. Hardware de UM

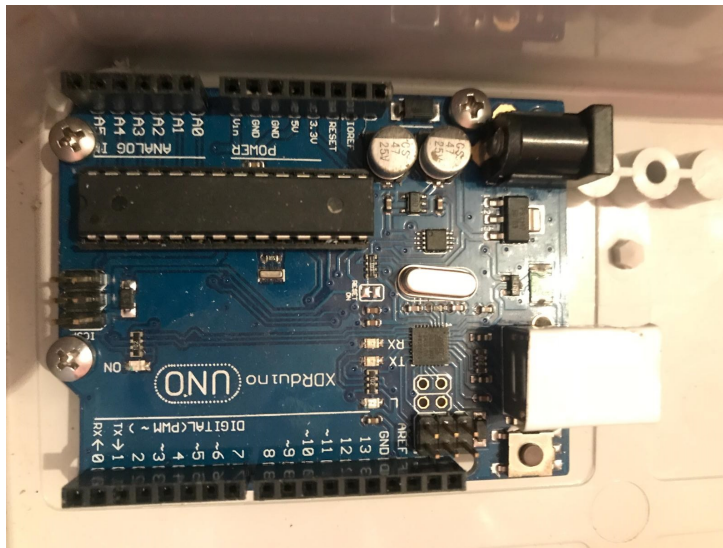


Figura 3.12: Arduino.

Encima del Arduino se incrusta el circuito de medición que ya queda sujeto por los mismos pines como se puede ver en la Figura 3.13

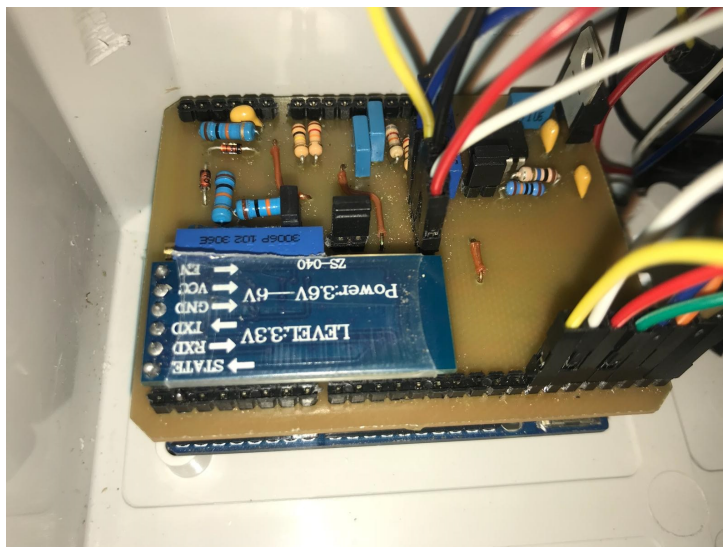


Figura 3.13: Circuito de medición.

Se le saca la carcasa a la fuente y se la aferra con tornillos dentro de esta caja también. Esto se puede ver en la Figura 3.14

Se necesita que la caja quede protegida contra agua, por lo que se le quita el ventilador a la fuente ya que de lo contrario es necesario hacer agujeros para que circule el aire. De todos modos esta fuente está diseñada para entregar hasta 200W, y como el circuito trabaja con corrientes del orden de los miliampere y voltajes del orden de decenas de volt, se espera un consumo mucho menor a 200W. Se probó

### 3.6. Implementación

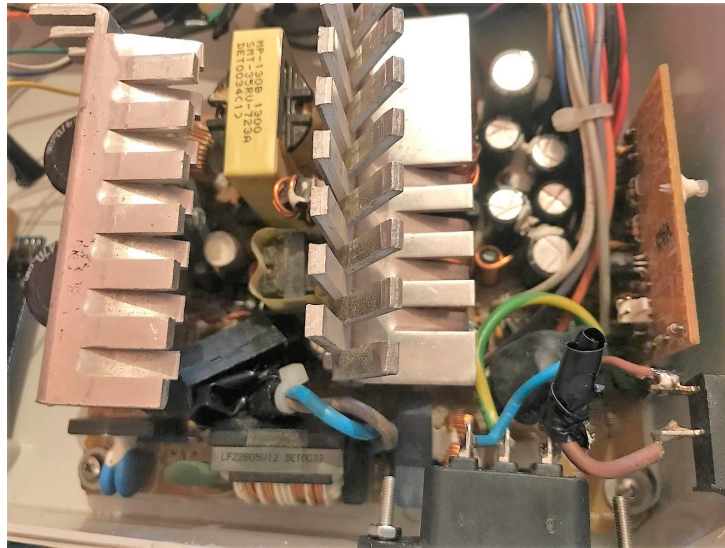


Figura 3.14: Fuente.

sin ventilador, con la caja cerrada y se verificó que la temperatura de la caja no superaba significativamente la temperatura ambiente.

Se perfora la caja para colocar un puerto para conectar a 220V y una llave de on-off como se ve en la Figura 3.15



Figura 3.15: Puerto y llave.

Se perfora la carcasa para sacar el cable que va para el organo y se pone un pasacable para que no entre agua. Y en la parte inferior se pegan dos caños para enhebrarle dos precintos, de este modo poder aferrar la caja al ducto rectangular que pasa por arriba de la fosa (ver Figura 2.1). Esto se puede ver en la Figura 3.16

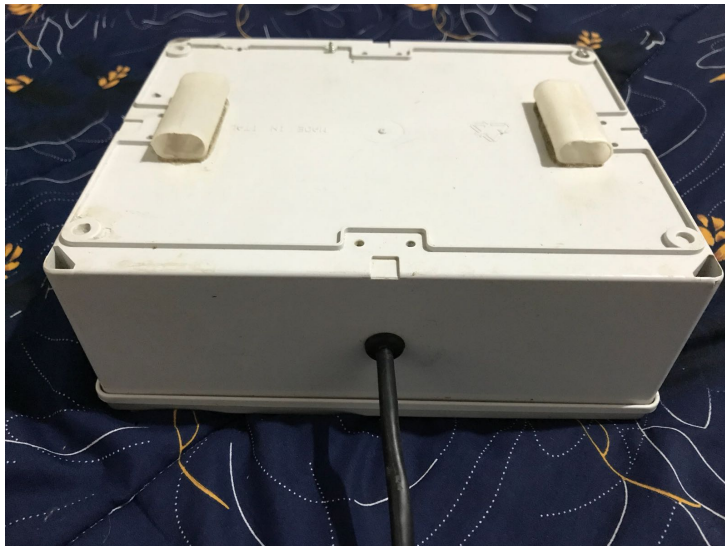


Figura 3.16: Cable y soportes.

# Capítulo 4

## Transductor de conductividad

### 4.1. Introducción

En este capítulo se detalla la construcción del transductor de conductividad. Como se menciona en la sección 1.2 es de extrema importancia medir la CE de cada cuarto. Por lo tanto, es necesario realizar modificaciones al organo para que esto sea posible ya que es el lugar físico de la máquina de ordeño donde se mezcla la leche. Tomando la idea de Lien et al. [20] se decide utilizar una celda por cuarto. Deben tenerse ciertas consideraciones al diseñar esta celda ya que el volumen del organo es limitado y no debe verse afectado su funcionamiento normal.

En el resto del capítulo se detallan las consideraciones necesarias a tener en cuenta al diseñar el circuito electrónico para realizar la medida de CE. Parte del diseño del circuito se basa en lo propuesto en el trabajo de Ferrero et al. [16], si bien algunos bloques solo mantienen la idea conceptual ya que el diseño original no cumple los requerimientos.

### 4.2. Modelado y diseño de celdas

La resistencia de una solución confinada a una celda con electrodos de área  $A$  separados una distancia  $l$  es:

$$R_m = \frac{l}{A\sigma} \quad (4.1)$$

Entonces la CE de una solución puede ser calculada como:

$$\sigma = \frac{K_{cell}}{R_m} \quad (4.2)$$

Donde  $K_{cell}(cm^{-1})$  es una constante determinada por las características geométricas de la celda ( $K_{cell} = \frac{l}{A}$ ).

Sin embargo, la constante de la celda no debe ser determinada por cálculos geométricos debido a las fugas de campo eléctrico en los bordes de los electrodos. Es más conveniente determinarla calibrando con soluciones patrones [22].

## Capítulo 4. Transductor de conductividad

La conductividad de una solución podría ser medida simplemente colocando dos electrodos en una solución y aplicando una tensión. Una corriente se generará en el circuito externo que conecta los dos electrodos. Sin embargo, una tensión continua generará electrólisis y polarización de los electrodos, reduciéndose la corriente que pasa a través del circuito a cero con el tiempo. Para evitar esto, se utiliza una tensión alterna [21].

Bajo estas condiciones se mide una impedancia compleja  $Z(f)$  dependiente de la frecuencia y representada como:

$$Z(f) = R(f) + jX(f) \quad (4.3)$$

Esta impedancia es la suma del término real resistivo,  $R(f)$ , y el término imaginario,  $X(f)$ . Se hace notar que la resistencia de la leche ( $R_m$ ) no necesariamente coincide con la parte real de  $Z(f)$  debido a los fenómenos físicos involucrados. Mabrook et al. [21] utilizan un modelo de celda similar al observado en la Figura 4.1, donde  $C_{pp}$  es la capacidad placa-placa,  $C_{pm}$  es la capacidad equivalente placa-leche,  $R_{pm}$  es la resistencia equivalente placa-leche y  $R_m$  es la resistencia de la leche (resistencia que se desea medir). A diferencia de [21] se agrega la resistencia  $R_{pm}$  ya que permite que el modelo se ajuste mejor a los datos. La dependencia de las resistencias ( $R$ ) con las dimensiones de la celda está dado por 4.1. La dependencia de las capacitancias ( $C$ ) con las dimensiones de la celda está determinada por:

$$C = \epsilon \frac{A}{l} \quad (4.4)$$

Donde  $\epsilon$  es la permitividad.

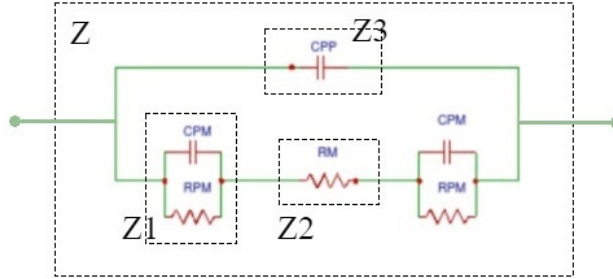


Figura 4.1: Modelo eléctrico equivalente de la celda

Las impedancias en Laplace son:

$$Z_1 = \frac{R_{pm}}{R_{pm}C_{pm} \cdot S + 1}$$

$$Z_2 = R_m$$

$$Z_3 = \frac{1}{C_{pp} \cdot S}$$

Por lo tanto, a partir de la Figura 4.1 se puede inferir que si existe  $fm$  que verifique:



## 4.2. Modelado y diseño de celdas

$$2Z_1(fm) \ll Rm \implies Rm + 2Z_1(fm) \simeq Rm \quad (4.5)$$

$$Z_3(fm) \gg Rm \implies Rm // Z_3(fm) \simeq Rm \quad (4.6)$$

Se cumple:

$$Z(fm) \simeq Rm \quad (4.7)$$

Por lo que se propone buscar una celda que minimice la frecuencia  $fm$  para la cual se verifiquen las condiciones dadas por las ecuaciones 4.5 y 4.6. Se busca la frecuencia mínima porque a alta frecuencias aparecen fenómenos eléctricos (capacidades parasitarias) que dificultan el análisis del circuito eléctrico. También cuanto mayor es la frecuencia mayores son los requerimientos en cuanto a slew rate y ancho de banda de los componentes eléctricos.

Para determinar cuáles son las dimensiones de la celda que minimiza la frecuencia  $fm$  se construyen tres celdas de distintas dimensiones de forma de analizar la dependencia de las dimensiones de la celda con su respuesta en frecuencia. Los electrodos se realizan de acero inoxidable ya que los de las celdas finales deben ser de ese material para no ser corroídos por los químicos utilizados en el lavado de la ordeñadora. Las celdas se observan en la Figura 4.2.

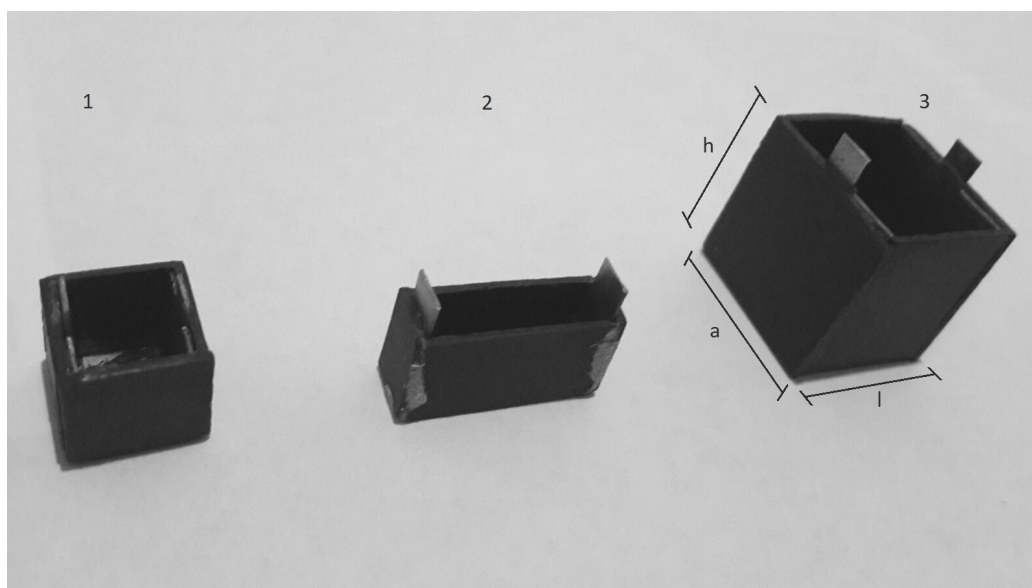


Figura 4.2: Celdas de ensayo

Las celdas se encuentran numeradas y sus lados identificados, distancia entre electrodos ( $l$ ), ancho ( $a$ ) y altura ( $h$ ). En la Tabla 4.1 se observan las dimensiones de las mismas.

Se utiliza el analizador de impedancias **Cypher Instruments C60** para adquirir la respuesta en frecuencia de las celdas para una solución de KCl con una concentración de 0.01 mol/kg. Según una tabla presentada en [17], una solución de KCl con una concentración de 0.01 mol/kg a una temperatura de  $19^{\circ}C$  presenta una conductividad de 1.246 mS/cm. En las Figuras 4.3-4.4-4.5 se observan el

## Capítulo 4. Transductor de conductividad

Tabla 4.1: Dimensiones de las celdas

Celda	l (cm)	a (cm)	h (cm)	a.h (cm <sup>2</sup> )
1	0.9	1.0	1.0	1.0
2	2.0	0.5	1.0	0.5
3	1.9	2.0	2.1	4.2

módulo y fase obtenida por el analizador para cada celda con la solución de KCL a 19°C. Para las tres celdas se observa una respuesta plana con fase casi nula (impedancia resistiva) en un entorno de los 10 kHz. Se realiza el ajuste de los datos al modelo, también mostrado en las figuras anteriores, obteniendo mejores resultados si se realiza en el rango 1kHz-100kHz, en lugar de todo el rango de adquisición.

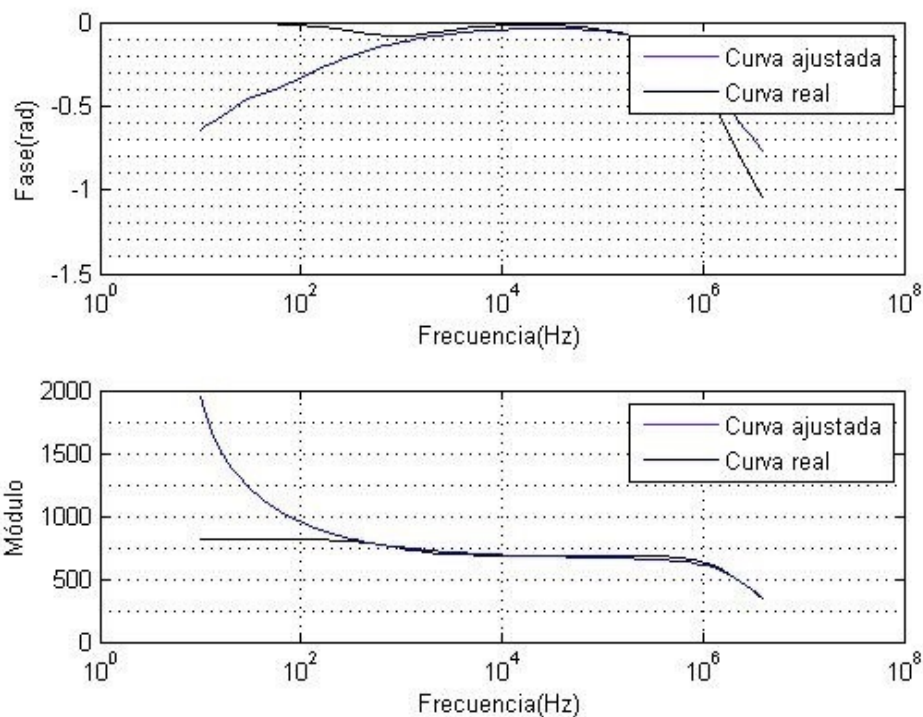


Figura 4.3: Módulo y fase de la impedancia de la celda 1 con una solución de KCL de concentración 0.01 mol/kg. En negro se tiene el modelo eléctrico equivalente ajustado en el rango 1-100kHz

## 4.2. Modelado y diseño de celdas

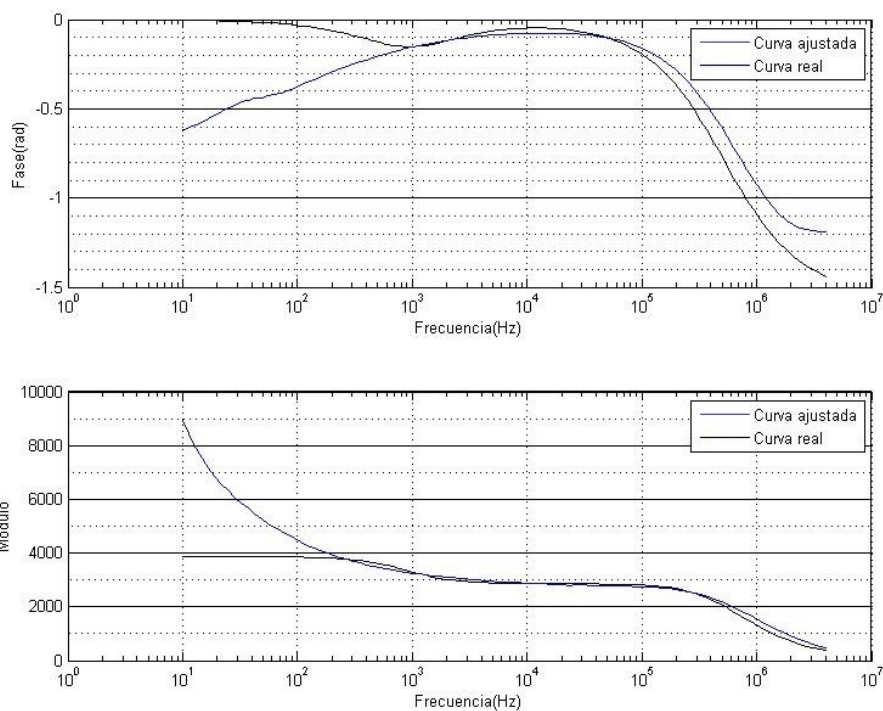


Figura 4.4: Módulo y fase de la impedancia de la celda 2 con una solución de KCl de concentración 0.01 mol/kg. En negro se tiene el modelo eléctrico equivalente ajustado en el rango 1-100kHz

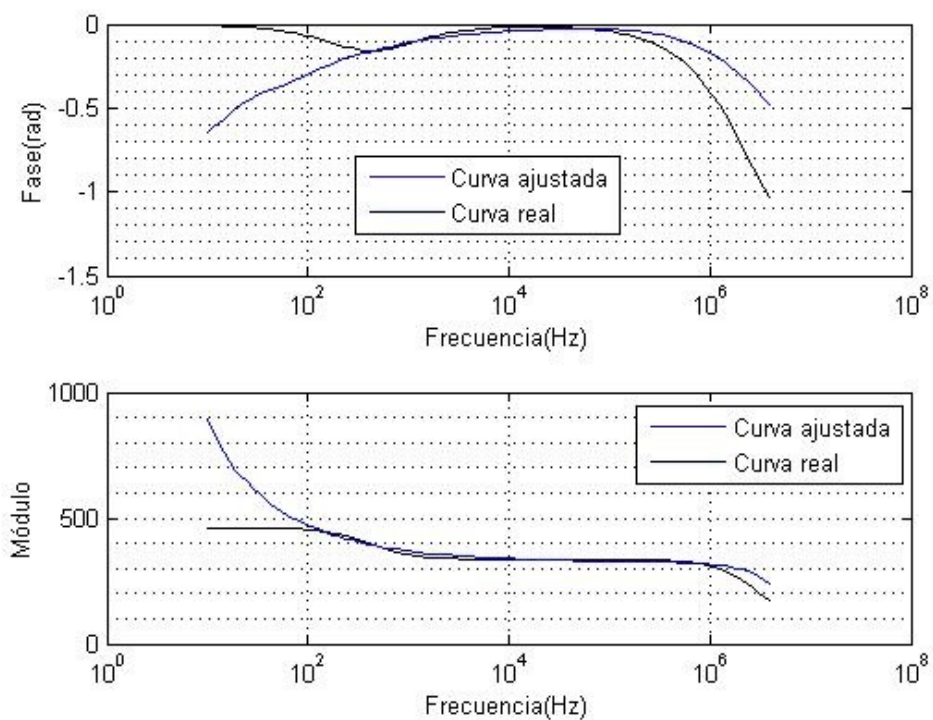


Figura 4.5: Módulo y fase de la impedancia de la celda 3 con una solución de KCl de concentración 0.01 mol/kg. En negro se tiene el modelo eléctrico equivalente ajustado en el rango 1-100kHz

## Capítulo 4. Transductor de conductividad

En la Tabla 4.2 se observan las relaciones esperadas teóricas entre las componentes eléctricas equivalentes de las celdas en relación a los valores de la celda 1.

Tabla 4.2: Relación entre las componentes eléctricas equivalentes entre celdas.  $\mathbf{A}$  es el área de la celda 1 y  $\mathbf{l}$  el largo. Se asume que al cambiar el área de la cara  $\mathbf{a.h}$  la separación entre la interfaz electrodo-leche se mantiene constante.

<b>celda 1</b>	$\mathbf{A=a.h}$	$\mathbf{l}$	$C_{pm}$	$R_{pm}$	$C_{pp}$	$R_m$
<b>celda 2</b>	$\mathbf{A/2}$	$\mathbf{2l}$	$C_{pm}/2$	$2R_{pm}$	$C_{pp}/4$	$4R_m$
<b>celda 3</b>	$\mathbf{4A}$	$\mathbf{2l}$	$4C_{pm}$	$R_{pm}/4$	$2C_{pp}$	$R_m/2$

En la Tabla 4.3 se observan los parámetros obtenidos en el ajuste. Se aprecia que los valores obtenidos  $C_{pp}$ ,  $C_{pm}$  y  $R_{pm}$  no cumplen cuantitativamente las relaciones de la Tabla 4.2. Entre otras cosas esto se debe a que las relaciones de la Tabla 4.2 están aproximadas. Sin embargo, se observa que cualitativamente se cumplen las relaciones. Como por ejemplo al disminuir el área de las caras de los electrodos,  $C_{pm}$  y  $C_{pp}$  disminuyen y  $R_{pm}$  aumenta. La resistencia  $R_m$  en cambio si cumple cuantitativamente las relaciones de la Tabla 4.2. Esto indica que hay algún fenómeno físico que no se está modelando adecuadamente (o ni siquiera se modela).

Tabla 4.3: Parámetros ajustados para cada celda

	<b>Cpm(<math>\mu F</math>)</b>	<b>Rpm(ohm)</b>	<b>Cpp(pF)</b>	<b>Rm(ohm)</b>
<b>celda 1</b>	3.389	63.78	100.0	688.4
<b>celda 2</b>	0.9354	501.0	80.00	2861
<b>celda 3</b>	6.915	43.56	200.0	337.0

Tabla 4.4: Error relativo entre la resistencia  $R_m$  determinada por el analizador y el módulo de la impedancia a 10 kHz

	<b>Rm(ohm)</b>	<b><math> Z _{@10kHz}</math></b>	<b>Error Relativo</b>
<b>celda 1</b>	688.4	692.2	0.6 %
<b>celda 2</b>	2861	2883	0.8 %
<b>celda 3</b>	337.0	340.3	1.0 %

En la Figura 4.6 se observa el módulo de cada impedancia del modelo comparado con el módulo de la impedancia total para los parámetros del ajuste de la celda 1 (se obtienen curvas similares para el resto de las celdas, por lo que no se presentan). El módulo de la impedancia  $Z_3$  es mucho mayor que el resto de las impedancias a frecuencias menores a 100 kHz por lo que no aporta en el rango de interés ya que está en paralelo. Las impedancias series ( $2Z_1$ ) tienen módulo despreciable a partir de los 10 kHz. Por lo tanto, se aprecia claramente que a 10 kHz

## 4.2. Modelado y diseño de celdas

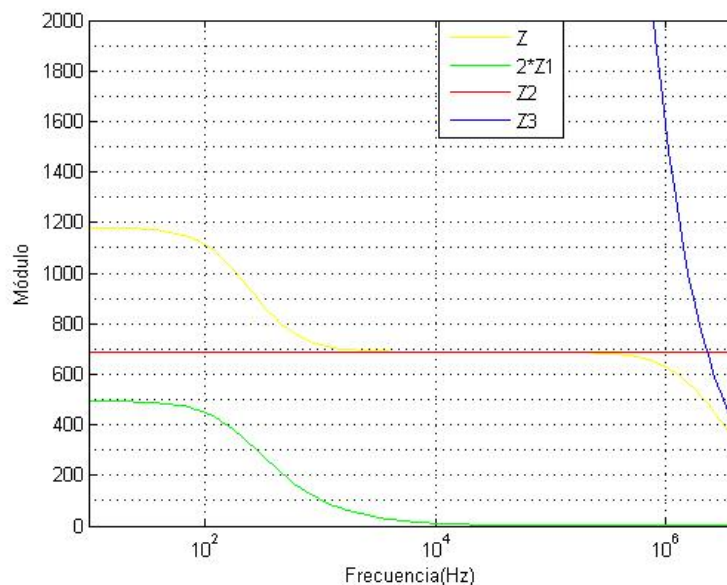


Figura 4.6: Módulo de impedancias del modelo eléctrico equivalente de la celda 1 con solución.

la impedancia total está mayormente determinada por la resistencia de la leche (Tabla 4.4).

Si se quiere tener un error relativo entre la resistencia de la solución y el módulo de la impedancia a 10 kHz menor al de la celda 1, se puede diseñar otra celda. Para esto, es necesario disminuir la frecuencia de corte de la impedancia Z1 manteniendo las otras variables del modelo constantes. Esta frecuencia de corte en el caso de la celda 1 tiene un valor de  $\frac{1}{2\pi C_{pm} R_{pm}} = 736 Hz$ . Para disminuir la frecuencia, se puede aumentar la capacidad. Una manera de lograr esto es aumentar el área de la celda. La celda 3 cumple con esta condición ya que tiene un área de  $4.2 cm^2$ . La frecuencia de corte de la impedancia Z1 en el caso de la celda 3 es de 360 Hz. Lo cual es razonable, ya que la capacitancia placa-leche para esta celda es de alrededor del doble de la de la celda 1 y la resistencia placa-leche apenas se decrementa (Tabla 4.3). Además, al aumentar la capacidad  $C_{pm}$  disminuye el módulo de la impedancia Z1. Sin embargo, para que la resistencia de la solución se conserve (de esta forma se garantiza que el error relativo disminuye), la longitud de la celda (l) se debe incrementar en la misma proporción que el área de la cara. Por lo que se estima que una nueva celda (celda 4) de dimensiones  $(l, a, h) = (2,0, 2,8, 2,8)$  tendría menor error relativo que la celda 1.

Sin embargo, para la elección de las dimensiones de las celdas aparte de que tenga bajo error relativo, las dimensiones tienen que ser tales que entren las cuatro en el órgano sobre un mismo plano. Se tomaron las medidas necesarias en el órgano y se llega que las celdas tienen que estar incluidas dentro de un cilindro de 3cm de radio por 3cm de altura. Por lo que la diagonal de la base de la celda debe cumplir  $l^2 + a^2 < 9$ . Para las dimensiones de la celda 4 esto no se cumple. La siguiente mejor

## Capítulo 4. Transductor de conductividad

en términos de error relativo es la celda 1. Además, esta celda tiene la ventaja que por ser tan chica tiene una tensión superficial tal que si se inclina  $45^\circ$  la leche no se derrama (la celda sigue llena). Esto es necesario ya que el organo suele moverse porque se encuentra colgando de la ubre. Mientras que una celda un poco más grande (por ejemplo una cúbica de 2cm de lado) ya no tiene esta propiedad.

En resumen se selecciona la celda 1 y una frecuencia de trabajo de 10 kHz.

La conductividad de la leche también depende de la temperatura. Esto se debe a que cuando la temperatura se incrementa, la movilidad de los iones aumenta por lo que la conductividad aumenta. Para compensar esto se utiliza la siguiente fórmula [16]:

$$\sigma = \sigma_o(1 + \alpha(T - T_o)) \quad (4.8)$$

Donde  $\sigma_o$  es la conductividad de la leche a la temperatura  $T_o = 25^\circ C$ . Se desea calcular el parámetro  $\alpha$  para realizar las correcciones pertinentes al tomar cada medida.

Despejando  $\alpha$  de la ecuación 4.8 y sustituyendo la conductividad por la expresión 4.2 se llega a la ecuación 4.9.

$$\alpha = \frac{1}{T - T_o} \left( \frac{R_{mo}}{R_m} - 1 \right) \quad (4.9)$$

Para calcular  $\alpha$  se releva la resistencia  $Z$  de la celda 1 a 10kHz a tres temperaturas distintas, como el error relativo a esta frecuencia es bajo, se asume  $R_m \approx Z$ . En la Tabla 4.5 se observan los valores.

Tabla 4.5:  $R_m$  a distintas temperaturas

T ( $^\circ C$ )	$R_m$ ( $\Omega$ )
21	200
27	178
33	154

Con estos datos se puede calcular tres estimadores de alfa.

Tabla 4.6: Cálculo de alfa

$\alpha$ ( $\%/^\circ C$ )	2,1	2,5	2,6
----------------------------	-----	-----	-----

Se observa que son parecidos (alrededor del  $2\frac{\%}{^\circ C}$ ). La influencia en la conductividad es baja por lo tanto no es necesario un valor preciso, solo se realiza un promedio de estos valores y no un ajuste lineal de los datos. Promediando se obtiene  $\alpha = 2,4\%/^\circ C$ . Este valor es similar al obtenido en [16].

### 4.3. Diseño mecánico

En la sección anterior se elige la celda cúbica de 1cm de lado. En esta sección se presenta cómo se implementa la pieza portadora de transductores para que los mismos puedan medir dentro del organo.

#### El organo

En la Figura 4.7 se presenta el organo, este cuenta con una pieza de acero inoxidable y otra de un plástico resistente posiblemente acrílico.



Figura 4.7: organo

En la Figura 4.8 se muestra las dos piezas por separado. La pieza metálica va conectada a las cuatro pezoneras que se encargan de extraer la leche de cada pezón mediante un sistema de vacío. Un caño de vacío y uno de leche corresponde a una pezonerá. Los dos caños de vacío sobrante van conectados a los dos caños principales de vacío ubicados arriba de la fosa.

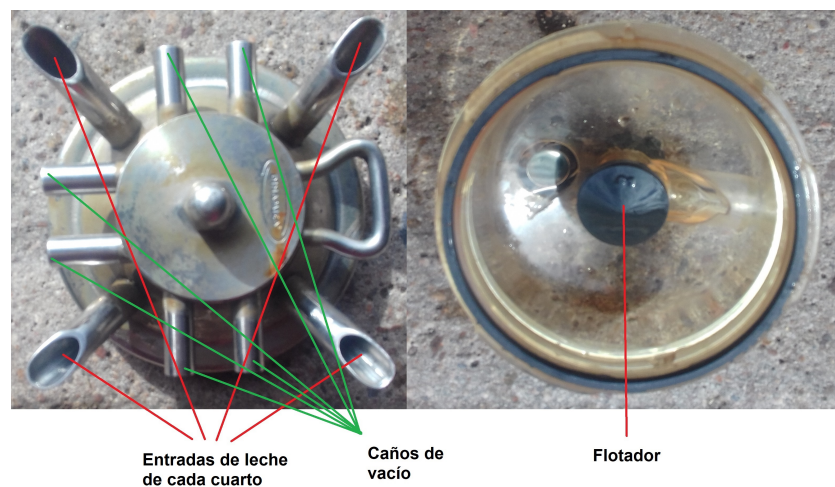


Figura 4.8: Piezas del organo

## Capítulo 4. Transductor de conductividad

La leche ingresa al organo por los cuatro ductos de la pieza metálica para caer dentro de la pieza de plástico. La pieza de plástico cuenta con un flotador, lo que dificulta ubicar ahí la pieza portadora de transductores, por lo que se decide ubicarla contra la pieza metálica. En la Figura 4.9 se muestra el interior de la pieza metálica. Se toman todas las medidas necesarias para realizar el diseño de la pieza portadora de transductores.



Figura 4.9: Interior pieza metálica organo.

El tambo donde se prueba el dispositivo tiene vacas que producen en promedio 18 l en 4 minutos por lo tanto cada cuarto tiene un caudal promedio de

$$Q = \frac{18000 \text{ ml}}{240 \text{ s} \times 4} \approx 19 \text{ ml/s}$$

El volumen de cada celda es:

$$V = 1 \text{ cm}^3 = 1 \text{ ml}$$

Por lo que en un segundo este caudal llenaría 19 celdas. Entonces si el fluido ingresa directamente en la celda provocará turbulencia lo que dificultará realizar una medida precisa de la conductividad.

Se planteó tomar una porción de ese flujo colocando un embudito enfrente a cada ducto, dejando una separación entre el agujero y la boca del embudo. De esta manera se escapa la porción de flujo restante por esa separación, volcando el excedente fuera de la celda. por lo tanto se reduce el caudal que llega a la misma.

Se realizó una prueba para evaluar la efectividad de la solución propuesta, se detallan en el Apéndice B.1.

Se encontró que el diámetro óptimo de la salida del embudo es de 1 mm.

### Método de fabricación

Para la fabricación de la pieza portadora de transductores, el instituto cuenta con una impresora 3D. Para realizar el diseño 3D se utiliza la herramienta Freecad.



## Diseño 3D

Se divide en dos piezas para facilitar el armado. Luego se incrustan las piezas y se pegan con pegamento acrílico (más conocido comercialmente como *la gotita*).

En la Figura 4.10 se observa la pieza inferior, la cual contiene las cuatro celdas unidas a una columna rectangular. Por el interior de la misma pasan los cables que se conectan a los electrodos de cada celda. Se observa un orificio que conduce a cada electrodo y un orificio para cada tirafondo, los tirafondos son para realizar la conexión cable-electrodo.

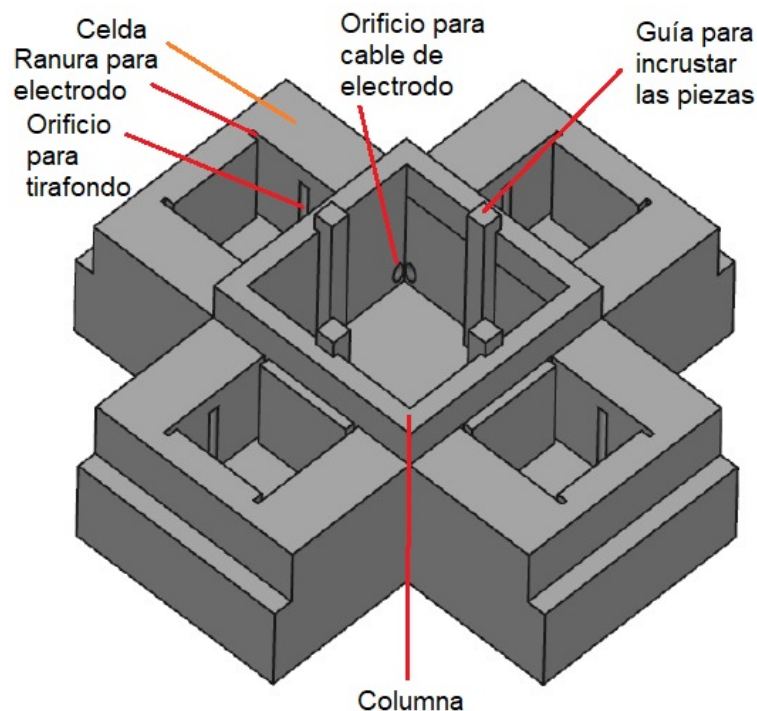


Figura 4.10: Vista superior de pieza inferior

En la Figura 4.11 se observa que los cables salen en la base de la celda, donde se enrollan en tirafondos de acero inoxidable (para que no los corra el lavado). Los tirafondos van atornillados en el plástico y a su vez empujando los electrodos de modo de que hagan contacto.

La pieza superior se puede ver en la Figura 4.12. Esta contiene orificios para los tornillos de acero inoxidable que aferran la pieza a la pieza metálica del organo. Además contiene el soporte para el LM35, orificios para los cables del LM35, los embudos y las extensiones de embudos que son para que cuando el caudal es muy bajo, no se escape leche por abajo y evitar que no pase por la celda.

En la Figura 4.13 se observa paredes para que no pase nada de leche de un cuarto a otro. Como se ve, falta una pared, por que se hace de silicona ya que va encima del puerto que conecta con el cable que va para el circuito. En el espacio para el puerto se observa los orificios de los cables que van conectados al mismo.

## Capítulo 4. Transductor de conductividad

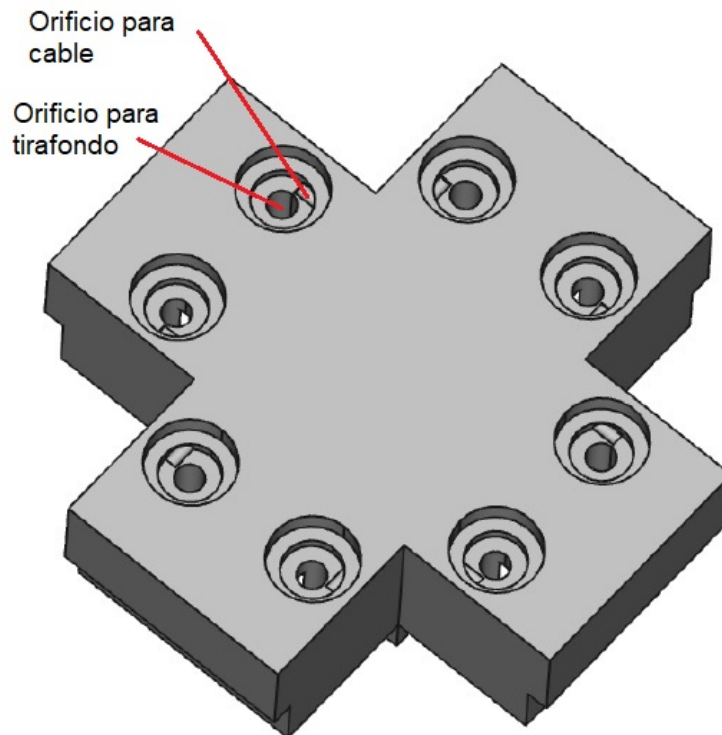


Figura 4.11: Vista inferior de pieza inferior

La pieza cuenta con agujeros para los cables y un soporte para el LM35. Además cuenta con orificios hexagonales para calzar las tuercas de los tornillos que aferran la pieza a la pieza metálica del organo y hendiduras para incrustar las piezas, calzando las guías en las hendiduras.

El diseño final descrito anteriormente es producto de varias iteraciones en las que se fue mejorando detalles constructivos, obteniendo un resultado satisfactorio.

### 4.4. Diseño electrónico

En esta sección se detalla el diseño del circuito electrónico para poder realizar la medida de conductividad a 10 kHz.

#### 4.4.1. Diseño

A continuación se listan los criterios de diseño:

- La relación entre la conductividad y la tensión de salida tiene que ser lineal.
- Corresponder 12mS con 5V con posibilidad de regulación.

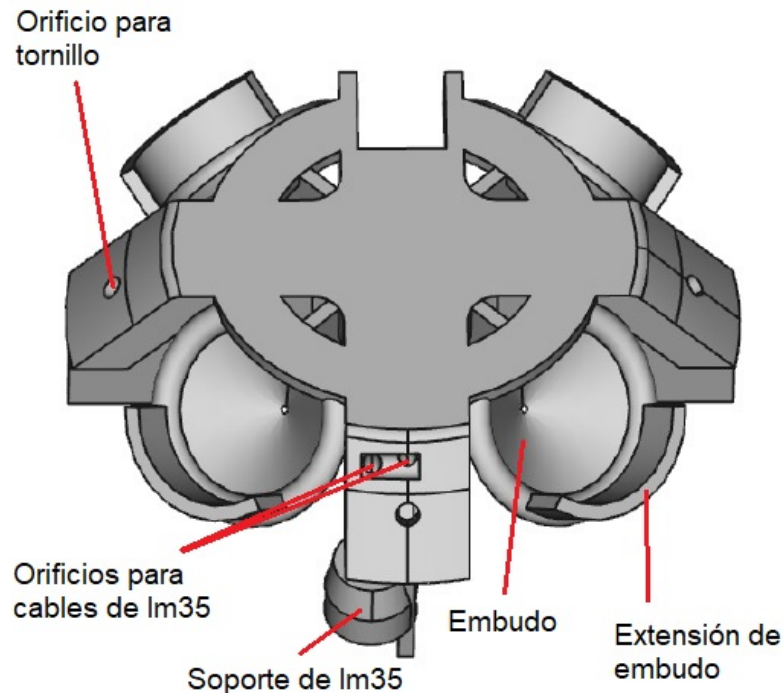


Figura 4.12: vista superior de pieza superior

- Error de no linealidad de 3%. En el trabajo de Ferrero et al. [16] obtienen un error de no linealidad del 4,2%. Se tiene como objetivo mejorar este error.
- Muestreo de conductividad de cada cuarto cada 1s ya que es la tasa de muestreo que se utiliza en [20] y el doble se utiliza en [16].

La idea de la medida de conductividad de la leche es aplicar una tensión conocida en las placas de las celdas y medir las corrientes que circulan para determinar la resistencia o conductividad (Ley de Ohm). Por los motivos expuestos anteriormente se utiliza una tensión sinusoidal de 10 kHz. En la Figura 4.14 se presenta el diagrama de bloques del circuito. El mismo cuenta con un oscilador de onda cuadrada de aproximadamente 10kHz, un filtro pasa banda para obtener la fundamental, un multiplexor analógico para compartir el circuito entre las celdas y no tener que fabricar un circuito para cada una, un amplificador en configuración inversora donde una de sus resistencias es la de la leche y un detector de picos que obtiene a la salida un voltaje continuo igual a la amplitud de la sinusoide de entrada. Se aclara que la Arduino no pertenece al circuito, solo se muestra para dar una vista general.

En la Figura 4.15 se presenta el esquemático del circuito, donde se encuentran recuadrados los bloques mencionados en la Figura 4.14.

Se observa que se agregan dos seguidores, estos son necesarios para obtener baja impedancia de salida en el filtro pasa banda y en el detector de picos.

La salida del integrado LM35 va conectada al pin JP7. Los pines JP13-16 van

## Capítulo 4. Transductor de conductividad

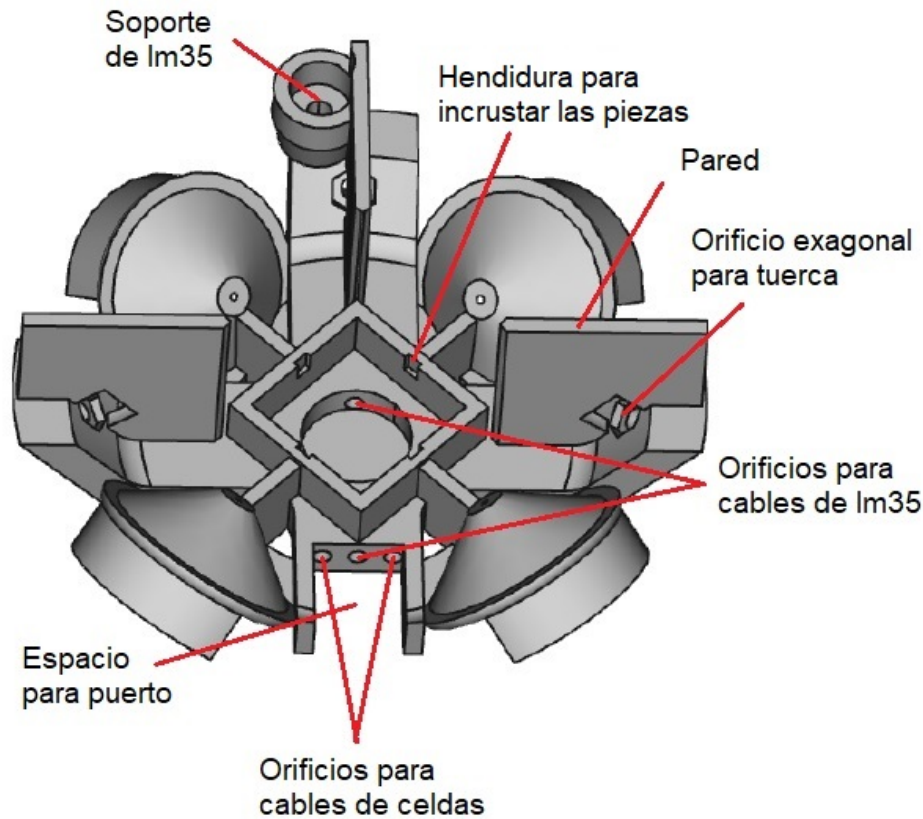


Figura 4.13: Vista inferior de pieza superior

conectados a un electrodo de cada celda y el otro electrodo correspondiente a cada celda van conectados a un mismo conductor que se conecta al pin JP11.

En la Figura 4.16 se muestra el diagrama de conexión de los electrodos. En esta Figura hay  $M_i$  en vez de  $J P_i$ . Las  $R_{m_i}$  son las resistencias equivalentes de cada celda.

### Oscilador

El oscilador a diseñar tiene que cumplir con una serie de requerimientos dado el uso que va a tener.

Considerando que el ordeño de una vaca insume aproximadamente 4 minutos, el oscilador debe estabilizarse en segundos. La frecuencia debe ser de aproximadamente 10 kHz y la amplitud de 1 V. La amplitud se eligió igual a la utilizada en el trabajo de Ferrero et al. [16] donde se utilizaba una celda de 1 cm de distancia entre electrodos.

Siguiendo [16] se diseñó un oscilador de Wien. Sin embargo en las pruebas se detectó que demoraba mucho en estabilizarse la amplitud (minutos) y se necesita que se estabilice en segundos. Por lo que se decide generar un tono utilizando el integrado 555 (generación de pulso rectangular con frecuencia fundamental 10

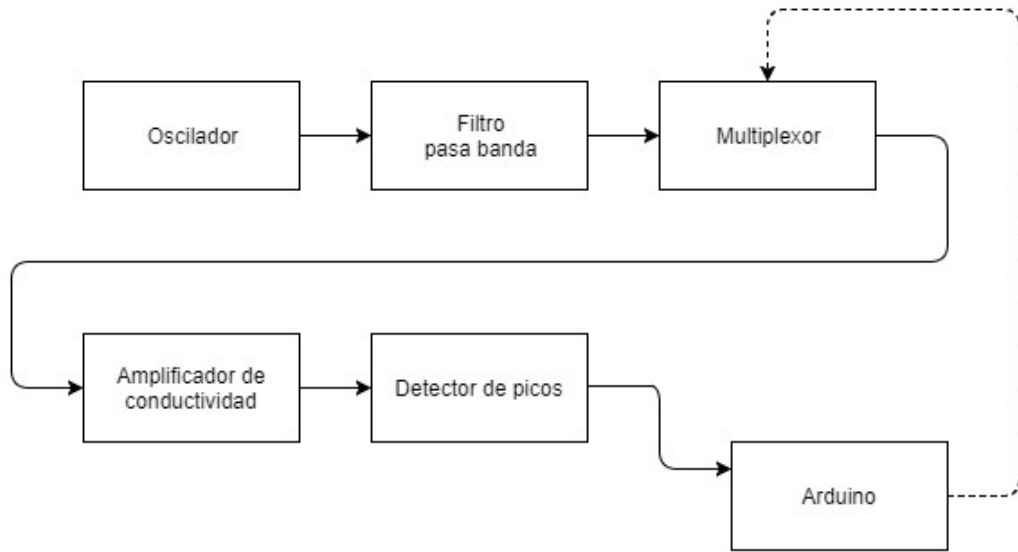


Figura 4.14: Diagrama de bloques del circuito

kHz) y un filtro pasabanda que permita solo pasar la fundamental.

El integrado LM555 en la configuración utilizada se comporta como un oscilador astable que genera una onda cuadrada de ciclo de trabajo 50% y frecuencia regulable dada por la ecuación 4.10.<sup>1</sup>

$$f = \frac{1}{2R_4C_5 \ln(2)} Hz \quad (4.10)$$

Se desea obtener una frecuencia de aproximadamente 10kHz. Se dispone de variedad de condensadores del orden de los 100nF, por lo tanto eligiendo  $R_4 = 130\Omega$  y  $C_5 = 470nF$  se obtiene

$$f = 11,8kHz$$

Para la elección de las  $C_4$  y  $R_5$  se siguieron las recomendaciones del fabricante.

Tabla 4.7: Componentes seleccionados

$R_4(\Omega)$	$R_5(\Omega)$	$C_4(nF)$	$C_5(nF)$
130	100k	10	470

La tensión de alimentación máxima es de 18V por lo que se lo puede alimentar a 12V. Como excursiona en todo el rango, generará entonces una onda cuadrada de amplitud 6V.

En la siguiente etapa se utiliza un filtro para obtener la fundamental y regular la amplitud a 1V.

<sup>1</sup>Por detalles de su derivación consultar Apéndice B.2.1.

## Capítulo 4. Transductor de conductividad

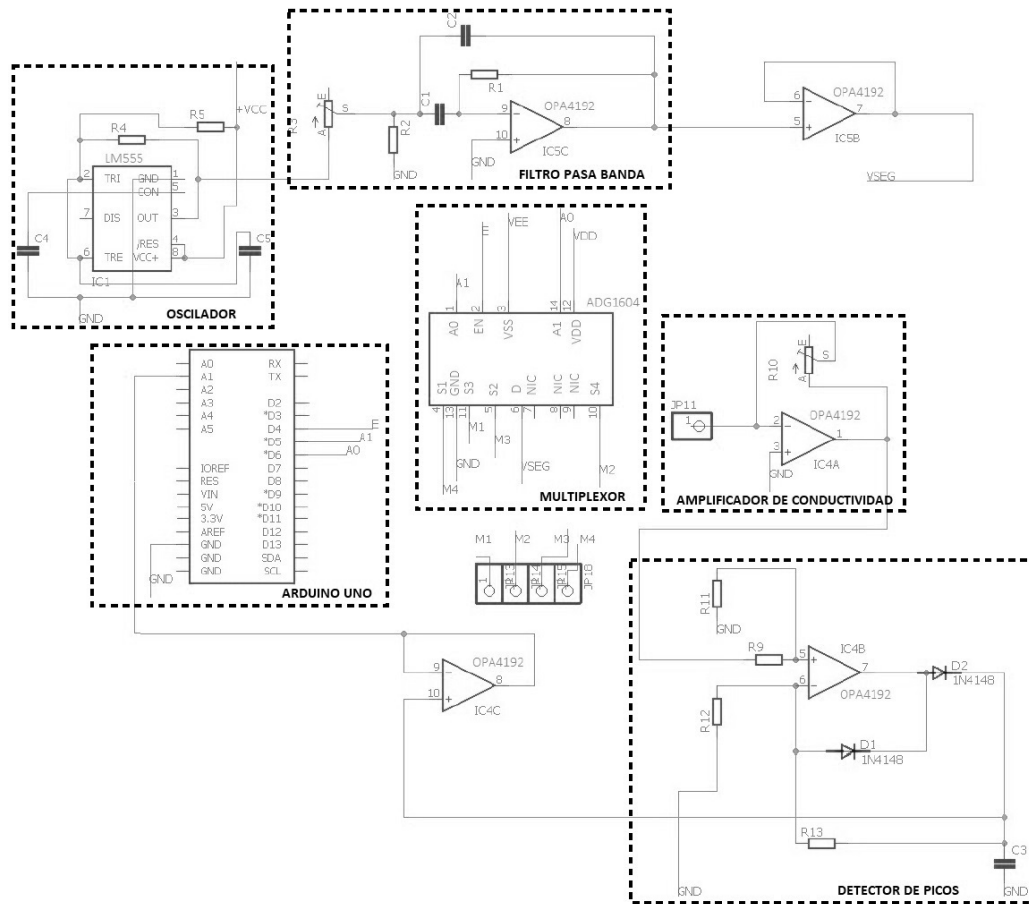


Figura 4.15: Esquemático del circuito

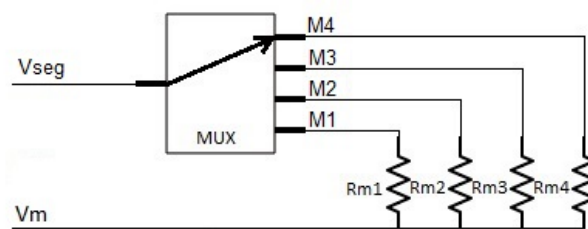


Figura 4.16: Diagrama de conexión de electrodos

### Filtro pasa banda MFB

Se opta por la configuración de la Figura 4.17 para implementar el filtro pasa-banda.

La frecuencia, ganancia y factor de calidad son los siguientes<sup>2</sup>.

Frecuencia central del filtro pasabanda

<sup>2</sup>Por detalles de la deducción consultar Apéndice B.2.2

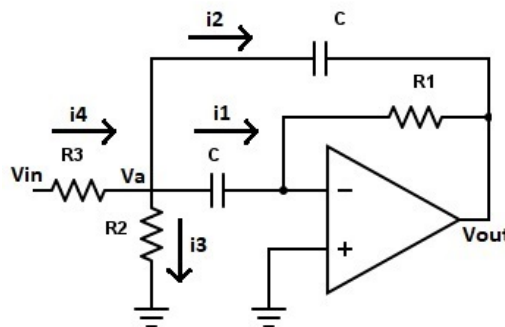


Figura 4.17: Filtro pasa banda MFB

$$f_c = \frac{1}{2\pi C} \sqrt{\frac{1}{R_1} \left( \frac{1}{R_2} + \frac{1}{R_3} \right)}$$

Ganancia a la frecuencia central

$$G = \frac{-R_1}{2R_3}$$

Factor de calidad

$$Q = \frac{1}{2} \sqrt{R_1 \left( \frac{1}{R_2} + \frac{1}{R_3} \right)} \quad (4.11)$$

### Elección de componentes

Se desea que se pueda regular la ganancia para poder ajustar la amplitud de la sinusoide a la salida y sin variar la frecuencia ni el factor de calidad. Por lo tanto eligiendo  $R_3$  como la resistencia variable y que cumpla  $R_3 \gg R_2$  se verifican esos requerimientos.

$$f_c \approx \frac{1}{2\pi C \sqrt{R_1 R_2}} \quad (4.12)$$

$$Q = \frac{1}{2} \sqrt{\frac{R_1}{R_2}} \quad (4.13)$$

Como se ve, no dependen de  $R_3$ . Un factor de calidad bueno<sup>3</sup> es de aproximadamente 10. Además se requiere que la frecuencia sea de 11,8kHz. Para utilizar un condensador del orden de los nanos y cumplir los requerimientos se elije  $R_1 = 82k$  y  $R_2 = 220\Omega$ , por lo tanto  $C = 3,3nF$ .

Como la onda cuadrada de entrada tiene amplitud 6V, la fundamental de esta onda cuadrada va a tener una amplitud cercana a 6V. A la salida se desea una

<sup>3</sup>Un factor de calidad de 10 quiere decir que a una frecuencia  $f_2 = 1,05f_c$  la ganancia ya cae 3dB por lo tanto el tercer armónico de  $f_c$  tiene una atenuación mucho mayor.

## Capítulo 4. Transductor de conductividad

sinusoide de amplitud 1V, por lo tanto la ganancia de filtro debe ser de alrededor de

$$|G| \approx \frac{1}{6} = \frac{R_1}{2R_3}$$
$$\Rightarrow R_3 \approx 246k\Omega$$

Por lo tanto con un preset de 500k es suficiente.

Tabla 4.8: Componentes seleccionados

$R_1(\Omega)$	$R_2(\Omega)$	$R_3(\Omega)$	$C$ nF
82k	220	246k	3,3

### Diagrama de Bode

En base a la transferencia en Laplace se construye el diagrama de bode. En la Figura 4.18 se muestra dicho diagrama realizado en MATLAB. El primer armónico es en  $3fc = 35kHz$ , como se observa a esa frecuencia la atenuación ya es de aproximadamente 30dB con respecto a la ganancia en  $fc$ .

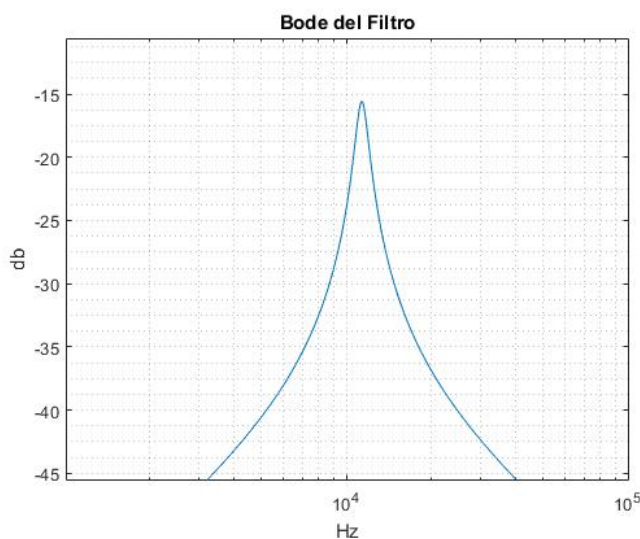


Figura 4.18: Filtro pasa banda MFB

### Simulación de filtro con LtSpice

Con los componentes seleccionados, se simuló el filtro. Se le introdujo una señal cuadrada de 6V de amplitud y frecuencia 11.8kHz como se ve (en verde) en la Figura 4.19. A la salida se obtuvo una sinusoide (en rojo) de la misma frecuencia que la cuadrada pero de amplitud 0.72V que es un poco menor a lo que se esperaba (1V). De todos modos la ganancia se deja regulable con un preset por lo tanto se puede ajustar a 1V.



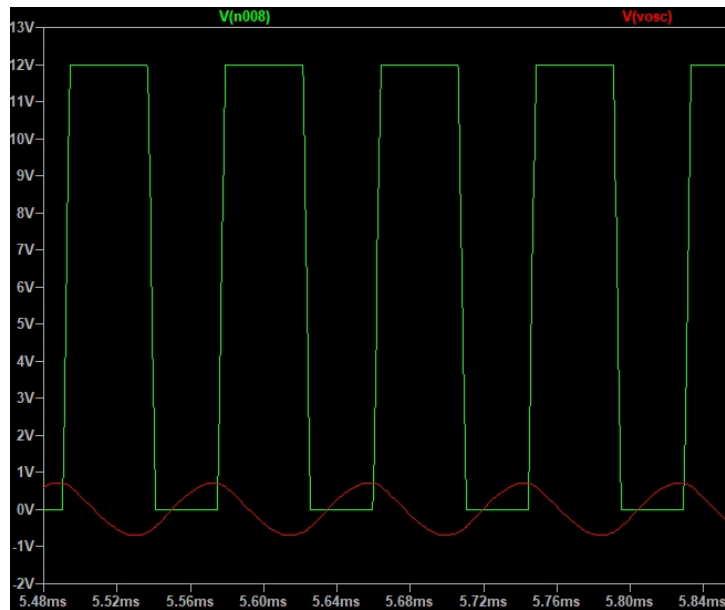


Figura 4.19: Filtro pasa banda MFB

## Multiplexor

El multiplexor tiene que cumplir con los siguiente requerimientos.

Se necesita un multiplexor que tenga por lo menos cuatro salidas.

La señal  $V_{seg}$  es la salida de el seguidor que está después del filtro, varía entre  $\pm 1V$ , por lo que el multiplexor tiene que poder excursionar entre  $\pm 1V$ .

Un multiplexor en conducción se modela con una resistencia asociada en serie ( $R_{on}$ ). Esta resistencia está en serie con la de la leche, por lo que tiene que ser lo suficientemente chica para que no afecte la medida. La resistencia mínima de la leche es:

$$R_{m_{min}} = \frac{l}{\sigma_{max} S} = \frac{1}{12 \times 10^{-3} \times 1} = 83\Omega$$

Como se tiene como objetivo un error del 3%, se calcula la resistencia máxima de conducción para que el error relativo sea del 3%.

$$0,03 = \frac{R_{on_{max}}}{R_{m_{min}}} \Rightarrow R_{on_{max}} = 2,5\Omega$$

Se elige el ADG1604 que cumple con dichos requerimientos.

En la Figura 4.20 se observa el diagrama del multiplexor.  $V_{SS}$  y  $V_{DD}$  son la alimentación y fijan los límites de excursión, según la la hoja de datos alimentándolo en  $\pm 5V$  la resistencia de conducción es de  $1,2\Omega$  máximo.

$A0$  y  $A1$  son pines digitales para elegir cuál de las cuatro salidas se conecta con la entrada. El pin digital  $EN$  conecta o desconecta la entrada.  $D$  es el pin de entrada de señal analógica y  $Sj$  son las salidas.

## Capítulo 4. Transductor de conductividad

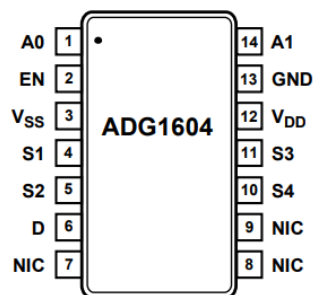


Figura 4.20: Pines del multiplexor. Imagen extraída de la hoja de datos del ADG1604 [3].

### Amplificador de conductividad

Este amplificador es un inversor por lo que la ganancia en valor absoluto es  $R_{10}/R_m$ . La  $R_m$  se puede calcular como:

$$R_m = \frac{l}{a \times h \times \sigma_m}$$

$$\Rightarrow G_c = \frac{a \times h \times \sigma_m}{l} R_{10} = \frac{\sigma_m}{K_{cell}} R_{10}$$

### Elección de componentes

Como la celda elegida tiene dimensiones  $(l, a, h) = (1\text{cm}, 1\text{cm}, 1\text{cm})$ , la ganancia es:

$$G_c = \sigma_m R_{10}$$

Como la señal de entrada es de amplitud 1V y se desea corresponder 12mS con 5V de amplitud, la ganancia debe ser de 5V/V. Despejando se encuentra que  $R_{10} = 417\Omega$ . Como se desea dejar regulable, se elije un preset de  $1k\Omega$ .

Tabla 4.9: Componentes seleccionados

$R_{10}(\Omega)$
1k

### Detector de picos

En Figura 4.21 se presenta el bloque detector de picos. Todas las  $R_j$  son de valor R. La ganancia es unitaria<sup>4</sup>. El error del detector está dado por la expresión:

$$error = 100 \times \left(1 - e^{-\frac{T}{RC}}\right)$$

<sup>4</sup>Por detalles de funcionamiento consultar Apéndice B.2.3

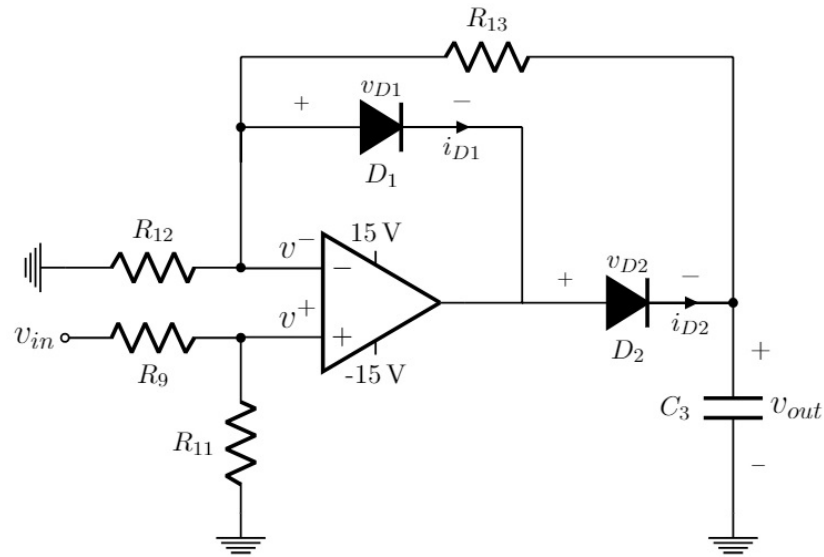


Figura 4.21: Diagrama del detector de picos

Donde  $T$  es el período de la sinusoide de entrada. Dado que la frecuencia del oscilador es  $11,8\text{kHz}$ , se tiene  $T = 84,7\mu\text{s}$ . Entonces para tener un error menor al  $3\%$ ,  $RC$  tiene que ser mayor a  $2,78\text{ms}$ .

Fijando  $R$  en un valor típico, por ejemplo  $10\text{k}\Omega$ , se necesita  $C_3 > 0,28\mu\text{F}$ . Se dispone de un condensador de  $1\mu\text{F}$ . Se utilizaron diodos de propósito general 1N4148.

Tabla 4.10: Valores de las componentes para el bloque

$R_9(\Omega)$	$R_{11}(\Omega)$	$R_{12}(\Omega)$	$R_{13}(\Omega)$	$C_3(\mu\text{F})$	Diodos
10k	10k	10k	10k	1	1N4148

### Simulación del detector en LTSpice

Se simuló el circuito con las componentes vistas en la Tabla 4.10 y señales de entrada de frecuencia  $10\text{kHz}$ . En la Tabla 4.11 se presenta los resultados de la simulación y el error relativo. Se puede obtener un estimador del error del detector de picos realizando un promedio con los mismos:  $e_{detector} = 2,8\%$

### Elección de amplificadores operacionales

Los amplificadores operacionales para amplificar correctamente deben cumplir ciertos requerimientos de slew rate, ancho de banda mínimo, offset máximo y tensión de alimentación. A continuación se hallan dichos requerimientos.

El slew rate para una sinusoide esta dado por la ecuación:

$$SR = 2\pi f \times V_{max}$$

## Capítulo 4. Transductor de conductividad

Tabla 4.11: Comportamiento detector de picos

$v_{in.pico}$ (V)	$v_{detector}$ (V)	error (%)
1,0	0,96	3,8
2,0	1,9	3,0
3,0	2,9	2,0
4,0	3,9	2,5
5,0	4,9	2,6

Como las señales a amplificar son sinusoides de 10kHz y amplitud 5V máximo<sup>5</sup>,  $SR = 2\pi \times 10^4 \times 5 = 3,14 \times 10^5 V/s$ , por lo tanto el SR del operacional debe ser

$$SR > 0,3V/\mu s$$

Se necesita un ancho de banda mínimo de 100kHz para que la senoide de 10kHz quede en banda pasante. La etapa con mayor ganancia es la del amplificador de conductividad que es de 5V/V, este bloque es de primer orden, por lo tanto el producto por ancho de banda debe ser

$$GBW > 500kHz$$

Se desea que la tensión de offset no influya en el error de la medida. Como el objetivo es tener un error relativo menor a 3%, se desea que el error relativo máximo producido por el offset sea despreciable por ende menor 0,3%. De la sección de requerimientos se tiene que la menor conductividad a medir es de  $1mS/cm$ , el voltaje de salida es proporcional a la conductividad y en  $12mS/cm$  vale 5V por lo tanto en  $1mS/cm$  vale 417mV. Como se desea que el offset a la salida sea menor al 0,3%, debe ser menor a  $417 \times 0,003 = 1,25mV$ . La etapa con ganancia mayor a 1 es la del amplificador de conductividad (5V/V) y dado que el offset a la entrada se puede ver como una fuente en la pata + del operacional, se tiene una configuración no inversora de ganancia 6V/V. Entonces la tensión de offset a la entrada debe ser

$$V_{off} < \frac{1,25}{6} = 208\mu V$$

Ya que se decide utilizar una fuente de computadora, el amplificador debe poder alimentarse entre  $\pm 12V$ .

Otro criterio que se tuvo en cuenta es que estén encapsulados de a cuatro operacionales para poder ahorrar espacio en el PCB.

Para cumplir con todos los requerimientos mencionados se selecciona el OPA4192 [1]:

- $SR = 20V/\mu s$
- $GBW = 10MHz$
- $V_{off} = 5\mu V$
- Supply voltage MAX  $\pm 20V$

---

<sup>5</sup>Amplitud máxima de entrada analógica del Arduino UNO

## Cable

Como se mencionó anteriormente, se utiliza un cable para conectar el circuito con las celdas. La resistencia de los conductores tiene que ser pequeña por que va en serie con las celdas y puede introducir error en la medida. Se procede a calcular la resistencia de cada conductor. Se sabe que la resistividad del cobre a  $25^{\circ}C$  es  $\rho = 17,1 \times 10^{-9} \Omega.m$ , el largo  $l = 3m$  y la sección  $S = 0,25 \times 10^{-6} m^2$  entonces:

$$R_c = \frac{\rho \cdot l}{S} = 0,205 \Omega \quad (4.14)$$

La resistencia mínima de la leche es de  $83 \Omega$ . Como una celda va conectada a dos conductores con resistencia  $R_c = 0,205 \Omega$ , se tiene que la resistencia total en serie es de  $0,41 \Omega$ . Por lo tanto el error relativo de la medida es de  $0,5\%$  que es menor al  $3\%$ .

## Transferencia total

Anteriormente se presentó la transferencia de cada bloque. Para una vista general de las dependencias es de interés calcular la transferencia total del transductor.  $v_{seg}$  es la amplitud de la salida del seguidor que esta después del filtro, esta pasa por el multiplexor que no genera cambios (se desprecia resistencia) y entra al amplificador de conductividad que tiene ganancia en valor absoluto  $R_{10} \sigma_m / K_{cell}$ . El detector de picos tiene ganancia 1 y  $v_{cond}$  es la salida del transductor. Por lo tanto la transferencia es:

$$v_{cond}(\sigma_m) = v_{seg} \frac{R_{10}}{K_{cell}} \sigma_m$$

Donde las unidades son las del SI. Si se regula  $v_{seg}$  a  $1V$ , la transferencia se reduce a:

$$v_{cond}(\sigma_m) = \frac{R_{10}}{K_{cell}} \sigma_m$$

### 4.4.2. Prueba

A continuación se presentan las pruebas realizadas a cada bloque del circuito de medición de conductividad.

## Oscilador

En la Figura 4.22 se muestra la salida del oscilador. Como se ve, es una onda cuadrada de  $11,0kHz$  y  $57\%$  de ciclo de trabajo que es próximo a lo que se esperaba.

Como se aprecia esta onda es cuadrada de  $5,2V$  de amplitud y una componente de continua de  $5,2V$ . Como la alimentación medida es de  $11,5V$  se esperaba que la amplitud fuera de  $5,8V$  aproximadamente. Se observa que se dista poco de la misma. Puede ser por el echo de que como el integrado está compuesto por transistores, la salida no puede excursionar en todo el rango de la alimentación por el voltaje de saturación, de todos modos esta variación no afecta por que se regula la amplitud con el filtro.

## Capítulo 4. Transductor de conductividad

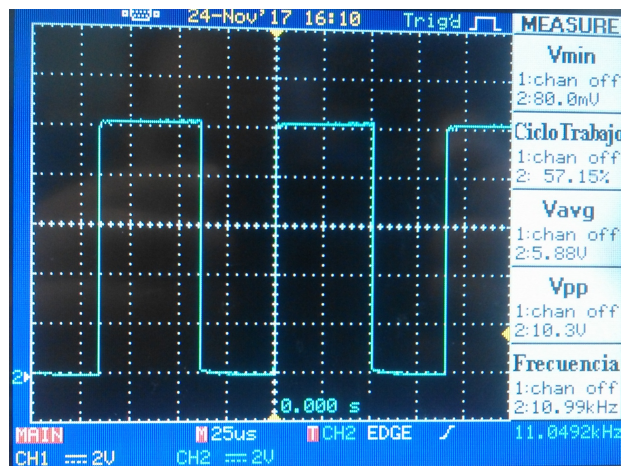


Figura 4.22: Salida del integrado LM555

## Filtro

En la Figura 4.23 se muestra la salida del filtro pasa banda, teniendo como entrada la salida del LM555. Se regula el preset ( $R_3 = 170k\Omega$ ) para que la salida sea de 1V de amplitud.

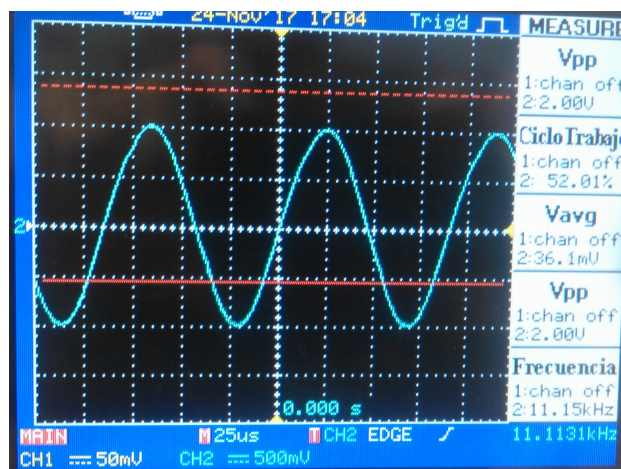


Figura 4.23: Salida del filtro pasa banda

## Multiplexor

Se verificó que copia la entrada a la salida seleccionada con los pines digitales y que la tensión de la salida es igual a la de entrada confirmando que la  $R_{on}$  es despreciable.

## 4.5. Integración de la pieza en el organo

### Amplificador de conductividad

Se conectó una resistencia de  $100\Omega$  entre una salida del mux (Mi) y Vm para simular la resistencia de la leche. Se corroboró que funciona, pero no se corrobora que la ganancia sea la adecuada, ya que no es de interés por que se regula en la prueba de campo.

### Detector de picos

En la Tabla 4.12, se encuentran los valores de pico de una señal de entrada a la que se le va variando la amplitud, la salida del detector y el error relativo calculado para los correspondientes valores.

Tabla 4.12: Comportamiento del detector de picos

$v_{in\_pico}$ (V)	0,505	1,00	1,51	1,99	2,49	2,96	3,50	4,00	4,50	5,00
$v_{detector}$ (V)	0,495	1,00	1,47	1,95	2,42	2,88	3,39	3,90	4,36	4,92
error (%)	1,98	0	2,65	2,01	2,81	2,70	3,14	2,50	3,11	1,60

El error máximo es 3,14% y el error promedio es 2,25% el cual es próximo al que se calculó en la simulación (2,8%).

### Cable y transductores de conductividad

El cable seleccionado se prueba para evaluar si introduce desfasaje o caída de tensión en la señal. Para ello se conecta en un extremo del cable, entre un conductor y tierra (la tierra va por un conductor y por la malla como se explicó anteriormente) una resistencia del orden de  $R_m$ . En el otro extremo del cable se inyecta una señal de 1V de pico y 10kHz entre el mismo conductor y tierra. Se observa con el osciloscopio la señal en el extremo del cable. No se encuentra diferencia apreciable entre la señal de entrada y la señal medida en el extremo del cable. Por lo tanto se concluye que no afectan las capacidades parásitas ni la resistencia del cable.

## 4.5. Integración de la pieza en el organo

### Puerto

Para la conexión entre el cable que va para el circuito de medición y los transductores se utiliza un puerto de tipo micrófono como se aprecia en la Figura 4.24. Se decide por utilizar este, ya que es robusto por ser metálico y por contener rosca para que no se desprenda el macho del hembra. En principio se utilizó un puerto Ethernet tipo jack pero se desprendía, por lo que es suplantado por este. La cantidad de vías <sup>6</sup> necesarias es 8. Este puerto cuenta con 10 vías por lo que sobran 2.

<sup>6</sup>en la subsección 3.6 se detalla qué es cada vía y cómo se mapea en el puerto

## Capítulo 4. Transductor de conductividad



Figura 4.24: Puerto

La parte hembra del puerto se instala en el organo perforando la pieza metálica como se puede ver en la Figura 4.25.a. Se agrega silicona en la unión para evitar que se pierda vacío en el organo.

Se suelda los cables al puerto hembra, se protege cada uno con vainas termo contraíbles, se agrega silicona para proteger donde no llega a cubrir bien la vaina y se le coloca una vaina termo contraíble más grande que cubra a todos juntos como se puede ver en la Figura 4.25.b.



#### 4.5. Integración de la pieza en el organo

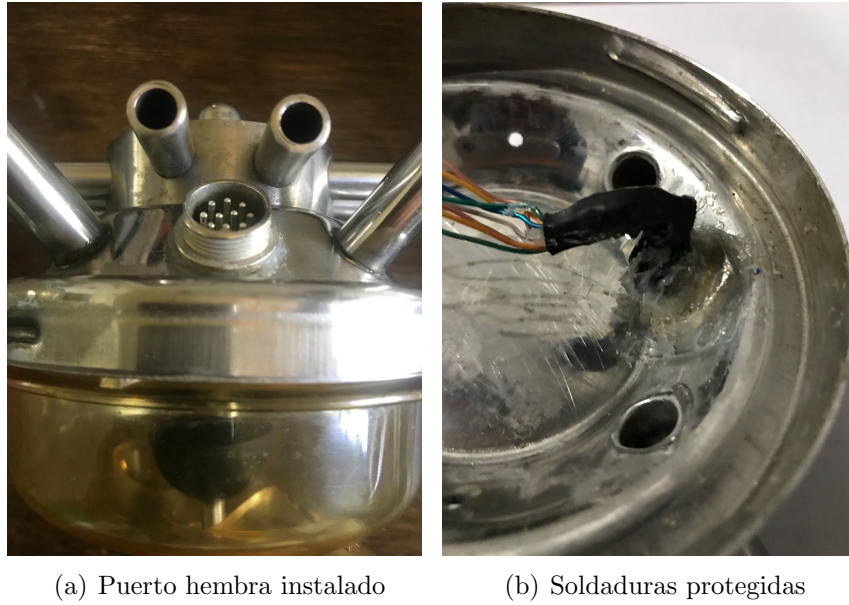


Figura 4.25: Puerto hembra

En la Figura 4.26 se muestra cómo queda el puerto macho ya soldado y protegido con la pieza metálica que trae.



Figura 4.26: Puerto macho

#### Piezas

Las piezas impresas quedan como se ve en las Figuras 4.27 a 4.30.

## Capítulo 4. Transductor de conductividad



Figura 4.27: Vista superior de pieza inferior impresa

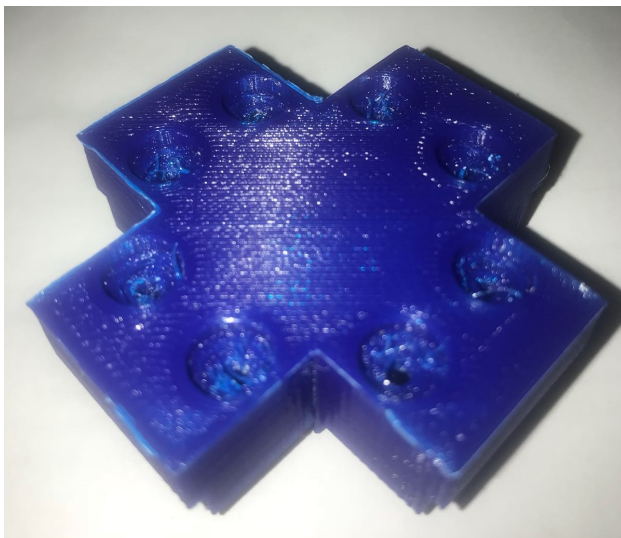


Figura 4.28: Vista inferior de pieza inferior impresa

#### 4.5. Integración de la pieza en el organo

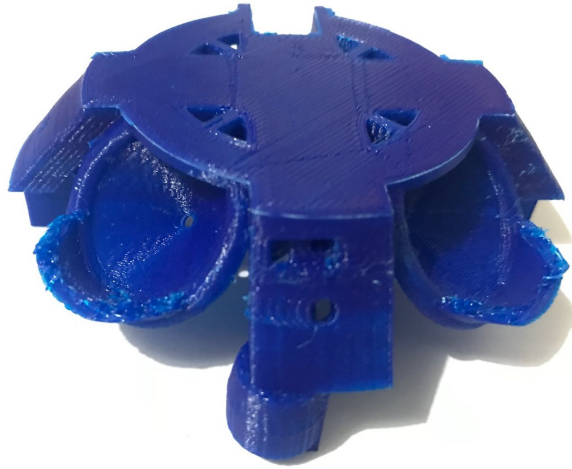


Figura 4.29: Vista superior de pieza superior impresa

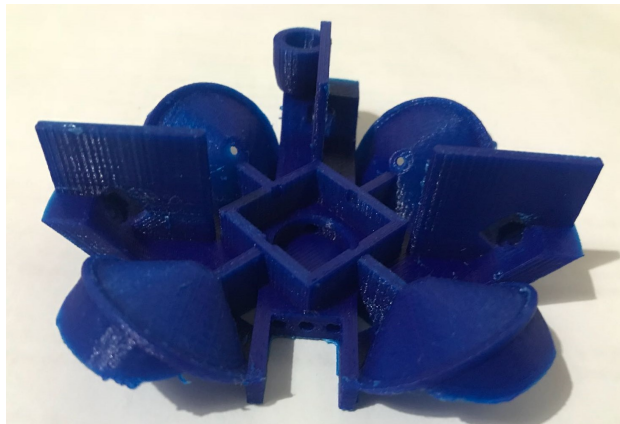


Figura 4.30: Vista inferior de pieza superior impresa

Los electrodos se construyen con chapa de acero inoxidable, se los recorta con tijera de cortar chapa y se les da terminación a lima. Las dimensiones son 1cm de alto por 1,1cm de largo de modo de poder incrustarlos en las ranuras hechas en las celdas. Puestos lucen como se ve en la Figura 4.31.

## Capítulo 4. Transductor de conductividad

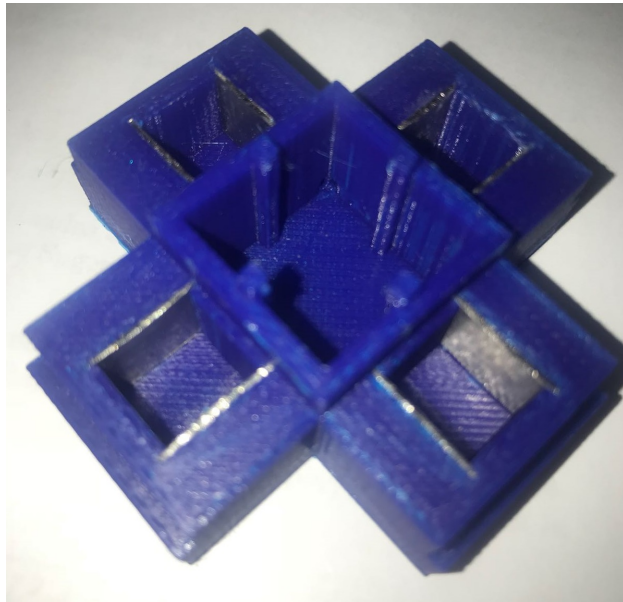


Figura 4.31: Electrodo puestos

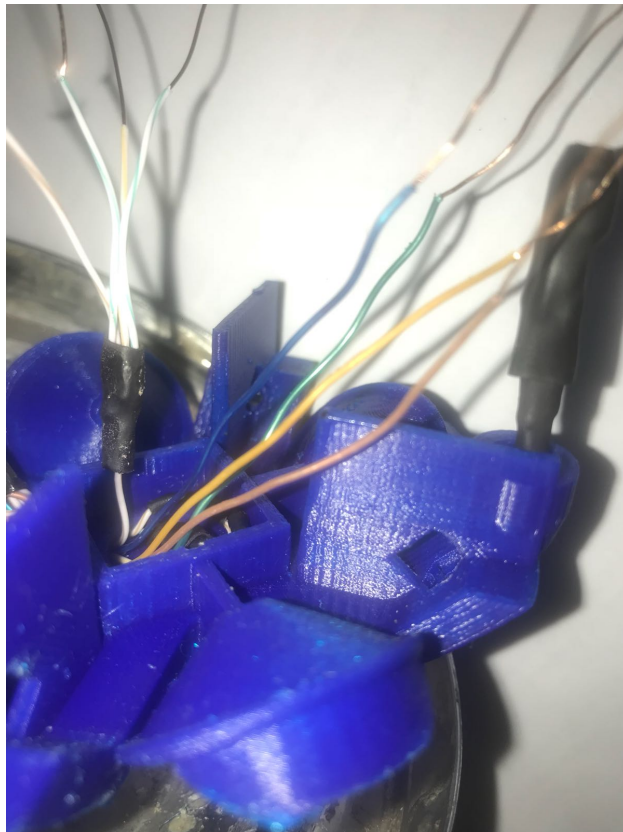
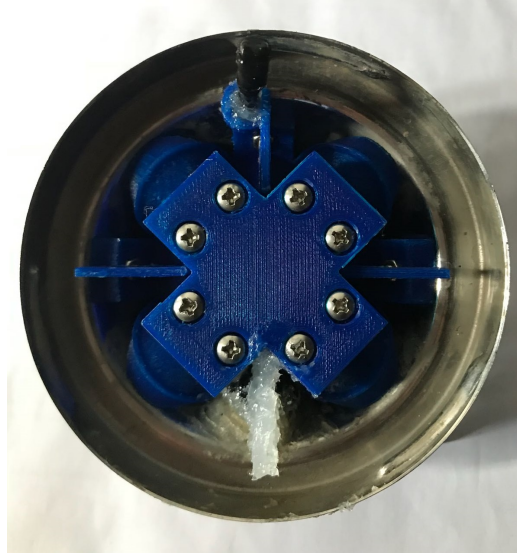


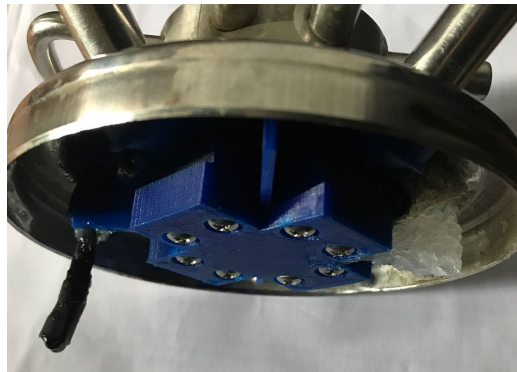
Figura 4.32: Pieza superior con cables y LM35

## 4.6. Calibración

La pieza superior con los cables y el LM35 protegido con termo contraíble se puede apreciar en la Figura 4.32.



(a) Vista superior



(b) Vista lateral

Figura 4.33: Piezas instaladas

Con los cables conectados a los electrodos con los tirafondos, la pieza superior atornillada al organo, las piezas pegadas y la pared de silicona, luce como se ve en la Figura 4.33. En la Figura 4.34 se muestra una vista de la UM completa.

## 4.6. Calibración

La tensión de salida del transductor es adquirida por el ADC del Arduino y enviada al LecPro. Es en el LecPro que se realiza la conversión de este valor numérico (0 a 1023) a un valor de conductividad (mS/cm).

## Capítulo 4. Transductor de conductividad

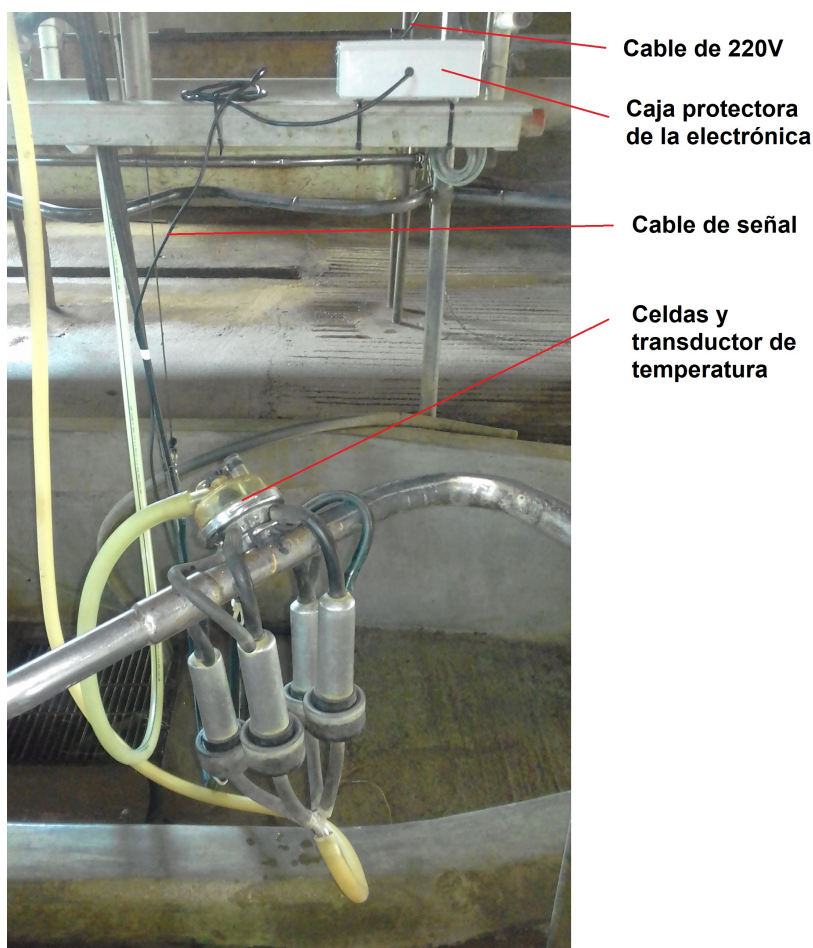


Figura 4.34: UM completa

Dicha conversión se realiza utilizando parámetros que deben ser calculados mediante el procedimiento de calibración. Se propone tres formas de calibrar y se obtiene una estimación del error. Para ello se releva los valores medidos en cada celda para conductividades conocidas mediante el siguiente experimento.

Se crean 5 soluciones distintas de NaCl disuelto en agua en el rango de interés (1mS - 12mS), para medir las soluciones se utiliza un conductivímetro Goodes *TDS&EC* E-1 Portable de precisión 0,1mS/cm. En la Tabla 4.13 se muestran las conductividades de cada solución.

Tabla 4.13: Soluciones de NaCl

Solución $N^{\circ}$	1	2	3	4	5
Conductividad (mS/cm)	1,7	3,2	4,7	6,9	9,3

Para simular el ordeño, estas soluciones se colocaron en botellas con un agujerito en cada tapa de manera de apretarlas y que salga un chorro finito continuo. Este chorro se emboca en los ductos del organo correspondiente a cada cuarto.

## 4.6. Calibración

En la UM se utiliza su software original. Sin embargo para el LecPro se implementa un software que despliegue en la terminal las muestras de conductividad enviadas por la UM y las almacene en un archivo de texto. No se realiza la compensación de temperatura por software porque todas las soluciones se mantienen a temperatura constante (temperatura ambiente). Se elige una celda y se comienza a echar la solución #5, cuando pasan aproximadamente 30 segundos se cambia a la solución #4 y así sucesivamente hasta llegar a la solución #1. Por último se limpia la celda con agua y se vacía para que la conductividad sea nula (aire). Luego se repite el procedimiento para otra celda de manera de relevar el comportamiento de las cuatro celdas.

Se numeraron las celdas correspondientes a cada cuarto. En la Figura 4.35 se muestra la serie que envía la UM cuando se realiza el procedimiento mencionado en la celda 3.



Figura 4.35: Cambios de solución celda 3

Claramente se ven los escalones correspondiente a los 5 cambios de soluciones. Se promedia cada uno cuando ya alcanzó su valor final para obtener al valor de ADC correspondiente a cada solución. Estos valores promedio se pueden ver en la Tabla 4.14.

## Ajuste

Con los puntos encontrados, para cada celda se realiza un ajuste lineal dado por la ecuación 4.15.

$$y = a \times x + b \tag{4.15}$$

## Capítulo 4. Transductor de conductividad

Tabla 4.14: Valores de ADC promedio al variar la conductividad en cada celda

Solución	1	2	3	4	5
Celda 1 (ADC)	57,6	109	163	214	274
Celda 2 (ADC)	45,3	79,5	125	174	232
Celda 3 (ADC)	51,1	92,5	140	195	245
Celda 4 (ADC)	51,8	94,4	141	199	252

Donde  $y$  esta en [mS/cm] y  $x$  es el número obtenido por el ADC.  $a$  y  $b$  son constantes a determinar. Para ello se utiliza la función `lsqcurvefit` de MATLAB. En la Figura 4.36 se observan las curvas ajustadas a los datos.

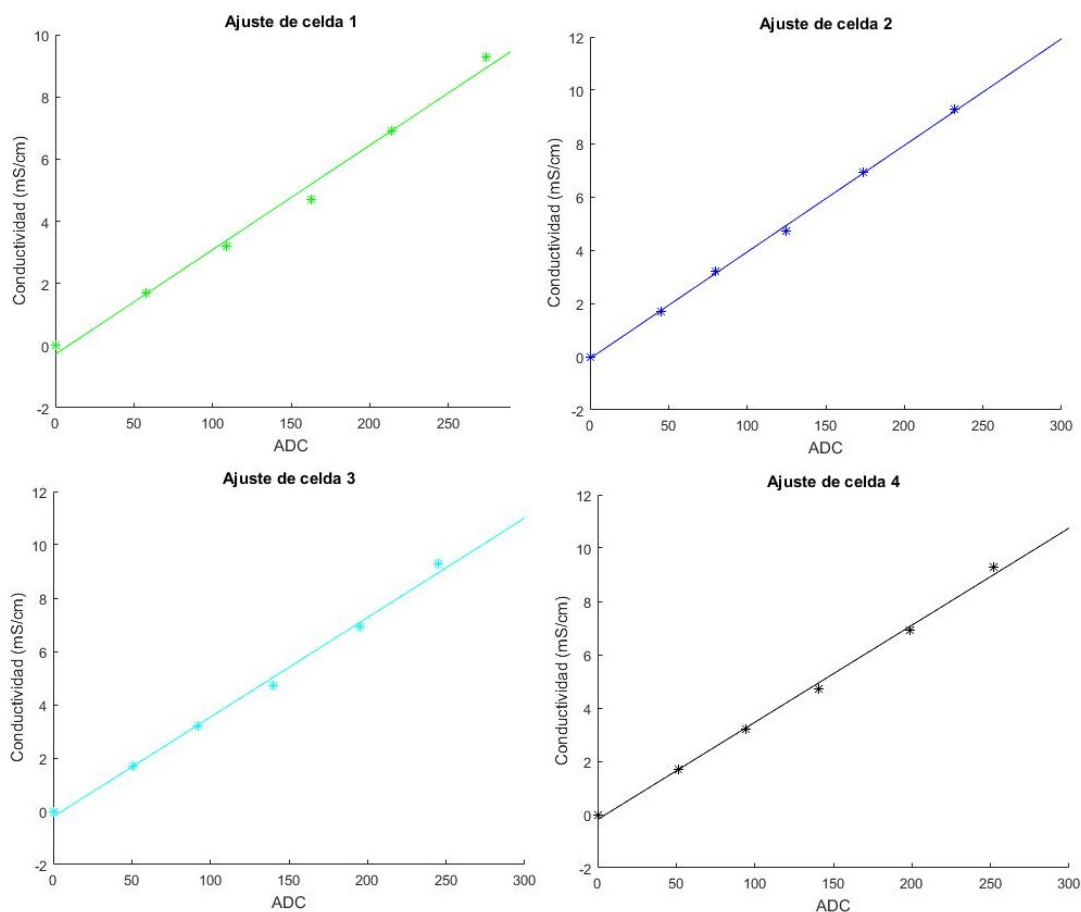


Figura 4.36: Ajuste de las celdas

En la Tabla 4.15 se presentan los parámetros de ajuste hallados para cada celda.

Con estos valores y la ecuación 4.15 se puede realizar la conversión de ADC a conductividad para cualquier punto con un error asociado que se estimará más



#### 4.6. Calibración

Tabla 4.15: Parámetros de ajuste

Celda	1	2	3	4
a	0.0336	0.0400	0.0374	0,0365
b	-0.280	-0.0733	-0.209	-0.186

adelante. Sin embargo este método tiene la desventaja que hay que caracterizar cada celda de cada UM para obtener los parámetros a y b de cada una pudiendo ser muy engorroso a gran escala.

En la Tabla 4.15 se puede apreciar que los valores calculados a y b tienen poca dispersión. Esto es debido a que las dimensiones de celdas son idénticas por diseño y simplemente tiene pequeñas diferencias constructivas por lo que si se utilizara una recta de calibración común a todas las celdas y el error resultante fuera bajo, simplificaría el procedimiento de calibración y uso del transductor.

La calibración común es la determinación de una recta única, correspondiente a una *celda media*.

Se estima la recta que caracteriza a la celda media con las cuatro celdas que se tiene caracterizadas. Para ello se ajusta todos los puntos de todas las celdas juntos por una recta como se ve en la Figura 4.37.

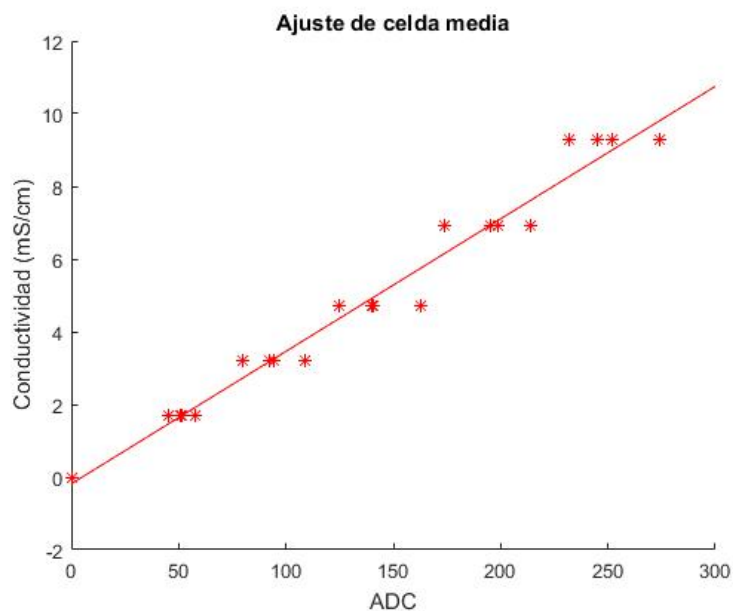


Figura 4.37: Ajuste de celda media

En la Tabla 4.16 se presentan los parámetros de ajuste de la celda media.

## Capítulo 4. Transductor de conductividad

Tabla 4.16: Parámetros de ajuste de celda media

<b>a</b>	0,0361
<b>b</b>	-0,118

### Cálculo de $K_{cell}$

Con el ajuste realizado es posible estimar el parámetro  $K_{cell}$  real ya que la transferencia es

$$V_{cond}(\sigma) = \frac{R_{10}}{K_{cell}}\sigma$$

En la Arduino, 5V corresponde a 1023 ADC, por lo tanto la transferencia se puede expresar en valores del ADC como:

$$x = \frac{1023}{5} \frac{R_{10}}{K_{cell}}\sigma \Rightarrow K_{cell} = \frac{1023}{5} R_{10} \frac{\sigma}{x}$$

Donde las unidades son las del SI .

Para la celda media,  $b = -0,118mS/cm$  que es bajo con respecto a los valores de conductividad de interés por lo tanto para un cálculo aproximado de  $K_{cell}$  se puede despreciar. La variable  $y$  esta en [mS/cm] por lo tanto para pasar a unidades del SI ([S/m]) hay que dividir por 10 ambos lados de la ecuación  $\Rightarrow \sigma \approx \frac{a}{10}x$ .

$$\Rightarrow K_{cell} \approx \frac{1023}{5} R_{10} \frac{a}{10} \quad (4.16)$$

$a = 0,0361$  y  $R_{10}$  se lo reguló en  $160\Omega$  por lo tanto:

$$K_{cell} = 118 m^{-1}$$

Mientras que el teórico calculado con las dimensiones es:

$$K_{cell} = \frac{l}{a \times h} = \frac{0,01}{0,01 \times 0,01} = 100 m^{-1}$$

Como se ve son parecidos.

### Cálculo del error

Se calcula el error asociado a la calibración por cada celda, por la celda media y por celda teórica. Para la celda teórica es necesario calcular el parámetro  $a$  a partir del  $K_{cell}$  teórico con la ecuación 4.16.

$$a = K_{cell} \frac{5}{1023} \frac{10}{R_{10}} = 100 \frac{5}{1023} \frac{10}{160} = 0,0305$$

El parámetro  $b$  de la celda teórica es cero.

Para calcular el error<sup>7</sup> (E) en porcentaje se utiliza la siguiente expresión:

---

<sup>7</sup>La expresión de  $y_{error}$  se extrajo de [23].

## 4.6. Calibración

$$E = 100 \times \frac{y\_error}{y}$$

$$y\_error = t_{\alpha/2, n-2} \cdot S \cdot \sqrt{\frac{1}{n} + \frac{n(x - \bar{x}_i)^2}{n\sum x_i^2 - (\sum x_i)^2}}$$

Donde  $y = a \cdot x + b$  y  $x$  es un valor de ADC donde se desea estimar el error.  $x_i$  son las muestras en ADC tomadas para calibrar,  $n$  es el número de muestras,  $\bar{x}_i$  es el promedio de las muestras,  $t_{\alpha/2, n-2}$  es el valor crítico de la distribución normal (en nuestro caso elegimos un intervalo de confianza del 95%  $\Rightarrow \alpha = 0,05$ ) y  $S$  es la desviación estándar de las muestras.

De la observación de las series obtenidas en la prueba de campo, los valores de ADC varían en general entre 200 y 400. Por lo que se toma este como rango de interés. Entonces para  $x$  en este rango, con un paso de 20 ADC, se calcula el error para cada celda como se puede ver en la Tabla 4.17.

Tabla 4.17: E.C1: Error celda 1, E.Cm: Error celda media, E.Ct Error celda teórica

$x$ (ADC/10)	20	22	24	26	28	30	32	34	36	38	40
<b>E.C1 (%)</b>	9,2	9,3	9,7	10	11	11	12	12	13	13	13
<b>E.C2 (%)</b>	7,5	7,6	7,9	8,3	8,8	9,2	9,6	10	10	11	11
<b>E.C3 (%)</b>	8,1	8,3	8,6	9,1	9,5	10	10	11	11	11	12
<b>E.C4 (%)</b>	8,3	8,5	8,8	9,3	9,7	10	11	11	11	12	12
<b>E.Cm (%)</b>	4,3	4,5	4,7	4,9	5,2	5,4	5,6	5,8	5,9	6,1	6,2
<b>E.Ct (%)</b>	9,7	10	11	11	12	12	13	13	13	14	14

Observe que el error en la celda media es menor que el de las celdas 1 a 4, esto no es por que tenga menor dispersión, es por que se tiene más muestras (para cada celda se tiene  $n=5$  y para la celda media se tiene  $n=20$ ). Para el cálculo del error de la celda teórica se utiliza las 20 muestras.

Como indicador del error para todo punto, se acota tomando el error máximo de la Tabla 4.17 para cada celda. Esto se puede ver en la Tabla 4.18

Tabla 4.18: Errores de calibraciones

	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>Cm</b>	<b>Ct</b>
<b>Error <math>\forall x</math> (%)</b>	13	11	12	12	6,2	14

En conclusión, el error de calibración por celda media es bajo (6,2%) por lo tanto es un método adecuado para utilizar. Sin embargo el de calibración celda teórica es un poco más grande (%14), este valor se podría bajar si se utiliza un método de construcción de electrodos más preciso.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 5

## Software de UM

### 5.1. Diseño de software

La UM debe realizar cuatro tareas principales: i) enviar y recibir datos vía Bluetooth, ii) controlar el multiplexor mediante señales digitales, iii) adquirir las cuatro muestras de conductividad y una de temperatura con un periodo de un segundo, y iv) señalar por medio de leds información estimada sobre el estado de la vaca, es decir, si se detectó mastitis o no. El software se estructuró con módulos que permiten abstraerse del hardware involucrado en cada tarea.

### 5.2. Descripción de Módulos

En la Figura 5.1 se presentan las bibliotecas creadas y externas<sup>1</sup>. Las flechas indican que se utilizan funciones del módulo al que apuntan. Sobre las mismas se encuentran los nombres de las funciones utilizadas de cada módulo. A continuación se dará una breve descripción de cada uno.

## Main

En este módulo se define la arquitectura de software propiamente dicha. En Arduino se cuentan con dos funciones: i) `setup()` que es la función que se llama una vez luego del reset, y ii) `loop()` que es llamada periódicamente constituyendo un loop infinito (de ahí su nombre) donde se consultan las banderas (definidas más adelante). Se crean las siguientes instancias globales:

- `HC-05()`: se crea la instancia de la clase `SoftwareSerial` definida en la biblioteca del mismo nombre. Cuenta con los métodos necesarios para inicializar, recibir y enviar datos al módulo HC-05.
- `Serial()`: se crea la instancia de la clase `Serial` que se encarga de las tareas referentes a la comunicación por medio de la UART del micro.

---

<sup>1</sup>Ambas bibliotecas vienen incluidas en el IDE de Arduino.

## Capítulo 5. Software de UM

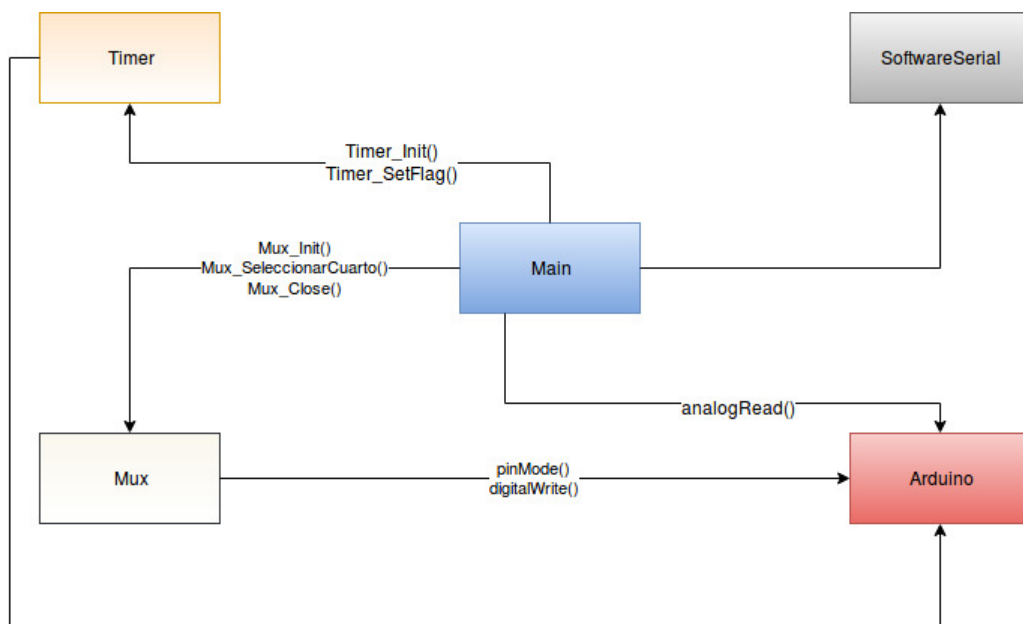


Figura 5.1: Diagrama de módulos. Las flechas salientes indican que se utilizan funciones de otros módulos. La biblioteca Timer utiliza registros definidos en la biblioteca Arduino. En el main se utiliza la clase SoftwareSerial definida en la biblioteca SoftwareSerial.

Se definen las variable globales:

- `mastitis`: booleano que se mantiene seteado por un tiempo predeterminado cuándo se está ante un caso de mastitis.
- `contador`: variable de tipo entera. Se incrementa una vez por segundo después que se detecta que una vaca tiene mastitis.
- `buff[]`: arreglo de tipo entero utilizado para almacenar las muestras de conductividad y temperatura durante un periodo.
- `banderaMuestreo`: booleano que es seteado en la rutina de interrupción (ISR<sup>2</sup>) del módulo timer.
- `ticks`: variable de tipo entera. Se incrementa cada 250 milisegundos. Indica que celda se está muestreando.

A continuación el pseudocódigo de la función `setup`:

```
void setup(void){  
  Mux_Init();  
}
```

<sup>2</sup>Interrupt Service Routine

```

Timer_Init ();
//se inicializa el multiplexor seleccionando la celda 0
Mux_SeleccionarCelda(0);
// Se inicializan las comunicaciones seriales de la UART y el
  HC05
Serial.begin(9600);
HC-05.begin(38400);
ticks = 0;
//Se inicializa el led
pinMode(LED_BUILTIN, OUTPUT);
digitalWrite(LED_BUILTIN, LOW);
}

```

A continuación se observa el pseudocódigo de la función `loop()`.

```

void loop(void){

    if (banderaMuestreo) {
        handlerMuestreo();
    }
    if (contador >= 20){
        handlerLeds();
    }
    if (HC-05.available()) {
        handlerNuevoMsj();
    }
}

```

Se pregunta por tres condiciones. La bandera **banderaMuestreo** es seteada cada 250 milisegundos por la ISR del timer. El manejador de esta bandera se define de la siguiente manera:

```

void handlerMuestreo(void){
    banderaMuestreo = 0;
    buff[ticks] = analogRead(A1);
    ticks++;
    if (ticks==4){
        buff[ticks] = analogRead(A0);
        ticks=0;
        SendBuff(buff);
        if (mastitis) contador++;
    }
    Mux_SeleccionarCelda(ticks);
}

```

La variable **ticks** es utilizada para indexar el buffer **buff**. Además, tiene como función controlar qué celda se muestrea en el instante<sup>3</sup> siguiente. La celda se selecciona llamando a la función `Mux_SeleccionarCelda()` pasándose como parámetro

<sup>3</sup>Se considera un instante al periodo de 250 milisegundos

## Capítulo 5. Software de UM

dicha variable. El sensor de temperatura está conectado a la entrada analógica A0 y el sensor de conductividad esta conectado a la entrada A1. El ADC tiene una resolución de 10 bits, es por esto que el buffer donde se almacenan las muestras es de tipo entero (2 bytes).

Cuando la variable **ticks** es igual a cuatro se está en el último cuarto del periodo por lo que después de muestrear la última celda se procede a muestrear la temperatura. Una vez que esto se realiza, se llama a la función **SendBuff()**. La misma se encarga de armar el paquete de datos<sup>4</sup> y enviarlo al módulo HC-05 por comunicación serial. Finalmente, si la variable **mastitis** se encuentra seteada se incrementa el contador.

Se verifica la condición **HC-05.available()** cuando hay un mensaje nuevo por parte del HC-05. Se definen los mensajes:

- Inicio: Al recibirse este mensaje se llama a la función *Timer\_SetFlag(int\*)* pasándose como parámetro la dirección de memoria de la variable bandera-Muestreo.
- HayMastitis: Se setea la variable mastitis a 1 y la variable contador a 0. Además, se enciende el led.

La condición **contador**  $\geq$  **20** se verifica luego de 20 segundos después que se encendió el led. El manejador de está bandera consiste en apagar el led y setear la variable mastitis a cero.

## Timer

Dado que se requiere muestrear cada 250 milisegundos, es necesario configurar un timer que permita cambiar **banderaMuestreo** con esa cadencia. El Arduino UNO tiene tres timer. Timer0 y timer2 son de 8 bits, mientras que timer1 es de 16 bits. El reloj del sistema es de 16 MHz. El máximo valor del contador para los timer de 8 bits es 256, mientras que para el de 16 bits es de 65536. Se necesita interrumpir cada 250 milisegundos. Se tienen cinco posibles valores de prescaler (1, 8, 64, 256, 1024). Si se selecciona 1024, la frecuencia del reloj del sistema dividido el prescaler da 15624 Hz. Dividiendo el resultado entre la frecuencia deseada (4 Hz) se tiene 3906.25. Lo que es menor a 65536 pero mayor a 256 por lo que lo más conveniente es utilizar el timer1. Se hace notar que se debe redondear el valor 3906.25 a 3906. Debido a este redondeo se tiene un error de 57.6 milisegundos por hora. Se tiene además la incertidumbre introducida por la frecuencia nominal del Arduino. Estos errores de desincronización son despreciables para esta aplicación.

Se configura el timer en modo CTC. Este modo permite que se llame a la ISR cuando el contador interno del timer alcanza el valor almacenado en el registro de comparación(3905<sup>5</sup> en este caso). Una vez que se alcance este valor, se limpia el contador en el siguiente intervalo de reloj y se reinicia el conteo.

---

<sup>4</sup>En 6.3 se detalla la estructura del mismo.

<sup>5</sup>La cuenta inicia en 0



El módulo está compuesto por:

**Variables:**

- *flag\_timer*: puntero privado al módulo de tipo entero.

**Funciones:**

- *Timer\_Init()*: Se encarga de configurar el timer 1 para que interrumpa cada 250 milisegundos.
- *Timer\_SetFlag(int\*)*: Asigna la dirección de memoria pasada como parámetro a el puntero privado al módulo (*flag\_timer*). De esta manera, al setear el lugar de memoria al que apunta el puntero privado, se esta seteando las variable externa al módulo (banderaMuestreo).
- Rutina de atención de interrupción del timer 1: Se setea la dirección de memoria a la cual apunta el puntero privado.

## Mux

Se encarga de manejar los pines de control del multiplexor.

**Variables:**

- *S0*: variable de tipo entero. Pin de control. Habilita o no el multiplexor.
- *S1*: variable de tipo entero. Pin de control.
- *S2*: variable de tipo entero. Pin de control.

**Funciones:**

- *Mux\_Init()* : Configura los pines digitales que manejan el multiplexor. Además habilita el mismo mediante el pin de enable.
- *Mux\_SeleccionarCelda()* : Recibe como parámetro la celda cuya conductividad se desea adquirir. A partir de dicho parámetro, setea los pines de control para seleccionar el canal del multiplexor correspondiente.
- *Mux\_Close()* : deshabilita el multiplexor seteando el pin *S0* a 0.

## SoftwareSerial

La UART del micro está conectada a los pines digitales 0 y 1 (Rx, Tx). Estos pines están conectados al USB que se utiliza para cargar el programa y para debugging.

El módulo HC-05 tiene una interfaz serial para comunicarse. Ahora, dado que el hardware UART del Arduino en un principio se utiliza para comunicarse con la computadora para desplegar mensajes de debugging, no es posible utilizar la misma para comunicarse con el módulo. Para esto, se utiliza la biblioteca SoftwareSerial.

## Capítulo 5. Software de UM

La misma, emula el comportamiento de una UART por software permitiendo utilizar cualquier pareja de pines digitales<sup>6</sup>(a excepción del 0 y 1 en este caso) para transmitir y recibir. Esta biblioteca cuenta con la clase `SoftwareSerial`, a cuyo constructor se le deben pasar los pines digitales a utilizar como Rx y Tx. Se utiliza un tamaño de dato de 8 bits. Esta clase presenta los métodos:

- `available()`: se recibió un nuevo dato
- `read()`: leer nuevo dato
- `write()`: escribir dato
- `begin()`: permite setear velocidad de la comunicación serial

---

<sup>6</sup>En el Arduino UNO

# Capítulo 6

## Hardware y Software de LecPro

### 6.1. Hardware de LecPro

Para la plataforma del LecPro se selecciona una Raspberry PI 3. Por dos razones. Primero, porque permite tener las funcionalidades de un computador moderno a un bajo precio (alrededor de 35 dólares americanos). Segundo, porque levanta el sistema operativo Linux en el cuál se pueden utilizar las bibliotecas de clasificación provistas por *scikit-learning*. Se utiliza como distribución de Linux, Ubuntu Core 16 para Raspberry PI 3.

Esta plataforma cuenta con una CPU quad-core de 64 bits ARM Cortex A53 con un reloj de 1.2 GHz. Además cuenta con Bluetooth 4.1, WIFI, cuatro puertos USB y un puerto Ethernet. Se requiere 5 V via MicroUSB o GPIO header para su alimentación.

Por otro lado, se utiliza un dongle USB dado que se requiere tener una aplicación servidor Bluetooth y a su vez conectarse como cliente a las UM (no es posible con un único dispositivo Bluetooth mantener conexiones como cliente y servidor en simultáneo).

Debido a que la sala de ordeño se lava con agua a presión, es necesario proteger el Raspberry del agua. Para ello se utiliza una caja de registro eléctrico con grado de protección IP65 <sup>1</sup>. La tapa contiene una junta de goma para que no entre el agua como se ve en la Figura 6.1.a. La misma se cierra con tornillos. Se utiliza un transformador 220/5 V para alimentar el Raspberry, por lo que se le hace una perforación a la caja para introducir el cable. Además se necesita otra apertura para el cable Ethernet que se conecta a una computadora para extraer los datos. Se coloca los cables y se rellena las perforaciones con masilla epóxica para que no entre agua como se aprecia en la Figura 6.1.b. El Raspberry se fija a la caja con tornillos. En la Figura 6.1.c se muestra una vista general de todos los componentes

---

<sup>1</sup>Este grado de protección indica que: El polvo no debe entrar bajo ninguna circunstancia. No debe entrar el agua arrojada a chorro (desde cualquier ángulo) por medio de una boquilla de 6,3 mm de diámetro, a un promedio de 12,5 litros por minuto y a una presión de 30 kN/m<sup>2</sup> durante un tiempo que no sea menor a 3 minutos y a una distancia no menor de 3 metros.

del LecPro.

### 6.2. Diseño de Software

El LecPro se debe encargar de mantener una comunicación confiable con las UM, además de procesar los datos que recibe de las mismas. Por otro lado, en forma simultánea, debe procesar los comandos que recibe desde la aplicación móvil. Para realizar estas tareas en paralelo se emplean tres procesos, de manera de poder utilizar 3 de los 4 core que tiene la Raspberry PI 3. El proceso padre, y dos procesos hijos. Un proceso hijo, se encarga de lanzar un servidor para atender la aplicación del móvil, mientras que el otro proceso hijo esta encargado de procesar los datos que recibe por parte de las UM y mantener la comunicación con las mismas. En este proceso hijo, se lanza un *thread* (hilo) por cada UM de manera de poder atender a cada una en forma simultánea e independiente. El proceso padre procesa los comandos ingresados a través del proceso hijo que levanta el servidor móvil y los transmite al proceso hijo encargado de gestionar las UM.

A continuación se describe la estructura de carpetas:

- **src**: directorio con los archivos fuente.
- **Modelo**: directorio donde se almacena el modelo de clasificación.
- **logs**: directorio donde se almacenan los logs del sistema.
- **Datos**: directorio donde se almacenan las muestras de cada vaca (un archivo por vaca).
- **main.py**: archivo fuente que inicia el sistema. El mismo es ejecutado automáticamente al encenderse el Raspberry.
- **dispositivos.txt**: interfaz que indica cuáles son las UM a las que hay que conectarse (direcciones Bluetooth).

#### 6.2.1. Diagrama de Módulos

Se define un módulo para cada una de las tareas principales. Estas son, mantener el servidor móvil corriendo y gestionar las UM. Además, se emplea un módulo extra como capa de abstracción de la tecnología de comunicación utilizada para con las UM y otro para encapsular las funciones referentes al procesamiento de las series temporales. Por último, se tiene el módulo main, en el cuál se implementa la lógica de control del sistema. El diagrama de módulos y cómo se comunican entre ellos se observa en la figura 6.2.



(a) Tapa



(b) Vista interior



(c) Vista general del LecPro

Figura 6.1

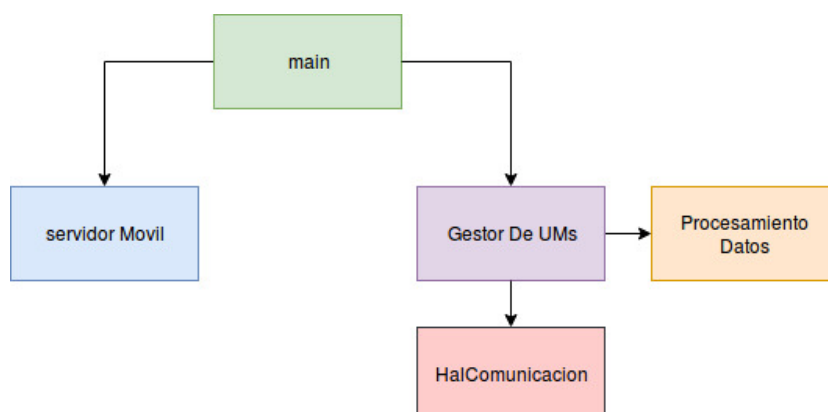


Figura 6.2: Diagrama de módulos del LecPro. Las flechas salientes indican que se utilizan funciones de otros módulos

El sistema tiene como eje central la atención de la UM. Esto consiste en procesar los datos que se reciben de la misma de manera de poder determinar que datos corresponden a un ordeño de una vaca y cuáles al intervalo entre ordeños. Una vez que se determina cuáles son los datos que conforman la serie temporal de conductividad eléctrica de una vaca, se utilizan las funciones del módulo Procesamiento para extraer las características<sup>2</sup> correspondientes de la misma. El modelo de clasificación a partir de estas características determina si se está ante un caso de mastitis o no, enviándose el comando correspondiente a la UM respectiva.

### 6.3. Implementación de Software

Como lenguaje se utiliza Python 3.5. Para la comunicación entre procesos se utiliza la función `Pipe()` de la biblioteca **multiprocessing**. Esta función retorna una pareja de objetos que representan ambos extremos de una tubería (Pipe). Los mismos poseen los métodos **send()**(enviar) y **recv()**(recibir). La comunicación entre ambos objetos es duplex, sin embargo en esta aplicación se utiliza comunicación simplex ya que el sistema se diseña de manera que el proceso encargado de manejar el servidor móvil, envíe mensajes al proceso principal y éste envíe mensajes al proceso encargado de gestionar las UM.

#### Main

En este módulo se define la subclase **procesoHijo** para manejar los procesos hijos. Esta clase es heredada de la clase **Process** de la biblioteca **multiprocessing**.

---

<sup>2</sup> Descriptores que representan la información del estado de la ubre de manera más clara que las series temporales crudas.

```

class procesoHijo(multiprocessing.Process):
    def __init__(self, fun, *args):
        multiprocessing.Process.__init__(self)
        self.pipe_parent, self.pipe_child = multiprocessing.Pipe()
        self.my_function = fun
        self.my_args = args

    def getPipe(self):
        return self.pipe_parent

    def run(self):
        self.my_function(*self.my_args)

    def join(self, timeout=None):
        multiprocessing.Process.join(self, timeout)
        self.pipe_parent.close()

```

Esta subclase define cuatro métodos adicionales. El método `__init__()`, es el constructor de la misma. El mismo recibe como parámetros de entrada la función a ejecutar en el proceso hijo, y sus parámetros. Se utiliza el operador `*` para indicar que se recibe una lista de argumentos de tamaño arbitrario. El constructor se encarga de inicializar los atributos de la clase. Se tiene cuatro atributos, la pareja de objetos generados por la función `Pipe()`, `self.pipe_parent` y `self.pipe_child`. Además, se tiene `self.my_function`, el cuál es un puntero a la función a ejecutar en el proceso hijo y `self.my_args` el cuál almacena la lista de argumentos a pasarle a la función anterior. Todos los atributos cuentan con la palabra reservada `self`, para indicar que los atributos son únicos a cada instancia creada.

El método `getPipe()`, retorna el objeto de un extremo de la tubería al proceso padre. El método `run()`, invoca a la función `self.my_function`. Este método sobrescribe al método `run()` de la clase `Process`. Este método, es invocado en un proceso aparte cuando se llama al método `start()` de la clase `Process`. Por último, se tiene el método `join()`, el cuál se utiliza para que el proceso padre, espere a que el proceso hijo finalice para así poder liberar los recursos utilizados para comunicar ambos procesos.

Se crean dos instancias de la subclase `procesoHijo`. Una para generar el proceso encargado de administrar el servidor móvil y otra para generar el proceso encargado de gestionar las UM. A continuación se presenta el pseudocódigo del módulo `main`:

```
#1) Inicializacion proceso servidor movil
servidorMovil = procesoHijo(funcServidor , paramsServidor)
servidorPipe = servidorMovil.getPipe()

#2) Inicializacion proceso gestorUM
gestorUM = procesoHijo(funcGestorUM , paramsGestorUM)
gestorPipe = gestorUM.getPipe()

#3) Inicio servidorMovil
servidorMovil.start()

#4) bucle
while True:
    try:
        msj = servidorPipe.recv()
        handlerMensaje(msj)
    except:
        handlerException()
        break

# 5) liberacion de recursos
gestorUM.join()
servidor.join()
```

En el main se inicia creando las instancias de la subclase **procesoHijo**. En el paso 3 se inicia el proceso referente al servidor móvil (el proceso encargado de gestionar las UM es iniciado luego que el usuario ingresa el comando de inicio del sistema).

En el paso 4 se tiene el bucle principal. En este bucle, el sistema queda bloqueado esperando a recibir mensajes del proceso referente al servidor móvil. En caso de recibir un mensaje se realiza la acción correspondiente. Hay dos maneras de salir del bucle principal. Una es cuando ocurra algún tipo de excepción. Si esto sucede, se registra cuál fue el tipo de excepción en el log del sistema antes de salir del bucle. El segundo caso, es cuando se recibe el mensaje de finalización del sistema.

En el paso 5 se espera a que finalicen los procesos hijos para así liberar los recursos utilizados de manera adecuada.

## HalComunicacion

Este es el módulo que define la clase para la comunicación Bluetooth con las UM. Python provee una biblioteca para comunicarse con dispositivos Bluetooth, **pybluez**. Sin embargo, no sé logró conectarse a múltiples dispositivos en simultáneo. Razón por la cuál se decide utilizar la aplicación **rfcomm** de linux. Esta aplicación permite conectarse a dispositivos Bluetooth remotos utilizando el protocolo RFCOMM(Radio Frequency Communications). Este protocolo emula la comunicación serial por un puerto RS-232. Primero en el archivo de configuración de la aplicación /etc/Bluetooth/rfcomm.conf se configura la interfaz emulada del puerto serie a utilizar. A continuación se muestra un ejemplo



```
rfcomm0 {
# Automatically bind the device at startup
bind no;
# Bluetooth address of the device
device AA:BB:CC:DD:EE:FF;
# RFCOMM channel for the connection
channel 1;
# Description of the connection
comment "This is Device 1's serial port.";
}
```

Al comienzo se indica el nombre de la interfaz serial que se creará en el directorio `/dev`. En este caso `rfcomm0`. Entre los brackets, se indica la configuración del mismo. La línea **bind no**, indica que no se cree la interfaz serial al inicio del sistema. Se define de esta manera para emular la clase **BluetoothSocket** definida en `pybluez` (la cuál cuenta con el método `bind`). La línea **device AA:BB:CC:DD:EE:FF**, indica cuál es el dispositivo Bluetooth local a utilizar. A continuación se tiene la línea **channel 1**. Esto indica cuál es el canal lógico a utilizar o puerto RFCOMM. El protocolo RFCOMM permite realizar más de 60 conexiones simultaneas entre dos dispositivos Bluetooth. Este campo permite seleccionar una de ellas. Sin embargo, para esta aplicación este campo no es relevante ya que el dispositivo HC-05 solo dispone de un único servicio.

Para crear la interfaz serial `/dev/rfcomm0` que enlaza el dispositivo Bluetooth local con el dispositivo Bluetooth remoto `XX:XX:XX:XX:XX:XX` se ingresa el siguiente comando

```
$ sudo rfcomm bind rfcomm0 XX:XX:XX:XX:XX:XX
```

Sin embargo, los dispositivos aún no están conectados. Para conectarlos se debe abrir la interfaz `/dev/rfcomm0` con alguna aplicación serial. Al realizar esta operación el driver Bluetooth se encarga de establecer la conexión entre los dispositivos Bluetooth.

De esta manera para conectarse a múltiples dispositivos Bluetooth remotos se crean las interfaces correspondientes y se abren las mismas utilizando la aplicación serial correspondiente. Sin embargo, se presenta un inconveniente, solo es posible formar una red Bluetooth de hasta 7 dispositivos conectados de manera simultanea. Uno es el máster y los otros 6 son los esclavos que están conectados a él. Este inconveniente podría ser resuelto agregando dongles USB y utilizando su dirección Bluetooth respectiva en el archivo de configuración.

A continuación se define la clase `BluetoothRFCOMM`:

```

class BluetoothRFCOMM:

    def __init__(self,x,y):
        self.bd_addr = x
        self.device = y

    def bind(self):
        os.system("sudo rfcomm bind "+ self.device+" "+self.bd_addr)

    def connect(self):
        self.serialDevice = serial.Serial(
            port="/dev/"+self.device,
            baudrate=115200,
            parity=serial.PARITY_NONE,
            stopbits=serial.STOPBITS_ONE,
            bytesize=serial.EIGHTBITS,
            timeout = 1
        )

    def write(self,argument):
        self.serialDevice.write(argument)

    def read(self,cantidad):
        return self.serialDevice.read(cantidad)

    def close(self):
        self.serialDevice.close()

    def __del__(self):
        os.system("sudo rfcomm release "+self.device)

```

Al crear una nueva instancia de la clase **BluetoothRFCOMM** se le pasan como parámetros de entrada al constructor la dirección del dispositivo Bluetooth remoto (x) y el nombre de la interfaz a crear en el directorio /dev/ (y). Seguidamente se llama al método bind() de la misma para crear la interfaz serial. Este método ejecuta el comando bind de la aplicación rfcomm en un subshell.

Para conectar los dispositivos Bluetooth, se llama al método connect(). Este abre la interfaz serial creada. Para esto se utiliza la clase **serial** definida en **py-Serial**. Se configura la comunicación serial para transmitir a 38400 baudios, sin bit de paridad, 1 bit de parada y 8 bit de datos. Además se configura para que el método read() de la clase serial sea no bloqueante con un timeout de 1 segundo.

Se definen los métodos write y read para escribir y leer datos. Luego se tiene el método close() para cerrar el puerto serial. Estos tres métodos simplemente encapsulan los métodos de la clase serial.

Por último se tiene el destructor de la clase, el cuál libera el enlace realizado por el comando bind de la aplicación rfcomm.

En este módulo se define también la función **dispositivos()**. Esta función obtiene del archivo dispositivos.txt las direcciones Bluetooth de las UM a las cuáles

hay que conectarse así como su identificador y las retorna.

## Gestor De UM

Este módulo es el encargado de controlar la comunicación con las UM y procesar los datos recibidos por las mismas.

Se crea un *thread* para cada UM para manejar las tareas asociadas a la misma. Para el manejo de *threads* en Python existe la clase ***threads*** definida en la biblioteca ***threading***. Al crear la instancia de la clase se ingresan como parámetros de entrada, la función que se desea ejecutar con sus parámetros. Se cuenta con el método ***start()*** para iniciar el *thread* y con el método ***join()*** para esperar a que finalice el *thread*. A continuación se observa un ejemplo de uso de la misma donde se crean dos *threads* que ejecutan una función para imprimir en la salida estándar.

```
def imprimir(id):
    print(id)

if __name__ == "__main__":
    # Numero de \textit{thread}s
    \textit{thread}s = 2
    #lista donde se almacenan las instancias
    jobs = []
    for i in range(0, \textit{thread}s):
        \textit{thread} = \textit{thread}ing.\textit{thread}
        }(target=imprimir(i))
        jobs.append(\textit{thread})

    #Se inicia cada treads
    for j in jobs:
        j.start()

    # Se espera a que todos los \textit{thread}s finalicen
    for j in jobs:
        j.join()
```

El pseudocódigo del gestor es el siguiente:

```
def gestorDeUM(conn):

    #1) se obtiene identificadores de UM.
    mac, ids = HalComunicacion.dispositivos()
    #2) se inicializa arreglo memoria compartida
    initMemoriaCompartida(ids)
    #3) se inician \textit{thread}s para atender cada UM
    \textit{thread}s = init\textit{thread}s(mac, ids)
    #4) bucle principal
    while True:
        msj = conn.recv()
        handlerMensaje(msj)

    #5) Esperar a finalizacion de \textit{thread}s hijos
    for t in \textit{thread}s:
        t.join()
    conn.close()
```

La primera tarea que se realiza en el gestor es determinar con cuántas UM hay que conectarse y cuáles son sus identificadores. Para esto se llama a la función **dispositivos()** definida en el módulo **HalComunicacion**. Los identificadores se utilizan para indexar la memoria compartida y para identificar los *threads*.

El segundo paso a realizar es inicializar la memoria compartida para comunicarse con los *threads*. Para la comunicación entre el *thread* principal y los hijos se utiliza un diccionario con tantos elementos como *threads* hijos se tengan. Se utiliza este tipo de dato porque es posible realizar la indexación por strings. En este caso se utiliza el identificador de la UM, el cuál es retornado por la función **dispositivos** de **HalComunicacion**. Cada elemento del diccionario es un arreglo de dos posiciones. En la primera posición, se tiene el dato que el *thread* principal desea enviar al *thread* hijo correspondiente. La segunda posición se utiliza como bandera. La misma se setea a uno, después que el *thread* padre escribe un nuevo dato. Una vez que el *thread* hijo consumió el dato, setea la bandera a cero.

Para evitar problemas de datos compartidos, se utiliza la primitiva de sincronización **Lock()** de *threading*. Esta primitiva posee dos estados, trancado (“locked”) y destrancado (“unlocked”). Se inicializa en el estado destrancado. Tiene dos métodos principales, **acquire()** y **release()**. Cuando el estado es destrancado, la llamada al método **acquire()** cambia el estado a trancado y retorna inmediatamente. Cuando el estado es trancado, **acquire()** bloquea hasta que la llamada al método **release()**, en otro *thread*, cambia el estado a destrancado. Entonces, la llamada a **acquire()** cambia el estado a trancado y retorna. Cuando más de un *thread* es bloqueado en **acquire()**, solo un *thread* continúa su ejecución cuando la llamada a **release()** cambia el estado a destrancado [?].

La primitiva **Lock()** es inicializada en **initMemoriaCompartida()** como variable global al módulo. Además también es inicializada en **False** la variable global **stop**.

Seguidamente se inicializan los *threads*. Estos ejecutan la función **UMhandler()**. Esta función se encarga de atender a la UM correspondiente. Se divide en dos partes, una encargada de la recepción de los paquetes enviados por la UM y

### 6.3. Implementación de Software

la otra del procesado de los datos encapsulados por los mismos.

Finalmente se tiene el bucle principal del gestor. El *thread* principal permanece bloqueado a la espera que el proceso padre le envíe un comando. Cuando esto ocurre, lo atiende. Se definen dos tipos de comandos: FIN, indica que se requiere que finalice el sistema. Para esto en el handler, se setea a True la variable global stop. La misma es poleada permanentemente por los *threads* hijos. Luego de setear la variable stop, se sale del bucle. El segundo comando es CAR XX. Este indica que se tiene la caravana de alguna de las vacas que se este ordeñando (XX es el identificador de la UM correspondiente). Por lo que almacena en la memoria compartida la misma y se setea a uno el flag de nuevo dato.

El finalizar el bucle se espera a que los *threads* hijos finalicen. Cuando esto sucede, se finaliza el gestor.

Se habrá notado que no se profundizó en la descripción de la función **UM-handler()**, la cuál es una parte fundamental del gestor. A continuación se explica en detalle.

Como se explica anteriormente, esta función tiene dos tareas principales. Recibir un nuevo paquete y procesarlo. A continuación se tiene el pseudocódigo de la misma.

```
def UMHandler(device ,name):
    initComunicacion ()
    while True:
        try:
            if (hayPaquete()):
                convertirDato ()
                procesarDato ()

            if stop:
                break

        except IOError:
            handlerException ()
    closeComunicacion ()
```

La primera tarea de la función es inicializar la comunicación con la UM. Para esto se utiliza la clase **BluetoothRFCOMM** definida en el módulo *HalComunicacion*. Esto consiste en crear el *socket*<sup>3</sup>, realizar el proceso de *binding* y conectarse con el dispositivo *Bluetooth* deseado.

Luego se tiene el bucle. En el mismo se *polea* la función de **hayPaquete()** y la variable global **stop**. Se utilizan la directiva **try...except** para manejar excepciones<sup>4</sup> debido a problemas con la comunicación (**IOError**). Ejemplo de este tipo de excepciones pueden ser que ocurra un timeout en la conexión o que no halla recursos Bluetooth disponibles. Si ocurre una excepción del tipo **IOError** se realiza un intento de reconexión.

<sup>3</sup>Un socket representa un extremo en un canal de comunicación

<sup>4</sup>Existen dos tipos de errores en python: errores de sintaxis y excepciones. Las excepciones se deben a errores en tiempo de ejecución y en algunos casos no son fatales. Por lo que puede ser de interés manejarlas adecuadamente.

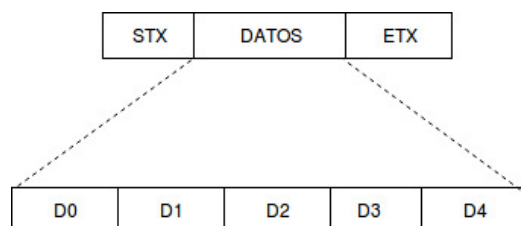


Figura 6.3: Trama del paquete enviado por UM

En la Figura 6.3 se observa la trama del paquete enviado por la UM. El paquete está compuesto por 22 bytes, 20 de datos y 2 que señalizan inicio y fin del mismo.

Los bytes de datos se dividen en 5 secciones (1 sección para cada dato de conductividad y 1 para el dato de temperatura). La UM utiliza el tipo de dato entero (dos bytes) para almacenar estas variables. Sin embargo, si se envían los mismos sin codificar, el inicio de trama no es distinguible de un byte de dato. Para evitar esto, se mandan los bytes en formato hexadecimal pero como *char*. Es decir, supongamos que se requiere enviar el entero 1024 en notación decimal. Su expresión en hexadecimal es 0400. Por lo que se envían los char '0','4','0' y '0'.

La función **hayPaquete()** se encarga de decodificar el paquete que se reciba y retornar True. En caso de que no se reciba ningún paquete, retorna False.

Si se recibe un paquete, se realiza la conversión de los datos. Los datos son los valores enteros de tensión adquiridos por sensor de temperatura y conductividad. Los mismos deben ser convertidos a las unidades adecuadas utilizando los parámetros obtenidos en la sección 4.6. Además las conductividades se deben convertir a su valor de conductividad equivalente a temperatura ambiente. Estas conversiones son realizadas por la función **convertirDato()**.

A continuación se ejecuta la función **procesarDato()**. La misma debe determinar si un nuevo dato<sup>5</sup> recibido es válido o inválido<sup>6</sup> y realizar las acciones correspondientes. En la Figura 6.4 se observa el árbol de decisión para un nuevo dato. Si el nuevo dato es válido y el dato anterior no fue válido, se crea un archivo de texto. El nombre sigue el formato "IdUM\_TimeStamp.txt"<sup>7</sup>. Tanto si el dato anterior es válido o inválido se almacena el nuevo dato en el archivo correspondiente y se setea el flag **DatoAnteriorValido** a uno. Si el nuevo dato es inválido y el anterior es inválido se setea el flag **DatoAnteriorValido** a cero. Si por el contrario, el dato anterior es válido, implica que acaba de terminar el ordeño de una vaca. Por lo tanto, se procede a clasificar la misma. Se envía el resultado a la UM, se setea el flag **DatoAnteriorValido** a cero y en caso de que se disponga del número de caravana del animal se cambia el nombre del archivo en el cuál se

<sup>5</sup>Para la función **procesarDato()**, un dato hace referencia a las 4 muestras de conductividad en mS a temperatura ambiente

<sup>6</sup> Se define un dato como válido, si se adquiere mientras la vaca se esta ordeñando. Los datos inválidos, son aquellos que se adquieren cuando el órgano que esta sensando la UM no está conectado a ninguna vaca.

<sup>7</sup>El **TimeStamp** es un entero que representa la hora y fecha actual en segundos desde el 1 de enero de 1970.

### 6.3. Implementación de Software

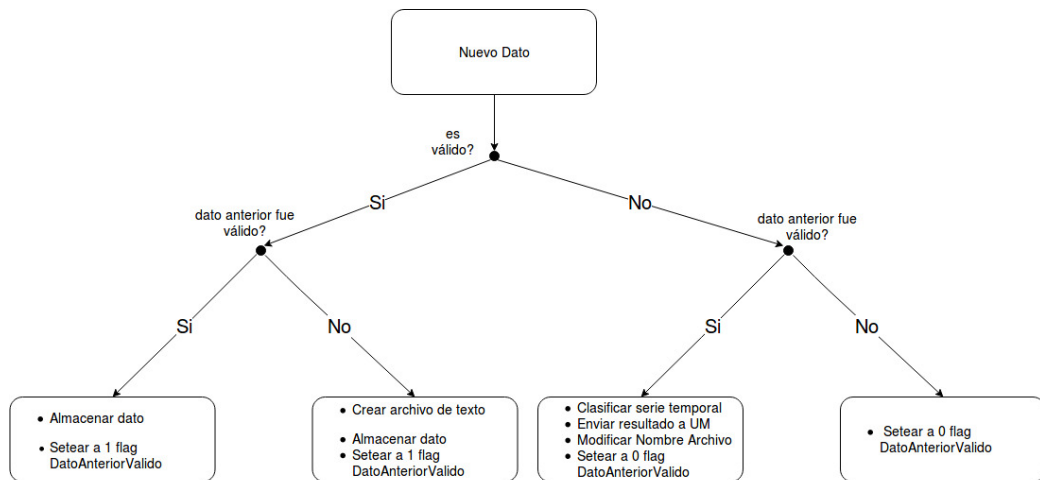


Figura 6.4: Árbol de decisión para un nuevo dato

almacenan los datos válidos por “IdUM\_CARAVANA.txt”, donde caravana es un entero.

A continuación se observa el extracto de código de la implementación del árbol de decisión para un nuevo dato.

```
def procesarDato(nuevoDato):  
  
    nonlocal DatoAnteriorValido, filename, name, clf  
    global pathDatos, lock, memCompartida  
  
    if (EsValido(nuevoDato)):  
        if not DatoAnteriorValido:  
            filename = pathDatos + name + "_" + str(TimeStamp()) + ".txt"  
  
            writeDataNewCow(nuevoDato, filename)  
            DatoAnteriorValido = True  
        else:  
            if DatoAnteriorValido:  
                #Clasificar Serie Temporal  
                series = np.genfromtxt(filename, dtype=float, delimiter=",")  
                características = procesamiento.transfCar(series)  
                pred = clf.predict(características)  
                #Enviar resultado UM  
                if pred == 1:  
                    #tiene mastitis  
                    sock_blu.write(b'1')  
                else:  
                    #no tiene mastitis  
                    sock_blu.write(b'0')  
                #Modificar Nombre archivo si corresponde  
                lock.acquire()  
                if memCompartida[name][1] == 1:
```

```
    cambiarNombreArchivo( filename , memCompartida[name][0] )
    memCompartida[name][1] = 0
    lock.release()

DatoAnteriorValido = False
```

La función **procesarDato()** recibe como parámetro de entrada una lista de cuatro elementos. Cada elemento representa la conductividad de cada cuarto. La palabra reservada **nonlocal** indica que las variables no son locales a la función si no que pertenecen al entorno en el cuál se define la misma. Por otro lado, la palabra reservada **global** indica que la variable es global en el módulo. Para determinar si un dato es válido se implementa la función **EsValido()**. La misma utiliza un buffer circular de diez muestras<sup>8</sup> donde se almacenan los últimos diez datos recibidos, válidos e inválidos. La lógica para determinar si un dato es válido o inválido depende de la categoría del dato anterior. Se considera que inicia el ordeño de la vaca cuando la media de las últimas diez conductividades de algunos de los cuartos supera un umbral. Este umbral es necesario ya que las celdas en las cuáles se miden las conductividades no quedan totalmente vacías al finalizar el ordeño. Esto hace que circule una corriente entre los electrodos, obteniéndose una conductividad no nula. Se toman diez conductividades para garantizar que el ruido no afecta a la clasificación del dato. Por lo tanto, si el dato anterior no es válido pero la media entre las últimas diez conductividades de alguno de los cuartos (considerando el nuevo dato) supera el umbral, se clasifica el nuevo dato como válido. Se considera que el ordeño finaliza cuando todas las conductividades de las últimas diez muestras son menores al umbral. Se utiliza esta lógica ya que puede pasar que las celdas se vacíen por unos instantes. Por ejemplo, cuando la vaca patea el órgano. Por lo que si el dato anterior es válido, pero todas los elementos del buffer (con el nuevo dato) son menores al umbral, se considera que el nuevo dato es inválido. El estado del dato anterior se almacena en la variable **DatoAnteriorValido**. Si es true, el dato anterior es válido. Es inválido en caso contrario.

La función **writeNewDataCow()** agrega el nuevo dato al archivo. Si no existe el archivo, se crea. Para acceder a la memoria compartida se utiliza la primitiva **lock** como se menciona anteriormente.

## Procesamiento

Este módulo se encarga de las tareas referentes a la clasificación de las series temporales. Para eso se cuenta con dos funciones. Una se encarga de cargar el modelo de clasificación en memoria **-load\_modelo()**- y otra se encarga de transformar las series temporales en características más adecuadas **-transfCar()**-. Como se construye el modelo y que transformaciones se realizan se detalla en la sección 6.4.

---

<sup>8</sup>Una muestra esta conformada por cuatro conductividades



## Servidor Móvil

Este módulo es el encargado de las tareas referentes al servidor Bluetooth. Para esto se utiliza la biblioteca **pybluez**. Pybluez utiliza el modelo de socket. Antes de poder utilizar un socket para enviar y recibir datos es necesario realizar una serie de pasos. Primero, se crea utilizando la clase **BluetoothSocket**(protocol) provista por pybluez. El constructor de la clase toma como parámetro el protocolo de transporte a utilizar. En este caso RFCOMM. Luego es necesario asignar los recursos del sistema operativo con el socket. Es decir, que tarjeta Bluetooth se utiliza y puerto. Para esto se cuenta con el método `bind()`, el cuál recibe como parámetros la dirección del dispositivo Bluetooth y el puerto. Luego es necesario poner el socket en modo escucha para que sea posible aceptar conexiones entrantes. Para esto, se cuenta con el método `listen(backlog)`. Este método requiere un único parámetro. Este parámetro se utiliza para saber cuántas conexiones entrante el sistema operativo debe aceptar (y mantenerlas en el estado pendientes) antes de que la aplicación del servidor las acepte. En este caso se setea a 1 porque solo se espera una conexión entrante (app móvil). El último paso antes de enviar y recibir datos en una aplicación servidor para Bluetooth es la aceptación de la conexión entrante. Esto se realiza con el método `accept()`. En esta biblioteca, el método no recibe parámetros pero retorna dos objetos. El primer objeto es una nueva instancia de **BluetoothSocket** conectada al cliente. El segundo, es un tupla que contiene la dirección y puerto al cuál se conectó el cliente. Luego de aceptada la conexión se utiliza los métodos `recv()` y `send()` del socket conectado al cliente para leer y enviar datos. Al finalizar se llama el método `close()` para liberar los recursos. A continuación se tiene el pseudocódigo de la aplicación del servidor:

```
def servidorMovil():

    server_sock=BluetoothSocket(RFCOMM)
    server_sock.bind((mac,PORTANY))
    server_sock.listen(1)
    client_sock , client_info = server_sock.accept()
    while True:
        try:
            msj = client_sock.recv()
            handlerMensaje(msj)

        except IOError:
            client_sock , client_info = server_sock.accept()

    client_sock.close()
    server_sock.close()
```

El bucle de la aplicación queda bloqueado esperando un nuevo mensaje por parte del cliente. Se definen tres mensajes:

- START: Inicio de Sistema

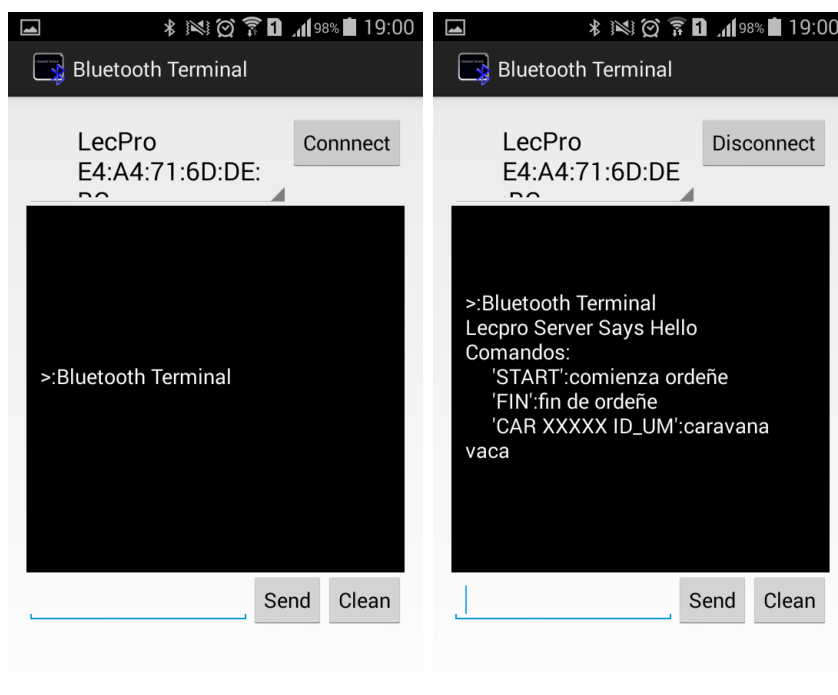
## Capítulo 6. Hardware y Software de LecPro

- FIN: fin de sistema
- CAR XXXXX UM\_ID: caravana de la vaca que se esta sensando por la UM\_ID.

Todos estos mensajes son pasados directamente al proceso padre utilizando la clase **Pipe()** de la biblioteca **multiprocessing**. Si ocurre una excepción relacionada a la comunicación Bluetooth (por ejemplo el cliente se desconecta), la aplicación queda esperando una nueva conexión.

### App Móvil

Desde el dispositivo móvil se utiliza la app gratuita Bluetooth Terminal<sup>9</sup>. En la Figura 6.5.a se observa la pantalla de inicio de la aplicación. En la Figura 6.5.b se observa la pantalla luego que se conecta al LecPro. Como se observa, se despliega un mensaje de bienvenida y se despliegan los comandos que el usuario puede ingresar.



(a) Ventana de inicio de la aplicación

(b) Mensaje de bienvenida desplegado por el servidor luego de conectarse al mismo

Figura 6.5: Aplicación móvil Bluetooth Terminal

<sup>9</sup> La misma fue creada por Juan Sebastian Ochoa Zambrano y se encuentra disponible para android en el Play Store.

## 6.4. Clasificación

En esta sección se detalla que criterio se utiliza para construir el modelo de clasificación. Además, se explica qué características se extraen de las series temporales de CE.

### Medidores de desempeño.

A continuación en la Tabla 6.1 se definen los términos utilizados para la clasificación de resultados:

<b>TP</b>	Casos de mastitis correctamente clasificados
<b>FP</b>	Casos sanos clasificados erróneamente
<b>FN</b>	Casos de mastitis clasificados erróneamente
<b>TN</b>	Casos sanos correctamente clasificados

Tabla 6.1: Términos utilizados para la clasificación de resultados

Se definen los siguientes estimadores:

$$\text{Precisión} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Tasa de Acierto} = \frac{TP + TN}{TN + TP + FP + FN}$$

$$\text{Precisión clase negativa} = \frac{TN}{TN + FN}$$

$$F_1 = 2 \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$F_1$  es una métrica que combina las métricas Precisión y Recall. Cuando estas métricas son similares,  $F_1$  puede ser considerada aproximadamente como la media de ambas.

### Base

A partir de las pruebas de campo fue posible armar una base de datos de 95 muestras distintas<sup>10</sup>. De estas 95 muestras, se tienen etiquetadas como sanas 63 muestras y 32 se encuentran etiquetados como casos de mastitis. Para cada muestra se tiene las cuatro series temporales de CE<sup>11</sup>. Además se cuenta con la siguiente información proporcionado por Santiago de Izaguirre (propietario del tambo): número de partos de la vaca, y el porcentaje de grasa y proteína para la

<sup>10</sup>Las pruebas de campo tomaron varios días. Debido a esto algunas vacas pasaron por el organo en el cuál se encontraba la UM más de una vez.

<sup>11</sup>En algunos casos no se ordeñan los cuatro cuartos. En dichos casos se registro la cantidad de cuartos ordeñados

## Capítulo 6. Hardware y Software de LecPro

mezcla de leche. Se requirió ir varias veces al tambo ya que por ordeño solo fue posible adquirir un máximo de 12 muestras (solo se cuenta con una UM en uno de los 23 puestos).

Se etiqueta una muestra como enferma si el RCS (Recuento de Células Somáticas) es mayor a 400.000 células/mililitro. Se cuenta con el RCS de la mezcla de leche y no de cada cuarto.

En la Figura 6.6 se observa las series temporales de CE para una muestra con mastitis.

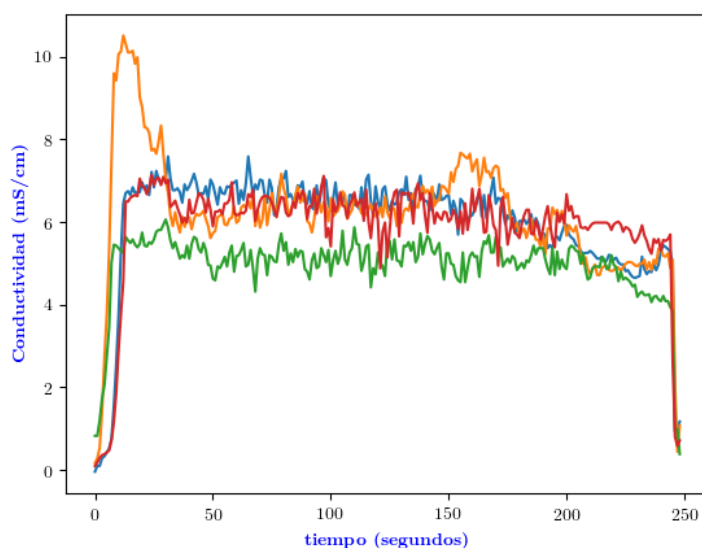


Figura 6.6: Series temporales de CE. La mezcla tiene un RCS de 3.399.000 células/mililitro.

## Evaluación

Para obtener los parámetros óptimos del algoritmo de clasificación se dividen los datos en dos conjuntos, uno de entrenamiento y otro de prueba. A partir de los datos del conjunto de entrenamiento se buscan los parámetros óptimos. Una vez obtenidos, se clasifica el conjunto de prueba. Esta división se realiza manteniendo la proporción entre las clases. Además el conjunto de prueba está conformado por un 20% del conjunto original siendo de 19 muestras. Por lo tanto el conjunto de entrenamiento está conformado por 76 muestras.

Para evaluar los resultados del conjunto de entrenamiento se realizan 10 validaciones cruzadas en 10 subconjuntos. Una validación cruzada en 10 subconjuntos consiste en dividir el conjunto de datos en 10 subconjuntos, entrenar el clasificador con 9 subconjuntos y clasificar el décimo. Luego cambiar el subconjunto a clasificar por otro de los 9, y entrenar con los 9 restantes. Esto, hasta clasificar los 10 subconjuntos. Finalmente se promedian los indicadores de desempeño obtenidos para cada clasificación de los subconjuntos. Realizar 10 validaciones cruzadas en 10 subconjunto es iterar el procedimiento descrito 10 veces. Se utiliza la técnica

de validación cruzada para obtener una idea del rendimiento del algoritmo más cercana a la realidad que en el caso de solo utilizar un único conjunto de prueba. Por ejemplo, puede suceder que el conjunto de prueba no sea representativo del espacio de muestras que nos interesa clasificar. En este caso al clasificar las muestras de prueba se tendrán malos resultados ya que el algoritmo fue optimizado utilizando datos con información distinta a los datos que se desean clasificar. La técnica de validación cruzada intenta solucionar este problema ya que se utilizan varios subconjuntos de prueba.

Para evitar el sobre-entrenamiento en la validación cruzada. Es decir, que en la optimización del algoritmo se utilice información extraída de los datos de prueba, cada vez que se entrena el algoritmo con un nuevo conjunto se buscan cuáles son los parámetros óptimos para el subconjunto de datos de entrenamiento de turno.

Como se mencionó anteriormente, la base de datos fue dividida en dos subconjuntos. Esto fue así para tener una idea de cómo serían los resultados al probar el sistema en el tambo. Como se aclaró, este subconjunto de prueba puede no ser necesariamente representativo por lo que se considera que las estimaciones obtenidas a partir de la validación cruzada (VC) reflejan mejor el rendimiento del algoritmo.

### Pre-procesamiento

Debido a algunos inconvenientes prácticos algunos datos no fueron debidamente almacenados. Por ejemplo, cuando una vaca se quita el órgano con una patada. En este caso, el sistema identifica que la conductividad de las celdas disminuye por debajo del umbral por lo que da por finalizado el ordeño. Luego, cuando el tambero vuelve a colocar el órgano a la vaca, el sistema interpreta que se trata de una nueva vaca. Por lo que para la misma vaca se tienen dos archivos distintos. Las últimas 10 muestras del primer archivo y las primeras 10 muestras del segundo archivo son eliminadas y se crea un nuevo archivo en el cuál se colocan las muestras de los dos archivos anteriores de manera consecutiva.

Sucede también que en algunos casos al terminar un ordeño, alguna de las celdas no se vacían debido a que el órgano queda inclinado en lugar de horizontal. En consecuencia, se tienen las series temporales de CE de dos vacas en un mismo archivo. Visualmente se identifica qué datos corresponden a la primera vaca, cuáles a la segunda y cuáles corresponden al intervalo entre vacas. Se almacenan los datos de cada vaca en archivos separados y se descartan los datos correspondientes al intervalo entre vacas.

### Clasificador

Random Forest es un clasificador que combina distintos árboles de clasificación entrenados con el mismo conjunto de entrenamiento utilizando un conjunto de características seleccionadas de manera aleatoria para cada árbol. Se entiende por árbol de clasificación a la estructura en forma de árbol que permite describir datos para predecir su clase. Cada hoja del árbol representa una etiqueta y las ramas representan las decisiones que se tomaron en cada nodo para llegar a dicha hoja.

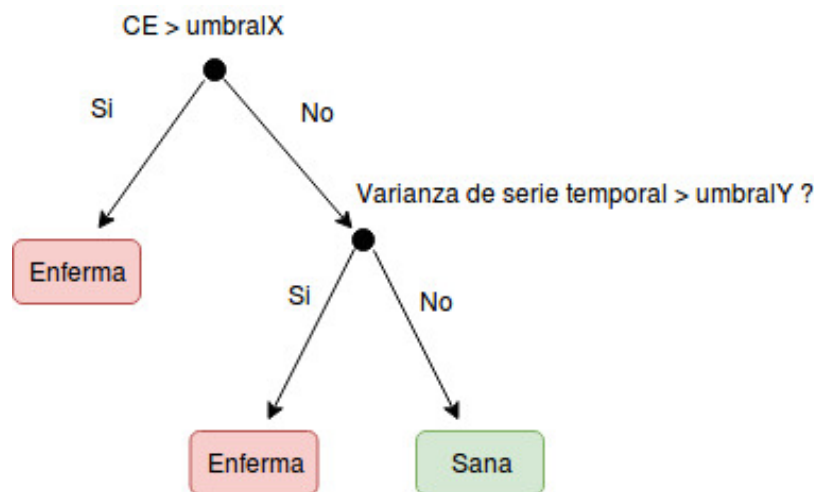


Figura 6.7: Árbol de decisión. Ejemplo artificial.

La característica que se utiliza en cada nodo para decidir que ramificación sigue la muestra es aquella que maximice el decremento de impureza de los subconjuntos generados, es decir, que maximice la pureza de los subconjuntos generados. En este trabajo se utiliza el criterio de impureza de Gini. En la Figura 6.7 se observa un ejemplo de árbol en donde se clasifican muestras (vacas) como **sanas** o **enfermas**. Si la conductividad eléctrica es mayor a cierto **umbralX**, se considera que es una muestra enferma. Si es menor, pero la varianza es mayor a cierto **umbralY**, también se clasifica como enferma. En caso contrario, se considera que la muestra es sana.

La cantidad de características dentro del conjunto de características seleccionadas ( $m$ ), junto con la cantidad de árboles ( $n$ ) son algunos de los parámetros del clasificador. Una vez construidos los árboles para la predicción de una nueva muestra, Random Forest hace lo siguiente: clasifica a ésta con cada árbol y la etiqueta que más veces se repita será la que se le asignará a la muestra.

Suponiendo que se tienen  $N$  muestras de entrenamiento con  $M$  características, se utiliza el siguiente algoritmo para construir cada árbol:

1. Se toman  $N$  muestras de manera aleatoria con remplazo del conjunto de datos original. Con las cuáles se entrenará.
2. Se eligen  $m \ll M$  características de manera aleatoria, las cuáles serán utilizadas de a una para la división de cada nodo. Se selecciona aquella que maximice el decremento en impureza.
3. Se crecen los árboles hasta la máxima extensión posible. No se realiza podado.

Breiman et al. [8] muestra que la tasa de error depende de la correlación entre los árboles y de la fuerza<sup>12</sup> de cada uno de estos. La tasa de error mínima se obtiene

<sup>12</sup>un árbol con baja tasa de error es un clasificador fuerte

cuando se encuentra un óptimo entre la mínima correlación entre los árboles y la máxima fuerza de cada uno de estos.

Otro punto importante, es la posibilidad de utilizar este clasificador para seleccionar características. El mismo pondera las características según con que frecuencia son utilizadas para particionar los nodos, las de mayor frecuencia son las seleccionadas. Una estrategia para disminuir la cantidad de características sería, primero entrenar el árbol con todas las características y luego volver a entrenarlo utilizando solamente las características más importantes.

Se está trabajando con un problema de dos clases des-balanceadas. Esto significa que la proporción de datos de las vacas sanas es distinta a la de las vacas enfermas. En esta base la relación es 3 (sanas) a 1 (enferma). Por lo que este clasificador tratará de minimizar la tasa de error general manteniendo baja la tasa de error de la clase mayoritaria mientras que la tasa de error de la clase minoritaria será alta. Este des-balance puede ser contrarrestado utilizando diferentes pesos para las clases. Además se encontró que utilizando el criterio de parada de mínimos pesos por hoja se obtiene mejores resultados que sin utilizar criterio de parada. Por lo tanto, los parámetros a optimizar del algoritmo son:

- *m\_features*: cantidad de características por nodo
- *n\_estimators*: cantidad de árboles
- *class\_weight*: peso que se le asigna a cada clase
- *min\_weight\_fraction\_leaf*: fracción de peso mínimo del peso total requerido para ser un nodo hoja.

Para encontrar los parámetros óptimos (aquellos que maximizan la métrica seleccionada) se utiliza el algoritmo **Grid Serch**. Este consiste armar una grilla de N dimensiones (4 en este caso) y evaluar el desempeño del clasificador para cada punto de la misma. Se seleccionan los parámetros en los cuáles se obtienen el máximo  $F_1$ .

## Características

Se extraen de las series temporales las características definidas en [15]. Estas consistían en calcular la media entre las 20 máximas medidas de CE válidas ( $X_{20}$ ) y la variación de todas las medidas de CE válidas dentro de un ordeño ( $\sigma_{EC}^2$ ) para cada cuarto de la ubre (8 indicadores, 2 por cuarto). A partir de estos se definen las siguientes cuatro características:

- $Max\_X_{20}$ : máximo  $X_{20}$  del ordeño.
- $Max\_σ_{EC}^2$ : máximo  $σ_{EC}^2$  del ordeño.
- $IQR\_X_{20}$ : cociente entre el máximo y mínimo  $X_{20}$  del ordeño.
- $IQR\_σ_{EC}^2$ : cociente entre el máximo y mínimo  $σ_{EC}^2$  del ordeño.

## Capítulo 6. Hardware y Software de LecPro

Tabla 6.2: Conjunto Entrenamiento. VC en 10 subconjuntos

	<b>F1</b>	<b>Precisión</b>	<b>Recall</b>
<i>Max_X<sub>20</sub></i>	46.63± 2.84	43.78±3.42	50.0±2.98
<i>Max-σ<sub>EC</sub><sup>2</sup></i>	<b>64.19±2.11</b>	<b>65.07±2.67</b>	<b>63.46±3.10</b>
<i>IQR_X<sub>20</sub></i>	36.42±3.77	34.65±3.096	38.46±4.87
<i>IQR-σ<sub>EC</sub><sup>2</sup></i>	35.09±2.16	33.18±2.32	37.31±2.46
<i>Cuartos</i>	11.0±6.0	30.86±15	7.30±4.01
<i>Grasa</i>	44.14±2.44	44.87±2.67	3.46±2.46
<i>Proteina</i>	19.09+3.66	18.63±3.43	19.62±4.015
<i>Duracion</i>	54.93±4.11	54.93±4.85	55.0±3.86

Tabla 6.3: Conjunto Prueba

	<b>F1</b>	<b>Precisión</b>	<b>Recall</b>
<i>Max_X<sub>20</sub></i>	50.0	50.0	50.0
<i>Max-σ<sub>EC</sub><sup>2</sup></i>	57.14	50.0	66.67
<i>IQR_X<sub>20</sub></i>	14.29	12.5	16.66
<i>IQR-σ<sub>EC</sub><sup>2</sup></i>	30.769	28.57	33.33
<i>Cuartos</i>	0	0	0
<i>Grasa</i>	30.77	28.57	33.33
<i>Proteina</i>	46.15	42.86	50
<i>Duracion</i>	33.3	33.3	33.3

Se cuenta además con la cantidad de cuartos ordeñados, el % de grasa en 100 ml y el % de proteína en 100 ml de la mezcla de leche. También se utiliza la duración del ordeño de cada vaca medida en segundos.

## Resultados

En las Tablas 6.2 y 6.3 se observan los resultados al clasificar los conjuntos de prueba y entrenamiento utilizando una característica. En las Tablas 6.4 y 6.5 se aprecian los resultados al utilizar una combinación de características. Se utilizan todas juntas, luego las tres mejores según el clasificador y finalmente solo las cuatro sugeridas por el artículo. El mejor resultado se obtiene al clasificar con una única característica. Esta es *Max-σ<sub>EC</sub><sup>2</sup>*. Se hace notar que para seleccionar la mejor característica no se tiene en cuenta los resultados sobre el conjunto de prueba, sino los obtenidos sobre el conjunto de entrenamiento al utilizarse la técnica de 10 validaciones cruzadas (VC) en 10 subconjuntos. Esto es así porque esta técnica captura mejor la variabilidad de los datos que utilizar solo el conjunto de prueba.



## 6.4. Clasificación

Tabla 6.4: Conjunto Entrenamiento. VC en 10 subconjuntos. Combinación de características

	<b>F1</b>	<b>Precisión</b>	<b>Recall</b>
<b>Todas</b>	42.66±2.05	55.9±5	3.62±2.3
<b>best3RF</b>	45.66±4.76	50.03±4.92	37.31±4.57
<b>Artículo</b>	37.88±2.5	41.47±2.87	35.0±3.19

Tabla 6.5: Conjunto Prueba. Combinación de características

	<b>F1</b>	<b>Precisión</b>	<b>Recall</b>
<b>Todas</b>	80	100	66.67
<b>best3RF</b>	54.55	60	50
<b>Artículo</b>	36.36	40	33.3

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 7

## Prueba de Campo

### 7.1. Instalación del detector en el tambo

Se saca el organo que está puesto y se coloca el nuestro que incluye el traductor de conductividad y el sensor de temperatura conectado a la UM como se puede ver en la Figura 7.1



Figura 7.1: Instalación del organo

A la derecha se observa un zoom, se aprecia que se aferra el cable que va para el organo a los caños de vacío. En el otro extremo del cable está la caja de la UM como se puede ver en la Figura 7.2

Se instala el LecPro provisoriamente arriba de una caja que contiene la electrónica del dispositivo que saca los organos como se puede ver en la Figura 7.3. Se aferra



Figura 7.2: Instalación de la caja de la UM

el cable Ethernet a los caños y también el de alimentación del LecPro y la UM.

## 7.2. Problemas Surgidos

A continuación se listan los problemas surgidos, algunos fueron solucionados transitoriamente para poder concluir el proyecto, en particular poder obtener muestras para definir y entrenar el clasificador, y otros necesitarán cambios mayores.

### Problemas Eléctricos

- Soldadura en electrodos se afloja.  
En el comienzo se conectó los cables a los electrodos mediante soldadura



Figura 7.3: Instalación del LecPro, cable Ethernet y cable de alimentación

con estaño, pero como los electrodos son de acero inoxidable, la soldadura no quedaba bien y aparecen resistencias parásitas en serie. Se solucionó haciendo el contacto con tirafondos en los que se enrosca el cable, el tirafondo se enrosca en la pared que linda con el electrodo, de manera de que lo quede empujando para hacer contacto.

- Falso contacto en los pines.  
Se utilizaron pines de embutir, el problema es que con la vibración que hay en los caños de la sala, los pines generan falso contacto. Se solucionó agregando estaño de manera de que entren más apretados pero la mejor solución sería cambiar por pines que se aprieten con tornillos.
- Roturas de LM35.  
Se rompió múltiples veces, generalmente por mala aislación del agua, de todos modos algo a mejorar sería utilizar un transductor más robusto como por ejemplo una PT100.
- Rotura de LM35 rompe amplificador.  
Se cambia el LM35 y con la estación de soldadura se cambia el amplificador ya que es encapsulado tssop.

### Problemas Mecánicos

1. Puerto del organo se desconecta.  
Se comenzó utilizando un puerto Ethernet tipo jack pero se desconectaba por lo que se lo cambia por uno de micrófono con rosca.

## Capítulo 7. Prueba de Campo

2. Pezones con poca leche no llegan a ingresar a la celda.  
Se agregan extensiones en la parte inferior de los embudos para que no escape la leche por abajo cuando el caudal es bajo.
3. Tirafondos galvanizados se corroen.  
Para el contacto con los electrodos en un momento se utilizó tirafondos galvanizados ya que la medida que se necesitaba no había en acero inoxidable. El problema es que se oxidaron y dejaban de hacer contacto con los electrodos, por lo tanto se utilizó la medida que había en acero inoxidable y se realizaron las modificaciones necesarias en el diseño de la pieza inferior para poder utilizar esos tirafondos.
4. Químicos rompe la pieza portadora de transductores.  
Los químicos que se utilizan para el lavado, van debilitando la pieza hasta que se rompe, en comienzo duró solo dos ordeños luego se la reforzó un poco mas y duro 7 ordeños, con eso ya fue suficiente para armar la base de datos.

## Otros Problemas

1. Conexión errónea del filtro del LM35.  
El LM35 necesita un RC que se coloca dentro de la pieza portadora de transductores. Su función es filtrar. La resistencia va en serie con el condensador pero se los conectó erróneamente en paralelo y como las piezas van pegadas con la gotita no se pueden desarmar. Por lo tanto lo que se hizo fue realizar el filtrado por software.
2. Regulación del preset  $R_{10}$ .  
En la implementación del circuito se reguló el preset en  $417\Omega$  pero cuando se lo prueba ordeñando la entrada analógica de Arduino satura a 1023, por lo que se lo baja  $160\Omega$ . Esto fue debido a que los picos de conductividad son mayores a lo que se esperaba (12mS). Esta diferencia con lo consultado en la bibliografía puede ser por la genética de las vacas o la alimentación.

# Capítulo 8

## Conclusiones y trabajo a futuro

Desde un punto de vista académico se cumplió con el objetivo de enfrentar un problema desconocido (la detección de mastitis vacuna en tiempo real) que por medio de investigación y conocimientos adquiridos en la carrera, como modelado de fenómenos físicos, tratamiento de datos, desarrollo de software y electrónica, se logra crear un prototipo funcional.

A lo largo del proyecto se presentaron diversas dificultades que se superaron adquiriendo nuevos conocimientos como el diseño 3D de la pieza para imprimirla, soldar componentes de montaje superficial (por ejemplo TSSOP), desarrollo de firmware, software y realización de acondicionamiento mecánico. Además tuvimos la oportunidad de profundizar el uso de algunas herramientas que ya conocíamos como la biblioteca **scikitlearn** para la clasificación.

Se obtuvieron resultados de estimación del estado de la ubre de interés desde un punto de vista práctico ya que por medio de un sistema accesible económicamente para el tambo, es posible dar alertas en el tambo. Utilizando la característica  $Max.\sigma_{EC}^2$  se obtuvo una precisión de detección de mastitis de 65 % y un recall del 64 %. Estos resultados indican que podría utilizarse el sistema para realizar una preselección de ganado para luego ser tratado por el veterinario. En la Tabla 8.1 se muestra la comparación con los resultados de otros trabajos.

El costo estimado del producto en materiales es de 100 USD para el LecPro y 145 USD cada UM <sup>1</sup>. Si un tambo necesita 20 UM, el costo total es de 3000 USD

---

<sup>1</sup>Por detalles de como se llega a estos valores, consultar Apéndice C.1

Tabla 8.1: Comparing results with other works.

	<b>Recall</b>	<b>Precisión</b>	<b>Especificidad</b>
<b>Este trabajo</b>	64	65	83
Norberg et al. [15]	45	–	85
Lien et al. [20]	46	31	84
Cavero et al. [12]	70	–	84
Biggadike et al. [18]	50	48	87

## Capítulo 8. Conclusiones y trabajo a futuro

(Sin contar la mano de obra pero se supone que con lo que se ahorra fabricando en masa, se paga la mano de obra). Se agrega un margen de ganancia y cubrimiento de posibles gastos de 1000 USD, por lo tanto el costo del producto en el mercado sería de 4000 USD. Si un tambo utiliza como método de detección RCS una vez al mes para 400 vacas le cuesta 1600 USD al año. Por lo tanto a partir de dos años y medio ya es más barato nuestro detector. Además se tiene la ventaja de que se analizan las vacas en cada ordeño mientras que el otro es una vez al mes (para este establecimiento), por lo que se obtiene un beneficio mucho mayor que redundaría en mayores ingresos. Si bien debería hacerse un análisis más cuidadoso, resulta prometedor. Por otro lado, la empresa *GEA* ofrece un detector automático con un costo de 1000 USD por organo por lo tanto para 20 organos el costo es de 20.000 USD (cinco veces más caro que el nuestro). A diferencia, este detector cuenta con medición del volumen de leche. Pero en el nuestro se podría implementar esta función sin cambiar el HW, solo agregando código como se explica en el Apéndice C.2.

El diseño mecánico se podría mejorar a futuro realizando una pieza de un material resistente al lavado de la ordeñadora (posiblemente el mismo plástico de una de las partes del organo). Otra mejora es un método de construcción de electrodos y pieza más preciso para disminuir el error de calibración por celda media y teórica. También se debería cambiar el LM35 por un transductor más robusto, por ejemplo una PT100 (realizando las modificaciones necesarias en el circuito acondicionador). Con respecto al circuito, una mejora pendiente es cambiarle los pines de alimentación y señal por unos diseñados para ambientes vibratorios (posiblemente con tornillos para apretar). En cuanto a la alimentación, se podría alimentar a todas las UM con una misma fuente de 24V y en cada UM generar las demás tensiones a partir de este. Sobre el módulo Bluetooth es necesario cambiarlo por uno con más alcance debido a que la longitud de la sala es de alrededor de 20m mientras que el alcance del HC05 es entre 5 y 10 m. Otra mejora sería agregar un botón de encendido general al sistema.

Como trabajo a futuro en cuanto al software, se requiere realizar modificaciones para poder comunicarse con más UM. Esto tiene dos soluciones. La primera consiste en utilizar más Dongles USB. Dado que la Piconet<sup>2</sup> tiene una limitación de 7 dispositivos, se requiere un dongle cada 6 vacas. La otra solución sería cambiar el protocolo de comunicación entre LecPro-UM para que sea del estilo Maestro/Escavo. Es decir, que las UM envíen datos solamente cuándo el LecPro lo requiere. De esta manera, sería posible rotar secuencialmente las 6 UM que están conectadas por instante. Para el control de que UM conectar puede utilizarse en el thread (UMhandler()) que las atiende, semáforos para manejar los recursos compartidos (las 6 comunicaciones simultáneas que un dispositivo configurado como maestro puede realizar en una red Bluetooth).

Por otro lado, queda pendiente automatizar el preprocesamiento de las series temporales. Detectar cuándo termina un ordeño de vaca pero las celdas quedan con leche por que el organo queda inclinado (esto se puede implementar utilizando como método de detección de fin de ordeño que la conductividad sea constante

---

<sup>2</sup>Red bluetooth.



en un intervalo de tiempo). Y detectar cuándo una vaca se saca el organo de una patada (esto se puede implementar estimando el tiempo de *no ordeñe* normal ( $t_n$ ) y el tiempo de *no ordeñe* ocasionado por una patada ( $t_p$ ), entonces si se mide un tiempo de *no ordeñe* ( $t$ ) y  $t < (t_n + t_p)/2$  se descarta la serie de conductividad siguiente).

Como mejora de la clasificación, sería entrenar con datos de RCS por cuarto (para llevar a cabo esto, es necesario enviar al laboratorio una muestra de leche por cuarto para analizar). También si se utiliza el historial de las series de conductividad eléctrica, al tener series temporales diarias por animal es posible realizar un modelado del animal en lugar de la clase, entonces si el comportamiento de conductividad eléctrica del animal se aparta de los parámetros normales, podría ser un indicativo de la presencia de mastitis (para llevar esto a cabo sería necesario etiquetar -análisis de laboratorio- las muestras de cada vaca una vez a la semana por un periodo en el cual la misma haya estado sana y enferma). Otra característica que podría ser útil sería utilizar los máximos relativos de las series temporales en lugar de los máximos absolutos.

Si se utiliza el historial de las series temporales, una mejora sería automatizar la lectura de caravanas por ejemplo con un lector de caravanas RFID.

Esta página ha sido intencionalmente dejada en blanco.

# Glosario

- **CE:** Método de detección de mastitis por conductividad eléctrica.
- **Celda:** Recinto con un par de electrodos en caras opuestas, que se utiliza para medir la conductividad de soluciones.
- **CMT:** Método de detección de mastitis con el reactivo California.
- **Corrales:** Lugar físico donde se ubican las vacas mientras son ordeñadas.
- **Electrodos:** Placas metálicas.
- **Fosa:** Desnivel donde el ordeñador se ubica para facilitar la colocación organo a las vacas.
- **LecPro:** Unidad de lectura y procesamiento de datos.
- **Ordeñador:** Operario de máquinas de ordeño.
- **Ordeñar:** Extraer la leche de las ubres de las vacas de manera manual ó automática.
- **Organo:** Dispositivo que extrae la leche de la ubre de la vaca.
- **Raspberri Pi:** Computador de placa simple (SBC) de bajo coste.
- **RCS:** Método de detección de mastitis mediante el recuento de células somáticas.
- **Tambero:** Dueño del tambo.
- **UM:** Unidad que realiza la medición.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice A

Plan Original de Proyecto



UNIVERSIDAD  
DE LA REPUBLICA  
URUGUAY



PROYECTO DE FIN DE ESTUDIOS

EN LA CARRERA

INGENIERÍA ELÉCTRICA

# Detección Automática de Mastitis Vacuna DAMAVA

## Plan de Proyecto

Autores: Fabián Vique  
Henry Marichal  
Alejandro Goday

Tutor: Leonardo Steinfeld

Fecha: 2 de agosto de 2018

# Índice

<b>1. Resumen</b>	<b>2</b>
<b>2. Descripción del proyecto</b>	<b>3</b>
<b>3. Antecedentes:</b>	<b>3</b>
<b>4. Objetivo General</b>	<b>4</b>
<b>5. Alcance</b>	<b>4</b>
<b>6. Criterios de éxito</b>	<b>4</b>
6.1. Criterios de éxito . . . . .	4
6.2. Requerimientos Conceptuales . . . . .	4
6.2.1. Sistema . . . . .	4
6.2.2. Unidad de Medición (UM) . . . . .	4
6.2.3. Lectura de caravanas y Procesamiento de datos(LecPro) . . . . .	5
<b>7. Actores</b>	<b>5</b>
<b>8. Supuestos</b>	<b>6</b>
<b>9. Restricciones</b>	<b>6</b>
<b>10.Especificación Funcional del Proyecto</b>	<b>6</b>
<b>11.Objetivos específicos</b>	<b>6</b>
<b>12.Tareas</b>	<b>8</b>
<b>13.WBS</b>	<b>11</b>
<b>14.Cronograma detallado del Proyecto</b>	<b>12</b>
14.1. Hitos . . . . .	12
<b>15.Análisis de Costos</b>	<b>15</b>
<b>16.Análisis de Riesgos</b>	<b>16</b>
<b>17.Anexo</b>	<b>18</b>
17.1. Funcionamiento de la Sala de ordeño . . . . .	18

# 1. Resumen

## Integrantes:

- Fabián Vique
  - C.I: 5390100-5
  - Correo electrónico: fab.vique@gmail.com
- Henry Marichal
  - C.I: 4862364-6
  - Correo electrónico: hmarichal@fing.edu.uy
- Alejandro Goday
  - C.I: 4681315-6
  - Correo electrónico: alejandrogoday@gmail.com

## Clientes:

- Tamberos

## Tutor:

- Leonardo Steinfeld

## Fecha prevista de Finalización:

30 de abril del 2018

## Total de horas a realizar por el grupo de proyecto:

2910 hs.

## Entregables intermedios:

- Hito 1:
  - Fecha: 15/09/17
  - Se entregara:
    - **Documentación de detección de mastitis**
      1. Investigación de métodos de detección.
      2. Selección de método de detección.
    - **Diseños**
      1. Diseño de funcionamiento de la UM.
      2. Diseño de HW de bloque de medición.
- Hito 2:
  - Fecha: 15/02/18
  - Se entrega:
    - **Documentación de la UM**
      1. Documentación de Firmware.
      2. Documentación de testeo de bloques.
      3. Documentacion de diseño de protección mecánica e hídrica.
      4. Documentación de testeo de protección mecánica e hídrica.
    - **Diseños**
      1. Diseño del LecPro.



## 2. Descripción del proyecto

La rentabilidad de un tambo está directamente determinada por la cantidad y calidad de la leche que produce. Esto es tan importante que las compañías recolectoras de leche penalizan a los tambos que no logren los niveles promedios permitidos de células somáticas. La mastitis bovina es el factor que provoca el mayor aumento de los recuentos de células, además de disminuir la cantidad de leche que produce el animal. Esta enfermedad consiste en la inflamación de la glándula mamaria, provocándose cambios en la composición bioquímica de la leche y en el tejido de la glándula.

Existen dos tipos de mastitis:

- **Mastitis subclínica:** Se caracteriza por la presencia de un microorganismo en combinación con un conteo elevado de células somáticas en leche. Es la más difícil de detectar debido a que no se observan alteraciones ni en la leche, ni en la ubre de la vaca. Además es de larga duración y más frecuente que los otros tipos.
- **Mastitis clínica:** Se observa inflamación del cuarterón afectado y el animal siente dolor al tocarlo. La leche se observa alterada con la presencia de descamaciones, coágulos, suero descolorido, y a veces sangre.

Dado que la mastitis clínica presenta síntomas visuales claros, el problema está en detectar la presencia de la mastitis subclínica. A partir de esto se han diseñado diversos métodos, algunos mas precisos que otros, para su detección. A continuación se mencionan los métodos [6] más relevantes en Uruguay y una breve descripción de cada uno.

- **Prueba de Wisconsin:** Consiste en enviar muestras a analizar en laboratorios para realizar el conteo de las células somáticas. Para esto, es necesario que el tambero tome las muestras de leche a cada vaca de manera manual.
- **Prueba California (CMT):** Es el método más utilizado para detectar este tipo de mastitis en Uruguay. Esta prueba se realiza en el establecimiento tomando manualmente una muestra de leche por vaca. Se requiere de un veterinario para realizarla dado que los resultados obtenidos son difíciles de interpretar, por lo que se necesita de alguien con experiencia y formación para hacerlo.
- **Conductividad:** Actualmente existen productos manuales y automáticos que detectan la mastitis midiendo la conductividad de la leche. Estos detectan un 50 % o más de falsos positivos. El producto automático viene incluido en un paquete que encarece excesivamente su precio, mientras que el manual es de bajo costo pero carece de practicidad para analizar un número grande de vacas.

## 3. Antecedentes:

A continuación en el cuadro 1 se definen los términos utilizados para la clasificación de resultados:

Cuadro 1: Clasificación de resultados

TP	Casos de mastitis subclínica correctamente clasificados
FP	Casos sanos clasificados erróneamente.
FN	Casos de mastitis subclínica clasificados erróneamente.
TN	Casos sanos correctamente clasificados.

Se definen los siguientes estimadores:

$$\text{Precisión} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN} \quad \text{Accuracy} = \frac{TP + TN}{TN + TP + FP + FN} \quad \text{Error} = \frac{FP}{FP + TP}$$

La mastitis implica un aumento de las concentraciones de sodio ( $\text{Na}^+$ ) y cloro ( $\text{Cl}^-$ ) en la leche, lo cual lleva a un aumento en su conductividad eléctrica. Estos cambios tienden a ocurrir antes de que se desarrollen signos visibles, por lo cual podrían ayudar a detectar esta infección.

En [20] se propone detectar mastitis subclínica utilizando la serie temporal de EC de la leche, obtenida por cuarto durante el ordeño. La serie temporal la procesan utilizando discriminantes lineales y realizando un análisis de regresión paso a paso. Obtienen un Recall de 46.2% y una Precisión de 30.7%. Concluyen que los resultados son buenos, debido al bajo costo de la implementación comparado con otros sistemas que utilizan el historial de EC.

Por otro lado, en [18] se propone utilizar el historial de EC por cuarto de los últimos 14 ordeños. Se establece una alerta de un posible caso de mastitis, si se supera cierto umbral determinado por la media del historial de EC. Los resultados que obtienen son sensibles al umbral utilizado. Concluyen que el método no es lo suficientemente preciso para ser utilizado como el único criterio para detectar mastitis.

Para el Proyecto de Fin de Carrera(PFC), se considera seguir una metodología similar a la propuesta en [20], debido a la dificultad de crear una base de datos con el historial de EC.

## 4. Objetivo General

Crear un dispositivo que detecte mastitis en cada vaca de manera automática. Esto permitirá al tambero tomar acción inmediata para controlar la mastitis, afectando de la menor forma posible su ingreso.

## 5. Alcance

- Realización de un dispositivo que detecte mastitis por vaca de forma automática, a colocarse en el órgano de la ordeñadora.
- En lo que a software refiere, en lo posible, se utilizarán librerías ya implementadas.
- El dispositivo va a estar ubicado en la sala de ordeño, por lo tanto debe ser impermeable, compacto, y resistente mecánicamente.
- Se realizará cableado para la alimentación de la red eléctrica.
- Se fabricará un prototipo para un solo órgano que se probará en la sala de ordeño, aunque el diseño del sistema contemplará todos los órganos de la sala. Además la lectura de caravanas se simulará.
- Se realizará la comunicación con el servidor y se elaborará una página web.

## 6. Criterios de éxito

### 6.1. Criterios de éxito

- Asociar correctamente los datos relevados en cada puesto con la vaca correspondiente.
- Funcionamiento continuo del prototipo por un período de dos semanas.
- Costo de fabricación de prototipo menor a USD 500.

### 6.2. Requerimientos Conceptuales

#### 6.2.1. Sistema

El sistema<sup>1</sup> estará compuesto por una Unidad de Medición(UM) en cada órgano y por una Unidad de Lectura de caravanas y Procesamiento de datos(LecPro), la cual envía los datos a un servidor para hacerlos disponibles en una página web.

#### 6.2.2. Unidad de Medición (UM)

##### Ubicación

- Debe estar ubicado en los órganos de la máquina de ordeño.

##### Requerimientos mecánicos

- Debe soportar golpes originados por el movimiento del ordeñador en la fosa (ocasionados al llevarselo por delante, o durante la colocación del órgano a la vaca).
- Debe ser impermeable.

##### Requerimientos de Interfaces

---

<sup>1</sup>Véase la sección 10

- Interfaz inalámbrica con el LecPro.

#### **Requerimientos de seguridad**

- No requiere seguridad de datos.

#### **Requerimientos Regulatorios**

- No contaminar la leche.
- La comunicación con el LecPro sería en una banda ISM de 2.4GHz.

#### **Requerimientos funcionales**

- El dispositivo detecta el comienzo de ordeño de la vaca.
- El dispositivo muestrea la conductividad de la leche durante todo el ordeño
- Enviar datos al LecPro para el procesamiento.

### **6.2.3. Lectura de caravanas y Procesamiento de datos(LecPro)**

#### **Ubicación**

- En la entrada de la sala de ordeño a una altura de entre 1.3 y 1.8 metros.

#### **Requerimientos de Protección**

- Impermeable.
- Resistencia a golpes ocasionados por un cabezazo de las vacas.

#### **Requerimientos de interfaces**

- Comunicación inalámbrica con la UM.
- Comunicación con servidor.

#### **Requerimientos regulatorios**

- La comunicación con el UM sería en una banda ISM de 2.4GHz.

#### **Requerimientos funcionales**

- Identificar vaca y guardar el orden de ingreso a la sala de ordeño.
- Recibir medidas realizadas por la UM.
- Realizar procesamiento de datos.
- Enviar resultado al servidor.

## **7. Actores**

Los actores son:

- Docentes de Proyecto
- Tutor
- Equipo de proyecto
- Tamberos
- Veterinarios (Para consultas).
- Facultad de Veterinaria UdelaR (Para consultas y análisis de muestras).

## 8. Supuestos

El proyecto se desarrolla bajo los siguientes supuestos:

- Se tendrá autorización para trabajar con un veterinario para realizar las pruebas de detección de mastitis.
- Se contará con acceso a tambo y permiso a hacer cambios.

## 9. Restricciones

- Se disponen de USD 800 para el proyecto
- El plazo límite para finalizar es el 30/04/18.

## 10. Especificación Funcional del Proyecto

El Sistema estará compuesto por una UM en cada órgano, la cuál realizará las medidas en la leche. Esta, enviará los resultados de las mediciones al LecPro, el cuál se encargará de asociarlas con la vaca correspondiente. El LecPro realizará un procesamiento sobre los datos para estimar si se esta ante un caso de mastitis. El resultado se enviará a un servidor, el cuál los desplegará a través de una pagina web.

## 11. Objetivos específicos

Se plantean los siguientes objetivos específicos para el proyecto<sup>2</sup>

1. Definición de métodos de detección de mastitis.

a) Investigación métodos existentes en el mercado y en la literatura científica.

- Resumen de los métodos más factibles de implementación en el marco del proyecto de fin de carrera.

2. Diseño y armado del bloque UM.

a) Diseño conceptual de la UM

- Cumplimiento de requerimientos finales

b) Implementación del software de la arquitectura de microprocesador

- No haber encontrado errores en los testeos.

c) Implementación de comunicación

- No haber encontrado errores en los testeos.

d) Implementación del bloque de medición.

- No haber encontrado errores en los testeos.

e) Implementación de alimentación

- No haber detectado anomalías en el funcionamiento

f) Implementación de protección mecánica e hídrica

- Resistencia al funcionamiento cotidiano del tambo.

g) Integración de bloques

- No haber encontrado errores en el testeo conjunto.

h) Documentación

- Haber registrado todas las etapas de la implementación de la UM.

---

<sup>2</sup>En el primer nivel de viñetas se indican los objetivos específicos. En el segundo nivel, los entregables de dichos objetivos. Y en el tercer nivel, los requisitos de aprobación.

### 3. Diseño y armado del bloque de LecPro.

- a) Diseño conceptual del LecPro
  - Cumplimiento de requerimientos finales
- b) Implementación de unidad de Procesamiento.
  - No haber encontrado errores en los testeos.
- c) Implementación de comunicación
  - No haber encontrado errores en los testeos.
- d) Implementación de Procesamiento de datos
  - No haber encontrado errores en los testeos
- e) Implementación de alimentación
  - No se detecten anomalías en el funcionamiento
- f) Implementación de protecciones
  - Resistencia al funcionamiento cotidiano del tambo.
- g) Integración de bloques
  - No encontrar errores en el testeo conjunto.
- h) Documentación
  - Haber registrado todas las etapas de la implementación del LecPro.

### 4. Integración.

- a) Diseño de Integración UM-LecPro
  - No encontrar errores en el testeo conjunto.

### 5. Servidor web, el cual se divide en los siguientes entregables:

- a) Implementación de conexión a internet.
  - No encontrar errores en el testeo.
- b) Página web.
  - No encontrar errores en el testeo.

### 6. Prueba de campo

- a) Documentación
  - Haber registrado los resultados de la prueba de campo

### 7. Documentación.

- a) Informe final
  - No haber encontrado errores en las últimas 3 revisiones completas

### **Se tiene el siguiente objetivo específico secundario:**

#### 1. Identificación automática de caravanas:

- a) Implementación de la lectura de caravanas
  - Identificar más del 90 % de las caravanas. El 10 % restante no se leería debido a que el sensor de lectura de caravana no puede realizar la medida.

## 12. Tareas

El proyecto está constituido por las siguientes tareas<sup>3</sup>:

1. Definición de métodos de detección de mastitis.
  - a) Investigar y preseleccionar métodos:  
Puesta al día en el estado del arte y preselección de métodos de detección.  
Duración: 50hrs. Recursos: 2
  - b) Documentación.  
Se documentarán todas las tareas relevantes a este objetivo.  
Duración: 30 hrs. Recursos: 2
2. Diseño y armado del bloque del UM.
  - a) Diseño conceptual UM  
Diseño de interconexión entre bloques y funcionamiento conjunto. Selección componentes para cada bloque.  
Duración: 9 hrs. Recursos: 3.
  - b) Desarrollo de SW microprocesador  
Duración: 125 hrs. Recursos: 2.
  - c) Testeo SW de microprocesador  
Duración: 50 hrs. Recursos: 2.
  - d) Desarrollo de SW de comunicación  
Programación de SW de comunicación UM-LecPro.  
Duración: 50 hrs. Recursos: 2.
  - e) Testeo de SW de comunicación  
Duración: 20 hrs. Recursos: 2
  - f) Diseño de HW de bloque de medición  
Diseño del bloque encargado de realizar las medidas pertinentes sobre la muestra de leche.  
Duración: 150 hrs. Recursos: 3
  - g) Desarrollo de SW de bloque de medición  
Duración: 120hrs. Recursos: 3
  - h) Implementación de bloque de medición  
Duración: 100 hrs. Recursos: 2
  - i) Testeo bloque medición  
Duración: 30 hrs. Recursos: 2
  - j) Diseño e implementación de alimentación.  
Duración: 60hrs. Recursos: 2
  - k) Testeo de Alimentación  
Duración: 10hrs. Recursos: 2
  - l) Integración de bloques y testeo  
Duración: 150 hrs. Recursos: 3
  - m) Diseño protección mecánica e hídrica  
Duración: 100 hrs. Recursos: 2
  - n) Adquirir componentes protecciones  
Duración: 10 hrs. Recursos: 1
  - ñ) Implementación protecciones mecánicas e hídricas  
Duración: 25 hrs. Recursos: 1
  - o) Testeo protecciones  
Duración: 10 hrs. Recursos: 1

---

<sup>3</sup>La duración de las tareas se mide en horas hombre

- p)* Adquisición de componentes electrónicos  
Compra de componentes electrónicos.  
Duración: 25 hrs. Recursos: 2
- q)* Prueba de campo UM  
Se instalara el dispositivo en el tambo. Se probará el funcionamiento durante 4 ordeños, guardando las medidas relevadas en una memoria. Se tomaran muestras manuales por vaca y se enviarán a analizar a un laboratorio.  
Duración: 17. Recursos: 1.
- r)* Documentación  
Se documentarán todas las tareas relevantes a este objetivo específico.  
Duración: 150 hrs. Recursos: 3

### 3. Diseño y armado del bloque de LecPro.

- a)* Diseño LecPro  
Diseño de interconexión entre bloques y funcionamiento conjunto. Selección de componentes para cada bloque.  
Duración: 9hrs. Recursos: 3
- b)* Desarrollo de SW de comunicación  
Duración: 75 hrs. Recursos: 3
- c)* Testeo bloque comunicación  
Duración: 15hrs. Recursos: 3
- d)* Implementación SW procesamiento de datos  
Se estudiarán e implementarán técnicas de procesamiento de datos para mejorar la estimación de la detección.  
Duración: 100 hrs. Recursos: 1
- e)* Desarrollo de SW de unidad de procesamiento  
Duración: 50hrs. Recursos: 2
- f)* Testeo SW unidad de procesamiento.  
Duración: 10 hrs. Recursos: 2
- g)* Integración y testeo conjunto bloques.  
Duración: 75 hrs. Recursos: 3
- h)* Diseño protección mecánica e hídrica.  
Duración: 30 hrs. Recursos: 2
- i)* Adquirir componentes protecciones.  
Realizar compras de protecciones en mercado local.  
Duración: 10 hrs. Recursos: 1
- j)* Implementación protecciones.  
Duración: 20 hrs. Recursos: 1
- k)* Testeo protección.  
Duración: 10 hrs. Recursos: 1
- l)* Documentación.  
Se documentarán todas las tareas relevantes a este objetivo específico.  
Duración: 90 hrs. Recursos: 3.

### 4. Servidor web

- a)* Implementación de conexión a internet.  
Duración: 100 hrs. Recursos: 2
- b)* Pagina web  
Duración: 100 hrs. Recursos: 2

### 5. Integración.

- a)* Diseño de integración UM-LecPro.  
Duración: 50 hrs. Recursos: 2

- b) Implementación  
Verificar el funcionamiento UM-LecPro  
Duración: 80 hrs. Recursos: 2
- c) Testear.  
Realizar test de cobertura.  
Duración: 75 hrs. Recursos: 3
- d) Documentación  
Se documentarán todas las tareas relevantes a este objetivo específico.  
Duración: 60 hrs. Recursos: 3.

6. Prueba de campo UM-LecPro

- a) Instalación de dispositivo en el tambo  
Ida a campo para instalar el dispositivo en el tambo.  
Duración: 30 hrs. Recursos: 2
- b) Prueba de campo  
Se probará el dispositivo durante 4 ordeñes para obtener una base de datos. Además se tomarán muestras de leche de las vacas de prueba. Estas muestras se enviarán a laboratorio, para realizar el recuento de células somáticas.  
Duración: 15 hrs. Recursos: 1
- c) Documentación  
Se documentarán todas las tareas relevantes a este objetivo específico.  
Duración: 60 hrs. Recursos: 3.

7. Documentación.

- a) Introducción  
Duración: 10 hrs. Recursos: 2
- b) Integrar documentos  
Duración: 75 hrs. Recursos: 3
- c) Conclusiones  
Duración: 45 hrs. Recursos: 3
- d) bibliografía  
Duración: 10 hrs. Recursos: 2

El total de horas es de 2395 hrs. Se planean dedicar 2910 hrs al proyecto. Por lo que, en caso de que no haya imprevistos en la realización de las tareas, se dedicará las horas restantes a los objetivos secundarios.

Las precedencias entre tareas se aprecian en la figura 2. En la figura 1 se observa la WBS.



# 13. WBS

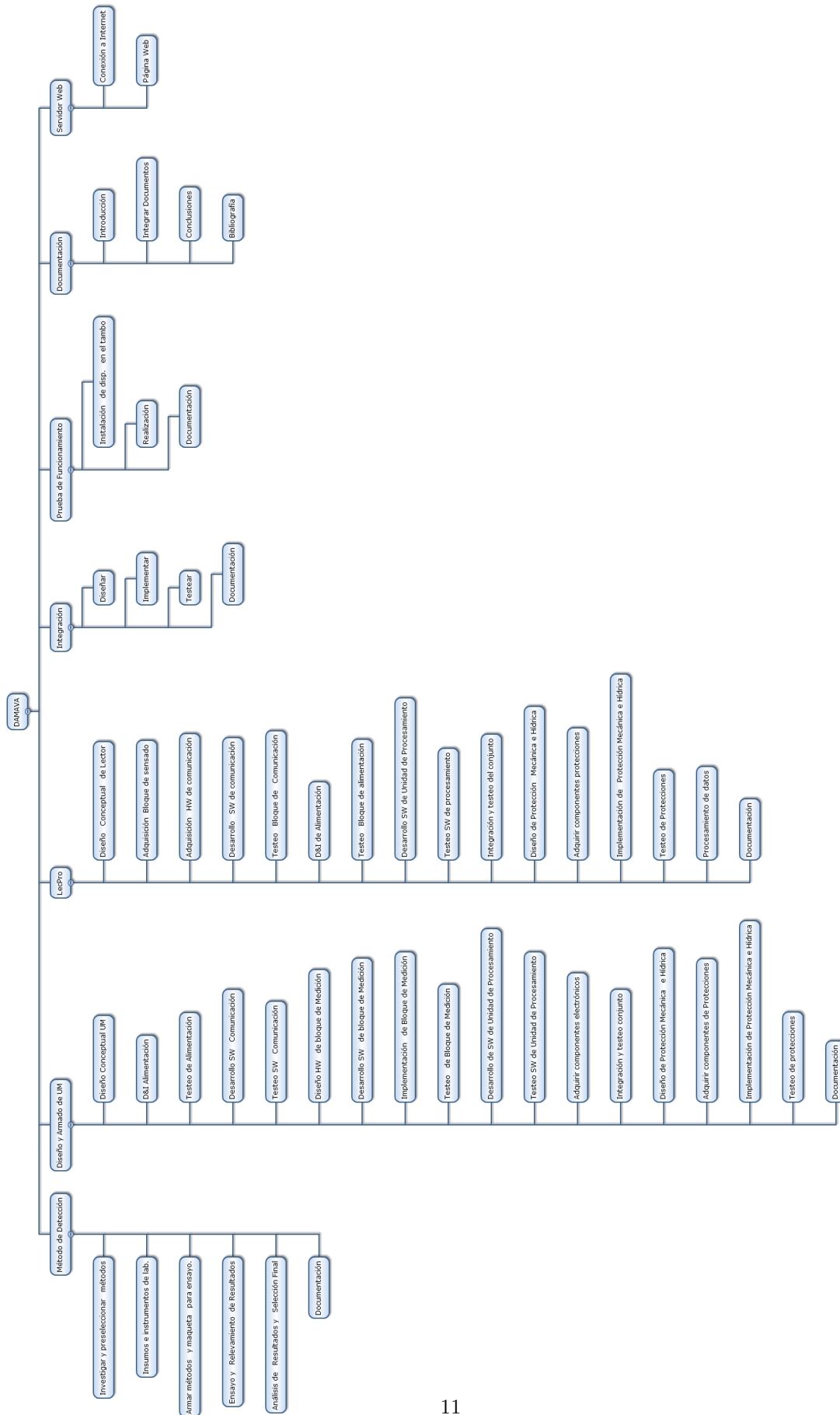


Figura 1: WBS

## 14. Cronograma detallado del Proyecto

### 14.1. Hitos

El proyecto cuenta con dos instancias de evaluación, el Hito 1 será el 15/09/17 y el Hito 2 será el 15/02/18. En estas instancias se deberá presentar los siguientes entregables.

#### Hito 1

##### Entregables:

- **Documentación de detección de mastitis**

1. Investigación de métodos de detección.
2. Selección de método de detección.

- **Diseños**

1. Diseño de funcionamiento de la UM.
2. Diseño de HW de bloque de medición.

#### Hito 2

##### Entregables:

- **Documentación de la UM**

1. Documentación de Firmware.
2. Documentación de testeo de bloques.
3. Documentación de diseño de protección mecánica e hídrica.
4. Documentación de testeo de protección mecánica e hídrica.

- **Diseños**

1. Diseño del LecPro.

En las figuras 2 y 3 se observa el diagrama de Gantt. Las fechas en naranja son los hitos y en rojo es el plazo máximo para terminar el proyecto. El periodo púrpura, es un periodo de exámenes.

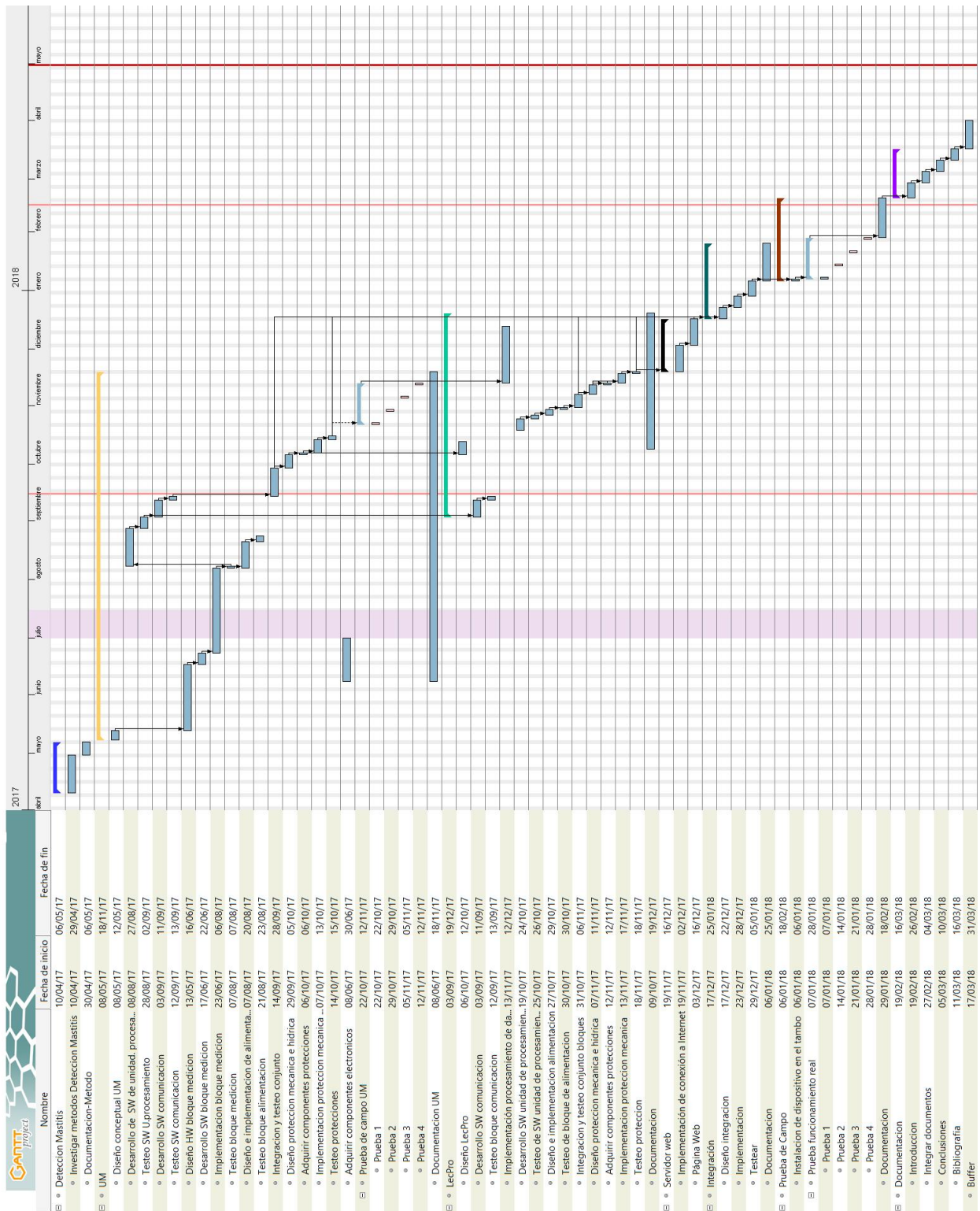


Figura 2: Diagrama de Gantt: Precedencia de tareas

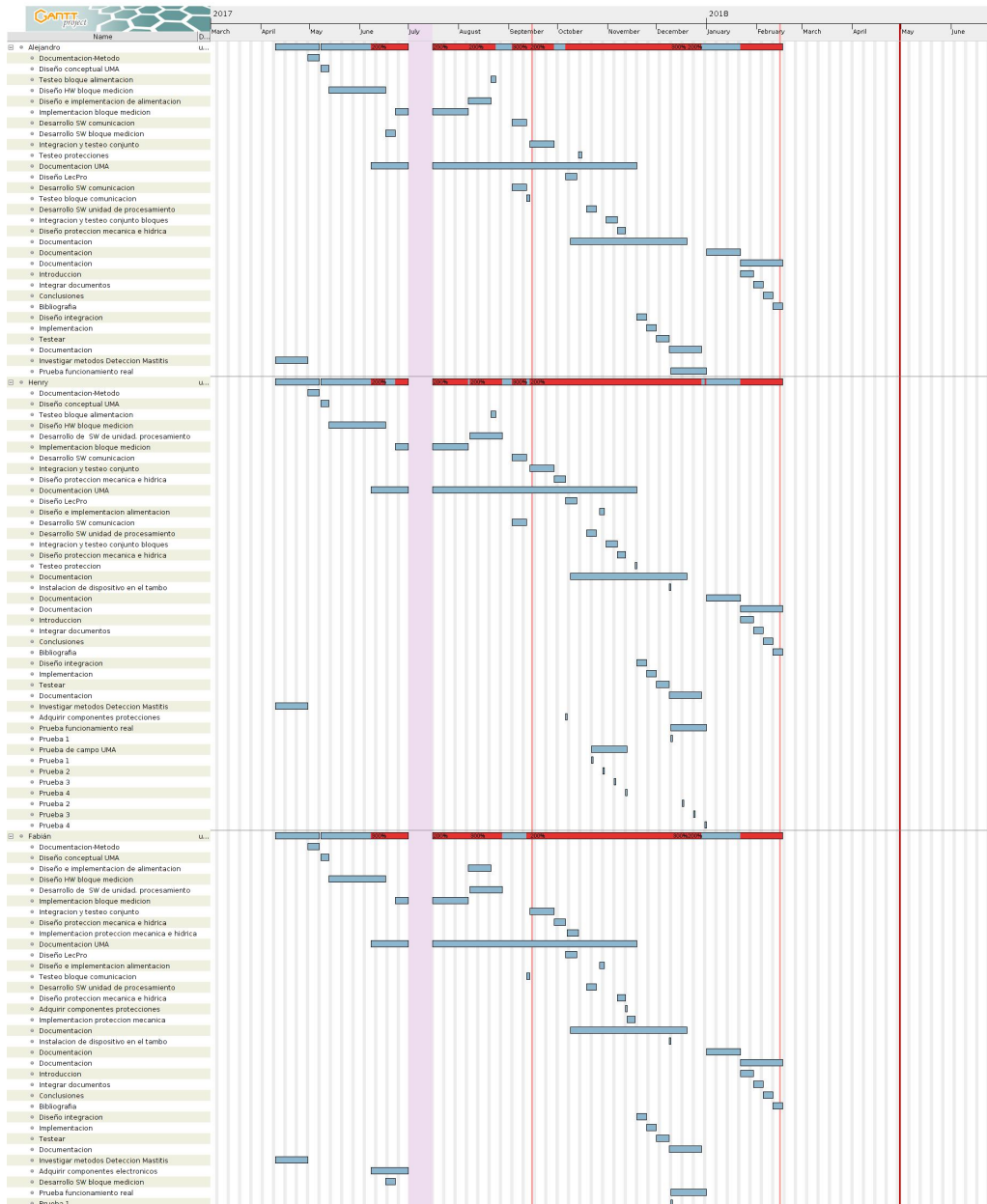


Figura 3: Diagrama de Gantt: Asignación de recursos

## 15. Análisis de Costos

A continuación se presenta el flujo de caja en el cuadro 2 , y el presupuesto en el cuadro 3.

Cuadro 2: Flujo de Caja

<b>Flujo de Caja</b>				
Mes	Concepto	Egreso (US\$)	Ingreso (US\$)	Flujo Efectivo (US\$)
Abril 2017	-Instrumentos de laboratorio	29	0	1059
	-120 Horas hombre	1030		
Mayo 2017	-Kit Comunicación	50	0	1239
	-Kit Microcontrolador	25		
	-Raspberry Pi	90		
	-120 Horas hombre	1030		
Junio 2017	-120 Horas hombre	1030	0	1030
Julio 2017	-120 Horas hombre	1030	0	1030
Agosto 2017	-Protección mecánica de UM	10	0	2084
	-Protección hídrica de UM	6		
	-Gabinete	10		
	-180 Horas hombre	2058		
Setiembre 2017	-Componentes electrónicos	50	0	2108
	-180 Horas hombre	2058		
Octubre 2017	-Protección mecánica LecPro	10	0	2118
	-Protección hídrica LecPro	6		
	-Viajes	34		
	-Pruebas laboratorio	0		
	-Gabinete	10		
	-180 Horas hombre	2058		
Noviembre 2017	-Materiales para Integración	7	0	2072
	-Cargador	7		
	-180 Horas hombre	2058		
Diciembre 2017	-Pruebas laboratorio	0	0	2072
	-180 Horas hombre	2058		
Enero 2018	-180 Horas hombre	2058	0	2058
Febrero 2018	-180 Horas hombre	2058	0	2058
Marzo 2018	-180 Horas hombre	2058	0	2058

Cuadro 3: Presupuesto del proyecto DAMAVA

<b>Presupuesto para proyecto DAMAVA</b>				
Concepto	Cantidad	P/U (US\$)	Total (US\$)	Observaciones
Kit microcontrolador	1	25	25	Arduinos y launchpad
Kit comunicación	2	25	50	
Componentes electrónicos	-	-	50	
Viajes (ida y vuelta)	2	17	34	Cada viaje cuesta (ida y vuelta) 500 pesos
Multímetro	1	9	9	
Cargador	1	7	7	
Raspberry Pi 3	1	90	90	70 (Amazon), 90 (MercadoLibre)
Gabinete (protección de electrónica)	2	10	20	
Protección hídrica	2	6	12	
Protección mecánica	2	10	20	
Materiales para integración	1	7	7	Cables, tornillos, etc.
Prueba SCC (Laboratorio)	80	-	488	
Imprevistos	-	-	24	(5 % de la suma total, que fue US\$ 464)
<b>Total</b>	-	-	488	

## 16. Análisis de Riesgos

Los riesgos mas relevantes son:

1. Retraso en la recepción de componentes electrónicos(demora aduana, courier).
2. Alcance del sensado de caravanas insuficiente.
3. Imposibilidad de instalar el dispositivo en el tambo.
4. Costo del prototipo superior a 800 USD.
5. Abandono de un integrante.

En el cuadro 4 y en la figura 4 se presentan respectivamente la estimación y la evaluación inicial de los riesgos.

Cuadro 4: Estimación Inicial de Riesgos

Riesgos	Prob. Ocurrencia	Nivel Impacto
1	Moderado	Alto
2	Muy probable	Alto
3	Poco probable	Extremo
4	Poco probable	Alto
5	Poco probable	Alto

		Probabilidad de ocurrencia		
		Poco probable	Moderado	Muy probable
Nivel de Impacto	Ninguno			
	Bajo			
	Medio			
	Alto	R4,R5	R1	R2
	Extremo	R3		

Figura 4: Evaluación de Riesgos

Se propone el siguiente plan de respuestas ante cada riesgo

1. Adelantar fecha de adquisición de los componentes.
2. Utilizar amplificador. En caso que aún no tenga el alcance requerido, cambiar la ubicación física del lector para que requiera un alcance menor.
3. Contactar a Facultad de Veterinaria para usar sus instalaciones. Como última opción, en caso de no conseguirse ningún establecimiento en donde instalar el dispositivo, se realizará un experimento de laboratorio simulando las condiciones reales.
4. Pedir préstamos a nuestras familias. En caso de no ser suficiente, limitar prestaciones del producto.
5. Redistribución de tareas entre los dos integrantes restantes. Revisar alcance.

En el cuadro 5 y en la figura 5 se observan respectivamente la estimación y la evaluación final de los riesgos.

Cuadro 5: Estimación final de riesgos

Riesgos	Prob. Ocurrencia	Nivel Impacto
1	Poco probable	Alto
2	Poco probable	Alto
3	Poco probable	Extremo
4	Poco probable	Alto
5	Poco probable	Alto

		Probabilidad de ocurrencia		
		Poco probable	Moderado	Muy probable
Nivel de Impacto	Ninguno			
	Bajo			
	Medio			
	Alto	R4,R5,R1,R2		
	Extremo	R3		

Figura 5: Evaluación final de Riesgos

## 17. Anexo

### 17.1. Funcionamiento de la Sala de ordeño

En la figura 6 se observa un esquema de la sala de ordeño del tambo. Las vacas ingresan a dicha sala por las entradas laterales (6). Cada vaca tiene acceso a un comedero (1), en el cual tiene ración para su alimentación. En muchos establecimientos, es el ordeñador (8) quien se encarga de ordeñar y racionar los comederos.

Durante la mayor parte del tiempo de ordeño, el ordeñador se encuentra en la fosa (5). Ésta se ubica en el centro de la sala de ordeño, a un nivel de aproximadamente un metro por debajo de los corrales donde se encuentran las vacas. Esto se puede observar en la figura 7, y es así para facilitarle al ordeñador la colocación de las pezoneras en la ubre de la vaca. Por el centro de la fosa, a una altura lo suficientemente elevada para no molestar al ordeñador, se ubica el caño central (2), por el cual transita el flujo de leche de todos los órganos (17) hacia el tanque de frío (4). En dicho tanque se almacena la leche hasta que el camión cisterna la retira.

La sala puede contener por turno el doble de vacas que de órganos. En la figura 7 se aprecia el caño para el agua (14), el caño por donde circula la leche (16) hacia el flujo central (2), el *recibidor de leche* (3), el caño de presión (15) que contiene aire comprimido para el sistema de bombeo de leche y la luminaria (13).

El funcionamiento del ordeño es el siguiente: Las vacas ingresan por ambas puertas laterales a la sala (ver figura 6). Primero se colocan los órganos sobre las vacas de un lateral (supongamos el de más abajo en la figura). A medida que van terminándose de ordeñar las vacas, se van colocando los órganos en las vacas del lateral de más arriba en la figura. Cuando todas las vacas del lateral de abajo terminan de ordeñarse, se abre la portera de salida (7). Las vacas salen de la sala. Se cierra la portera de salida, y se abre la de entrada (6) para que ingresen otras vacas. Se cierra la portera de entrada. Y nuevamente, a medida que las vacas del lateral de arriba van terminando de ordeñarse, se van colocando los órganos en las vacas del lateral de abajo. Este proceso se repite, hasta que se ordeñan todas las vacas.

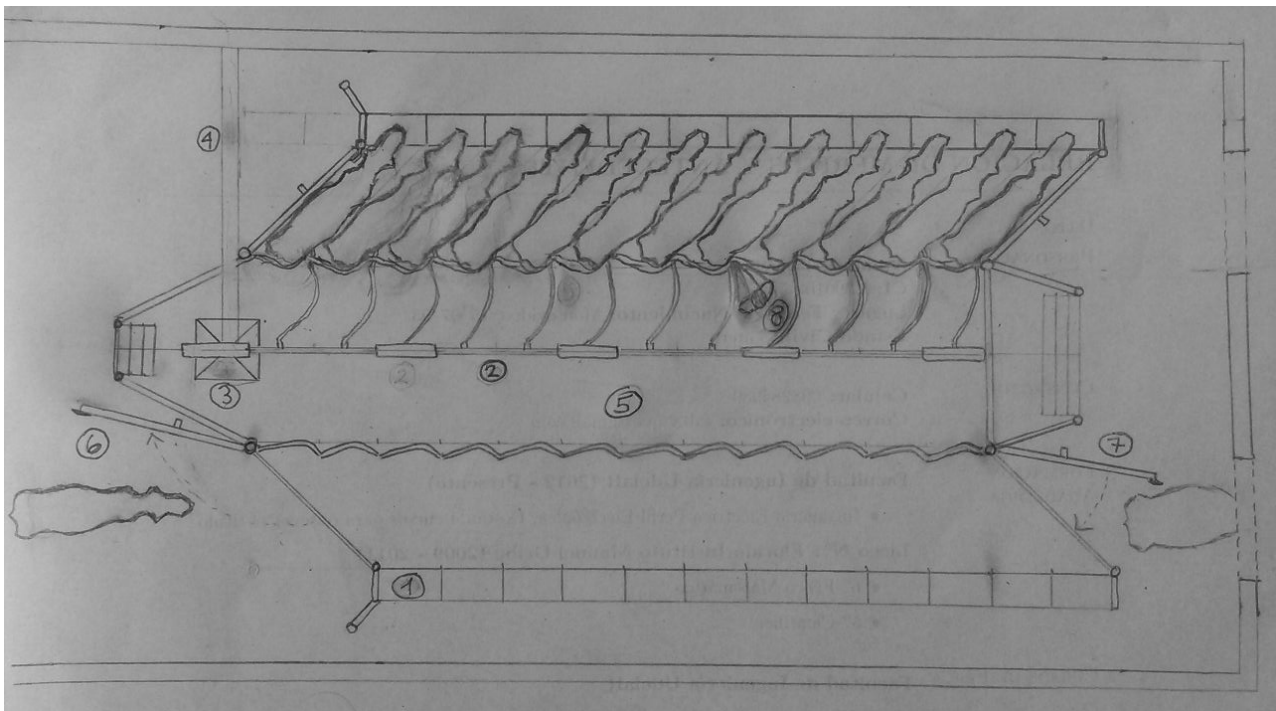


Figura 6: Vista horizontal sala de ordeño



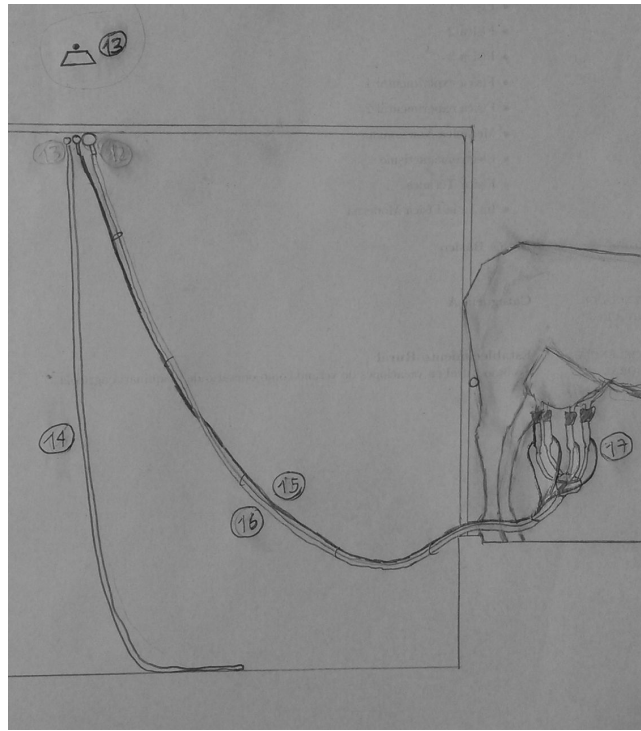


Figura 7: Vista vertical sala de ordeње

Esta página ha sido intencionalmente dejada en blanco.

# Apéndice B

## Transductor de Conductividad

En este Apéndice se detalla pruebas experimentales y descripciones de funcionamiento de bloques del transductor de conductividad.

### B.1. Prueba de embudo

A continuación se detalla el experimento para verificar la solución propuesta de colocar un embudo y encontrar el diámetro óptimo de salida del embudo.

**Materiales:**

- Pieza metálica del organo
- Una celda cuadrada de 1cm de lado
- Una manguera
- Una taza de 250ml
- Una botella de plástico
- Cinta adhesiva
- Tijera

**Procedimiento**

Se construye un embudo recortando de la botella, un cuadrado de aproximadamente 5cm x 5cm, se lo enrolla en forma de cono tratando de dejarle el agujero de salida cerrado, se lo pega con cinta para que no se desenrolle y se le empareja la entrada con una tijera.

Se conecta la manguera a una canilla con agua corriente, se emboca la punta en una taza y se regula la apertura de la canilla de manera que llene la taza (250ml) en 13s, de esta manera se obtiene un caudal de 19ml/s que es el caudal medio de los cuartos.

## Apéndice B. Transductor de Conductividad

Se embute la manguera en uno de los ductos de la pieza metálica y se sostiene la celda en la posición que iría ( vertical y cerca del eje de simetría de la pieza ). Primero se prueba sin el embudo y se observa que el fluido está muy turbulento dentro de la celda como se esperaba. Luego con la tijera se le realiza un corte en la punta del cono de manera que se forme un agujero muy pequeño, ahora se lo coloca antes de la celda y se observa que el fluido dentro de la celda es estacionario. Si se llena demasiado lento ( mayor a 10s ), se realiza otro corte y se observa el fluido, cuando se agrandó demasiado el agujero, se observa fluido turbulento en la celda.

Se encontró que el agujero óptimo es de diámetro 1mm.

### Tiempo de llenado

El tiempo de llenado de la celda se puede calcular teóricamente. El diámetro del ducto es de  $d_d = 9,3mm$  por lo que si la salida del embudo se deja en  $d_e = 1mm$  se estaría tomando una porción del caudal. Un calculo grueso del caudal  $Q'$  a la salida del embudo es:

$$Q' \approx Q \times \frac{\pi(d_e/2)^2}{\pi(d_d/2)^2} = Q \times \frac{d_e^2}{d_d^2} = 0,22ml/s$$

Que llenaría una celda en 4,6s que es menor a la que se deseaba ( 10s ) .

## B.2. Circuito

En esta sección se detalla el funcionamiento de algunos bloques del circuito del transductor de conductividad.

### B.2.1. Oscilador

En la Figura B.1 se muestra un diagrama del integrado LM555, que conectando resistencias y condensadores como se muestra, se logra un ciclo de trabajo 50%. Esta configuración se encuentra en el sitio web [?].

Como se observa en la figura B.1 hay un divisor resistivo y dos comparadores, uno compara el voltaje en la pata 6 con  $2V_{cc}/3$  y el otro el voltaje en la pata 2 con  $V_{cc}/3$ . Estos comparadores son los que hacen conmutar el flip-flop entre 0 y  $V_{cc}$ .

La resistencia R5 se elige suficientemente grande para que la corriente por la misma sea despreciable comparada con la que circula por R4.

Si  $V_{out} = V_{cc}$ , C5 se carga a través de R4 (por que la corriente por R5 es despreciable) hasta que el voltaje en la pata 6 llega a  $2V_{cc}/3$ . En ese momento  $V_{out}$  conmuta a cero y C5 se descarga a través de R4 hasta que el voltaje en la pata 2 llega a  $V_{cc}/3$  y  $V_{out}$  conmuta nuevamente a  $V_{cc}$ .

Como se carga y descarga a través de la misma resistencia, el tiempo de carga y descarga es el mismo, por lo que calculando el tiempo que demora en descargarse

## B.2. Circuito

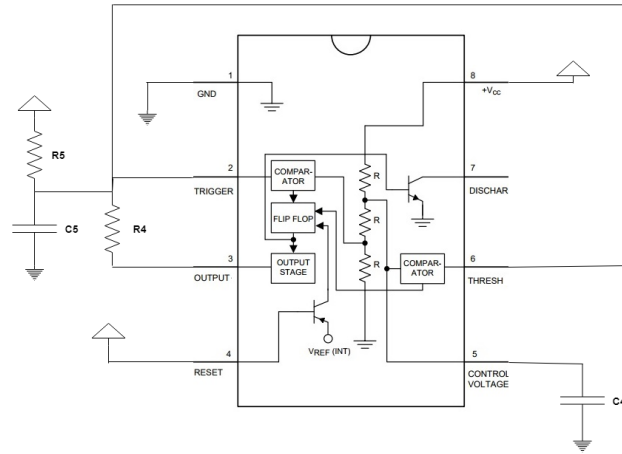


Figura B.1: Diagrama de conexionado del LM555. Imagen extraída de [2], con modificaciones

de  $2V_{cc}/3$  a  $V_{cc}/3$  se obtiene la mitad del período ( $T/2$ ).

$$e^{-\frac{t_1}{R_4 C_5}} = \frac{2}{3}$$

$$e^{-\frac{t_2}{R_4 C_5}} = \frac{1}{3}$$

$$\frac{T}{2} = t_2 - t_1 = R_4 C_5 (\ln(3) - \ln(\frac{3}{2})) = R_4 C_5 \ln(2)$$

$$T = 2R_4 C_5 \ln(2)$$

Por lo tanto la frecuencia es:

$$f = \frac{1}{2R_4 C_5 \ln(2)} \text{ Hz}$$

### B.2.2. Filtro pasa banda MFB

Aplicando nodos en  $V_a$

$$\frac{V_{in} - V_a}{R_3} = \frac{V_a}{R_2} - \frac{V_o}{R_1} + (V_a - V_o)Cs$$

Ademas

$$V_a Cs = \frac{-V_o}{R_1}$$

Operando con estas ecuaciones se llega a la transferencia:

$$\frac{V_o}{V_{in}} = \frac{\frac{-s}{R_3 C}}{s^2 + \frac{2s}{R_1 C} + \frac{R_3 + R_2}{R_3 R_2 R_1 C^2}}$$

## Apéndice B. Transductor de Conductividad

Por lo tanto se obtienen los siguientes parámetros de una transferencia de segundo orden (frecuencia angular natural y factor de amortiguamiento)

$$\omega_n = \frac{1}{C} \sqrt{\frac{1}{R_1} \left( \frac{1}{R_2} + \frac{1}{R_3} \right)}$$

$$\zeta = \sqrt{\frac{R_3 R_2}{(R_3 + R_2) R_1}}$$

La frecuencia central del filtro pasa banda es:

$$f_c = \frac{\omega_n}{2\pi}$$

La ganancia a esta frecuencia es:

$$G = \frac{-R_1}{2R_3}$$

El ancho de banda del filtro es la diferencia entre las frecuencias a la que la ganancia cae 3db con respecto a la ganancia a la frecuencia central. Planteando esta definición se llega a que las frecuencias y el ancho de banda es:

$$f_2 = \frac{\omega_n}{2\pi} (\zeta + \sqrt{\zeta^2 + 1})$$

$$f_1 = \frac{\omega_n}{2\pi} (-\zeta + \sqrt{\zeta^2 + 1})$$

$$B = f_2 - f_1 = \frac{\omega_n \zeta}{\pi}$$

El factor de calidad es la frecuencia central dividida por el ancho de banda

$$Q = \frac{f_c}{B} = \frac{1}{2\zeta} = \frac{1}{2} \sqrt{R_1 \left( \frac{1}{R_2} + \frac{1}{R_3} \right)} \quad (\text{B.1})$$

### B.2.3. Detector de Picos

Para los diodos se utiliza el modelo  $V_\gamma$ . Este modelo se resume en el cuadro B.1. Ambos diodos son idénticos.

Tabla B.1: Modelo  $\gamma$  diodo

Estado	Hipótesis	Verificación
ON	$v_D = V_\gamma$	$i_D > 0$
OFF	$i_D = 0$	$v_D \leq V_\gamma$

La tensión de entrada es de la forma:

$$v_{in}(t) = A \sin(\omega t) \quad \forall t \geq 0$$

Todas las resistencias  $R_j$  son iguales y su valor es  $R$

Se analiza el circuito en etapas. Se estudia el transitorio con  $D_1$  OFF y  $D_2$  ON. En régimen, el circuito funciona en dos etapas, intercalándose entre  $D_1$  ON y  $D_2$  OFF, y  $D_1$  OFF y  $D_2$  ON.

Desde  $t = 0$  hasta  $t = t_1$

En esta etapa, se supone  $D_1$  OFF y  $D_2$  ON. Estas suposiciones se traducen en lo expuesto en la Tabla B.2.

Tabla B.2: Hipótesis y condiciones a verificar en primera etapa.

Desde $t = 0$ hasta $t = t_1$		
Diodo	Hipótesis	Condición a Verificar
$D_1$	$i_{D1} = 0$	$v_{D1} \leq V_\gamma$
$D_2$	$v_{D2} = V_\gamma$	$i_{D2} < 0$

Dado que por hipótesis  $i_{D1} = 0$ , se tiene un divisor de tensión entre  $R_{12}$  y  $R_{13}$  de forma tal que:

$$v^- = \frac{v_{out}}{2}$$

Además, a causa del divisor de tensión formado por  $R_9$  y  $R_{11}$ , sabemos que:

$$v^+ = \frac{v_{in}}{2}$$

Por tanto, considerando que hay un cortocircuito virtual ( $v^- = v^+$ ), se tiene:

$$v_{out}(t) = v_{in}(t) \quad \forall t \leq t_1 \quad (\text{B.2})$$

Donde  $t_1$  es el instante en que deja de cumplirse la condición más restrictiva de las expuestas en el cuadro B.2.

Condición para  $D_1$ :

Planteando las caídas de tensión en  $D_1$ ,  $D_2$  y  $C_3$ , tenemos:

$$v_{D1} = v^- - (v_{D2} + v_{out})$$

Sabemos que  $v_{out}(t) = v_{in}(t)$  y además por hipótesis  $v_{D2} = V_\gamma$ . Sustituyendo en  $v_{D1}$  obtenemos:

$$v_{D1} = -\left(\frac{v_{in}}{2} + V_\gamma\right)$$

Imponiendo  $v_{D1} \leq V_\gamma$  se obtiene la condición:

$$v_{in}(t) \geq -4V_\gamma \quad (\text{B.3})$$

Suponiendo que  $t_1 < \frac{\pi}{w}$  se tiene que  $v_{in} \geq 0$ . Por lo tanto, esta condición se verifica.

## Apéndice B. Transductor de Conductividad

Condición para  $D_2$ :

Por ley de nodos se tiene que la corriente  $i_{D_2}$  es la suma de la corriente por el condensador y la corriente por  $R_4$ . Además se impone  $i_{D_2} > 0$ . Entonces

$$i_{D_2} = C_3 \frac{dv_{in}(t)}{dt} + \frac{v_{in}(t)}{2R} > 0$$

Por lo que se debe verificar

$$\frac{dv_{in}(t)}{dt} > \frac{-v_{in}(t)}{2RC_3} \Rightarrow Aw \cos(wt) > \frac{-A \sin(wt)}{2RC_3}$$

$$2RC_3 w > -\tan(wt) \quad (\text{B.4})$$

La condición B.4 se verifica mientras  $t < \frac{\pi}{2w}$ . Por lo tanto

$$t_1 = \frac{\pi}{2w} \quad (\text{B.5})$$

Desde  $t = t_1$  hasta  $t = \frac{2\pi}{w}$

En esta etapa, se supone  $D_1$  ON y  $D_2$  OFF. Estas suposiciones se traducen en lo expuesto en la Tabla B.3.

Tabla B.3: Hipótesis y condiciones a verificar en segunda etapa.

Desde $t = t_1$ hasta $t = \frac{2\pi}{w}$		
Diodo	Hipótesis	Condición a Verificar
$D_2$	$i_{D_2} = 0$	$v_{D_2} \leq V_\gamma$
$D_1$	$v_{D_1} = V_\gamma$	$i_{D_1} \geq 0$

Al igual que en la parte anterior, se considera un cortocircuito virtual ( $v^- = v^+$ ). Entonces por  $R_1$  circula una corriente:

$$i_3(t) = \frac{v_{in}(t)}{2R}$$

Al condensador ingresa una corriente de

$$i_c(t) = \left( \frac{v_{in}(t)}{2} - v_c(t) \right) \frac{1}{R} = C_3 \frac{dv_c(t)}{dt}$$

Entonces la corriente que circula por  $D_1$  es

$$i_{D_1} = -(i_c + i_3)$$

Se utiliza la transformada de Laplace para resolver la ecuación diferencial lineal de primer orden:

$$\frac{dv_c(t)}{dt} + \frac{v_c(t)}{RC_3} = \frac{v_{in}(t)}{2RC_3} \quad (\text{B.6})$$



Se utilizan las siguientes transformadas:

$$L[v_{in}(t)] = A \frac{w}{s^2 + w^2}$$

$$L \left[ \frac{dv_c(t)}{dt} \right] = sV_c(s) - v_c(t_1)$$

Obteniéndose:

$$V_c(s) = \frac{A}{2} \frac{w}{(s^2 + w^2)(1 + \tau s)} + \frac{\tau v_c(t_1)}{1 + \tau s} \quad \text{con} \quad \tau = RC_3 \quad v_c(t_1) = A$$

Utilizando:

$$L^{-1} \left[ \frac{w}{(s^2 + w^2)(1 + \tau s)} \right] = \frac{\tau w e^{-\frac{t}{\tau}}}{1 + (\tau w)^2} + \frac{\text{sen}(wt - \text{argtan}(\tau w))}{\sqrt{1 + (\tau w)^2}}$$

$$L^{-1} \left[ \frac{1}{1 + \tau s} \right] = \frac{e^{-\frac{t}{\tau}}}{\tau}$$

Al antitransformar se obtiene:

$$v_c(t) = A \left( \frac{1}{2} \left[ \frac{\tau w e^{-\frac{t}{\tau}}}{1 + (\tau w)^2} + \frac{\text{sen}(wt - \text{argtan}(\tau w))}{\sqrt{1 + (\tau w)^2}} \right] + e^{-\frac{t}{\tau}} \right)$$

$$i_{D1} = A \left( \frac{Cw}{2} \left[ \frac{-e^{-\frac{t}{\tau}}}{1 + (\tau w)^2} + \frac{\cos(wt - \text{arctan}(\tau w))}{\sqrt{1 + (\tau w)^2}} \right] - \frac{e^{-\frac{t}{\tau}}}{\tau} + \frac{\text{sen}(wt)}{2R} \right)$$

Condición para  $D_1$ :

Se tiene que verificar que  $i_{D1}$  es positiva para  $t > t_1$

En la figura B.2 se observa el gráfico de la corriente  $i_{D1}$  para  $t > t_1$ . Esta condición se cumple hasta  $t_2 < \frac{5T}{4}$  con  $T = \frac{2\pi}{w} = 0,1ms$ . El valor de  $t_2$  se obtiene numéricamente. Este es:  $t_2 = 0,1221ms$ .

Condición para  $D_2$ :

Se tiene que verificar que la tensión en bornes del diodo sea menor a  $V_\gamma$ . Se tiene

$$v_{D2} = v_{amp} - v_c = \frac{v_{in}(t)}{2} - V_\gamma - v_c < V_\gamma$$

$$\frac{v_{in}}{2} < v_c + 2V_\gamma$$

Dado que la frecuencia de el oscilador es 11,8kHz entonces  $T = 84,7\mu S$ , se elije  $\tau = RC = 2,78ms$  de modo que la descarga del condensador sea menor al 3%. Por lo que se verifica la condición ya que  $v_c(t = t_1) = A$ .

Entre  $t_2$  y  $\frac{5T}{4}$  se vuelve al caso 1.

## Apéndice B. Transductor de Conductividad

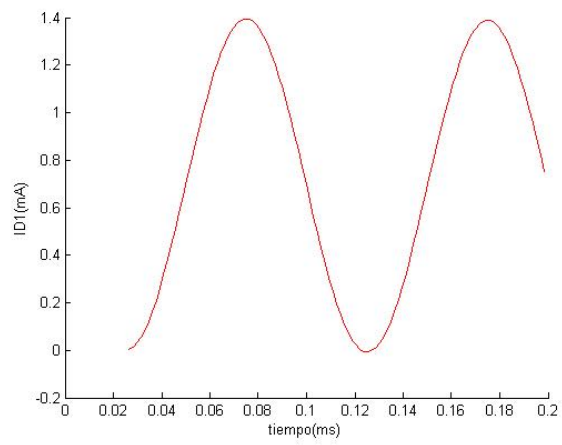


Figura B.2: Corriente por D1

# Apéndice C

## Conclusiones

En este Apéndice se presenta en detalle resultados utilizados en las conclusiones.

### C.1. Costos materiales de UM y LecPro

En las tablas C.1 y C.2 se presenta respectivamente los costos de materiales del LecPro y la UM.

Tabla C.1: Costo de lecPro

Componente	Costo (USD)
Raspberry pi	77
Caja	13
Dongle	10
<b>Total</b>	<b>100</b>

### C.2. Medición de volumen de leche

En esta sección se propone un método de medición de volumen de leche que produce una vaca, utilizando el hardware actual.

De las series temporales de conductividad, se puede calcular la pendiente de subida ( $m_i$ ) de cada cuarto. Esta es proporcional al caudal ( $Q_i$ ) y a la conductividad de cuando se llena la celda ( $\sigma_{i,full}$ ) entonces  $Q_i = k \times m_i / \sigma_{i,full}$  donde  $k$  es una constante a determinar que depende del volumen de la celda y de el diámetro de salida del embudo. Multiplicando por el tiempo de duración de cada serie ( $t_i$ ) se obtiene el volumen  $V_i = k \times t_i \times m_i / \sigma_{i,full}$ . El volumen total producido por una vaca es la suma del de los 4 cuartos entonces  $V = k \sum_{i=1}^4 t_i \times m_i / \sigma_{i,full}$ . En los datos de las vacas proporcionados Santiago de Izaguirre se encuentra el de leche que produce cada vaca por lo tanto se puede estimar el parámetro  $k$ .

## Apéndice C. Conclusiones

Tabla C.2: Costo de la UM

<b>Componente</b>	<b>Costo (USD)</b>
LM35	7
Caja	17
Puerto	10
HC-05	10
Fuente	17
Arduino	17
Fabricación de PCB	33
Cable	7
Componentes electrónicos	17
Otros	10
<b>Total</b>	<b>145</b>

# Referencias

- [1] OPAx192 36-V, Precision, Rail-to-Tail Input/Output, Low Offset Voltage, Low Input Bias Current Op Amp with e-trim. <http://www.ti.com/lit/ds/symlink/opa4192.pdf>, December 2013.
- [2] LM555 Timer. <http://www.ti.com/lit/ds/symlink/lm555.pdf>, February 2015.
- [3] ADG1604. <http://www.analog.com/media/en/technical-documentation/data-sheets/ADG1604.pdf>, 2016.
- [4] L7808 STMicroelectronics Linear Voltage Regulators. <https://www.mouser.com/datasheet/2/389/178-974043.pdf>, November 2016.
- [5] LM35 Precision Centigrade Temperature Sensors. <http://www.ti.com/lit/ds/symlink/lm35.pdf>, December 2017.
- [6] C.C. Bedolla, V.H. Castañeda, and W. Wolder. Métodos de detección de la mastitis bovina. *Revista electrónica de Veterinaria*, VII(9), 2007.
- [7] Omar Fernando Fernández Bolaños, José Eduardo Trujillo Graffe, John Jai-ver Peña Cabrera, Jefferson Cerquera Gallego, and Yury Tatiana Granja Salcedo. Mastitis bobina: generalidades y métodos de diagnóstico. *Revista Veterinaria REDVET*, (13), 2012.
- [8] L. Breiman. Random forests. *Machine Learning*, pages 5–32, 2001.
- [9] Omelio Cepero, Caridad Camacho, Julio César Castillo, and José Salado. Conductividad eléctrica y californina mastitis test en la detección de la mastitis subclínicas. *REDVET*, VI(3), 2005.
- [10] Ana Beatriz da Costa Ribeiro, Joice Sifuentes dos Santos, and Daniel Zanol. Evaluation of an electrical conductivity portable device as an alternative for subclinical mastitis detection. *Salud Animal*, 38(2), 2016.
- [11] M.V. Daniel G. Ferraro. Concepto de calidad de leche. su importancia para la calidad del producto final y para la salud del consumidor. *Aprocal*.
- [12] D.Cavero, K.H.Tolle, G.Rave, C.Buxade, and J.Krieter. Analysing serial data for mastitis detection by means of local regression. *Livestock Science*, (107):101–110, 2007.

## Referencias

- [13] D.S. Whyte, P.T. Johnstone, and R.W. Claycomb and G.A. Mein. Online sensors for earlier, more reliable mastitis detection. 2004.
- [14] E.F. Elizalde, M.L. Signorini, V.R. Canavesio, A. Cuatrin, H.D. Tarabla, and L.F. Calvino. Medición de la conductividad eléctrica en leche como método de diagnóstico de mastitis subclínica bovina. 2009.
- [15] E. Norberg, H. Hogeveen, I.R. Korsgaard, N.C. Friggens, K.H.M.N. Sloth, and P. Lovendahl. Electrical conductivity of milk: Ability to predict mastitis status. *American Dairy Science Association*, (87):1099–1107, 2004.
- [16] F.J. Ferrero, M. Valledor, and J.C. Campo. Screening method for early detection of mastitis in cows. *The Veterinary record*, 2002.
- [17] William M. Haynes. Crc handbook of chemistry and physics, 91th edition.
- [18] H.H. Biggadike, I. Ohnstad, R.A. Laven, and J.E. Hillerton. Evaluation of measurements of conductivity of quarter milk samples for the early diagnosis of mastitis. *The Veterinary record*, 2002.
- [19] Eloy Guillermo De la Cruz González and Rocío Contero. Correlación de los métodos california mastitis test (cmt), conductividad eléctrica (ce) y conteo de células somáticas (ccs) en el laboratorio de calidad de leche de la ups. *Ingeniería Agropecuaria CAYAMBE - Tesis de Pregrado*, 2011.
- [20] Cheng-Chang Lien, Ye-Nu Wan, and Ching-Hua Ting. Online detection of dairy cow subclinical mastitis using electrical conductivity indices of milk. *Engineering in agriculture, environment and food*, (9):201–207, 2015.
- [21] M.F. Mabrook and M.C. Petty. Effect of composition on the electrical conductance of milk. *Journal of food engineering*, (60):321–352, 2003.
- [22] Diego Ramírez Muñoz and Silvia Casan Berga. An analog electronic interface to measure electrical conductivity in liquids. *Measurement*, (38):181–187, 2005.
- [23] Annette D. Shine. Fitting experimental data to straight lines. 2006.

# Índice de tablas

3.1. Comparación de LM35 con termocupla . . . . .	16
3.2. Componentes seleccionados . . . . .	16
3.3. Alimentación . . . . .	19
3.4. Colores de conductores . . . . .	23
4.1. Dimensiones de las celdas . . . . .	30
4.2. Relación entre las componentes eléctricas equivalentes entre celdas. A es el área de la celda 1 y l el largo. Se asume que al cambiar el área de la cara <b>a.h</b> la separación entre la interfaz electrodo-leche se mantiene constante. . . . .	32
4.3. Parámetros ajustados para cada celda . . . . .	32
4.4. Error relativo entre la resistencia Rm determinada por el analizador y el módulo de la impedancia a 10 kHz . . . . .	32
4.5. Rm a distintas temperaturas . . . . .	34
4.6. Cálculo de alfa . . . . .	34
4.7. Componentes seleccionados . . . . .	41
4.8. Componentes seleccionados . . . . .	44
4.9. Componentes seleccionados . . . . .	46
4.10. Valores de las componentes para el bloque . . . . .	47
4.11. Comportamiento detector de picos . . . . .	48
4.12. Comportamiento del detector de picos . . . . .	51
4.13. Soluciones de NaCl . . . . .	58
4.14. Valores de ADC promedio al variar la conductividad en cada celda	60
4.15. Parámetros de ajuste . . . . .	61
4.16. Parámetros de ajuste de celda media . . . . .	62
4.17. E.C1: Error celda 1, E.Cm: Error celda media, E.Ct Error celda teórica . . . . .	63
4.18. Errores de calibraciones . . . . .	63
6.1. Términos utilizados para la clasificación de resultados . . . . .	87
6.2. Conjunto Entrenamiento. VC en 10 subconjuntos . . . . .	92
6.3. Conjunto Prueba . . . . .	92
6.4. Conjunto Entrenamiento. VC en 10 subconjuntos. Combinación de características . . . . .	93
6.5. Conjunto Prueba. Combinación de características . . . . .	93

## Índice de tablas

8.1. Comparing results with other works. . . . .	99
B.1. Modelo $\gamma$ diodo . . . . .	130
B.2. Hipótesis y condiciones a verificar en primera etapa. . . . .	131
B.3. Hipótesis y condiciones a verificar en segunda etapa. . . . .	132
C.1. Costo de lecPro . . . . .	135
C.2. Costo de la UM . . . . .	136



# Índice de figuras

1.1. Dispositivo para determinación de la conductividad eléctrica de la leche. Imagen extraída de [6]	3
2.1. Sala de ordeño	9
2.2. Fosa. Se observa un organo en funcionamiento. En el tambo en el cual se prueba el sistema se tienen 23.	11
2.3. Diagrama de bloques del sistema completo	11
2.4. Diagrama del tambo con el sistema implementado	12
3.1. Diagrama del sistema completo en el tambo	14
3.2. Transductor LM35. Imagen extraída de <a href="https://www.internetdelascosas.cl/wp-content/uploads/2012/05/arduino-LM35-sensor-pines-278x300.png">https://www.internetdelascosas.cl/wp-content/uploads/2012/05/arduino-LM35-sensor-pines-278x300.png</a>	15
3.3. Amplificador de LM35	16
3.4. RC. Imagen extraída de hoja de datos LM35 [5]	17
3.5. Conexión del HC-05	18
3.6. Conexión L7808 extraído de [4].	19
3.7. Diseño PCB	20
3.8. Vista superior del PCB armado	21
3.9. Vista inferior del PCB armado	22
3.10. Diagrama de conexión del puerto macho.	22
3.11. Caja de registro.	23
3.12. Arduino.	24
3.13. Circuito de medición.	24
3.14. Fuente.	25
3.15. Puerto y llave.	25
3.16. Cable y soportes.	26
4.1. Modelo eléctrico equivalente de la celda	28
4.2. Celdas de ensayo	29
4.3. Módulo y fase de la impedancia de la celda 1 con una solución de KCL de concentración 0.01 mol/kg. En negro se tiene el modelo eléctrico equivalente ajustado en el rango 1-100kHz	30
4.4. Módulo y fase de la impedancia de la celda 2 con una solución de KCL de concentración 0.01 mol/kg. En negro se tiene el modelo eléctrico equivalente ajustado en el rango 1-100kHz	31

## Índice de figuras

4.5. Módulo y fase de la impedancia de la celda 3 con una solución de KCL de concentración 0.01 mol/kg. En negro se tiene el modelo eléctrico equivalente ajustado en el rango 1-100kHz . . . . .	31
4.6. Módulo de impedancias del modelo eléctrico equivalente de la celda 1 con solución. . . . .	33
4.7. organo . . . . .	35
4.8. Piezas del organo . . . . .	35
4.9. Interior pieza metálica organo. . . . .	36
4.10. Vista superior de pieza inferior . . . . .	37
4.11. Vista inferior de pieza inferior . . . . .	38
4.12. vista superior de pieza superior . . . . .	39
4.13. Vista inferior de pieza superior . . . . .	40
4.14. Diagrama de bloques del circuito . . . . .	41
4.15. Esquemático del circuito . . . . .	42
4.16. Diagrama de conexionado de electrodos . . . . .	42
4.17. Filtro pasa banda MFB . . . . .	43
4.18. Filtro pasa banda MFB . . . . .	44
4.19. Filtro pasa banda MFB . . . . .	45
4.20. Pines del multiplexor. Imagen extraída de la hoja de datos del ADG1604 [3]. . . . .	46
4.21. Diagrama del detector de picos . . . . .	47
4.22. Salida del integrado LM555 . . . . .	50
4.23. Salida del filtro pasa banda . . . . .	50
4.24. Puerto . . . . .	52
4.25. Puerto hembra . . . . .	53
4.26. Puerto macho . . . . .	53
4.27. Vista superior de pieza inferior impresa . . . . .	54
4.28. Vista inferior de pieza inferior impresa . . . . .	54
4.29. Vista superior de pieza superior impresa . . . . .	55
4.30. Vista inferior de pieza superior impresa . . . . .	55
4.31. Electrodo puestos . . . . .	56
4.32. Pieza superior con cables y LM35 . . . . .	56
4.33. Piezas instaladas . . . . .	57
4.34. UM completa . . . . .	58
4.35. Cambios de solución celda 3 . . . . .	59
4.36. Ajuste de las celdas . . . . .	60
4.37. Ajuste de celda media . . . . .	61
5.1. Diagrama de módulos. Las flechas salientes indican que se utilizan funciones de otros módulos. La biblioteca Timer utiliza registros definidos en la biblioteca Arduino. En el main se utiliza la clase SoftwareSerial definida en la biblioteca SoftwareSerial. . . . .	66
6.1. . . . .	73
6.2. Diagrama de módulos del LecPro. Las flechas salientes indican que se utilizan funciones de otros módulos . . . . .	74

6.3. Trama del paquete enviado por UM . . . . .	82
6.4. Árbol de decisión para un nuevo dato . . . . .	83
6.5. Aplicación móvil Bluetooth Terminal . . . . .	86
6.6. Series temporales de CE. La mezcla tiene un RCS de 3.399.000 células/mililitro. . . . .	88
6.7. Árbol de decisión. Ejemplo artificial. . . . .	90
7.1. Instalación del organo . . . . .	95
7.2. Instalación de la caja de la UM . . . . .	96
7.3. Instalación del LecPro, cable Ethernet y cable de alimentación . .	97
B.1. Diagrama de conexionado del LM555. Imagen extraída de [2], con modificaciones . . . . .	129
B.2. Corriente por D1 . . . . .	134



Esta es la última página.  
Compilado el jueves 2 agosto, 2018.  
<http://iie.fing.edu.uy/>