

UNIVERSIDAD NACIONAL DE HUANCABELICA

(CREADA POR LEY N° 25265)



FACULTAD DE INGENIERÍA ELECTRÓNICA - SISTEMAS
ESCUELA ACADÉMICO PROFESIONAL DE ELECTRÓNICA

TESIS

**“SISTEMA DE PROTECCIÓN PARA REPELER AVES
HUAQUIAS EN LA EMPRESA PISCÍCOLA DE
HUIÑAMALCA PARCO”**

LÍNEA DE INVESTIGACIÓN:
CONTROL Y AUTOMATIZACIÓN

PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO ELECTRÓNICO

PRESENTADO POR LOS BACHILLERES:
ORDOÑEZ CASTILLO, Alexander
PEÑALOZA CONTRERAS, Ángeles Gregorio

PAMPAS 2014



ACTA DE SUSTENTACION DE TESIS

En el Auditorio de la Facultad de Ingeniería Electrónica – Sistemas, a los 14 días del mes de AGOSTO del año 2014, a horas 4:40 P.M., se reunieron; el Jurado Calificador, conformado de la siguiente manera:

Presidente: ING. INE. JAVIER FRANCISCO HARQUE CAMARENA

Secretario: ING. LUCIO BUISPE CARRION

Vocal: M.B. INE. HIPOLITO CARBAJAL MORAN

Designados con Resolución N° 254-2014-F.I.E.S.-U.N.H.; del: proyecto de investigación (Tesis), Titulado: "SISTEMA DE PROTECCION PARA REPETIR AVES HUABUJAS EN LA EMPRESA PISCICOLA DE HUINA MALCA PARCO"

Cuyos autores son los graduado (s):

BACHILLER (S): ANGELES GREGORIO PENALOZA CONTRERAS
ALEXANDER DR. DONER CASTILLO

A fin de proceder con la evaluación y calificación de la sustentación del proyecto de investigación, antes citado.

Finalizado la evaluación; se invito al público presente y a los sustentantes a abandonar el recinto; y, luego de una amplia deliberación por parte del jurado, se llegó al siguiente el resultado:

APROBADO POR MAYORIA

DESAPROBADO

En conformidad a lo actuado firmamos al pie.

[Firma]
 Presidente

[Firma]
 Secretario

[Firma]
 Vocal

A nuestra querida familia por su apoyo incondicional en nuestra formación profesional.

AGRADECIMIENTOS

A Dios, por brindarnos la oportunidad de realizarnos como persona y como profesional.

Al Ing. Raúl Padilla Sánchez, por habernos dado la asesoría y el apoyo necesario para la presente tesis.

A los docentes de la Escuela Académico Profesional de Electrónica de la Facultad de Ingeniería Electrónica-Sistemas de la Universidad Nacional de Huancavelica.

A nuestros padres, Doña Valeriana Castillo Baltazar y Don Filomeno Ordoñez Galarza que ya está gozando de la presencia del señor, de la misma manera a Doña Paula Contreras García y Don Leoncio Peñaloza Londoño, quienes nos apoyaron incondicionalmente a lo largo de este camino lleno de alegrías y dificultades.

Al administrador de la empresa Piscícola Huiñamalca, por facilitarnos la información y acceso a los ambientes necesarios para la realización de la presente tesis.

ÍNDICE

	Página
CAPÍTULO I.....	1
PROBLEMA.....	1
1.1. Planteamiento del problema.....	1
1.2. Formulación del problema.....	2
1.2.1. Problema general.....	2
1.2.2. Problemas específicos.....	2
1.3. Objetivos.....	2
1.3.1. Objetivo general.....	2
1.3.2. Objetivos específicos.....	2
1.4. Justificación.....	2
2.1 Antecedentes.....	4
2.2 Bases teóricas.....	6
2.2.1. La trucha Arco Iris (<i>Oncorhynchus mykiss</i>).....	6
2.2.1.1 Biología de la trucha.....	6
2.2.1.2 Etapas de desarrollo de la trucha.....	6
2.2.1.3 Ciclo biológico.....	7
2.2.1.4 Taxonomía.....	8
2.2.1.5 Parámetros de cultivo.....	9
2.2.1.6 Instalaciones para el cultivo.....	10
2.2.1.7 Factores que afectan la producción.....	11
2.2.1.8 Alimentación.....	12
2.2.1.9 Sanidad piscícola.....	14
2.2.2. Microcontrolador PIC16F877.....	16
2.2.2.1 Arquitectura Básica.....	17
2.2.2.2 PIC de Microchip.....	18
2.2.2.3 Nomenclatura.....	18
2.2.2.4 Gamas.....	19
2.2.2.5 Recursos auxiliares.....	22

2.2.2.6	PIC 16F877A	23
2.2.2.7	Características del PIC16F877A.....	24
2.2.2.8	Instrucciones para programar el PIC16F877	25
2.2.2.9	Qué se necesita para programarlos y el procedimiento.....	41
2.2.3.	Sensores	42
2.2.3.1	Sensor de presencia	42
2.2.3.2	Sensor fotoeléctrico LDR.....	44
2.2.4.	Pantalla LCD 16X2.....	46
2.2.5.	Fuente de alimentación	47
2.2.6.	Automatización	47
2.3	Hipótesis.....	50
2.3.1.	Hipótesis general.....	50
2.3.2.	Hipótesis específicos.....	50
2.4	Variables de estudio	51
2.4.1.	Variable independiente.....	51
2.4.2.	Variable dependiente	51
3.1	Ámbito de estudio.....	52
3.2	Tipo de investigación.....	52
3.3	Nivel de investigación.....	52
3.4	Método de investigación.....	52
3.5	Diseño de investigación.....	53
3.6	Población, muestra, muestreo	53
3.6.1.	Población	53
3.6.2.	Muestra	53
3.6.3.	Muestreo	54
3.7	Técnicas e instrumentos de recolección de datos	54
3.8	Procedimiento de recolección de datos.....	54
3.8.1.	Fase de pre-campo	54
3.8.2.	Fase de campo.....	54
3.8.3.	Fase de gabinete.....	54
3.9	Técnicas de procesamiento y análisis de datos	55

4.1 Presentación de resultados 56

 4.1.1 Diagrama de bloques del sistema de protección..... 56

 4.1.2 Diagrama de flujo del sistema de protección..... 57

 4.1.3 Diseño e implementación del sistema de protección..... 59

 4.1.4 Prueba de hipótesis..... 62

4.2 Discusión 68

CONCLUSIONES 69

RECOMENDACIONES..... 70

REFERENCIAS BIBLIOGRÁFICAS..... 71

ARTÍCULO CIENTÍFICO 73

ANEXOS..... 82

ÍNDICE DE FIGURAS

	Página
Figura N° 2.1: Ciclo de vida de la trucha arco iris	8
Figura N° 2.2: Partes de la trucha arco iris	9
Figura N° 2.3: Arquitectura Von Neumann de microprocesador	17
Figura N° 2.4: Arquitectura Harvard de microcontrolador (acceso a memorias).....	17
Figura N° 2.5: Las tres gamas en que se dividen los PIC.....	20
Figura N° 2.6: Diagrama de un PIC16F877A.....	24
Figura N° 2.7: Instrucciones de PIC 16F877A	26
Figura N° 2.8: Encapsulado del sensor de movimiento S320140	43
Figura N° 2.9: Zona de cobertura vista de planta del sensor de movimiento S320140.....	43
Figura N° 2.10: Conexión del sensor de movimiento S320140.....	44
Figura N° 2.11: Ubicación de LDR.....	45
Figura N° 2.12: El LDR	45
Figura N° 2.13: Conector de la pantalla LCD 16x2	46
Figura N° 2.14: Resumen de tipos de automatización	50
Figura N° 3.1: Diseño de investigación	53
Figura N° 4.1: Diagrama de bloques del sistema de protección.....	57
Figura N° 4.2 Diagrama de flujo.....	58
Figura N° 4.3: Elementos que forman parte del sistema de protección	59
Figura N° 4.4: Simulación de activación del sistema por la noche.....	60
Figura N° 4.5: Simulación de activación del sistema en el día.....	60
Figura N° 4.6: Simulación del estado de no activación del sistema	61
Figura N° 4.7: Circuito físico de sistema de protección implementado en placa.....	61
Figura N° 4.8: Maqueta de la Piscigranja con el circuito físico implementado	62
Figura N° 4.9: Prueba de t-Student.....	63
Figura N° 4.10: Resultados de grupo control	64
Figura N° 4.11: Resultados de grupo experimental	65
Figura N° 4.12: Resumen estadístico	65
Figura N° 4.13: Ubicación de tc	66

ÍNDICE DE TABLAS

	Página
Tabla N° 1.1: Resumen de pérdidas por ataques de aves Huaquias.....	1
Tabla N° 2.1: Racionamiento por día.....	13
Tabla N° 4.1: Pérdidas de alevinos en periodo de 7 meses	64
Tabla N° 4.2: Prueba de hipótesis con la "t" de Student	65

RESUMEN

El trabajo de investigación tuvo por objetivo principal: Determinar la influencia del diseño e implementación del sistema de protección para repeler las aves Huaquias en la empresa Piscícola Huiñamalca Parco, y como objetivos específicos: Describir el nivel de la tasa de pérdidas de alevinos, con el sistema de protección para repeler aves Huaquias en la Empresa piscícola Huiñamalca Parco y establecer la relación que existe entre el sistema de protección para repeler las aves Huaquias con la tasa de pérdidas de alevinos en la Empresa Piscícola Huiñamalca Parco. Se empleó como método: Método general de investigación el método comparativo, como método específico el método experimental y como método particular el que utiliza el control y automatización. Como resultado: Se diseñó e implementó el sistema de control para repeler aves Huaquias; se diseñó e implementó el sistema utilizando el PIC 16F877A y el lenguaje de programación C, se manejó las variables sistema de protección y tasa de pérdidas de alevinos, el sistema se activa si el PIR S320140 detecta la presencia de un ave y además si es de día activa una bocina y si es de noche una lámpara, en la memoria del microcontrolador se va guardando el número de presencia de aves de día y de noche.

Se trabajó con dos grupos uno experimental y otro control, se pudo observar que las aves Huaquias tienen una mejor respuesta esperada ante el sistema de día, esto se explica porque existe un personal que ronda permanentemente la piscigranja, como conclusión: Se ha demostrado con un nivel de confianza del 95% que la implementación del sistema de protección para repeler aves Huaquias ha influido significativamente en la disminución de la pérdida de alevinos de la empresa Huiñamalca Parco.

Palabras claves:

Sistema de protección, tasa de pérdida alevinos, repeler aves Huaquias, empresa piscícola.

INTRODUCCIÓN

La implementación del Sistema de Protección para Repeler Aves Huaquias, la necesitan muchas empresas piscícolas que quieren mejorar su sistema de producción de alevinos.

El sistema de protección no exige requisitos para su aplicación, ha sido elaborada para que las apliquen a las empresas piscícolas, sin importar su origen geográfico, social o cultural, la extensión de la aplicación de los factores que consideren la política de la empresa, la naturaleza las actividades y las condiciones que se encuentren.

Para empresas que logren implementar el sistema de protección obtienen beneficios que podemos mencionar:

Imagen: Las empresas que implementa el sistema de protección se ven beneficiadas en su imagen interna y externa.

Negociación: Al adoptar el sistema de protección las empresas tienen mayor poder de negociación, debido a que sus riesgos estarán identificados y controlados por procedimientos claramente identificados.

Competitividad: El hecho de asumir el sistema de protección, hará que las empresas puedan competir de igual a igual en los mercados nacionales e internacionales, Actualmente la globalización elimina las fronteras y las barreras de los diferentes productos y servicios que se ofrecen en los mercados mundiales.

Para su mayor comprensión, la presente investigación se ha dividido en 4 capítulos los cuales son los siguientes:

El Capítulo I, que trata sobre el problema de investigación; en donde se visualiza el planteamiento y formulación del problema, la justificación, los respectivos objetivos de investigación, seguido de la justificación e importancia.

El Capítulo II, del marco teórico conceptual de la investigación; donde se aprecia primero los antecedentes y luego la información teórica relevante sobre las variables de estudio, sustentado en teorías y bibliografía actualizada, seguido de la hipótesis y el sistema de variables.

El Capítulo III, del marco metodológico, en el cual se detalla el tipo, nivel método y diseño de investigación, además de las técnicas de recolección de datos y el procesamiento de información.

El Capítulo IV, de los resultados; donde se detalla los pormenores del análisis y procesamiento de la información tanto del pre test como del post test, seguido de la prueba de hipótesis.

Al final se complementa con las conclusiones, recomendaciones, referencias bibliográficas según el estilo Vancouver, y los respectivos anexos del presente trabajo de investigación.

Los autores.

CAPÍTULO I

PROBLEMA

1.1. Planteamiento del problema

Según información de la señora Caridad Cóndor Casachagua, una de las responsables del cuidado de las 07 pozas de crianzas de truchas de las cuales 03 son de alevinos de la empresa piscícola Huiñamalca-Parco, existen aves conocidas con el nombre de Huaquias de unos 25 cm de altura, color plomo, pecho blanco, cuello alargado y pico largo que atacan a los alevinos de una longitud comprendida entre los 5 a 12 cm, inicialmente este ataque se realizaba durante el día, con una frecuencia de 02 por día, siendo este el problema, se optó por espantar a las aves haciendo bullicios, frente a estos resultados las aves comenzaron a atacar por la madrugada a partir de las 3:30 de la mañana, por lo que se optó otro método para ahuyentarlos, prendiendo los reflectores de luz.

El Ing. Miguel Cóndor Casachagua hizo un estimado de las pérdidas que generan estos ataques a la empresa los cuales se resumen en el siguiente cuadro:

Tabla N° 1.1: Resumen de pérdidas de alevinos por ataques de aves Huaquias

AÑO	ES	PÉRDIDAS (Kg)
2012	octubre	17
2012	noviembre	15
2012	diciembre	22
2013	enero	17
2013	febrero	15
2013	marzo	28

Fuente: Empresa Piscícola Huiñamalca - Parco

1.2. Formulación del problema

1.2.1. Problema general

¿En qué medida el sistema de protección para repeler aves Huaquias, influirá en disminuir el ataque a los alevinos en la empresa Piscícola Huiñamalca Parco?

1.2.2. Problemas específicos

- a) ¿Cuál es la tasa de pérdidas de alevinos, con el sistema de protección para repeler aves Huaquias en la empresa Piscícola Huiñamalca Parco?
- b) ¿Qué relación existe entre el sistema de protección para repeler aves Huaquias con la tasa de pérdidas de alevinos en la empresa Piscícola Huiñamalca Parco?

1.3. Objetivos

1.3.1. Objetivo general

Determinar la influencia del sistema de protección para repeler las aves Huaquias en la empresa Piscícola Huiñamalca Parco.

1.3.2. Objetivos específicos

- a) Describir el nivel de la tasa de pérdidas de alevinos, con el sistema de protección para repeler aves Huaquias en la Empresa piscícola Huiñamalca Parco.
- b) Establecer la relación que existe entre el sistema de protección para repeler las aves Huaquias con la tasa de pérdidas de alevinos en la Empresa Piscícola Huiñamalca Parco.

1.4. Justificación

El Perú, y en especial la quebrada del Mantaro, departamento de Junín, presentan condiciones y recursos naturales posibles de utilizar, como son las numerosas vertientes que ofrecen condiciones ecológicas óptimas para la crianza de trucha arco iris. La alternativa tecnológica de crianza de truchas en el sistema de

estanques se basa en la oportunidad de la utilización de canales naturales, y en cuyas partes más amplias permite la crianza de truchas.

Para la presente tesis se ha considerado la trucha por ser el salmónido más apropiado para el cultivo industrial y para la producción de truchas de consumo, ya que posee una gran facilidad de adaptación y exhibe un rápido desarrollo.

Para tener un sistema de producción sostenida de truchas en el Perú en general, la quebrada del Mantaro en particular, es necesario superar el problema de los ataques de las aves Huaquias y proteger a los alevinos durante los dos primeros meses.

Con el desarrollo de esta tesis de investigación y los resultados que se obtengan se contribuirá a minimizar la tasa de pérdidas de alevinos debido a los ataques de estas aves, consiguiendo una óptima producción de truchas arco iris.

Consideramos que estas razones son suficientes para viabilizar la ejecución de la presente tesis; el mismo que permitirá en gran medida solucionar un problema crítico que limita el desarrollo de la acuicultura regional y nacional.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

2.1.1. **Jorge David Higa Díaz, Miguel Ángel Cataño Sánchez. Diseño del Sistema de Audio para un Espantapájaros Electrónico. Pontificia Universidad Católica del Perú, Lima Perú, 2009.**

En los campos agrícolas, las aves son un gran problema para los agricultores ya que estas se alimentan de los sembradíos y de las cosechas, ocasionando grandes pérdidas económicas. Asimismo, las plantaciones dan un lugar ideal para que las aves se multipliquen y así puedan ocasionar enfermedades a los animales de granja y corral. Las aves no sólo afectan a las zonas agrícolas, también a las urbanas, por ejemplo, las palomas se alimentan de los desperdicios y de alimentos que encuentran en la ciudad, anidando en casi cualquier lugar como templos, fachadas, entre otros.

Se espera que el espantapájaros electrónico ayude a resolver el problema de las aves, ya que al tener un sistema automático para el ahuyentamiento de aves esta puede ser más eficiente que los sistemas tradicionales. Además que se puede modificar la estrategia de ahuyentamiento mediante programación.

Actualmente, la universidad ha desarrollado una tesis con un primer modelo de espantapájaros. Con esta tesis se pretende mejorar la etapa de generación de sonido necesario para el espantapájaros. El sistema de audio es una etapa importante.

El objetivo de la tesis es diseñar el diagrama esquemático de un sistema de audio para el espantapájaros, sistema que sea pequeño, necesite poca energía y reproduzca las señales almacenadas de sonido. Con esto se espera que los impactos sean favorables para el campo y las ciudades, dependiendo donde sea aplicado, elevaría la productividad del agricultor y uno de los aspectos más importantes es que no elimina a las aves sino que solo las asusta con sonidos específicos para ese fin.¹

2.1.2. Juan Francisco Rodríguez Ayala. Diseño y Construcción de un Sistema Electrónico de Ahuyentamiento de Aves por medio de Recursos Sonoros y Visuales para la Protección de Campo de Cultivo, Pontificia Universidad Católica del Perú, Lima Perú, 2009

El consumo furtivo de uvas por parte de las aves es uno de los problemas más comunes en los campos de cultivo de la vid. En este sentido las aves son consideradas como una plaga, ya que en altas poblaciones perjudican la propiedad, la salud, los bienes y el ambiente del hombre.

En el departamento de Ica, la cosecha de uva para consumo doméstico y para producción de vinos y de pisco, se ve afectada por la plaga de las aves. En una coyuntura de resurgimiento económico regional e incremento de la exportación de productos de bandera como el Pisco, los agricultores se encuentran dispuestos a invertir recursos económicos para controlar este problema.

Existen actualmente productos electrónicos en el mercado que prometen reducir el ataque de las aves, pero que por desconocimiento por parte de los usuarios, no se aplican como es debido ni gozan de un soporte técnico que les permita maximizar la efectividad de estos aparatos.

Nuestra investigación aborda dos aspectos de esta problemática. El primer aspecto es la correcta aplicación de métodos de control de aves, lo cual nos lleva a definir un sistema de ahuyentamiento de aves. Otro aspecto de nuestra investigación consiste en el desarrollo de dispositivos electrónicos

¹ [Fecha de acceso: 06 de abril de 2013]. URL disponible en: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/698>

para aumentar la eficiencia de un sistema de ahuyentamiento de aves. Dichos dispositivos son potenciales productos accesibles a los usuarios del campo y por ende deben cumplir con los requisitos de operatividad y resistencia al medio ambiente rural.

Gran parte de la información respecto a ésta problemática fue obtenida de empresarios y especialistas iqueños. Realizamos experimentos en el fundo La Caravedo, en Ica, para probar la efectividad de nuestros métodos propuestos, y además recopilamos testimonios en audio y en video, los cuales nos ayudarán a comprender mejor el marco problemático de éste fenómeno.²

2.2 Bases teóricas

2.2.1. La trucha Arco Iris (Oncorhynchus mykiss) [URL1]

2.2.1.1 Biología de la trucha

Esta especie se caracteriza por tener el cuerpo cubierto con finas escamas y de forma fusiforme, la coloración de la trucha varía de acuerdo al ambiente en que vive, edad, estado de maduración sexual. La denominación de trucha arco iris se debe a la presencia de una franja decolores de diferentes tonalidades, con predominio de una franja rojiza sobre la línea lateral en ambos lados del cuerpo.

2.2.1.2 Etapas de desarrollo de la trucha

- El desarrollo biológico de la trucha comprende 5 etapas:
- a. Ova.- Son los huevos fecundados que después de un promedio aproximado de 30 días de incubación, eclosionan para convertirse en larva.
 - b. Alevino.- Son peces pequeños que miden de 3 cm. A 10 cm. Con un peso que oscila entre 1.5 gr. A 20 gr.

² [Fecha de acceso: 06 de abril de 2013]. URL disponible en: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/274>

- c. Juvenil.- Son peces que miden de 10 cm. A 15 cm. Cuyo peso es generalmente de 20 gr. A 100 gr.
- d. Comercial.- Es la etapa especial, donde los peces han recibido el proceso de engorde para ser comercializados, estos miden 15 cm. A 22 cm. Con un peso de 100 a 200 gr.

2.2.1.3 Ciclo biológico

Al cabo de cierto tiempo (aprox. 90-100 días para nuestra zona) el embrión ya desarrollado rompe la envoltura que lo contuvo y tenemos al alevino en sus primeros estados donde ya tiene movilidad propia y respiración branquial, no desprendiéndose de su saco vitelino que le servirá en estos primeros pasos de alimentación. Una vez dejado el saco ya se alimenta por si solo, adquiere forma y características propias de su especie. INPRINTED es un período fundamental donde el alevino de 6 a 8 cm. adquiere, a través de su dotación genética, las características del medio acuoso que lo rodea llevándolas consigo toda su vida, es decir, copia para sí los datos del lugar donde nació para posteriormente volver (ya siendo adulto) a ese mismo sitio aproximadamente para aparearse y recomenzar nuevamente el ciclo.

Esto así descripto hipotéticamente es ideal, pero el camino en la realidad muestra obstáculos a veces insalvables, no llegando muchas ocasiones a poder lograrlo, ya que no todos los que nacen tienen la suerte de poder desarrollarse. Ya desde la cama de desove no todos los huevos son fecundados y por ende no todos se transformarán en embriones. Viene luego la lucha por la supervivencia ante el acoso de los depredadores naturales que hacen disminuir en un gran porcentaje a aquellos que lograron nacer.

En estado natural solo conseguirán llegar a adultos capaces de

procrear muy pocos del total de huevos depositados. Una hembra adulta ovipone aproximadamente de 1.000 a 1.500 huevos por kilo de su peso, es decir, si por ejemplo una de 4 Kg. depositó aproximadamente 5.000 de ellos, lograrán nacer alrededor de 350 alevinos (7 % de natalidad en ambientes naturales). De aquí hasta llegar a ser reproductores solo lo alcanzarán 2 (dos) ejemplares de aquellos 5.000 huevos originales.

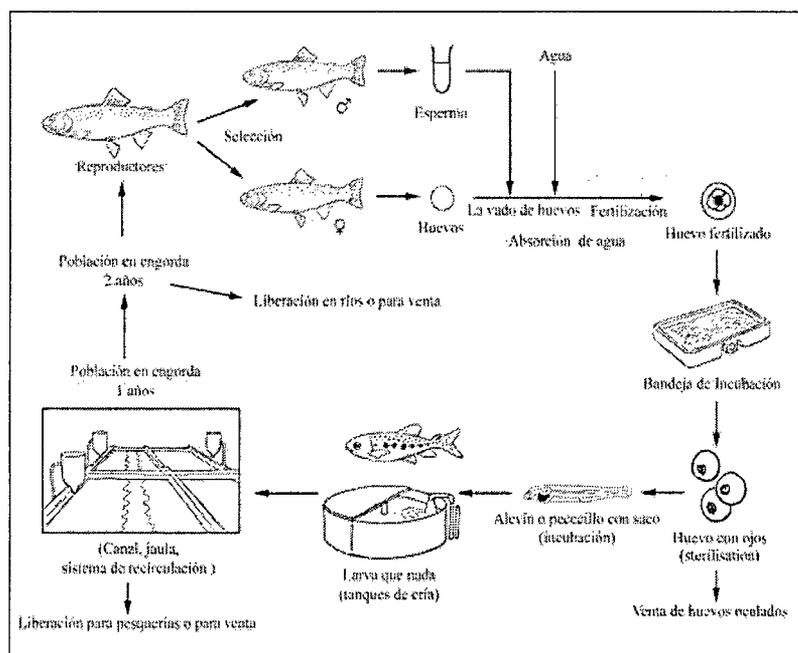


Figura Nº 2.1: Ciclo de vida de la trucha arco iris

2.2.1.4. Taxonomía

- Reino : Animal
- Sub Reino : Metazoa
- Phylum : Chordata
- Sub Phylum : Vertebrata
- Clase : Osteichthyes
- Sub Clase : Actinopterygii

Orden : Isospondyli
Sub Orden : Salmoneidei
Familia : Salmonidae
Género : Oncorhynchus
Especie : Oncorhynchusmykiss
Nombre Vulgar : Trucha arco iris

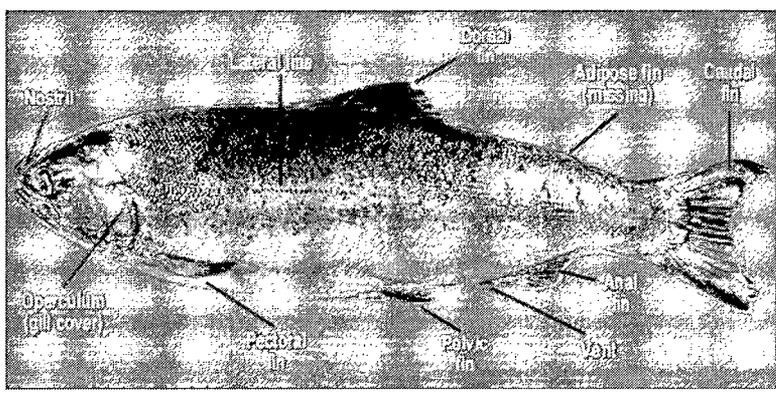


Figura N° 2.2: Partes de la trucha arco iris

2.2.1.5 Parámetros de cultivo

- **Recurso Hídrico:** El cuerpo de agua a utilizar, debe poseer características adecuadas en cuanto a su cantidad (caudal) y calidad (factores físico – químicos y biológicos). Las propiedades físicas, como temperatura, pH, oxígeno, transparencia, turbidez, etc. Pueden estar sometidas a variaciones bruscas por la influencia de factores externos. Fundamentalmente a cambios atmosféricos y climáticos. Las propiedades químicas, sin embargo son mucho más estables y sus variaciones son mínimas, salvo casos excepcionales en los que una contaminación pueda producir efectos irreversibles. La calidad del agua desde el punto de vista biológico, está condicionada a la ausencia o presencia de

organismos vivos en el ecosistema acuático, así como a la mayor o menor presencia de agentes patógenos.

- **Terreno:** Se debe asegurar una extensión de terreno suficiente, de preferencia de consistencia arcillosa, a fin de evitar filtraciones y pérdidas de agua. El terreno debe estar ubicado cerca al recurso hídrico y tener una pendiente topográfica moderada, entre 2 a 3%. Algunas propiedades físicas y químicas de un cuerpo de agua para la truchicultura

- **Rango Óptimo**

Temperatura del agua : 10 – 16°C

Oxígeno Disuelto : 6,5 – 9 ppm

PH : 6,5 – 8,5

CO2 : < 7ppm

2.2.1.6 Instalaciones para el cultivo

- **Estanques:** Recinto cerrado donde se almacena y circula una determinada cantidad del recurso hídrico, a fin de permitir el confinamiento de los peces para lograr su crianza y desarrollo, a expensas de una alimentación ofrecida por el piscicultor. Un estanque hace las veces de un hábitat artificial capaz de satisfacer las exigencias biológicas del animal en su medio natural, siendo de responsabilidad del piscicultor a su vez, la atención de las necesidades alimenticias y de protección sanitaria de los peces en cultivo, a fin de obtener resultados favorables en los niveles de producción esperados.

- **Tipo de Estanques:**

Estanque semi-natural.- Cuerpo de agua confinado que sufren cierto acondicionamiento por parte del hombre y se utiliza de preferencia aquel que se encuentran sobre terreno arcilloso, a fin de evitar filtraciones.

Estanque artificial.- Diseñado y construido especialmente con fines piscícolas, puede ser a tajo abierto o con material de concreto armado (cemento, ladrillo, refuerzo de piedras, etc.)

Estanque de presa.- Puede construirse a manera de un embalse y también como una secuencia de estanques aprovechando un declive del terreno, también es conocido como estanque con dique o de interceptación, generalmente se instala en la parte más baja de un valle, construyéndose un muro transversal que forma una pequeña presa de contención. El agua para este estanque proviene generalmente de un manantial o pequeños cursos de agua.

Estanques de derivación.- Se construyen aprovechando las características topográficas del terreno, de tal manera que el agua que los abastece es derivada del río, riachuelo o manantial hacia los estanques mediante un canal. Según la topografía del terreno y la cantidad de agua a utilizar dentro de los estanques de derivación.

Para la crianza intensiva de truchas, se debe diseñar y construir estanques con características adecuadas a las etapas de crianza o biológicas de la especie, puede emplearse cualquier forma o tamaño de estanques para cualquier etapa de crianza, pero con ciertas limitaciones de manejo, sin embargo una adecuada distribución de estanques para cada etapa biológica podrá permitir una crianza periódica, rotativa de alevines, juveniles, pre comerciales, comerciales y reproductores, y a la vez posibilitará el uso racional del agua.

2.2.1.7 Factores que afectan la producción [URL2]

- **Asociado al pez.-** Comportamiento del animal según estadio biológico, requerimiento nutricional, exigencia medio – ambiental, tasa de crecimiento, asimilación de alimento,

historial sanitario, factor de condición, canibalismo, captación de oxígeno, generación de sólidos fecales, stress, adaptación a elevadas densidades de carga, etc.

- **Asociado al agua.**- Oxígeno Disuelto, nitrito, nitratos, alcalinidad total, pH, caudal, amonio, sólidos suspendidos, sólidos sedimentados, temperatura, transparencia, contaminantes, DBO, viscosidad.
- **Asociado al Confinamiento.**- Volumen de agua, velocidad de agua, profundidad del estanque, recambio de agua, diseño de salidas, forma del estanque, etc.
- **Asociado a la nutrición.**- Tasa de alimentación, forma de alimentación, calidad nutricional del alimento, almacenamiento de alimento, etc.
- **Asociado al Manejo.**- Técnicas de muestreo, frecuencia de muestreos, técnicas de alimentación, densidad de carga, mantenimiento de los estanques, limpieza de los estanques, clasificación de tallas, manipuleo de reproductores, planificación de la producción, etc.³

2.2.1.8 Alimentación

Es un aspecto muy importante que se debe tener en cuenta a fin de proporcionarles el alimento adecuado, la ración adecuada en el momento adecuado. El alimento debe cubrir las necesidades de los peces tanto en lo que a energía se refiere, como a los diferentes tipos de aminoácidos y nutrientes que son requeridos para su desarrollo y crecimiento.

En la acuicultura se utilizan alimentos artificiales balanceados puesto que la trucha arco iris es una especie carnívora. Como nutrientes necesarios se puede citar proteínas, hidratos de

³ [Fecha de acceso: 2 de Abril del 2013]. URL disponible en: http://www.cedepperu.org/img_upload/c55e8774db1993203b76a6afddc95dc/manual_truchas_antamina.pdf

carbono, grasas, minerales, fibras y vitaminas. La formulación del alimento y tasa de alimentación diaria, se hace de acuerdo a los requerimientos del pez, tomando como referencia determinados parámetros como: tamaño, peso y estadio sexual del animal.

Para estimar la cantidad de alimento a suministrar diariamente a un estanque o jaula, se debe tener en cuenta la temperatura del agua, estadio del pez, biomasa total por estanque. Hay que tener en cuenta que la calidad y rendimiento del alimento se puede medir a través del índice de conversión alimenticia (cantidad de alimento que come y se transforma en peso vivo). Tabla de racionamiento por día.

Tabla N° 2.1: Racionamiento por día

Peso (g)	Talla (cm)	Tasa alimento	Conversión alimento	Dieta	Estado
0.19-0.7	2.56-4.0	10%	1.7:1	Inicio (polvo) 0.6 mm	Dedinos
0.7-3.0	4.0-6.5	8%	1.8:1	Inicio (1.0mm)	Alevinos
3.0-11	6.5-10	7%	1.9:1		
11.0-40	10-15	6%	1.7:1	Crecim 1(1.5mm)	Juveniles
40-90	15-20	4%	1.8:1	2(3.0mm)	
90-180	20-25	3%	1.9-1	Acabado (4.8 mm)	Talla comercial
180-333	25-31	2%	2-1		
333	31	1%	1.5:1	Acabado granulado (6mm)	Reproductores

Fuente: Empresa Piscícola Huiñamalca

Reglas de alimentación:

- La alimentación diaria y el cuidado de los peces en los estanques tiene prioridad.
- Un buen programa de alimentación incluye alimentar a los peces los 7 días de la semana.
- Se debe tener cuidado de no dar alimento cerca de la compuerta de salida donde la corriente puede llevarse al alimento fuera del estanques antes que el pez pueda consumirlo.

- El alimento deberá aumentarse cada 3 días.
- Los peces deben muestrearse cada cierto tiempo para determinar si están logrando la tasa de crecimiento esperado, de lo contrario la ración debe ser modificada.
- Los peces deben mantenerse sin alimentación 24 horas antes de seleccionarlos, manipularlos y/o transportarlos.
- Se debe llevar registros individuales en los estanques, las conversiones, porcentajes de ganancia, los flujos de agua, el oxígeno disuelto y la mortalidad.

2.2.1.9 Sanidad piscícola

Dentro de la tecnología de cultivo, la sanidad ocupa un lugar de interés por la necesidad que existe de conocer los procedimientos para prevenir y controlar las enfermedades que potencialmente limitan la producción.⁴

La prevención de las enfermedades es el mejor elemento de control y juega un papel importante en los cultivos de peces, teniendo en cuenta los cuidados de higiene de los estanques, el manejo de una densidad de carga adecuada, etc. El piscicultor debe ser capaz de detectar algunos de los problemas de carencia nutricional o de enfermedades infecciosas más comunes, debe aprovechar operación es de clasificación para examinar, medir y pesar a todos los peces, o cuando menos, a una muestra significativa, pero hay tener en cuenta lo siguiente:

Equipos: Deberían ser mantenidos separados (uno para cada estanque si es posible) y ser desinfectados después de cada uso, si esto no es posible por lo menos una vez por semana.

Estanques: Limpiar y desinfectar antes de usar si la naturaleza del estanque lo permite.

⁴ [Fecha de acceso: 06 de Mayo de 2013]. URL disponible en: <http://tr.scribd.com/doc/12867272/ TRUCHA-SEPARATA>

Movimiento de peces: De tiempo en tiempo es necesario clasificarlos, pero las diferentes poblaciones o grupos de peces no deberán ser mezclados. Si un grupo muestra signos de enfermedad, este deberá estar mantenido en estanques individuales y no distribuido por toda la granja.

Peces muertos: Deberán ser removidos diariamente e incinerados de inmediato. Las pérdidas de peces deberán ser registradas. Si el número de peces muertos se va incrementando podría deberse a condiciones inadecuadas del medio o a una enfermedad.

Las enfermedades de los peces, al igual que la de otros seres vivientes, se debe procurar prevenirlas en lo posible, así como detectar y controlarlas temprano. Para tales propósitos es necesario poner atención a los cuidados diarios con el fin de conocer bien el estado de salud de los peces y descubrir rápidamente cualquier anomalía.

Cuando se descubre la presencia de alguna enfermedad no deberá hacerse ningún tratamiento (medicación) bajo un criterio profano, por el riesgo de empeorar el cuadro con un diagnóstico y tratamiento erróneo, sino que se deberá solicitar los estudios pertinentes a las instituciones o profesionales especializados en el área.

Entre los aspectos a vigilar está la revisión de las branquias, las mismas que deben presentarse sanas, ya que su eficacia en la captación de oxígeno influye en su tasa de crecimiento.

Las branquias son una de las partes más sensibles del organismo del pez, por lo que resultan buenos indicadores de su estado general. Asimismo se debe revisar ojos, piel y aletas que permitan detectar a tiempo problemas sanitarios, antes de que se produzcan perjuicios relevantes para la población de truchas.

Causas de enfermedades:

- **Físicas:** Temperatura, contenido de materiales en suspensión, turbidez.
- **Químicas:** Cambios de pH., presencia de contaminantes orgánicos o inorgánicos, deficiencia de oxígeno, incremento del CO₂, etc.
- **Nutricional:** Carencia y desequilibrio de los principales nutrientes del alimento, como vitaminas y minerales de orden biológico.
- Deficiente manejo durante las mediciones, selección traslado, limpieza.⁵

2.2.2. Microcontrolador PIC16F877 [URL3]

Un microcontrolador es un circuito integrado que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una tarea determinada y debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de «controlador incrustado» (embeddedcontroller). Se dice que es “la solución en un chip” porque su reducido tamaño minimiza el número de componentes y el costo.

El microcontrolador es un computador dedicado. En su memoria sólo reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan la conexión de sensores y actuadores del dispositivo a controlar. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada.

Un microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se designa a gobernar una sola tarea.

⁵[Fecha de acceso: 06 de Mayo de 2013]. URL disponible en: [http://tr.scribd.com/doc/45639651/ Manual - Crianza-de-Truchas](http://tr.scribd.com/doc/45639651/Manual-Crianza-de-Truchas)

2.2.2.1 Arquitectura Básica

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de Von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

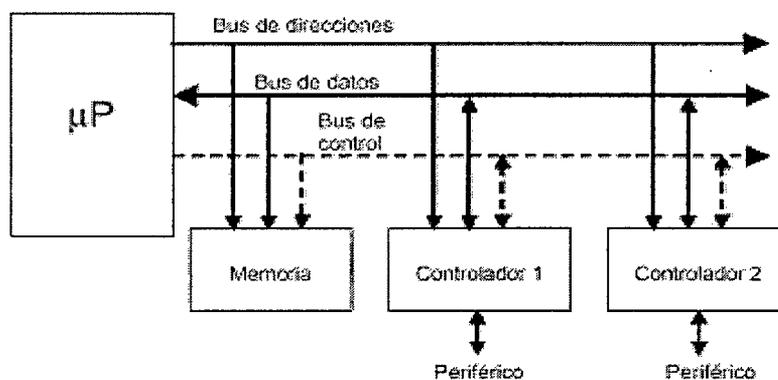


Figura Nº 2.3: Arquitectura Von Neumann de microprocesador

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

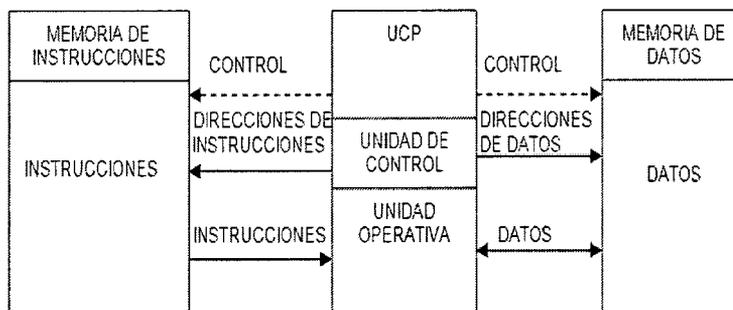


Figura Nº 2.4: Arquitectura Harvard de microcontrolador (acceso a memorias).

2.2.2.2 PIC de Microchip

En lo que sigue del documento comentaremos algunas de las características de los PIC de MicroChip que, por lo que hemos podido averiguar y a diferencia de otras marcas se consiguen fácilmente en las casas de electrónica de Montevideo, además de ser de los más vendidos del mundo.

Para conocer un chip particular es necesario revisar su Hoja de Datos. Es un compendio de todas las características de un chip, su juego de instrucciones, sus registros, las memorias, las funciones incorporadas, etc...

2.2.2.3 Nomenclatura

Un microcontrolador se identifica por su nombre, y este tiene varias partes.

Marca	: Fabricante de microcontrolador.
Gama	: Escala de microcontroladores (14 baja, 16 media, 18 alta).
Memoria Programable	: Tipo de memoria para el área de programa (F Flash, el micro se puede reprogramar, C el micro se puede programar una sola vez).
Modelo	: Modelo del microcontrolador.
Frecuencia	: Máxima frecuencia que soporta el microcontrolador usando reloj externo (en MHz).

Ejemplo, modelo de microcontrolador: PIC16F628/04

Marca	: PIC es el tipo de microcontrolador, fabricado por MicroChip.
Gama	: 16 indica la gama, este chip es de gama media.
Memoria Programable	: F memoria Flash.
Modelo	: 628.
Frecuencia	: 04 MHz.

2.2.2.4 Gamas

Para resolver aplicaciones sencillas se precisan pocos recursos; en cambio, las aplicaciones grandes requieren numerosos y potentes. Siguiendo esta filosofía, Microchip construye diversos modelos de microcontroladores orientados a cubrir, de forma, las necesidades de cada proyecto. Así, hay disponibles microcontroladores sencillos y baratos para atender las aplicaciones simples y otros complejos y más costosos para las de mucha envergadura.

Con las tres gamas de PIC se dispone de gran diversidad de modelos y encapsulados, pudiendo seleccionar el que mejor se acople a las necesidades de acuerdo con el tipo y capacidad de las memorias, el número de líneas de E/S y las funciones auxiliares precisas. Sin embargo, todas las versiones están construidas alrededor de una arquitectura común, un repertorio mínimo de instrucciones y un conjunto de opciones muy apreciadas, como el bajo consumo y el amplio margen del voltaje de alimentación. En la Figura N° 2.5 se muestra la distribución de los modelos de PIC en las tres gamas.

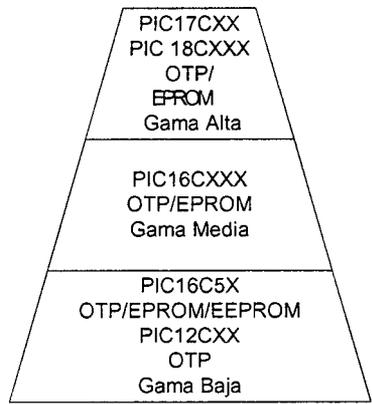


Figura N° 2.5: Las tres gamas en que se dividen los PIC.

▪ **Gama Baja**

La memoria de programa puede contener 512, 1 k. y 2 k palabras de 12 bits, y ser de tipo ROM, EPROM. También hay modelos con memoria OTP, que sólo puede ser grabada una vez por el usuario. La memoria de datos puede tener una capacidad comprendida entre 25 y 73 bytes. Sólo disponen de un temporizador (TMR0), un repertorio de 33 instrucciones y un número de patitas para soportar las E/S comprendido entre 12 y 20. El voltaje de alimentación admite un valor muy flexible comprendido entre 2 y 6,25 V, lo cual posibilita el funcionamiento mediante pilas corrientes teniendo en cuenta su bajo consumo (menos de 2 mA a 5 V y 4 MHz).

▪ **Gama Media**

En esta gama sus componentes añaden nuevas prestaciones a las que poseían los de la gama baja, haciéndoles más adecuados en las aplicaciones complejas. Admiten interrupciones, poseen comparadores de magnitudes analógicas, convertidores A/D, puertos serie y diversos temporizadores.

Algunos modelos disponen de una memoria de instrucciones del tipo OTP (One Time Programmable), que sólo la puede

grabar una vez el usuario y que resulta mucho más económica en la implementación de prototipos y pequeñas series.

Hay modelos de esta gama que disponen de una memoria de instrucciones tipo EEPROM, que, al ser borrables eléctricamente, son mucho más fáciles de reprogramar que las EPROM, que tienen que ser sometidas a rayos ultravioleta durante un tiempo determinado para realizar dicha operación.

- **Gama Alta**

Los dispositivos PIC17C4X responden a microcontroladores de arquitectura abierta pudiéndose expansionar en el exterior al poder sacar los buses de datos, direcciones y control. Así se pueden configurar sistemas similares a los que utilizan los microprocesadores convencionales, siendo capaces de ampliar la configuración interna del PIC añadiendo nuevos dispositivos de memoria y de E/S externas. Esta facultad obliga a estos componentes a tener un elevado número de patas comprendido entre 40 y 44. Admiten interrupciones, poseen puerto serie, varios temporizadores y mayores capacidades de memoria, que alcanza el 8 k palabras en la memoria de instrucciones y 454 bytes en la memoria de datos.

Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y casi 60 los de la alta.

Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. La característica de instrucciones de largo fijo beneficia la optimización del uso de zona de memoria de programa y facilita la construcción de ensambladores y compiladores.

2.2.2.5 Recursos auxiliares

Funcionalidades propias de cada microcontrolador que se ejecutan en paralelo con el programa.

Funciones incorporadas:

- **Circuito de reloj.**

El microcontrolador incluye un reloj para ser usado como reloj del micro, este generalmente no alcanza la mayor velocidad para la que el micro está preparado y es algo impreciso, sobre todo cuando sube la temperatura, si se requiere es posible alimentar al micro con una señal de reloj externa y no utilizar el reloj interno, de todas maneras tiene la gran ventaja de estar dentro del chip, lo que reduce problemas de confiabilidad (tener un reloj externo es un punto más de falla).

- **Temporizadores.**

El micro incluye varios timers que están disponibles en versiones de 8 o 16 bits, usarán la señal de reloj vigente para generar sus incrementos.

- **Perro guardián (Watchdog).**

Es una forma de control que permite al PIC reaccionar frente a iteraciones infinitas y cuelgues, es un contador que está decrementándose todo el tiempo, y cuando llega a 0, resetea el micro. Se debe ser cuidadoso al utilizarlo dado que, por programa se debe impedir que el contador llegue a 0, porque si lo hace reinicia el micro.

- **Convertidores A/D y D/A.**

Los convertidores Analógico Digitales y Digitales Analógicos están integrados a los chips de la gama alta y son capaces de realizar las conversiones adecuadas.

- **Comparadores analógicos.**

Estos comparadores incluidos desde algunos micros de la gama media, comparan dos señales analógicas y determinan si son iguales.

- **Protección ante fallos de alimentación.**

El micro está preparado para reaccionar a una caída de tensión reseteándose y no volviendo a operar hasta que el nivel de tensión sea el adecuado.

- **Estado de reposo en bajo consumo.**

El micro puede entrar en modo Stand By por su propia programación, esto es particularmente útil cuando se están usando baterías en lugar de estar conectado a una línea.

Las funcionalidades recién detalladas están disponibles dependiendo del modelo de controlador.

2.2.2.6 PIC 16F877A

El 16F877A es de la familia del 876 y tiene sus mismas características, excepto que tiene 40 patas de las que 33 son I/O.

Los detalles de arquitectura y configuración son los mismos del modelo anterior por ser de la misma familia.

La distribución de las patas de un chip de este modelo es la que sigue:

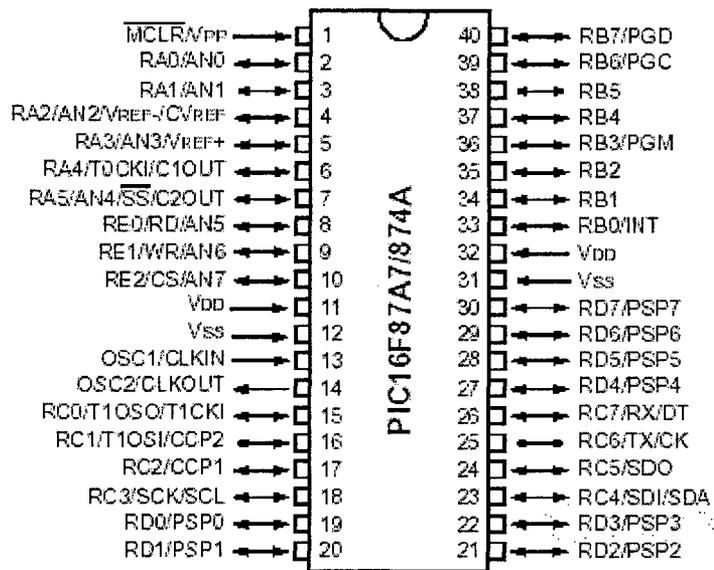


Figura N° 2.6: Diagrama de un PIC16F877A

2.2.2.7 Características del PIC16F877A

Microprocesador RISC de alto desempeño

- Instrucciones de un ciclo, excepto para los saltos
- Velocidades de operación desde DC hasta 20MHz
- 8k de memoria de programa
- 368 bytes de memoria de RAM
- 256 bytes de EEPROM
- 35 instrucciones de 1 palabra

Periféricos

- TIMER0: contador/temporizador de 8 bits con un prescaler programable de 8 bits
- TIMER1: contador/temporizador de 16 bits con capacidad para reloj o cristal externo
- TIMER2: contador/temporizador de 8 bits con registro de periodo de 8bits, prescaler y ostscaler
- Dos módulos de captura, comparación y modulación de ancho de pulso PWM:

- ✓ Captura de 16 bits, resolución máxima 12,5ns
- ✓ Comparación de 16 bits, resolución máxima 200ns
- ✓ Resolución máxima PWM de 10 bits
- Puerto de serie sincrónico (SSP) con SPITM (maestro) e I2CTM (maestro/esclavo)
- Receptor transmisor sincrónico asincrónico universal (USART/SCI), con detección de direcciones de 9 bits
- Puerto paralelo esclavo (PSP) de 8 bits con control externo.

Características analógicas

- Convertidor A/D de 8 canales / 10 bits
- Reset por desvanecimiento (BOR)
- Comparador analógico
 - ✓ Dos comparadores
 - ✓ Referencia de voltaje programable en el chip
 - ✓ La salida del comparador son accesibles externamente.

Características especiales del microcontrolador

El fabricante en su hoja de especificaciones afirma que la memoria del programa se puede grabar y borrar unas 100.000 veces, la memoria de datos EEPROM 1000000 de veces y sus datos permanecen almacenados por más de 40 años. Otra de las características son las programaciones en serie en el circuito (ICSPTM) que requiere un total de 5 pines, WDT con oscilador independiente para mayor confiabilidad, protección de código programable, modo de bajo consumo (sleep), múltiples opciones de oscilador externo, depuración en el circuito por medio de 2 pines, rango amplio de voltaje de operación (desde 2V a 5.5V) y bajo consumo.

2.2.2.8 Instrucciones para programar el PIC16F877

Estos microcontroladores responden a una serie de instrucciones o códigos que se deben grabar en su memoria de programa, en total son 35. A continuación se encuentra una tabla con la lista

completa y después una descripción de cada una de ellas con el fin de facilitar su aprendizaje. [3]

Si d = 0 el resultado de la operación se almacena en el registro W
Si d = 1 el resultado se almacena en el registro utilizado

Operaciones orientadas a registros			
Nemotécnico	Operación	Cód. de operación (14 bits)	Estados afectados
ADEWF f,d	Sumar W y f	00 0111 dfff ffff	C,DC,Z
ANDWF f,d	AND entre W y f	00 0101 dfff ffff	Z
CLRF f	Limpiar f	00 0001 1fff ffff	Z
CLRWF	Limpiar W	00 0001 0XXX XXXX	Z
COMF f,d	Complementar f	00 1001 dfff ffff	Z
DECf f,d	Decrementar f	00 0011 dfff ffff	Z
DECFSZ f,d	Decrementar f, saltar si cero	00 1011 dfff ffff	Z
INCF f,d	Incrementar f	00 1010 dfff ffff	Z
INCFSZ f,d	Incrementar f, saltar si cero	00 1111 dfff ffff	Z
IORWF f,d	OR entre W y f	00 0100 dfff ffff	Z
MOVF f,d	Mover f	00 1000 dfff ffff	Z
MOVWF f	Mover W a f	00 0000 1fff ffff	Z
NOP	No operación	00 0000 0XXX 0000	
RLF f,d	Rotar a la izquierda a través del carry	00 1101 dfff ffff	C
RRF f,d	Rotar a la derecha a través del carry	00 1100 dfff ffff	C,DC,Z
SUBWF f,d	Restar W de f	00 0010 dfff ffff	Z
SWAPF f,d	Intercambiar nibbles de f	00 1110 dfff ffff	Z
XORWF f,d	OR exclusiva entre W y f	00 0110 dfff ffff	
Operaciones orientadas a bits			
BCF f,b	Limpiar bit b de f	01 00bb bfff ffff	
BSF f,b	Activar bit b de f	01 01bb bfff ffff	
BTFSZ f,b	Probar bit b de f, saltar si es cero	01 10bb bfff ffff	
BTFS f,b	Probar bit b de f, saltar si es uno	01 11bb bfff ffff	
Operaciones con constantes y de control			
ADDLW k	Sumar literal k a W	11 111X kkkk kkkk	C,DC,Z
ANDLW k	AND entre k y W	11 1001 kkkk kkkk	Z
CALL k	Llamar subrutina	10 0kkk kkkk kkkk	
CLRWDIT	Limpiar WDT	00 0000 0110 0100	T0,PD
GOTO k	Salta a dirección k	10 1kkk kkkk kkkk	
IORLW k	OR entre k y W	11 1000 kkkk kkkk	Z
MOVLW k	Cargar a W con literal k	11 00XX kkkk kkkk	
RETFIE	Retornar de interrupción	00 0000 0000 1001	
RETLW k	Retornar y cargar a W con k	11 01XX kkkk kkkk	
RETURN	Retornar de subrutina	00 0000 0000 1000	T0,PD
SLEEP	Ir al modo de bajo consumo	00 0000 0110 0011	C,DC,Z
SUBLW k	Restarle k a W	11 110X kkkk kkkk	Z
XORLW k	OR exclusiva entre k y W	11 1010 kkkk kkkk	

Figura N° 2.7: Instrucciones de PIC 16F877A

Instrucción ADDLW: Suma un valor constante k al contenido del registro W, el resultado se guarda en el registro W.

Sintaxis : ADDLW k

Operación : (W) + k

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: C, DC, Z

Ejemplo: ADDLW d'15'

Antes de la instrucción registro W = d'10'

Después de la instrucción registro W = d'25'

Instrucción ADDWF: Suma el contenido de un registro f al contenido del registro W, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : ADDWF f,d

Operación : (W) + (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: C, DC, Z

Ejemplo: ADDWF 15h,0

Antes de la instrucción registro W = 17h registro 15h = 0C2h

Después de la instrucción registro W = 0D9h registro 15h = 0C2h

Instrucción ANDLW: Operación lógica AND entre una constante k y el registro W, la operación se hace bit a bit, el resultado queda en el registro W.

Sintaxis : ANDLW k

Operación : (W) AND (k)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo: ANDLW b'10101010'

Antes de la instrucción registro W = b'11110000'

Después de la instrucción registro W = b'10100000'

Instrucción ANDWF: Operación lógica AND entre un registro f y el registro W, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : ANDWF f,d

Operación : (W) AND (f)

Ciclos de instrucción : 1

Instrucciones para programar el PIC16F877

5 Bits del registro de estados que se afectan: Z

Ejemplo: ANDWF FSR,1

Antes de la instrucción registro W = b'00010111' registro FSR = b'11000010'

Después de la instrucción registro W = b'00010111' registro FSR = b'00000010'

Instrucción BCF: Pone en cero el bit b del registro f.

Sintaxis : BCF f,b

Operación : 0 (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo: BCF regis,7

Antes de la instrucción regis = b'11111111'

Después de la instrucción regis = b'01111111'

Instrucción BSF: Pone en uno el bit b del registro f.

Sintaxis : BSF f,b

Operación : 1 (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo: BSF regis,2

Antes de la instrucción regis = b'00000000'

Después de la instrucción regis = b'00000100'

Instrucción BTFSC: Pregunta por el bit b del registro f, si dicho bit está en cero el programa se salta una línea y ejecuta la instrucción que allí se encuentre, si el bit estaba en uno no salta una línea sino que ejecuta la instrucción inmediatamente siguiente.

Sintaxis : BTFSC f,b

Operación : salta si (f) = 0

Ciclos de instrucción : 1 ó 2

Bits del registro de estados que se afectan: Ninguno

Ejemplo: línea 1 BTFSC regis,0

Línea 2 GOTO inicio

Línea 3

Antes de la instrucción apuntador de programa = línea 1

Después de la instrucción si el bit 0 del registro regis = 0

Apuntador de programa = línea 3

Si el bit 0 del registro regis = 1

Apuntador de programa = línea 2

Instrucción BTFSS: Pregunta por el bit b del registro f, si dicho bit está en uno el programa se salta una línea y ejecuta la instrucción que allí se encuentre, si el bit estaba en cero no salta una línea sino que ejecuta la instrucción inmediatamente siguiente.

Sintaxis : BTFSS f,b

Operación : salta si (f) = 1

Ciclos de instrucción : 1 ó 2

Bits del registro de estados que se afectan: Ninguno

Ejemplo: línea 1 BTFSS regis,0

línea 2 GOTO inicio

línea 3

Antes de la instrucción apuntador de programa = línea 1

Después de la instrucción si el bit 0 del registro regis = 1

Apuntador de programa = línea 3

Si el bit 0 del registro regis = 0

Apuntador de programa = línea 2

Instrucción CALL: Llama una subrutina que está ubicada en la posición de memoria o etiqueta "k".

Sintaxis : CALL k

Operación : Carga el apuntador de programa con la dirección "K"

Ciclos de instrucción : 2

Bits del registro de estados que se afectan: Ninguno

Ejemplo: aquí CALL rutina

Antes de la instrucción contador de programa = aquí

Después de la instrucción contador de programa = rutina

En la pila se guarda la dirección "aquí" para regresar

Instrucción CLRF: Borra el contenido del registro "f", lo carga con 00.

Sintaxis : CLRF f

Operación : borra el contenido del registro f (lo carga con 00)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo: CLRF regis

Antes de la instrucción regis = 5Ah

Después de la instrucción regis = 00

Instrucción CLRW: Borra el contenido del registro W (lo carga con 00)

Sintaxis : CLRW

Operación : Registro W = 00

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo: CLRW

Antes de la instrucción registro W = 5Ah

Después de la instrucción registro W = 00

Instrucción CLRWDT: Reinicia en cero el conteo del temporizador interno Watchdog Timer, para que no sea reseteado el microcontrolador.

Sintaxis : CLRWDT

Operación : Contador del temporizador watchdog timer = 00

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: TO, PD

Ejemplo: CLRWDT

Antes de la instrucción Contador WDT =?

Después de la instrucción Contador WDT = 00

Bits del registro de estados: TO = 1, PD = 1

Instrucción COMF: Complementa el contenido del registro "f" (cambia unos por ceros y viceversa), el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : COMF f,d

Operación : (complementar registro f)

Ciclos de instrucción : 1

8Bits del registro de estados que se afectan: Z

Ejemplo: COMF regis,0

Antes de la instrucción regis = b'00101100', W = ?

Después de la instrucción regis = b'00101100', W = b'11010011'

Instrucción DECF: Decrementa en uno el contenido del registro "f", el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : DECF f,d

Operación : (f) - 1

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo: DECF regis,1

Antes de la instrucción regis = d'13'

Después de la instrucción regis = d'12'

Instrucción DECFSZ: Decremento en uno el contenido del registro "f", si el contenido queda en 00 el micro salta una línea del programa, el resultado obtenido se guarda en el registro escogido según el valor del bit d.

Sintaxis : DECFSZ f,d

Operación : (f) - 1, salta una línea si el resultado es cero

Ciclos de instrucción : 1 (2)

Bits del registro de estados que se afectan: Ninguno

Ejemplo: aquí DECFSZ regis, 1
 GOTO ciclo
 Continúa.....

Antes de la instrucción contador de programa = aquí

Después de la instrucción regis = regis - 1

si regis = 0, entonces

contador de programa = continua

si regis no es 0, entonces

Contador de programa = aquí + 1 (ejecuta GOTO ciclo)

Instrucción GOTO: El contador de programa salta a la dirección "k"

Sintaxis : GOTO k

Operación : El contador de programa salta a la dirección k

Ciclos de instrucción : 2

Bits del registro de estados que se afectan: Ninguno

Ejemplo: GOTO ciclo

Antes de la instrucción contador de programa = ?

Después de la instrucción contador de programa = ciclo

Instrucción INCF: Incrementa en uno el contenido del registro "f", el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : INCF f,d

Operación : (f) + 1

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo : INCF regis,1

Antes de la instrucción regis = d'24'

Después de la instrucción `regis = d'25'`

Instrucción INCFSZ: Incrementa en uno el contenido del registro "f", si el contenido de "f" queda en 00 el micro salta una línea del programa, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : `INCFSZ f,d`

Operación : $(f) + 1$, salta si el resultado es cero

Ciclos de instrucción : 1 (2)

Bits del registro de estados que se afectan: Ninguno

Ejemplo: aquí `DECFSZ regis, 1`

GOTO ciclo

Continúa.....

Antes de la instrucción contador de programa = aquí

Después de la instrucción `regis = regis + 1`

Si `regis = 0`, entonces

Contador de programa = continúa

Si `regis` no es 0, entonces

Contador de programa = aquí + 1 (ejecuta GOTO ciclo)

Instrucción IORLW: Operación lógica OR entre el contenido del registro W y la constante k, el resultado queda en el registro W.

Sintaxis : `IORLW k`

Operación : $(W) \text{ OR } (k)$

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo: `IORLW b'10101100'`

Antes de la instrucción registro W = b'00001111'

Después de la instrucción registro W = b'10101111'

Instrucción IORWF: Operación lógica OR entre el registro W y el registro "f", el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : IORWF f,d

Operación : (W) OR (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo: IORWF regis,0

Antes de la instrucción regis = b'00110011 , W = b'11110000'

Después de la instrucción regis =b'00110011' , W = b'11110011'

Instrucción MOVLW: Carga el registro W con el valor constante k.

Sintaxis : MOVLW k

Operación : (W) se carga con el valor k

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo: MOVLW 5Ah

Antes de la instrucción W =?

Después de la instrucción W = 5Ah

Instrucción MOVF: Mueve el contenido del registro "f" hacia el registro W.

Sintaxis : MOVF f

Operación : (W) se carga con (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo: MOVF regis,0

Antes de la instrucción registro W = ?, regis = d'32'

Después de la instrucción registro W = d'32'

Instrucción MOVWF: Mueve el contenido del registro W al registro "f".

Sintaxis : MOVWF f

Operación : (W) se mueve al registro (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo: MOVWF 20h

Antes de la instrucción registro 20h = ? , registro W = d'48'

Después de la instrucción registro 20h = d'48', registro W = d'48'

Instrucción NOP: No hace nada.

Sintaxis : NOP

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Instrucción RETURN: Retorno desde una subrutina.

Sintaxis: RETURN

Operación: carga el contador de programa con la dirección donde debe regresar luego de ejecutar la subrutina, la cual estaba guardada en la pila o stack.

Ciclos de instrucción: 2

Bits del registro de estados que se afectan: Ninguno

Ejemplo : RETURN

Después de la interrupción, contador de programa = dato de la pila

Instrucción RETLW: Retorno desde una subrutina y adicionalmente carga el registro W con el valor constante k.

Sintaxis : RETLW k

Operación : contador de programa se carga con el valor de la pila, además (W) = k

Ciclos de instrucción: 2

Bits del registro de estados que se afectan: Ninguno

Ejemplo : CALL tabla
tabla NOP
NOP

RETLW k1

Antes de la instrucción registro W = ?

Después de la instrucción registro W = k1, contador de programa = dato de la pila

Instrucción RETFIE: Retorna al programa luego de ejecutar una interrupción.

Sintaxis: RETFIE

Operación: pila k contador de programa, 1 k intcon,gie

Ciclos de instrucción: 2

Bits del registro de estados que se afectan: Ninguno

Ejemplo: RETFIE

Antes de la instrucción contador de programa = ?

Después de la instrucción contador de programa = dato de la pila, las interrupciones se habilitan nuevamente.

Instrucción RLF: Rotar el contenido del registro "f" un bit a la izquierda, usando el carry como bit intermedio, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : RLF f,d

Operación : Rota el contenido del registro "f" a la izquierda a través del carry.

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: C

Ejemplo: RLF regis,0

Antes de la instrucción bit carry = 0, regis = 1110 0110, W=?

Después de la instrucción bit carry = 1, regis = 1110 0110

W = 11001100

Instrucción RRF: Rotar el contenido del registro "f" un bit a la derecha, usando el carry como bit intermedio, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : RRF f,d

Operación : Rota el contenido del registro "f" a la derecha a través del carry

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: C

Ejemplo: RRF regis,0

Antes de la instrucción bit carry = 0, regis = 1110 0101, W=?

Después de la instrucción bit carry = 1, regis = 1110 0101

W = 0111 0010

Instrucción SLEEP: El micro entra en un estado de standby (dormido), de ese modo solamente sale si ocurre una interrupción.

Sintaxis : SLEEP

Operación : bit TO = 1, bit PD = 0

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: TO, PD

Ejemplo: SLEEP

Instrucción SUBLW: Resta el contenido del registro W de la constante k usando el método de complemento a dos, el resultado se guarda en el registro W, el bit de carry indica el signo de la respuesta.

Sintaxis : SUBLW k

Operación : $k - (W)$

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: C, DC, Z

Ejemplo 1: SUBLW 5h

Antes de la instrucción W = 1h, bit de carry = ?

Después de la instrucción W = 4h, bit de carry = 1 (positivo)

Ejemplo 2: SUBLW 5h

Antes de la instrucción W = 5h, bit de carry = ?

Después de la instrucción W = 0, bit de carry = 1 (positivo)

Ejemplo 3: SUBLW 5h

Antes de la instrucción W = 6h, bit de carry = ?

Después de la instrucción W = 0FFh , bit de carry = 0 (negativo)

Instrucción SUBWF: Resta el contenido del registro W del registro "f" usando el método de complemento a dos, el bit de carry indica el signo de la respuesta, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : SUBWF f,d

Operación : (f) - (W)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: C, DC, Z

Ejemplo 1: SUBWF regis,1

Antes de la instrucción regis = 5h, W = 2h, bit de carry = ?

Después de la instrucción regis = 3h, W = 2h, bit de carry = 1
(positivo)

Ejemplo 2: SUBWF regis,1

Antes de la instrucción regis = 2h, W = 2h, bit de carry = ?

Después de la instrucción regis = 0, W = 2h, bit de carry = 1
(positivo)

Ejemplo 3: SUBWF regis,1

Antes de la instrucción regis = 5h, W = 6h, bit de carry = ?

Después de la instrucción regis = 0FFh, W = 6h, bit de carry = 0
(negativo)

Instrucción SWAPF: Intercambia los cuatro bits altos y los cuatro bits bajos del registro "f", el nuevo dato obtenido se guarda en el registro seleccionado según el valor del bit d.

Sintaxis : SWAPF f,d

Operación : los bits f<3:0> se intercambian de posición con los bits f<7:4>

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo 1: SWAPF regis,0

Antes de la instrucción regis = b'10101100', W = ?

Después de la instrucción regis = b'10101100', W = b'11001010'

Instrucción XORLW: Operación lógica XOR entre el registro W y la constante k, el resultado se guarda en el registro W.

Sintaxis : XORLW k

Operación : (W) XOR (k)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo: XORLW b'10101100'

Antes de la instrucción registro W = b'11110000'

Después de la instrucción registro W = b'01011100'

Instrucción XORWF: Operación lógica XOR entre el registro W y el registro "f", el resultado obtenido se guarda en el registro seleccionado según el bit d.

Sintaxis : XORWF f,d

Operación : (W) XOR (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo: XORWF regis,1

Antes de la instrucción regis = b'11110000', W = b'10101010'

Después de la instrucción regis = b'01011010', W = b'10101010'9

2.2.2.9 Qué se necesita para programarlos y el procedimiento

Los componentes necesarios para programar un PIC son:

- Un editor de texto,
- Un compilador para el lenguaje que se decida usar (para el modelo adecuado),
- Un programador (tanto el circuito como el utilitario que baja el binario al PIC).

Es usual utilizar entornos de desarrollo del tipo MicroChip MPLAB, este es un software desarrollado por la empresa MicroChip que se baja gratis de su página, que permite editar código fuente y luego compilarlo en una variada cantidad de micros.

Si se cuenta con un programador de MicroChip (el PICSTAR), se lo puede utilizar para además cargar el micro con el programa. Si no será necesario otro software como el ICProg que es un utilitario capaz de pasarle al micro un código binario a través de un programador genérico.

Un programador es un circuito que se conecta al PC por un puerto (hay programadores para puerto serial, para puerto paralelo e inclusive algunos modelos para USB, siendo el más común y simple el serial), y baja al PIC el programa insertándolo en la memoria FLASH, carga la EPROM con datos (si se incluyen en el ICProg, por ejemplo), y por último se carga el seteo de la palabra de configuración.

Los lenguajes de programación que provee Microchip se reducen al Assembler para PIC en la gama baja y media, pero en la gama alta se encuentran compiladores para C y otros lenguajes menos conocidos de alto nivel.

Existen fabricantes de software que han desarrollado compiladores como PIC-Basic o PICC que se venden por

separado, estos si se compilan para varios micros de la gama media y alta.

El MPLAB incluye un simulador que es capaz de ejecutar como el PIC todo el programa, inclusive permite que se den pulsos en las distintas patas tanto en forma preorganizada (a través de archivos) como en forma puramente asincrónica mediante botones. Además se pueden ejecutar las distintas sentencias paso a paso e ir viendo como evolucionan los estados de corriente de cada pata o los valores de cada registro. De todas maneras tiene limitaciones y aún no todos los chips son soportados. [1]

Por lo tanto el procedimiento es sencillo:

- Se diseña y construye el circuito adecuado para resolver el problema.
- Se diseña y programa la fuente para comandar la tarea con el MPLAB.
- Se compila la fuente.
- Se usa el simulador para corroborar que esté todo funcionando como es esperado.
- Se carga el microchip con el ICProg.
- Se inserta el micro en el circuito.
- Se alimenta y comienza a funcionar.

2.2.3. Sensores

2.2.3.1 Sensor de presencia [URL4]

Los sensores de presencia que se obtienen en la industria son resultado de la necesidad de contar con indicadores de posición en los que no existe contacto mecánico entre el actuador y el detector. Pueden ser de tipo lineal (detectores de desplazamiento)

o de tipo conmutador (la conmutación entre dos estados indica una posición particular). Hay dos tipos de detectores de proximidad y/o presencia muy utilizados en la industria: inductivos y capacitivos⁶.

El sensor PIR detector de movimientos por infrarrojos resulta muy adecuado para su empleo en robots, gracias a su pequeño tamaño y bajo consumo. El sensor incluye una lente tipo fresnel de plástico que le proporciona un alcance de 5 metros y un ángulo de detección de 60°. La señal de salida es compatible TTL y la alimentación es de 5V con un consumo de tan solo 350 uA mientras está en reposo. Sus reducidas dimensiones de solo 25 x 35 x 18 mm hacen posible su utilización en todo tipo de robots y dispositivos sensores. [2]

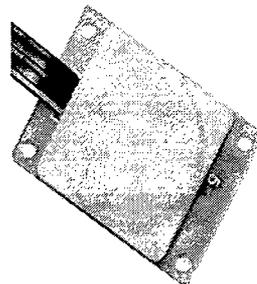


Figura N° 2.8: Encapsulado del sensor de movimiento S320140

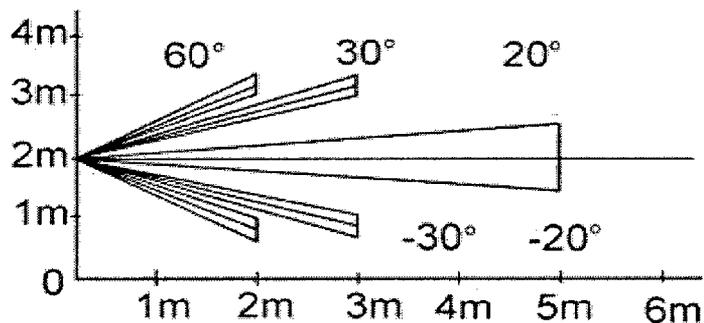


Figura N° 2.9: Zona de cobertura vista de planta del sensor de movimiento S320140

⁶ [Fecha de acceso: 19 de setiembre del 2013]. http://robots-argentina.com.ar/Sensores_general.htm

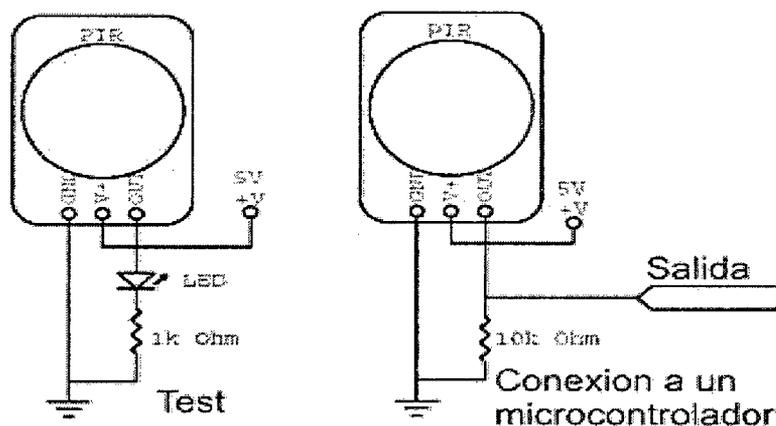


Figura N° 2.10: Conexión del sensor de movimiento S320140

2.2.3.2 Sensor fotoeléctrico LDR

Un sensor fotoeléctrico es un dispositivo electrónico que responde al cambio en la intensidad de la luz. Estos sensores requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor. Todos los diferentes modos de sensado se basan en este principio de funcionamiento. Están diseñados especialmente para la detección, clasificación y posicionado de objetos; la detección de formas, colores y diferencias de superficie, incluso bajo condiciones ambientales extremas.

Los sensores de luz se usan para detectar el nivel de luz y producir una señal de salida representativa respecto a la cantidad de luz detectada. Un sensor de luz incluye un transductor fotoeléctrico para convertir la luz a una señal eléctrica y puede incluir electrónica para condicionamiento de la señal, compensación y formateo de la señal de salida.

El sensor de luz más común es el LDR - Light Dependant Resistor o Resistor dependiente de la luz.-Un LDR es básicamente un

resistor que cambia su resistencia cuando cambia la intensidad de la luz. Existen tres tipos de sensores fotoeléctricos, los sensores por barrera de luz, reflexión sobre espejo o reflexión sobre objetos

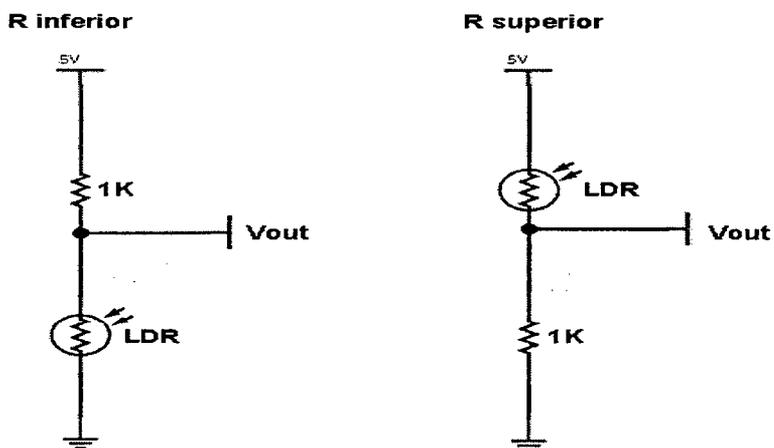


Figura N° 2.11: Ubicación del LDR.

Si utilizamos el LDR como resistencia inferior del divisor de tensión, nos dará la tensión máxima cuando tengamos el LDR en plena oscuridad, ya que estará oponiendo el máximo de su resistencia al paso de la corriente derivándose esta por V_{out} al completo, si lo utilizamos como resistencia superior, el resultado será el inverso, tendremos la tensión máxima cuando esté completamente iluminado, ya que se comportará prácticamente como un circuito abierto, con una resistencia de 50Ω o 100Ω .

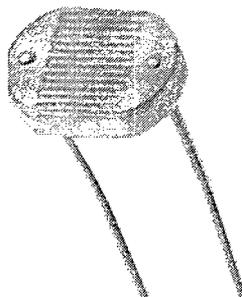


Figura N° 2.12: El LDR

2.2.4. Pantalla LCD 16X2

Una pantalla LCD se trata de un visualizador alfanumérico (solo se pueden ver números, letras y/o algún otro carácter). Este tipo de LCD se define por el número de líneas que es capaz de mostrar a la vez el número de caracteres por líneas. Un LCD de 8x1 es un LCD de 8 caracteres y una línea. Los más comunes son de 8, 16, 20, 24 y 40 caracteres y de 1, 2 y 4 líneas el LCD es actualmente el circuito más barato y confiable para mostrar datos en un proceso de monitoreo su interfaz con los controladores se realiza a través de un conector de 14 pines, cuya configuración es respetada por la mayoría de los fabricantes.⁷

El módulo LCD lleva integrado a sus circuitos una memoria ROM (U1) conocida como "generador de caracteres" que habrá de generar los patrones de la matriz de puntos (5 x 7 ó 7 x 9) que forman los caracteres en la pantalla. También tiene una RAM interna (U2) que almacena los caracteres y los exhibe en el módulo LCD. Esta LCD consta de 16 pines para su conexión (Figura N° 2.10): usa 11 líneas de comunicación (3 de control y 8 de datos), 2 de alimentación del texto y 2 para la luz de fondo.

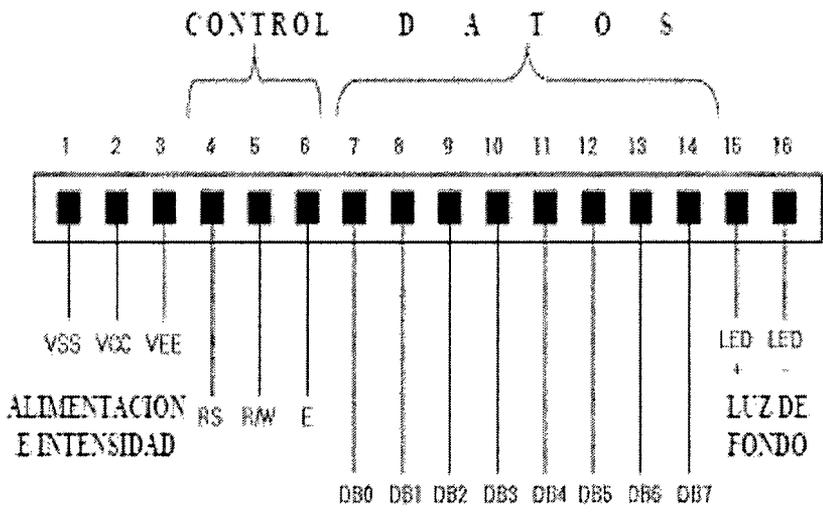


Figura N° 2.13: Conector de la pantalla LCD 16x2

⁷[Fecha de acceso: 23 de octubre del 2013]. URL disponible en: <http://tr.scribd.com/doc/94511479/Cerradura-Electronic-A-Con-PIC>

Algunas de sus especificaciones aparecen a continuación:

- Dimensiones de la pantalla: 65mm x 14mm.
- Dimensiones de la unidad: 82mm x 37mm.
- Alimentación de +5V DC.
- Soporta interfaces de 4 u 8 bits.
- Luz de fondo azul con texto en negro.
- 16 caracteres x 2 líneas.

2.2.5. Fuente de alimentación

La fuente es un dispositivo muy importante para un sistema ya que de ahí se toma la energía necesaria para la alimentación del mismo. Esta es un subsistema que convierte la corriente alterna en otro tipo de corriente eléctrica adecuado para nuestra cerradura. Este otro tipo de corriente será de tipo continuo, ya nosotros queremos que se mantenga en un solo valor, para nuestro caso se tendrá dos valores de voltaje de salida (12V y 5V) nuestro fuente consta de 4 etapas

- Atenuación: Disminuye el voltaje y la corriente de corriente alterna (voltaje de lineal (hasta este punto 12Vd.c.). y la corriente disminuye a 1A
- Rectificación: Se elimina la parte negativa de la señal de corriente alterna, esto se logra utilizando un Rectificador de Onda Completa.
- Filtrado: Se suavizan los rizados de la señal saliente del rectificador, tratando de lograr el valor de voltaje lineal.
- Regulación: Su función es mantener y controlar el valor deseado a la carga.

2.2.6. Automatización [URL5]

Hay tres clases muy amplias de automatización industrial: automatización fija, automatización programable y automatización flexible.

La **automatización fija**.- se utiliza cuando el volumen de producción es muy alto, por tanto, se puede justificar económicamente el alto costo del

diseño de equipo especializado para procesar el producto con rendimiento alto y tasas de producción elevadas. Un posible inconveniente de la automatización fija es su ciclo de vida que va de acuerdo a la vigencia del producto en el mercado.

La **automatización programable**.- se emplea cuando el volumen de producción es relativamente bajo y hay una diversidad de producción a obtener. En este caso el equipo de producción es diseñado para adaptarse a las variaciones de configuración del producto; esta adaptación se realiza por medio de un programa (Software).

La **automatización flexible**.- es más adecuada para un rango de producción medio. Estos sistemas poseen características de la automatización fija y de la automatización programada. Los sistemas flexibles suelen estar constituidos por una serie de estaciones de trabajo interconectadas entre sí por sistemas de almacenamiento y manipulación de materiales, controlados en su conjunto por una computadora.⁸

Uno de los puntos significativos cuando se habla de automatización es el hardware, la cual se puede llevar a cabo por computadores personales, microcontroladores, PLC's, FPGA's y DSP, con estos se pueden dar soluciones al desarrollo del medio de la industria y aplicaciones muy especiales en forma eficiente y segura.

- Un microcontrolador es un dispositivo electrónico en forma de chip que consta de las tres unidades básicas de un ordenador, la unidad de procesamiento central o CPU, la memoria y los puertos de entrada y salida, estos microcontroladores funcionan mediante un programa almacenado en su memoria, estos programas pueden estar escritos en distintos lenguajes de programación, los microcontroladores son ampliamente utilizados gracias a que son diseñados para disminuir el coste económico y el consumo de energía de un sistema en particular,

⁸[Fecha de acceso: 15 de julio del 2013]. URL disponible en:<http://www.gestiopolis.com/administracion-estrategia/automatizacion-en-procesos-como-mejores-practicas.htm>

además son utilizados como cerebros de una gran variedad de sistemas que controlan máquinas.

- Los controladores lógicos programables (PLC) son dispositivos electrónicos operados digitalmente utilizando una memoria interna para el almacenamiento de instrucciones o tareas con el fin de que cumpla unas funciones específicas tales como lógicas, secuenciales, tiempo y aritméticas, así controlando varios tipos de máquinas o procesos por medio de entradas y salidas lógicas o digitales.
- Campo de matriz de puertas programables (FPGA) es dispositivo semiconductor que tiene bloques de lógica cuya interconexión y funcionalidad que permite configurar en el sitio mediante un lenguaje de programación. La lógica programable de este admite reproducir funciones sencillas como puerta lógica o un sistema combinado hasta funciones de mayor complejidad como son los sistemas de chip:
- Los procesadores de señal digital (DSP) es un tipo de procesador con alta velocidad en el procesamiento de datos en tiempo real. A un DSP le llega una señal digital la procesa de manera que sea más clara, una imagen más nítida o datos con mayor velocidad. Los DSP tienen un conjunto de características básicas que hace más útil su funcionamiento, entre ellas están:
 - Alta velocidad en el cálculo aritmético.
 - Transferencia de datos hacia y desde un mundo real.
 - Memorias de múltiple acceso.

Para realizar una automatización como complemento al software se debe tener en cuenta cualquier tipo de hardware anteriormente mencionados, la selección de estos se hacen dependiendo de los requerimientos que se necesiten, escogiendo el software y hardware que presente las mejores características, menor coste y una buena funcionalidad para la industria. [4]

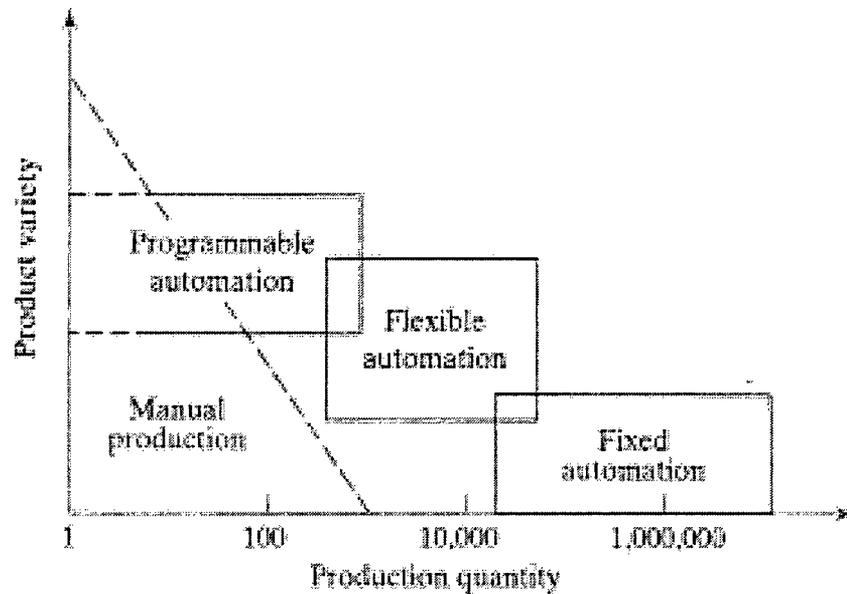


Figura N° 2.14: Resumen de tipos de automatización

2.3 Hipótesis

2.3.1. Hipótesis general

El sistema de protección influye significativamente repeliendo las aves Huaquias para la protección de alevinos en la empresa piscícola Huiñamalca Parco.

2.3.2. Hipótesis específicos

- a) El nivel de la tasa de pérdidas de alevinos, con el sistema de protección para repeler aves Huaquias, en la empresa Piscícola Huiñamalca Parco es menor a 5%.
- b) Existe una relación inversa entre el sistema de protección para repeler las aves huaquias y la tasa de pérdidas de alevinos en la empresa piscícola Huiñamalca Parco.

2.4 Variables de estudio

2.4.1. Variable independiente

Sistema de protección para repeler a las aves Huaquias.

2.4.2. Variable dependiente

Tasa de pérdidas de alevinos por ataque de aves Huaquias.

Variables	Operacionalización de las Variables	Indicador	Unidades
<u>Independiente</u> a) Sistema de protección =SP	El sistema de protección para repeler aves Huaquias.	a) Presencia = P	a) Se mide en und.
<u>Dependientes</u> a) Tasa de Perdidas de Alevinos=TPA	Indicador demográfico que señala el número de defunciones de una población por cada 1.000 habitantes	a) Porcentaje de fallecidos=PF	a) Porcentaje %

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1 Ámbito de estudio

El estudio se desarrolló en una piscigranja ubicado en la localidad de Huiñamalca, distrito Parco, provincia de Jauja, región Junín.

3.2 Tipo de investigación

De acuerdo al propósito de la investigación, naturaleza de los problemas y objetivos formulados en el trabajo, el presente estudio reúne las condiciones suficientes para ser calificado como una investigación tecnológica.

3.3 Nivel de investigación

El nivel de investigación es correlacional, ya que en el trabajo se requirió conocer la relación que existe entre las variables tasa de pérdida de alevinos y el sistema de protección para repeler aves Huaquias.[5]

3.4 Método de investigación

En la presente investigación se empleó como método general de investigación el método comparativo, como método específico el método experimental.

3.5 Diseño de investigación

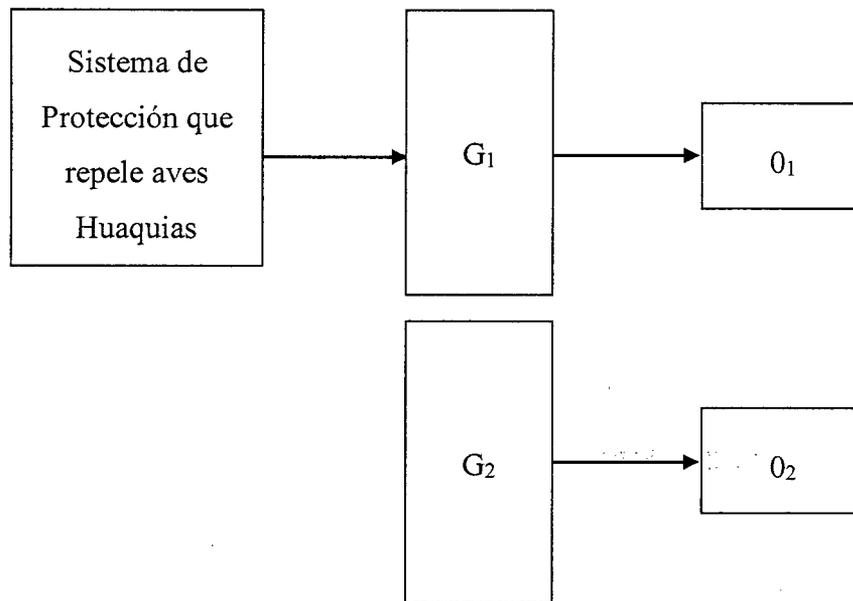


Figura N° 3.1: Diseño de investigación

Donde G1 representa al grupo de alevinos experimental, al que se aplica el sistema de protección para repeler aves Huaquias, G2 representa al grupo control al cual no se le aplica el sistema de protección, O1 es la tasa de mortalidad Post-prueba con aplicación del sistema de protección y O2 es la tasa de mortalidad Post-prueba sin aplicación del sistema de protección.[6]

3.6 Población, muestra, muestreo

3.6.1. Población

La población objetivo a la cual fue dirigida esta investigación fueron 03 piscigranjas de trucha arcoíris del distrito de Parco, provincia de Jauja, región Junín.

3.6.2. Muestra

Lo constituye 01 piscigranja de trucha arcoíris del distrito de Parco, provincia de Jauja, región Junín, la misma que cuenta con una población de 100000 alevinos.

3.6.3. Muestreo

El muestreo fue aleatorio simple porque la piscigranja fue seleccionada de tal manera que cada una de ellas tuvo igual probabilidad de ser seleccionada de la población. Específicamente el muestreo aleatorio fue sistemático porque las muestras la obtuvimos de una manera directa y ordenada.

3.7 Técnicas e instrumentos de recolección de datos

Se utilizó la técnica de observación de campo con el instrumento guía de observación de campo, en el cual registramos la cantidad de pérdidas de alevinos durante el desarrollo del trabajo de investigación como se muestra en el Anexo D.

3.8 Procedimiento de recolección de datos

3.8.1. Fase de pre-campo

Al inicio de la investigación se diseñó el instrumento guía de observación de campo.

3.8.2. Fase de campo

En la ejecución de la investigación se aplicó el instrumento de la fase pre-campo, se implementó el sistema para repeler aves Huaquias y se recolectaron datos de presencia de aves de día y de noche con el sistema de protección en funcionamiento.

3.8.3. Fase de gabinete

Al final de la ejecución del trabajo de investigación se hizo el procesamiento de la información, análisis de la información y presentación de resultados.

3.9 Técnicas de procesamiento y análisis de datos

Para procesar los datos recolectados de la tasa de mortalidad de los alevinos se ingresó al programa Excel para representar mediante gráficos estadísticos la cantidad de pérdidas desde el mes de mayo hasta el mes de noviembre.

Además para la prueba de hipótesis se hizo cálculos de media, desviación estándar, variación, grado de asimetría y finalmente se utilizó el método estadístico T-student con un nivel de confianza del 95% para nuestra variable en estudio.

CAPÍTULO IV

RESULTADOS

4.1 Presentación de resultados

El trabajo de investigación titulado "Sistema de Protección para Repeler Aves Huaquias en la Empresa Piscícola de Huiñamalca Parco", se desarrolló entre los meses de mayo a noviembre del 2013. En la investigación se trabajó con el diseño pre experimental de dos grupos (dos piscigranjas) uno control y el otro experimental.

El sistema se aplicó al grupo experimental y de manera paralela se registró los datos de mortalidad en alevinos en el grupo control; todas estas pruebas son posteriores al sistema de protección para repeler aves Huaquias.

Este sistema de protección se ha implementado en la iniciación de la presente investigación, es decir en los meses de enero a abril del 2013, con la finalidad de disminuir la tasa de mortalidad de alevines dentro de la Empresa Piscícola Huiñamalca.

4.1.1 Diagrama de bloques del sistema de protección

En el diagrama de bloques de la Figura N° 4.1 del sistema de protección, el sensor PIR S320140 entrega un voltaje de 5V cuando hay presencia y 0V caso contrario, el LDR entrega diferentes niveles de tensión (entre 0V y 5V) al microcontrolador PIC 16F877A para su procesamiento correspondiente de acuerdo al algoritmo implementado.

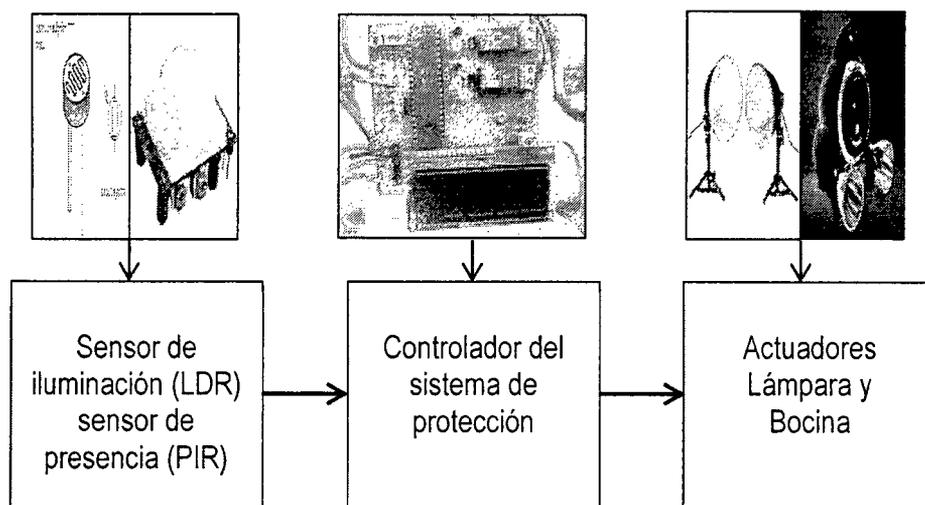


Figura N° 4.1: Diagrama de bloques del sistema de protección

4.1.2 Diagrama de flujo del sistema de protección

Para nuestro sistema de protección se tiene como dispositivo principal al microcontrolador 16F877A, además del sensor de luz LDR y el sensor PIR S320140.

En el diagrama de flujo de la Figura N° 4.2 se puede observar que primero se inicializa el microcontrolador, seguidamente se lee el sensor LDR y a continuación se presenta los siguientes casos:

PRIMERO.- Si es de noche y existe presencia de ave huaquia, entonces se enciende la lámpara.

SEGUNDO.- Si no es de noche y hay presencia de ave huaquia entonces se enciende la bocina.

TERCERO.- En ambos casos si no existe presencia de ave huaquia entonces se vuelve a leer el sensor LDR.

El programa del diagrama de flujo fue implementado en lenguaje de programación C utilizando el software de compilación CCS para microcontroladores (ver anexo A.1.).

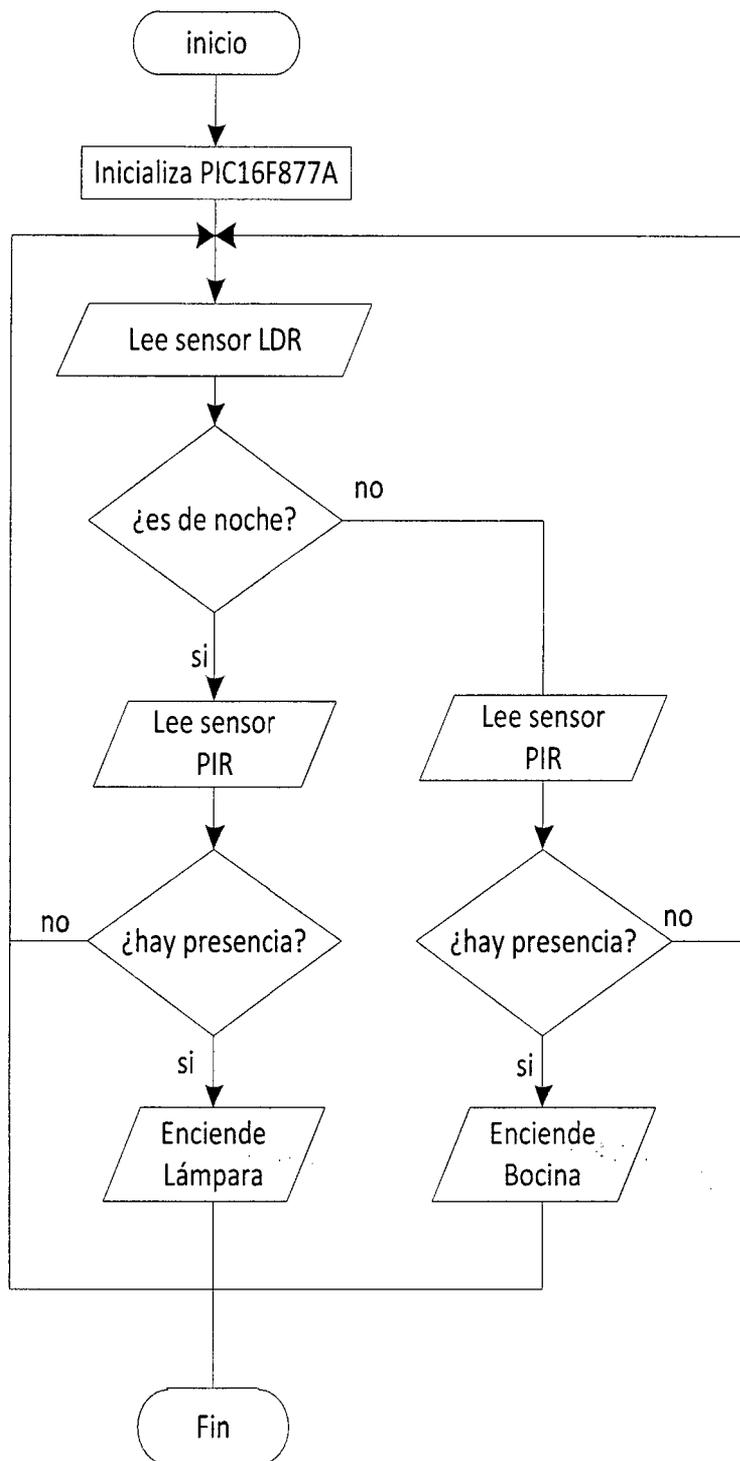


Figura N° 4.2 Diagrama de flujo del programa implementado en el PIC 16F877A

4.1.3 Diseño e implementación del sistema de protección

Sistema implementado en el programa Isis Proteus, donde se puede observar el uso de los diferentes componentes y dispositivos utilizados en la etapa de implementación como se muestra en la Figura N° 4.3.

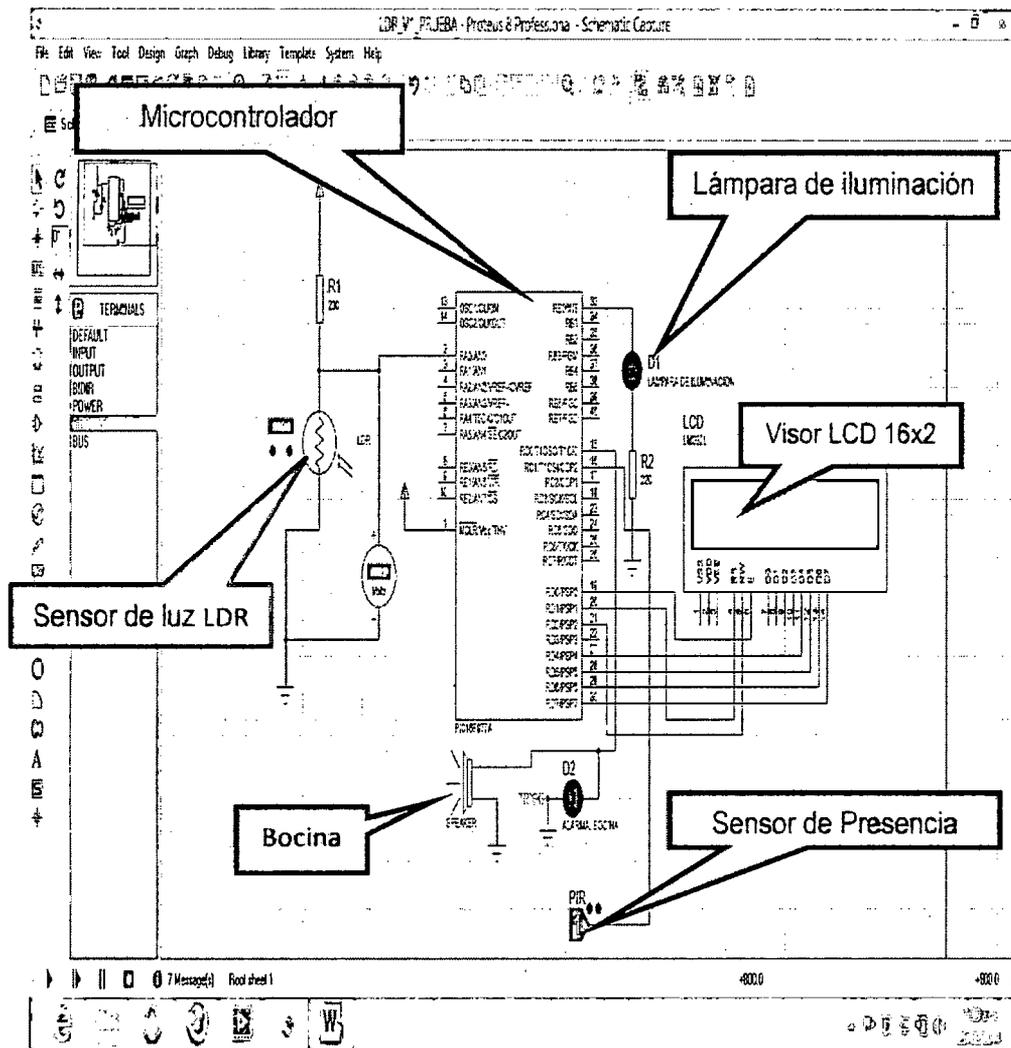


Figura N° 4.3: Elementos que forman parte del sistema de protección

Simulación del sistema en funcionamiento de noche, en el que se puede observar que la lámpara de iluminación está activa debido a que es un mecanismo para ahuyentar a las aves Huaquias, este comportamiento se puede observar en la Figura N° 4.4.

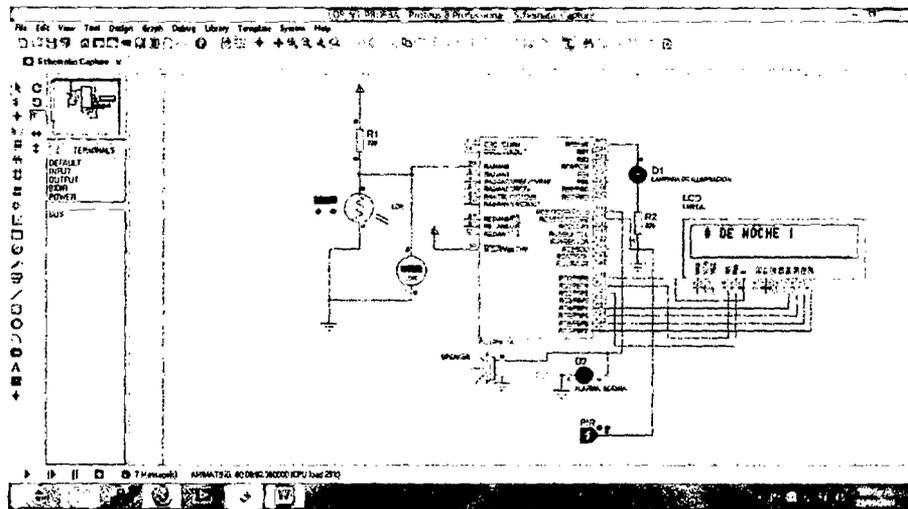


Figura N° 4.4: Simulación de activación del sistema por la noche

Simulación del sistema en funcionamiento de día, en el que se puede observar que la bocina está activa debido a que es un mecanismo para ahuyentar a las aves Huaquias, ver Figura N° 4.5.

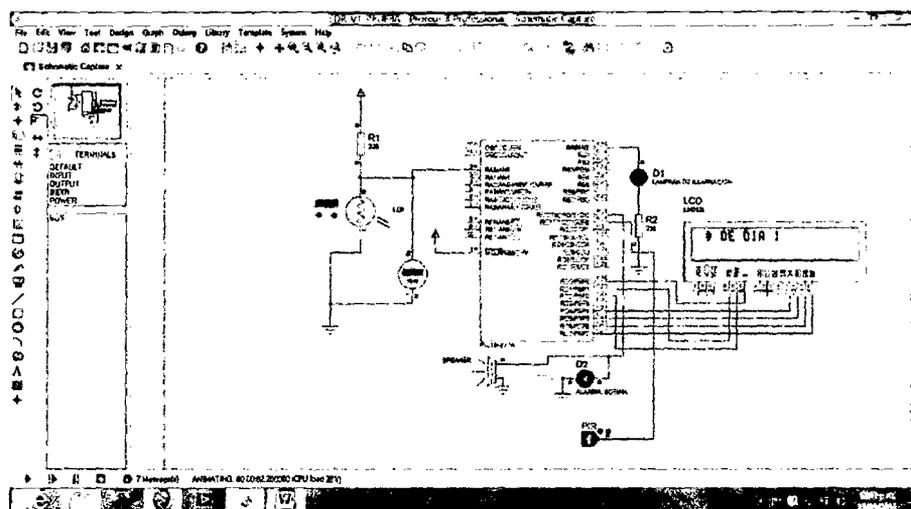


Figura N° 4.5: Simulación de activación del sistema en el día

Simulación del sistema en funcionamiento de día, en el que se puede observar que tanto la bocina y la lámpara de iluminación están desactivadas debido a que no existe presencia de las aves Huaquias, ver Figura N° 4.6.

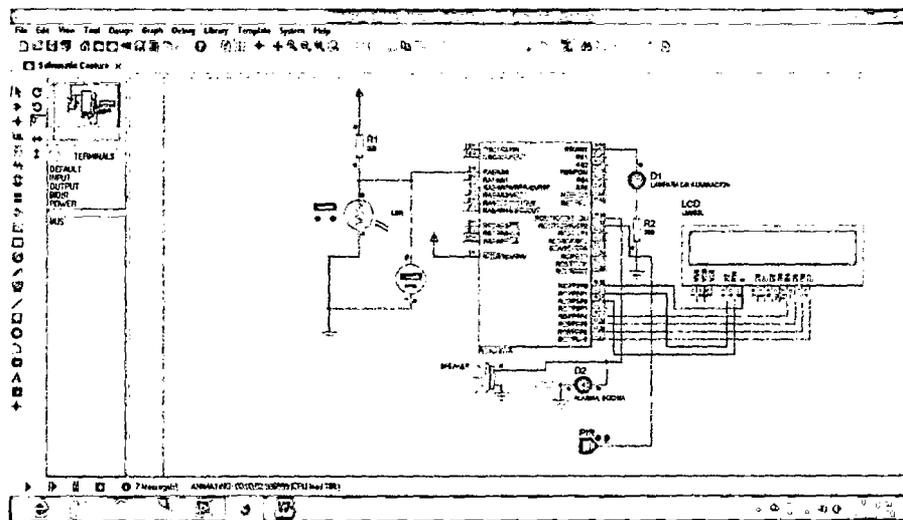


Figura N° 4.6: Simulación del estado de no activación del sistema

Circuito implementado con el microcontrolador PIC 16F877A, en el que se visualiza al oscilador de cristal de 4 MHz, ver Figura N° 4.7.

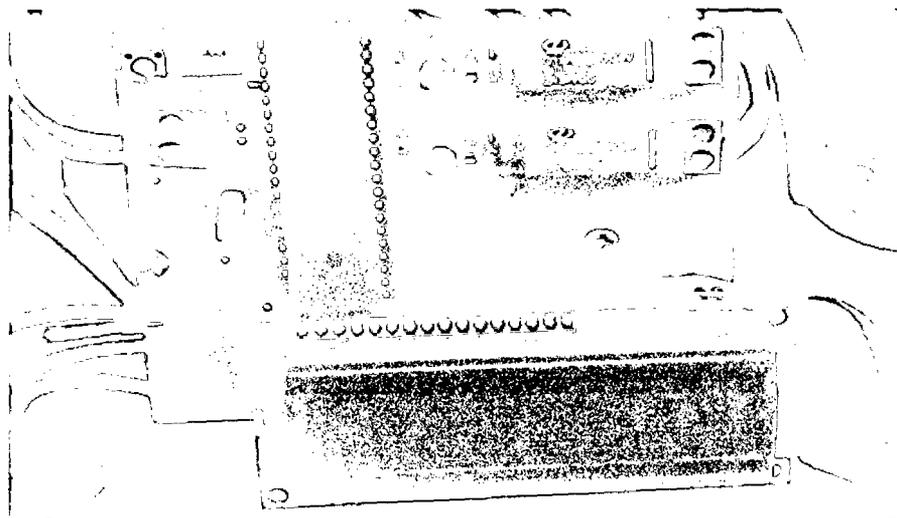


Figura N° 4.7: Circuito físico de sistema de protección implementado en placa

Maqueta implementada con los circuitos del sistema de protección completos, en el que se puede observar el microcontrolador 16F877A, el sensor PIR S320140, la bocina, la lámpara, el display, ver Figura N° 4.8.



Figura N° 4.8: Maqueta de la Piscigranja con el circuito físico implementado

4.1.4 Prueba de hipótesis

El proceso que permite realizar el contraste de hipótesis requiere ciertos procedimientos. Se ha podido verificar los planteamientos de diversos autores y cada uno de ellos con sus respectivas características y peculiaridades, motivo por el cual se decidió por la prueba de hipótesis con distribución t de Student.

Prueba de hipótesis de investigación

H_0 : El sistema de protección no influye significativamente repeliendo las aves Huaquias para la protección de alevinos en la empresa piscícola Huiñamalca Parco.

H₁ : El sistema de protección influye significativamente repeliendo las aves Huaquias para la protección de alevinos en la empresa piscícola Huiñamalca Parco.

Implica que la investigación tiene 95% de seguridad para generalizar sin equivocarse y solo 5% en contra. En términos de probabilidad, 0.95 y 0.05, respectivamente, ambos suman la unidad.

gl = (n₁-1)+(n₂-1); gl = (7-1)+(7-1); gl=12

gl : Grados de libertad

n₁, n₂: Número de filas de la tabla de registro

Buscando a 5% de significancia y 12 grados de libertad en la prueba t de Student se tiene que el punto crítico o "t" teórica es igual a 1.782; por lo tanto:

Valor crítico = **1.782**

Aceptar Ho si $-1.782 < t_c < 1.782$

Rechazar Ho si $t_c \leq -1.782$ o $1.782 \leq t_c$

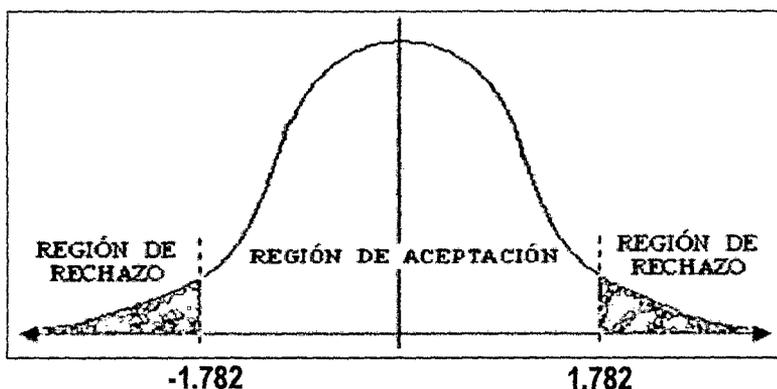


Figura N° 4.9: Prueba de t-Student

El estadígrafo de prueba más apropiado para este caso es la Prueba t de Student, como el diseño es experimental, se tomó dos grupos de investigación uno control y el otro experimental, los datos que se presenta en la tabla 4.1 sobre pérdidas de alevinos en el periodo de 7

meses fueron recogidos por la guía de observación de campo del Anexo D.

Tabla 4.1: Pérdidas de alevinos en periodo de 7 meses

Meses de la aplicación	Pérdida de alevinos en el grupo control (Kg)	Pérdida de alevinos en grupo experimental (Kg)
Mayo	10	0.00
Junio	12	0.00
Julio	15	0.10
Agosto	12	0.15
Setiembre	18	0.20
Octubre	15	0.10
Noviembre	18	0.15
Total	100	0.7

Fuente: Elaboración propia

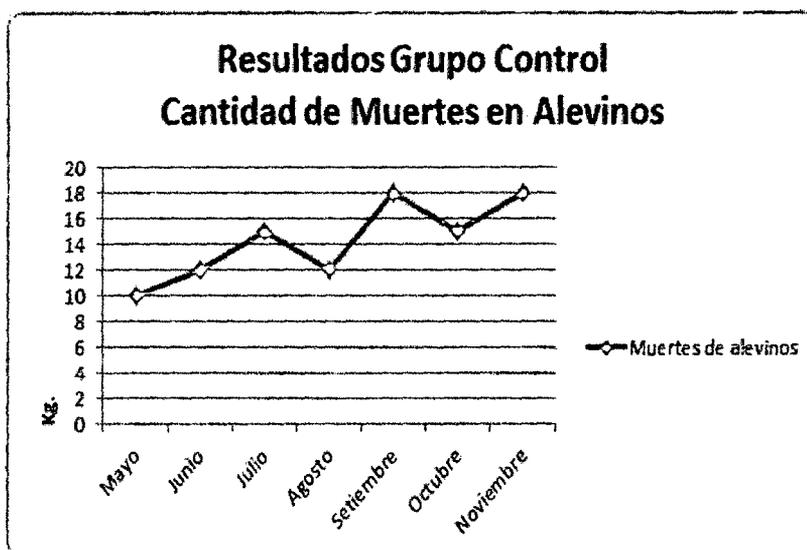


Figura N° 4.10: Resultados de grupo control

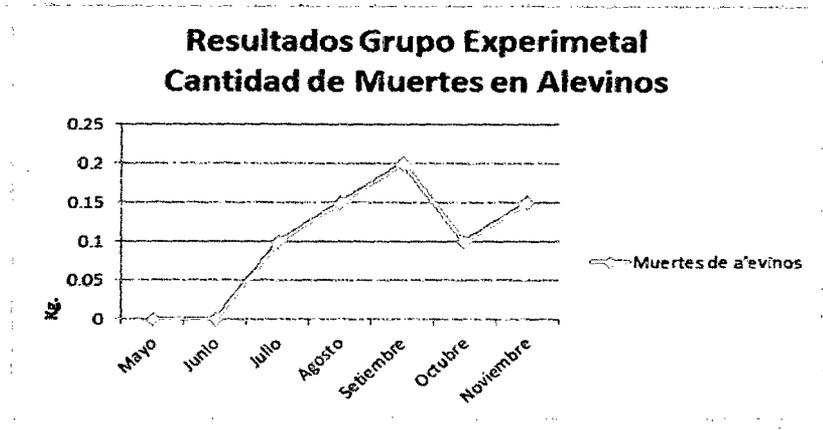


Figura N° 4.11: Resultados de grupo experimental



Figura N° 4.12: Resumen estadístico

Calculemos ahora la prueba de hipótesis con la "t" de Student.

Para el cálculo de la media, desviación estándar, variación, grado de asimetría y la prueba estadística de T-student se utilizaron las fórmulas estadísticas del programa Excel.

Tabla 4.2: Prueba de hipótesis con la "t" de Student.

Media	14.286	0.100
Desviación	3.094	0.076
Variación	9.571	0.006
Asimetría	0.005	-0.393

Fuente: Elaboración propia

Para hallar el valor de t de Student:

$$t = \frac{\overline{X(1)} - \overline{X(2)}}{\sqrt{\left(\frac{S^2(1)}{n(1)-1} + \frac{S^2(2)}{n(2)-1}\right)}}$$

Donde:

$S^2(1)$ y $S^2(2)$ son las varianzas de las muestras 1 y 2

$n(1)$ y $n(2)$ son los tamaños de las muestras 1 y 2

$X(1)$ y $X(2)$ son las respectivas medias de la muestra 1 y la muestra 2

Ahora podremos valernos de esta fórmula, reemplazando valores:

$$t = \frac{14.286 - 0.1}{\sqrt{\left(\frac{9.571}{7-1} + \frac{0.006}{7-1}\right)}}$$

$$t = 11.228$$

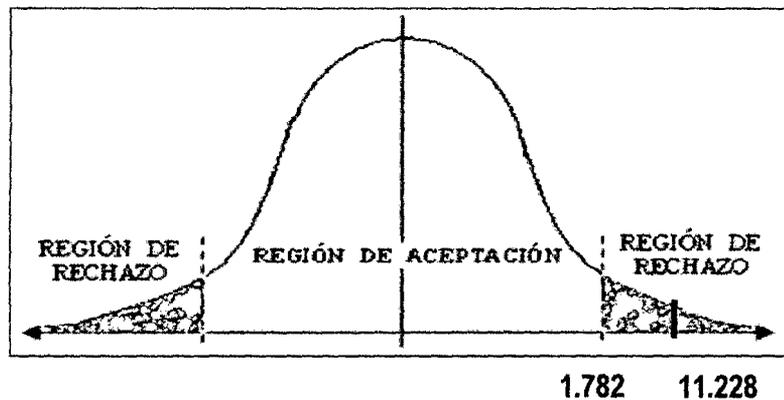


Figura N° 4.13: Ubicación de t_c

Decisión Estadística:

Puesto que t calculada es mayor que la t teórica; es decir ($11.228 > 1.782$), en consecuencia se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alterna (H_1).

Conclusión Estadística:

Se ha demostrado con un nivel de confianza del 95% y la prueba t de Student que la implementación del sistema de protección para repeler

aves Huaquias en la empresa piscícola de Huiñamalca Parco influye significativamente en la reducción de la pérdida de alevinos de la Empresa piscícola Huiñamalca Parco.

a) Prueba de hipótesis específica 1

H_{E1} : El nivel de la tasa de pérdidas de alevinos, con el sistema de protección para repeler aves Huaquias, en la empresa Piscícola Huiñamalca Parco es menor a 5%.

Para probar la hipótesis hacemos el cálculo de la tasa de pérdidas con los grupos experimental y control, de la Tabla N° 4.1 se obtiene los siguientes resultados:

Tasa de pérdida con el sistema de protección = $(0.7 \cdot 100\%) / 100 = 0.7\%$.

Decisión Estadística:

Puesto que $0.7\% < 5\%$, se verifica la validez de la hipótesis específica 1.

b) Prueba de hipótesis específica 2

H_{E2} : Existe una relación inversa entre el sistema de protección para repeler las aves huaquias y la tasa de pérdidas de alevinos en la empresa piscícola Huiñamalca Parco.

Como puede observarse en la Tabla N° 4.1 sin sistema de protección tenemos 100 Kg de pérdidas durante los meses de mayo a noviembre y con el sistema de protección tenemos 0.7 Kg de pérdidas, por lo que se puede concluir que la relación entre las variables sistema de protección y la tasa de pérdidas es inversa.

Decisión Estadística:

Puesto que con la aplicación del sistema de protección la tasa de pérdidas es de 0.7 Kg, se verifica la validez de la hipótesis específica 2.

4.2 Discusión

La variable independiente presencia de aves Huaquias en el trabajo de investigación, es una variable discreta que inicialmente se establece con el valor de 0, significando esto ausencia de presencia.

Cuando el PIR detecta la presencia de un organismo vivo, el sistema lo interpreta como presencia y dependiendo de la hora enciende la bocina si es de día o la lámpara si es de noche.

Se pudo observar que el sensor no detecta ninguna presencia cuando el cuerpo que ingresa a la zona de cobertura no es un organismo vivo, realzando que el funcionamiento de este PIR también detecta la presencia de calor, ya que es un tipo de sensor infrarrojo.

CONCLUSIONES

1. El nivel de la tasa de pérdidas de alevinos, con el sistema de protección para repeler aves Huaquias en la Empresa piscícola Huiñamalca Parco fue de 0% en los meses de mayo y junio, alcanzándose un porcentaje de 0.08% en el mes de setiembre, ya que se pudo observar una mayor presencia de estas aves en este mes, haciendo un total de 0.28% de pérdidas entre los meses de mayo a noviembre, representando este porcentaje a 280 alevinos de un total de 100000.
2. La relación que existe entre el sistema de protección para repeler las aves Huaquias con la tasa de pérdidas de alevinos en la Empresa Piscícola Huiñamalca Parco, es inversa como se muestra en la tabla de registro entre los meses de mayo a noviembre del 2013.
3. En la investigación se ha demostrado con un nivel de confianza del 95% y la prueba t de Student que la implementación del sistema de protección para repeler aves Huaquias ha influido significativamente en la disminución de la pérdida de alevinos de la empresa Huiñamalca Parco, Región Junín en el 2013; ya que se ha disminuido ostensiblemente las pérdidas tal como se pudo registrar durante los seis meses de aplicación del sistema.

RECOMENDACIONES

1. Se recomienda implementar una base de datos con sonidos diferentes, para adicionar al sistema, con el objetivo de evitar el adiestramiento condicionado de las aves.
2. Para asegurar el funcionamiento del sistema de protección contra el ataque de las aves Huaquias se recomienda la implementación de un sistema de alimentación ininterrumpida, que asegurará en un 100% el funcionamiento del sistema ante la falta de energía eléctrica en esta zona.
3. Replicar la presente investigación en otras empresas piscícolas de la región y del país; por el mismo responsable o por otros investigadores; y así conseguir una mayor confiabilidad de sus resultados y conclusiones.
4. Es muy importante controlar la calibración del nivel de tensión del sensor de presencia en los estanques de los alevines para un mejor control.
5. La aplicación de este tipo de sistemas para la protección contra los ataques de aves predatoras es escasa por lo que se debe implementar para mejorar la producción de alevines ya esta etapa es crítica en el desarrollo biológico de trucha.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Penagos J., Como Programar en Lenguaje C los Microcontroladores PIC16F88, 16F628A y 16F877A, 1a ed. Ecuador. Mc. Graw Hill Editorial; 2010.
 - [2] Decroyé S. Diseño de un Sistema RF para Lecturas de Sensores en Aplicaciones Docentes de Proyectos Básicos de Ingeniería, España, junio 2012.
 - [3] Penagos J. Instrucciones para programar. 1a ed. Ecuador. Mc. Graw Hill Editorial; 2010.
 - [4] Ogata K, Ingeniería de Control Moderna. 3a ed. Japón: Mc. Graw Hill Editorial; 2003
 - [5] Hernández R., Fernández C., Baptista M. Metodología de la Investigación 5a ed. México: Mc. Graw Hill; 2010.
 - [6] Caballero, A. Metodología Integral Innovadora para Planes y Tesis. 1a ed. Lima: Instituto Metodológico Alen Caro E.I.R.L. Editorial; 2011.
- [URL1] Proyecto peruanos, Trucha arco iris, Lima Perú 2001 [Fecha de acceso: 15 de mayo de 2013]. URL disponible en: <http://www.proyectosperuanos.com/truchas.html>.
- [URL2] Ceped. Municipalidad de Ragash. Antamina, Manual de Crianza Trucha (Oncorhynchus mykiss), Ragash Perú, 2009 [Fecha de acceso: 2 de junio de 2013]. URL disponible en: http://www.cedepperu.org/img_upload/c55e8774db1993203b76a6afddc995dc/manual_truchas_antamina.pdf
- [URL3] López G, Margni S., Funcionamiento de microcontroladores, Universidad de la República Oriental del Uruguay, Uruguay, 2003 [Fecha de acceso: 23 de junio del 2013]. URL disponible en: <http://www.google.com.pe/url?sa=t&rct=j&q=%22funcionamiento%20de%20microcontroladores%22%20filetype%3Adoc&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Fwww.fing.edu.uy%2FInco%2Fgrupos%2Fmina%2FpGrado%2Fconstruccion2003%2FDocumentos%2FInroduccionPics.doc&ei=FS7mT42tGoi8gTArrmvAQ&usg=AFQjCNEf8CJfE4YWYUvQO1usgGwoEZvriA&cad=rja>.
- [URL4] Sensores y detectores para robótica, 2002 – 2014 INTPLUS, [Fecha de acceso: 27 de junio del 2013]. URL disponible en: <http://www.superrobotica.com/S320140.htm>

[URL5] Almazán B, Cotilla I, Osuna S, Valdés K, Automatización, México, 2008
[Fecha de acceso: 23 de junio del 2013]. URL disponible en:
[http://www.gestiopolis.com/administracion-estrategiaautomatizacion-en
procesos-como-mejores-practicas.htm](http://www.gestiopolis.com/administracion-estrategiaautomatizacion-en-procesos-como-mejores-practicas.htm).

ARTÍCULO CIENTÍFICO

**SISTEMA DE PROTECCIÓN PARA REPELER AVES HUAQUIAS EN LA EMPRESA
PISCÍCOLA DE HUIÑAMALCA PARCO
PROTECTION SYSTEM FOR HUAQUIAS BIRDS REPEL IN FISH COMPANY OF
HUIÑAMALCA PARCO**

Alexander Ordoñez Castillo, Ángeles Gregorio Peñaloza Contreras
Universidad Nacional de Huancavelica, Facultad de Ingeniería Electrónica – Sistemas,
Escuela Académico Profesional de Electrónica

RESUMEN

Objetivos: Determinar la influencia del diseño e implementación del sistema de protección para repeler las aves Huaquias en la empresa Piscícola Huiñamalca Parco. Métodos: Es una investigación tecnológica, con nivel correlacional. Se empleó como método general de investigación el método comparativo, como método específico el método experimental y como método particular el que utiliza el control y automatización. Se manejó las variables sistema de protección y tasa de pérdidas de alevinos y se trabajó con dos grupos: uno experimental y otro de control [1]. Resultados: Se pudo observar que las aves Huaquias tienen una mejor respuesta esperada ante el sistema durante día, esto se explica porque existe un personal que ronda permanentemente la piscigranja. Conclusión: Se ha demostrado con un nivel de confianza del 95% que la implementación del sistema de protección para repeler aves Huaquias ha influido significativamente en la disminución de la pérdida de alevinos de la empresa Huiñamalca Parco [2].

Palabras claves:

Sistema de protección, tasa de pérdida alevinos, repeler aves Huaquias, empresa piscícola.

ABSTRACT

Objectives: To determine the influence of the implementation and design of protection system to repel the Huaquias birds at Huiñamalca Parco trout farm enterprise. Methods: It is a technological research, with correlational level. As a general method of research it was used the comparative method, as a specific method the experimental method and as a particular method the one that uses control and automation. Variables like the protection

system and the fry lost rate were managed and two groups: an experimental group and a control group were worked [1]. Results: It was looked that Huaquias birds have a better expected response to the system during the day, it is explained because exist a person who rounds permanently the trout farm. Conclusion: It has been demonstrated with a 95% of confidence level that the implementation of the protection system to repel Huaquias birds has influenced significantly on the diminishing of the fry lost at Huiñamalca Parco enterprise [2].

Keywords:

Protection system, loss rate of fingerlings, Huaquias repel birds, fish company.

INTRODUCCIÓN

La implementación del Sistema de Protección para Repeler Aves Huaquias, la necesitan muchas empresas piscícolas que quieren mejorar su sistema de producción de alevinos.

El sistema de protección no exige requisitos para su aplicación, ha sido elaborada para que las apliquen a las empresas piscícolas, sin importar su origen geográfico, social o cultural, la extensión de la aplicación de los factores que consideren la política de la empresa, la naturaleza las actividades y las condiciones que se encuentren.

Para empresas que logren implementar el sistema de protección obtienen beneficios que podemos mencionar:

Imagen: Las empresas que implementa el sistema de protección se ven beneficiadas en su imagen interna y externa.

Negociación: Al adoptar el sistema de protección las empresas tienen mayor poder de negociación, debido a que sus riesgos estarán identificados y controlados por procedimientos claramente identificados.

Competitividad: El hecho de asumir el sistema de protección, hará que las empresas puedan competir de igual a igual en los mercados nacionales e internacionales, Actualmente la globalización elimina las fronteras y las barreras de los diferentes productos y servicios que se ofrecen en los mercados mundiales.

MATERIALES Y MÉTODOS

En la investigación se empleó el microcontrolador PIC16F877A, cuyo código de programación se desarrolló en el lenguaje de microcontroladores CCS, [3] también se emplea como sensor de presencia el PIR S320140, también el sensor LDR (Light Dependent Resistor) para detectar el día y la noche. La población en la que se desarrolló este trabajo para validar la operación del sistema de protección diseñado en el distrito de Huiñamalca-Parco, está conformado por los 100,000 alevinos, los datos observados sobre tasa e pérdidas de alevinos se registraron en la guía de observación de campo. Por otro lado la población del grupo control está constituido por 80000 alevinos de otra piscigranja en el mismo distrito. El método que se empleó fue el cuasiexperimental, complementado con las técnicas estadísticas que permitieron realizar el tratamiento de los datos y la comprobación de la hipótesis, empleando la media, desviación, variación, asimetría y la T-Student. [4]

RESULTADOS

- En el diagrama de bloques de la Figura N° 1 del sistema de protección, el sensor PIR S320140 entrega un voltaje de 5V cuando hay presencia y 0V caso contrario, el LDR entrega diferentes niveles de tensión (entre 0V y 5V) al microcontrolador PIC 16F877A para su procesamiento correspondiente de acuerdo al algoritmo implementado.[5]

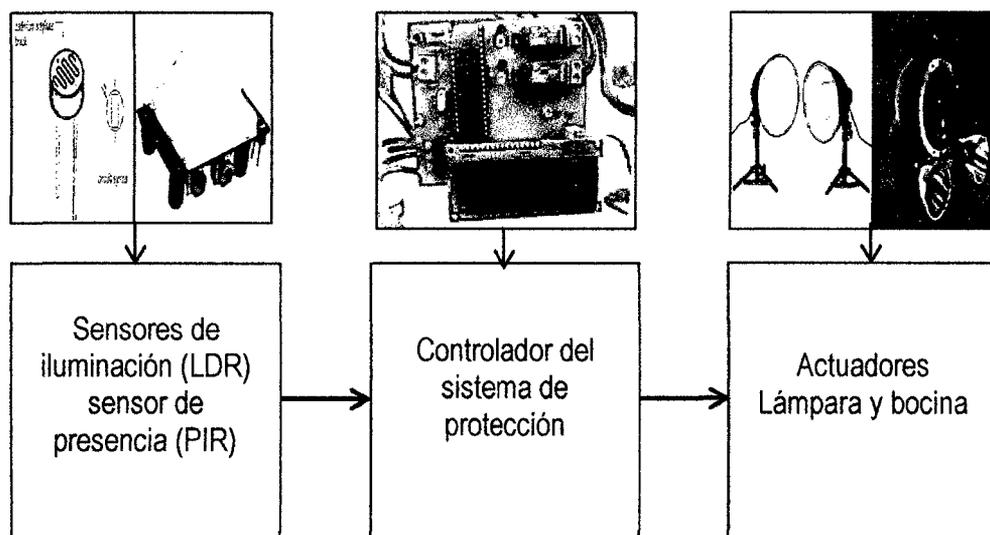


Figura N°1: Diagrama de bloques del sistema de protección

- El circuito del sistema de protección que permite repeler aves huaquias, se indica en la figura 2. donde se puede observar el uso de los diferentes componentes y dispositivos utilizados en la etapa de implementación. El modulo sensorial tiene como elemento principal el microcontrolador PIC 16F877A.[6]

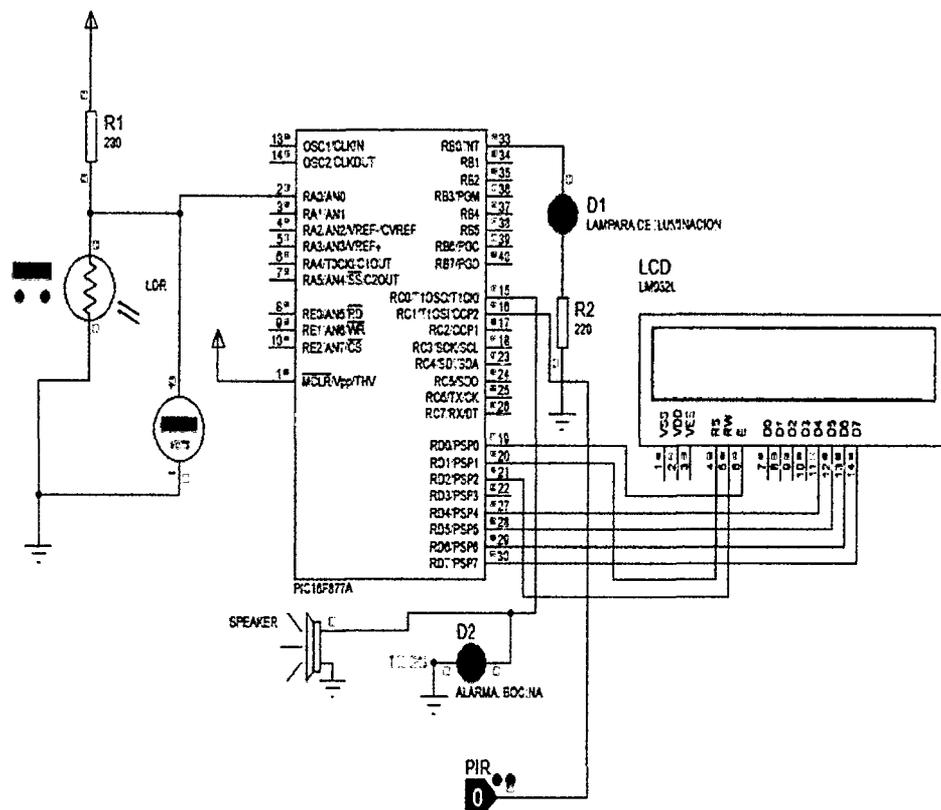


Figura 2. Circuito del sistema de protección para repeler aves huaquias.

- En el diagrama de flujo de la figura 3, se tiene el algoritmo que se implementó en el microcontrolador PIC16F877A.[7]

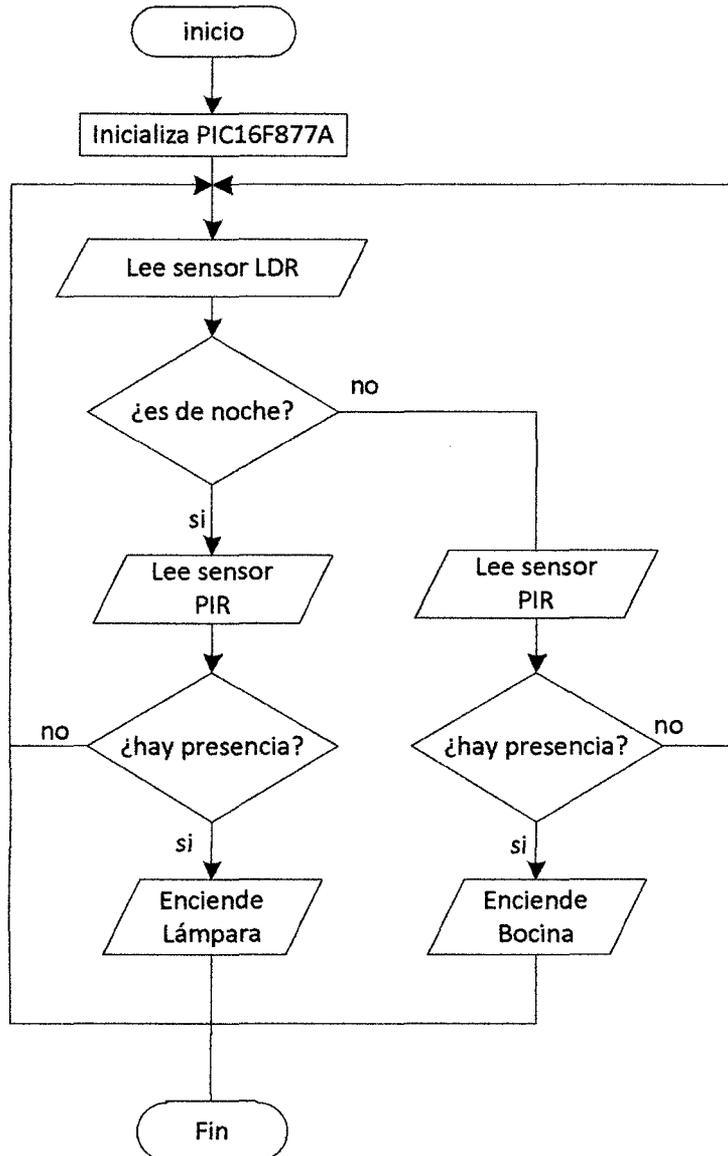


Figura 3. Diagrama de flujo del algoritmo implementado en el PIC 16F877A del sistema de protección para repeler aves huaquias.

- Simulación del sistema en funcionamiento de noche, en la figura N° 4, se puede observar que la lámpara de iluminación está activa debido a que es un mecanismo para ahuyentar a las aves Huaquias.[8]

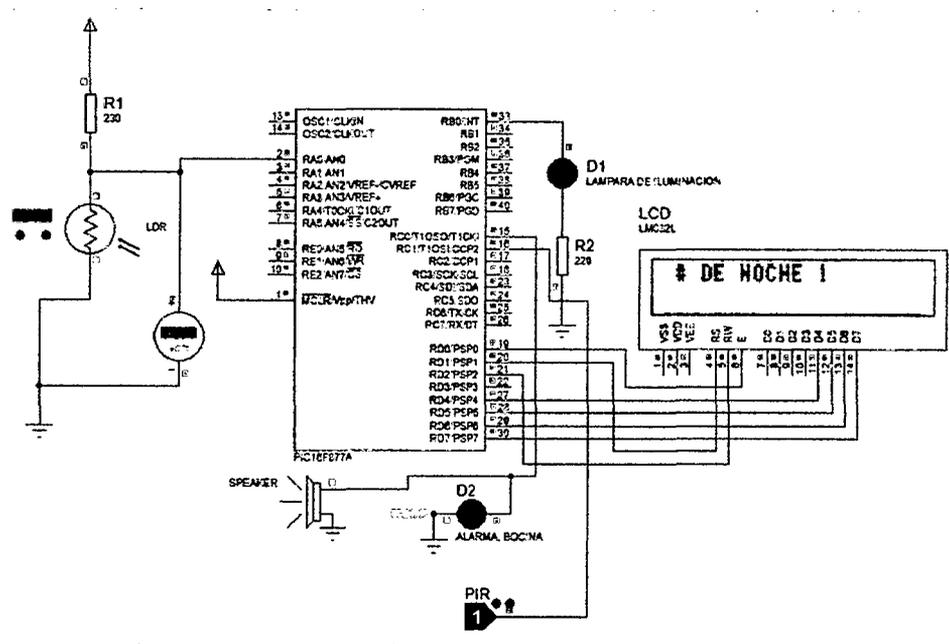


Figura N° 4. Simulación de activación del sistema por la noche

- Simulación del sistema en funcionamiento de día, en la figura 5 se puede observar que la bocina está activa debido a que es un mecanismo para ahuyentar a las aves Huaquias.

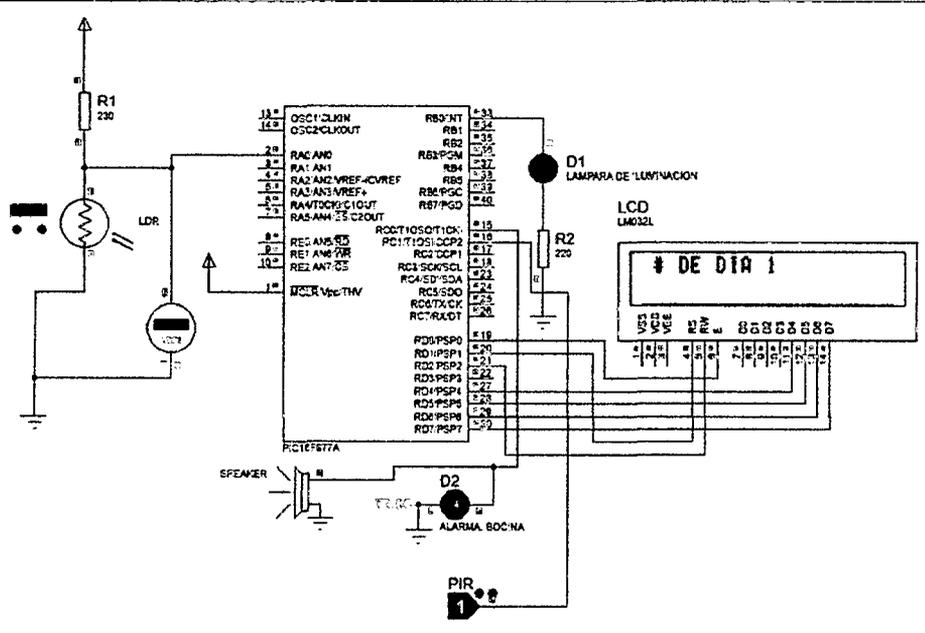


Figura 5. Simulación de activación del sistema en el día

DISCUSIÓN

La variable independiente presencia de aves Huaquias en el trabajo de investigación, es una variable discreta que inicialmente se establece con el valor de 0, significando esto ausencia de presencia.

Cuando el PIR detecta la presencia de un organismo vivo, el sistema lo interpreta como presencia y dependiendo de la hora enciende la bocina si es de día o la lámpara si es de noche.

Se pudo observar que el sensor no detecta ninguna presencia cuando el cuerpo que ingresa a la zona de cobertura no es un organismo vivo, realizando que el funcionamiento de este PIR también detecta la presencia calor, ya que es un tipo de sensor infrarrojo.

CONCLUSIONES

1. El nivel de la tasa de pérdidas de alevinos, con el sistema de protección para repeler aves Huaquias en la Empresa piscícola Huiñamalca Parco fue de 0% en los meses de mayo y junio, alcanzándose un porcentaje de 0.08% en el mes de setiembre, ya que se pudo observar una mayor presencia de estas aves en este mes, haciendo un total de 0.28% de pérdidas entre los meses de mayo a noviembre, representando este porcentaje a 280 alevinos de un total de 100000.
2. La relación que existe entre el sistema de protección para repeler las aves Huaquias con la tasa de pérdidas de alevinos en la Empresa Piscícola Huiñamalca Parco, es inversa como se muestra en la tabla de registro entre los meses de mayo a noviembre del 2013.
3. En la investigación se ha demostrado con un nivel de confianza del 95% y la prueba t de Student que la implementación del sistema de protección para repeler aves Huaquias ha influido significativamente en la disminución de la pérdida de alevinos de la empresa Huiñamalca Parco, Región Junín en el 2013; ya que se ha disminuido ostensiblemente las pérdidas tal como se pudo registrar durante los seis meses de aplicación del sistema.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Caballero, A. Metodología Integral Innovadora para Planes y Tesis. 1a ed. Lima: Instituto Metodológico Alen Caro E.I.R.L. Editorial; 2011.
- [2] Proyecto peruanos, Trucha arco iris, Lima Perú 2001 [Fecha de acceso: 15 de mayo de 2013]. URL disponible en: <http://www.proyectosperuanos.com/truchas.html>.
- [3] Penagos J. Instrucciones para programar. 1a ed. Ecuador. Mc. Graw Hill Editorial; 2010.
- [4] Hernández R., Fernández C., Baptista M. Metodología de la Investigación 5a ed. México: Mc. Graw Hill; 2010.
- [5] Ogata K, Ingeniería de Control Moderna. 3a ed. Japón: Mc. Graw Hill Editorial; 2003
- [6] López G, Margni S., Funcionamiento de Microcontroladores, Universidad de la República Oriental del Uruguay, Uruguay, 2003 [Fecha de acceso: 23 de junio del 2013]. URL disponible en: <http://www.google.com.pe/url?sa=t&rct=j&q=%22funcionamiento%20de%20microcontroladores%22%20filetype%3Adoc&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Fwww.fing.edu.uy%2Ffinco%2Fgrupos%2Fmina%2FpGrado%2Fconstruccion2003%2FDocumentos%2FIntroduccionPics.doc&ei=FS7mT42tGoi8gTArrmvAQ&usg=AFQjCNEf8CJfE4YWyUvQO1usgGwoEZvriA&cad=rja>.
- [7] Penagos J., Como Programar en Lenguaje C los Microcontroladores PIC16F88, 16F628A y 16F877A, 1a ed. Ecuador. Mc. Graw Hill Editorial; 2010.
- [8] García, E. Compilador C CCS y Simulador Proteus para Microcontroladores PIC. 1ª ed. México: Alfaomega Marcombo Ediciones Técnicas. Editorial; 2008.

ANEXOS

ANEXO A.1

CÓDIGO DEL SISTEMA DE PROTECCIÓN DESARROLLADO EN C++ DE CCS

```
#include <16F877A.h>
#device adc=10 // Conversor Analógico Digital de 10 bit el PIC 16F876A puede trabajar
con 8 o 10 bit de resolución.
#FUSES XT,NOWDT
#use delay(clock=4000000)
#use rs232(uart1, baud=9600)//usart1 -->ajuste de XMIT y RCV para la USART 1
#USE fast_IO (B)
#define trig pin_c0
#define pir input(pin_c1)
#define espera 300 // Constante para delay espera
#define pinLed PIN_B0 // Sirve para activar un LED, RELE....
#define bocina PIN_D2 // Sirve para ACTIVAR BOCINA, SI HAY PRECENCIA DE AVES
#define CONTROL 800 // Constante para ajustar según la iluminación de la habitación
#include "lcd.c"
// con iluminación de la lampara Vout medido es de 4voltios (+/-)
// 4V corresponden con binSensor = 818
// Programa principal
void main()
{
// Variables locales
int16 binSensor; // Valor de la conversión del ADC
float volt; // Tensión de entrada en AN0
int16 min=1024;
int16 max=0;
int a=0;
int b=0,c;
setup_adc_ports(AN0); // Configuramos RA0 como pin Analógico
setup_adc(ADC_CLOCK_INTERNAL); // Reloj R-C
```

```

setup_psp(PSP_DISABLED);
setup_spi(SPI_SS_DISABLED);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);
SET_TRIS_B(0x00);    // PORTB como salida
output_low(pinLed); // LED, Lampara...en el instante de encendido o reset
                    // siempre apagado
set_adc_channel(0); // AN0
delay_ms(150);     // Tras el encendido del sistema o Reset
//lcd_init(); // inicializa lcd
// Bucle infinito
lcd_init();        //inicia LCD
while(TRUE){
    // Leemos la conversión y obtenemos la tensión Vout
    delay_us(20);
    binSensor = read_adc();
    volt = (5.0/1023.0) *(binSensor);

    // Obtención de mín y max del ADC
    if ( binSensor < min){
        min = binSensor;
    }
    if (binSensor > max){
        max = binSensor;
    }
    // Encendemos "Lampara" cuando no hay luz
    if (binSensor > CONTROL){ // Si estamos en oscuridad entonces

```

```

if (pir==1)
{
output_low(trig);
output_high(pinLed); // enciende LED, Lampara...
//delay_ms(2000);
a++;
delay_ms(1500);
//printf(" PRECENCIA DE AVES, DE NOCHE \n\r");
printf(lcd_putc, "\f # DE NOCHE %d ", a); //Se muestra en pantalla
información
delay_ms (1500);
//delay_ms(espera);
lcd_putc("\f"); //Limpia pantalla
}
else
{
output_low(pinLed);
}
}
else {
// Si no mantén apagado lámpara.
if (pir==1)
{
output_low(pinLed);
output_high(trig); // enciende alarma...presencia de aves
//delay_ms(2000);
b++;
delay_ms(1500);
//printf(" PRECENCIA DE AVES, DE DIA,\n\r");
printf(lcd_putc, "\f # DE DIA %d ", b); //Se muestra en pantalla información
delay_ms (1500);
}
}

```

```
        //delay_ms(espera);
        lcd_putc("\f");    //Limpia pantalla
    }
    else
    {
        output_low(trig);
    }

}

// Escribimos la información en el TX
//printf("ADC:%4ld max:%4lu min:%4lu Volt:%4.1f\n\r",binSensor,max,min,volt);
//delay_ms(espera);
}
}
```

ANEXO A.2

CÓDIGO DE LA FUNCIÓN DEL LCD

Programa LCD.c

```

struct lcd_pin_map {          // This structure is overlaid
    BOOLEAN enable;          // on to an I/O port to gain
    BOOLEAN rs;              // access to the LCD pins.
    BOOLEAN rw;              // The bits are allocated from
    BOOLEAN unused;          // low order up. ENABLE will
    int  data : 4;           // be pin B0.
} lcd;

#if defined use_portb_lcd
    ##locate lcd = getenv("sfr:PORTB") // This puts the entire structure over the port
    #ifdef __pch__
        #locate lcd = 0xf81
    #else
        #locate lcd = 6
    #endif
    #define set_tris_lcd(x) set_tris_b(x)
#else
    ##locate lcd = getenv("sfr:PORTD") // This puts the entire structure over the port
    #ifdef __pch__
        #locate lcd = 0xf83
    #else
        #locate lcd = 8
    #endif
    #define set_tris_lcd(x) set_tris_d(x)
#endif

#ifndef lcd_type
#define lcd_type 2           // 0=5x7, 1=5x10, 2=2 lines
#endif

```

```

#define lcd_line_two 0x40 // LCD RAM address for the second line
BYTE const LCD_INIT_STRING[4] = {0x20 | (lcd_type << 2), 0xc, 1, 6};
    // These bytes need to be sent to the LCD
    // to start it up.
    // The following are used for setting
    // the I/O port direction register.

struct lcd_pin_map const LCD_WRITE = {0,0,0,0,0}; // For write mode all pins are
out
struct lcd_pin_map const LCD_READ = {0,0,0,0,15}; // For read mode data pins are
in
BYTE lcd_read_byte() {
    BYTE low,high;
    set_tris_lcd(LCD_READ);
    lcd.rw = 1;
    delay_cycles(1);
    lcd.enable = 1;
    delay_cycles(1);
    high = lcd.data;
    lcd.enable = 0;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(1);
    low = lcd.data;
    lcd.enable = 0;
    set_tris_lcd(LCD_WRITE);
    return( (high<<4) | low);
}

void lcd_send_nibble( BYTE n ) {
    lcd.data = n;
    delay_cycles(1);
    lcd.enable = 1;

```

```

    delay_us(2);
    lcd.enable = 0;
}

void lcd_send_byte( BYTE address, BYTE n ) {
    lcd.rs = 0;
    while ( bit_test(lcd_read_byte(),7) );
    lcd.rs = address;
    delay_cycles(1);
    lcd.nw = 0;
    delay_cycles(1);
    lcd.enable = 0;
    lcd_send_nibble(n >> 4);
    lcd_send_nibble(n & 0xf);
}

void lcd_init() {
    BYTE i;
    set_tris_lcd(LCD_WRITE);
    lcd.rs = 0;
    lcd.nw = 0;
    lcd.enable = 0;
    delay_ms(15);
    for(i=1;i<=3;++i) {
        lcd_send_nibble(3);
        delay_ms(5);
    }
    lcd_send_nibble(2);
    for(i=0;i<=3;++i)
        lcd_send_byte(0,LCD_INIT_STRING[i]);
}

void lcd_gotoxy( BYTE x, BYTE y ) {
    BYTE address;

```

```
if(y!=1)
    address=lcd_line_two;
else
    address=0;
address+=x-1;
lcd_send_byte(0,0x80|address);
}
void lcd_putc( char c) {
    switch (c) {
        case '\f' : lcd_send_byte(0,1);
                    delay_ms(2);
                    break;
        case '\n' : lcd_gotoxy(1,2);    break;
        case '\b' : lcd_send_byte(0,0x10); break;
        default  : lcd_send_byte(1,c);  break;
    }
}
char lcd_getc( BYTE x, BYTE y) {
    char value;
    lcd_gotoxy(x,y);
    while ( bit_test(lcd_read_byte(),7) ); // wait until busy flag is low
    lcd.rs=1;
    value = lcd_read_byte();
    lcd.rs=0;
    return(value);
}
```

ANEXO B
REPORTE FOTOGRAFICO

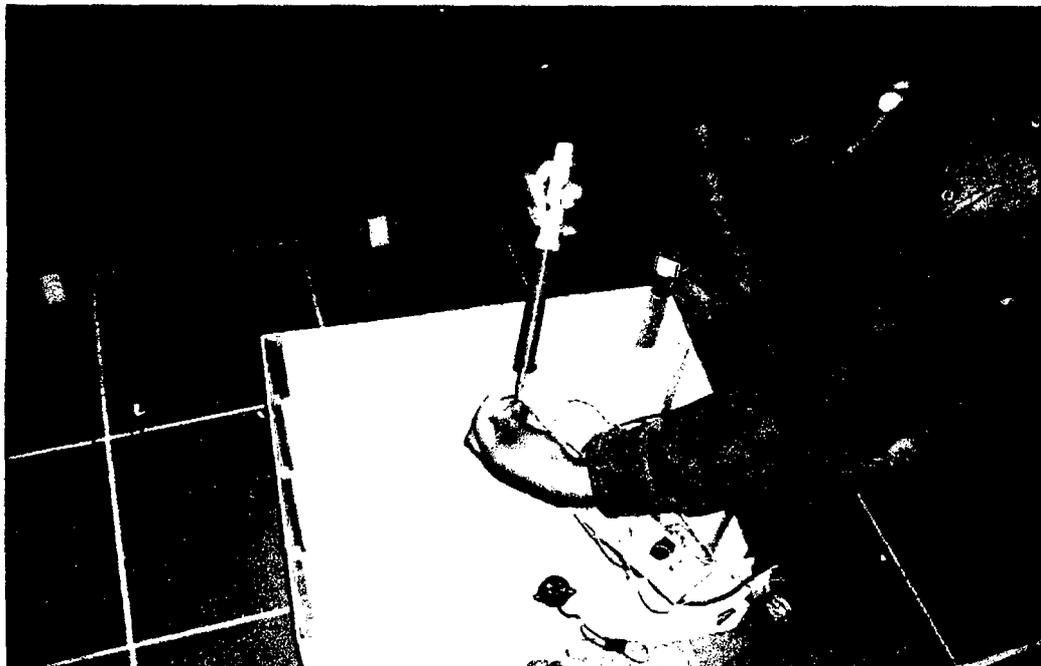


Foto N° 01: instalación del circuito en laboratorio



Foto N° 02: Pruebas de conectividad

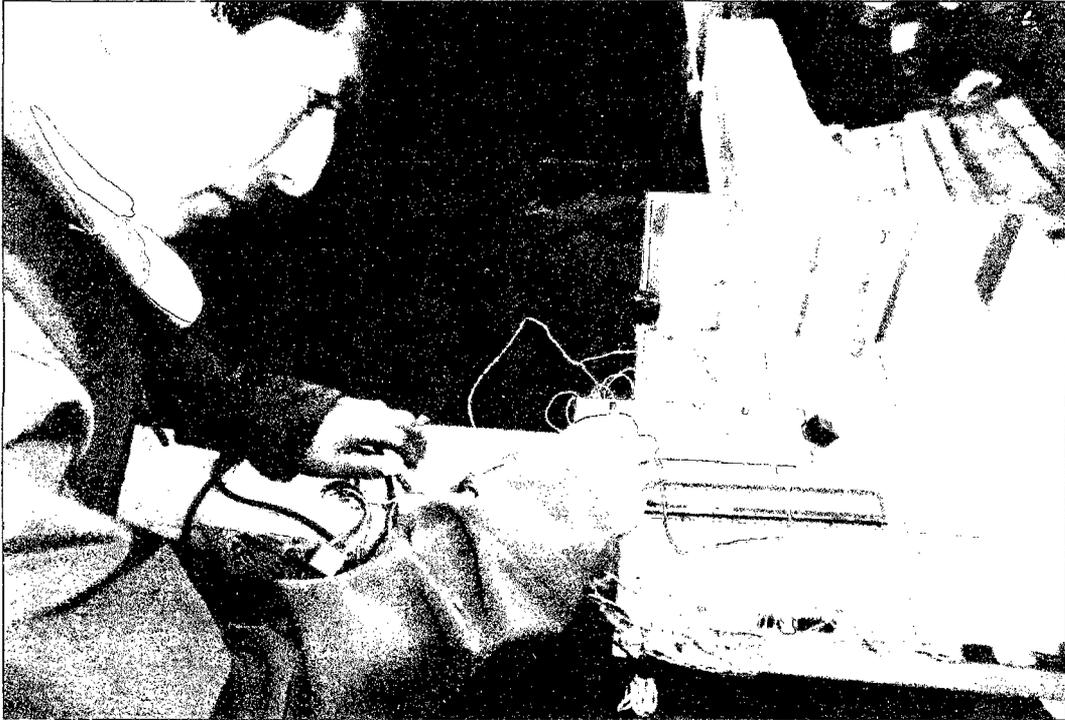


Foto N° 03: Alimentación del circuito para realizar pruebas en la localidad asignada

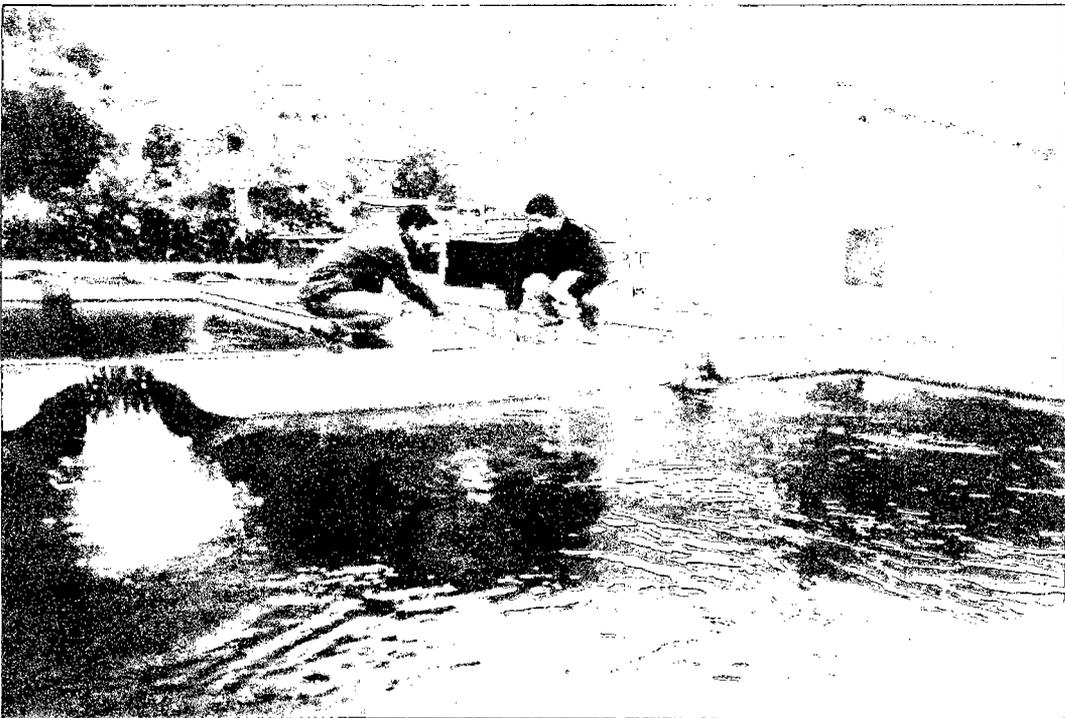


Foto N° 04: Realizando pruebas del proyecto implementado en la piscícola de Huiñamalca Parco.

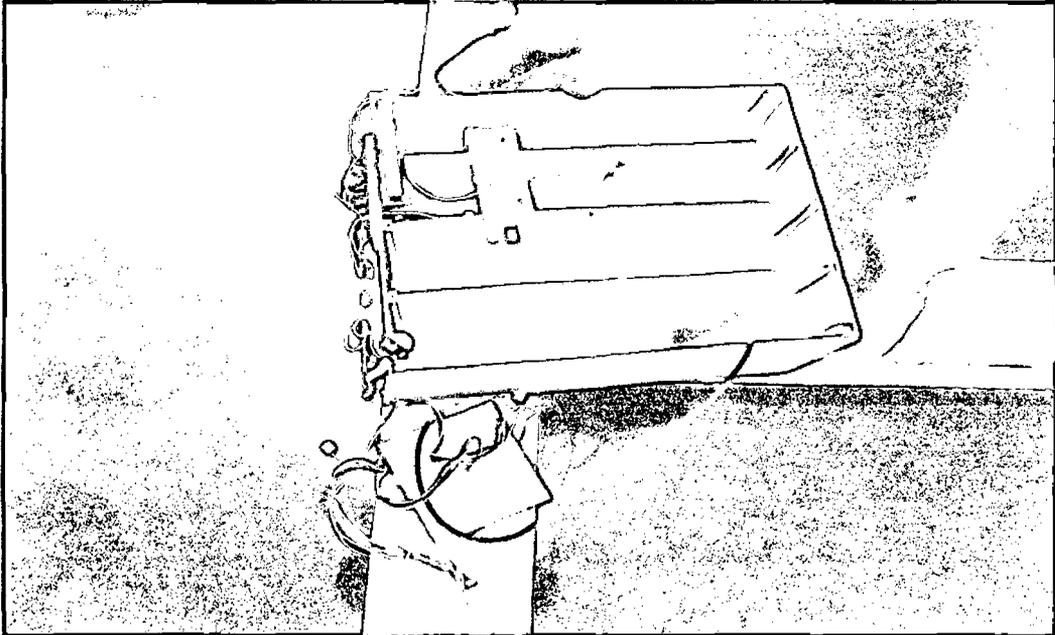


Foto N° 05: Circuito implementado y realizando pruebas de presencia y activación de bocina

ANEXO D

GUÍA DE OBSERVACIÓN DE CAMPO, REGISTRO DE PÉRDIDA DE ALEVINOS EN KG

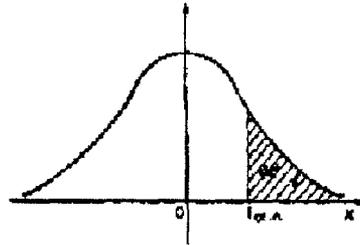
Día	MESES													
	MAYO		JUNIO		JULIO		AGOSTO		SETIEMBRE		OCTUBRE		NOVIEMBRE	
	C	E	C	E	C	E	C	E	C	E	C	E	C	E
01	0,3	0	0,4	0	0,2	0,01	0,3	0,01	0,4	0,01	0,5	0	0,5	0
02	0,3	0	0,4	0	0,3	0	0,3	0	0,7	0	0,3	0,01	0,6	0,01
03	0,3	0	0,3	0	0,2	0,01	0,3	0	0,6	0,01	0,6	0	0,5	0
04	0,4	0	0,4	0	0,4	0	0,4	0,01	0,7	0	0,5	0,01	0,7	0,01
05	0,2	0	0,5	0	0,2	0,01	0,2	0	0,5	0,01	0,4	0	0,4	0
06	0,3	0	0,4	0	0,3	0	0,5	0,01	0,8	0	0,5	0	0,7	0,01
07	0,3	0	0,4	0	0,2	0	0,3	0	0,4	0,01	0,3	0,01	0,6	0
08	0,4	0	0,3	0	0,3	0,01	0,3	0,01	0,6	0	0,6	0	0,5	0,01
09	0,4	0	0,5	0	0,2	0	0,3	0	0,5	0,01	0,4	0	0,7	0
10	0,3	0	0,4	0	0,4	0	0,5	0,01	0,7	0	0,5	0	0,6	0,01
11	0,2	0	0,3	0	0,3	0,01	0,5	0	0,6	0,01	0,6	0,01	0,7	0
12	0,3	0	0,4	0	0,4	0	0,4	0,01	0,5	0	0,5	0	0,6	0,01
13	0,2	0	0,3	0	0,3	0,01	0,5	0	0,6	0,01	0,3	0	0,4	0
14	0,4	0	0,4	0	0,4	0	0,4	0,01	0,7	0	0,4	0,01	0,6	0,01
15	0,3	0	0,4	0	0,3	0	0,3	0,01	0,6	0,01	0,6	0	0,5	0
16	0,3	0	0,4	0	0,4	0	0,4	0	0,5	0	0,5	0	0,7	0,01
17	0,3	0	0,4	0	0,3	0,01	0,5	0,01	0,8	0,01	0,6	0,01	0,6	0
18	0,4	0	0,6	0	0,3	0	0,3	0	0,5	0,01	0,4	0	0,5	0,01
19	0,3	0	0,4	0	0,4	0	0,5	0	0,7	0	0,5	0	0,7	0
20	0,3	0	0,4	0	0,2	0	0,3	0,01	0,4	0,01	0,4	0,01	0,5	0,01
21	0,3	0	0,5	0	0,3	0,01	0,3	0	0,7	0	0,4	0	0,8	0
22	0,4	0	0,3	0	0,4	0	0,4	0	0,6	0,01	0,6	0	0,5	0,01
23	0,3	0	0,4	0	0,3	0	0,5	0,01	0,8	0,02	0,5	0,01	0,8	0
24	0,3	0	0,5	0	0,4	0	0,5	0	0,4	0,01	0,5	0	0,6	0,01
25	0,3	0	0,4	0	0,3	0,01	0,3	0,01	0,6	0	0,6	0	0,7	0
26	0,4	0	0,5	0	0,4	0	0,4	0	0,7	0,01	0,5	0,01	0,6	0,01
27	0,4	0	0,3	0	0,3	0	0,5	0	0,6	0,02	0,6	0	0,5	0
28	0,3	0	0,6	0	0,4	0	0,4	0,01	0,5	0,01	0,4	0	0,6	0,01
29	0,4	0	0,5	0	0,5	0	0,5	0,01	0,7	0	0,5	0,01	0,7	0
30	0,3	0	0,4	0	0,4	0,01	0,4	0	0,6	0,01	0,6	0	0,6	0,01
31	0,4	0			0,3	0	0,3	0,01			0,4	0		
SUB TOTAL	10	0	12	0	10	0.10	12	0.15	18	0.20	15	0.10	18	0.15

C: Grupo control

E: Grupo experimental

ANEXO E

Tabla T-Student



$\alpha/2$	0,40	0,30	0,20	0,10	0,050	0,025	0,010	0,005	0,001	0,0005
1	0,325	0,727	1,376	3,078	6,314	12,71	31,82	63,66	318,3	636,6
2	0,289	0,617	1,061	1,886	2,920	4,303	6,963	9,925	22,31	31,60
3	0,277	0,584	0,978	1,638	2,353	3,182	4,541	5,841	10,22	12,94
4	0,271	0,569	0,941	1,533	2,132	2,776	3,747	4,604	7,173	8,610
5	0,267	0,559	0,920	1,476	2,015	2,571	3,365	4,032	5,893	6,859
6	0,263	0,553	0,906	1,440	1,943	2,447	3,143	3,707	5,208	5,959
7	0,263	0,549	0,896	1,415	1,895	2,365	2,998	3,499	4,785	5,405
8	0,262	0,546	0,889	1,397	1,860	2,306	2,896	3,355	4,501	5,041
9	0,261	0,543	0,883	1,383	1,833	2,262	2,821	3,250	4,297	4,781
10	0,260	0,542	0,879	1,372	1,812	2,228	2,764	3,169	4,144	4,587
11	0,260	0,540	0,876	1,363	1,796	2,201	2,718	3,106	4,025	4,437
12	0,259	0,539	0,873	1,356	1,782	2,179	2,681	3,055	3,930	4,318
13	0,259	0,538	0,870	1,350	1,771	2,160	2,650	3,012	3,852	4,221
14	0,258	0,537	0,868	1,345	1,761	2,145	2,624	2,977	3,787	4,140
15	0,258	0,536	0,866	1,341	1,753	2,131	2,602	2,947	3,733	4,073
16	0,258	0,535	0,863	1,337	1,746	2,120	2,583	2,921	3,686	4,013
17	0,257	0,534	0,863	1,333	1,740	2,110	2,567	2,898	3,646	3,965
18	0,257	0,534	0,862	1,330	1,734	2,101	2,552	2,878	3,611	3,922
19	0,257	0,533	0,861	1,328	1,729	2,093	2,539	2,861	3,579	3,883
20	0,257	0,533	0,860	1,325	1,725	2,086	2,528	2,845	3,552	3,850
21	0,257	0,532	0,859	1,323	1,721	2,080	2,518	2,831	3,527	3,819
22	0,256	0,532	0,858	1,321	1,717	2,074	2,508	2,819	3,505	3,792
23	0,256	0,532	0,858	1,319	1,714	2,069	2,500	2,807	3,485	3,767
24	0,256	0,531	0,857	1,318	1,711	2,064	2,492	2,797	3,467	3,745
25	0,256	0,531	0,856	1,316	1,708	2,060	2,485	2,787	3,450	3,725
26	0,256	0,531	0,856	1,315	1,706	2,056	2,479	2,779	3,435	3,707
27	0,256	0,531	0,855	1,314	1,703	2,052	2,473	2,771	3,421	3,690
28	0,256	0,530	0,855	1,313	1,701	2,048	2,467	2,763	3,408	3,674
29	0,256	0,530	0,854	1,311	1,699	2,045	2,462	2,756	3,396	3,659
30	0,256	0,530	0,854	1,310	1,697	2,042	2,457	2,750	3,385	3,646
40	0,255	0,529	0,851	1,303	1,688	2,021	2,423	2,704	3,307	3,551
50	0,255	0,528	0,849	1,298	1,676	2,009	2,403	2,678	3,262	3,495
60	0,254	0,527	0,848	1,296	1,671	2,000	2,390	2,660	3,232	3,460
80	0,254	0,527	0,846	1,292	1,664	1,990	2,374	2,639	3,195	3,415
100	0,254	0,526	0,845	1,290	1,660	1,984	2,365	2,626	3,174	3,389
200	0,254	0,525	0,843	1,286	1,653	1,972	2,345	2,601	3,131	3,339
500	0,253	0,525	0,842	1,283	1,648	1,965	2,334	2,586	3,106	3,310
∞	0,253	0,524	0,842	1,282	1,645	1,960	2,326	2,576	3,090	3,291