

UNIVERSIDAD MAYOR DE SAN ANDRES

FACULTAD DE TECNOLOGIA
ELECTRONICA Y TELECOMUNICACIONES



" DISEÑO E IMPLEMENTACION DE UN PROTOTIPO
DISPENSADOR PARA ALIMENTOS AUTOMATICOS GESTIONADO
DIRECTAMENTE Y REMOTAMENTE "

Proyecto de Grado Técnico presentado para la obtener el Grado de Licenciatura

POR: Univ. Jorge Luis Yavincha Balboa
TUTOR: M.Sc. Luis Richard Marquéz Gonzales

LA PAZ – BOLIVIA
JUNIO, 2023

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE TECNOLOGIA
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

Proyecto de Grado

**" DISEÑO E IMPLEMENTACION DE UN PROTOTIPO
DISPENSADOR PARA ALIMENTOS AUTOMATICOS
GESTIONADO DIRECTAMENTE Y REMOTAMENTE "**

Presentado por: Jorge Luis Yavincha Balboa

Para optar del grado académico de Licenciado en Electrónica y Telecomunicaciones

Nota numeral:

Nota literal:

Ha sido:

Lic. Julia Torrez Soria

Directora de la Carrera de Electrónica y Telecomunicaciones

Tutor: M. Sc. Luis Richard Marquéz Gonzales

Tribunal M.Sc. Edwin Félix Ibarra García

Tribunal Ing. José Arturo Ríos Altamirano

Tribunal Lic. Eder Tomas Jurado Moya

DEDICATORIA

Dedico este trabajo a quienes me dieron la oportunidad con su amor y desvelo la realidad de este sueño, mis padres: Martires Yavincha Balboa y Rosa Balboa Argani; mis hermanos Julio Cesar Yavincha Balboa, José Luis Yavincha Poma y Daniela Yavincha Balboa por su labor en apoyarme a culminar mis estudios, y a la amada Facultad y carrera donde nos brindan y comparten los saberes necesarios para el desarrollo de nuestra vida.

AGRADECIMIENTO

Quiero agradecer a los docentes de la carrera; a mi Tutor de Proyecto, M. Sc. Luis Richard Marqués Gonzales, quienes con esmero y pasión dirigieron mi camino en el campo de la electrónica y las telecomunicaciones; así mismo agradecer a mi familia y amigos y en especial a Andrés Reynaldo Ramos Calizaya quienes me apoyaron en los buenos y malos momentos fuera y dentro de la facultad y la carrera.

RESUMEN DEL PROYECTO

El objetivo del proyecto consiste en implementar una solución para las empresas que se oriente a la distribución de alimentos de tipo cereal y así mismo está dirigido para las granjas que desempeñan la labor de alimentar a sus animales, siendo que el proyecto está destinado a un dispensador de alimentos programable y remoto.

Esto ha consistido en crear un sistema basándose en la tecnología del microcontrolador ATMEGA 328p y Android, dando facilidad al usuario, brindándole así una herramienta para una organización automatizada en el manejo del presente prototipo.

En una primera etapa damos a conocer la solución que la electrónica puede brindar al problema. Además, se estudia el área donde este podría tener efecto y es donde se va a centrar el proyecto.

Posteriormente y como resultado de la etapa inicial se implementa una solución técnica que complementa con las conclusiones obtenidas en el estudio inicial, teniendo claras las especificaciones que debemos cumplir para tomar las decisiones oportunas en la elección de los componentes y los servicios a brindar que integra el presente proyecto.

La solución técnica consiste en el manejo de la programación y control hacia los propietarios con respecto a una mejor organización y automatización de la dispensadora de alimentos.

INDICE

INDICE GENERAL	Pag.
DEDICATORIA	I
AGRADECIMIENTOS	II
RESUMEN DEL PROYECTO	III
INDICE GENERAL	IV
CAPITULO I	1
1.1. INTRODUCCION	1
1.2. ANTECEDENTES	2
1.3. PLANTEAMIENTO DEL PROBLEMA	2
1.4. OBJETIVOS	3
1.4.1. Objetivo General	3
1.4.2. Objetivos Especificos	3
1.5. JUSTIFICACION	3
1.5.1. Justificacion tecnologica	3
1.5.2. Justificacion Social	3
1.5.3. Justificacion Academica	4
1.6. DELIMITACION DEL PROYECTO	4
1.6.1. Temporal	4
1.6.2. Espacial	4
1.6.3. Tematica	4
1.7. ALCANCES	5
1.8. MARCO METODOLOGICO	5
1.8.1. Definicion de la Investigacion aplicada	5
1.8.2. Diagrama en Bloques de la Investigacion Aplicada	5
1.8.3. Diseño de la Investigacion	5

1.8.4. Actividad y Tarea	6
1.8.5. Metodos y Tecnicas	6
CAPITULO II	7
2.1. MARCO TEORICO	7
2.1.1. Empresas de ALimentos	7
2.1.2. Procesos que se llevan a cabo en las empresas de alimentos	7
2.1.3. Principales actividades de las empresas de alimentos	8
2.1.4. Empresas en Bolivia	9
2.2. DISPENSADORES	10
2.2.1. Dispensadores comerciales	10
2.2.2. Dispensador por gravedad	11
2.2.3. Dispensadores Automaticos	14
2.3. MECANISMO DE DOSIFICACION	19
2.3.1. Dosificadores por Volumen	19
2.3.2. Dosificadores por Gravedad	19
2.3.3. Dosificadores de Tornillo Transportador	20
2.3.4. Dosificadores Rotativos	20
2.4. ALIMENTACION EN ANIMALES Y MASCOTAS	21
2.4.1. Cantidad de comidas diarias	21
2.5. SISTENSIS DEL CAPITULO	22
2.6. MICROCONTROLADORES	23
2.6.1. Microcontrolador ATmega328p	24
2.6.2. Características del Microcontrolador ATmega 328p	26
2.6.3. Registro de Memoria	30
2.6.4. Dispositivos Acoplables al Microcontrolador	33
2.6.5. Display LCD	33
2.6.6. Comunicación LCD	36

2.7. COMUNICACIÓN SERIAL	37
2.7.1. Comunicación Serie	37
2.7.2. Comunicación Serial-Serie	38
2.8. COMUNICACIÓN GPRS - GSM	38
2.8.1. Frecuencias	39
2.9. SERVOMOTORES	40
2.10. MOTOR PASO A PASO	41
2.10.1. Como funciona un motor paso a paso	42
2.10.2. Motor paso a paso - trabajo en modo de paso completo	43
2.10.3. Tipos de motores paso a paso	44
2.10.4. Motor paso a paso de iman permanente	45
2.10.5. Motor paso a paso de reluctancia variable	45
2.10.6. Motor paso a paso Hibrido	45
2.11. MOTORES UNIPOLARES Y BIPOLARES	46
2.11.1. Control de motor paso a paso	48
2.12. EL SENSOR ULTRASONICO	49
2.12.1. Especificaciones tecnicas	50
2.13. PROGRAMACION DE MICROCONTROLADORES	51
2.13.1. Estructura de un Sketch	51
2.13.2. Lenguaje de Programacion Arduino	52
2.14. ANDROID	54
2.14.1. Arquitectura Android	55
2.14.2. Versiones de Android	56
2.14.3. Usos y dispositivos	57
3.14.4. Aplicaciones	58

CAPITULO III	59
3.1. MARCO PRACTICO (INGENIERIA DE PROYECTO)	59
3.1.1 Introduccion	59
3.1.2. Descripcion General	59
3.1.3. Microcontrolador	59
3.1.4. Sim 800L	59
3.2. DIAGRAMA DE BLOQUES DEL PROYECTO	60
3.3. PROGRAMACION	61
3.3.1. Firmware de configuraciones	61
3.3.2. Software de configuracion	61
3.4. DESARROLLO DEL TRABAJO	61
3.4.1. Etapas	61
3.5. SOFTWARE CONFIGURACION DEL SISTEMA	62
3.5.1. Sistema Android con herramienta de Desarrollo	62
3.5.2. Proceso de configuracion	62
3.6. FIRMWARE DEL SISTEMA	66
3.6.1. IDE Arduino Herramienta de desarrollo	67
3.6.2. Diagrama de flujo	68
3.6.3. Programacion elaborada en IDE SETUP	70
3.7. HARDWARE DEL SISTEMA	74
3.7.1. Microcontrolador	74
3.7.2. Modulo Sim 8001 GPRS - GSM	75
3.7.3. Modulo RTC	76
3.7.3.1. Especificaciones y características	77
3.7.3.2. Funciones RTC	77
3.8. DIAGRAMA COMPLETO DEL CIRCUITO	78
3.8.1. Circuito del sistema dispensador	78
3.9. PRUEBAS Y RESULTADOS DE FUNCIONAMIENTO	81

CAPITULO IV	82
4.1. ANALISIS DE COSTOS	82
4.1.1. Costos de diseño de Software	82
4.1.1.1. Analisis de costo del Diseño del Software	85
4.1.2. Costos en la Construccion del Hardware	86
4.1.3. Costo Final del Proyecto	87
CAPITULO V	88
5.1. CONCLUSIONES Y RECOMENDACIONES	88
5.1.1. Conclusiones	88
5.1.2. Recomendaciones	88
CAPITULO VI	89
6.1. REFERENCIAS BIBLIOGRAFICAS	89
6.2. PAGINAS WEB	89
ANEXOS	91

INDICE DE FIGURAS

Figura 1.1 Diagrama en bloques de investigacion.....	5
Figura 2.1 Empresas habilitadas en bolivia.....	9
Figura 2.2 Dispensador industrial basico	10
Figura 2.3 Dispensador de gravedad.....	11
Figura 2.4 Dispensador automatico petsafe.....	12
Figura 2.5 Dispensador animal planet para mascotas.....	13
Figura 2.6 Dispensador de alimentos en granja	14
Figura 2.7 Dispensador automatico de alimentos en granja	15
Figura 2.8 Diagrama en bloques de dosificacion.....	19
Figura 2.9 Dosificador tipo tornillo	20
Figura 2.10 Dosificadores tipo rotativo.....	20
Figura 2.11 Microcontrolador atmega328p.....	25
Figura 2.12 Diagrama en bloques de microcontrolador.....	27
Figura 2.13 Diagrama interno ATMEGA328p	28
Figura 2.14 Unidad de control de reloj AVR.....	29
Figura 2.15 Diagrama en Bloques atmega328p.....	31
Figura 2.16 Grupos de memoria RAM	32
Figura 2.17 Iluminacion CCFL.....	35
Figura 2.18 Iluminacion led	35
Figura 2.19 Contraste	35
Figura 2.20 Display LCD16x2.....	36
Figura 2.21 Comunicación Serial - Serie.....	38
Figura 2.22 Servomotor	40
Figura 2.23 motor paso a paso	44
Figura 2.24 Motor Bipolar	47
Figura 2.25 Motor unipolar.....	47
Figura 2.26 Modulo ultrasonico.....	51

Figura 2.27 Estructura principal del programa	52
Figura 2.28 Arquitectura principal de android	55
Figura 3.1 creacion de nuevo proyecto app inventor	62
Figura 3.2 Nombre de proyecto	63
Figura 3.3 Division de disposiciones APPinventor	63
Figura 3.4 Selección de interfaz	64
Figura 3.5 Entorno final APK	64
Figura 3.6 programacion entorno ventana usuario	65
Figura 3.7 programacion de entorno registro de datos	65
Figura 3.8 programacion del entorno envio de datos	66
Figura 3.9 diagrama de flujo telefono apk	68
Figura 3.10 diagrama de flujo microcontrolador	69
Figura 3.11 diagrama del prototipo	74
Figura 3.12 Microcontrolador Atmega 328p	75
Figura 3.13 Sim 800 L	76
Figura 3.14 Modulo RTC	78
Figura 3.15 Simulacion de proteus del prototipo	79
Figura 3.16 Diseño PCB Placa	80
Figura 3.17 Pruebas de funcionamiento	81

INDICE DE TABLAS

Tabla 2.1 Dispensadores Industrilaes Basicos	18
Tabla 2.2 Tabla de alimentos	22
Tabla 2.3 Banda de Frecuencias GSM	40
Tabla 2.4 Verciones y Lanzamientos de Android	57
Tabla 4.1 Coeficionetes a, b, c, d según el tipo de proyecto	83
Tabla 4.2 Valoracion de los Conductores de coste para determinar la FAE	85
Tabla 4.3 Valoracion del Hardware	87

INDICE DE ECUACIONES

Ecuacion 4.1 Ecuacion de cocomo intermedio.....	82
Ecuacion 4.2 Factor de ajuste del esfuerzo	84
Ecuacion 4.3 costo economico de software.....	86

CAPITULO I

1.1. INTRODUCCION

El presente proyecto de grado está dirigido a un diseño de un prototipo dispensador para alimentos automáticos gestionado directamente y de manera remota.

En el presente documento se establece los antecedentes por el cual es necesario el desarrollo del diseño de este prototipo. La implementación del dispensador de alimentos va dirigido a dos campos en los cuales podremos brindar una solución: área de Granjas este dirigido para animales, como también para empresas distribuidoras en el rubro de alimentos de tipo Granulado, ya que se desea hacerlo con un sistema de control mediante programación para un accionamiento remoto, para el fácil uso del usuario. El prototipo va ser controlado desde el celular con una aplicación y sistema de red GSM (Sistema Global de Comunicaciones Móviles).

El diseño de este proyecto podría ser implementado dentro de la sociedad para el fácil manejo y control de alimentación de sus mascotas, además para estudiantes del área de electrónica y Telecomunicaciones, que les permita dar un enfoque de base y mejoramiento en el área que deseen investigar o desarrollar.

1.2. ANTECEDENTES.

En la actualidad podemos observar cuanto a llegado a ocupar la electrónica y la automatización en nuestro diario vivir, por lo que se desea dar a conocer la factibilidad de realizar un dispensador automatizado de alimentos y que se puede controlar por programación de manera remota.

El estudio de factibilidad se lo realiza tomando en cuenta aspectos importantes al área, para conocer si un proyecto es o no viable; estos son: Estudio Técnico y el Estudio de programación.

En el documento se muestra la idea del perfil de proyecto que consiste en: el diseño y la fabricación de un dispensador automático de alimentación, que puede ser programado de manera remota.

Se desea que el dispensador puede ser controlado a través de programación, por medio de dos alternativas posibles y crear horarios de acuerdo a las necesidades de alimentación o necesidad de como desea que se dispensen los productos de tipos granulado, para una respectiva empresa de esa área.

1.3. PLANTEAMIENTO DEL PROBLEMA.

El diario vivir de la población conlleva una vida de muchas actividades por lo que el tiempo es insuficiente para cumplir con las tareas propuestas, ya sea dado por una empresa que siempre busca efectivizar su producto buscando realizar mayores tareas en poco tiempo al igual que una persona que dirige una granja, por lo que se debe plantear un desarrollo obligado en todos sus campos para buscar soluciones que traigan consigo innovación y agilización; se está viviendo la era donde el ser humano desea realizar una gran cantidad de tareas de una manera rápida y eficiente, por ello las soluciones deben ser prácticas, sencillas, eficientes y oportunas.

Para evitar todas estas preocupaciones y darles solución se propone construir un dispensador controlado remotamente, por un temporizador el cual les permita a las personas ahorrar tiempo y estar vigilando la maquinaria. Así que en cuestión de la nutrición se deben garantizar las siguientes condiciones: que se administre en las proporciones adecuadas, que se provea en horarios regulares y asegurarse que el alimento contenga los nutrientes necesarios.

1.4. OBJETIVOS.

1.4.1. Objetivo general.

Diseñar e implementar un sistema prototipo para un dispensador automático de alimentos gestionada de forma directa y remota para el hogar o proyectada para el área avícola.

1.4.2. Objetivo específico

- Realizar el programa base para el circuito con el microcontrolador.
- Diseñar e implementar el circuito electrónico para el control de los periféricos.
- Diseñar el control de sistema de acuerdo a los requerimientos físicos.
- Determinar las características del prototipo que se ajuste a las necesidades del usuario.

1.5. JUSTIFICACION.

1.5.1. Justificación tecnológica

Se tomó el interés de usar la tecnología y la reducción de costos para el desarrollo del prototipo de dispensador, se hará uso del software y hardware libre.

1.5.2. Justificación social

Con este trabajo se pretende presentar un prototipo que daría solución a un grupo del área de alimentación, realizando un dispensador de alimento eficiente y logrando dar así un control totalmente a disposición del usuario interesado.

1.5.3. Justificación académica

Para el diseño se aplicará todo el conocimiento adquirido en la carrera de electrónica y telecomunicaciones, realizando así un prototipo funcional para la solución del problema planteado, de acuerdo a las materias aprendidas en el área de la carrera.

1.6. DELIMITACION DEL PROYECTO.

En este proyecto se implementará: un prototipo de control sobre un dispensador automático de alimentos, facilitando y mejorando la calidad de vida en mascotas oh área de granja avícola implementando un prototipo para el usuario que tendrá que programar mediante tiempos oh control remoto de dispensación.

1.6.1. Temporal

El tiempo de desarrollo, en las magnitudes del proyecto se plantea en un aproximado de 4 meses o más para garantizar el uso del sistema del presente prototipo, por lo que es factible para su aplicación.

1.6.2. Espacial

Al ser un prototipo este nos muestra de manera general el funcionamiento del dispensador, pero dependiente a la empresa que desee implementar o adquirir el proyecto, se debe modificar el tamaño y volver adaptable a las necesidades del usuario que desee adquirir o implementar este proyecto.

1.6.3. Temática

El conocer cada empresa o persona que desee adquirir y para que alimento desea el proyecto nos llevó a formar este prototipo de manera general para un uso adaptable a diferentes productos y áreas en el ámbito de dispensación de alimentos.

1.7. ALCANCES

el sistema de prototipo que se desea pueda ser adaptable a las diferentes áreas de mercado está diseñado para el uso de dispensación de alimentos de manera automática, controlada remotamente para mejorar la producción y ahorrar tiempo al usuario que vea las necesidades de mejoramiento.

1.8. MARCO METODOLÓGICO

1.8.1. Definición de la investigación aplicada

La Investigación Aplicada tiene por objetivo resolver un determinado problema o planteamiento específico, enfocándose en la búsqueda y consolidación del conocimiento para su aplicación y, por ende, para el enriquecimiento del desarrollo cultural y científico.

1.8.2. Diagrama en Bloques de la Investigación Aplicada



Figure 1.1 Diagrama en bloques de investigación

Fuente: J.J. Quispe Chávez (Proyecto de Grado)

1.8.3. Diseño de la Investigación

En este caso se quiere aplicar una tecnología que permita el control programable y remoto en el área de dispensación de alimentos. Por consiguiente, el tipo de ámbito al que se aplica es muy específico y bien delimitado, ya que no se trata de explicar una amplia variedad de situaciones, sino que más bien se intenta abordar un problema específico.

Esta investigación aplicada busca la generación de conocimiento con aplicación directa a los diferentes problemas, se basa fundamentalmente en los hallazgos tecnológicos de la investigación básica, ocupándose del proceso de enlace entre la teoría y el producto.

1.8.4. Actividad y Tarea

- Escoger el algoritmo necesario.
- Realizar la programación en el microcontrolador respectivos para el manejo de los módulos.
- Realizar el respectivo funcionamiento del dispensador de alimentos programable y remoto.
- Elaboración del programa e interface para el funcionamiento del proyecto.

1.8.5. Métodos y Técnicas

De acuerdo al planteamiento del problema, se tomó en cuenta el tipo de estudio aplicativo; es decir, una investigación centrada en encontrar mecanismos o estrategias que permitan lograr un objetivo concreto, como conseguir un elemento o bien que pudiese ser de utilidad. Frecuentemente el propósito del investigador es describir situaciones y eventos, es decir cómo es y cómo se manifiesta determinado fenómeno. Los estudios aplicativos buscan la generación de conocimiento con aplicación directa a los diferentes problemas, se basa fundamentalmente en los hallazgos tecnológicos de la investigación básica, ocupándose del proceso de enlace entre la teoría y el producto, en este perfil de proyecto se aplicará una tecnología que es el control automático programable remoto para un dispensador de alimentos.

2 CAPITULO II

2.1. MARCO TEORICO

2.1.1. Empresas de alimentos

Las empresas de alimentos o empresas alimenticias son aquellas que buscan abastecer a los clientes con productos de consumo de buena calidad. Este tipo de empresas también se encargan de implementar procesos de calidad en sus líneas de producción, además de la elaboración, transformación, preparación, conservación y envasado de los alimentos para consumo humano e incluso animal.¹

La industria alimentaria es la parte de la industria que se encarga de todos los procesos relacionados con la cadena alimentaria. Se incluyen dentro del concepto las fases de transporte, recepción, almacenamiento, procesamiento, conservación, y servicio de alimentos de consumo humano y animal. Las materias primas de esta industria consisten principalmente de productos de origen vegetal (agricultura), animal (ganadería) y fúngico (perteneciente o relativo a los hongos). Gracias a la ciencia y la tecnología de alimentos el progreso de esta industria se ha visto incrementado y nos ha afectado actualmente en la alimentación cotidiana, aumentando el número de posibles alimentos disponibles en la dieta. El aumento de producción ha ido unido con un esfuerzo progresivo en la vigilancia de la higiene y de las leyes alimentarias de los países intentando regular y unificar los procesos y los productos, y de la industrialización de productos alimentarios.

2.1.2. Procesos que se llevan a cabo en las empresas de alimentos

La empresa de alimentos se clasifica en cinco procesos diferentes la fabricación de alimentos: proceso de manipulación, extracción, elaboración, conservación y envasado de los alimentos.

¹ <https://www.quiminet.com/empresas/empresas-de-alimentos-2736068.htm#:~:text=Este%20tipo%20de%20empresas%20tambi%C3%A9n,consumo%20humano%20e%20incluso%20animal.>

1. Procesos de manipulación de alimentos: se refiere al momento en que el personal capacitado tiene contacto directo con los alimentos durante los diferentes procesos.
2. Proceso de extracción: se extraen los elementos que sean externos al producto a través de calor, secado o filtrado del mismo.
3. Procesos de elaboración: su objetivo es la transformación del alimento a otro mediante la cocción, destilado, secado y fermentación (en el caso de bebidas).
4. Procesos de conservación: en esta etapa se termina con la actividad microbiana, prolongando la vida de alimento, puede ser a través de pasteurización, esterilización con antibiótico, esterilización por radiación la radiación ionizante (usando en control de envases) y la radiación de microondas.
5. Proceso de envasado: una vez que el alimento está listo es empacado o envasado de acuerdo a su estado (sólido o líquido) y se inicia con la comercialización.

2.1.3. Principales actividades de las empresas de alimentos

Dentro de las principales actividades de una empresa de alimentos se encuentran:

- Preparación de productos alimentación
- Fabricación de productos alimenticios
- Transformación de productos alimenticios
- Elaboración de productos alimenticios
- Envasado de productos alimenticios
- Almacenamiento de productos alimenticios
- Transporte de productos alimenticios
- Distribución de productos alimenticios
- Manipulación de productos alimenticios
- Venta de productos alimenticios
- Suministro y servicio de productos alimenticios

Las empresas de alimentos tienen diversas expectativas alimenticias al generar más variedades de alimentos, trabajan con estrategias muy definidas y un estricto control de calidad.²

2.1.4. Empresas en Bolivia

Existen alrededor de 330 empresas en Bolivia entre los cuales algunos son parte de la OEA³

Asociación de Caficultores	-	LA PAZ	Café arábica verde
Asociación de Productores de Café Ecológico Regional Larecaja-Apcerl	-	LA PAZ	Café en grano verde
Asociación de Productores de Café Ecológico Río Mercedes	-	LA PAZ	Café extra orgánico
Avichoi Agro Export S.R.L.	-	LA PAZ	Ají seco en vaina
Bodeco S.R.L.	-	LA PAZ	Stevia La Bolivianita
Bolivian Maca Export	-	LA PAZ	Tea Mack
Celccar LTDA.	-	LA PAZ	Grano oro verde
Centro de Comercialización de Alto Beni S. R. L.	-	LA PAZ	Banano
Cervecería Boliviana Nacional S.A.	-	LA PAZ	
Ciapec LTDA. Coop. Int. Agri. De Pro.	-	LA PAZ	Café orgánico
Coaine Cooperativa Agropecuaria Nor Este	-	LA PAZ	Café verde en grano
Comercializadora Buenaventura S.R.L.	-	LA PAZ	
Comercializadora Mony Import Export S.R.L.	-	LA PAZ	
Compañía Cervecera Boliviana	-	LA PAZ	Judas, trimalta
Compañía de Alimentos LTDA.	-	LA PAZ	Helado
Cooperativa Agropecuaria y Comercialización Alto Sajama	-	LA PAZ	
Cooperativa Agropecuario de Comercialización Corpus Cristhv Illampu LTDA.	-	LA PAZ	Grano oro verde

Figura 2.1 Empresas habilitadas en Bolivia

Fuente: <http://vci.produccion.gob.bo/siexco/web/app.php/empresa/sector/6>

² <https://www.quiminet.com/empresas/empresas-de-alimentos-2736068.htm#:~:text=Este%20tipo%20de%20empresas%20tambi%C3%A9n,consumo%20humano%20e%20incluso%20animal.>

³ <http://vci.produccion.gob.bo/siexco/web/app.php/empresa/sector/6>

2.2. DISPENSADORES

En el mercado existen diferentes tipos de dispensadores, pudiendo dividirse en dos tipos, los automáticos y gravedad. En la actualidad los dispensadores automáticos tienen una gran acogida en el campo comercial por su factibilidad y prestaciones que posee, pero esto no pueden ser adquiridos por tener un precio elevado. Esto impide que algunos propietarios posean un dispensador.

2.2.1. Dispensadores comerciales

Actualmente en el mercado existen una infinidad de dispensadores de alimentos, que tienen diferentes características y prestaciones. Los dispensadores van desde simples comederos mecánicos que dosifican manualmente el alimento, hasta dispositivos semiautomáticos con sensores, interfaz gráfica y programación de horarios e incluso algunos poseen cámaras web con las cuales los propietarios pueden tener un mayor acercamiento con su mascota y/o tener una visualización de su empresa en el desarrollo de las dispensaciones.⁴



Figura 2.2 Dispensador industrial básico

Fuente: <http://vci.produccion.gob.bo/siexco/web/app.php/empresa/sector/6>

⁴ <https://dspace.ups.edu.ec/bitstream/123456789/19384/1/UPS-CT008857.pdf>

2.2.2. Dispensador por gravedad

Cuenta con una unidad de almacenamiento y un recipiente unido al mismo que permite que el alimento se deslice al recipiente a medida



Figura 2.3 Dispensador de gravedad

Fuente: https://es.made-in-china.com/tag_search_product/Food-Dispenser_usngsghn_1.html

Los dosificadores por gravedad siempre dosifican el alimento hasta que se agota toda su capacidad. No presenta ningún tipo de control ni automatización y en su mayoría tienen una capacidad muy limitada de almacenamiento. Son fabricados en materiales sencillos y de bajo costo. Son de fácil transporte, no requieren alimentación eléctrica, son factibles de conseguir y económicos⁵.

⁵ <https://dspace.ups.edu.ec/bitstream/123456789/19384/1/UPS-CT008857.pdf>

En el mercado encontramos dosificadores de alimento con mayor innovación como es el dosificador Animal planeta, el cual tiene mayor implementación tecnológica, capaz de programar los horarios de alimentación de los animales.

Construido en materiales resistentes con piezas desmontables que facilitan su limpieza y mantenimiento. Es un dispositivo portable que funciona con alimentación eléctrica de 3 baterías de 1.5V, cuenta con una pantalla LCD que muestra la hora y el nivel de alimento, tiene una capacidad no mayor a 5Kg requiere una constante llenado y supervisión en caso del uso dedicado a animales de distintos tipos de razas.



Figura 2.4 Dispensador automático petsafe

Fuente: www.google.com/search?q=dosificador+de+animal+planet&rlz

Pesare, ofrecen dispensadores con mayor capacidad e implementación de nuevas características como es el caso de Healy Pat Simple Reed, este es un dosificador con capacidad de almacenamiento de hasta 10 Kg de Alimento, fabricado en plástico y componentes mecánicos de acero inoxidable, capaz de ser programado para dosificar hasta doce veces en un día y también de programar la cantidad a dosificar desde $\frac{1}{4}$ de taza hasta 4 tazas de alimento por porción. Alimentado eléctricamente por 4 baterías de 1.5 V o por un adaptador de corriente.



Figura 2.5. Dispensador animal planet para mascotas

Fuente: <https://www.ebay.com/itm/304565944456>

Existen dispensadores más sencillos que requieren mayor supervisión humana para verificar que estén llenos como es el caso del alimentador rotacional de 5 comidas de Pesare como , es un dispensador que tiene forma de plato con 5 compartimientos para el alimento, que rota abriendo cada porción para que el animal se alimente, funciona con 4 baterías de 1.5 v. en este dispensador se puede programar para que sea accionado en horarios determinados, presenta limitaciones en la capacidad de almacenamiento de alimento, requiere ser llenado con frecuencia, tiene la limitación de poder dosificar 5 porciones fijas. El equipo está concebido para la alimentación de animales pequeños, fabricado con materiales de plástico y el acero inoxidable.



Figura 2.6 Dispensadores de alimentos en granja

Fuente: evmross.top/products.aspx?cname=comederos+automaticos+para+pollos

2.2.3. Dispensadores Automáticos

Existen dispensadores automáticos, pero una de sus grandes desventajas es que no cuenta con un sensor detector de monóxido de carbono que frecuentemente provocaría el envenenamiento de la mascota. A través del sensor el dispositivo envía una alerta al dispositivo móvil del propietario en caso de existir monóxido de carbono, con esto se consigue salvaguarda la vida la mascota. Estos sistemas no son tan comunes en el país debido a sus costos elevados provocados por su importación.





Figura 2.7 Dispensadores automático de alimentos en granja

Fuente: evmross.top/products.aspx?cname=comederos+automaticos+para+pollos

El gran aumento de mascota a nivel mundial es considerable ya que forman parte del círculo familiar. Para las mascotas se busca una alimentación adecuada basada en una dieta balanceada y equilibrada. Por estos motivos los dispensadores de alimento para mascotas se han vuelto comerciales, como se puede apreciar en la Tabla, en la que se detallan algunos de los dispensadores más comercializados según sus características.

Dosificadores Programables Presenciales

Nombre	Precio en dólares (Ecuador)	Precio en euros (España)	Fabricante	Descripción
<p>AutomaticPat Feeder</p> 	300	199.95	Reed & Go	<p>-Tiene una interfaz programable a través de una aplicación instalada en el teléfono inteligente.</p> <p>-Puede conectarse a una red inalámbrica a internet por lo que se puede tener acceso 24/7 y desde cualquier lugar del mundo.</p> <p>-Tiene integrado una cámara con la que se puede ver a la mascota para mayor cuidado.</p> <p>-Funciona conectándolo a la energía eléctrica [11].</p>
<p>Diamond Series V3</p> 	130	61.88	Crown Majestic	<p>-Puede administrar la comida en cantidades de 1/2; 3/4; 1 y 2 tazas de alimento seco.</p> <p>-Tiene una pantalla LCD que se programa la cantidad y la hora que se desea administrar el alimento a la mascota.</p> <p>-Permite el uso de 4 pilas AA (no incluidas) o combinarlo con el adaptador de energía eléctrica [12].</p>

<p>Mvpower</p> 	<p>80</p>	<p>45.99</p>	<p>MVpower ®</p>	<ul style="list-style-type: none"> -El funcionamiento es con 3 pilas tipo D no incluidas. -Es programable con 3 comidas diarias a las horas que deseemos según la rutina de nuestros animales. -No posee control remoto. -Tiene una capacidad de 5.5 Kg. se rellena cuando es necesario de forma manual. -Tiene un micrófono con el que se puede grabar un breve mensaje con la voz del propietario cada vez que se dispensa la comida [15].
<p>Aspen Pat Lebistro</p> 	<p>120</p>	<p>68.09</p>	<p>Aspen Pat</p>	<ul style="list-style-type: none"> -Tiene pantalla LCD para programar comidas y establecer tamaños de porciones. -Funciona con 3 pilas D (no incluidas). -No posee control remoto. -Tiene una capacidad de 1.6 Kg. se rellena cuando es necesario de forma manual [16].



<p>Petcorp Food</p> 	120	72.53	Petcorp Food Dispenser	<ul style="list-style-type: none"> -Alimenta en forma automática, para un período determinado. -Se programa según la cantidad y periodicidad que su mascota necesita (hasta 4 veces al día de volumen variable). -Tiene un temporizador con un respaldo de baterías. - Tiene una capacidad de 2.5 Kg [17].
<p>Perfect Pat Feeder</p> 	600	450	Perfect Pat Feeder	<ul style="list-style-type: none"> -Cuenta con un sensor que le permite calcular el peso exacto de la ración de alimento. -Funciona conectado a la red eléctrica y también posee una batería recargable. -Tiene una capacidad de 5Kg [17].

Tabla 2.1 Dispensadores Industriales Básicos

Fuente: <https://dspace.ups.edu.ec/bitstream/123456789/19384/1/UPS-CT008857>

2.3. MECANISMO DE DOSIFICACIÓN

Existen dispensadores comerciales fabricados de acuerdo con las características del producto, al proceso de dosificación y a la cantidad a dosificar.

Los dispensadores se clasifican por volumen y gravedad, ambos tipos trabajan de forma continua o por intervalos de tiempo

2.3.1. Dosificadores por Volumen

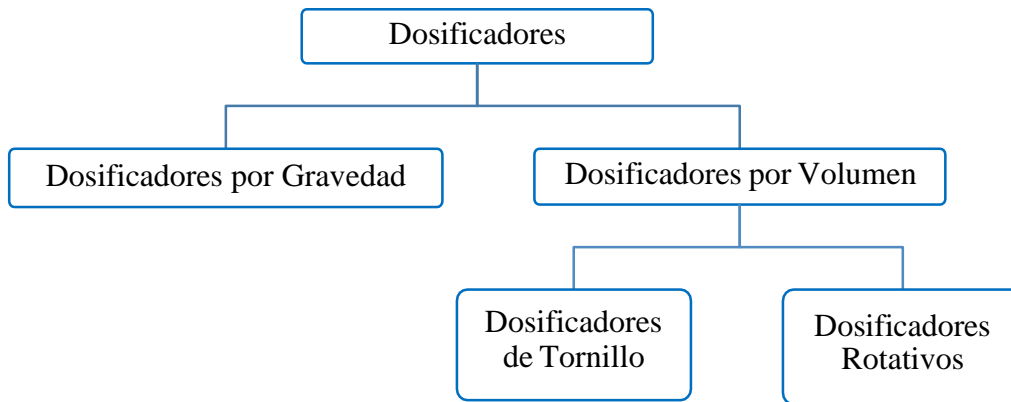


Figura 2.8 Diagrama Bloques de Dosificación

Fuente: www.evmross.top/products.aspx?cname=comederos+automatico

Los mecanismos para la generación de flujo del alimento, por lo general usan tornillos de transporte, bandas, compuertas rotativas y válvulas para mover el producto. Dosifican el producto por una velocidad constante del flujo del producto controlando los tiempos de funcionamiento al volumen dosificado.

2.3.2. Dosificadores por Gravedad

Generan movimiento del producto a partir de mecanismos que tienen un control de peso a la salida y un control en la velocidad de los mecanismos, que determina la cantidad de material dosificado de manera precisa.

2.3.3. Dosificadores de Tornillo Transportador

El dosificador de tornillo transportador es a partir del giro de un tornillo sinfín dentro de un canalón fijo por donde se transporta longitudinalmente el producto, el tornillo está conectado a la unidad de almacenamiento que lo alimenta de materia prima, mientras que hay un orificio de salida del producto. Es uno de los más usados por su simplicidad y adaptabilidad a cualquier tipo de materia prima, es capaz de dispensar en respectivos intervalos de tiempo.

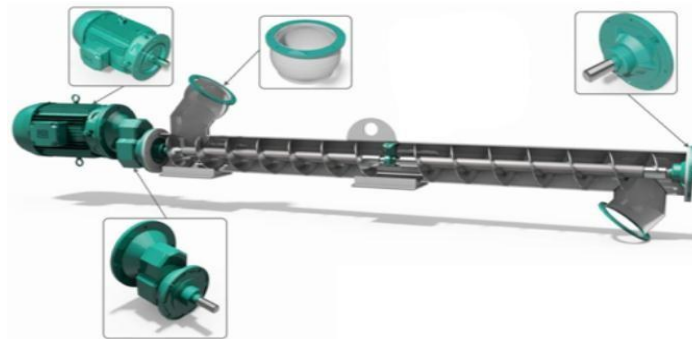


Figura 2.9 Dosificador tipo tornillo

Fuente: ingemecanica.com/tutorial_semanal/tutorialn143.html

2.3.4. Dosificadores Rotativos

El elemento principal es la compuerta rotativa. La compuerta rotativa es de construcción simple y robusta, pero de menos precisión.



Figura 2.10 Dosificador tipo rotativo

Fuente: www.palamaticprocess.fr/sites/default/files/file/dosification-palamatic-process

2.4. ALIMENTACIÓN EN ANIMALES Y MASCOTAS

La alimentación en las mascotas varía dependiendo la raza las cuales pueden considerarse (pequeñas, medianas, grandes); por tanto, una proporción adecuada en horarios definidos, asegura un buen crecimiento, peso adecuado y junto con la actividad física darán como resultado una mascota sana.

El alimento balanceado correctamente elaborado y con los nutrientes adecuados es necesario para una mascota sana; en este sentido, se debe tener asesoramiento o recomendación propicia de un médico veterinario que despejará de alguna duda o situación que se encuentre la mascota. Además, es el que siempre ayudará a suministrarle vitaminas y minerales.⁶

A continuación, en la Tabla se puede apreciar que, según el peso de la mascota, se tiene un suministro en gramos de alimentación. De acuerdo a estudios relacionados con el cuidado de la mascota, se estableció un estándar de alimentación para cada raza de perros pequeños, medianos y grandes.⁷

2.4.1. Cantidad de Comidas Diarias

- Hasta los 3 meses: 4 comidas.
- De los 3 meses a los 6 meses: 3 comidas.
- De los 6 meses a los 17 meses: 2 comidas.
- Después de los 18 meses se podrá optar por 1 comida (sin embargo, se recomienda siempre 2 ingestas para una mejor asimilación).⁸

⁶ <https://www.palamaticprocess.fr/sites/default/files/file/dosification-palamatic-process.pdf>

⁷ <https://www.palamaticprocess.fr/sites/default/files/file/dosification-palamatic-process.pdf>

⁸ <https://www.biopasos.com/documentos/087.pdf>

Alimentar a una mascota es algo que se tiene que aprender, de la alimentación depende gran parte la salud de la mascota, no debemos sobrealimentarlo, especialmente en las primeras etapas de la vida. Una deficiente alimentación puede ser la sentencia para poseer una mascota débil y enferma.⁹

Peso (Kg)	Suministro Cantidad (Gramos)	Cantidad (Porción en Tazas)
Razas Pequeñas		
1Kg – 3Kg	50 a 90 g	0.5 a 1
3Kg – 5Kg	90 a 120 g	1 a 1.25
5Kg – 10Kg	120 a 190 g	1.25 a 1.75
Razas Medianas		
10Kg – 15Kg	190 a 260 g	1.75 a 2.5
15Kg – 20Kg	260 a 310 g	2.5 a 3
Razas Grandes		
20Kg – 30Kg	310 a 410 g	3 a 4
30Kg – 40Kg	410 a 500 g	4 a 4.75

Tabla 2.2 Tabla de alimentos

Fuente: <https://www.biopasos.com/documentos/087.pdf>

2.5. SÍNTESIS DEL CAPITULO

En este capítulo se abordaron los dos tipos de dispensadores comerciales como son de gravedad y automáticos en el mercado. Se detalló su funcionamiento y características, especificando el aporte que tiene cada dispensador para las mascotas. Además, se analizaron los mecanismos de dosificación para un mejor dispensado de alimento.

⁹ <https://www.biopasos.com/documentos/087.pdf>

Luego se analizaron las cantidades diarias de comida para que las mascotas estén sanas, manteniendo una dieta equilibrada y balanceada.

2.6. MICROCONTROLADORES

Es un circuito integrado programable 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. Estas últimas características es la que le confiere la denominación de “controlador incrustado”. El microcontrolador es un computador dedicado. En su memoria solo reside un programa destinado a gobernar una aplicación determinada, sus líneas de entrada y salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar, y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos.

Una vez programado y configurado el microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se destina a gobernar una sola tarea.

El número de productos que funcionan en base a uno o varios microcontroladores aumenta de forma exponencial. No es aventurado pronosticar que en el siglo XXI habrá pocos elementos que carezcan de microcontrolador. En esta línea de prospección del futuro, la empresa Dataquest calcula que en cada hogar americano existirán varios centenares de microcontroladores en los comienzos del tercer milenio.

Los electrodomésticos de línea blanca (lavadoras, hornos, lavavajillas, etc.) y de línea marrón (televisores, videos, aparatos musicales, etc.) incorporan numerosos microcontroladores.

Igualmente, los sistemas de supervisión, vigilancia y alarma en los edificios utilizan estos chips.

También se emplean para optimizar el rendimiento de ascensores, calefacción, aire acondicionado, alarmas de incendio, robo, etc.

Las comunicaciones y sus sistemas de transferencia de información utilizan profusamente estos pequeños computadores incorporándolos en los grandes automatismos y en los modernos teléfonos.

La instrumentación y el electro medicina son dos campos idóneos para la implantación de estos circuitos integrados. Una importante industria consumidora de microcontroladores es la de automoción, que los aplica en el control de aspectos tan populares como la climatización, la seguridad y los frenos ABS.

Las comunicaciones y los productos de consumo general absorben más de la mitad de la producción de microcontroladores. El resto se distribuye entre el sector de la automoción, los computadores y la industria.

2.6.1. Microcontrolador ATmega328p

Es un circuito integrado de alto rendimiento que está basado en un microcontrolador RISC, combinando 32kb (ISP) flash una memoria con la capacidad de leer mientras escribe, 1KB de memoria EEPROM, 2KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles / contadores con modo de comparación, interrupciones internas y externas, programador de modo USART, una interfaz serial orientada a byte de 2 cables, SPI puerto serial, 6 canales 10 bit conversor A / D (Canales en TQFP y QFN / MLF packages), temporizador watchdog programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software. El dispositivo opera entre 1.8 y 5.5 voltios. Por medio de la ejecución de poderosas instrucciones en un solo ciclo de reloj, el dispositivo alcanza una respuesta de 3 MIPS, balanceando consumo de energía y velocidad de proceso.

Un replazo alternativo común al ATmega328 es el ATmega328P (pico power) que consiste en un ahorro de energía que es tan solo una característica adicional del ATmega328, pero su funcionamiento es el mismo.

Hoy el ATmega328 se usa comúnmente en múltiple proyectos y sistemas autónomos donde se requiere un microcontrolador simple, de bajo consumo y bajo costo. Talvez la implementación más común de este chip es en la plataforma Arduino, en sus módulos uno y nano.

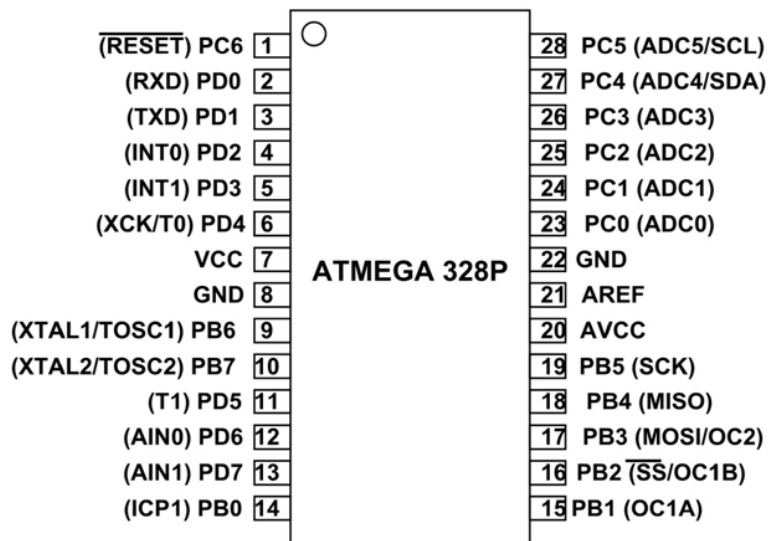


Figura 2.11. Microcontrolador ATMEGA328P

Fuente: <https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet>

A través del diagrama de pines podemos entender las configuraciones de los pines de cualquier dispositivo electrónico, por lo que está trabajando en cualquier proyecto de ingeniería y lo primero debes leer son los pines de los componentes.

VCC es un suministro de voltaje digital

AVCC es un pin de voltaje de suministro para convertidor analógico a digital. Que en todas las imágenes lo muestran como VCC, es el pin 20.

GND denota Ground y tiene 0V.

El **puerto A** consta de los pines de PA0 a PA7. Estos pines sirven como entrada analógica a convertidores analógico a digital, el puerto A actúa como un puerto de entrada / salida digital bidireccional de ocho (8) bits.

El **puerto B** consiste en los pines de PB0 a PB7. Este puerto es un puerto bidireccional de 8 bits que tiene una resistencia pull – up interno.

El **puerto C** consiste en los pines de PC0 a PC7. Los buffers de salida del puerto C tienen características de unidad simétricas con capacidad de fuente alta de corriente.

El **puerto D** consiste en los pines de PD0 a PD7. También es un puerto de entrada / salida de 8 bits que tiene una resistencia pull – up interno.

AREF es un pin de referencia analógico para convertidor analógico a digital.

Los pines de entrada analógicos soportan conversiones analógico – digital (ADC de 10 bit utilizando la función analogRead (). Las entradas analógicas pueden ser también usadas como pines digitales. Entrada analógica 0 como pin digital 14 hasta la entrada analógica 5 como pin digital 19. Las entradas analógicas 6 y 7 no pueden ser utilizadas como pines digitales.

2.6.2. Características del Microcontrolador Atmega 328p

El funcionamiento interno de un microcontrolador se puede explicar con un diagrama de bloques o esquema lógico, donde se ven en cada bloque cada unidad interna del microcontrolador y como se comunica con el resto de unidades.

Diagrama de bloques simplificado de un microcontrolador. Se compone de tres bloques fundamentales: la CPU (Central Processing Unit), memoria (RAM y ROM) y las entradas y salidas.

Los bloques se conectan entre sí mediante grupos de líneas eléctricas denominadas buses o pistas. Los buses pueden ser de direcciones (si transportan direcciones de memoria o entrada y salida), de datos (si transportan datos o instrucciones) o de control (si transportan señales de control diversas). La CPU es el cerebro central del microprocesador y actúa bajo control del programa almacenado en la memoria. La CPU se ocupa básicamente de traer las instrucciones del programa desde la memoria, interpretarlas y hacer que se ejecuten.

La CPU también incluye los circuitos para realizar operaciones aritméticas y lógicas elementales con los datos binarios, en la denominada Unidad Aritmética Lógica (ALU: Arithmetic logic Unit).

Diagrama de Bloques del Microcontrolador

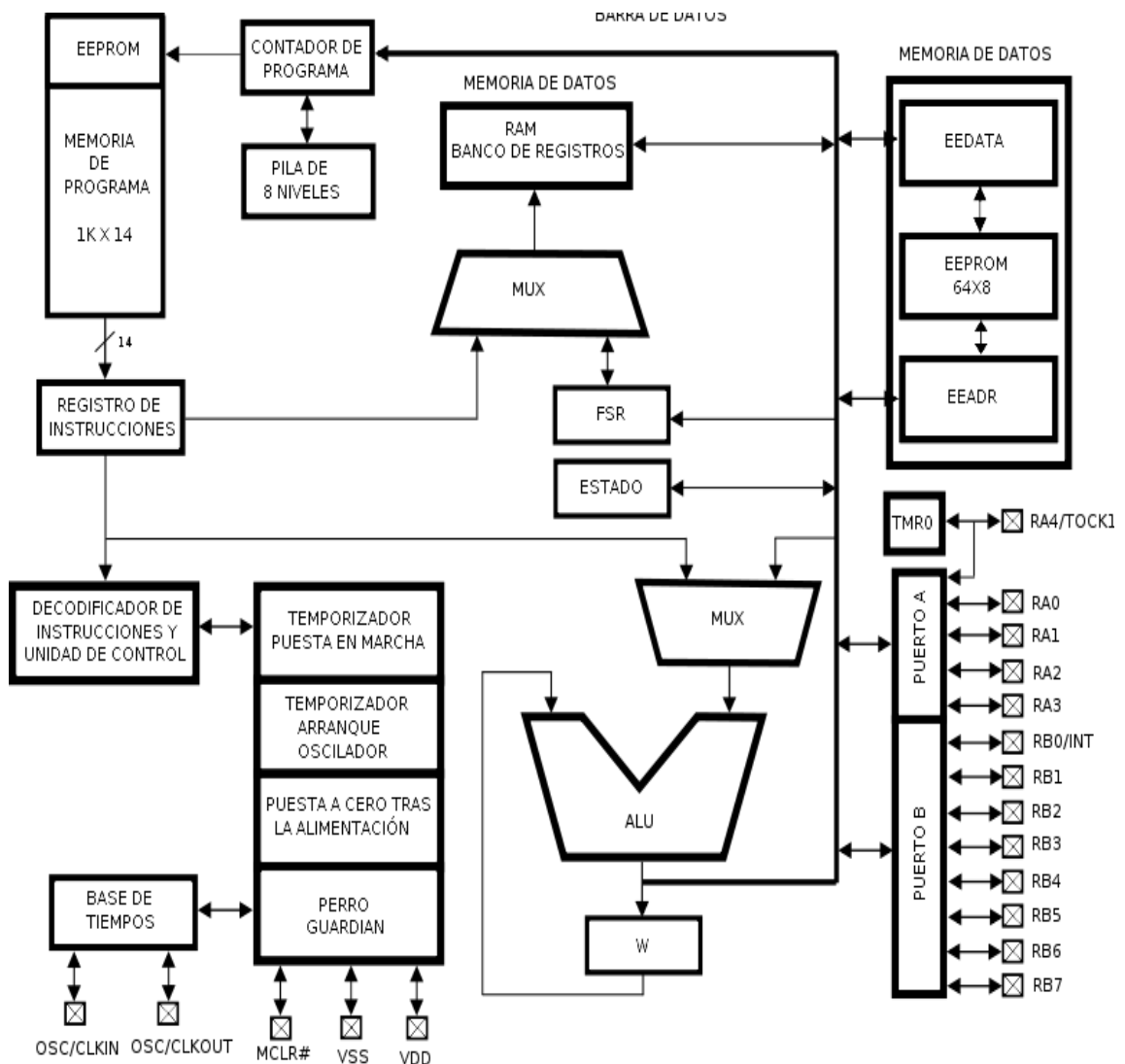


Figura 2.12 Diagrama de Bloques del Microcontrolador

Fuente: https://www.researchgate.net/figure/Figura-2-Diagrama-de-bloques-del-microcontrolador-PIC-16F88_fig2_45337900

El microcontrolador AVR de Atmel ATmega 328p

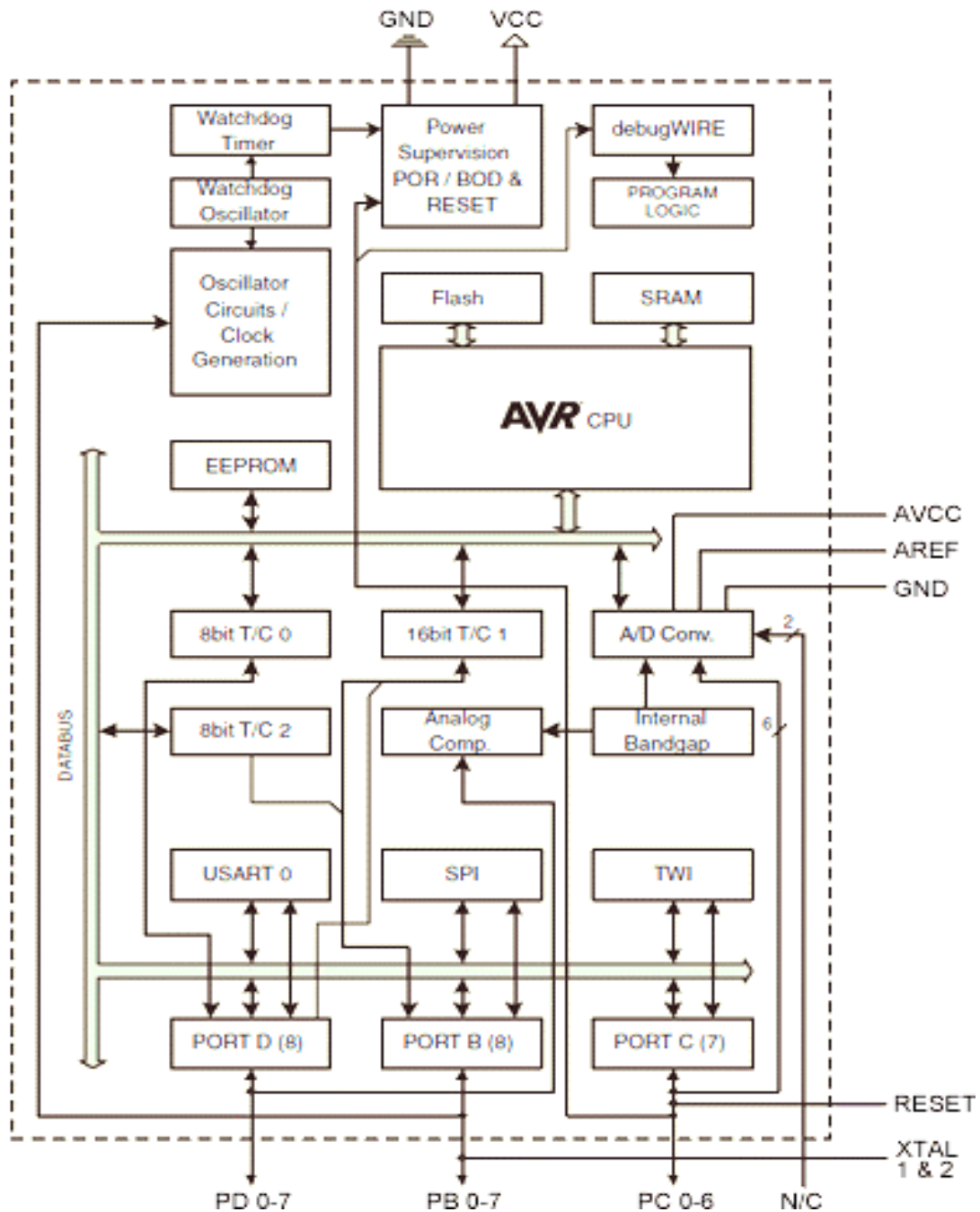


Figura 2.13 Diagrama interno ATmega 328p

Fuente: <http://electgpl.blogspot.com/2016/06/el-atmega328p.html>

El sistema de reloj determina la velocidad de trabajo del microcontrolador. Con 16 MHz se ejecuta una instrucción en 62,5 nanos segundos (1 / 16 MHz), correspondiente a 1 ciclo de máquina. El microcontrolador tiene diferentes opciones de circuito de reloj tal como muestra la imagen.

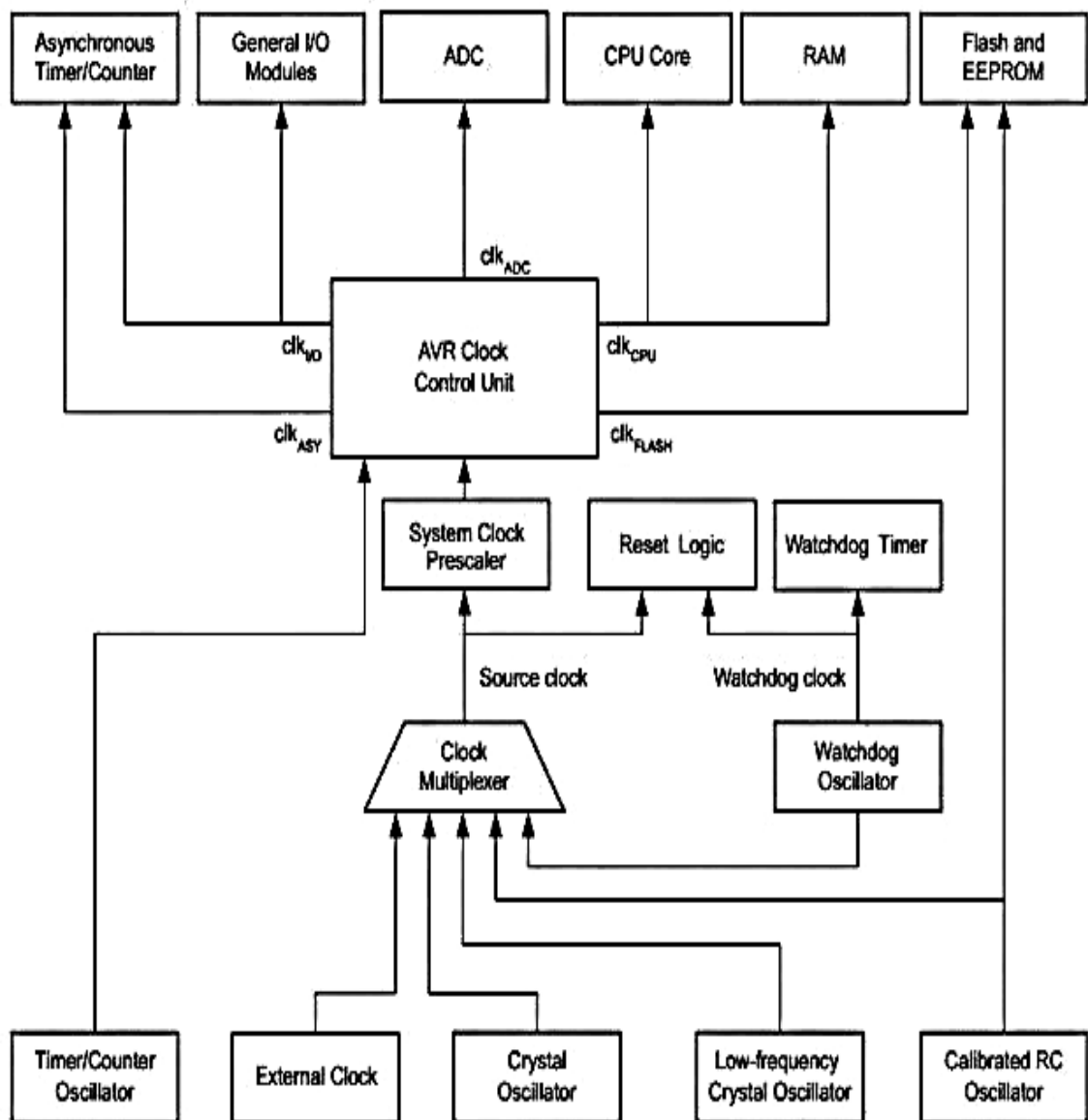


Figura 2.14 Unidad de Control de Reloj AVR

Fuente: <https://aprendiendoarduino.wordpress.com/tag/arquitectura-microcontrolador/>

Es un registro interno del microcontrolador se encuentran 5 opciones diferentes de reloj que son seleccionados por medio de un multiplexor. De este multiplexor sale la señal de reloj, la cual pasa a través de un prescaler, este prescaler se puede utilizar para reducir la frecuencia, reducir el consumo de energía y mejorar la estabilidad de la señal de reloj.

La señal de reloj es distribuida por la unidad de control a los diferentes bloques existentes: la CPU, las memorias, los módulos de entrada / salida, los contadores / timers, el SPI y la USART, al igual que el conversor Análogo Digital ADC.

El microcontrolador ATmega 328 tiene tres timers (timer 0, timer 1, timer2) que también se pueden usar como contadores. Los timers 0 y 2 son de 8 bits y el timer 1 de 16. Estos timers tienen un módulo de prescalado para su propia señal de reloj que puede provenir de su sistema de reloj interno o por pines externos (modo contador).

Son módulos que funcionan en paralelo a la CPU y de forma independiente a ella. El funcionamiento básico consiste en aumentar el valor del registro del contador al ritmo que marca su señal de reloj.

Usando el reloj interno o un cristal externo puede ser utilizado para medir tiempos puesto que utiliza una señal periódica, precisa y de frecuencia conocida; mientras que si la señal viene de un pin externo puede contar eventos que se produzcan en el exterior y que se reflejen en cambios de nivel de tensión de los pines.

Estos contadores también forman parte del generador de señales PWM y permiten configurar tanto la frecuencia como el ciclo de trabajo.

2.6.3. Registro de Memoria

Todos los microcontroladores tienen un conjunto de instrucciones que suele ser un conjunto pequeño al tratarse de arquitectura RISC. La CPU cuenta con ese número de instrucciones que sabe ejecutar.

En el caso del ATmega328p, tiene una arquitectura RISC con 131 instrucciones, la mayoría de ellas ejecutadas en un solo ciclo de reloj.

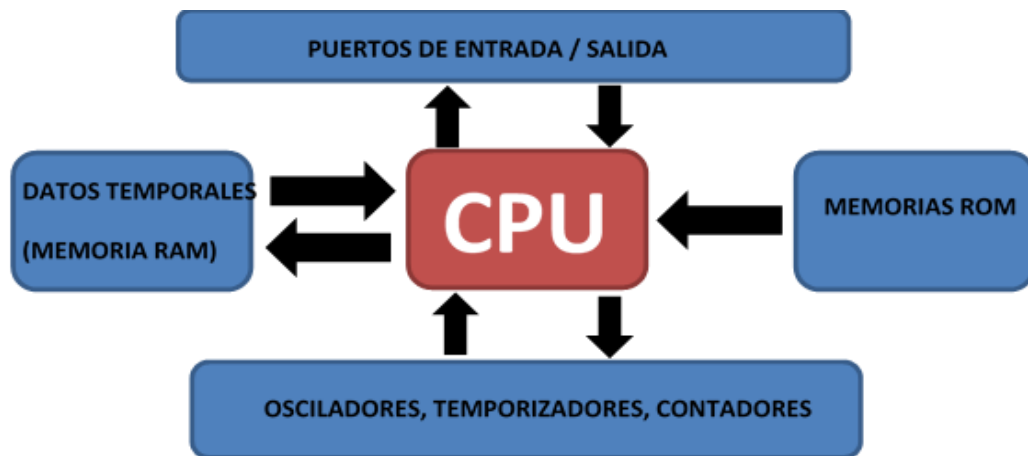


Figura 2.15 Diagrama en Bloques ATmega328p

Fuente: Proyecto de grado, Jorge luis Yavincha Balboa

Los registros son unas zonas concretas de la memoria RAM accesibles directamente desde la CPU o desde otros elementos del microcontrolador que permite hacer operaciones directamente y de forma más rápida.

Trabajar con registros de memoria puede ser difícil si sólo se escribe un programa en lenguaje ensamblador. Al utilizar el lenguaje de programación de alto nivel como es C basta con

escribir el nombre del registro y su dirección de memoria, a partir de esa información, el compilador selecciona el registro necesario. Las instrucciones apropiadas para la selección del registro serán incorporadas en el código durante el proceso de la compilación.

La memoria RAM en el ATmega328p se divide en varias partes, todos los grupos de registros se ponen a cero al apagar la fuente de alimentación. La SRAM del 328p se distribuye de la siguiente forma:

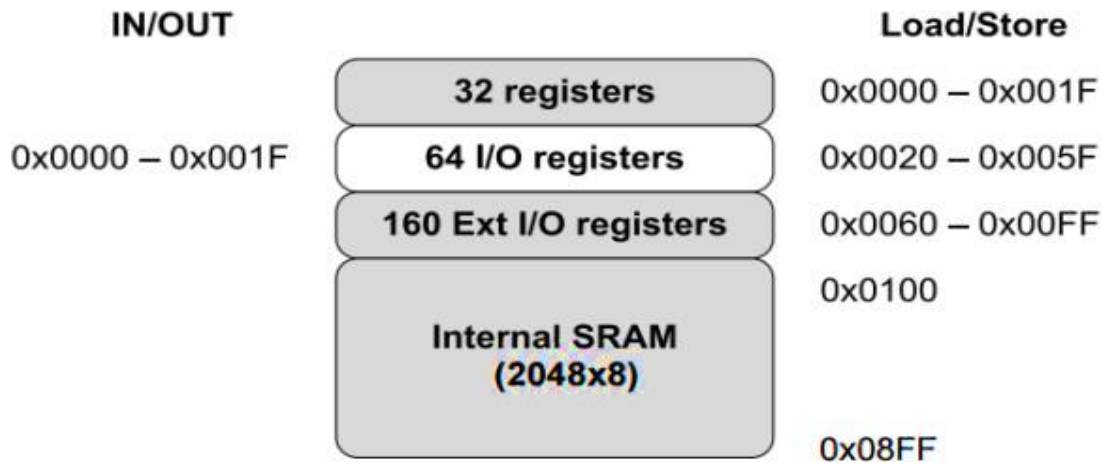


Figura 2.16 Grupos de Memoria RAM

Fuente: <https://es.slideshare.net/edurp14/grupos-4-memoria-ram-rom>

Las primeras 32 localizaciones de la memoria son el fichero de registros (Register File). Las siguientes 64 localizaciones de memoria es el standard I/O memory y después las 160 siguientes localizaciones son la Extended I/O memory. Por último, las siguientes 2K localizaciones son la memoria interna SRAM.

Las 5 diferentes modos de direccionamiento para los datos de memoria son:

- **Direct.** - El direccionamiento directo llega a todo el espacio de datos.
- **Indirect With Displacement.** - El modo indirecto con desplazamiento alcanza 63 direcciones.
- **Indirect.** - En el archivo de registros, los registros R26 a R31 presentan los puntos de registro de direccionamiento indirecto.
- **Indirect with Pre – decrement.** - El registro de dirección X, Y y Z son decrementados.
- **Indirect with Post – increment.** - El registro de dirección X, Y y Z son incrementados.

Los 32 registros de propósito general, los 64 I/O Registros, los 160 Extended I/O Registros y los 2K bytes de SRAM interna en el dispositivo son todos accesibles mediante estos modos de direccionamiento.

Los **registros de propósito general** se utilizan para almacenar los datos temporales y los resultados creados durante el funcionamiento de la ALU. Los 32 General Purpose Working Registers están directamente conectados a la ALU, permitiendo ser accedidos dos registros de forma independiente en una sola instrucción ejecutada en un ciclo de reloj. Los **I/O registros** localizados en las direcciones 0x20 a 0xFF, a diferencia de los registros de propósito general, su propósito es predeterminado durante el proceso de fabricación y no se pueden cambiar. Como los bits están conectados a los circuitos particulares en el chip (convertidor A/D, módulo de comunicación serial, etc.), cualquier cambio de su contenido afecta directamente al funcionamiento del microcontrolador o de alguno de los circuitos. Esta es la forma en que a bajo nivel se interactúa por ejemplo con los pines del microcontrolador.

2.6.4. Dispositivos Acoplables al Microcontrolador

Los dispositivos que pueden proporcionarnos algunas tareas para propósitos específicos en el uso de los microcontroladores ATMEGA328P, pueden ser muchos, los cuales estarán conectados mediante cables o directamente acoplados, muchos disponen de librerías que deberemos adjuntar al programa para poderlos usar.

Para utilizar los métodos de sensores y actuadores digitales debemos tener en cuenta que solo tenemos dos posibles valores, HIGH representa el nivel alto y LOW el nivel bajo.

En el caso de los analógicos su uso es levemente más complejo, pero también más configurables ya que tiene que leer y escribir un voltaje de 0 a 5 voltios que se representa en 10 bits (lectura) o en 8 bits (escritura), es decir la tensión puede tener 1024 (lectura) o 256 (escritura) y valores distintos.

2.6.5. Display LCD

El uso de las LCD se ha visto muy requerido tanto en la industria como en los proyectos escolares o de medianas empresas, ya que su uso es bastante agradable a la vista, los display LCD tiene una gama más abierta de aplicaciones, desde relojes, calculadoras, electrodomésticos, impresoras, etc.

Las siglas LCD significan pantalla de cristal líquido. Es una pantalla plana basada en el uso de una sustancia líquida atrapada entre dos placas de vidrio, haciendo pasar por este una corriente eléctrica a una zona específica, para que así esta se vuelva opaca, y además cuenta con iluminación trasera.

Las pantallas LCD de color, cada pixel individual se divide en tres celdas o sub píxeles con los colores RGB (Rojo, verde y Azul) respectivamente. Y así cada pixel puede controlarse para producir una gran variedad de colores distintos.

- El tamaño de un panel LCD generalmente se mide a lo largo de su diagonal, expresado generalmente en pulgadas. Sin embargo existen más características que pueden describir las dimensiones aproximadas, como por ejemplo la LCD 16x2 (negro sobre fondo azul) se refiere a que tiene la capacidad de tener el mismo tiempo 16 caracteres de manera horizontal en dos renglones.
- Esta se expresa con las dimensiones horizontal y vertical. Las pantallas HD tienen una resolución de 1920x1080. Y esta puede alcanzar con esta resolución una gran variedad de tamaño.
- La luminosidad de la pantalla también es importante analizarla, ya que según la aplicación en la que se encuentra esta, requerirá más luz para poder apreciarse, o viceversa. Por lo que la mayoría cuentan con una luz trasera y la posibilidad de poder controlar la luminosidad.
- **Iluminación CCFL** esta iluminación básicamente consta poner detrás de la pantalla una matriz de CCFL, o bien en las orillas o bordes de la pantalla. Sin embargo, es más consumo que el led y tiene un menor tiempo de vida, por lo que poco a poco se ha ido poniendo en segundo plano.



Figura 2.17 Iluminación CCFL

Fuente: <https://dspace.ups.edu.ec/bitstream/UPS-CT004404>

Iluminación LED Esta iluminación puede presentarse en dos maneras, en un solo color, (generalmente blanco) o bien en RGB, los blancos suelen



Figura 2.18 Iluminación Led

Fuente: <https://www.xatakahome.com/iluminacion-y-energia/que-es-la-iluminacion-led-especial-iluminacion-led>

- **Contraste** Es la relación entre la intensidad más brillante y la más oscura.

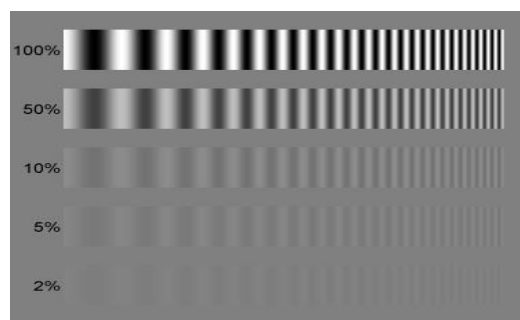


Figura 2.19 Contraste

Fuente: <https://visualled.com/glosario/contraste-pantallas/>

- **Angulo de Visión** Es el ángulo máximo en el que el usuario puede visualizar lo que está en la LCD sin que se pierda mucha calidad.
- **Numero de caracteres** Hay diversos tamaños de LCD y con ello nos limitamos o nos expandamos la posibilidad de mostrar en el display cierto número de caracteres, los tamaños estándar que manejamos son 16x2, 20x4, 8x2.

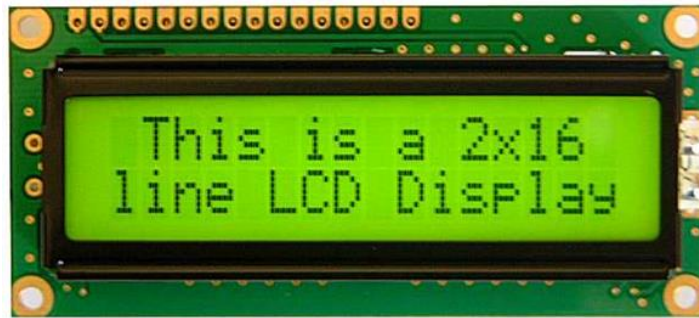


Figura 2.18 Display LCD 16x2

Fuente: https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=9

2.6.6. Comunicación LCD

Existen básicamente tres maneras de hacer comunicación con muestra LCD, aunque básicamente todas son en paralelo, cambian por el accesorio que se utiliza para entablar esta comunicación.

- **Comunicación en Paralelo** Es la más directa de utilizar, ya que se puede realizar haciendo directamente la comunicación con el ATmega 328p, esta puede utilizar del pin 0 al 7 o bien del 0 al 4, mandando la mitad de los datos, ya que, aun mandando la mitad de los datos, (haciendo doble proceso de envió) es tan rápido el registro, que no es posible percibirlo con el ojo humano.
- **Comunicación Serial a Paralelo** Para utilizar esta forma de comunicación se requiere un circuito integrado (como el PCF8574A) el cual se conecta por I²C o SPI al puerto SDA mientras que el reloj al SCL mediante una librería wire. Esta librería permite la comunicación con dispositivos I²C / TWI. Este se hace mediante un shift register.

- **Comunicación Serial UART** Este método funciona de manera serial utilizando el puerto TX y RX de un microcontrolador, mediante el uso de una herramienta llamada Back Pack, Sparkfun tiene dos, una que soporta de 16 o 20 caracteres y es para 2 o 4 líneas, mientras la otra es para gráficos entre 160x128 y 128x64. Esta puede ser utilizada con un baud entre 2400 y 38400 bps.

2.7. COMUNICACIÓN SERIAL

En la transmisión de datos en serie se utilizan pulsos binarios para transmitir los datos. El dígito binario uno está representado por cinco voltios o una lógica ALTA. Por el contrario, el cero binario se denota con una lógica BAJA o cero voltios. Para implementar la comunicación en serie, se requieren un origen y un destino. También se les conoce como emisor y receptor. Se pueden emplear varios tipos de comunicación serie y se designan como Simplex, Half Duplex y Full Duplex.¹²

Hay dos formas básicas de transferencia de datos. Estas son la comunicación en paralelo y en serie. La comunicación en serie conduce la transmisión de datos bit por bit.

Cuando envía datos, la endianidad de la transmisión es importante. En una transmisión de datos binarios de 8 bits de 11001110, ¿qué bit se envía primero? Podría ser el bit más significativo – MSB (7º bit) o el bit menos significativo – LSB (0º bit). El siguiente diagrama muestra una transmisión Little Endian donde el LSB se mueve primero.

2.7.1. Comunicación Serie

En esta representación, el transmisor envía un solo bit al receptor por cada pulso de reloj. En lugar de mover datos bit a bit, la comunicación en paralelo puede mover 8,16, o 32 bits de datos simultáneamente. Ejemplos de comunicación en paralelo son las impresoras y copiadoras que se benefician de la velocidad más rápida de transferencia de datos.

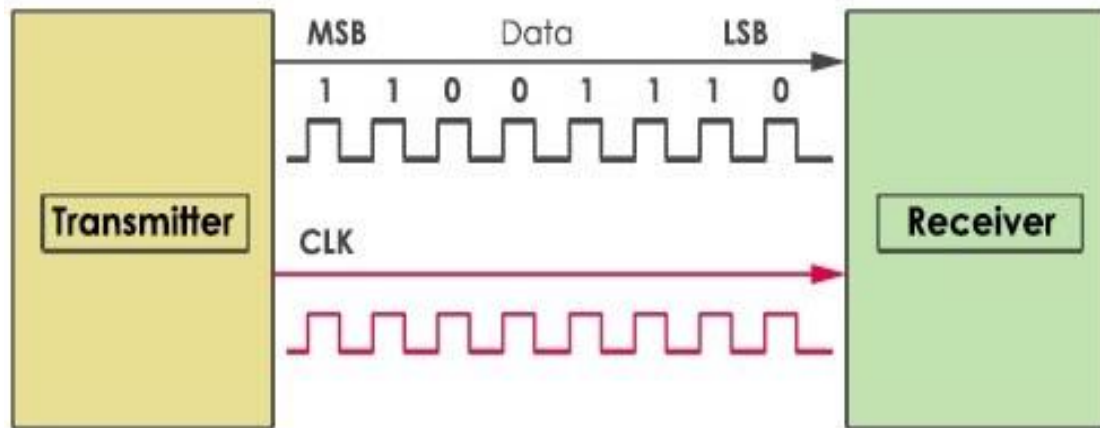


Figura 2.21 Comunicación Serial-Serie

Fuente: <https://www.serial-port-monitor.org/es/articles/serial-communication>

2.7.2. Comunicación Serial-Serie

La comunicación en serie envía un bit de datos a la vez. En consecuencia, se necesitan menos líneas de E/S (entrada-salida) para implementarlo que para la transmisión en paralelo.

Esto da como resultado menos interferencia y una reducción en los requisitos de espacio. También reduce el costo del sistema integrado y le permite transferir datos de manera confiable a largas distancias.

2.8. COMUNICACIÓN GPRS-GSM

La Asociación GSM (GSMA o *GSM Association*), dice que GSM es el estándar en telecomunicaciones móviles más extendido en el mundo, con un 82% de los terminales mundiales en uso.¹⁰ GSM cuenta con más de 3000 millones de usuarios en 159 países distintos, siendo el estándar predominante en Europa, América del Sur, Asia y Oceanía, y con gran extensión en América del Norte.

¹⁰ Asociación GSM World, estadística de junio 2008

La ubicuidad del estándar GSM ha sido una ventaja tanto para consumidores (beneficiados por la capacidad de itinerancia y la facilidad de cambio de operador sin cambiar de terminal, simplemente cambiando la tarjeta SIM) como para los operadores de red (que pueden elegir entre múltiples proveedores de sistemas GSM, al ser un estándar abierto que no necesita pago de licencias).

El GSM se implementó por primera vez el servicio de mensajes cortos de texto (SMS), que posteriormente fue extendido a otros estándares. Además, en GSM se define un único número de emergencias a nivel mundial, el 112, que facilita que los viajeros de cualquier parte del mundo puedan comunicar situaciones de emergencia sin necesidad de conocer un número local.¹¹

2.8.1. Frecuencias

La interfaz de radio de GSM se ha implementado en diferentes bandas de frecuencia.

Banda	Nombre	Canales	Uplink (MHz)	Downlink (MHz)	Notas
GSM 850	GSM 850	128–251	824,0–849,0	869,0–894,0	Usada en Estados Unidos, Sudamérica y Asia
GSM 900	P-GSM 900	0–124	890,0–915,0	935,0– 960,0	La banda con que nació GSM en Europa y la más extendida
	E-GSM 900	974–1023	880,0–890,0	925,0– 935,0	<i>E-GSM</i> , extensión de GSM 900
	R-GSM 900	n/a	876,0–880,0	921,0–925,0	<i>GSM ferroviario</i> (GSM-R).

¹¹ www.gsmlaa.org

GSM1800	GSM 1800	512– 885	1710,0–1785,0	1805,0– 1880,0	
GSM1900	GSM 1900	512–810	1850,0–1910,0	1930,0– 1990,0	Usada en Norteamérica, incompatible con GSM-1800 por solapamiento de bandas.

Tabla 2.3 Banda de Frecuencias GSM

Fuente: www.gsmlaa.org

2.9. SERVOMOTORES

Normalmente los motores habituales lo que hacen es transformar la energía eléctrica en un giro continuo que podemos usar para desarrollar trabajo mecánico.

En la sesión 13 utilizamos un pequeño motor de la corriente continua y regulamos la velocidad de giro mediante una señal PWM.

Los servos son también motores de corriente continua, pero en lugar de diseñarse para obtener un giro continuo que podemos aprovechar, se diseñan para que se muevan un ángulo fijo en respuesta a una señal de control, y se mantengan fijos en esa posición.

Estos servos o servomotores son muy frecuentes en Aero modelismo y en robótica, por la capacidad que presentan para moverse a un ángulo concreto y mantenerse allí. De hecho, se suelen diseñar para que giren un ángulo proporcional a una señal PWM, de forma que su control es muy preciso.



Figura 2.22 Servomotor

Fuente: <https://www.aranacorp.com/es/controla-un-servo-con-arduino/>

Un servo suele estar formado por un circuito de control reductor, un motor de CC, un circuito de control. Habitualmente los servomotores tienen un margen de operación, es decir, pueden moverse entre 0° y ángulo dado, que suele ser de 180°, pero existen modelos comerciales de todas las características.

Estos servos funcionan sobre 5V y el control se realiza mediante una señal de control PWM, en la que el ancho el pulso indica el ángulo que deseamos adopte el eje.

Un servo tiene un conector de 3 hilos, 5V (rojo), GND (negro o marrón) y el otro control (amarillo o blanco).

Podríamos escribir un pequeño programa en Atmega328p que controle estos pulsos y los relacione con el ángulo buscado, los servos son tan habituales, que ya ha habido gente que han escrito estos programas y los han puesto a nuestra disposición.

2.10. MOTOR PASO A PASO

El motor paso a paso es un motor de corriente continua sin escobillas en el que la rotación se divide en un cierto número de pasos resultantes de la estructura del motor. Normalmente, una revolución completa del eje de 360° se divide en 200 pasos, lo que significa que se realiza una sola carrera del eje cada 1,8°. También hay motores donde la carrera del eje se realiza cada 2; 2,5; 5, 15 o 30°. La funcionalidad descrita es posible gracias a la construcción especial del **motor paso a paso**, que se discutirá más adelante en el texto.

Debido al hecho de que la rotación completa del eje se divide en ciertas secciones discretas, el motor paso a paso no gira suavemente, sino que realiza saltos y cruza estados intermedios, por lo que el funcionamiento del motor paso a paso se acompaña de un sonido y vibración característicos.

Actualmente los controladores de motores de paso a paso se realizan sobre la base de circuitos digitales que controlan los drivers aumentando la capacidad de carga de las salidas.

Por lo general, en un controlador de este tipo encontramos un microcontrolador, pero no necesariamente, porque dicho controlador se puede hacer basado en puertas y flip-flops sin mucha dificultad. El propio método de control del motor depende de su tipo, el número de fases y si es con o sin retroalimentación. En algunos controladores, la corriente que fluye a través de los devanados se puede ajustar usando la forma de onda PWM, mientras que la dirección de rotación y el control de los pasos (saltos) se llevan a cabo usando formas de onda cuadradas. Sin embargo, cuando se trata de un modelo de motor específico y su controlador, para determinar el método de control correcto, uno debe consultar sus fichas de catálogo.

Los devanados que son la carga en las salidas del controlador son bobinas con cierta inductancia y capacitancia. Su reactancia aumenta al aumentar la frecuencia, lo que limita la corriente que fluye y limita la frecuencia de conmutación máxima. Al seleccionar un motor para una aplicación específica, siempre hay algunas compensaciones a considerar, como, por ejemplo, la velocidad angular máxima y el par requerido con respecto a la resolución.

El artículo proporciona información básica sobre el principio de funcionamiento de los motores paso a paso bipolares y unipolares y a lo que debe prestar atención al elegir un motor para una aplicación determinada.

2.10.1. Como funciona un Motor paso a paso

El motor paso a paso está compuesto por rotor y estator. El estator es una parte estacionaria, mientras que el rotor montado en el eje con un cojinete gira siguiendo el campo magnético giratorio creado alrededor del estator. El estator, hecho de acero u otro metal, es el marco de un conjunto de electroimanes, que son bobinas montadas en lugares específicos alrededor del rotor. Cuando la corriente fluye a través de las bobinas del estator, se crea un campo magnético a su alrededor. Los flujos magnéticos particulares tienen una dirección e intensidad que dependen de la intensidad y la dirección de la corriente que fluye a través de una bobina determinada.

Cuando la bobina se energiza, se forma un electroimán atrayendo a un imán (diente) montado en el rotor, desplazado por un cierto desplazamiento con respecto a él. Luego, el rotor y el eje giran en el ángulo en el que su posición se opone menos al flujo magnético o la resultante de múltiples flujos. Después de pasar por este desplazamiento, se enciende otro electroimán (bobina o bobinas) en el estator y se tira del rotor a su nueva posición. Al conmutar bobinas sucesivas, es posible realizar más saltos hacia adelante o hacia atrás, o completar o girar parcialmente el rotor y el eje con él. Con base en la descripción dada, es posible imaginar un motor paso a paso como una serie de electroimanes que atraen el imán del rotor. En realidad, sin embargo, la situación es mucho más compleja ya que el imán es atraído por el campo resultante alrededor del conjunto del electroimán, lo que permite no solo la operación de paso completo, sino también la operación de medio paso (división de la carrera por 2) o menos, lo que se denomina operación de micro paso.¹²

2.10.2. Motor paso a paso – trabajo en modo de paso completo

El principio de funcionamiento del **motor paso a paso** en el modo de paso completo se ilustra en el dibujo 1. En este modo, el motor gira en un ángulo resultante de su construcción, que puede ser, por ejemplo, $1,8^\circ$. Como es fácil de calcular, en tal caso es necesario dar 200 pasos para un giro completo ($200 \times 1,8^\circ = 360^\circ$).

El recorrido del eje se realiza después de energizar una o dos bobinas. El funcionamiento de una sola bobina requiere una potencia mínima del controlador. En funcionamiento bifásico, con suministro de bobinas opuestas, se requiere el doble de potencia, pero también aumentan la velocidad y el par.

¹² <https://www.tme.eu/es/news/library-articles/page/41861/Motor-paso-a-paso-tipos-y-ejemplos-del-uso-de-motores-paso-a-paso/#:~:text=El%20motor%20paso%20a%20paso,eje%20cada%201%2C8%C2%B0.>

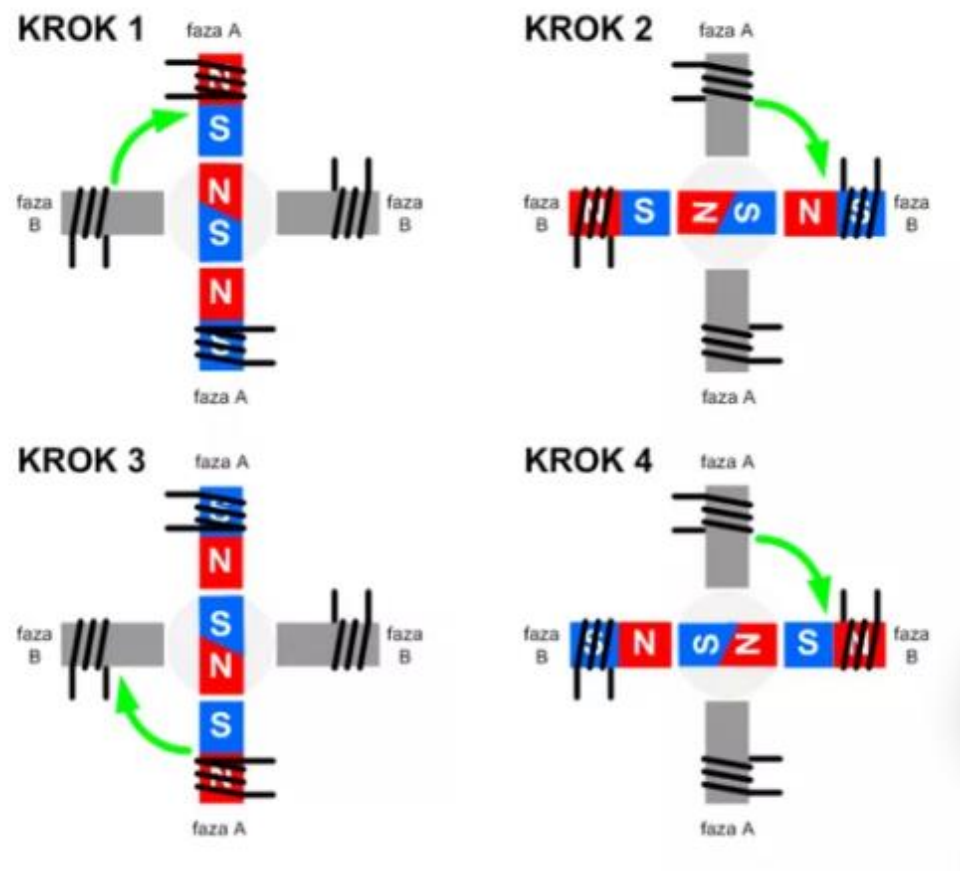


Figura 2.23 motor paso a paso

Fuente: <https://www.tme.eu/es/news/library-articles/page/41861/Motor-paso-a-paso-tipos-y-ejemplos-del-uso-de-motores-paso-a>

2.10.3. Tipos de motores paso a paso

El motor paso a paso, en comparación con un motor de CC con escobillas típico, no es particularmente complicado y de construcción más compleja, pero requiere más precisión. Los motores BLDC modernos son muy similares a los motores paso a paso de imanes permanentes, que además se controlan de manera muy similar.

Según el criterio básico, los tipos de motores paso a paso se dividen según la forma en que están contruidos y el número de fases necesarias para alimentar las bobinas. Dependiendo del diseño, los tipos individuales difieren en su propósito (aplicación de destino), resolución y par alcanzado.

2.10.4. Motor paso a paso de imán permanente

Un motor de imán permanente podría tener dos bobinas (electroimanes) formando los cuatro polos variables, y el rotor del motor estaría magnetizado radialmente. Cambiar la posición del rotor cambiaría la dirección del flujo de corriente en las bobinas, lo que cambiaría los polos magnéticos. Si la dirección de la corriente se cambiara en consecuencia, el rotor de dicho motor giraría 90° . Un solo paso de dicho motor, aunque podría ser útil en algunas aplicaciones, sería muy grande y no muy precisa. Por lo tanto, de hecho, los motores de imanes permanentes tienen más polos de rotor y, para aumentar el número de pasos y la precisión de posicionamiento, se montan varios imanes en su rotor.

Normalmente, los motores paso a paso de imán permanente tienen un paso de $7,5^\circ$ a 15° , dando de 48 a 24 carreras por revolución completa. Los polos de rotor magnetizados aumentan la inducción magnética, por lo que los motores de imanes permanentes se caracterizan por un par elevado. La construcción sencilla da como resultado un precio moderado del motor con una resolución bastante baja.

2.10.5. Motor paso a paso de reluctancia variable

Los motores de reluctancia variable estuvieron entre los primeros modelos de motores paso a paso. Hoy en día son bastante raros y usados.

En este tipo de motor, el rotor consta de varios dientes hechos de hierro dulce. Cuando las bobinas del estator son alimentadas por una corriente continua, el diente del rotor es atraído por el campo magnético. Gracias a la conmutación secuencial, el rotor gira en el ángulo determinado por la estructura del motor.

2.10.6. Motor paso a paso híbrido

El motor híbrido es uno de los tipos de motores paso a paso más utilizados en la industria. Se caracteriza por una alta resolución: su rotor realiza saltos de $0,9^\circ$ a $3,6^\circ$ (de 400 a 100 golpes). Este tipo de motor supera al resto en cuanto a fiabilidad, par, par de retención y velocidad alcanzada.

El rotor del motor híbrido está compuesto de imanes permanentes, pero a diferencia de los motores de imanes permanentes discutidos anteriormente, no están montados radialmente sino magnetizados axialmente. Normalmente, el rotor consta de dos anillos magnetizados de forma diferente colocados en el eje del motor. Cada anillo tiene ranuras que forman los dientes del rotor.

2.11. MOTORES UNIPOLARES Y BIPOLARES

Otra división de los motores paso a paso resulta del método de hacer el devanado en motores de 2 fases. Dependiendo de él, los motores se dividen en unipolares y bipolares. La principal diferencia es que el motor unipolar funciona con una polaridad de corriente (voltaje), mientras que el motor bipolar funciona con dos polaridades, lo que significa que la dirección del flujo de corriente en la bobina es variable. Otra diferencia es que las bobinas del motor deben estar conectadas de tal manera que sea posible transferir energía desde el final de una bobina al principio de la otra. Este método de conexión permite el uso de corriente (voltaje) de una polaridad. Las diferencias en la construcción de ambos tipos de motores se ilustran de forma simplificada en los dibujos 3 y 4.

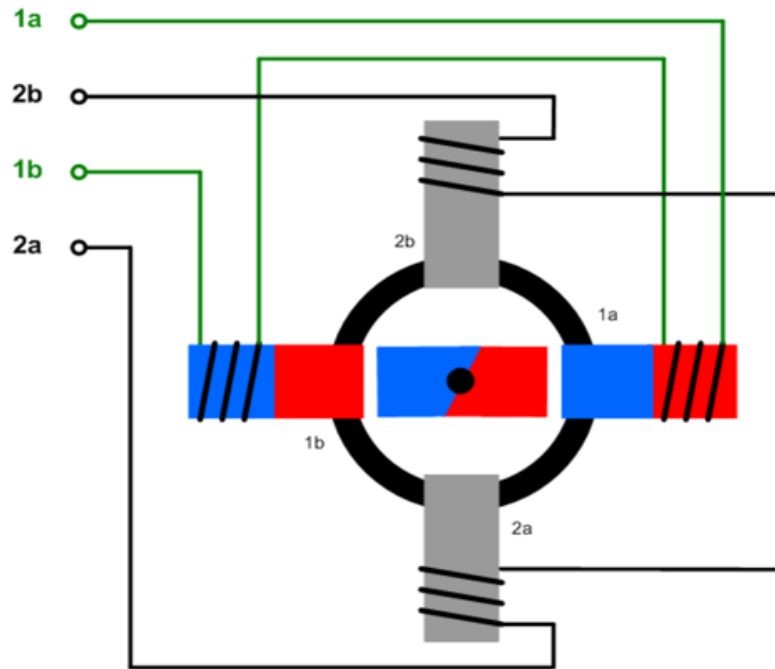


Figura 2. 24 motor bipolar y el método de conectar sus devanados.

Fuente: <https://dSPACE.UPS.EDU.EC/bitstream/123456789/19384/1/UPS-CT008857.pdf>

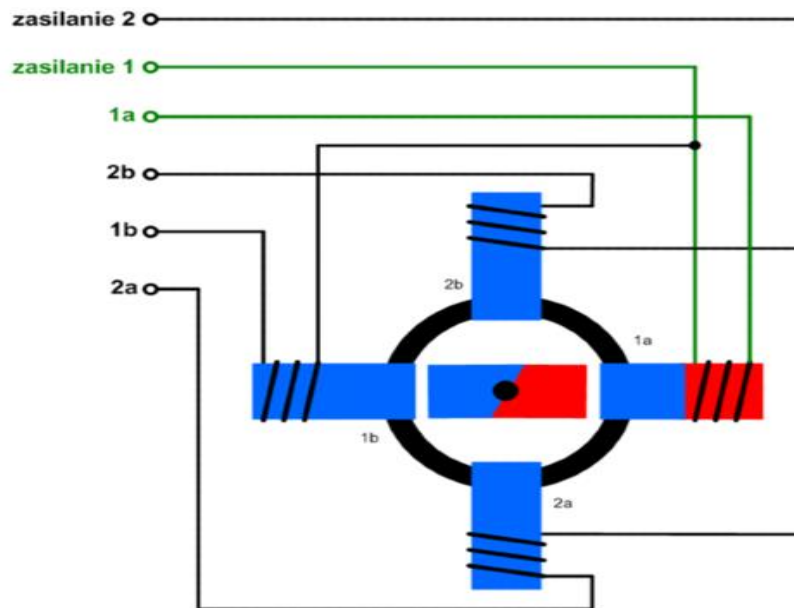


Figura 2.25 Dibujo conceptual de un motor unipolar

Fuente: <https://epyelectronica.com/catalogo-epy-electronica/>

Un motor bipolar tiene más par que un motor unipolar, aunque esto se produce a expensas de controles más complejos.

2.11.1. Control de motor paso a paso

Es una buena práctica construir accionamientos para tener en cuenta todo el conjunto, es decir, el motor con el controlador, porque es imposible considerar las propiedades de todo el sistema de accionamiento sin tener en cuenta los parámetros de ambos. En pocas palabras, se puede decir que incluso el mejor motor no funcionará correctamente sin el controlador correcto y viceversa. La selección del motor + unidad de control tendrá un impacto significativo en las características del sistema de transmisión.

El desarrollo principal de los motores paso a paso es reducir la inercia del motor al tiempo que aumenta la resolución (número de pasos), el par y la eficiencia energética. Por lo tanto, en la práctica, además de los tipos de motores antes mencionados, puede encontrar muchas modificaciones destinadas a mejorar los parámetros mencionados anteriormente. Los motores particulares, según la aplicación, también difieren en el número de devanados y, por lo tanto, también en el algoritmo de control.

En Internet puede encontrar muchos diseños de controladores de motor paso a paso, desde los más fáciles de construir, equipados con un potenciómetro para regular la velocidad de rotación y un botón para cambiar la dirección de rotación o un botón de inicio/parada, construido con el uso de componentes discretos, como puertas, flip-flops y llaves de transistores, hasta muy complicado, por ejemplo, realizado sobre la base de circuitos integrados especializados de controladores y un procesador DSP. Sin embargo, parece que dichos proyectos serán más útiles para la construcción de dispositivos no profesionales o experimentales que, por ejemplo, para su uso en la industria. En este caso, buscaremos soluciones listas para usar de fabricantes probados.

En el uso no profesional, la forma más fácil es construir un controlador de motor basado en Arduino y un amplificador de corriente o controlador de motor apropiado. La elección de la placa de expansión dependerá del motor utilizado.

En un motor bipolar, la corriente debe fluir en dos direcciones. Por lo tanto, para cambiar la dirección del flujo magnético en el núcleo, un solo interruptor de palanca que consta de conmutados alternativamente transistores (diseño de medio puente). Al controlar un motor unipolar, el flujo de corriente en un devanado determinado debe asegurarse solo en una dirección, por lo que es suficiente usar una sola llave de transistor por bobina. Como puede ver, el control unipolar requiere menos transistores clave, pero como recordamos de la lectura anterior, solo la mitad del devanado está funcionando en un momento dado, por lo que el motor unipolar en sí mismo produce menos torque que el motor bipolar. Por lo tanto, un motor bipolar requiere un sistema de control más extenso, pero muchos fabricantes de semiconductores ofrecen circuitos integrados especializados que incluyen dos puentes de transistores completos.

Circuitos de protección contra sobrecalentamiento, sobre corriente y sobretensión y puertas lógicas para simplificar el control del motor. Estos circuitos integrados se pueden encontrar en la cartera de STM, Toshiba Electric y otros.¹³

2.12. EL SENSOR ULTRASONICO

El sensor HC-SR04 es un sensor de distancia de bajo costo que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. El sensor HC-SR04 es el más utilizado dentro de los sensores de tipo ultrasonido, principalmente por la cantidad de información y proyectos disponibles en la web.

El funcionamiento del sensor es el siguiente: el emisor piezoeléctrico emite 8 pulsos de ultrasonido(40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y así se puede calcular la distancia al objeto.

¹³ <https://www.tme.eu/es/news/library-articles/page/41861/Motor-paso-a-paso-tipos-y-ejemplos-del-uso-de-motores-paso-a-paso/#:~:text=El%20motor%20paso%20a%20paso,eje%20cada%201%2C8%C2%B0>.

El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar).

La distancia se puede calcular utilizando la siguiente fórmula:

$$\text{Distancia(m)} = \{(\text{Tiempo del pulso ECO}) * (\text{Velocidad del sonido}=340\text{m/s})\} / 2$$

El sensor US-016 es similar al HC-SR04 pero con salida de tipo analógico, otro sensor ultrasonido es el sensor US-100 con salida de tipo uart/serial.

2.12.1. Especificaciones Técnicas

- Voltaje de Operación: 5V DC
- Corriente de reposo: < 2mA
- Corriente de trabajo: 15mA
- Rango de medición: 2cm a 450cm
- Precisión: +- 3mm
- Angulo de apertura: 15°
- Frecuencia de ultrasonido: 40KHz
- Duración mínima del pulso de disparo TRIG (nivel TTL): 10 µS
- Duración del pulso ECO de salida (nivel TTL): 100-25000 µS
- Dimensiones: 45*20*15 mm
- Tiempo mínimo de espera entre una medida y el inicio de otra 20ms (recomendable 50ms)



Figura 2.26 Modulo Ultrasónico

Fuente: <https://epyelectronica.com/catalogo-epy-electronica/>

2.13. PROGRAMACIÓN DE MICROCONTROLADORES

Programar consiste en traducir a líneas de código las tareas automatizadas que queremos hacer leyendo los sensores y en función de las condiciones del entorno programar la interacción con el mundo exterior mediante unos actuadores.

Arduino proporciona un entorno de programación sencillo y potente para programar, pero además incluye las herramientas necesarias para compilar el programa en la memoria flash del microcontrolador. Además, el IDE nos ofrece un sistema de gestión de librerías y placas muy práctico. Como IDE es un software sencillo que carece de funciones avanzadas típicas de otros IDEs, pero suficiente para programar.

2.13.1. Estructura de un Sketch

Un programa de Arduino se denomina sketch o proyecto y tiene extensión (.ino), para que funcione el sketch, el nombre del fichero debe estar en un directorio con el mismo nombre que el sketch.

No es necesario que un sketch este en un único fichero, pero si es imprescindible que todos los ficheros estén dentro del mismo directorio que el fichero principal.

```

1  void setup () {
2  // put your setup code here, to run once:
3  }
4
5  void loop () {
6  // put your main code here, to run repeatedly:
7  }

```

Figura 2.27 Estructura Principal de Programa

Fuente: http://dfists.ua.es/~jpomares/arduino/page_04.htm

La estructura básica de un sketch de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes son obligatorias y encierran bloques que contienen declaraciones, estamentos o instrucciones. Ambas funciones para que el programa trabaje.

- **Setup ()** Es la parte encargada de recoger la configuración.
- **Loop ()** Es la que contiene el programa que se ejecuta cíclicamente.

2.13.2. Lenguaje de Programación Arduino

El lenguaje de programación de Arduino es C++. No es un C++ puro, sino que es una adaptación que proviene de AVR – libc que provee de una librería de C de alta calidad para usar con GCC (compilador de C y C++) en los microcontroladores AVR de Atmel y muchas utilidades específicas para las MCU AVR de Atmel como avrdude.

Aunque se hable que hay un lenguaje propio de programación de Arduino, no es cierto, la programación se hace en C++, pero Arduino ofrece una Core que facilitan la programación de los pines de entrada y salida y de los puertos de comunicación, así como otras librerías para operaciones específicas. El propio IDE ya incluye estas librerías de forma automática y no es necesario declararlas expresamente. Otra diferencia frente a C++ standard es la estructura del programa que ya hemos visto anteriormente.

- **Librerías** Las librerías son trozos de código hechos por terceros que usamos en nuestro sketch. Esto nos facilita mucho la programación y hace nuestro programa sea más sencillo de hacer o modificar una librería, pero en este curso debemos ser capaces de buscar una librería, instalarla, aprender a usar cualquier librería y usarla en un sketch. Las librerías normalmente incluyen los siguientes archivos comprimidos en un archivo ZIP o dentro de un directorio. Estas siempre contienen:

- Un archivo .cpp (código de C++)
- Un archivo .h o encabezado de C, que contiene las propiedades y métodos o funciones de la librería.
- Un archivo Keywords.txt, que contiene las palabras clave que se resaltan en el IDE.
- Muy posiblemente la librería incluye un archivo readme con información adicional de lo que hace y con instrucciones de cómo usarla.
- Directorio denominado examples con varios sketches de ejemplo que nos ayudara a entender cómo usar la librería.

La instalación de librerías se puede hacer directamente desde el gestor de librerías o manualmente.

Además de estas librerías oficiales que podemos obtener desde el gestor de librerías y las publicadas en el playground de Arduino, existen muchos desarrollos disponibles en Internet y los propios fabricantes de componentes HW publican las librerías para poder usarlo o incluso la propia comunidad de usuarios.

2.14. ANDROID

Android es un sistema operativo móvil desarrollado por Google, basado en kernel de Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores. Inicialmente fue desarrollado por Android.inc, empresa de Google respaldado económicamente y que adquirió en 2005. Android fue presentado en 2007 junto con la fundación del Open Handset Alliance (un consorcio de compañías de hardware y software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El código fuente principal de Android se conoce como Android Open Source Project (AOSP), que se licencia principalmente bajo la licencia Apache. Android es el sistema operativo más utilizado del mundo, con una cuota de mercado superior al 80% al año 2017, muy por encima de IOS. Según los documentos secretos filtrados en 2013 y 2014, el sistema operativo es uno de los objetivos de las agencias de inteligencia internacionales.

2.14.1. Arquitectura Android

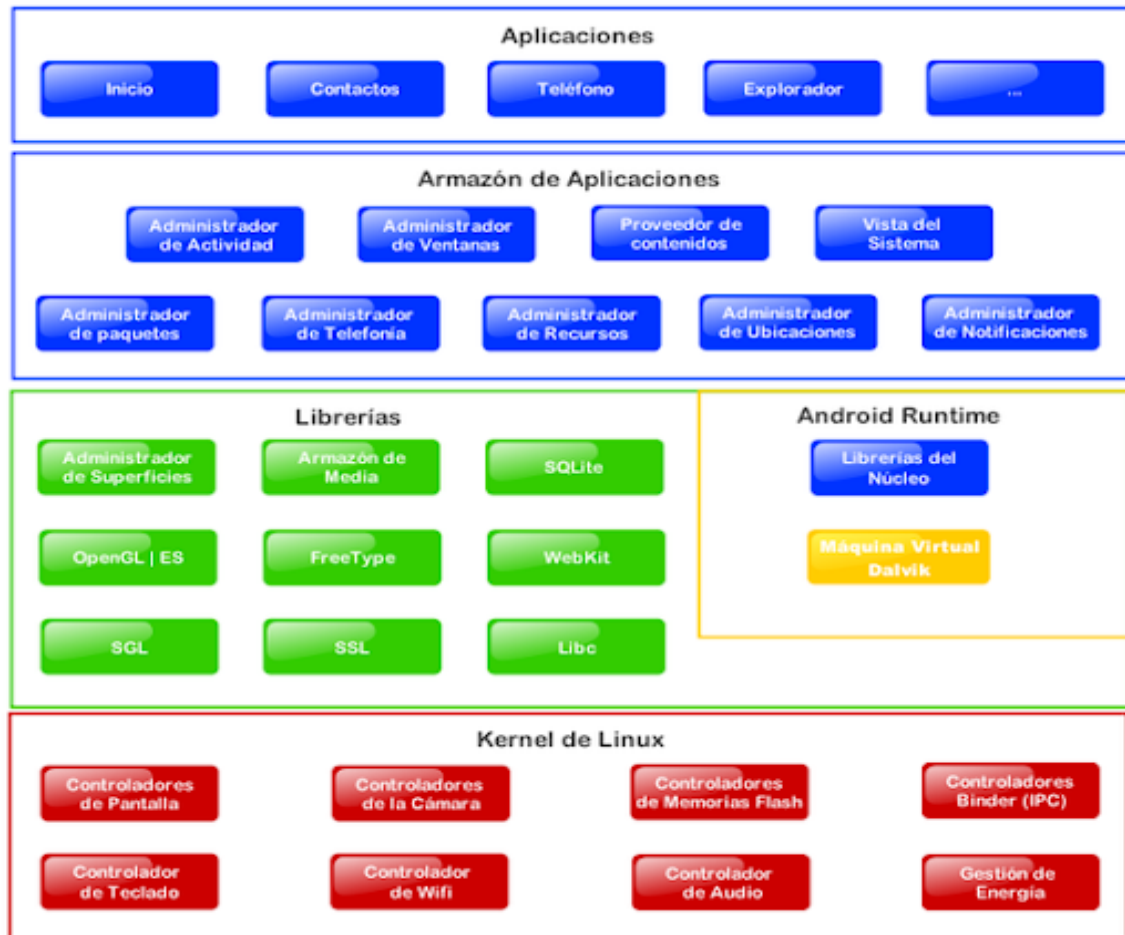


Figura 2.28 Arquitectura Principal de Android

Fuente: <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

Los componentes principales del sistema operativo de Android son:

- **Aplicaciones** Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contacto y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

- **Marco de Trabajo de Aplicaciones** Los desarrolladores tienen acceso completo a los mismos API del entorno de trabajo usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes, cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android. Algunas son: System C library (Implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráfico, 3D y SQLite, entre otras.

2.14.2. Versiones de Android

Las versiones de Android recibían hasta la versión 9, en inglés, el nombre de diferentes postres o dulces. En cada versión el postre o dulce elegido empieza por una letra distinta, conforme a un orden alfabético:

Nombre Código	Numero de Versión	Fecha de Lanzamiento	Nivel de API
Apple Pie	1.0	23 /09/2008	1
Banana Bread	1.1	9/02/2009	2
Cup cake	1.5	25/04/2009	3
Donut	1.6	15/09/2009	4
Eclair	2.0 – 2.1	26/10/2009	5 – 7
Froyo	2.2 – 2.2.3	20/05/2010	8
Gingrebread	2.3 – 2.3.7	6/12/2010	9 – 10
Honeycomb	3.0 – 3.2.6	22/02/2011	11 – 13

Ice Cream Sándwich	4.0 – 4.0.5	18/10/2011	14 – 15
Jelly Bean	4.1 – 4.3.1	9/07/2012	16 – 18
KitKat	4.4 – 4.4.4	31/10/2013	19 – 20
Lollipop	5.0 – 5.1.1	12/11/2014	21 – 22
Marshmallow	6.0 – 6.0.1	5/10/2015	23
Nougat	7.0 – 7.1.2	15/07/2016	24 – 25
Oreo	8.0 – 8.1	21/08/2017	16 – 27
Pie	9.0	6/08/2018	28
Android 10	10.0	3/09/2019	29

Tabla 2.4 Versiones y Lanzamientos de Android

Fuente: <https://andro4all.com/guias/android/versiones-android-historia>

2.14.3. Usos y Dispositivo

El sistema operativo Android se usa en teléfonos inteligentes, ordenadores portátiles, tabletas, Google TV, relojes de pulsera, auriculares, receptores decodificadores integrados y otros dispositivos.

La plataforma de hardware principal de Android es la arquitectura ARM. Hay soporte para X86 en el proyecto Android X86, y Google TV utiliza una versión especial de Android x86.

A principios de 2010 Google ha colaborado con HTC para lanzar su producto estrella en dispositivos Android, el Nexus One. Google ha continuado la comercialización de la gama Nexus en 2010 con el Samsung Nexus S, en 2011 con el Galaxy Nexus en 2012 con el Nexus 4 y las tabletas Nexus 7 y Nexus 10 en 2014 con el Nexus 6 y la tableta Nexus 9. Estos dispositivos son utilizados para el desarrollo e implementación de Android, siendo los dispositivos que estrenan las nuevas versiones disponibles. En la actualidad existen aproximadamente 1,000,000 de aplicaciones para Android y se estima que 1,500,000 teléfonos móviles se activen diariamente, y en 2013 se llegó a los 1000 millones de teléfonos inteligentes Android en el mundo.

2.14.4. Aplicaciones

Las aplicaciones se desarrollan habitualmente en el lenguaje JAVA con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un kit de Desarrollo Nativo para Aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varios marcos de aplicaciones basadas en la web multitelefono. También es posibles usar las bibliotecas Qt gracias al proyecto Necesitas SDK.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de JAVA y estar en posesión de kit de desarrollo de software o SDK provisto por Google el cual se puede descargar gratuitamente.

Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

Google Play es la tienda y plataforma en line de software desarrollado por Google para dispositivos Android. Es una aplicación que se encuentra instalada en la mayoría de los dispositivos Android que permite a los usuarios descargar aplicaciones publicadas para diferentes necesidades del usuario como apps de música, juegos, noticias, clima, educación, compras, salud, deportes, mapas y más por los desarrolladores tanto de Google y otros más, aunque hay aplicaciones que se deben comprar para descargar por medio de ciertas formas de pago que Google play pone a disposición como por PayPal, tarjeta de crédito y canjeo de códigos de play store para poder usar algunas aplicaciones no gratuitas como juegos principalmente.

Para que el usuario pueda usar esta plataforma de Play Store es necesario asociar una cuenta de Gmail con contraseña. Google retribuye a los desarrolladores el 70 % del precio de las aplicaciones. Esta aplicación reemplazo a Market. Por otra parte, los usuarios pueden instalar aplicaciones desde otras tiendas virtuales, o directamente en el dispositivo si se dispone del archivo APK de la aplicación.

3 CAPITULO III

3.1. MARCO PRACTICO (INGENIERIA DE PROYECTO)

3.1.1. Introducción

En esta etapa del proyecto desarrollaremos una descripción del diseño e implementación de un prototipo dispensador para alimentos automáticos gestionado directamente y remotamente "

3.1.2. Descripción General

El sistema cumple con el objetivo general del proyecto, es controlado mediante un celular, módulo sim800l, módulo Puente H. Los mensajes del celular son enviados al módulo y son transmitidos al Microcontrolador ATMEGA328p el cual se encarga de mandar la opción de habilitación para la automatización del dispensador de alimentos, mostrando en pantalla la dosificación para alimentos.

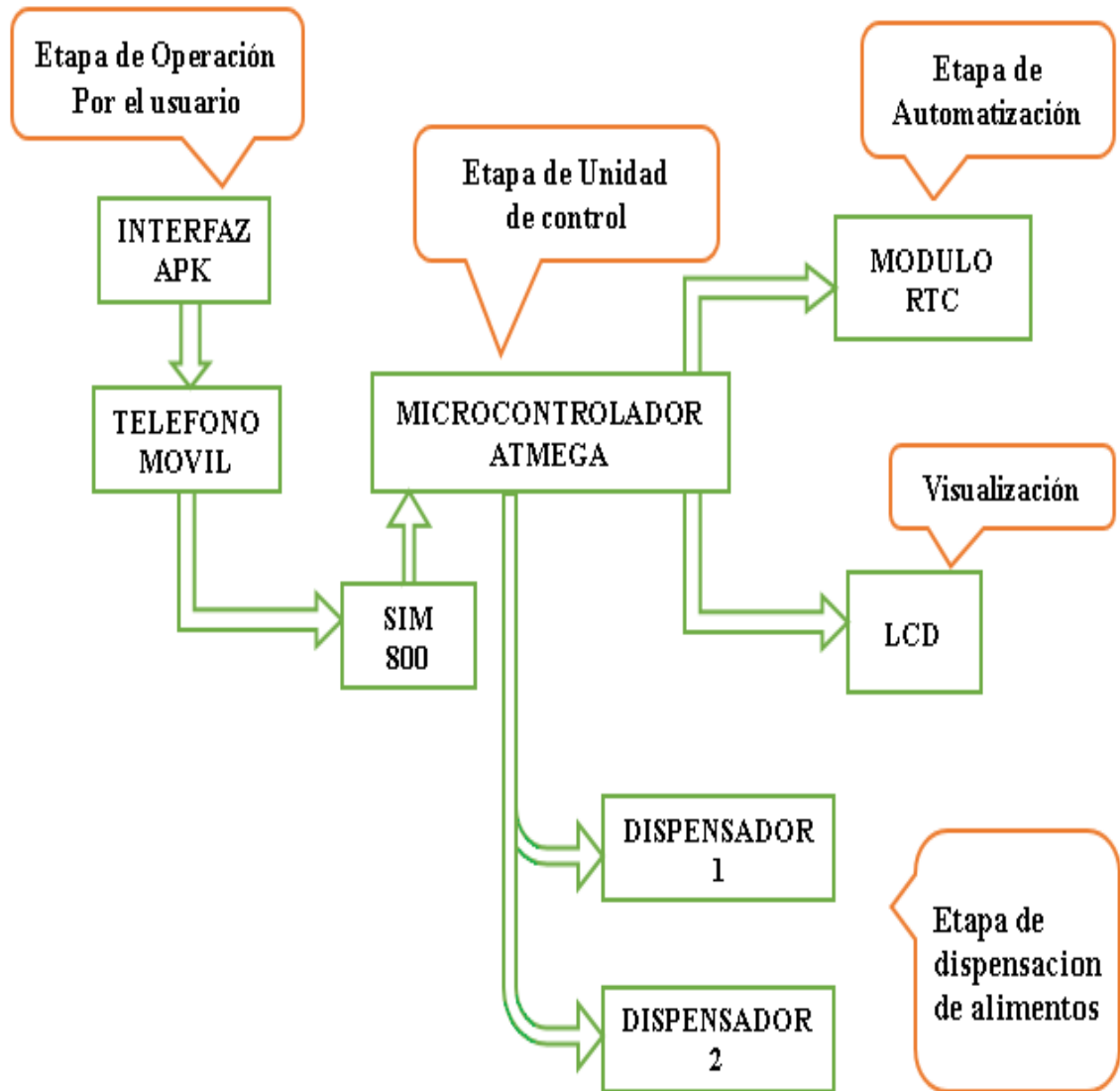
3.1.3. Microcontrolador

El microcontrolador ATMEGA328p, el cual se encarga de realizar las acciones definidas por el firmware mediante los datos recibidos por el sim800L para controlar los eventos externos.

3.1.4. Sim800l

El módulo sim800L nos permite la comunicación de dos dispositivos para compartir información ya se envió o recepción de datos, en este caso para la recepción de datos y poderlo comunicar al microprocesador. Así mismo poder enviar datos recibidos del Sensor Ultrasónicos para tener una reacción en el Módulo de puente H

3.2. DIAGRAMA DE BLOQUES DEL PROYECTO



La explicación se detalla en: **3.4.1 Etapas**

3.3. PROGRAMACION

3.3.1. Firmware de Configuración

El firmware es el código o programa que se encuentra grabada en el microcontrolador para realizar la comunicación con el móvil. Proporciona lógica y está programado en lenguaje de Programación. El firmware recibe órdenes externas y responde operando el dispositivo.

3.3.2. Software de configuración

El software de configuración es la interfaz gráfica entre un celular y el usuario, es compatible solo con celulares con sistema Android.

3.4. DESARROLLO DEL TRABAJO

3.4.1. Etapas

El presente trabajo se desarrolla en 4 diferentes etapas:

1. Etapa de operación por el usuario de envió de datos del teléfono móvil al circuito, esta se desarrolla en una aplicación Android que pueda conectarse con el módulo sim800l que será la interfaz del circuito, y una vez conectada la aplicación pueda recibir datos para poder enviarlos al circuito.
2. Etapa de la unidad de control del microcontrolador ATmega 328p. es una etapa principal del trabajo, este microcontrolador recibe el dato enviado por el Interfaz sim800l, para hacer una comparación y realice una determinada acción.
3. Etapa de conexión, donde estarán conectados en un orden al sistema de Modulo RTC, este cumplirá la acción de Programar los horarios de dispensación y automatizarlos, visualizando en la pantalla LCD.
4. Etapa de almacenamiento, donde el dispensador se encargará de distribuir el alimento enviado por el teléfono móvil al circuito.

3.5. SOFTWARE DE CONFIGIRACION DEL SISTEMA

En la configuración del sistema se utiliza el programa de MIT App Inventor, una herramienta de desarrollo para hacer una aplicación en Android.

3.5.1. Sistema Android como Herramienta de desarrollo

Android es un sistema operativo diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, relojes Smart pulseras, inicialmente desarrollado por Android.

3.5.2. Proceso de configuración

Para poder trabajar con Android se tiene que instalar el entorno de desarrollo MIT de App Inventor.

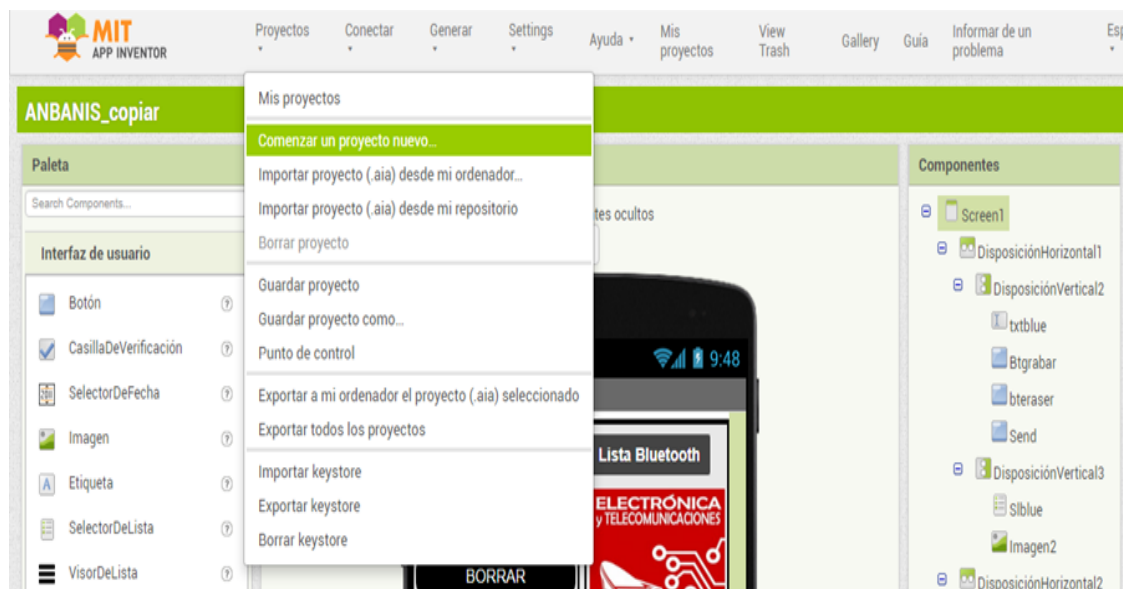


Figura 3.1 Creación de Nuevo Proyecto App Inventor

Fuente: Propia

Elegimos el nombre de la aplicación que le vamos a asignar al Proyecto.

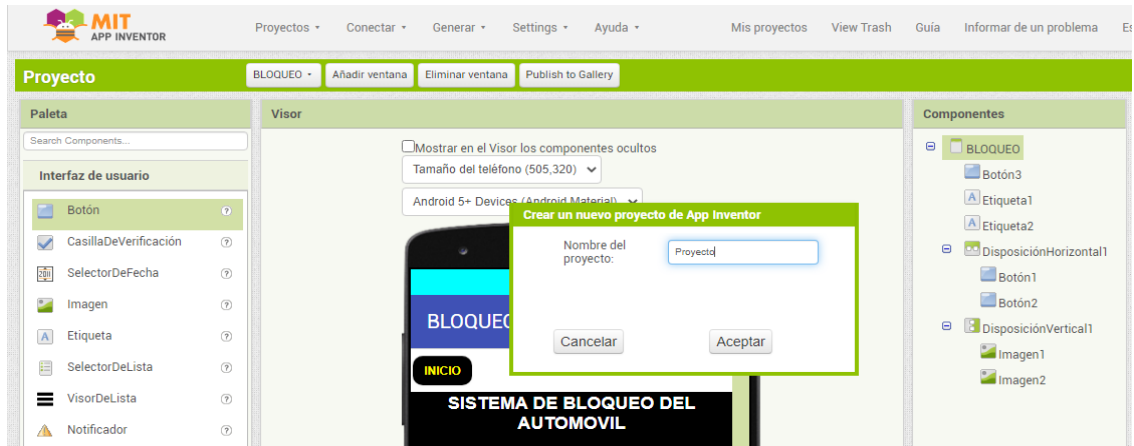


Figura 3.2 Nombre del Proyecto

Fuente: Propia

Nos muestra la pantalla de manera de screen1 y con la ayuda de la herramienta paleta usamos las Disposiciones Horizontales y verticales para asignar áreas de botones y etiquetas para el diseño de nuestra aplicación.

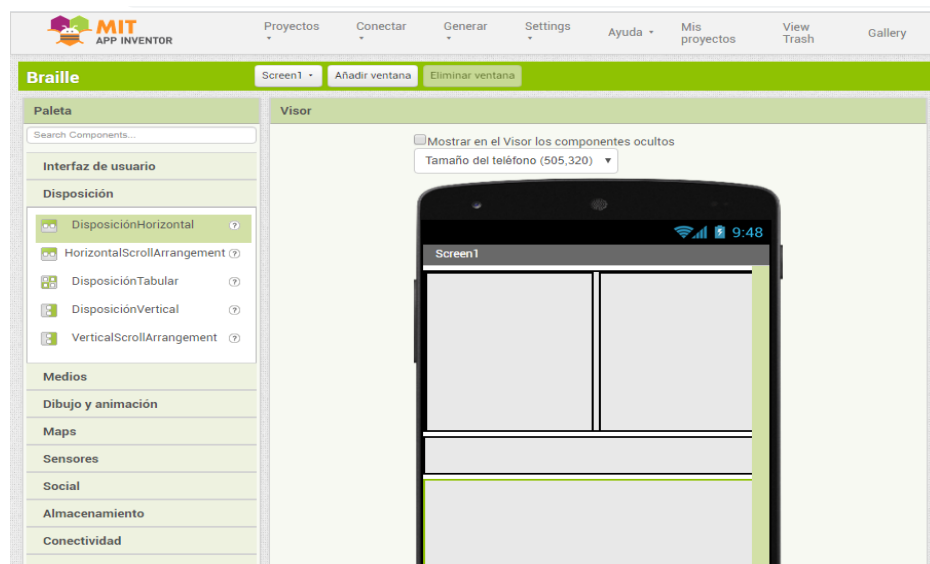


Figura 3.3 División de Disposiciones App Inventor

Fuente: Propia

Seleccionamos en la interfaz de usuario los botones, las etiquetas y la lista de menú para la conectividad con el circuito, realizamos todo el entorno de pantallas que deseamos ver en nuestro teléfono móvil.



Figura 3.4 Selección de Interfaz usuario

Fuente: Propia

Desarrollamos el entorno final de la aplicación, de manera amigable para que el usuario tenga la facilidad de usarla.



Figura 3.5 Entorno Final de la APK en App Inventor

Fuente: Propia

Después del entorno final, realizamos la programación de la acción que procederá cada uno de los botones y ventanas, en la pestaña de programación de bloques que pusimos en nuestro entorno.

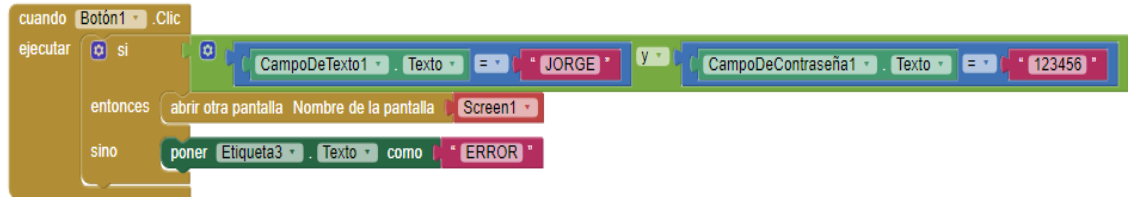


Figura 3.6 Programación del entorno ventana usuario

Fuente: Propia

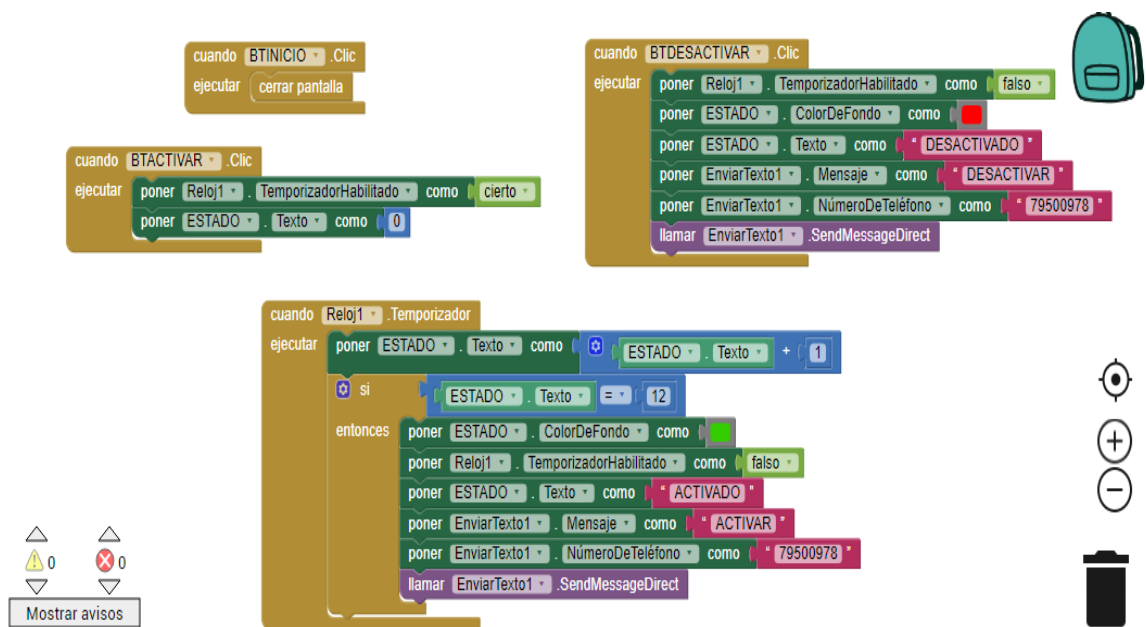


Figura 3.7 Programación del Entorno registro de datos

Fuente: Propia

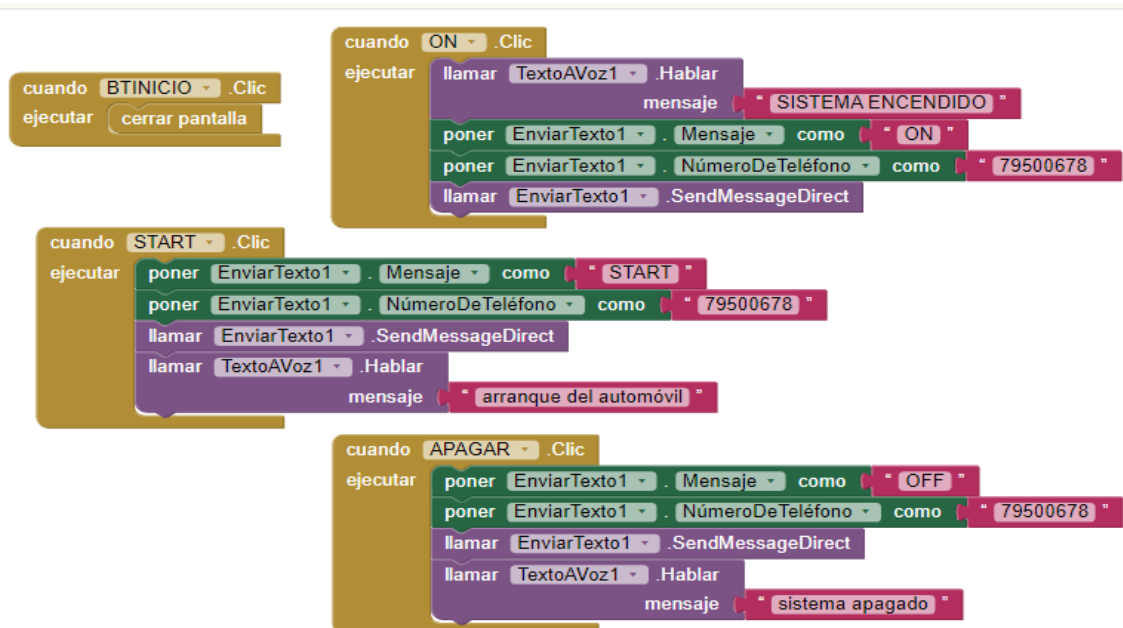


Figura 3.8 Programación del Entorno envió de datos

Fuente: Propia

3.6. FIRMWARE DEL SISTEMA

El microcontrolador ATmega 328p de 28 pines ejecutara el firmware del sistema, para compilar usaremos IDE de Arduino.

El programa en el microcontrolador debe realizar las siguientes tareas para cumplir con los objetivos planteados.

- El sistema de identificación biométrica conjunta con el reconocimiento de comandos de voz, deberá habilitar el encendido del automóvil.
- El microcontrolador debe establecer un enlace con el Sistema GPRS-GSM para recibir el dato enviado por el Teléfono Móvil y esperar para realizar accionamiento en el sistema de seguridad.
- Una vez establecida la comunicación entre los dos dispositivos, el microcontrolador debe cumplir la acción de enviar la información solicitada: Sistema de advertencia contra intrusos o ubicación GPS al teléfono móvil.

3.6.1. IDE Arduino Herramienta de desarrollo

El programa principal del microcontrolador empieza con una definición de las variables, la asignación de entrada y salida y la comunicación tipo serial, importamos las librerías para la configuración serial y la configuración LCD,

El void setup es la primera función en ejecutarse, declaramos la iniciación de columnas y filas del LCD. Además de la velocidad de transferencia del serial a 9600 baudios, la designación de pines para la conexión Serial de las diferentes etapas designadas, la salida de los sistemas GPRS-GSM, GPS, Biométrico y Reconocimiento de voz.

El void loop es la que se ejecuta un número infinito de veces, limpiamos el LCD y ponemos el cursor para mostrar el mensaje, declaramos la condición para el evento serial, definimos la instrucción de comparación para las acciones que debe tomar en el sistema de seguridad, envió de mensaje al propietario si existe algún intruso en el automóvil, envió de ubicación de GPS del automóvil y encendido automatizado por medio de comandos de voz e identificación biométrica. También puede llegar a encender el automóvil por medio del teléfono móvil.

3.6.2. Diagrama de Flujo

Diagrama de Flujo del programa Implementado en el Microcontrolador ATmega328p, para la ejecución del proyecto:

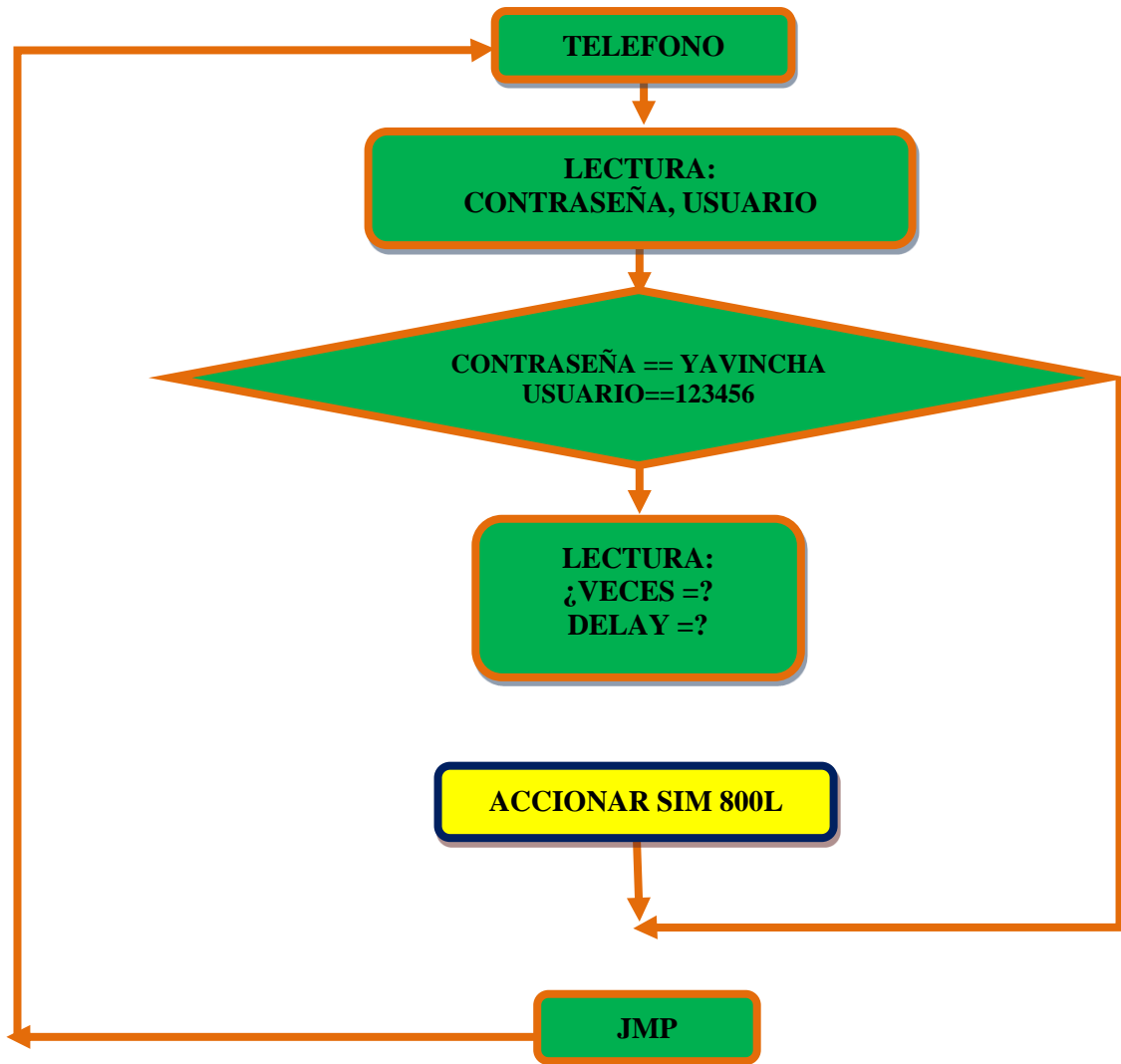


Figura 3.9 Diagrama de flujo Teléfono APK

Fuente: Propia

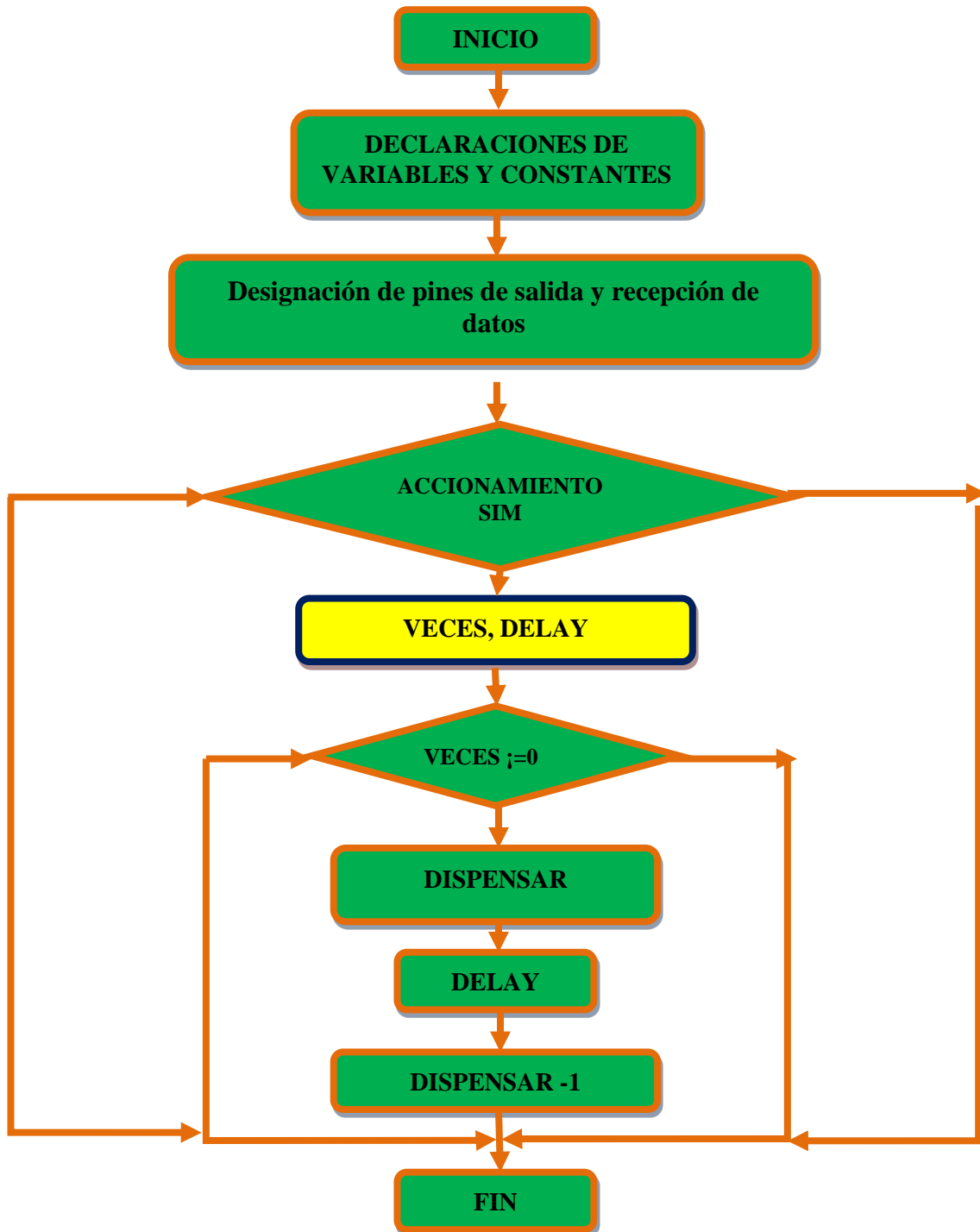


Figura 3.10: Diagrama de flujo Microcontrolador

Fuente: propia

3.6.3. Programación elaborada en IDE SETUP

El compilador crea un proyecto en donde se guardará el archivo para poder cargar al microcontrolador.

```
#include <Wire.h>
#include "RTClib.h"
#include <SoftwareSerial.h> //Incluimos la librería Software Serial

SoftwareSerial SerialComSim800(11,12); //RX11, tx12
String valor;
const int output Pin = LED_BUILTIN;
const int led=2;
bool state = false;
// RTC_DS1307 rtc;
RTC_DS3231 rtc;

String daysOfTheWeek [7] = {"Domingo", "lunes", "martes", "miércoles", "jueves",
"viernes", "sábado"};
String monthsNames [12] = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio",
"Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"};

void setup () {
pin Mode (13, OUTPUT);

Serial. Begin (9600);
SerialComSim800.begin(9600);

Serial.println("Iniciando...");
delay (2000);
SerialComSim800.println("AT");
serial Check ();

SerialComSim800.println("AT+CMGF=1");
serial Check ();

//SerialComSim800.println("AT+CMGS="+59175868288+"\r\n"); //envio de numero
SerialComSim800.println("AT+IPR=9600");
```

```

serial Check ();
SerialComSim800.println("AT+CNMI=1,2,0,0,0");
serial Check ();
//led=OUTPUT;
Serial. Begin (9600);
delay (1000);
if (! rtc. Begin ()) {
    Serial.println(F ("Couldn't find RTC"));
    while (1);
}

if (rtc. lost Power ()) {
    rtc. Adjust (Date Time(F(__DATE__), F(__TIME__)));
}
}

// Comprobar si está programado el encendido
bool isScheduledON (Date Time date)
{
    int weekday = date. dayOfTheWeek ();
    float hours = date. Hour () + date. Minute () / 60.0;
    // De 09:30 a 11:30 y de 21:00 a 23:00
    bool hour Condition = (hours > 19.00 && hours < 19.20) || (hours > 21.00 && hours <
23.00);
    // Miercoles, Sabado o Domingo
    bool day Condition = (weekday == 3 || weekday == 6 || weekday == 0);
    if (hour Condition && day Condition)
    {
        return true;
    }

    return false;
}

void print Date (Date Time date)
{
    Serial. Print (date. Year (), DEC);
    Serial. Print ('/');
}

```

```

Serial. Print (date. Month (), DEC);
Serial. Print ('/');
Serial. Print (date. Day (), DEC);
Serial. Print (" ");
Serial. Print (daysOfTheWeek [date. dayOfTheWeek ()]);
Serial. Print (" ");
Serial. Print (date. Hour (), DEC);
Serial. Print (':');
Serial. Print (date. Minute (), DEC);
Serial. Print (':');
Serial. Print (date. Second (), DEC);
Serial.println();
}

void loop () {
  Date Time now = rtc. Now ();
  print Date(now);
  if (state == false && isScheduledON(now)) // Apagado y debería estar encendido
  {
    digital Write (output Pin, HIGH);
    state = true;
    Serial. Print("Activado");
    digital Write (led, HIGH);
  }

  else if (state == true &&! isScheduledON(now)) // Encendido y debería estar apagado
  {
    digital Write (output Pin, LOW);
    state = false;
    Serial. Print("Desactivar");
    digital Write (led, LOW);
  }

  delay (3000);
  serial Check ();
  if (SerialComSim800.available()) {
    valor=SerialComSim800.readString();
  }
}

```

```
if (SerialComSim800.read() == "ON") {
  digital Write (13, HIGH);
  Serial.println("encendido");
  delay (3000);
}

else {
  digital Write (13, LOW);
  Serial.println("apagado");
  delay (3000);
}
}

void serial Check () {
  delay (500);
  while (Serial. Available ()) {
    SerialComSim800.write(Serial. Read ());
  }
  while (SerialComSim800.available()) {
    Serial. Write (SerialComSim800.read());
  }
}
}
```

3.7. HARDWARE DEL SISTEMA

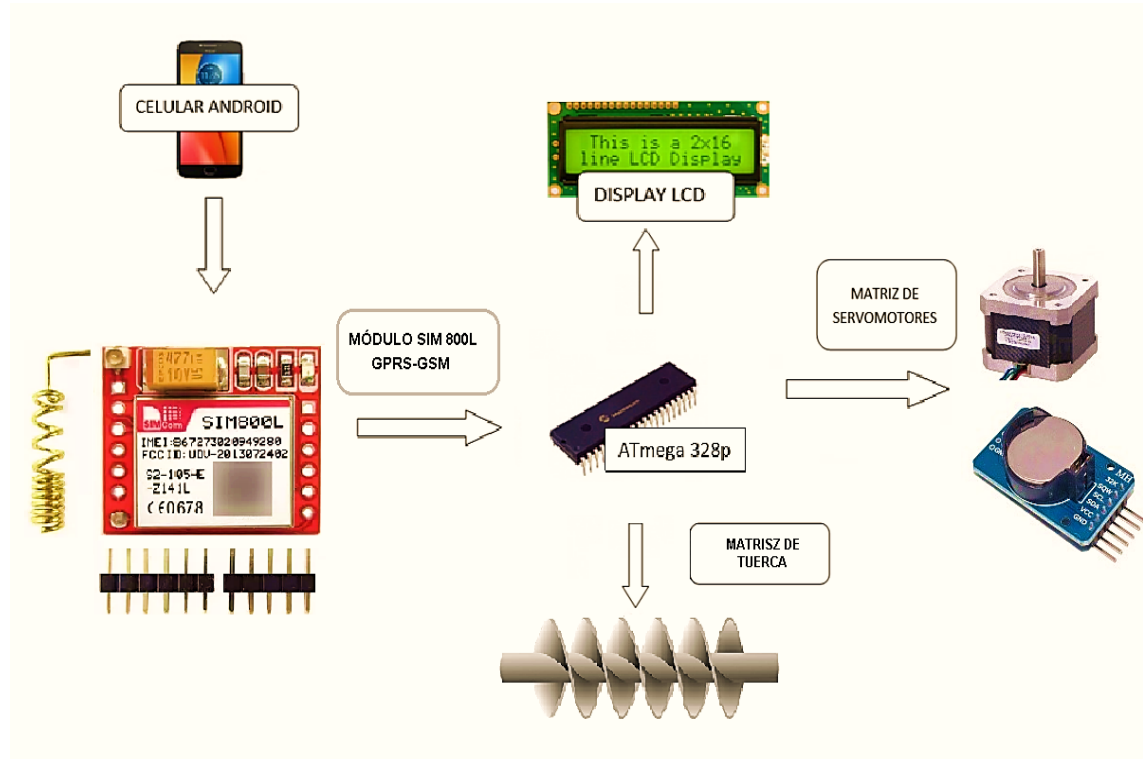


Figura 3.11 Diagrama del prototipo

Fuente: Propia

3.7.1. Microcontrolador

El microcontrolador ATmega 328p marca Atmel es la parte central del sistema, en el reside el firmware y ejecuta todas las acciones que esta solicita.

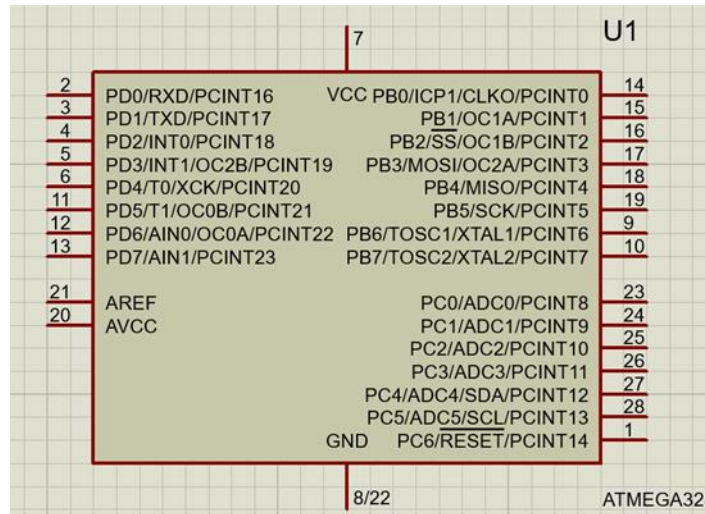


Figura 3.12 Microcontrolador ATmega 328P (Isis Proteus)

Fuente: Labcenter/Isisproteus

3.7.2. Módulo Sim 8001 GPRS-GSM

Hemos visto varias formas de conectar nuestro **Arduino** al exterior, como Ethernet y wifi, pero puede que alguna vez queramos comunicarnos con él y no tengamos acceso a ninguna de estas redes, o no queramos depender de ellas. Puede ser muy útil, por ejemplo, para colocar en una estación meteorológica.

Para este tipo de propósitos podemos utilizar un módulo **GSM/GPRS** con una tarjeta SIM, de forma que podamos comunicarnos con él como si se tratase de un teléfono móvil. Y es que esta tarjeta basada en el módulo **SIM900** nos permite enviar y recibir llamadas y SMS y conectarnos a Internet, transformando nuestro Arduino en un teléfono móvil.

EL **GSM** (*Global System for Global Communications*) es el sistema de comunicaciones que más se utiliza en teléfonos móviles y es un estándar en Europa.

La primera funcionalidad de esta tecnología es la transmisión de voz, pero también permite la transmisión de datos (SMS, Internet), eso sí, a una velocidad muy baja de 9kb/s.

El **GPRS** (*General Packet Radio Service*) es una extensión del GSM basada en la transmisión por paquetes que ofrece un servicio más eficiente para las comunicaciones de datos, especialmente en el caso del acceso a Internet.

La velocidad máxima (*en teoría*) del GPRS es de 171kb/s, aunque en la práctica es bastante más pequeña.

Para poder comunicarnos vía comandos AT tendremos que cargar un programa para la comunicación serie como hemos hecho tantas veces anteriormente. Crearemos una instancia llamada SIM808 y seleccionaremos los pines del Arduino que queramos usar para comunicarnos (Rx y Tx). Nosotros hemos elegido el 7 y el 8, pero podéis usar cualquiera que sea compatible con la librería. También podéis cambiar la velocidad de comunicación, pero debe ser la misma para el puerto serie y para la instancia que hemos creado.



Figura 3.13 Sim800l

Fuente: <https://uelectronics.com/producto/sim800l-modulo-gprs-gsm-sim800-con-antena/>

3.7.3. Módulo RTC

El Módulo RTC DS3231 Reloj de Tiempo Real podrás integrarlo a tus proyectos para que tengan la capacidad de almacenar y llevar la cuenta de fecha y hora por medio de la interfaz de comunicación I2C. El módulo está basado en el RTC DS3231 de MAXIM y la EEPROM AT24C32 de ATMEL.

Ambos circuitos integrados comparten el mismo bus comunicación con el Protocolo I2C. El RTC DS3231 es la evolución del clásico RTC DS1307. La principal diferencia con el DS1307 es el oscilador interno compensado por temperatura, lo que hace que su precisión sea muy alta. La memoria EEPROM AT24C32 permite almacenar 32Kbits (4K Bytes) de datos de manera permanente.

3.7.3.1. Especificaciones y características

- Voltaje de Alimentación DC: 3.3V ~ 5V
- Tipo de Comunicación: I2C
- Bajo consumo de energía
- Compatible con Arduino, Raspberry Pi y otras tarjetas.
- Exactitud Reloj: 2ppm
- Dirección I2C del DS3132: Read (11010001) Write (11010000)
- Memoria EEPROM AT24C32 (4K * 8bit = 32Kbit = 4KByte)
- RTC de alta precisión DS3231 con oscilador interno
- Puede ser usado en cascada con otro dispositivo I2C, la dirección del AT24C32 puede ser modificada (por defecto es 0x57)
- No incluye **batería CR2032**
- Dimensiones: 38 mm x 22 mm x 12 mm

3.7.3.2. Funciones RTC:

- Fecha con segundos, minutos, horas, numero día, día de la semana, mes y año.
- Compensación de año bisiesto.
- Formato de hora configurable en 12 o 24 horas.
- 2 alarmas configurables
- Circuito compensador de temperatura para el voltaje de referencia interno

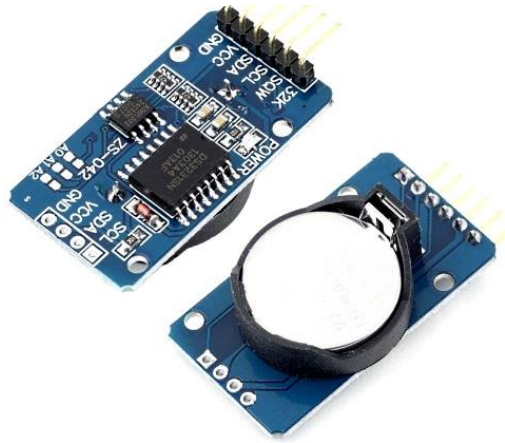


Figura 3.14 Modulo RTC

Fuente: Epy electronic

3.8. DIAGRAMA COMPLETO DEL CIRCUIO

La presentación del circuito que conforma el hardware, se realiza el diseño con ares para la placa.

3.8.1. Circuito del Sistema Dispensador

En el diagrama se observa la parte central del circuito completo. Las conexiones para las salidas y el módulo en los sistemas GPRS-GSM, Modulo RTC, motor paso a paso. además de una visualización por el display LCD.

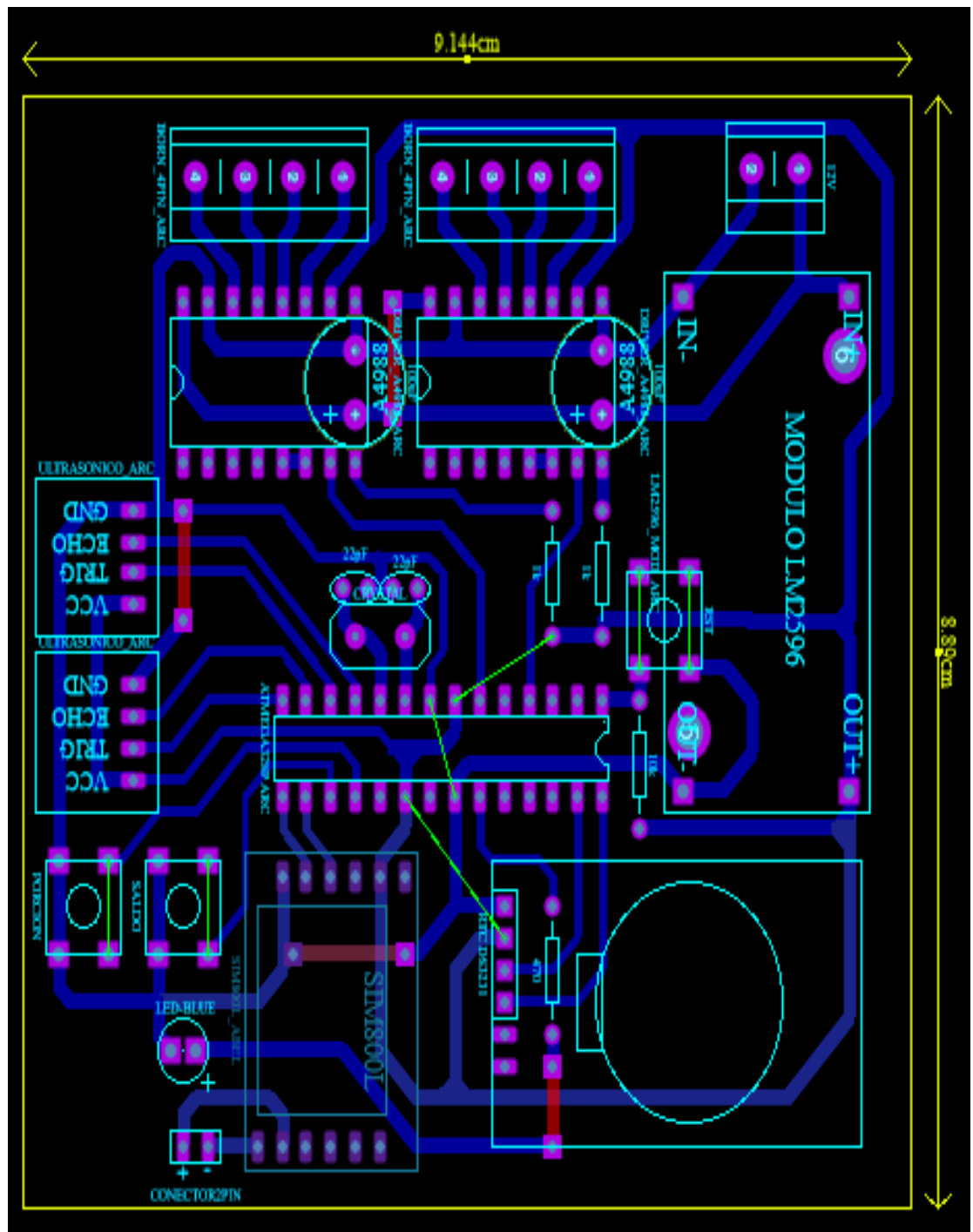


Figura 3.16 Diseño Placa PCB

Fuente: Propia ARES Proteus

3.9. PRUEBAS Y RESULTADOS DE FUNCIONAMIENTO

En esta sección de ingeniería se presentan las imágenes de la implementación física del circuito y algunos detalles relevantes al respecto. En el documento se encuentra el análisis de los resultados en base de pruebas de funcionamiento del circuito diseñado.

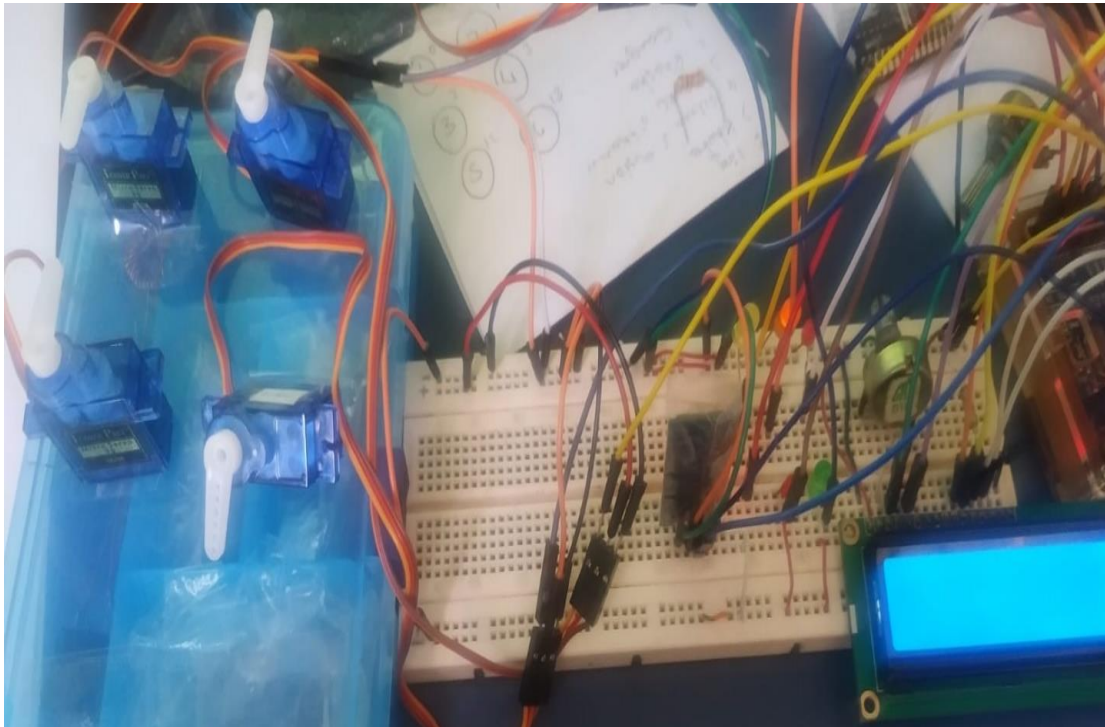


Figura 3.17 Pruebas de funcionamiento

Fuente: propia

4 CAPITULO IV

4.1. ANALISIS DE COSTOS

En esta sección se encuentra el análisis de costos del proyecto. Este se divide en el costo del desarrollo de software, el costo de construcción del hardware del prototipo.

4.1.1. Costos de diseño de software

El análisis de costo de diseño del software se emplea el modelo constructivo de coste empírico. Según Boehm, quien es el desarrollador y está definido para tres tipos de proyectos: Orgánico, semi-acoplado y empotrado.

Tenemos las siguientes ecuaciones:

$$E = A_b * KLDC * B_b * FAE$$

$$D = C_c * E * D_d$$

$$N = E/D$$

Ecuación 4.1 Ecuaciones de Cocomo intermedio

"COCOMO → es donde mide el esfuerzo de realizacion"

Donde:

E → Esfuerzo aplicado en personas – mes

KLDC → Numero de lineas de codigo (en miles)

FAE → Factor de ajuste del esfuerzo

D → Tiempo de desarrollo (en meses)

N → Numero de personas necesarias para el desarrollo del software

A_b, B_b, C_b, D_b → Son coeficientes extraidos de la tabla 4.1

Se utiliza para obtener una primera aproximación rápida del esfuerzo y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

Proyecto de Software	A (difícil)	B (semidifícil)	C (medio fácil)	D (fácil)	Descripción
Orgánico (mental)	3.20	1.05	2.5	0.38	Aplicaciones bien comprendidas, desarrolladas por equipos pequeños.
Semi - orgánico (mental - físico)	3.00	1.12	2.5	0.35	Proyectos más complejos en los que no tienen la misma experiencia todos los miembros del equipo. Hay requisitos más y menos rígidos.
Empotrado (físico)	2.80	1.20	2.5	0.32	El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

Tabla 4.1 Coeficientes a, b, c, d según el tipo de proyecto

Para este ajuste, al resultado de la fórmula general se multiplica por el coeficiente sugerido de aplicar los atributos seleccionados.

El factor de ajuste del esfuerzo FAE se determina multiplicando 15 factores extraídos de la tabla 4.2.

$$FAE = Fr * Tbd * Cp * Rt * Ra * Vmv * Tr * Ca * Ea * Cpr * Eso * El * Pm * Us * L$$

Ecuación 4.2 Factor de ajuste del esfuerzo

Unidades de dimensiones que miden el espacio de la realización del proyecto

Atributos	Valoración					
	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Atributos del Software						
Fiabilidad requerida del software (Fr)	0.75	0.88	1.00	1.15	1.40	-
Tamaño de la base de datos (Tbd)	-	0.94	1.00	1.08	1.16	-
Complejidad del Producto (Cp)	0.70	0.85	1.00	1.15	1.30	1.65
Atributos de hardware						
Restricciones del Tiempo de Ejecución (Rt)	-	-	1.00	1.11	1.30	1.66
Restricción del almacenamiento principal (Ra)	-	-	1.00	1.06	1.21	1.56
Vitalidad de la máquina virtual (Vmv)	-	0.87	1.00	1.15	1.30	-
Tiempo de respuesta del ordenador (Tr)	-	0.87	1.00	1.07	1.15	-
Atributos de personal						
Capacidad del Analista (Ca)	1.46	1.19	1.00	0.86	0.71	-
Experiencia en la aplicación (Ea)	1.29	1.13	1.00	0.91	0.82	-

Capacidad de los programadores (Cpr)	1.42	1.17	1.00	0.86	0.70	-
Experiencia en S.O. utilizados (Eso)	1.21	1.10	1.00	0.90	-	-
Experiencia en el lenguaje de programación (El)	1.14	1.07	1.00	0.95	-	-
Atributos del Proyecto						
Prácticas de programación modernas (Pm)	1.24	1.10	1.00	0.91	0.82	-
Utilización de herramientas (Us)	1.24	1.10	1.00	0.91	0.83	-
Limitaciones de planificación del Proyecto (L)	1.22	1.08	1.00	1.04	1.10	-

Tabla 4.2 Valoración de los conductores de coste para determinar la FAE

4.1.1.1. Análisis de costo del diseño del software

Consideramos como un proyecto de software semi-acoplado para el análisis de costo.

$$FAE = 1.15 * 0.94 * 1 * 1.11 * 1 * 0.87 * 1 * 1 * 1.29 * 1.17 * 1.10 * 1.07 * 1 * 0.91 * 1.08$$

$$FAE = 1,82$$

$$KLCD = \frac{470}{1000}$$

$$KLDC = 0,47$$

$$E = 3 * 0,47^{1,12} * 1,82$$

$$E = 2,34 \text{ [personas - mes]}$$

$$D = 2,5 * 2,34^{0,35}$$

$$D = 3,36 \text{ [meses]}$$

$$N = \frac{2,34}{3,36} = 0,69 \cong 1 \text{ [personas]}$$

para determinar el costo del software:

$$C_s = E * I_p$$

Ecuación 4.3 Costo económico de software

Donde:

$E \rightarrow$ Proviene de las ecuaciones de Cosomo

$I_p \rightarrow$ Remuneración mensual de un profesional en el área

$$C_s = 2,34 * 2.250 = 5265 Bs$$

4.1.2. Costos en la construcción del hardware

En las tablas se muestra la lista de componentes empleados, precios unitarios y cantidades, además de elementos adicionales en la construcción.

Nro.	Componente	Valor	Precio Unitario (Bs)	Cantidad	Precio Total (Bs)
1	Resistores	1K Ω / 1/4W	0,30	4	1,20
2	Diodo led	1	0,50	1	0,50
3	Capacitor	22pF,50u	1,00	5	5,00
4	Sim 800L	MINI	90,00	1	90,0
5	Display LCD	LCD 20x4	58,00	1	58,00
6	Motor Paso a Paso	10k Ω / 1/4W	90,00	2	180,00
7	Módulo Regulador 5v	LM2596	15,00	1	15,00
8	Puente H	L298	17,00	2	34,00
9	Módulo Distancia	Hc-04	14,00	2	28,00
10	Modelado 3d, diseño propio	Tornillo,	90,00	2	180,00
		Tapa,	37,50	2	75,00
		despachador	37,50	2	75,00
		Soporte hc04	25,00	2	50,00
11	Módulo I2c	I2c	11,00	1	11,00

12	Circuito Integrado	Atmega328p	34,00	1	34,00
13	Crystal	16MHz	3,00	1	3,00
14	Placa PCB		10,00	1	10,00
15	conectores		1,00	22	22,00
16	Bornes		2,00	1	2,00
17	pulsador	2pins	1,00	3	3,00
18	Fuente Alimentación	12v,2A	32,00	1	32,00
19	Zocalo de 28	Dip28	2,00	1	2,00
20	Tubos, tapas	pbc	11,00	2	22,00
21	Modulo RTC	Ds3231	23,00	1	23,00
22	Peineta	Hembra, macho	3,00	3	9,00
PRECIO TOTAL					964,70

Tabla 4.3 Valoración del Hardware

4.1.3. Costo final del Proyecto

El costo final se determina mediante la siguiente ecuación:

$$CFp = \text{costo total de diseño de software} + \text{costo total de componentes}$$

$$CFp = 5265,00 + 964,70$$

$$CFp = 6229,70 Bs$$

Es posible aclarar que el costo final del proyecto es para un prototipo con fines académicos.

5 CAPITULO V

5.1. CONCLUSIONES Y RECOMENDACIONES

5.1.1. Conclusiones

Luego de realizar el diseño, construcción, pruebas de funcionamiento del prototipo y el análisis económico, se destacan las siguientes conclusiones.

- Se realizo satisfactoriamente la transmisión de datos del microcontrolador ATmega328p vía comunicación Serial.
- Se diseño e implemento el prototipo para la dispensación de alimentos automático y remotamente.
- Al ser una plataforma de código y hardware abierto nos permite la comunicación libre con el sistema hasta comercializar sin necesidad de procedimientos legales ya que no se paga licencias.
- La plataforma SIM GSM y Android, al ser un código abierto nos reduce un costo de licencias y nos permite aplicar en nuestro hardware para un funcionamiento optimo

5.1.2. Recomendaciones

Al momento de conectarnos al módulo para la interacción debemos tomar en cuenta la conexión a este mismo, ya que pueden existir varias conexiones en el área donde uno se encuentra.

Se considera factible el desarrollo de aplicaciones para teléfonos celulares, para hacer que el entorno de control sea más amigable que el del presente trabajo.

6 CAPITULO VI

6.1. REFERENCIAS BIBLIOGRAFICAS

- Clavijo J.R. (2011). Diseño y Simulación de Sistemas microcontrolados en lenguaje C. Colombia: ISBN
- López J.L. (2008). Microcontrolador PIC16F84 Desarrollo de Proyectos. (2. Ed.). Estados Unidos de América: Alfaomega-Rama
- Guachalla R. (2012). Guías de Laboratorio de Microprocesadores 1. Bolivia-La Paz. Recuperado de la base de datos Repositorio Facultad de Tecnología

6.2. PAGINAS WEB

- Llamas, L. (2017, julio 11). Array separado por comas por puerto serie en Arduino. Luis Llamas. <https://www.luisllamas.es/arduino-array-separado-comas/>
- Pérez, A. (s/f). Prácticas y código ejemplo para cadena de texto o string con Arduino Uno – Parte 1 (MIC031S). Com.mx. Recuperado el 3 de mayo de 2023, de <https://www.incb.com.mx/index.php/articulos/78-microcontroladores-y-dsps/2533-practicas-y-codigo-ejemplo-para-cadena-de-texto-o-string-con-arduino-uno-parte-1-mic031s>
- MIT app inventor. (s/f). Mit.edu. Recuperado el 3 de mayo de 2023, de <https://appinventor.mit.edu/>
- App inventor. Trucos, consejos sugerencias Tips and Tricks App inventor. Pantalla completa F11 Full Screen. (s/f). Kio4.com. Recuperado el 3 de mayo de 2023, de http://kio4.com/appinventor/100_trucos.htm
- (S/f). Patagoniatec.com. Recuperado el 3 de mayo de 2023, de <https://saber.patagoniatec.com/2014/06/atmega328p-pu-arduino-argentina-ptec/>

- Torres, A. L. (s/f). *La historia del Braille, un alfabeto accesible*. *Laciudadaccesible.com*. Recuperado el 3 de mayo de 2023, de <http://periodico.laciudadaccesible.com/portada/opinion-la-ciudad-accesible/item/4605-la-historia-del-braille-un-alfabeto-accesible>
- Jecrespom, P. (s/f). *arquitectura microcontrolador –. Aprendiendo Arduino*. Recuperado el 3 de mayo de 2023, de <https://aprendiendoarduino.wordpress.com/tag/arquitectura-microcontrolador/>
- (S. f.). *Discapnet.es*. Recuperado 3 de mayo de 2023, de <http://www.dicapnet.es/areas-tematicas/disenio-para-todos/accesibilidad-de-comunicacion/>

ANEXOS

ANEXO A – Código de Proyecto

```
#include <SoftwareSerial.h>
#include <Wire.h> // incluye librería para interfaz I2C
#include <LiquidCrystal_I2C.h>
#include <RTClib.h> // incluye librería para el manejo del módulo RTC

RTC_DS3231 rtc; // crea objeto del tipo RTC_DS3231
SoftwareSerial SIM900(9,10); // for gem module RX, TX
LiquidCrystal_I2C lcd(0x27,20,4);

#define led Estado A0
#define saldo 11
#define Bvueltas 12
int Porciones = 1;

//VARIABLES DEL GPS + SIM900 + LCD + MILLIS + OTROS
String Numero = "AT+CMGS=\"+59175868288\""; //PARA MENSAJES
String Numero Llamada = "ATD+59175868288;"; //PARA LLAMADAS
String Numero Saldo = "ATD*33;"; //PASA EL SALDO ENTEL
//*****SMS
String estado="";
String mensaje = ""; // Se le asigna los mensajes que se reciben
String recibido; //utilizamos una variable para recibir el SMS
//*****SMS

//MOTORES
# define PULL_M1 2
# define PULL_M2 4
# define EN_M1 A3
```

```

# define EN_M2 A1
int Vueltas, hora1, min1, hora2, min2, hora3, min3;
char c;
//ultrasonico
#define trig 6      // Pin Trigger
#define echo 5      // Pin Eco
#define trig2 8     // Pin Trigger
#define echo2 7     // Pin Eco
Long duraci3n, distancia; // Duraci3n para calcular la distancia
Long duracion2, distancia2; // Duraci3n para calcular la distancia
int D comida Vacía, DcomidaVacía2, Ccomida= 2; //x cm de comida
unsigned long tiempo Distancia;

void setup () {

    pin Mode (ledEstado, OUTPUT);
    digital Write (ledEstado, LOW);
    pin Mode (saldo, INPUT_PULLUP);
    pin Mode (B vueltas, INPUT_PULLUP);
    pin Mode (EN_M1, OUTPUT);
    pin Mode (EN_M2, OUTPUT);
    digital Write (EN_M1, HIGH);
    digital Write (EN_M2, HIGH);
    pin Mode (PULL_M1, OUTPUT);
    pin Mode (PULL_M2, OUTPUT);
    digital Write (PULL_M1, LOW);
    digital Write (PULL_M2, LOW);
    pin Mode (echo, INPUT);
    pin Mode (trig, OUTPUT);

```



```

    delay (1000);
}

SIM900.println("AT+CMGF=1"); //Set SMS Text Mode
delay (1000);
SIM900.println("AT+CNMI=2,2,0,0,0"); //procedure, how to receive messages from the
network
delay (1000);

//SIM900.setTimeout(100);

CalculoDistacia ();
CalculoDistacia2();
DcomidaVacía = distancia-Ccomida;
DcomidaVacía2 = distancia2-Ccomida;
tiempoDistancia = millis ();

lcd. set Cursor (0, 1); lcd.print("Hr1:00:00");
lcd. set Cursor (0, 2); lcd.print("Hr2:00:00");
lcd. set Cursor (0, 3); lcd.print("Hr3:00:00");
}

void Credit Button () {
    if (digital Read(saldo) == LOW)
    {
        digital Write (ledEstado, HIGH);
        delay (100);
        SIM900.println(NumeroSaldo);
        delay (100);
    }
}

```

```

SIM900.println();
delay (20000);
SIM900.println("ATH");
delay (100);
digitalWrite (led Estado, LOW);
}
}

```

```

void NumPorciones () {
  if (digitalRead (Bvueltas) == LOW) {
    digitalWrite (led Estado, HIGH);
    delay (100);
    Porciones++;
    if (Porciones >9) {Porciones = 1;}
    delay (100);
    digitalWrite (led Estado, LOW);
  }
}

```

```

void PorcionesLCD () {
  lcd. set Cursor (12, 2);
  lcd.print("Porción");
  lcd. set Cursor (15, 3);
  lcd.print (Porciones);
}

```

```

void CalculoDistacia () { // Esta función determina la distancia de un objeto con el sensor
ultra sónico
digitalWrite (trig, LOW);

```

```

delayMicroseconds (4);
digitalWrite (trig, HIGH); //A través del pin de disparo (Trigger) enviaremos una señal
de activación al sensor, lo activaremos durante 10 microsegundos y lo desactivaremos
delay Microseconds (10);
digital Write (trig, LOW);
duration=pulse in (echo, HIGH);
distancia = (duración/58.4); //solo esto y me medirá en cm
delay (50);
}

```

```

void DistanciaLCD () {
  lcd. set Cursor (12, 0);
  LCD. Print("Di1:"+String(distancia)+"cm ");
}

```

```

void CalculoDistacia2() { // Esta función determina la distancia de un objeto con el sensor
ultra sónico
  digitalWrite (trig2, LOW);
  delayMicroseconds (4);
  digitalWrite (trig2, HIGH); //A través del pin de disparo (Trigger) enviaremos una señal
de activación al sensor, lo activaremos durante 10 microsegundos y lo desactivaremos
  delay Microseconds (10);
  digital Write (trig2, LOW);
  duracion2=pulse in (echo2, HIGH);
  distancia2 = (duracion2/58.4); //solo esto y me medirá en cm
  delay (50);
}

```

```

void DistanciaLCD2() {
  lcd. set Cursor (12, 1);
  lcd. Print("Di2:"+String(distancia2) +"cm ");
}

```

```

void HoraTreal_LCD () {
  Date Time now = rtc. Now ();
  lcd. set Cursor (1,0); //C, F
  if (now. Hour () < 10) {
    lcd. Print ("0");
  }

```

```

  lcd. Print (now. Hour (), DEC); //imprime hora
  lcd. Print (':');
  if (now. Minute () < 10) {
    lcd. Print ("0");
  }

```

```

  lcd. Print (now. Minute (), DEC); //imprime minutos
  lcd. Print (':');
  if (now. Second () < 10) {
    lcd. Print ("0");
  }

```

```

  lcd. Print (now. Second (), DEC); //imprime Segundos
}

```

```

void Horas LCD () {
  lcd. set Cursor (0,1); //C, F

```



```
lcd. Print("Hr1:");
if (hora1 < 10) {lcd. Print ("0");}
lcd. Print(hora1);
lcd. Print (':');
if (min1 < 10) {lcd. Print ("0");}
lcd. Print(min1);
```

```
lcd. set Cursor (0,2); //C, F
lcd. Print("Hr2:");
if (hora2 < 10) {lcd. Print ("0");}
lcd. Print(hora2);
lcd. Print (':');
if (min2 < 10) {lcd. Print ("0");}
lcd. Print(min2);
```

```
lcd. set Cursor (0,3); //C, F
lcd. Print("Hr3:");
if (hora3 < 10) {lcd. Print ("0");}
lcd. Print(hora3);
lcd. Print (':');
if (min3 < 10) {lcd. Print ("0");}
lcd. Print(min3);
```

```
// lcd. set Cursor (0, 1); lcd. Print ("Hr.1: "+String(hora1) +":"+String(min1) +":00");
// lcd. set Cursor (0, 2); lcd. Print ("Hr.2: "+String(hora2) +":"+String(min2) +":00");
// lcd. set Cursor (0, 3); lcd. Print ("Hr.3: "+String(hora3) +":"+String(min3) +":00");
}
```

```

void loop () {
    CreditoButton ();
    NumPorciones ();
    CalculoDistacia ();
    CalculoDistacia2();
    PorcionesLCD ();
    DistanciaLCD ();
    DistanciaLCD2();
    HoraTreal_LCD ();
    Horas LCD ();

    // if((millis()-tiempoDistancia) > 1000) {
    //
    //     tiempoDistancia = millis ();
    // }

    if (SIM900.available()>0) {
        digital Write (ledEstado, HIGH);
        //recibido = SIM900.readString();
        c = SIM900.read();
        if (c == 'H') {hora1 = SIM900.parseInt();}
        if (c == 'M') {min1 = SIM900.parseInt();}
        if (c == 'I') {hora2 = SIM900.parseInt();}
        if (c == 'N') {min2 = SIM900.parseInt();}
        if (c == 'J') {hora3 = SIM900.parseInt();}
        if (c == 'L') {min3 = SIM900.parseInt();}
        delay (250);
        digitalWrite (led Estado, LOW);
        delay (10);
    }
}

```

```

lcd. set Cursor (9, 1);
lcd. Print (" ");
}

```

```

Date Time fecha = rtc. Now ();

```

```

if (fecha. hour () == hora1 && fecha. Minute () == min1 && fecha. second () == 0) {

```

```

    Calculo Distancia ();

```

```

    Distancia LCD ();

```

```

    if (distancia >= D comida Vacía) {

```

```

        digitalWrite (EN_M1, LOW);

```

```

        for (int i = 0; i < 200*Porciones; i++) { // 200 pasos para motor de 0.8 grados de

```

```

Angulo de paso

```

```

            digitalWrite (PULL_M1, HIGH); delay (10);

```

```

            digitalWrite (PULL_M1, LOW); delay (10);

```

```

        }

```

```

    lcd. set Cursor (9, 1); lcd.print("Y");

```

```

    envió Mensaje ("Dispensador 1: Si comió");

```

```

}

```

```

else {lcd. set Cursor (9, 1); lcd.print("N");

```

```

    envió Mensaje ("Dispensador 1: No comió");

```

```

}

```

```

CalculoDistacia2();

```

```

DistanciaLCD2();

```

```

if (distancia2 >= D comida Vacía) {

```

```

    digitalWrite (EN_M2, LOW);

```

```

        for (int i = 0; i < 200*Porciones; i++) { // 200 pasos para motor de 0.8 grados de
Angulo de paso
        digital Write (PULL_M2, HIGH); delay (10);
        digital Write (PULL_M2, LOW); delay (10);
        }

        lcd. set Cursor (10, 1); lcd.print("Y");
        envió Mensaje ("Dispensador 2: Si comió");
        }

        else {lcd. set Cursor (10, 1); lcd.print("N");
        envió Mensaje ("Dispensador 2: No comió");
        }

digitalWrite (EN_M1, HIGH);
digitalWrite (EN_M2, HIGH);
}

if (fecha. hour () == hora2 && fecha. Minute () == min2 && fecha. second () == 0) {
    CalculoDistacia ();
    DistanciaLCD ();
    if (distancia >= D comida Vacía) {
        digitalWrite (EN_M1, LOW);
        for (int i = 0; i < 200*Porciones; i++) { // 200 pasos para motor de 0.8 grados de
Angulo de paso
        digital Write (PULL_M1, HIGH); delay (10);
        digital Write (PULL_M1, LOW); delay (10);
        }

```

```

    lcd. set Cursor (9, 2); lcd.print ("Y");
    envió Mensaje ("Dispensador 1: Si comió");
}

else {lcd. set Cursor (9, 2); lcd.print("N");
    envió Mensaje ("Dispensador 1: No comió");
}

CalculoDistacia2();
DistanciaLCD2();
if (distancia2 >= DcomidaVacia) {
    digitalWrite (EN_M2, LOW);
    for (int i = 0; i < 200*Porciones; i++) {// 200 pasos para motor de 0.8 grados de
Angulo de paso
        digitalWrite (PULL_M2, HIGH); delay (10);
        digitalWrite (PULL_M2, LOW); delay (10);
    }

    lcd. set Cursor (10, 2); lcd.print("Y");
    envió Mensaje ("Dispensador 2: Si comió");
}

else {lcd. set Cursor (10, 2); lcd.print("N");
    envió Mensaje ("Dispensador 2: No comió");
}

```

```

digitalWrite (EN_M1, HIGH);
digitalWrite (EN_M2, HIGH);
}

if (fecha. hour () == hora3 && fecha. Minute () == min3 && fecha. second () == 0) {
    CalculoDistacia ();
    DistanciaLCD ();
    if (distancia >= DcomidaVacía) {
        digitalWrite (EN_M1, LOW);
        for (int i = 0; i < 200*Porciones; i++) { // 200 pasos para motor de 0.8 grados de
Angulo de paso
            digitalWrite (PULL_M1, HIGH); delay (10);
            digitalWrite (PULL_M1, LOW); delay (10);
        }

        lcd. set Cursor (9, 3); lcd.print("Y");
        envió Mensaje ("Dispensador 1: Si comió");
    }

    else {lcd. set Cursor (9, 3); lcd.print("N");
        envió Mensaje ("Dispensador 1: No comió");
    }

    CalculoDistacia2();
    DistanciaLCD2();
    if (distancia2 >= D comida Vacía) {
        digitalWrite (EN_M2, LOW);
        for (int i = 0; i < 200*Porciones; i++) { // 200 pasos para motor de 0.8 grados de
Angulo de paso

```

```

    digital Write (PULL_M2, HIGH); delay (10);
    digital Write (PULL_M2, LOW); delay (10);
}

lcd. set Cursor (10, 3); lcd.print("Y");
envió Mensaje ("Dispensador 2: Si comió");
}

else {lcd. set Cursor (10, 3); lcd.print("N");
    envió Mensaje ("Dispensador 2: No comió");
}

digitalWrite (EN_M1, HIGH);
digitalWrite (EN_M2, HIGH);
}
}

void envió Mensaje (String estado) {
    SIM900.println(Numero);//reemplazar por el número a enviar el mensaje
    delay (1000);
    SIM900.println(estado);// Reemplazar por el texto a enviar
    delay (1000);
    //Finalizamos este comando con el carácter de sustitución (?) código ASCII 26 para él
envió del SMS
    SIM900.println((char)26);
    delay (1000);
    SIM900.println();
    delay (1000);
}

```

ANEXO B – Diseño Final



