



**UNIVERSIDAD NACIONAL DE INGENIERIA**  
**Facultad de Electrotecnia y Computación**

**Trabajo Monográfico Para optar a título de Ingeniero**  
**Electrónico**  
**Carrera**  
**Ingeniería Electrónica**

**Tema:**

**“Desarrollo de prototipo de máquina expendedora basado en  
FPGA, para viviendas estudiantiles de la universidad nacional de  
ingeniería”.**

**Autor(es):**

Br. Alinton Alexander Cortez.                      Carnet: 2015-0436U

Br. Isaac Ernesto Macias Diaz.                      Carnet: 2015-0231U

Br. Kevin Rafael Balladares Roys.                      Carnet: 2015-0975U

**Tutor:**

MSc.Ing. Juan Miguel Mairena

**MANAGUA, NICARAGUA, JULIO DEL AÑO 2023**

## **Dedicatoria.**

Dedicamos primeramente el presente a Dios que, gracias a él, todo el proceso del proyecto ha sido desarrollado con éxito, brindando salud, sabiduría y todo lo necesario para que hoy día podamos culminar nuestra carrera, en donde sin su ayuda y misericordia, esto no se hubiera realizado con éxito, teniendo en cuenta que sin él no estuviéramos donde estamos.

A nuestros padres, quienes gracias a su apoyo permitió poder continuar con mi formación académica y de esta manera poder culminar con la carrera ingeniería electrónica, siendo ellos uno de los pilares fundamentales en el desarrollo de cada uno como persona y profesional.

A nuestros compañeros que, gracias a ellos con su colaboración y apoyo, logramos trabajar de manera organizada, dando como resultado la entrega del presente de manera exitosa y funcional.

Por último, pero no menos importante, a los profesores, en especial al ING. Juan Mairena e ING. Carlos Ortega (en paz, descanse), quienes nos brindaron parte de su conocimiento y experiencia como ingenieros dedicados, formando carácter en cada uno de nosotros en lo que respecta en adentrarnos al campo laboral.

## Tabla de contenido

<b>I</b>	<b>INTRODUCCIÓN.....</b>	<b>7</b>
<b>II</b>	<b>JUSTIFICACIÓN.....</b>	<b>8</b>
<b>III</b>	<b>OBJETIVO.....</b>	<b>9</b>
	3.1 Objetivo general.....	9
	3.2 Objetivos específicos.....	9
<b>IV</b>	<b>MARCO TEORICO.....</b>	<b>10</b>
	4.1 Máquinas Vending.....	10
	4.2 Tipos de Máquinas Exendedoras.....	10
	4.2.1 Máquinas expendedoras mecánicas.....	10
	4.2.2 Máquinas expendedoras electrónicas.....	11
	4.3 Arreglos de compuertas lógicas programables en sitio (FPGA).....	12
	4.3.1 Características de las FPGA.....	12
	4.3.2 Arquitectura de los circuitos FPGA.....	12
	4.3.3 Familias de FPGA.....	13
	4.4 Lenguaje de programación.....	14
	4.4.1 Características del lenguaje de programación.....	14
	4.4.2 Tipos de lenguajes de programación.....	14
	4.5 Actuadores.....	16
	4.5.1 Actuadores eléctricos.....	16
	4.6 Teclado matricial.....	19
	4.6.1 Funcionamiento de un teclado matricial.....	19
	4.7 Monitor LCD.....	20
	4.7.1 Características pantalla LCD.....	20
	4.7.2 Ventajas de monitor LCD.....	20
	4.7.3 Desventajas de monitor LCD.....	20
	4.8 Monedero electrónico.....	21
	4.8.1 Función de un monedero electrónico.....	21
	4.8.2 Característica de un monedero electrónico.....	21
	4.9 Fuente de alimentación conmutada.....	22
	4.9.1 Partes fundamentales de una fuente conmutada.....	22

4.10	Sistema de alimentación ininterrumpida (UPS).	24
4.10.1	Partes de una UPS.	24
4.10.2	Tipos de UPS.	24
4.11	Regulador de voltaje XL4015.	25
4.11.1	Especificación y características	26
V	Diseño Metodológico.	27
VI	Análisis y presentación de resultados.	30
6.1	Población y muestra de la investigación.	30
6.1.1	Muestra.	30
6.1.2	Resultados de la encuesta.	31
6.2	Fase de desarrollo y diseño.	34
6.2.1	Fase de control.	34
6.2.2	Sistema de Control de la Maquina Basada en FPGA.	37
6.2.3	Lenguaje de Programación VHDL.	42
6.2.4	Fase de sistema electrónico de detección de moneda.	44
6.2.5	Fase de diseño de sistema selector de producto.	47
6.2.6	Fase de diseño de etapa de potencia.	52
6.2.7	Fase de simulación.	57
6.2.8	Fase estructural.	62
6.2.9	Fase de ensamblaje.	65
VII	Costo del proyecto.	66
VIII	Conclusiones.	68
IX	Recomendaciones.	69
X	Bibliografía.	70
XI	Anexos.	72

## Índice de figuras.

Figura 1 Máquina expendedora mecánica (Vera Costa, 2019).....	11
Figura 2 Máquina expendedora electrónica. (Vera Costa, 2019) .....	11
Figura 3 Arquitectura básica de una FPGA. (Ortega & Garcias Perez , 2020) .	13
Figura 4 Captura programación en el lenguaje VHDL. (Canel, 2018) .....	15
Figura 5 Acción del campo magnético en una espira. (Corona Ramirez, Abarca Jiméñez, & Mares Correño, 2014).....	16
Figura 6 Motor de corriente continua. (Motorex, 2019) .....	17
Figura 7 Motor de corriente alterna. (Garrigos, 2011) .....	18
Figura 8 Parte interna de un motor paso a paso. (Ingeniería Mecafenix, 2017)	18
Figura 9 Teclado matricial con sus filas y sus columnas. (Reyes, 2016) .....	19
Figura 10 Monitor LCD y su estructura. (Soto, 2019).....	21
Figura 11 Monedero electrónico programable. (Camarasa, 2021).....	22
Figura 12 Diagrama en bloques de una fuente conmutada. (Fernandez, 2006)	23
Figura 13 Diagrama en bloques de una UPS. (Peña, 2018) .....	25
Figura 14 Regulador de voltaje XL4015. (Baker, 2020) .....	25
Figura 15 Captura porcentaje de realización de la encuesta. (questionpro, s.f.) .....	32
Figura 16 Estadísticas de la segunda pregunta de la encuesta. (questionpro, s.f.) .....	32
Figura 17 Resultados de la sexta pregunta de la encuesta. (questionpro, s.f.) .	33
Figura 18 Datos de la séptima pregunta de la encuesta. (questionpro, s.f.) .....	33
Figura 19 Diagrama en bloques de prototipo de máquina expendedora. Fuente propia. ....	34
Figura 20 Lógica de la máquina de estado del proyecto. Fuente propia. ....	36
Figura 21 Vista frontal BASYS 3. (Digilent, s.f.) .....	37
Figura 22 Diagrama sistema de alimentación BASYS 3. (Digilent, s.f.) .....	39
Figura 23 Circuito de programación ARTIX 7. (Digilent, s.f.) .....	39
Figura 24 Asignaciones de pines entre la FPGA y el dispositivo flash. (Digilent, s.f.) .....	40
Figura 25 Definición de pines VGA y circuitos BASYS 3. (Digilent, s.f.) .....	41
Figura 26 Partes y puertos de conexión de monedero electrónico. (Camarasa, 2021) .....	45
Figura 27 Conexión del monedero y la BASYS 3. Fuente propia.....	47
Figura 28 Estructura interna del teclado matricial. Fuente propia. ....	48
Figura 29 Pantalla Green Yi. Fuente ficha técnica. ....	49
Figura 30 Convertidor VGA-RCA y sus conexiones. (evoltapc, s.f.) .....	50
Figura 31 Conexión fase selección de producto. Fuente propia. ....	51

Figura 32 Motor 28BYJ-48. fuente propia. ....	53
Figura 33 Tarjeta controladora ULN2003. Fuente propia.....	53
Figura 34 Multiplexor de 16 canales CD74HC4067. Fuente propia. ....	54
Figura 35 Conexión de la BASYS 3 a los mux. Fuente propia.....	56
Figura 36 Conexión de los multiplexores y la BASYS. Fuente propia.....	57
Figura 37 Bloque decodificador del formato de moneda. Fuente propia. ....	57
Figura 38 Bloque sumador a partir de la información del decodificador. Fuente propia. ....	58
Figura 39 Bloque decodificador de la señal de teclado. Fuente propia.....	58
Figura 40 Bloque decodificador del VGA. Fuente propia. ....	59
Figura 41 Bloque de la máquina de estado. Fuente propia.....	59
Figura 42 Bloque PWM. Fuente propia. ....	60
Figura 43 Esquema de simulación de las salidas. Fuente propia. ....	61
Figura 44 esquema de simulación de las entradas. Fuente propia.....	61
Figura 45 Diseño 2D máquina expendedora. Fuente propia.....	62
Figura 46 Diseño 3D, parte frontal e interna de la máquina expendedora. Fuente propia. ....	63
Figura 47 Proceso de ensamblaje de estructura general. Fuente propia.....	64
Figura 48 Elaboración de las bandejas. Fuente propia. ....	65
Figura 49 Ensamblaje completo de la máquina expendedora. Fuente propia. .	65

### Índice de tablas.

Tabla 1 Valor de Z según el nivel de confianza. (Roque, 2010).....	31
Tabla 2 Voltajes de alimentación BASYS 3. (Digilent, s.f.) .....	38
Tabla 3 Niveles de abstracción en el diseño de circuitos mediante lenguaje HDL. (Aguirre Dobernack, 2017) .....	43
Tabla 4 Características de pantalla Green Yi. Fuente ficha técnica.....	49
Tabla 5 Tabla de verdad para el funcionamiento del multiplexor. ....	55

## I INTRODUCCIÓN.

El desarrollo del presente documento da a conocer la elaboración de un prototipo de máquina expendedora, que pretende suministrar productos de venta libre y accesos generales (Snack, mascarillas, etc.), teniendo en cuenta su funcionamiento a cualquier hora del día, sin la necesidad de un agente personal.

Por lo antes mencionado, se implementó un sistema interactivo, en el que el usuario sea capaz de realizar una gestión de compra para obtener un producto determinado, enfocando su utilización en el área de la vivienda estudiantil UNI, y dando solución al problema que se origina a raíz de no poder obtener algunos productos en horas de la noche.

El presenta escrito, inicia en una descripción teórica, en el que se detallan los conceptos básicos utilizando técnicas de investigación. Posteriormente, se presenta el análisis y presentación de resultados, en el que se da a conocer cada una de las fases que conforman el sistema, y que a su vez se detalla la selección de hardware y software que se utilizan en la elaboración del prototipo.

## II JUSTIFICACIÓN.

En Nicaragua, las máquinas expendedoras están empezando a experimentar su auge, los consumidores se familiarizan cada vez más con estos servicios, por lo que la demanda de productos va en constante aumento. Cabe destacar que este tipo de tecnología ayuda en gran manera a los usuarios como a los distribuidores de estos mismos, ya que estos trabajan de manera autónoma las 24 horas, los siete días de la semana.

En la universidad nacional de ingeniería los estudiantes, principalmente en el área de la vivienda estudiantil, se determina una necesidad de requerir diversos productos (fármacos y uso personal) que ayuden en un momento específico; aliviando o satisfaciendo en carácter personal ciertos problemas sobre todo a altas horas de la noche donde no se tiene permitido salir después de las 10:00 pm por reglamento interno de la directiva becaria. Cabe destacar que los distintos comercios aledaños a la universidad terminan sus labores después de las 7:00 pm.

Por dichas razones se propone el diseño y desarrollo de un prototipo de máquina expendedora económica que brinde productos orientados a las necesidades antes mencionadas dentro de la vivienda estudiantil, principalmente en las noches, la cual es el punto crítico de adquirirlos dentro de la universidad.

Cabe mencionar que uno de los objetivos principales del proyecto es diseñar un sistema basado en tecnología FPGA y lenguaje de VHDL, aprovechando los recursos de este hardware para un óptimo procesamiento de datos y de esta manera descartar cualquier inconveniente con las señalizaciones hacia los distintos actuadores que integran dicho proyecto.

### **III OBJETIVO.**

#### **3.1 Objetivo general**

“Desarrollar un prototipo de máquina expendedora basado en FPGA para las viviendas estudiantiles de la universidad nacional de ingeniería”.

#### **3.2 Objetivos específicos**

- Determinar un sistema electrónico de selección y entrega de producto, para una interacción efectiva entre máquina y usuario.
- Diseñar un sistema de recepción de datos que permita el conteo monetario y gestión de compra del producto.
- Implementar un sistema de decodificación que permita la detección de cualquier tipo de moneda utilizada por el usuario.

## **IV MARCO TEORICO.**

En el presente proyecto, se propone el desarrollo de un prototipo de máquina expendedora, donde se implementan técnicas sobre la ingeniería electrónica; a esto se atribuye la elaboración de un sistema de selección y entrega de producto, teniendo en cuenta la realización de un sistema de recepción de datos, donde se almacenan las variables del conteo monetario y que a su vez es decodificado para ser procesado por la FPGA junto a las variables de digitación obtenida por el teclado.

### **4.1 Máquinas Vending.**

Las máquinas expendedoras o más conocidas popularmente por el anglicismo “VENDING” son cualquier tipo de máquina que dispensa un producto de manera automática a través de un sistema de pago tradicional, como son las monedas o billetes, algunas más modernas como tarjeta.

En Japón, existen 1, 093,000 máquinas expendedoras en las que se venden todo tipo de productos: como calcetines, plumas, libros de historietas, Snack etc. (Matute, 2013)

### **4.2 Tipos de Máquinas Expendedoras.**

#### **4.2.1 Máquinas expendedoras mecánicas.**

Aquellas en que todo su funcionamiento es mecánico, sin intervención de ningún mecanismo eléctrico o electrónico. Son máquinas sencillas. (Iupercio, pinos, & uday, 2013)



*Figura 1 Máquina expendedora mecánica  
(Vera Costa, 2019).*

#### **4.2.2 Máquinas expendedoras electrónicas.**

Superan gradualmente el uso de las dispensadoras mecánicas gracias a su óptima presentación, cuenta con dispositivos electrónicos, convirtiéndolos en equipos de alta autonomía que predisponen monederos o sistemas de cobro automáticos inteligentes, permitiendo transacciones ya sea cobrando moneda o billetes, devolviendo cambio en caso de que sea necesario, cobros fiables sin necesidad de agente de venta. (Iupercio, pinos, & uday, 2013)



*Figura 2 Máquina expendedora electrónica.  
(Vera Costa, 2019)*

### **4.3 Arreglos de compuertas lógicas programables en sitio (FPGA).**

Una FPGA, es un complejo circuito integrado digital programable compuesto por bloques lógicos configurables y puertos de entrada/salida, cuya interconexión y funcionalidad puede ser programada mediante un lenguaje de descripción especializado. (Ortega & Garcias Perez , 2020)

#### **4.3.1 Características de las FPGA.**

- En una FPGA se programa su hardware, a diferencia de los microcontroladores/microprocesadores, en los que solo existe un hardware fijo y se programa su software (firmware).
- Las FPGA, además de contener puertas lógicas AND y OR, tienen memoria RAM, controladores de reloj, etc., por lo que son muy apropiadas para el diseño de sistemas embebidos con microprocesador.
- pueden reprogramarse para un trabajo específico o cambiar sus requisitos después de haberse fabricado.
- El tamaño óptimo de los módulos lógicos y los requerimientos de interconexión son completamente dependientes del tipo de aplicación que se desea implementar en el dispositivo.
- Los módulos en un FPGA se interconectan por medio de canales configurables, mediante un proceso conocido como enrutamiento.

#### **4.3.2 Arquitectura de los circuitos FPGA.**

Una FPGA se compone de una gran cantidad de elementos denominados Bloques Lógicos Configurables; también llamados Logic Cells que están interconectados por una serie de uniones programables llamadas Fabric, que sirven de rutas para el intercambio de señales entre los bloques lógicos. La interfaz que controla la comunicación entre la FPGA y los dispositivos externos está compuesta por los bloques de entrada/salida (I/O). (Ortega & Garcias Perez , 2020)

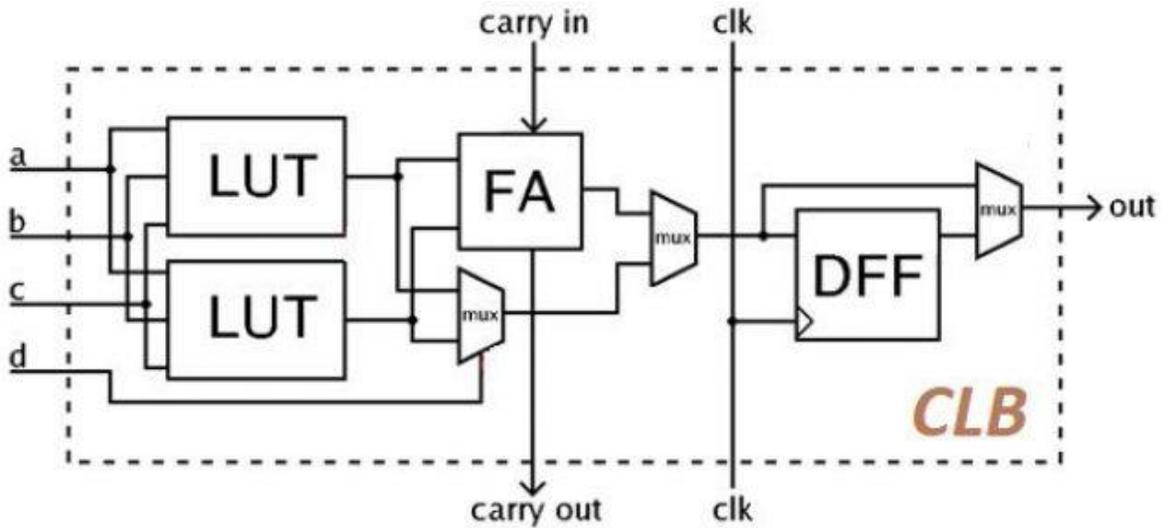


Figura 3 Arquitectura básica de una FPGA. (Ortega & Garcias Perez , 2020)

#### 4.3.3 Familias de FPGA.

Existen diferentes tipos de FPGA; cada una de estas familias presenta una arquitectura única, y también difieren en sus tecnologías básicas de fabricación. La aparición de mayores densidades de componentes en los FPGA ha motivado al desarrollo de nuevas tecnologías de programación para estos dispositivos, mientras que en todo los PLD's se han adoptado los métodos de programación de PROM's y EPROM. A continuación, se presentan algunas de las familias más representativas de estos dispositivos:

- **Arreglo de Celdas Lógicas (LCA):** son arreglos colocados alrededor del dispositivo, cada uno de los cuales puede ser configurado para usarse como una entrada, una salida en tercer estado, o una terminal de E/S bidireccional.
- **Dispositivos ACT:** Los módulos lógicos consisten de tres multiplexores de dos a uno y una compuerta OR, formando un arreglo. Debido a su simplicidad, estos módulos son más generales que los CLB que conforman a los LCA, teniendo cada uno ocho entradas y ningún registro Inter construido.
- **Arreglo Reconfigurable Eléctricamente:** él ERA está basado en tecnología RAM estática, pudiendo ser reprogramados al momento de operación.

## **4.4 Lenguaje de programación.**

Un lenguaje de programación es aquel que sirve únicamente para comunicarse con una máquina y controlar su comportamiento.

Existen una gran cantidad de lenguajes de programación creados para diferentes objetivos. Todos ellos tienen un conjunto de reglas sintácticas y semánticas que sirven para definir el tipo de datos con los que se puede trabajar con ellos y, en consecuencia, el tipo de acciones que se pueden llevar a cabo con ellos. (Assembler Institute, 2022)

### **4.4.1 Características del lenguaje de programación.**

Programar viene a ser el proceso de crear un software fiable mediante la escritura, prueba, depuración, compilación o interpretación, y mantenimiento del código fuente de dicho programa informático. Básicamente, este proceso se define aplicando lógicamente los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).
- Compilación o interpretación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.
- Desarrollo de la documentación.

(Assembler Institute, 2022)

### **4.4.2 Tipos de lenguajes de programación.**

Los tipos de lenguajes de programación se pueden clasificar principalmente como lenguajes de programación de bajo y alto nivel.

- Los **lenguajes de bajo nivel** incluyen lenguajes ensambladores y de máquina.

Un **lenguaje ensamblador** contiene una lista con instrucciones básicas y es mucho más difícil de leer que un lenguaje de alto nivel.

El **lenguaje de máquina** se entiende directamente por la unidad de procesamiento del ordenador. (Barber & Ferris, 2004)

- los **lenguajes de alto nivel** están diseñados para ser fáciles de leer y entender, permiten así a los programadores escribir el código fuente al usar palabras y símbolos lógicos y significativos.

Las siguientes actividades se pueden realizar con lenguajes de programación de alto nivel:

- Desarrollo de programas y aplicaciones.
- Desarrollo de inteligencia artificial.
- Desarrollo de base de datos.
- Desarrollo de videojuegos.
- Desarrollo de controladores e interfaz de hardware.
- Desarrollo de internet y páginas web.
- Desarrollo de scripts.

(Barber & Ferris, 2004)

```

1  -- 15.08.19 ----- Susana Canel ---- FFJK_cl_ps.vhdl
2  -- FLIP-FLOP JK, DISPARADO POR FLANCO POSITIVO,
3  -- CLEAR Y PRESET ASINCRONICOS.
4
5  library ieee;
6  use ieee.std_logic_1164.all;
7  -----
8  entity FFJK_cl_ps is
9  port(j_i   : in  std_logic;
10     k_i   : in  std_logic;
11     clk_i  : in  std_logic;
12     cl_i   : in  std_logic;
13     ps_i   : in  std_logic;
14     q_o   : out std_logic);
15  end entity FFJK_cl_ps;
16  -----
17  architecture Arq of FFJK_cl_ps is
18     signal q   : std_logic;
19  begin

```

Figura 4 Captura programación en el lenguaje VHDL. (Canel, 2018)

## 4.5 Actuadores.

Las magnitudes físicas pueden ser transformadas en otro tipo de magnitudes con propiedades que logran interactuar con el entorno; al final, dichas propiedades se reflejan como un cambio en el estado de un sistema. Las variables involucradas con estas magnitudes suelen ser de fuerzas, posición, velocidad y aceleración. Los dispositivos que realizan la transformación de las magnitudes se conocen como actuadores y se clasifican, según su funcionamiento, en eléctricos, hidráulicos y neumáticos. (Corona Ramirez, Abarca Jiménez, & Mares Correño, 2014)

En este caso se enfoca principalmente en los actuadores eléctricos, la cual son utilizados en el presente proyecto.

### 4.5.1 Actuadores eléctricos.

Estos actuadores transforman la energía eléctrica en energía mecánica. Su principio de funcionamiento se fundamenta en el efecto que produce una espira conductora que se encuentra dentro de la acción de un campo magnético y por la que circula una corriente eléctrica. En estas condiciones se experimenta una fuerza electromagnética en la espira, la cual induce un desplazamiento perpendicular a las líneas de acción del campo magnético. (Corona Ramirez, Abarca Jiménez, & Mares Correño, 2014)

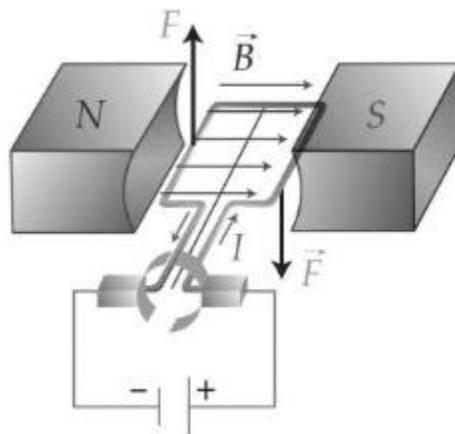


Figura 5 Acción del campo magnético en una espira. (Corona Ramirez, Abarca Jiménez, & Mares Correño, 2014)

#### 4.5.1.1 Clasificación de los actuadores eléctricos.

Es muy frecuente clasificar a los actuadores eléctricos de acuerdo con el tipo de energía eléctrica que utilizan para su funcionamiento, tal es el caso de los motores de corriente continua y los motores de corriente alterna. En este caso, nos enfocamos principalmente en los motores de corriente continua.

- **Motores de corriente continua (DC).**

Este tipo de motor consta de un rotor y un estator. El rotor es la parte móvil que proporciona la fuerza que actúa sobre la carga mecánica, mientras que el estator es la parte fija que provee el magnetismo necesario para inducir la fuerza electromotriz. Una característica muy importante de este motor es que su velocidad de rotación es proporcional al voltaje, mientras que el torque es proporcional a la corriente que circula por su devanado. (Corona Ramirez, Abarca Jiménez, & Mares Correño, 2014)

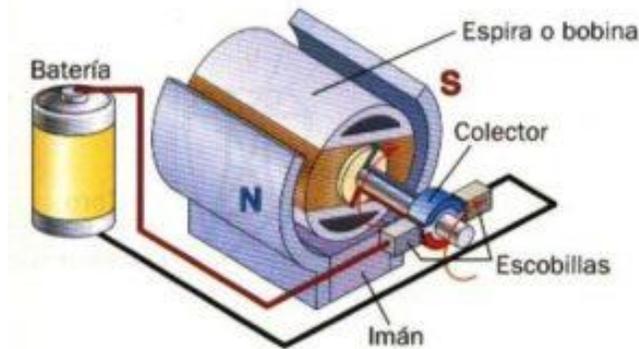


Figura 6 Motor de corriente continua. (Motorex, 2019)

- **Motores de corriente alterna (AC).**

Los motores basan su funcionamiento en la obtención de un campo magnético giratorio. Dentro de este puede haber un electroimán, que gira a la misma velocidad que el campo; este es un motor síncron. (Motorex, 2019)

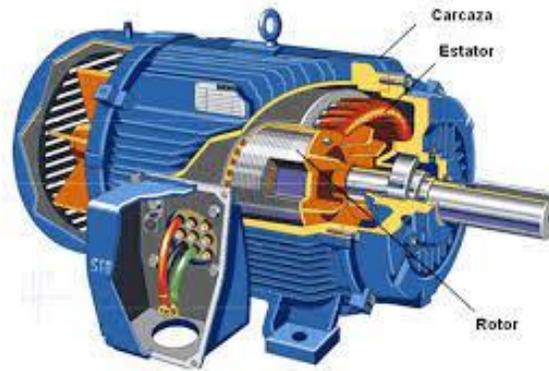


Figura 7 Motor de corriente alterna. (Garrigos, 2011)

### Motores de paso a paso (Step Motors).

El motor paso a paso está constituido por un rotor de imanes permanentes y varias bobinas que configuran el estator. El rotor se encuentra en el interior de una armadura o jaula y se encuentra magnetizada con el mismo número de polos que los que puede crear una de las bobinas.

La correspondencia entre el rotor (polos fijos) y el estator (polos variables) es la causa que provoca el giro escalonado del rotor, ya que las bobinas, arrolladas a unas masas polares, pueden ser alimentadas alternativamente, creando sobre las masas campos magnéticos con polaridad opuesta a la armadura del imán, de tal modo que se produce desplazamiento del rotor hasta la posición siguiente, es decir una fracción. (Guarella, 2011)

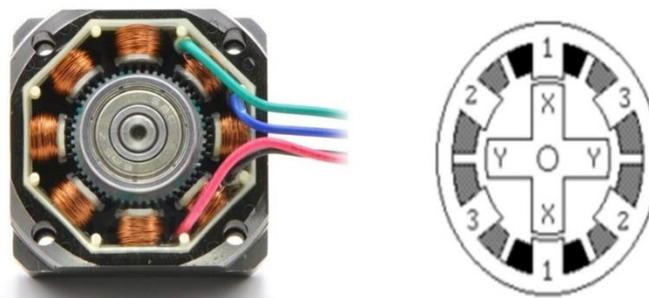


Figura 8 Parte interna de un motor paso a paso. (Ingeniería Mecafenix, 2017)

## 4.6 Teclado matricial.

Un teclado matricial es un dispositivo que agrupa varios pulsadores y permite controlarlos empleando un número de conductores inferior al que necesitaríamos al usarlos de forma individual.

Estos dispositivos agrupan los pulsadores en filas y columnas formando una matriz, disposición que da lugar a su nombre. Es frecuente una disposición rectangular pura de  $N \times M$  columnas, aunque otras disposiciones son igualmente posibles. (Allasia, 2019)

### 4.6.1 Funcionamiento de un teclado matricial.

Para este caso se encuentra a tierra un extremo del pulsador, y el otro está conectado a una entrada digital con una resistencia de pull-up.

Para leer todas las teclas se produce un barrido por filas. En primer lugar, todas las filas se encuentran conectadas a 5 V, y son definidas todas las columnas como entradas con resistencia de pull-up. Progresivamente, se pone una fila a 0 V, y se leen las entradas de las columnas. Una vez realizada la lectura se ponen nuevamente a 5 V, pasando a la siguiente fila, y volviendo a realizar el progreso hasta recorrer todas las filas.

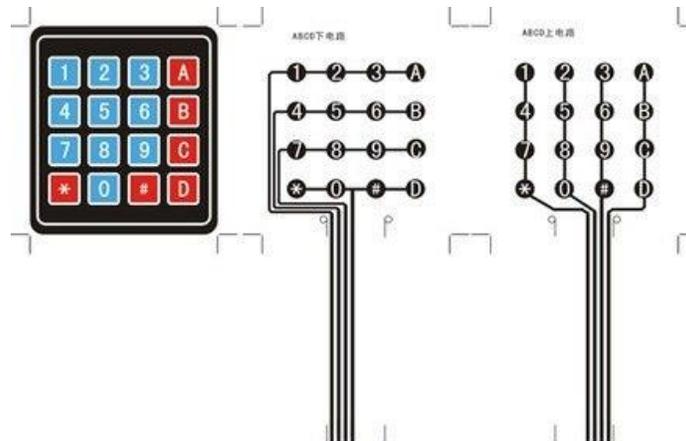


Figura 9 Teclado matricial con sus filas y sus columnas.  
(Reyes, 2016)

## 4.7 Monitor LCD.

La estructura de una pantalla LCD es un “sándwich” de muchas capas, con una emisión de luz blanca en el fondo. Toda la luz puede atravesar la pantalla LCD mientras no se le aplique carga eléctrica. Alterando el voltaje entre los electrodos se logra que la luz gire más o menos y, por lo tanto, que sea más o menos filtrada por un segundo filtro hasta llegar a la oscuridad total. Para que ofrezca imágenes en color basta con dividir cada célula LCD en tres subcelulares y aplicar diferentes filtros de color a cada una, de forma que generen luz roja, verde y azul, cuya combinación forma todos los colores. (Arranz Sancho & Conde Hernández, 2008)

### 4.7.1 Características pantalla LCD.

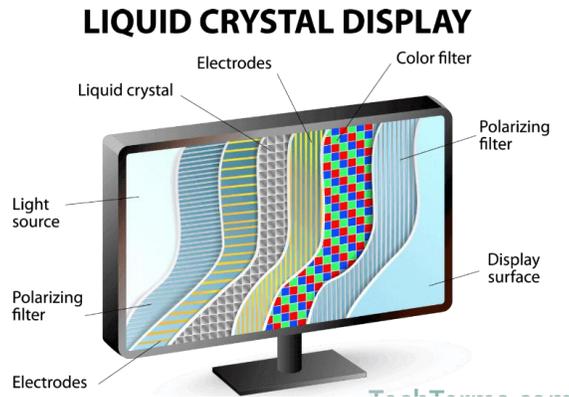
- **Tamaño:** La unidad de medida es la pulgada (“). Los más comunes son de 15.6", 17", 19", 20", 22", 24 pulgadas y monitores pequeños para cámaras de autos.
- **Tecnología:** Se le conoce como estática, ya que la pantalla no se actualiza, sino que permanece quieta hasta que la computadora envíe señal para un cambio de color
- **Resolución:** Se refiere a la cantidad máxima de píxeles que es capaz de desplegar en la pantalla. Un píxel es cada uno de los puntos de color que la pantalla muestra. (Torres, 2022)

### 4.7.2 Ventajas de monitor LCD

- El grosor es inferior por lo que pueden utilizarse en portátiles.
- La geometría es siempre perfecta, lo determina el tamaño del píxel.
- Usa menos energía.

### 4.7.3 Desventajas de monitor LCD

- Solo pueden reproducir fielmente la resolución nativa.
- Por sí solas no producen luz, necesitan una fuente externa.
- Si no se mira dentro del cono de visibilidad adecuado, desvirtúan los colores. (Torres, 2022)



*Figura 10 Monitor LCD y su estructura. (Soto, 2019)*

## **4.8 Monedero electrónico.**

Un monedero electrónico es una memoria que almacena información de dinero ingresado en un sistema. La mayoría de los monederos electromecánicos usados en los equipos expendedores están conectados a través del puerto DEX a la tarjeta controladora que actúa como maestro. Este dispositivo es un esclavo, y como tal recibe instrucciones de la tarjeta controladora. (González, 2009)

### **4.8.1 Función de un monedero electrónico.**

La función del monedero electromecánico es recibir y validar las monedas del importe para algún producto, habilitar las opciones o botones de los productos que cubre el importe validado, registrar una sola selección de algún botón, entregar el cambio si hay y llenar los tubos con las monedas que entraron y/o enviar el dinero restante a la bolsa.

### **4.8.2 Característica de un monedero electrónico.**

Los monederos electromecánicos poseen una serie de inductores que crean un campo electromagnético. Este campo es perturbado por los objetos que pasan por él, incluyendo todo tipo de monedas. Cada tipo de moneda está acuñada con un metal, tamaño, y grosores característicos que la identifica, incluyendo las monedas falsas; estas características tienen un efecto de perturbación distinto sobre el campo magnético, las distintas mediciones ayudan a identificar las monedas depositadas.

Los monederos más avanzados están programados con algoritmos que ajustan automáticamente los rangos de medición para mantener su precisión en la detección de monedas. Estos monederos también pueden transmitir la información de su estado y registros de operaciones mediante DEX o MDB, del estándar EV A-DTS. (González, 2009)



*Figura 11 Monedero electrónico programable. (Camarasa, 2021)*

#### **4.9 Fuente de alimentación conmutada.**

Una fuente conmutada es un sistema que entrega una potencia a una carga mediante la modificación de algún parámetro de control como la frecuencia o ciclo útil de la señal de salida. A diferencia de una fuente de alimentación lineal, la corriente consumida no depende enteramente de la carga, sino también de la conmutación realizada en la etapa de potencia de la fuente, como menciona. Zhang et al. (2013).

##### **4.9.1 Partes fundamentales de una fuente conmutada.**

- **Filtro EMC.** Su función es absorber los problemas eléctricos de la red, como ruidos, armónicos, transitorios, etc. También evita que la propia fuente envíe interferencias a la red.
- **Puente rectificador.** Solo deja pasar la corriente en un sentido, de modo que convierte la corriente alterna en corriente pulsante, es decir, que oscila igual que la corriente alterna, aunque únicamente en un sentido.

- **Corrector del factor de potencia.** En determinadas circunstancias, la corriente se desfasa respecto a la tensión, lo que provoca que no se aproveche toda la potencia de la red. El corrector se encarga de solventar este problema.
- **Condensador.** Amortigua la corriente pulsante para convertirla en corriente continua con un valor estable.
- **Transistor.** Se encarga de cortar y activar el paso de la corriente. De este modo se convierte a la corriente continua en corriente pulsante.
- **Controlador.** Activa y desactiva el transistor. Esta parte del circuito suele tener varias funciones, como protección contra cortocircuitos, sobrecargas, sobretensiones... También controla al circuito de corrección del factor de potencia. Además, mide la tensión de salida de la fuente, y modifica la señal entregada al transistor, para regular la tensión y mantener estable la salida.
- **Transformador.** Reduce la tensión, y además aísla físicamente la entrada de la salida.
- **Diodo.** Convierte la corriente alterna del transformador a corriente pulsante.
- **Filtro.** Convierte la corriente pulsante en continua.
- **Opto acoplador.** Enlaza la salida de la fuente con el circuito de control, pero manteniéndolos físicamente separados. (Fernandez, 2006)

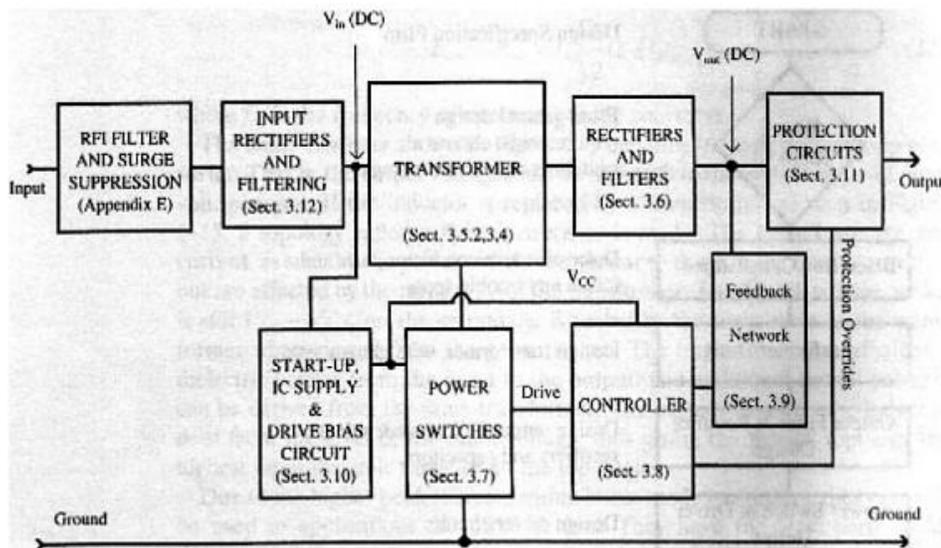


Figura 12 Diagrama en bloques de una fuente conmutada. (Fernandez, 2006)

## 4.10 Sistema de alimentación ininterrumpida (UPS).

Dicho dispositivo permite tener flujo de energía eléctrica mediante baterías, cuando el suministro eléctrico falla; de la misma manera, sirven para proteger los dispositivos que se encuentran conectados cuando hay una elevación o disminución de tensión, o sostener su funcionamiento cuando suceden pequeños cortes de energía.

### 4.10.1 Partes de una UPS.

- **Rectificador:** revisa la corriente alterna que entra al UPS y suministra corriente continua a la batería para que se mantenga cargada.
- **Batería:** suministra energía en caso de corte del flujo eléctrico. El tiempo de duración de la batería dependerá de la capacidad de almacenaje que esta tenga.
- **Inversor:** transforma la corriente continua en corriente alterna. La corriente alterna alimenta a los artefactos que se encuentran conectados a la salida del UPS.
- **Conmutador:** puede estar en dos posiciones. Permite conectar la salida con la entrada del artefacto, o con la salida del inversor. (TECSA, 2022)

### 4.10.2 Tipos de UPS.

Respecto a los tipos de **UPS**, podemos mencionar al **SPS**, que se encarga de revisar la energía que entra, pero si detecta problemas en esta energía entrante, cambia y comienza a suministrar energía por medio de la batería. Este cambio es realizado por el **SPS** en milésimas de segundos, tornándose imperceptible.

Luego, se encuentra el **UPS on-line**, el cual evita que el corto lapso de tiempo que existe entre el cambio de energía externa a la batería se interrumpa, siempre se encuentra proveyendo de energía al inversor. (TECSA, 2022)

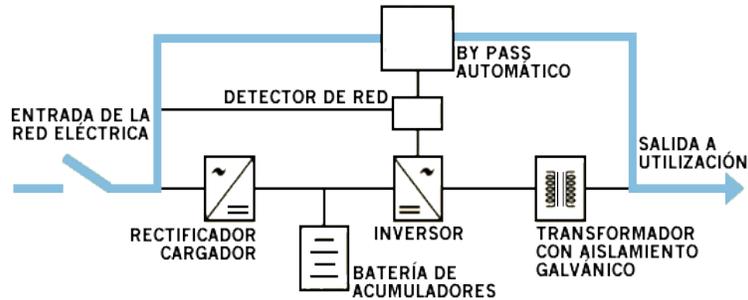


Figura 13 Diagrama en bloques de una UPS. (Peña, 2018)

#### 4.11 Regulador de voltaje XL4015.

El Convertidor DC-DC XL4015 Buck Reductor se encarga de entregar un nivel de tensión a la salida inferior al nivel de tensión que se presenta a la entrada (Buck).

El corazón del módulo se encuentra un chip XL4015, que es un convertidor CC/CC reductor PWM de frecuencia fija de 180 KHz, capaz de impulsar una carga de 5 A con alta eficiencia, baja ondulación y excelente regulación de línea y carga. El circuito de control PWM puede ajustar la relación de trabajo linealmente de 0 a 100%. Una función de protección contra sobre corriente está integrada en el interior para que cuando ocurra un cortocircuito, la frecuencia de operación se reduzca a 48 KHz. (Baker, 2020)

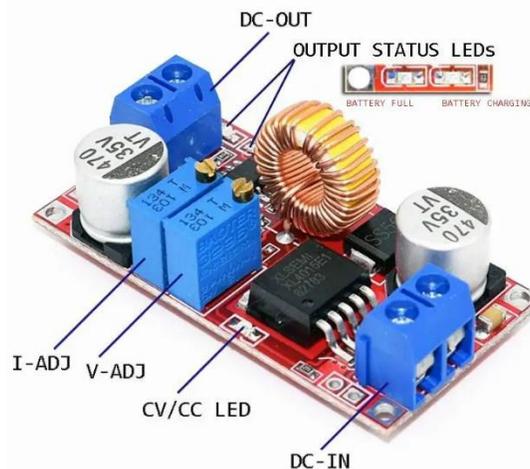


Figura 14 Regulador de voltaje XL4015. (Baker, 2020)

#### 4.11.1 Especificación y características

- Tipo: Buck, Step-Down
- Voltaje: Entrada: 8 a 34V, salida: 1.25 a 32V, regulación:  $\pm 2.5\%$
- Corriente de salida ajustable: 5A máximo
- Potencia: 75W
- Frecuencia de operación: de 48 KHz hasta 180KHz
- Onda de salida: Voltaje: 50mV, ancho de Banda: 20M
- Regulación de Carga: Promedio  $\pm 0.5\%$
- Eficiencia: 96%
- Protección: Corto circuito: Si, hasta 8<sup>a</sup>, sobre calentamiento: Si, Automático se apaga la salida.
- Protección frente a Inversión de polaridad: Ninguna
- Temperatura de funcionamiento: -40°C a 85°C
- Dimensiones: 26 mm x 52 mm x 15.5 mm  
(Baker, 2020)

## V Diseño Metodológico.

- **Ubicación del estudio.**

La ubicación del estudio del proyecto estará dirigido a la comunidad estudiantil de la Universidad Nacional de Ingeniería UNI, Departamento de Managua.

Área del estudio: Sistema de control y automatización

- **Tipo de investigación.**

Aplicada, ya que se deben buscar las estrategias y procedimientos que permitan alcanzar los objetivos planteados de forma concreta y específica.

**Experimental**, debido a la manipulación de variables para ver su efecto en el sistema y de la mayoría se tendrá el control, excepto de algunas, ya que serán controladas por factores externos como, por ejemplo: Teclado matricial, monedero electrónico etc.

- **Estrategia de investigación.**

La investigación es de carácter cuantitativo; se tomarán en cuenta aspectos relacionados con cálculos estadísticos de la comunidad estudiantil, así como también cálculos fisicomatemáticos relacionados con características del motor, análisis de datos basados en máquinas de estado aplicando teorías y leyes relacionadas con la rama de la física, la mecánica y programación.

- **Técnica de recolección de información.**

Encuestas en la comunidad estudiantil, observaciones a la operación del sistema actual y el desempeño, la información teórica se obtendrá de fuentes primarias, secundarias y terciarias.

- **Población y muestra de investigación.**

El prototipo se implementará dentro del campus universitario RUSB 100 estudiantes en la vivienda interno, siendo este dato, de suma importancia, ya que es aquí donde se tomará muestra del impacto de prototipo de máquina expendedora que se basa en FPGA para productos de venta libre.

- **Fases de desarrollo y diseño.**

En el presente estudio existirán diversos diseños que forman parte de proceso de elaboración del mismo, desde diseño de diagramas de máquina de estado para representar el funcionamiento de lógica de trabajo y Diagrama de esquemático electrónico de conexión para todos los sistemas involucrados, además de verificar el funcionamiento en simulaciones y medios gráficos para la representación del mismo, sin dejar a un lado adopción de las tecnologías a utilizar en el proyecto.

- **Fase de control.**

La fase de control se llevará a cabo a través de la programación y a su vez del manejo de la herramienta que será capaz de controlar todo el sistema de la máquina, en este caso la FPGA. Esta se programará y configurará con cada una de las sentencias capaces de manejar el sistema en general, tales como: actuadores, monitor, recolectores de datos, etc.

- **Fase diseño de sistema electrónico de detección de moneda.**

Se determinará a través de un sistema de decodificación en la programación de la FPGA, que permitirá el uso de un tipo de moneda, esto con el fin de obtener un valor determinado en la máquina y poder ser utilizado por el usuario; a su vez este en su programación hará un conteo presentando un valor y de esta manera gestionar el producto.

- **Fase diseño de Sistema de selector de producto.**

El diseño de esta fase, tiene como objetivo la interacción del usuario en la gestión de selección y compra del producto, en donde se hace uso de las señales del teclado matricial, la cual es procesada en la FPGA la cual es la encargada de proyectar en el monitor el dato que el usuario introduce.

- **Fase diseño de Sistema de etapa de potencia.**

Se encuentran los actuadores, la cual son la parte final en el funcionamiento de la máquina; son los responsables de responder a las opciones del usuario, cabe mencionar que básicamente estos son los que reciben la orden de todo el proceso de información que entra en la FPGA, para la entrega del producto.

- **Fase de simulación.**

Una vez finalizado el diseño, se debe comprobar que funcionamiento adecuado. En esta fase se determina el diagrama de circuito programado en la FPGA esta fase básicamente va de la mano de la programación del proyecto.

- **Fase estructural.**

En esta fase se determinarán cada una de las dimensiones generales de la máquina, ya que esta contara con una estructura que ayuda a un mejor manejo, desde su parte interior, que es donde se encuentra el producto y el sistema como tal, hasta su parte exterior, que permita una mejor interacción con el usuario.

- **Fase de ensamblaje (o final).**

En este caso, una vez concluida las fases anteriores, se procederá al ensamblaje de toda la máquina desde toda su estructura, la interconexión de todos los elementos electrónicos y control, hasta ver finalmente el funcionamiento completo del prototipo, como una herramienta tecnológica completa que suplirá la necesidad deseada.

## VI Análisis y presentación de resultados.

El siguiente apartado presenta cada uno de los procedimientos que se realizaron para determinar la elaboración del proyecto, desde su aceptación, hasta la presentación de los resultados, en este caso fue dividido en fases, lo cual ayuda a tener una mejor organización de las técnicas de trabajo y de esta manera tener como resultado el funcionamiento del proyecto.

### 6.1 Población y muestra de la investigación.

La población y muestra es muy importante determinarla o tenerla en cuenta, ya que a través de esta se permite desarrollar la aceptación del proyecto, en el caso de ponerlo a trabajar en un lugar determinado, esto aprobará si el proyecto podría tener un impacto considerable con respecto a la población.

#### 6.1.1 Muestra.

Se entiende como un subconjunto más o menos representativo de una población estadística, aislado del resto con fines de evaluación y estudio. Es decir, se trata de un fragmento de la totalidad de elementos a estudiar, compuesta por un número manejable de ellos, seleccionados (idealmente) al azar.

##### 6.1.1.1 Cálculo de la muestra.

Para el cálculo de la muestra se hace uso de la siguiente fórmula:

$$n = \frac{N * Z_{\alpha}^2 * p * q}{e^2 * (N - 1) + Z_{\alpha}^2 * p * q}$$

Se utiliza esta fórmula, ya que con esta se determina una muestra finita, en este caso la población no es muy grande.

Donde: **n** Tamaño de muestra buscado.  
**N** Tamaño de la población o universo.  
**Z** Parámetros estadísticos que depende el nivel de confianza.  
**e** error de estimación máximo aceptado.  
**p** Probabilidad de que ocurra el evento estudiado (éxito).  
**q (1-p)** probabilidad de que no ocurra el evento estudiado.

$$n = \frac{100 * 1.96^2 * 50\% * 50\%}{4.83\%^2 * (100 - 1) + 1.96^2 * 50\% * 50\%} \quad n = 80.60$$

Para este caso, se toma la población, la cual es de 100 y se multiplica por **Z** en donde está se muestra dependiendo el nivel de confianza que se en la siguiente figura; por otro lado, como no se conoce la probabilidad de que ocurra el evento, se toma un 50% de que ocurra y un 50% de que no ocurra. El error estimado se tomó del promedio de los errores estándar de todas las preguntas, dando un resultado de 4.83%; por tanto, el resultado total para la muestra es de **80.60**, siendo este la parte representativa de toda la población, cabe destacar que se muestra casi igual a la población, ya que este es pequeño.

*Tabla 1 Valor de Z según el nivel de confianza. (Roque, 2010)*

Nivel de confianza	Z <sub>alfa</sub>
99.7%	3
99%	2,58
98%	2,33
96%	2,05
95%	1,96
90%	1,645
80%	1,28
50%	0,674

Para este punto se determinó una encuesta, la cual permitió realizar unas preguntas a las personas que serán las más involucradas con el impacto del proyecto.

### **6.1.2 Resultados de la encuesta.**

En este caso, para disminuir un poco más de tiempo y poder tener unas respuestas acertadas sin tantas complicaciones, se procedió a utilizar la web, teniendo en cuenta el uso de la tecnología y de esta manera facilitar el trabajo, obteniendo datos más exactos y con una mejor visualización de la encuesta.

Para la realización de esta encuesta se hizo uso de una página llamada [www.questionpro.com](http://www.questionpro.com), la cual permite tener una cuenta en donde se edita y se especifica la encuesta. A su vez, esta calcula datos importantes para realizar una conclusión mucho más detalla. A continuación, se muestran capturas sobre la encuesta ya empleada.

## maquina expendedora - Dashboard



Figura 15 Captura porcentaje de realización de la encuesta. (questionpro, 2023)

La figura presenta las cantidades de visto y principalmente cuantas veces se respondió en su totalidad. Esto es de gran importancia, ya que permite saber si la muestra que se tomó anteriormente realizó la encuesta en su totalidad.

### **Q2. Tiene algunos problemas para acceder a ciertos productos especialmente en horas de la noche.**

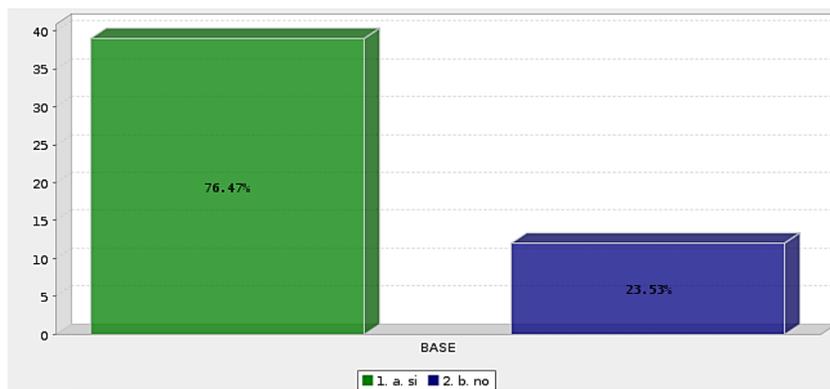


Figura 16 Estadísticas de la segunda pregunta de la encuesta. (questionpro, 2023)

Como se puede observar en la figura anterior, en un 76.47% de los encuestados cuentan con el problema para obtener algún producto en horas de la noche, esto principalmente porque tienen un horario establecido para salir de la universidad, por tanto, presenta un punto muy importante a la hora de determinar la aceptación del proyecto.

**Q7. ¿Qué producto compraría con mas frecuencia en horas de la noche?**

104212905	crocantes	104162942	Comida
104211612	maruchan	104162843	Snacks, bebidas energéticas. Galletas
104211472	snack	104119132	Cafe
104188269	te	104107226	snack gaseosa
104179727	Cafe	104030834	Medicamento cuando esté enfermo y algún tipo snack
104179030	Medicamento problema estomacal	103995903	Medicina, snacks
104178036	chocolate	103991674	Snacks
104177372	snak	103964479	Captopril
104176316	mentas	103958868	Medicamentos genéricos
104175995	maní	103958658	Bebidas
104173201	Snaks	103952532	De todo lo que tenga
104170064	Gaseosa,snacks	103943793	Condomes
104168964	Snack y medicinales	103941710	insumos medicos y snack
104167489	Para comer .	103935707	Bocadillos/Frituras
104164752	Bebidas	103935518	Acetaminophen
104163913	Frituras, agua, pastillas para la gripe	103935219	Snack
104163582	Productos personales y alimenticios	103934504	Snack
103933285	Medicamentos	103934136	Cafe
103933271	Alimentos	103933455	Cafe
103932658	Snacks	103933440	galletas
103932338	Energizantes	103933410	Cosas personales
103931635	Maruchan		

Figura 17 Resultados de la sexta pregunta de la encuesta. (questionpro, 2023)

La figura anterior presenta los posibles productos que pueden ser incluidos a la máquina expendedora, esto es muy importante, ya que de aquí es donde se puede determinar estos productos específicos de la máquina, dependiendo de las más necesitadas por los usuarios, en la cual se presenta gran importancia por productos médicos.

**Q6. Pagaría un poco más por el producto, debido al servicio de la máquina.**

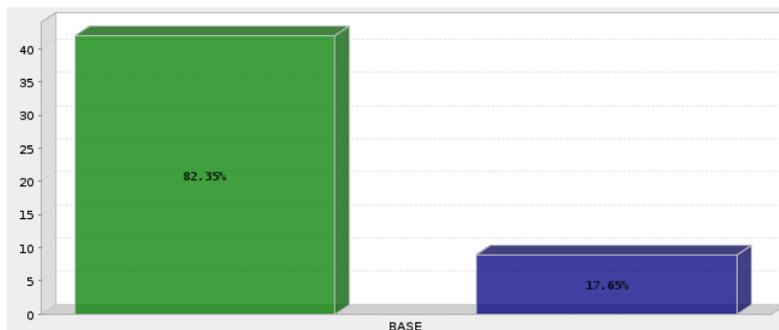


Figura 18 Datos de la séptima pregunta de la encuesta. (questionpro, 2023)

Con la aprobación de un 82.35%, se puede determinar, que el servicio de la máquina expendedora es muy bien aprobado para la muestra tomada, ya que es de aquí donde se puede saber si el proyecto va a ser rentable con respecto a la venta de productos e ingresos monetarios.

## 6.2 Fase de desarrollo y diseño.

En esta etapa, hay que tener en cuenta las sub-fases, donde se determina el orden del proceso de elaboración del proyecto, estas cumplen con los objetivos específicos, dando a conocer los componentes que serán utilizados y como estos se comportan al trabajar en conjunto.

### 6.2.1 Fase de control.

La fase de control detalla todo el funcionamiento del sistema, partiendo de un punto gráfico, que permite entender como las etapas se desempeñan a la hora de hacer su función, es por esto que, con el uso de un diagrama de bloques, la máquina de estado se presenta todo el sistema, con su lógica de funcionamiento. La importancia de esta representación, se enfoca en el punto de partida de la ideología del algoritmo a la hora de programar las etapas.

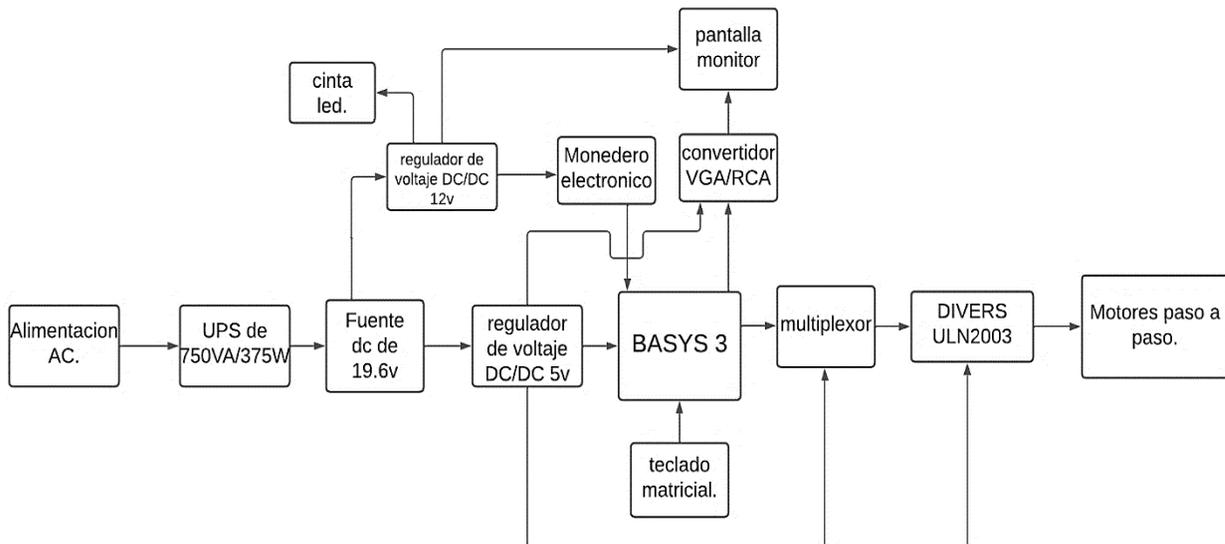


Figura 19 Diagrama en bloques de prototipo de máquina expendedora. Fuente propia.

En la presente figura se muestra el diagrama en bloques del proyecto en su totalidad, este parte del suministro eléctrico AC, a su vez se conecta una UPS (sistema de alimentación ininterrumpida) la cual, no solo se encarga de suministrar energía a todo el sistema del proyecto, sino que también permite tener protección a la hora de que ocurra una elevación o disminución de tensión, esta permite también mantener su funcionamiento cuando suceden pequeños cortes de energía. Por otro lado, a esta se conecta la fuente principal que consta de un voltaje de 19.6v DC que se encargaran de alimentar a dos reguladores de voltajes.

La fuente principal, con la ayuda de estas tarjetas reguladoras, suministra dos tipos de voltajes uno de 5V, que es la destinada para alimentar el sistema de control principal como lo es la BASYS 3, los drivers encargados de controlar los motores paso a paso, el multiplexor encargado de aumentar las salidas de los pines de la BASYS y el convertidor VGA/RCA, encargado de convertir la imagen transmitida hacia la pantalla; por otro lado, el regulador dos que suministra un voltaje 12V DC, es necesario para hacer funcionar el monedero electrónico, la pantalla y la cinta led que iluminara la parte interna de la máquina.

El sistema cuenta con un control principal (BASYS 3) de donde, recibe dos tipos de señales, la del monedero y la del teclado matricial, estos dispositivos se encargan que el usuario interactúe con la máquina; al recibir estas dos señales la BASYS procesa la información que se digita en el teclado y la cantidad de dinero detectado por el monedero, que es transmitida a la pantalla; a su vez teniendo en cuenta la opción del usuario manda una señal en binario al multiplexor que es el encargado de activar el motor necesario para dispensar el producto.

#### **6.2.1.1 Máquina de estado.**

Una máquina de estado, es un modelo de comportamiento de un sistema con entradas y salidas en donde las salidas dependen no solo de las señales de entradas actuales, sino también de las anteriores.

Este proyecto, esta es representado como una máquina de estado finito (FSM), ya que al ponerse en marcha el proceso, este al final se terminará ejecutándose teniendo un número finito de estados, en este caso 3.

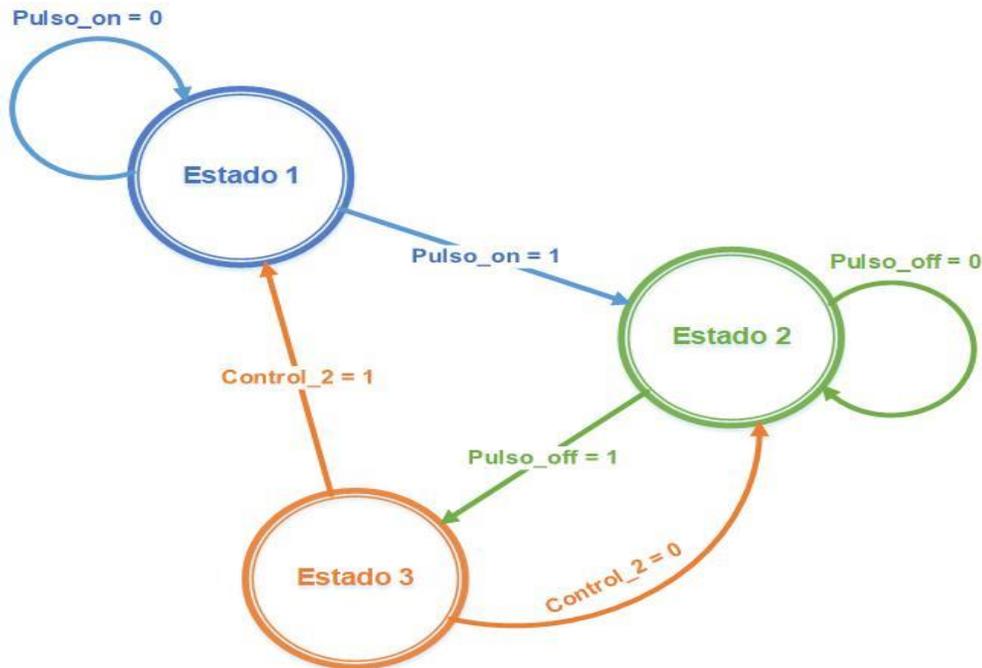


Figura 20 Lógica de la máquina de estado del proyecto. Fuente propia.

La explicación de los estados se detalla de la siguiente manera:

- **Estado 1:** Estar en este estado va en dependencia de **pulso\_on**, este representa las monedas utilizadas dependiendo de la denominación, en este punto el **pulso\_on=0** representa que no se ha insertado moneda y se determina la pertenencia en el estado 1. En el caso de que se detecte una moneda, manda un **pulso\_on=1** de un bit, que hace que se desplace al estado 2.
- **Estado 2:** este estado presenta una señal de control de 1 bit, la cual es **pulso\_off**, si esta señal equivale a un 0 lógico entonces el sistema está a esperas que se introduzca el primer dígito del teclado matricial; cuando se introduce el segundo dígito del teclado entonces **pulso\_off** tomará el valor de 1, lo cual indica que el usuario ya terminó de seleccionar su producto y desplazara el algoritmo de la máquina de estado al estado siguiente.

- **Estado 3:** Representa la relación precio del producto y cantidad dinero disponibles, en este caso si la selección del producto no es la adecuada a la cantidad monetaria manda un pulso **control\_2=0** en donde se retorna al estado anterior y se renueva el conteo de las monedas. Por otro lado, si en este estado 3 se selecciona con respecto a la cantidad dinero disponible, esta manda un pulso **control\_2=1** que representa la entrega del producto, en donde una vez entregado retorna a su estado inicial y vuelve a realizar el mismo proceso.
- **Características de la máquina de estado.**
  1. La máquina solo acepta múltiplos de P1 (p1).
  2. La máquina puede dispensar basándose en la cantidad de códigos que esta posee, cada código existente determina variaciones de producto.
  3. Tiene un botón Cancelar.
  4. Solo una entrada puede estar activa a la vez.
  5. Un producto se puede dispensar en un ciclo de reloj.
  6. Si se insertan 5 o 1 Córdoba, el dinero será incrementado hasta tener la cantidad deseada del costo de producto.

### 6.2.2 Sistema de Control de la Máquina Basada en FPGA.

La plataforma para desarrollar el sistema de control se basará en FPGA. Se trata de un dispositivo semiconductor de circuitos digitales que contiene bloques de lógica cuya interconexiones y funcionalidad puede ser configurada mediante un lenguaje de programación especializado, en este caso, el VHDL.



Figura 21 Vista frontal BASYS 3. (Digilent, 2019)

El uso de este controlador, se basa a sus características y principalmente su forma de programación en hardware, ya que abarca un amplio campo en lo que respecta al desarrollo de sistemas robustos. Entre las características principales están:

- **Alimentación BASYS 3.**

La placa usa un bus de energía de entrada, que se puede controlar desde un cable USB mediante suministro de enchufe de pared de 2,1 mm con centro positivo de 5 V CC a 15 V CC o desde un paquete de baterías. El circuito USB siempre recibe alimentación del cable USB; si no se conecta ningún cable USB, el circuito USB se deja sin alimentación.

*Tabla 2 Voltajes de alimentación BASYS 3. (Digilent, 2019)*

<b>Fuente.</b>	<b>Circuitos.</b>	<b>Dispositivos.</b>	<b>Corriente (max/typical)</b>
3.3V	FPGA I/O, USB ports, Clocks, Flash, PMODs	IC10: LTC3633	2A/0.1 to 1.5A
1.0V	FPGA Core	IC10: LTC3633	2A/ 0.2 to 1.3A
1.8V	FPGA Auxiliary and Ram	IC11: LTC3621	300mA/ 0.05 to 0.15A

Algunos dispositivos requieren suministros de 1.0 V y 1.8 V además del suministro principal de 3.3 V, y estos suministros adicionales son creados por reguladores que toman su entrada del suministro principal de 5.0 V externos. Los suministros primarios son generados por reguladores de conmutación altamente eficientes de Linear Technology.

Esta placa también puede suministrar energía a una placa periférica conectada a un conector Pmod o al conector de expansión grande de 100 pines.

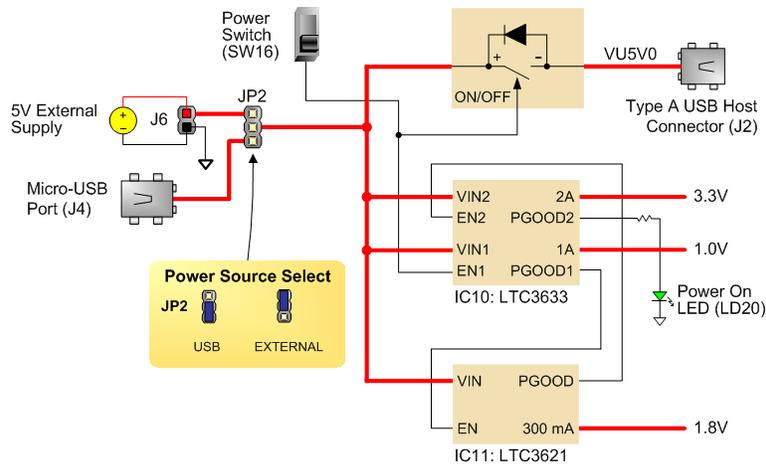


Figura 22 Diagrama sistema de alimentación BASYS 3. (Digilent, 2019)

- **Configuración de FPGA y Plataforma Flash.**

El FPGA en la placa BASYS 3 debe ser configurado (o programado) por el usuario antes de que pueda realizar cualquier función, pero para configurar la FPGA tiene la opción de 3 formas.

1. Una PC puede usar el circuito USB-JTAG de Digilent (puerto J4, etiquetado como "PROG") para programar el FPGA en cualquier momento que esté encendido.
2. Un archivo almacenado en el dispositivo flash serié no volátil (SPI) se puede transferir a la FPGA mediante el puerto SPI.
3. Se puede transferir un archivo de programación desde una memoria USB conectada al puerto USB HID.

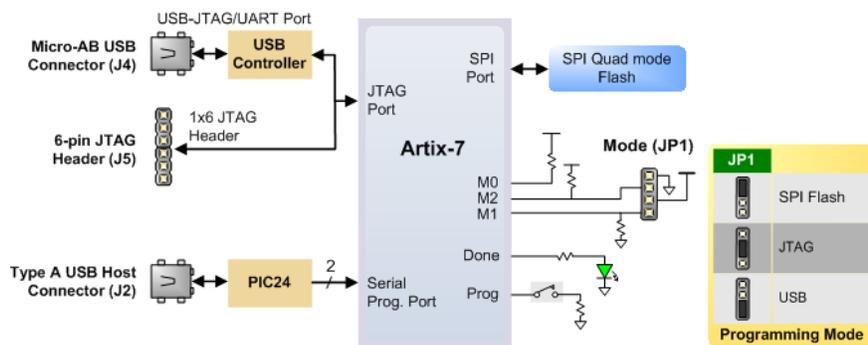


Figura 23 Circuito de programación ARTIX 7. (Digilent, 2019)

- **Memoria.**

Contiene un dispositivo Flash serié no volátil de 32 Mbit, que se conecta a la FPGA Artix 7 mediante un bus SPI dedicado de modo cuádruple (x4). Los archivos de configuración de FPGA se pueden escribir en Quad SPI Flash, y las configuraciones de modo están disponibles para hacer que FPGA lea automáticamente una configuración de este dispositivo al encenderlo.

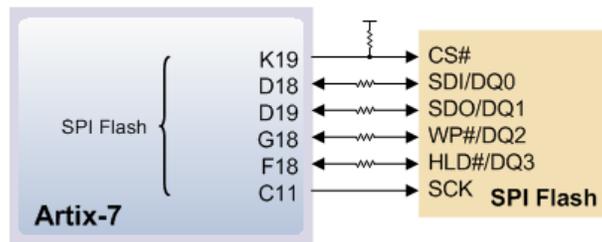


Figura 24 Asignaciones de pines entre la FPGA y el dispositivo flash. (Digilent, 2019)

- **Relojes.**

Incluye un solo oscilador de 100 MHz conectado al pin W5 (W5 es una entrada MRCC en el banco 34). El reloj de entrada puede impulsar MMCM o PLL para generar relojes de varias frecuencias y con relaciones de fase conocidas que pueden ser necesarias a lo largo de un diseño. Algunas reglas restringen qué MMCM y PLL pueden ser controlados por el reloj de entrada de 100 MHz.

- **Puerto USB.**

Los controladores de puerto USB-COM convierten los paquetes USB en datos de puerto serial/UART. Los datos del puerto serie se intercambian con la FPGA mediante un puerto serie de dos hilos (TXD/RXD). El FT2232HQ también se usa como controlador para el circuito USB-JTAG de Digilent, pero las funciones USB- UART y USB-JTAG se comportan de manera completamente independiente entre sí. La combinación de estas dos funciones en un solo dispositivo permite que Basys 3 se programe, se comunique con UART y se alimente desde una computadora conectada con un solo cable Micro USB.

- **Puerto VGA.**

La BASYS utiliza 14 señales FPGA para crear un puerto VGA con 4 bits por color y las dos señales de sincronización estándar (HS: sincronización horizontal y VS: sincronización vertical). Las señales de color utilizan circuitos divisores de resistencia que funcionan junto con la resistencia de terminación de 75 ohmios de la pantalla VGA para crear 16 niveles de señal en cada una de las señales VGA roja, verde y azul.

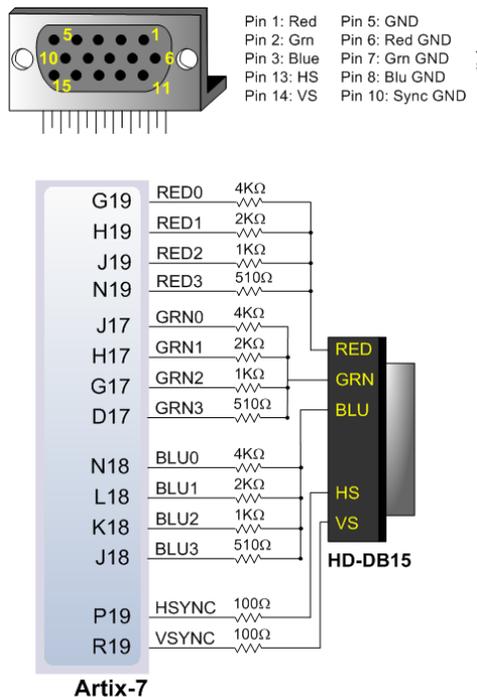


Figura 25 Definición de pines VGA y circuitos BASYS 3. (Digilent, 2019)

Este circuito produce señales de color de video que proceden en incrementos iguales entre 0 V (totalmente apagado) y 0,7 V (totalmente encendido). Usando este circuito, se pueden mostrar 4096 colores diferentes, uno para cada patrón único de 12 bits. Se debe crear un circuito controlador de video en el FPGA para controlar las señales de sincronización y color con la sincronización correcta para producir un sistema de visualización que funcione.

**Toda esta información fue obtenida de:** (Digilent, 2019)

### 6.2.3 Lenguaje de Programación VHDL.

Este proyecto hace uso del lenguaje VHDL, por ser un lenguaje de alto nivel de abstracción que ha sido aceptado como un estándar de diseño. Estos lenguajes son los llamados lenguajes de descripción de hardware (HDL "Hardware Description Language"), entre los que se encuentran: VHDL, Verilog y ABEL.

#### 6.2.3.1 Ventajas de los Lenguajes HDL.

Existen muchos motivos para usar lenguajes HDL en la fase de diseño de un sistema electrónico digital. Sus principales ventajas son:

- Posibilidad de verificar el funcionamiento del sistema dentro del proceso de diseño, sin necesidad de implementar el circuito.
- Las simulaciones en la fase de diseño permiten tomar decisiones en cuanto a la arquitectura y a los cambios a realizar.
- Se apoyan en las herramientas de síntesis, que convierten una descripción en HDL y la optimizan según la tecnología utilizada.
- Permite la migración de diseños de una tecnología a otra sin pérdida de funcionalidad.
- Las herramientas de síntesis proveen la forma de transformar el lenguaje HDL en un circuito lo más compacto y rápido posible. (Aguirre Dobernack, 2017)

#### 6.2.3.2 Dominio Descriptivos y Niveles de Abstracción en VHDL.

VHDL soporta varios estilos para la descripción de diseño. Estos, denominados dominios descriptivos:

- **El dominio de comportamiento:** Domino o algorítmico es el nivel de abstracción más elevado que soporta VHDL.
- **El dominio del flujo de datos:** Describe el circuito en términos de como los datos se mueven a través del sistema y forma en que la información fluye a través de los registros del circuito.

- **El dominio Físico:** Se usa para describir circuitos en términos de sus componentes. Puede ser usado para crear una descripción de bajo nivel, como la descripción a nivel de transistor o una descripción a nivel de diagrama de bloques. (Aguirre Dobernack, 2017)

Tabla 3 Niveles de abstracción en el diseño de circuitos mediante lenguaje HDL. (Aguirre Dobernack, 2017)

Nivel de Abstracción	Valores	Medidas
<b>Sistema</b>	Relaciones entre subsistemas, sincronización y protocolos.	Ancho de banda, MIPS.
<b>Algorítmico</b>	Estructuras abstractas. Se usan las dependencias en lugar del tiempo.	Latencia, cadencia de datos, número de módulos.
<b>RT (Register Transfer)</b>	Palabras con valores discretos. Control y procesamiento en tiempo discreto.	Tiempos de ciclo, márgenes y puertas equivalentes.
<b>Lógico</b>	Valores lógicos. Computación en tiempo continuo.	Tiempos de conmutación, Skew y áreas equivalentes.
<b>Circuito</b>	Valores continuos. Todo es electrónica en tiempo continuo.	Tiempos de subida, bajada y consumos de área.

### 6.2.3.3 Características Generales de VHDL

VHDL contiene elementos que son parecidos a los encontrados en los lenguajes de programación, aunque hay que tener presente el hardware que generan al momento de ser usados.

- Uso de variables y constantes, para almacenar valores que pueden cambiar o valores que no cambian durante la ejecución del programa, respectivamente.
- Funciones y procedimientos, que son segmentos de código que pueden ser llamados desde distintos puntos
- Uso de bibliotecas y paquetes, en donde se puede almacenar las definiciones de componentes, funciones, procedimientos o constantes, con la finalidad de tener componentes reutilizables, y así usar el mismo código para más de un proyecto.

- Instrucciones de entrada y salida, usadas solo durante la simulación. Con este tipo de instrucciones es posible leer datos de un archivo, los cuales pueden ser los valores de una señal de entrada. (Aguirre Dobernack, 2017)

#### **6.2.4 Fase de sistema electrónico de detección de moneda.**

La detección de moneda para el proyecto es de suma importancia, ya que es aquí donde se obtiene uno de los parámetros esenciales para la gestión del producto, como es el valor y la cantidad de monedas. En este, se explica cómo fue, está elaborada y qué componentes se utilizaron.

##### **6.2.4.1 Identificación de pago.**

Como sistema de identificación de pago, tras una amplia búsqueda, se ha elegido colocar una solución del modelo JY-936, descartando otras soluciones existentes, debido a que en este modelo encontramos el sistema ideal para la gestión de moneda de la máquina expendedora, este monedero electrónico sustituye a los que aceptan de un solo valor y no lleva de muestra insertada. Se programa haciendo que reconozca las diferentes monedas que se utilizaran, indicándole la cantidad de pulsos (créditos) que cada una de ellas representa. Si se inserta una moneda, para la cual no está programado, la rechazará y el cliente la recuperará.

- **Características.**
  1. Este monedero se alimenta de 12V DC y entrega un pulso eléctrico de unos 5 VDC a través de su pin "COIN", cada vez que se inserta una moneda correcta en su interior.
  2. También está dotado de un pin de salida denominado "COUNTER" que se encarga de enviar un pulso, que pueda ser leído por un contador electrónico externo, en nuestro caso no se utiliza por que el contador se genera internamente dentro de la BASYS.

3. Mediante el interruptor NC/NO podemos seleccionar en qué niveles lógicos va a trabajar el pulso de salida del monedero, es decir, o estado normalmente abierto (NO) o normalmente cerrado (NC).
4. mediante otro interruptor, podemos seleccionar la velocidad correcta de aceptación de la moneda, en definitiva, la velocidad del pulso de salida (pin COIN). Esta velocidad de pulso se encuentra entre los 100 milisegundos (velocidad lenta), 50 milisegundos (velocidad media) y 30 milisegundos (velocidad rápida).

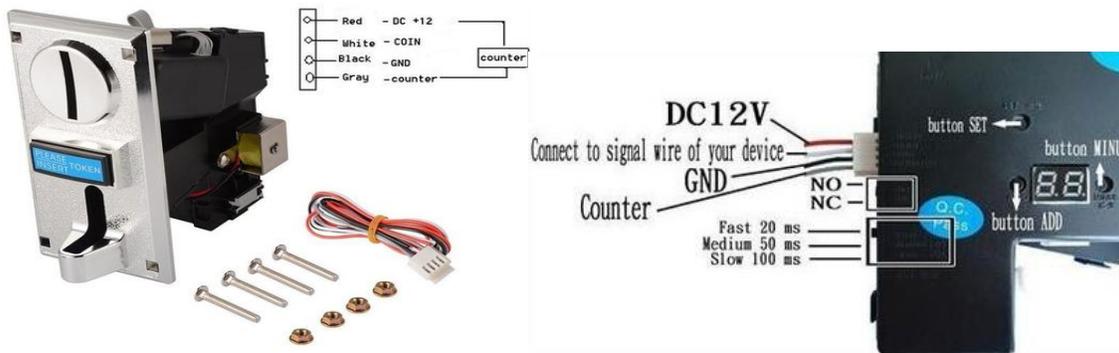


Figura 26 Partes y puertos de conexión de monedero electrónico. (Camarasa, 2021)

#### 6.2.4.2 Manual de instrucciones para programación de monedas.

1. Aplicar tensión de 12V al cable rojo (positivo) y negro (negativo)
2. Pulsar los botones ADD y MINUS al mismo tiempo durante unos segundos hasta que aparezca la letra A en la pantalla y luego soltarlos.
3. Pulsar el botón SET durante unos segundos hasta que aparezca la letra E y soltar
4. Pulsar el botón ADD mostrará del 1 al 6 dependiendo de cuantas monedas quiera usar al mismo tiempo y luego mantener pulsado SET durante unos segundos hasta ver H1 en la pantalla.
5. Pulsar el botón ADD para indicar cuantas muestras queremos hacer de cada moneda queremos hacer (se recomienda 30, luego meterás 30 veces la misma moneda para que aprenda el monedero) y luego mantener pulsado el

botón ADD unos segundos para finalizar. Aparecerá indicado P1 en la pantalla.

6. Pulsar el botón ADD tantas veces como sea necesario para indicar la duración del pulso. El máximo es de 50 milisegundos. Se recomienda usar el valor 50. Para aceptar, mantener pulsado el botón SET y se mostrará F1.
7. Pulsar el botón tantas veces como sea necesario para seleccionar la precisión (número de entre 1 y 30, siendo 1 la mayor precisión y 30 la peor). Se recomienda un valor entre 5 y 10. Mantener pulsado SET y empezaremos a configurar meter la primera moneda.
8. Ahora insertar la misma moneda 30 veces seguidas para que el monedero aprenda. Se indicará mostrando la letra A en pantalla, indicando que el proceso ha finalizado, repetimos el paso 5 (H1) y 7 (F1)
9. Mantener pulsado el botón SET hasta que se muestra la letra E.
10. Apagar y encender de nuevo el monedero y los cambios quedarán guardados.

#### **6.2.4.3 Conexión del monedero con BASYS 3.**

Se conecta al pin JA1 de la placa, en donde a la hora de programar hay que tener en cuenta, que este pin tiene que ser entrada para que la BASYS pueda recibir el pulso de 5v que genera el monedero a la hora que se le ingresa una moneda o varios pulsos en el caso de que sean varias monedas o una moneda de 5 córdobas, la cual este pulso será recibido y procesado internamente por un contador, un sumador y posteriormente la máquina de estado internamente programado.

Es muy importante recalcar, que el monedero está configurado como normalmente cerrado, esto debido a que se está trabajando con un pulso en alto donde de esta manera pueda ser detectado por la BASYS. Este también trabaja con el switch de velocidad de detección de moneda en fast; esto determina que detectara rápidamente las moneda.

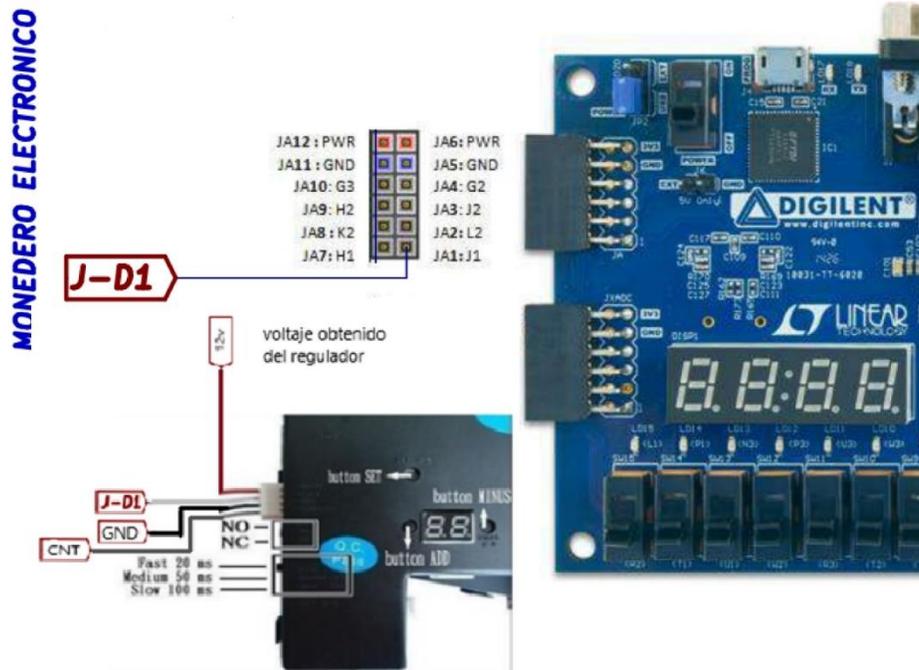


Figura 27 Conexión del monedero y la BASYS 3. Fuente propia.

### 6.2.5 Fase de diseño de sistema selector de producto.

El sistema selector es el encargado de brindar la interacción del usuario con la máquina, es por esto, que en conjunto con el sistema identificador de pago y el monitor permitirán al usuario poder obtener de un producto determinado.

#### ¿Cómo se selecciona el producto?

Por la facilidad y sencillez a la hora de elegir un producto se utiliza **un teclado matricial** alfanumérico rígido. Este teclado funciona igual que el de membrana, es una opción de reemplazo rígida. Tiene una configuración 4x4 y un grado IP54 de aislamiento.

Entre sus características principales tenemos las siguientes:

- Teclas: 16 (4 x 4) 0 a 9, A a D, \*, #.
- Color: Negro.
- Material: ABS.
- Incluye: 1 Teclado numérico.
- Grado: IP54.

- Voltaje de funcionamiento: 3.3/5V.
- Ciclos: 1 millon.
- Dimensiones: 65.04x68.69x09.53 mm.
- Peso: 21 g.

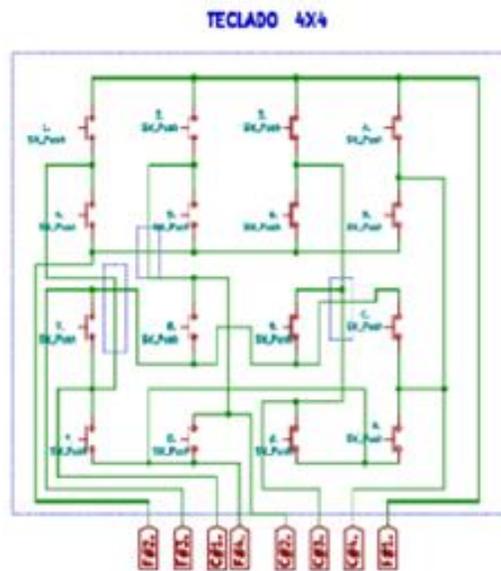


Figura 28 Estructura interna del teclado matricial. Fuente propia.

### ¿Cómo interactúa el usuario con la maquina?

Para interacción visual del usuario con la máquina, se eligió un monitor LCD modelo Green Yi-8, se puede conectar con su equipo de video como VCD, DVD, GPS, receptor de satélite, En esta era de tecnología este panel proporcionara al usuario un gran confort debido a su facilidad y sencillez a la hora de elegir un producto.

Es una parte clave de un sistema de visión, así que definitivamente mejorará su interacción con el sistema. Tiene un tamaño de 5 pulgadas, proporciona una mejor claridad, muestra una mejor imagen que otros monitores, mayor brillo y resolución.

Tabla 4 Características de pantalla Green Yi. Fuente ficha técnica.

Marca	Green Yi
Dimensiones LxAxP	5.12x3.23x0.47 Pulgadas
Tamaño	5 pulgadas
Tecnología de pantalla	LCD
Tipo de instalación	Montaje en superficie o parabrisas
Resolución	800x480
Voltaje	12 voltios
Tipo de conector	RCA

Este tipo de pantalla cuenta con dos entradas de vídeo para la pantalla. El enchufe amarillo es para conectarse a cámara de visión, también ajusta el brillo del monitor, contraste y saturación. Cómo funciona este monitor de pantalla de respaldo AHD, cuando se conecta a la fuente de alimentación, se muestra la pantalla azul si no hay entrada de señal de vídeo, después de varios segundos, la pantalla se apagará automáticamente.



Figura 29 Pantalla Green Yi. Fuente ficha técnica.

En este caso, la BASYS 3 al tener una salida VGA y la pantalla una entrada RCA, se hace uso de un convertidor para poder visualizar los datos en el monitor.

- **Convertidor VGA-RCA.**

Este conversor de video es un adaptador de gran importancia que permite conectar una salida de video analógica RCA/S-Video aun monitor/TV con entrada VGA.

Algunas de sus características importantes son:

1. Una entrada de señales VGA, una salida de señales VGA.
2. Una salida de RCA, una salida de S-Video.
3. Un ACPI de 5V
4. Resoluciones de las señales VGA: 640x480 @ 60/72/75 / 75HZ, 800x600 @ 60 / 75Hz, 1024x768 @ 60 / 75Hz, 1280x1024 @ 60Hz.
5. Al mismo tiempo, emita las señales VGAVideoS-Video.
6. Muestre simultáneamente imágenes en el monitor y en el televisor.
7. Apoye el modo de visualización OVERSCAN y UNDERSCAN.
8. Conveniente de llevar porque es ligero, delgado, corto y pequeño.
9. Tamaño: 100 x 56 x 36mm.

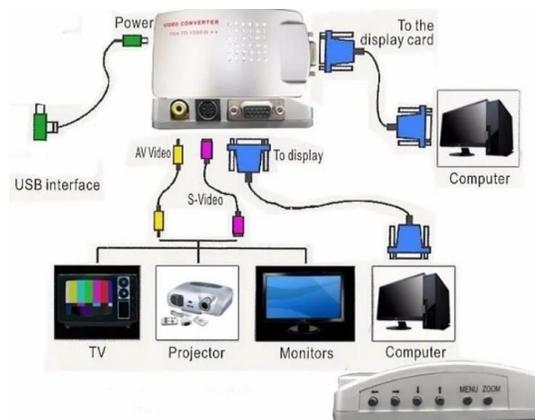


Figura 30 Convertidor VGA-RCA y sus conexiones. (evoltapc, s.f.)

### 6.2.5.1 Conexión de la fase de selección de producto.

La conexión de esta fase, parte de saber como está configurado el teclado matricial teniendo en cuenta las filas-columnas y de esta manera conocer el estado de los botones; en este caso cada tecla es un pulsador conectado a una fila y a una columna que cuando es pulsada se cierra una conexión única entre una fila y una columna; partiendo de esto, el teclado a utilizar cuenta de la siguiente configuración:

- Pin1: fila 2
- Pin2: fila 3
- Pin3: columna 1
- Pin4: fila 4
- Pin5: columna 2
- Pin6: columna 3
- Pin7: columna 4
- Pin8: fila 1

La conexión en la BASYS se determinan dividiendo las filas y las columnas en los pines, donde estos son definidos en el usf como entradas partiendo de la siguiente manera:

Fila 1 hasta la fila 4 se conectan del pin JA7 hasta el pin JA10, por otro lado, de la columna 1 hasta la columna 4 se conectan del pin JXAC7 hasta el pin JXAC10. Internamente el decodificador es el encargado de recibir esta señal y que a su vez esta sea reflejada. Cabe destacar que al ser una fase en la que el usuario interactue con la maquina, se requiere la visualización de los valores insertados, es por esto que se envia a través de la conexión VGA la señal procesada, para esto se elaboraron arreglos basados en los datos que queremos que muestre con el fin de poder ver lo seleccionado con el teclado (la señal VGA es procesada o convertida, ya que la pantalla consta de una entrada RCA).

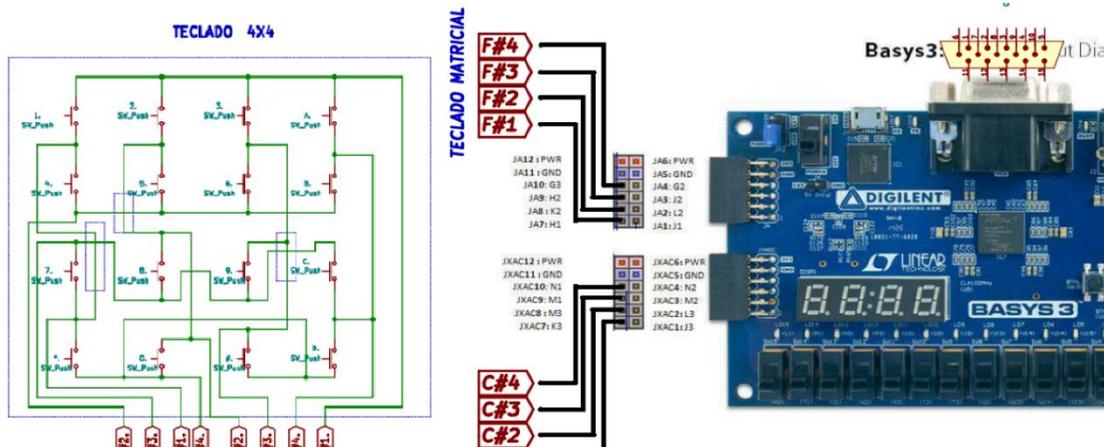


Figura 31 Conexión fase selección de producto. Fuente propia.

## **6.2.6 Fase de diseño de etapa de potencia.**

El diseño de esta etapa se considera la parte final del proyecto, donde se determina la entrega del producto al usuario, para esto se especifican los actuadores, las cuales son los encargados en desplazar el producto y entregarlo, hay que tener en cuenta que la selección del tipo de motor es de suma importancia, ya que de este depende: la calibración y el movimiento del colcho.

### **6.2.6.1 Sistema de actuadores.**

El sistema mecánico será mediante motor paso a paso modelo **28BYJ-48**; un actuador electromagnético rotatorio que convierte mecánicamente entradas de pulso digitales a movimiento rotatorio incremental de la flecha, entre sus características se encuentran las siguientes:

1. Motores sin escobillas digitalmente controlados que rota un número específico de grados (un paso).
2. Alta precisión y bajo torque.
3. El consumo de energía del motor es de alrededor de 240mA.
4. Su tipo de control se selecciona según el uso (paso completo, fase simple, doble fase, medio paso y micro paso).

### **¿Como funcionan estos motores paso a pasos?**

Estos motores paso a paso utilizan una rueda dentada (que tiene 32 dientes) y cuatro electroimanes para hacer girar la rueda un "paso" a la vez. Cada pulso ALTO enviado energiza la bobina, atrae los dientes más cercanos de la rueda dentada e impulsa el motor un paso.

El motor tiene cuatro bobinados que son excitados en una precisa secuencia para hacer girar. Por eso, en la secuencia de medio paso (half step, en inglés), la variación de señales necesaria para el funcionamiento en uno u otro sentido tiene 8 combinaciones.

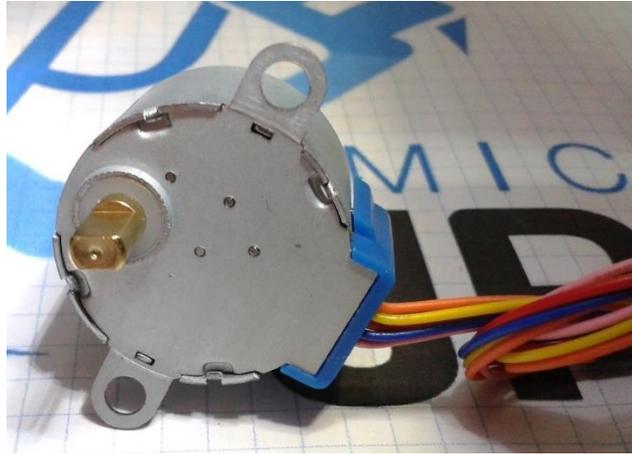


Figura 32 Motor 28BYJ-48. fuente propia.

Cabe destacar, que este tipo de motores, hace uso de un controlador para su funcionamiento, el cual describe lo siguiente:

- **Controlador ULN2003.**

Este es uno de los circuitos integrados de control de motor más comunes, que consta de una matriz de 7 pares de transistores Darlington; cada par es capaz de conducir cargas de hasta 500 mA y 50 V. Cuatro de los siete pares se utilizan en este tablero.

En su descripción, esta placa tiene cuatro LED que muestran actividad en las cuatro líneas de entrada de control (para indicar el estado de avance). Proporcionan una buena visual al girar. La placa también viene con un puente ON/OFF para aislar la alimentación del motor paso a paso.



Figura 33 Tarjeta controladora ULN2003.  
Fuente propia.

- **Multiplexor analógico CD74HC4067.**

Se hizo uso de este módulo, ya que permite controlar un mayor número de elementos con un número menor de entradas o salidas, esto debido a las limitaciones de pines con las que cuenta la BASYS 3.

Este MUX permite leer o escribir hasta 16 dispositivos con solo 5 pines. 4 de ellos controlan el dispositivo a leer. El pin restante recoge la señal leída o escrita en el canal seleccionado, presentando las siguientes características:

1. Multiplexor – Demultiplexor Analógico / Digital de 16 CH 74HC4067
2. Voltaje de Operación: 2V a 6V
3. Corriente de Salida: 20mA
4. Resistencia “encendido”: 70 ohmios a 4,5 V

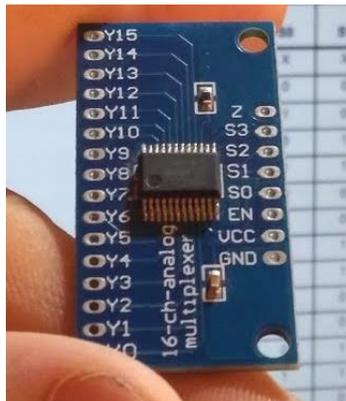


Figura 34 Multiplexor de 16 canales CD74HC4067. Fuente propia.

En la siguiente tabla se presenta como se activan los canales de salidas según el número en binario en el multiplexor, esto permite que cada canal se active a la vez, al ser cuatro señales de entrada, sus posibles combinaciones se representan como  $2^4$  en el que se detallan 16 posibles combinaciones iniciando en 0000 y terminando en 1111.

Tabla 5 Tabla de verdad para el funcionamiento del multiplexor.

S0	S1	S2	S3	$\bar{E}$	SELECTED CHANNEL
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

H= High Level  
L= Low Level  
X= Don't Care

### 6.2.6.2 Conexión de la fase de la etapa de potencia.

Esta fase es básicamente la etapa final del proceso de compra en la máquina expendedora, para este caso los pines de la BASYS determinados para los motores se programa como pines de salidas, en el cual se manda la señal para que los motores puedan funcionar. Por lado, debido a la cantidad de pines a utilizar por la cantidad de motores, existen limitaciones de pines que conlleva a utilizar un multiplexor capaz de permitir aumentar la cantidad de salidas de la BASYS.

Internamente, para que la BASYS 3 pueda hacer funcionar los motores se hace uso de un PWM, la cual es programado en un bloque internamente, cabe destacar que un PWM básicamente consiste en activar una salida digital durante un tiempo y mantenerla apagada durante el resto generando así pulsos positivos que se repiten de manera constante, por lo tanto, para esta fase a la BASYS se le asignan 4 salidas definidas como **S0** hasta **S3**, directamente asignadas al pin **JB10** hasta **JB7**.

Se definen también 4 signal determinados como, **SIG-1** hasta **SIG-4**, especificados en los pines **JB1** hasta **JB4**, estos signal funcionan como referencia definiéndose en **LOW** para el multiplexor.

**Basys3:** Pmod Pin-Out Diagram

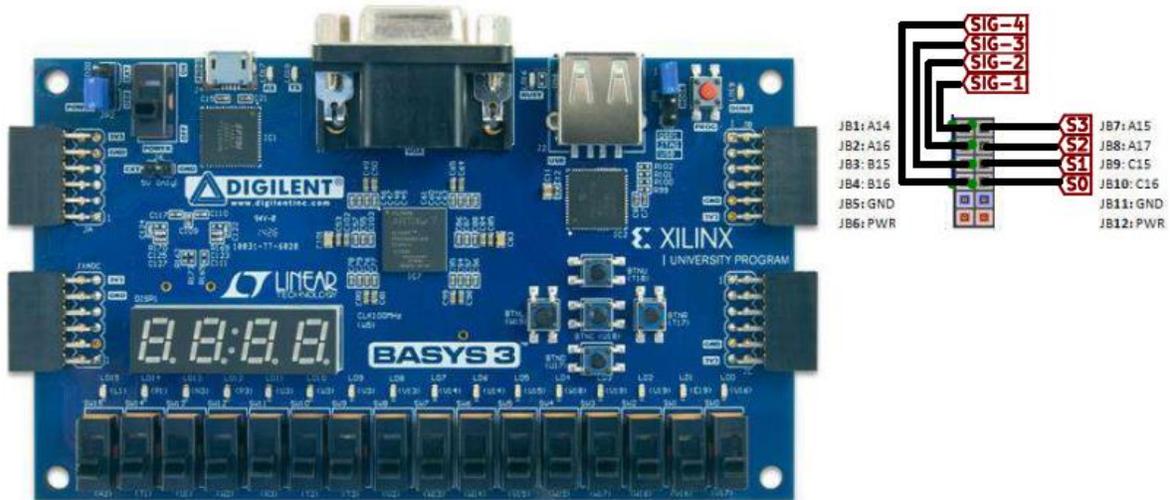


Figura 35 Conexión de la BASYS 3 a los mux. Fuente propia.

La conexión de los multiplexores es un caso especial, ya que se utilizan 4 de estos, donde cada uno trata la señal que proviene de la BASYS y pasa hacia el controlador ULN2003, que en este caso cuenta con 4 entradas debido a que estos aumentan la capacidad de carga de la salida.

Cada multiplexor controla una bobina del motor, por tanto, el mult 1 controla el movimiento en los primeros 90 grados de los 15 motores en uso, el mult 2 controla hasta los 180 grados, el mult 3 controla hasta 270 grados y el mult 4 controla el último cuarto de vuelta. Por consiguiente, si la BASYS manda una señal **0001** a las entradas del multiplexor, esto representa que solo **S0** va a estar en alto y según tabla de verdad de los mult manda una señal en alto a todos los canales de salidas 1 y por ende manda a activar el controlador y el motor 1.

A continuación, se muestra la conexión de los multiplexores con las salidas de la BASYS al igual que su salida hacia los controladores.

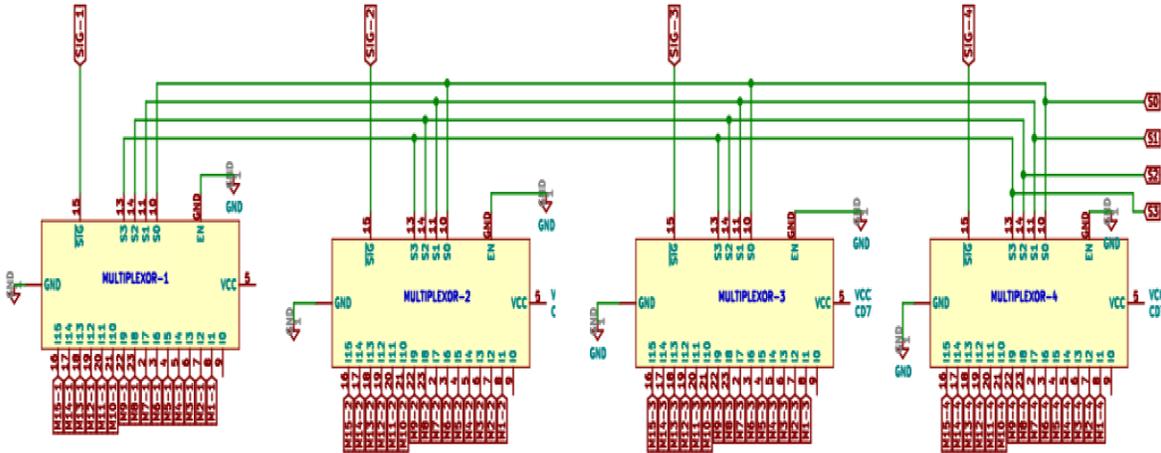


Figura 36 Conexión de los multiplexores y la BASYS. Fuente propia.

### 6.2.7 Fase de simulación.

- **Bloque conta\_m.**

Explicando sobre el esquemático del proyecto en ise design suite, se tiene el bloque conta\_m, la cual es, donde se realiza la decodificación del formato de moneda.

Partiendo del funcionamiento del monedero electrónico, se consta con la siguiente característica que se presenta cuando se introduce una moneda valorada en un Córdoba, en esta parte, el monedero produce un pulso a su salida, teniendo a este como punto de partida en la realización de la programación.

En este bloque se toma como referencia la moneda de un córdoba, esto es muy importante, ya que es relacionada en la producción de pulsos a la hora de identificar una moneda de 5, esta toma 5 pulsos como que, si fueran 5 monedas de 1, entonces este módulo realiza la decodificación de esos pulsos en señales de 4 bits que sería el valor de la moneda introducida convertida a una señal de 4 bits.

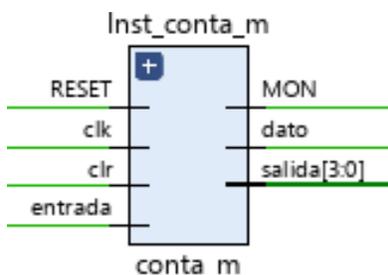


Figura 37 Bloque decodificador del formato de moneda. Fuente propia.

- **Bloque sum.**

El bloque de suma, es el bloque del código donde se toma la señal generada por el bloque **conta\_m** con el fin de sumar el valor del número de monedas leídas, es decir, si se introduce una moneda de 1 córdoba y luego una de 5 córdobas, la suma sería 6 córdobas que es el total de dinero introducido por el usuario

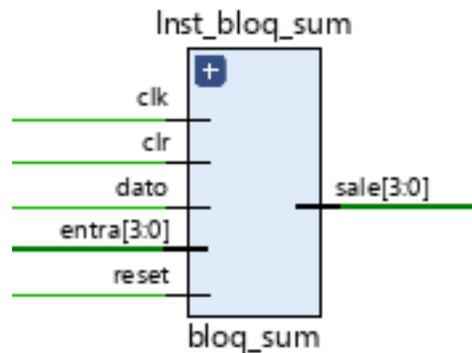


Figura 38 Bloque sumador a partir de la información del decodificador. Fuente propia.

- **Bloque dec\_tec1.**

Este bloque tiene como entrada los datos de salida del módulo de 7 segmentos; parte del código decodifica la señal del teclado de 7 segmentos y lo digitaliza para luego esta señal ser tratada dentro del código con el fin que se refleje el número de producto que el cliente eligió.

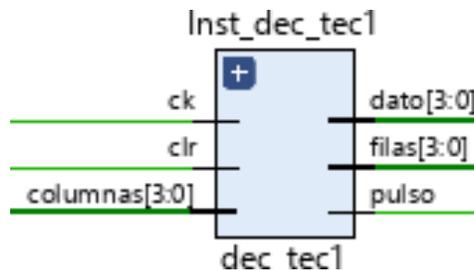


Figura 39 Bloque decodificador de la señal de teclado. Fuente propia.

- **Bloque vga\_prom\_top.**

En este bloque del proyecto se realizan los arreglos que serán reflejados en la pantalla VGA, los valores que se reflejarán será el monto del dinero introducido por el usuario y el producto que eligió, los arreglos tendrán salidas VGA para que se muestren en la pantalla VGA a color.

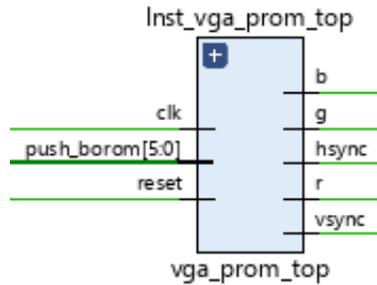


Figura 40 Bloque decodificador del VGA. Fuente propia.

- **Bloque maq\_1.**

Este bloque de programación se realizó con el fin de tener las configuraciones de estados de acuerdo al funcionamiento del proyecto, este se relaciona principalmente con la entrega del producto, es decir sin este bloque el producto no sería entregado, ya que se crean las condicionales en donde convergen varias salidas de otros bloques en este caso, el de las monedas y el teclado.

Este bloque toma como entradas los valores del monto del dinero introducido y la digitación del producto deseado, donde se tiene que cumplir que haya la cantidad de dinero con respecto al producto seleccionado, en dependencia de eso esta manda a su salida el movimiento del motor donde se encuentra lo seleccionado por el usuario; del mismo modo tiene una salida VGA, que es utilizada para la visualización de la entrega de este mismo.

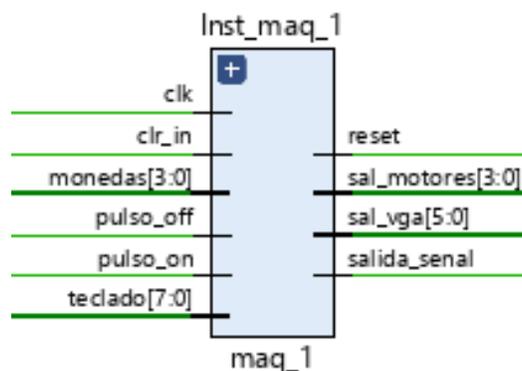


Figura 41 Bloque de la máquina de estado. Fuente propia.

- **Bloque PWM.**

Para este punto, teniendo en cuenta el uso de un motor paso a paso, se genera el presente bloque en donde este es el encargado mandar una señal PWM y hacer funcionar los motores en de dependencia de la salida de señal del bloque **maq 1**, que este es el que controla cuando un motor a realizar el giro.

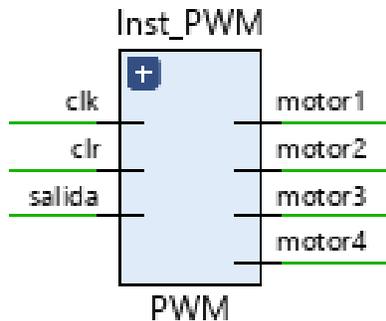


Figura 42 Bloque PWM. Fuente propia.

- **Entradas y salidas esenciales.**

1. **Columnas\_tec:** es una entrada de 4 bits, la cual recibe la señal del teclado matricial que es de 4 bits, para este caso, son los pulsos que manda el teclado matricial de manera binaria.
2. **Clk:** es una señal de reloj, se utiliza porque dentro del código se utilizarán señales de reloj, las cuales se utilizarán para divisores de frecuencias; para contadores, etc.
3. **Clr,** es una señal de barrido la cual se activará al presionar un push button.
4. **Reset,** es una señal de reset para el proceso que se dará dentro de las actualizaciones de estado dentro del funcionamiento del prototipo.
5. **Monedero:** es una entrada, la cual se usará para recibir los pulsos que expulsará el monedero; al detectar que moneda (5 o 1 Córdoba) es la que se está introduciendo, si es una moneda de 1 Córdoba se recibirá un pulso y si es de 5 córdobas se recibirá 5 pulsos respectivamente.
6. **R, G, B:** Son salidas que provienen del controlador vga que brinda que color será lo que se proyectará en el monitor VGA.

7. **H\_S & V\_S:** Son señales de salida provenientes del controlador vga la cual brinda las sincronías verticales y horizontales que tendrán los arreglos al proyectarse en el monitor vga
8. **Salida motores:** es una salida que controlan cuál motor se activara de acuerdo a las condicionales que se programen dentro de la máquina de estados del prototipo.
9. **Motores:** es una salida que controla el torque del motor, es una salida pwm, la cual activara y hará funcionar al motor paso a paso de acuerdo al motor que se mandara activar a través la salida de salida motores.

- **Esquemático general.**

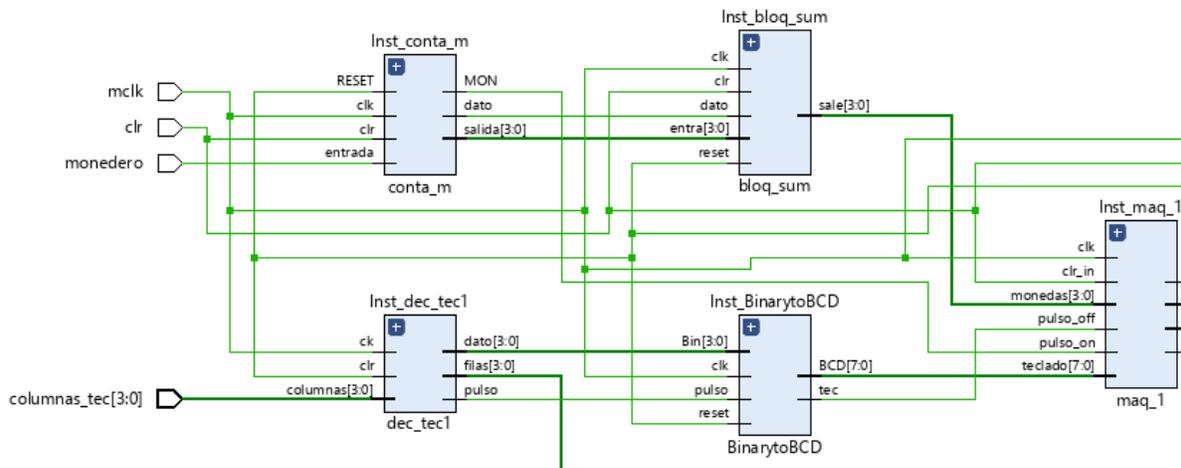


Figura 44 esquema de simulación de las entradas. Fuente propia.

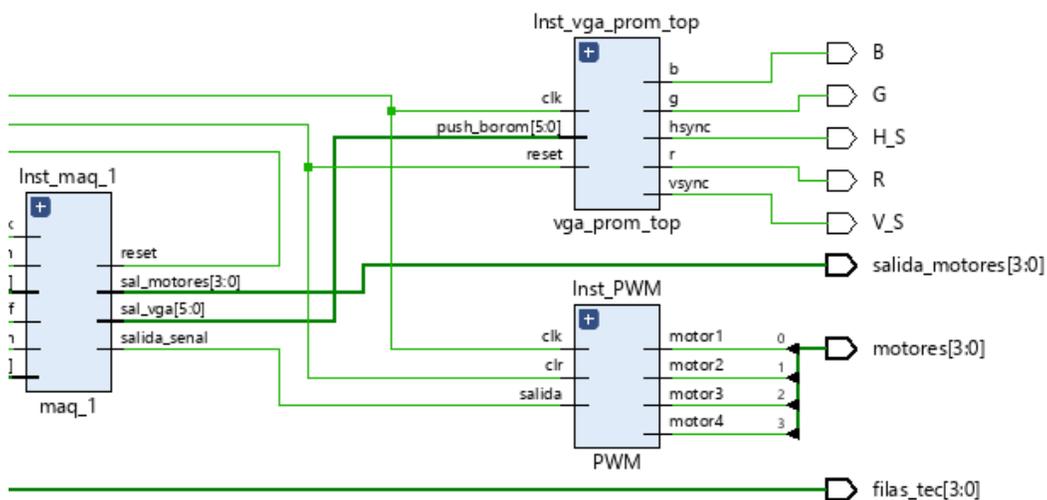


Figura 43 Esquema de simulación de las salidas. Fuente propia.

De esta manera, se presenta la convergencia de cada uno de los bloques, para el funcionamiento correcto de toda la máquina expendedora. Algunos bloques dependen de las entradas y las salidas proporcionadas por otros bloques, teniendo en cuenta la lógica de programación de la máquina de estado y sus condiciones.

**6.2.8 Fase estructural.**

Para esta fase, se toma en cuenta que el prototipo está diseñado a una escala real y de tal forma también su funcionamiento. Cabe destacar, que la estructura cuenta con medidas exactas, de todo el diseño, teniendo en cuenta también la parte interna electrónica funcional. Esta estructura trata de minimizar problemas con el manejo interno de los productos (abastecimiento).

La elaboración estructural se divide en tres partes, definiendo un orden en el que es muy importante seguir para la realización exitosa de esta fase.

- **Diseño.**

El diseño es la parte inicial y muy importante de esta estructura, ya que es de aquí donde se determinan las medidas y la forma del proyecto en sí, para esto se recurre a la realización de dibujos con medidas exactas que posteriormente serán tomadas.

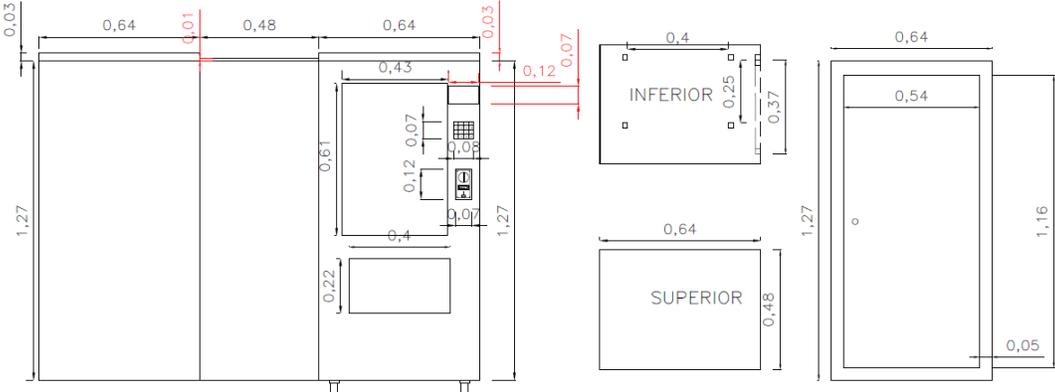


Figura 45 Diseño 2D máquina expendedora. Fuente propia.

El diseño del dibujo fue realizado en la plataforma de AutoCAD, para determinar cada una de las medidas y los componentes una vez terminado el proyecto. El proceso de elaboración permite trabajar con organización y de esta manera obtener unos resultados exitosos a la hora de ponerlos en práctica.

Cada una de estas medidas se tratan de manejar de forma exacta, ya que el funcionamiento debe ser óptimo a la hora de instalar los otros componentes que complementan todo el sistema, y de esta manera no existan problemas a la hora que el usuario trate de obtener un producto. Por otro lado, en dependencia de todo este diseño, se toman las condiciones necesarias para la manipulación de variables utilizadas en el controlador.

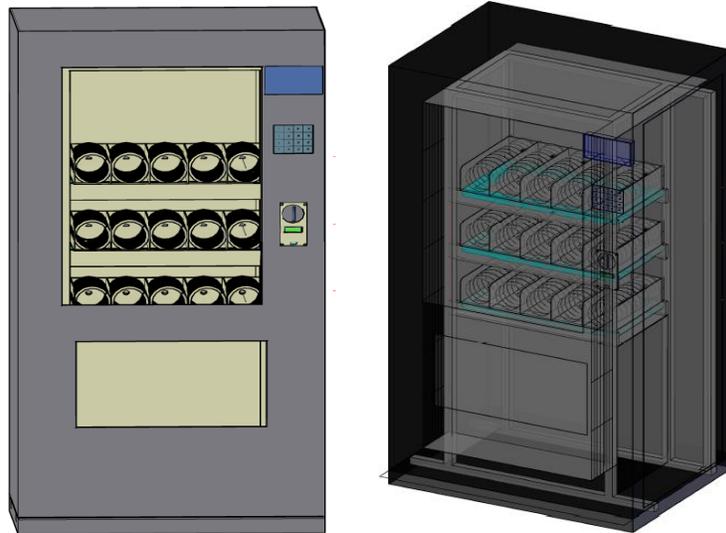


Figura 46 Diseño 3D, parte frontal e interna de la máquina expendedora. Fuente propia.

- **Trazos métricos o cortes.**

Se determina trazos métricos o cortes de cada una de las partes que serán utilizadas en la estructura.

A través de las medidas de los planos se hacen cortes de los materiales que formaran los soportes de toda la máquina, estos cuentan con las medidas siguientes:

- 4 angulares largos de 1.27 m.
- 6 soportes de tubo cuadrados de 1.27 m.
- 1 soporte de tubo cuadrado de 0.43 m.
- 4 angulares de 0.64 m.
- 4 angulares de 0.48 m.
- 3 tubos cuadrados de 0.54 m.
- 2 tubos cuadrados de 1.16 m.

Con respecto a los cortes de las láminas se determinan los siguientes:

- 2 láminas con dimensiones de 1.30 m de alto por 0.64 m de ancho en la que una de ellas contará con detalles en su parte interna por que será la parte frontal, esta lámina frontal se le quitará el diseño anteriormente mostrado.
- 2 láminas de 1.30 m de alto por 0.48 m de ancho, en una de esta se le quitará las dimensiones de la puerta que es de 1.16 m de alto por 0.37 m de ancho.

Hay que tener en cuenta también en los cortes de las bandejas que consta de las siguientes dimensiones, 23 cm de ancho por 40 cm de largo, la base, mientras las divisiones constan de 8 cm de ancho por 23 cm de largo.

- **Ensamblado general de estructura.**

Se presenta la estructura de toda la máquina expendedora haciendo uso de los cortes anteriormente mencionados.



*Figura 47 Proceso de ensamblaje de estructura general. Fuente propia.*

El proceso presentado da a conocer cada uno de los detalles determinado en el diseño y puesto en prácticas con los materiales, lo presentado es la parte física de la estructura, cada detalle determina un buen funcionamiento en los que respecta al uso y el abastecimiento de productos.



*Figura 48 Elaboración de las bandejas. Fuente propia.*

Las bandejas son de suma importancia, ya que aquí es donde estará el producto; a estas se les instaló el rodo y de esta manera facilitar el control de inventario, de igual manera las dimensiones facilitan la entrega de cualquiera de los 15 productos que la máquina pueda vender.

### **6.2.9 Fase de ensamblaje.**

Se determina como el punto de culminación en el cual, se realiza y se presenta todo el proyecto, en donde se une la parte física y la parte electrónica, aprovechando cada uno de los espacios y teniendo en cuenta la organización de cada una de las conexiones, es muy importante mencionar, que en esta fase se presenta de forma física el orden, conexión y funcionamiento presentado en el diagrama de bloques.



*Figura 49 Ensamblaje completo de la máquina expendedora. Fuente propia.*

## VII Costo del proyecto.

Descripción	Cantidad	Unidad	Costo unitario	Costo total
<b>Materiales estructura</b>				
Alambre de aluminio 1/0	15m	15	C\$40.00	C\$600.00
angular de aluminio de 1"x1"x6m		1.5	C\$180.00	C\$270.00
Angulares de 2"x2"x6m chap.1/4		2	C\$950.00	C\$1,900.00
Bisagra "		4	C\$90.00	C\$360.00
Cañuela	3m	3	C\$70.00	C\$210.00
Cerradura		1	C\$120.00	C\$120.00
Disco de pulir corte de 9"		1	C\$180.00	C\$180.00
Disco de corte de 9"		1	C\$180.00	C\$180.00
Lamina de acrilico	8m cuadrado	8	C\$405.00	C\$3,240.00
Lamina de aluminio 2x5	6m cuadrado	6	C\$217.00	C\$1,302.00
Lamina de metalv 2x6	12m cuadrado	12	C\$205.00	C\$2,460.00
lija #100		2	C\$80.00	C\$160.00
lija #120		2	C\$80.00	C\$160.00
lija #60		1	C\$80.00	C\$80.00
lija #80		1	C\$85.00	C\$85.00
Pasta para auto 1/4 galon		2	C\$530.00	C\$1,060.00
Pegamento para acrilico		13	C\$280.00	C\$3,640.00
Perling 5"x2"x2m	2m	2	C\$480.00	C\$960.00
Pintura anticorrosiva blanco 1/4	0.25	0.25	C\$550.00	C\$137.50
pintura anticorrosiva negro mate	1 Gal.	1	C\$550.00	C\$550.00
Pintura anticorrosiva plateado 1/4	0.25	0.25	C\$550.00	C\$137.50
Remache de 2"		20	C\$180.00	C\$3,600.00
Rieles para gavetas	3 par	3	C\$360.00	C\$1,080.00
Ruedas para muebles		4	C\$360.00	C\$1,440.00
sentitizador para pasta para auto		2	C\$360.00	C\$720.00
Soldadura 6011/3:32	4lb	4	C\$180.00	C\$720.00
Tiner	1Gal.	1	C\$170.00	C\$170.00
Tornillo jipson punta broca		50	C\$0.50	C\$25.00
Tornillo moska		20	C\$0.50	C\$10.00
Tubo cuadrado de 1"x1"x6m chap.1/4		2	C\$850.00	C\$1,700.00
silicona industrial		1	C\$230.00	C\$230.00
vidrio 0.40mx0.60m		1	C\$560.00	C\$560.00

<b>Materiales Electrica /electronica</b>				
Alambre 4x18	13m	13	C\$28.00	C\$364.00
Alambre multipar telefonico	1m	1	C\$78.00	C\$78.00
Estaño		4	C\$70.00	C\$280.00
Alambre yumper	4 juego	4	C\$75.00	C\$300.00
Potenciometro 10k		2	C\$40.00	C\$80.00
Pasta de soldar		1	C\$95.00	C\$95.00
cinta de cobre para remover estaño	20cm	20	C\$2.00	C\$40.00
Conectores embas 1mm		5	C\$20.00	C\$100.00
Conectores macho 1mm		12	C\$30.00	C\$360.00
Termo encogible#18	3m	3	C\$20.00	C\$60.00
Termo encogible#16	3m	3	C\$20.00	C\$60.00
Teype super 33		1	C\$130.00	C\$130.00
aislante plastico		12	C\$5.00	C\$60.00
Motor paso a paso 28BYJ-48		15	C\$220.00	C\$3,300.00
Controlador ULN2003		15	C\$140.00	C\$2,100.00
Multiplexor Analógico CD74HC4067		4	C\$72.00	C\$288.00
Monedero electronico JY-936		1	C\$1,440.00	C\$1,440.00
Monitor LCD modelo Green Yi-8		1	C\$1,000.00	C\$1,000.00
Conversor Vga A Rca		1	C\$760.00	C\$760.00
Tarjeta de control Nexys2		1	C\$7,200.00	C\$7,200.00
UPS		1	C\$3,060.00	C\$3,060.00
Fuente DC 19.8V Dell		1	C\$750.00	C\$750.00
Regulador DC-DC XL4015 5A 75W		2	C\$200.00	C\$400.00
Cinta led 12v	2m	2	C\$80.00	C\$160.00
<b>Costo mano de obra programacion</b>		<b>1</b>	<b>C\$18,000.00</b>	<b>C\$18,000.00</b>
<b>Costo mano de obra electrica/electronic.</b>		<b>1</b>	<b>C\$7,200.00</b>	<b>C\$7,200.00</b>
<b>Costo mano de obra estructura</b>		<b>1</b>	<b>C\$5,400.00</b>	<b>C\$5,400.00</b>
<b>Total</b>				<b>C\$81,112.00</b>
			Tiempo de ejecución del proyecto	
			90 dias	

Debido a que es un primer prototipo, el proyecto presenta un valor de 81,112 córdobas, a este se le atribuye el precio de los componentes estructurales, los componentes electrónicos utilizados y los costos de mano de obra (programación, eléctrico/electrónico y estructura). Este valor puede disminuir tomando en cuenta las réplicas, esto gracias al tipo de programación, ya que al ser basado en hardware este puede ser encapsulado disminuyendo costos de producción.

Cada componente utilizado fue seleccionado tomando en cuenta principalmente su óptimo funcionamiento dependiendo de sus características.

## VIII Conclusiones.

La elaboración del prototipo de máquina expendedora, es un punto de inicio en la experimentación de nuevas tecnologías, ya que hace uso de una herramienta bastante completa en lo que respecta al procesamiento de datos, como es la FPGA y que, en tal caso, cumplió con el objetivo de solucionar el problema de obtener algunos productos a cualquier hora del día en la vivienda estudiantil UNI. Por tal razón, a raíz de la elaboración del presente se pudo concluir lo siguiente:

- Al establecer un sistema autónomo de selección y entrega, enfocado al suministro de snack, golosinas y fármacos, se obtuvo resultados eficientes en su funcionamiento, debido a la interacción usuario-máquina, la cual es intuitiva por su fácil manejo, a esto se destaca la buena convergencia de sus procesos para entrega del producto.
- El diseño de un sistema de bajo costo y de fácil implementación en comparación con los equipos comerciales, cumple con el funcionamiento electrónico requerido para la recepción de datos, haciendo uso de la decodificación de moneda, para el bloque lógico de adición que determina la cantidad de dinero insertado y ejecuta la gestión del producto.
- Se implementó con éxito el funcionamiento lógico del prototipo de máquina expendedora, basado en el procesamiento de datos, utilizando arreglos lógicos programables (FPGA) el cual, permitió la detección y decodificación de moneda utilizada por el usuario.

## IX Recomendaciones.

El prototipo presentado es un punto de partida a futuros proyectos, que brindará un mejor desarrollo de conocimientos en alumnos que se profundicen en tecnologías similares; cabe destacar que el presente, es funcional con posibles mejoras a corto o largo plazo, como las mencionadas a continuación:

- El prototipo presenta una característica, la cual determina que esta reconozca solo monedas de 1 y 5 córdobas, por tal razón, se recomienda determinar la lectura de billetes de bajas denominaciones y de esta manera tener una mejor diversidad en el uso monetario para los usuarios.
- La determinación de la entrega de cambio, es una recomendación muy importante, ya que la máquina no cuenta con esta característica debido a que hace uso de valores monetarios exactos.
- Para una mejor interacción máquina-usuario, se recomienda agregar al sistema, un método de pago por tarjeta, que determine de manera rápida la compra del producto y a su vez ampliar nuevos conocimientos con respecto a la innovación tecnológica.
- Desarrollar un control de inventario guardado en la nube, capaz de notificar de forma remota al administrador, esto ayudará a que la máquina expendedora siempre cuente con productos disponibles para los usuarios.
- Teniendo en cuenta el funcionamiento de una pantalla LCD, se recomienda la implementación de avisos publicitarios de carácter general, principalmente comerciales, y de esta manera generar un servicio amplio con respecto a ingresos.

## X Bibliografía

- Aguirre Dobernack, N. (22 de Abril de 2017). *docplayer*. Obtenido de <https://docplayer.es/33860803-1-introduccion-y-objetivos.html>
- Allasia, L. J. (2019). Cerradura con teclado matricial y sensor biométrico.
- Arranz Sancho, V., & Conde Hernández, C. (Noviembre de 2008). *infor.uva.es*. Obtenido de [https://www.infor.uva.es/~cevp/FI\\_II/fichs\\_pdf\\_teo/Trabajos\\_Ampliacion/Monitores.pdf](https://www.infor.uva.es/~cevp/FI_II/fichs_pdf_teo/Trabajos_Ampliacion/Monitores.pdf)
- Assembler Institute. (23 de Febrero de 2022). *assemblerinstitute.com*. Obtenido de <https://assemblerinstitute.com/blog/que-son-lenguajes-programacion-cual-aprender/>
- Baker, E. (9 de Junio de 2020). *www.eeweb.com*. Obtenido de <https://www.eeweb.com/xl4015-improves-dc-module-with-cv-cc-control-a-quick-review/>
- Barber, F., & Ferris, R. (Mayo de 2004). *informatica.uv.es*. Obtenido de [http://informatica.uv.es/iiguia/AED/oldwww/2004\\_05/AED.Tema.02.pdf](http://informatica.uv.es/iiguia/AED/oldwww/2004_05/AED.Tema.02.pdf)
- Camarasa, L. M. (07 de Abril de 2021). *arcades-retroal.com*. Obtenido de <https://arcades-retroal.com/producto/monedero-electronico-programable-multimoneda>
- Canel, S. (6 de septiembre de 2018). *susycursos.com*. Obtenido de <https://susycursos.com/tag/if-then/>
- Corona Ramirez, L. G., Abarca Jiménez, G. S., & Mares Correño, J. (2014). *Sensores y actuadores*. Mexico D. F.: GRUPO EDITORIAL PATRIA, S.A. DE C.V.
- Digilent. (s.f.). *digilent.com*. Obtenido de <https://digilent.com/reference/programmable-logic/basys-3/reference-manual>
- evoltapc. (s.f.). *evoltapc.cl*. Obtenido de <https://evoltapc.cl/convertidores/375-conversor-de-vga-a-rca---s-video-conecta-el-pc-a-la-tv.html>
- Fernandez, D. M. (2006). *adjuntos.yoreparo.com*. Obtenido de <https://adjuntos.yoreparo.com/default/0006/83/6d6940c6b4a1d482dd5732c6f547dee4485ee0b6.pdf>
- Garrigos, J. (Noviembre de 2011). *platea.pntic.mec.es*. Obtenido de [http://platea.pntic.mec.es/~jgarrigo/SAP/archivos/1eva/introduccion\\_motores\\_ca.pdf](http://platea.pntic.mec.es/~jgarrigo/SAP/archivos/1eva/introduccion_motores_ca.pdf)
- González, G. E. (13 de febrero de 2009). *repositorio.tec.mx*. Obtenido de <https://repositorio.tec.mx/bitstream/handle/11285/629058/33068001076506.pdf?sequence=1>

- Guarella, J. (28 de Septiembre de 2011). *www.academia.edu*. Obtenido de [https://www.academia.edu/download/52326408/APUNTE\\_SENSORES\\_Y\\_ACTUADORES.pdf](https://www.academia.edu/download/52326408/APUNTE_SENSORES_Y_ACTUADORES.pdf)
- Ingeniería Mecafenix. (20 de abril de 2017). Obtenido de <https://www.ingmecafenix.com/electricidad-industrial/motor-paso-a-paso/>
- Iupercio, pinos, M., & uday. (Noviembre de 2013). *dspace.ups.edu.ec*. Obtenido de <https://dspace.ups.edu.ec/handle/123456789/5564>
- Matute, V. P. (2013). Diseño y desarrollo de un sistema de ubicacion monitoreo y control de una maquina vending dispensador de bebidas automatico mediante dispositivos avi.
- Motorex. (7 de octubre de 2019). *motorex.com.pe*. Obtenido de <https://www.motorex.com.pe/blog/cual-es-la-diferencia-entre-los-motores-electricos-ca-y-cc/>
- Ortega, J. M., & Garcias Perez , J. A. (Marzo de 2020). *akka-technologies*. Obtenido de <https://www.akka-technologies.com/fpga/>
- Peña, B. (6 de Julio de 2018). *ritsasv.com*. Obtenido de <https://www.ritsasv.com/2018/07/06/analisis-ups/>
- questionpro. (s.f.). *www.questionpro.com*. Obtenido de <https://www.questionpro.com>
- Reyes, L. A. (3 de junio de 2016). <https://ingenieriaelectronica.org>. Obtenido de <https://ingenieriaelectronica.org/utilizar-el-teclado-matricial-4x4-con-arduino/>
- Roque, M. (2010). *Introduccion del calculo del tamaño muestral*. Barcelona.
- Soto, J. A. (30 de Abril de 2019). *geeknetic.es*. Obtenido de <https://www.geeknetic.es/Guia/1605/Tipos-de-Paneles-de-Monitor.html>
- TECSA. (1 de Abril de 2022). <https://www.tecsagro.com.mx>. Obtenido de <https://www.tecsagro.com.mx/blog/sistema-alimentacion-ininterrumpida/>
- Torres, S. (16 de Abril de 2022). *folou.co*. Obtenido de <https://folou.co/dispositivos/pantalla-lcd/>
- Vera Costa, G. E. (2019). *dspace*. Obtenido de <http://dspace.utb.edu.ec/handle/49000/7435>

## XI anexos

### Encuesta.

Apreciados compañeros estudiantes el motivo de esta encuesta es de carácter educativo, con el fin de determinar un numero de aceptación del proyecto máquina expendedora en la vivienda estudiantil de la universidad nacional de ingeniería.

- Con que frecuencia hace uso de las máquinas expendedoras ubicadas en la universidad nacional de ingeniería.

a. Poca frecuencia.

b. Frecuentemente.

- Tiene algunos problemas para acceder a ciertos productos especialmente en horas de la noche.

a. Si.

b. No.

- ¿Qué le parecería una máquina expendedora en la vivienda estudiantil?

a. Excelente.

b. No me gustaría.

- Le gustaría que la maquina expenda productos médicos y de uso personal

a. Si.

b. No.

- Tiene problema que la maquina expenda tres variedades de productos tales como: de uso médico (para enfermedades frecuentes), de uso personal y consumibles como los snacks.

a. Solo productos médicos.

b. Solo productos de uso personal.

a. Solo productos consumibles  b. las tres variedades de productos.

- ¿Qué producto compraría con más frecuencia en horas de la noche?

Escribe el producto.

- Pagaría un poco más por el producto, debido al servicio de la máquina.

a. Si.

b. No.

## CODIGO MAQUINA DE ESTADO

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
use IEEE.STD_LOGIC_arith.ALL;

signal estado_a : std_logic_vector (1 downto 0) := "00"; --estado actual
signal estado_s : std_logic_vector (1 downto 0) := "00"; --estado siguiente
signal control_2 : std_logic ; --señal de control de inicio
--signal mon : std_logic_VECTOR(1 DOWNTO 0) := "00";
--signal tec : std_logic_VECTOR(1 DOWNTO 0) := "00";
signal c : integer range 5002000 downto 0; ----5002000
signal c_2 : STD_LOGIC_VECTOR (7 downto 0);
signal retardo : std_logic := '0';
signal control_3 : std_logic := '0';
signal reset_s : std_logic := '0';
signal sal_motores_s : STD_LOGIC_VECTOR (3 downto 0);
signal sal_vga_1 : STD_LOGIC_VECTOR (5 downto 0);
signal sal_vga_2 : STD_LOGIC_VECTOR (5 downto 0);

begin
  inicio_maquina: process(clk)
  begin
    if (rising_edge(clk)) then
      estado_a <= estado_s;
    end if;
  end process;

  maquina : process(estado_a,pulso_on,pulso_off,control_2)
  begin
    case estado_a is
      when "00" =>
        if (pulso_on = '0') then
          mon <= "00";
          tec <= "00";
          estado_s <= "00";
        elsif (pulso_on = '1') then --se recibe la primer moneda
          estado_s <= "01";
        end if;

      when "01" =>
        if (pulso_off = '0') then
          tec <= "01";
          mon <= "01";
          estado_s <= "01";
        elsif (pulso_off = '1') then --se recibe el segundo num_teclado
          mon <= "01";
          tec <= "10";
          estado_s <= "10";
        end if;
    end case;
  end process;
end;
```

```

        if (control_2='1') then          --se comprobó y despachó el producto
            estado_s <= "00";
--            mon <= "00";
--            teo <= "00";
        elsif(control_2='0') then        --se devuelve al estado dos con
--            mon <= "01";
--            teo <= "01";
            estado_s <= "01";
        end if;
        when others => estado_s <= "00";
    end case;

end process;

salida_mon : process(pulso_on,MONEDAS,CLK)
begin
IF (RISING_EDGE(CLK)) THEN
    if (pulso_on = '1') then          -- es unmenor o igual, sal vga 2 es la entrada de avilitación del segundo mux
        case monedas is
            sal_vga_1 <= "000001";
        when "0000010" =>
            sal_vga_1 <= "000010";
        when "0000011" =>
            sal_vga_1 <= "000011";
        when "0000100" =>
            sal_vga_1 <= "000100";
        when "0000101" =>
            sal_vga_1 <= "000101";          --5
        when "0000110" =>
            sal_vga_1 <= "000110";          --6
        when "0000111" =>
            sal_vga_1 <= "000111";          ---7
        when "0001000" =>
            sal_vga_1 <= "001000";          -- ---8
        when "0001001" =>
            sal_vga_1 <= "001001";          ---9
        when "0001010" =>
            sal_vga_1 <= "001010";          ---10
        when "0001011" =>
            sal_vga_1 <= "001011";          ---11
        when "0001100" =>
            sal_vga_1 <= "001100";          --12
        when "0001101" =>
            sal_vga_1 <= "001101";          ----13
        when "0001110" =>
            sal_vga_1 <= "001110";          -----14
        when "0001111" =>
            sal_vga_1 <= "001111";          -----15
        when "0010000" =>
            sal_vga_1 <= "010000";          -----16
        when "0010001" =>
            sal_vga_1 <= "010001";          -----17
        when "0010010" =>
            sal_vga_1 <= "010010";          -----18
        when "0010011" =>
            sal_vga_1 <= "010011";          -----19
        when "0010100" =>
            sal_vga_1 <= "010100";          -----20
        when "0010101" =>
            sal_vga_1 <= "010101";          -----21
        when "0010110" =>
            sal_vga_1 <= "010110";          -----22
        when "0010111" =>
            sal_vga_1 <= "010111";          -----23
        when "0011000" =>
            sal_vga_1 <= "011000";          -----24

```

```

when "0011001" =>
    sal_vga_1 <= "011001"; -----25
when "0011010" =>
    sal_vga_1 <= "011010"; -----26
when "0011011" =>
    sal_vga_1 <= "011011"; -----27
when "0011100" =>
    sal_vga_1 <= "011100"; -----28
when "0011101" =>
    sal_vga_1 <= "011101"; -----29
when "0011110" =>
    sal_vga_1 <= "011110"; -----30
when others =>
    sal_vga_1 <= "000000"; --cambiar codigo a cuando no hay valor
end case;
else
    sal_vga_1<="000000";          -- cero
end if;
END IF;
end process;

salida_tec : process(PULSO_ON,PULSO_OFF,teclado,CLK)
begin
    IF RISING_EDGE(CLK) THEN
    if (pulso_on = '1' and pulso_off = '0') then
        case teclado is          --agregar señal sobre la entrada del primer numero del teclado
            when "00010000" =>          ---1 EN TECLADO PRIMER NUMERO
                sal_vga_2<="100000";
            when "00100000" => --2          ---2 EN TECLADO PRIMER NUMERO
                sal_vga_2<="010000";
            when "00110000" => --3          ---3 EN TECLADO PRIMER NUMERO
                sal_vga_2<="110000";
            when others =>
                sal_vga_2<="000000";
        end case;
    elsif(pulso_off = '1') then
        case teclado is
            when "00010001" =>          ---- 1-1  TECLADO EN 1ER Y 2DO NUMERO
                sal_vga_2<= "000001"; --1
            when "00010010" =>          ---- 1-2  TECLADO EN 1ER Y 2DO NUMERO
                sal_vga_2<= "000010"; --2
            when "00010011" =>          ---- 1-3  TECLADO EN 1ER Y 2DO NUMERO
                sal_vga_2<="000011"; --3
            when "00010100" =>          -----1-4
                sal_vga_2<="000100"; --4
            when "00010101" =>          ----- 1-5
                sal_vga_2<="000101"; --5
            when "00010110" =>          -----1-6
                sal_vga_2<="000110"; --6
            when "00010111" =>          -----1-7
                sal_vga_2<="000111"; --7 BIN
            when "00011000" =>          -----1-8
                sal_vga_2<="001000"; --8 BIN
            when "00011001" =>          -----1-9
                sal_vga_2<="001001"; --9 BIN
            when "00100000" =>          -----2-0
                sal_vga_2<="001010"; --10 BIN
            when "00100001" =>          -----2-1
                sal_vga_2<="001011"; --11 BIN

```

```

        when "00100010" =>          -----2-2
            sal_vga_2<="001100"; --12 BIN
        when "00100011" =>          -----2-3
            sal_vga_2<="001101"; --13 BIN
        when "00100100" =>          -----2-4
            sal_vga_2<="001110"; --14 BIN
        when "00100101" =>          -----2-5
            sal_vga_2<="001111"; --15 BIN
        when others =>
            tec<="01";          -- una prueba
            sal_vga_2<="000000";
    |
    |         end case;
    |
    |     else
    |         sal_vga_2<="000000";
    |
    |     end if;
    | END IF;
    | end process;

    sal_vga <= sal_vga_1 & sal_vga_2;
PROCESS(CLK,CLR_in)
begin
    if (clr_in = '1') then
        c <= 0;
    elsif (rising_edge(clk)) then
        if(c < 1002000) then      --1002000
            c <= c + 1;
        else
            c <= 0;
            retardo <= not (retardo);
        end if;
    end if;
end process;

motores_sal : process(pulso_off,teclado,monedas)
begin
    if (pulso_off = '1') then
        case teclado is
            when "00010010" =>          --- 1- 2 tec
                if(monedas="001011") then
                    sal_motores_s<="0001";
                    control_2<='1';

```

```

else
    control_2<='0';
end if;
when "00010011" => -----1 3 tec
    if(monedas="000111") then
        sal_motores_s<="0010";
        control_2<='1';
    else
        control_2<='0';
    end if;
when "00010100" => -----1 4 tec
    if(monedas="000101") then
        sal_motores_s<="0011";
        control_2<='1';
    else
        control_2<='0';
    end if;
severity note failure;
when "00010101" => -----1 5tec
    if(monedas="000110") then
        sal_motores_s<="0100";
        control_2<='1';
    else
        control_2<='0';
    end if;
when "00010110" => -----1 6tec
    if(monedas="001111") then
        sal_motores_s<="0101";
        control_2<='1';
    else
        control_2<='0';
    end if;
when "00010111" => -----1 7tec
    if(monedas="000101") then
        sal_motores_s<="0110";
        control_2<='1';
    else
        control_2<='0';
    end if;
when "00011000" => -----1 8tec
    if(monedas="001101") then
        sal_motores_s<="0111";
        control_2<='1';
    else
        control_2<='0';
    end if;
when "00011001" => -----1 9tec
    if(monedas="001110") then

```

```

        if(monedas="001110") then
            sal_motores_s<="1000";
            control_2<='1';

        else
            control_2<='0';
        end if;
    when others =>
        sal_motores_s<="0000";
        control_2<='0';
    end case;
end if;

end process;

process(retardo,pulso_off,reset_s)
begin
    if (pulso_off = '1' or reset_s = '1') then
        if rising_edge(retardo)then
            if (c_2 <= "11000000") then -- 11000000
                c_2 <= c_2 + '1' ;
                reset_s <= '1';
                control_3 <= '1';
            else
                c_2 <= "00000000";
                reset_s <= '0';
                control_3 <= '0';
            end if;
        end if;
    end if;
end process;

process(retardo,sal_motores_s)
begin
    if (rising_edge(retardo)) then
        if ((sal_motores_s = "0001" or sal_motores_s = "0010" or sal_motores_s = "0011" or sal_motores_s = "0100" or sal_motores_s = "0101" or sal_motores_s = "0110" or sal_motores_s = "0111" or sal_motores_s = "1000" or sal_motores_s = "1001" or sal_motores_s = "1010" or sal_motores_s = "1011" or sal_motores_s = "1100" or sal_motores_s = "1101" or sal_motores_s = "1110" or sal_motores_s = "1111")) then
            reset <= '1';
            sal_motores <= sal_motores_s;
            salida_senal <= '1';
        else
            sal_motores <= "0000";
            reset <= '0';
            salida_senal <= '0';
        end if;
    end if;
end process;

end Behavioral;

```

## CODIGO CONTADOR

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
use IEEE.STD_LOGIC_arith.ALL;
entity conta_m is
    Port ( clk : in  STD_LOGIC;
          clr : in  STD_LOGIC;
          entrada : in  STD_LOGIC;
          RESET : in  STD_LOGIC;
          MON : out  STD_LOGIC;
          salida : out  STD_LOGIC_VECTOR (6 downto 0));

end conta_m;

architecture Behavioral of conta_m is
    signal suma :          std_logic_vector (6 downto 0) := (others => '0');
    signal dato :          std_logic:= '0' ;
    signal c :              integer range 10000000 downto 0;
    signal retardo :       std_logic := '0';
    signal MON_S :         std_logic := '0';
    signal control :       std_logic := '0';
    signal tmp :           std_logic_vector (9 downto 0) := (others => '0');
    signal sal_s :         std_logic_vector (9 downto 0) := (others => '0');
    signal i :             std_logic_vector (3 downto 0) := (others => '0');
    signal s_signal_1 :    std_logic_vector (6 downto 0) := (others => '0');
    signal dato_s :        std_logic:= '0' ;
    signal d_c :           std_logic:= '0' ;

begin
    divisor_frecuencia : process(clk,clr)
    begin
        if (clr = '1') then
            c <= 0;
        elsif (rising_edge(clk)) then
            if(c < 5000000) then
                c <= c + 1;
            else
                c <= 0;
                retardo <= not (retardo);
            end if;
        end if;
    end process;

    lectura : process(retardo,entrada,reset)
    begin
        if(reset = '1') then
            sal_s <= "0000000000";
            MON_S <= '0';
        end if;
    end process;
end architecture Behavioral;
```

```

elsif (rising_edge(retardo)) then
  if (entrada='1' and control='0') then
    tmp(9 downto 0) <= (entrada & tmp(9 downto 1));
    control <= '1';
    mon_s <= mon_s;

  elsif(control = '1')then
    if (i >= "1001") then
      i <= "0000";
      control <= '0';
      mon_s <= '1';
      dato_s <= '1';
      sal_s(9 downto 0) <= tmp(9 downto 0);

    else
      i <= i + '1';
      dato_s <= '0';
      tmp(9 downto 0) <= (entrada & tmp(9 downto 1));
      control <= '1';
      mon_s <= mon_s;

    end if;
  else
    control <= '0';

    dato_s <= '0';
  end if;
end if;
end process;

MON <= MON_S;    -- reconoce la puesta en uno de la señal

salida_1 : process(sal_s,reset)
begin
  if(reset ='1') then
    s_signal_1 <= "0000000";
  elsif sal_s = "0101010101" then
    s_signal_1 <= "0000101";
  elsif sal_s = "0000000001" then
    s_signal_1 <= "0000001";
  else
    s_signal_1 <= "0000000";
  end if;
end process;

suma_p : process(s_signal_1,dato,reset,clr)
begin
  if( clr = '1' or reset ='1') then --clr = '1' or
    SUMA <= "0000000";
  elsif(rising_edge(dato))then
    suma <= suma + s_signal_1 ;
  end if;
end process;
salida <= suma;

process (retardo)
variable ct : std_logic_vector(1 downto 0);
begin

```

```

        if rising_edge(retardo) then
            if(dato_s = '1' or d_c = '1') then
                if (ct < "01") then
                    ct := ct + '1';
                    dato<= '0';
                    d_c <= '1';
                elsif(ct < 10) then
                    ct := ct + '1';
                    dato<= '1';
                    d_c <= '1';
                else
                    dato<= '0';
                    ct := "00";
                    d_c <= '0';
                end if;
            end if;
        end if;
    end process;
end Behavioral;

```

## CODIGO TECLADO

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity dec_tecl is
    Port ( columnas : in STD_LOGIC_VECTOR (3 downto 0);
          ck : in STD_LOGIC;
          clr : in STD_LOGIC;
          filas : out STD_LOGIC_VECTOR (3 downto 0);
          dato : out STD_LOGIC_VECTOR (3 downto 0);
          pulso : out STD_LOGIC);
end dec_tecl;

architecture Behavioral of dec_tecl is
    signal cont: std_logic_vector(1 downto 0);
    signal dec_filas: std_logic_vector(3 downto 0);
    signal dec_dato: std_logic_vector(3 downto 0);
    signal c3, c2, c1, c0: std_logic;
    signal c : integer range 6 downto 0;
    signal pulsoT: std_logic;
    signal retardo : std_logic := '0';

begin
    --Proceso para el contador cont.
    Process(ck)
    begin

```

```

    if (rising_edge(ck)) then
        if(c < 2) then
            c <= c + 1;
        else
            c <= 0;
            --retardo <= not (retardo);
            cont <= cont + 1;
        end if;
    end if;
end process;
Process(ck)
begin
    if (rising_edge(ck)) then
        if (columnas="1000") then
            c3<='1';
            pulsoT <= '1';
        elsif(columnas="0100") then
            c2<='1';
            pulsoT <= '1';
        elsif(columnas="0010") then
            c1<='1';
            pulsoT <= '1';
        elsif (columnas="0001") then
            c0<='1';
            pulsoT <= '1';
        elsif(cont = "11" and(c3='1' or c2='1' or c1='1' or c0='1'))then
            c0<='0'; c1<='0'; c2<='0'; c3<='0';
        elsif(cont = "11" and c3='0' and c2='0' and c1='0' and c0='0')then
            pulsoT <= '0';
        end if;

    end if;
end process;

process(ck,clr)
begin
    if clr='1' then
        dato <="0000";
    elsif(falling_edge(ck))then
        if (columnas/="0000" and(c0='1' or c1='1' or c2='1' or c3='1'))
            dato<=dec_dato;
        end if;
    end if;
end process;

--genera una secuencia de salida para las filas, utilizando el contador
dec_filas <= "0001" when cont="00" else
    "0010" when cont="01" else
    "0100" when cont="10" else
    "1000" when cont="11" else
    "0000";
filas<= dec_filas;

```

```

dec_dato <= "0001" when dec_filas="0010" and columnas="0001" else
           "0010" when dec_filas="0010" and columnas="0010" else
           "0011" when dec_filas="0010" and columnas="0100" else
           "1010" when dec_filas="0010" and columnas="1000" else

           "0100" when dec_filas="0100" and columnas="0001" else
           "0101" when dec_filas="0100" and columnas="0010" else
           "0110" when dec_filas="0100" and columnas="0100" else
           "1011" when dec_filas="0100" and columnas="1000" else

           "0111" when dec_filas="1000" and columnas="0001" else
           "1000" when dec_filas="1000" and columnas="0010" else
           "1001" when dec_filas="1000" and columnas="0100" else
           "1100" when dec_filas="1000" and columnas="1000" else

           "1110" when dec_filas="0001" and columnas="0001" else
           "0000" when dec_filas="0001" and columnas="0010" else
           "1111" when dec_filas="0001" and columnas="0100" else
           "1101" when dec_filas="0001" and columnas="1000" else
           "0000";

pulso <= pulsoT;

end Behavioral;

```

## DIVISOR DE FRECUENCIA

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
entity BinarytoBCD is
    Port ( Bin : in  STD_LOGIC_VECTOR (3 downto 0);
          pulso : in std_logic;
          clk : in std_logic;
          reset : in std_logic;
          tec : out std_logic;
          BCD : out  STD_LOGIC_VECTOR (7 downto 0));
end BinarytoBCD;
architecture Behavioral of BinarytoBCD is

    signal    cont :          STD_LOGIC:= '0';
    signal    BCD_s :        STD_LOGIC_VECTOR (7 downto 0);
    signal    cont_div :     STD_LOGIC_VECTOR (7 downto 0);
    signal    c :            integer range 25000000 downto 0;
    signal    retardo :      std_logic := '0';

begin

    divisor_frecuencia : process(clk)
    begin
        if (rising_edge(clk)) then
            if(c < 5000000) then --3125000 funciona bien
                c <= c + 1;
            else
                c <= 0;
                retardo <= not (retardo);
            end if;
        end if;
    end process;
end process;

```

```

lectura_tec : process(Bin,retardo,reset,pulso)
begin
  if(reset='1')then
    cont<='0';
    bcd_s<="00000000";
    tec <= '0';
  elsif (rising_edge(retardo) and pulso='1')then
    if(bin="0001" or bin="0010" or bin="0011" or bin="0100" or bin="0101"
      if (cont='0') then
        cont <='1';
        tec <= '0';
        BCD_s <= bin & "0000";
      elsif(cont='1')then
        BCD_s(3 downto 0) <= bin;
        cont <='0';
        tec <= '1';
      end if;
    else
      bcd_s <= bcd_s;
      cont <= cont;
      tec <= '0';
    end if;
  end if;
end process;

```

```

bcd<= bcd_s;

```

```

end Behavioral;

```

## CODIGO VGA TOP

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;

entity vga_prom_top is
  Port ( clk : in STD_LOGIC;
        reset : in STD_LOGIC;
        hsync : out STD_LOGIC;
        vsync : out STD_LOGIC;
        r : out STD_LOGIC;
        g : out STD_LOGIC;
        b : out STD_LOGIC;
        push_borom : in STD_LOGIC_vector (11 downto 0) );

end vga_prom_top;

--
architecture Behavioral of vga_prom_top is

```

---

```
    COMPONENT ctrl_t
  PORT(
    clk : in std_logic;
    addr : IN std_logic_vector(10 downto 0);
    entrada : IN std_logic_vector (11 downto 0);
    salida_mux : OUT std_logic_vector(539 downto 0)
  );
END COMPONENT;
```

```
COMPONENT vga_initials
PORT(
  vidon : IN std_logic;
  hc : IN std_logic_vector(10 downto 0);
  vc : IN std_logic_vector(10 downto 0);
  M : IN std_logic_vector(539 downto 0);
  --sw : IN std_logic_vector(7 downto 0);
  rom_addr4 : OUT std_logic_vector(10 downto 0);
  red : OUT std_logic;
  green : OUT std_logic;
  blue : OUT std_logic
);
END COMPONENT;
```

---

```
COMPONENT vga_sync
PORT(
  clk : IN std_logic;
  reset : IN std_logic;
  hsync : OUT std_logic;
  vsync : OUT std_logic;
  video_on : OUT std_logic;
  pixel_x : OUT std_logic_vector(10 downto 0);
  pixel_y : OUT std_logic_vector(0 to 10)
);
END COMPONENT;
```

```
signal rom_addr_top : std_logic_vector (10 downto 0);
--signal M_top1 : std_logic_vector (0 to 127);
--signal M_top2 : std_logic_vector (0 to 127);
signal sali_mux : std_logic_vector (539 downto 0);
signal pixel_count_x : std_logic_vector (10 downto 0);
signal pixel_count_y : std_logic_vector (10 downto 0);
signal video_on_top : std_logic;
begin

  Inst_ctrl_t: ctrl_t PORT MAP(
    clk => clk,
    addr => rom_addr_top,
    entrada => push_borom,
    salida_mux => sali_mux
  );
```

```

    Inst_vga_initials: vga_initials PORT MAP(
    vidon => video_on_top,
    hc => pixel_count_x,
    vc => pixel_count_y,
    M => sali_mux,
    rom_addr4 => rom_addr_top,
    red => r,
    green => g,
    blue => b
    );

    Inst_vga_sync: vga_sync PORT MAP(
    clk => clk,
    reset => reset,
    hsync => hsync,
    vsync => vsync,
    video_on => video_on_top,
    pixel_x => pixel_count_x,
    pixel_y => pixel_count_y
    );
--video_on <= video_on_top;
end Behavioral;

```

## CODIGO INSTANCIA VGA

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity vga_initials is
port (vidon: in std_logic;
    hc : in std_logic_vector(10 downto 0);
    vc : in std_logic_vector(10 downto 0);
    M: in std_logic_vector(539 downto 0 );
    --sw: in std_logic_vector(7 downto 0);
    rom_addr4: out std_logic_vector(10 downto 0);
    red : out std_logic;
    green : out std_logic;
    blue : out std_logic
    );
end vga_initials;
architecture vga_initials of vga_initials is
constant vbp: std_logic_vector(10 downto 0) := "00000011111";
constant w: integer := 539;
constant h: integer := 359;
signal C1, R1: std_logic_vector(10 downto 0);
signal rom_addr, rom_pix: std_logic_vector(10 downto 0);
signal spriteon, R, G, B: std_logic;

begin
--set C1 and R1 using switches
C1 <= "000000" & "00001"; --000000000001
R1 <= "000000" & "00001"; --000000000001
rom_addr <= vc - vbp - R1;
rom_pix <= hc - hbp - C1;
rom_addr4 <= rom_addr(10 downto 0);
--Enable sprite video out when within the sprite region
spriteon <= '1' when (((hc >= C1 + hbp) and (hc < C1 + hbp + w))
and ((vc >= R1 + vbp) and (vc < R1 + vbp + h))) else '0';

```

```

process(spriteon, vidon, rom_pix, M)
  variable j: integer;
  begin
    red <= '0';
    green <= '0';
    blue <= '0';

    if spriteon = '1' and vidon = '1' then
      j := conv_integer(rom_pix);
      IF M(j)= '0' THEN
        R <= '1';
        G <= '1';
        B <= '1';
        red <= R;
        green <= G;
        blue <= B;
      ELSE
        R <= '0';
        G <= '0';
        B <= '1';
        red <= R;
        green <= G;
        blue <= B;
      END IF;
    end if;
  end process;
end vga_initials;

```

## CODIGO PWM

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity PWM is
  Port ( clk : in  STD_LOGIC;
        clr : in  STD_LOGIC;
        salida : in  STD_LOGIC;
        motor1 : out STD_LOGIC;
        motor2 : out STD_LOGIC;
        motor3 : out STD_LOGIC;
        motor4 : out STD_LOGIC
  );
end PWM;

architecture PWM of PWM is

  signal c : integer range 5000000 downto 0;
  signal c2 : integer range 7 downto 0;
  signal retardo : std_logic := '0';

begin

  divisor_frecuencia : process(clk)
    begin

```

```

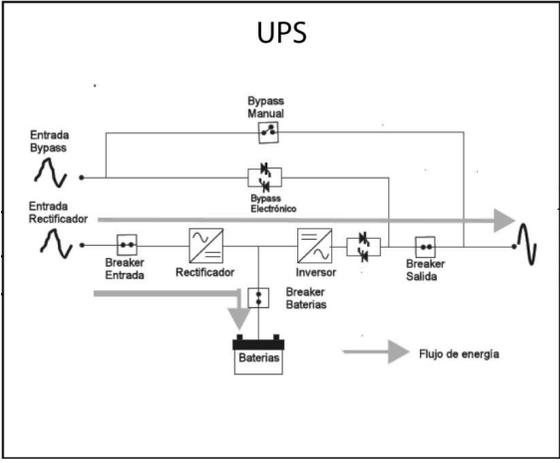
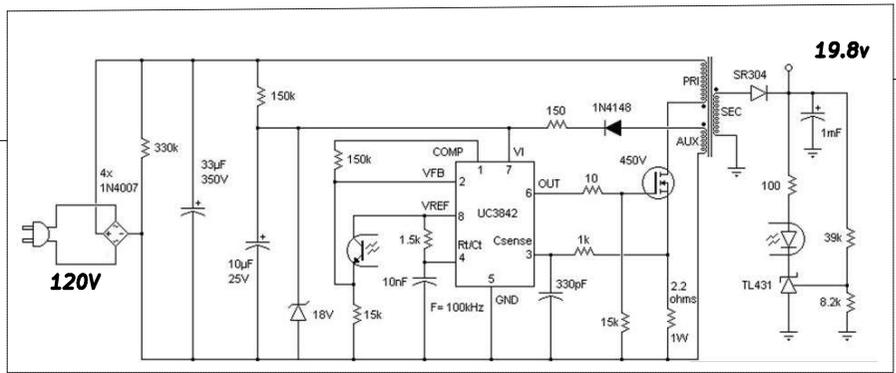
        if (clr = '1') then
            c <= 0;
        elsif (rising_edge(clk)) then
            if(c < 100000) then
                c <= c + 1;
            else
                c <= 0;
                retardo <= not (retardo);
            end if;
        end if;
    end process;

divisor_frecuencia2 : process(retardo)
begin
    if (clr = '1') then
        c2 <= 0;
    elsif (rising_edge(retardo)) then
        if(c2 < 7) then
            c2 <= c2 + 1;
        else
            c2 <= 0;
        end if;
    end process;

    | salida_pasos: process (salida,clk)
    | begin
    |     if (salida = '1') then
    |         if (c2 = 0) then
    |             motor1 <= '1';
    |             motor2 <= '0';
    |             motor3 <= '0';
    |             motor4 <= '0';
    |         elsif (c2 = 1) then
    |             motor1 <= '1';
    |             motor2 <= '1';
    |             motor3 <= '0';
    |             motor4 <= '0';
    |         elsif (c2 = 2) then
    |             motor1 <= '0';
    |             motor2 <= '1';
    |             motor3 <= '0';
    |             motor4 <= '0';
    |         elsif (c2 = 3) then
    |             motor1 <= '0';
    |             motor2 <= '1';
    |             motor3 <= '1';
    |             motor4 <= '0';
    |         elsif (c2 = 4) then
    |             motor1 <= '0';
    |             motor2 <= '0';
    |             motor3 <= '1';
    |             motor4 <= '0';
    |         elsif (c2 = 5) then
    |             motor1 <= '0';
    |             motor2 <= '0';
    |             motor3 <= '1';
    |             motor4 <= '1';
    |         elsif (c2 = 6) then
    |             end if;
    |             end if;
    |         end process salida_pasos;

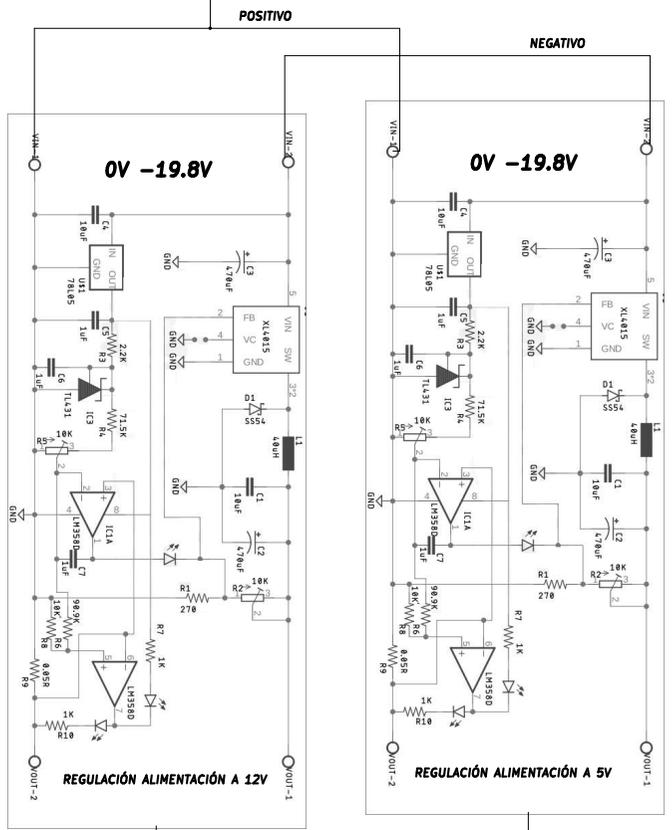
end pwm;

```



RED DE ALIMENTACIÓN 120V

Salida a Fuente de Alimentación 120V



REGULACIÓN ALIMENTACIÓN A 12V

REGULACIÓN ALIMENTACIÓN A 5V

ALIMENTACIÓN MONEDERO ELECTRONICO

ALIMENTACIÓN PANTALLA LED 7.6PULG.

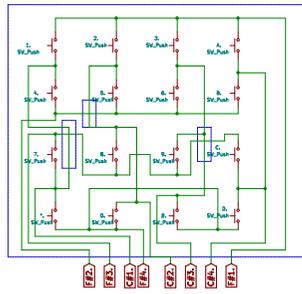
ALIMENTACIÓN TARJETA DE CONTROL NEXYS 2

CONVERTIDOR VGA A RCA

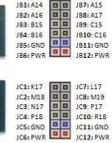
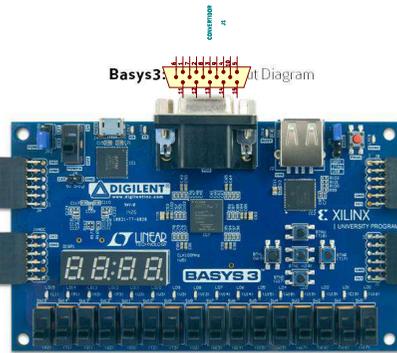
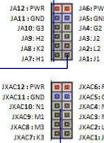
ETAPA DE POTENCIA DE MOTORES MODULO MULTIPLEXORES 74HC4067

ETAPA DE POTENCIA DE MOTORES MODULO ULN2003

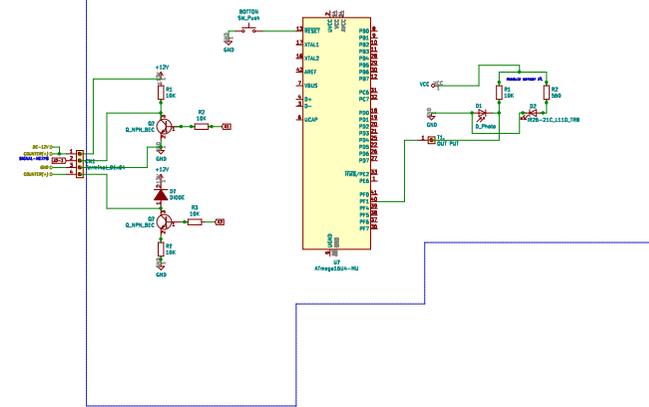
TECLADO 4X4



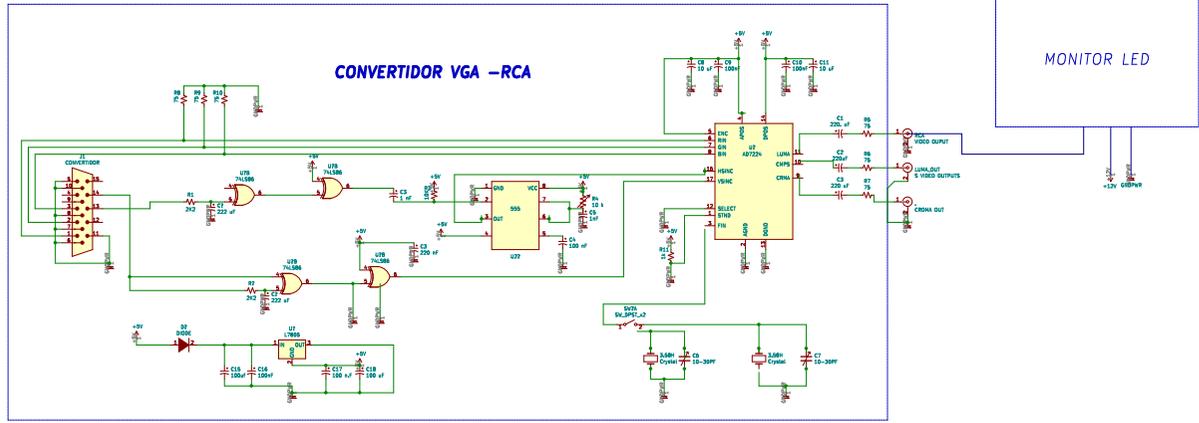
MONEDERO ELECTRONICO  
TECLADO MATRICIAL



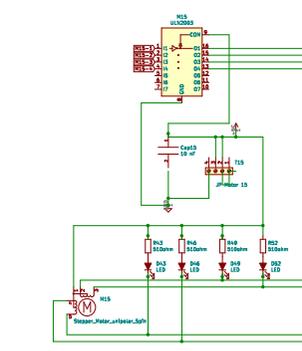
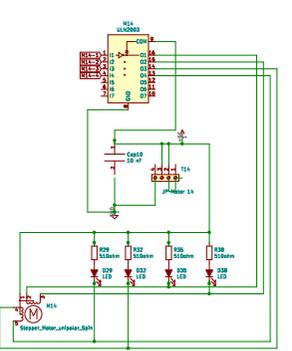
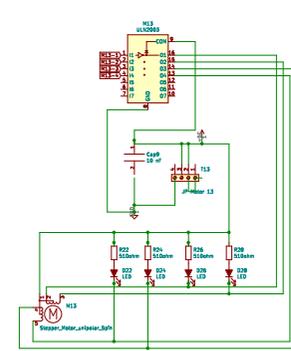
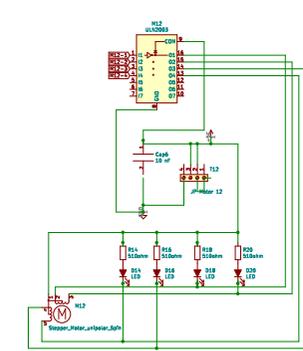
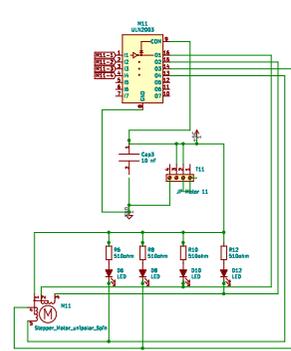
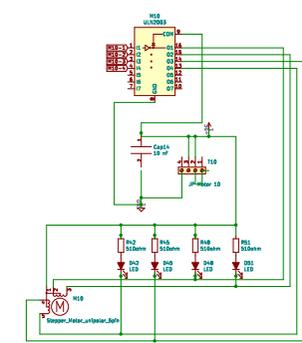
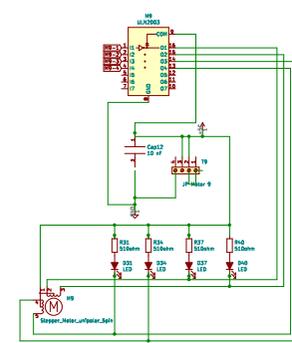
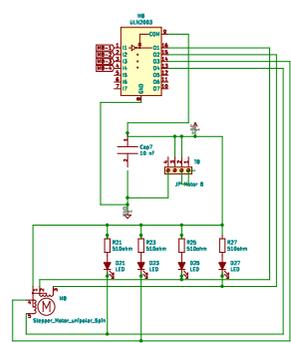
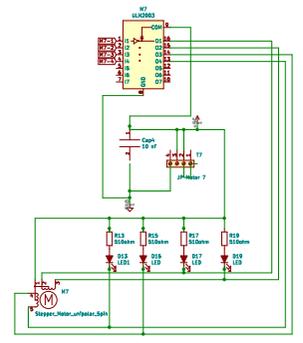
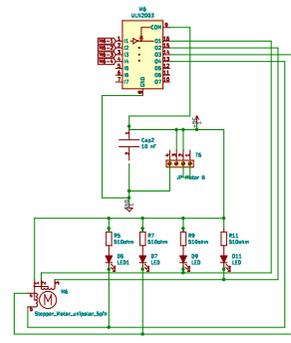
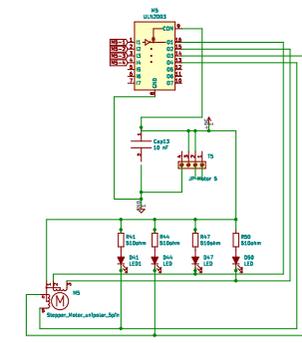
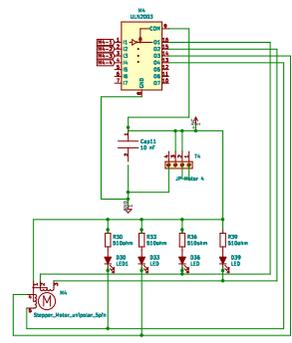
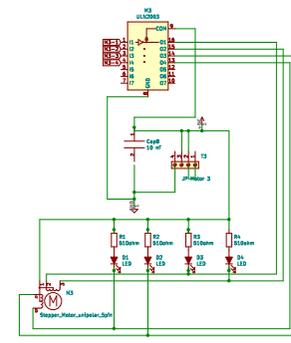
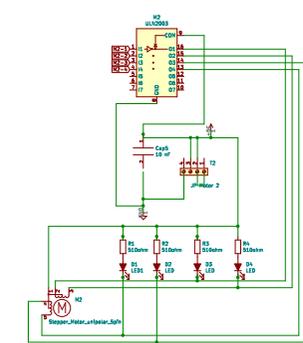
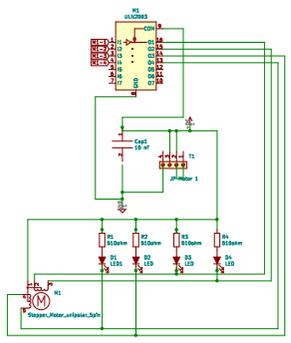
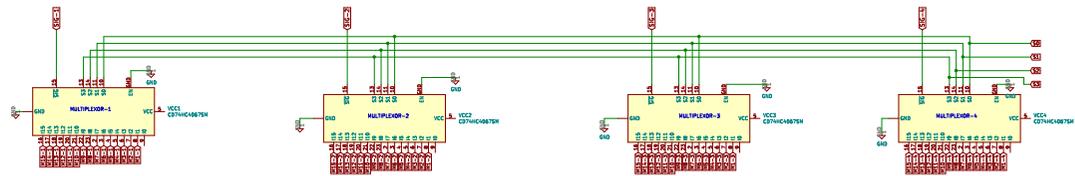
MONEDERO ELECTRONICO

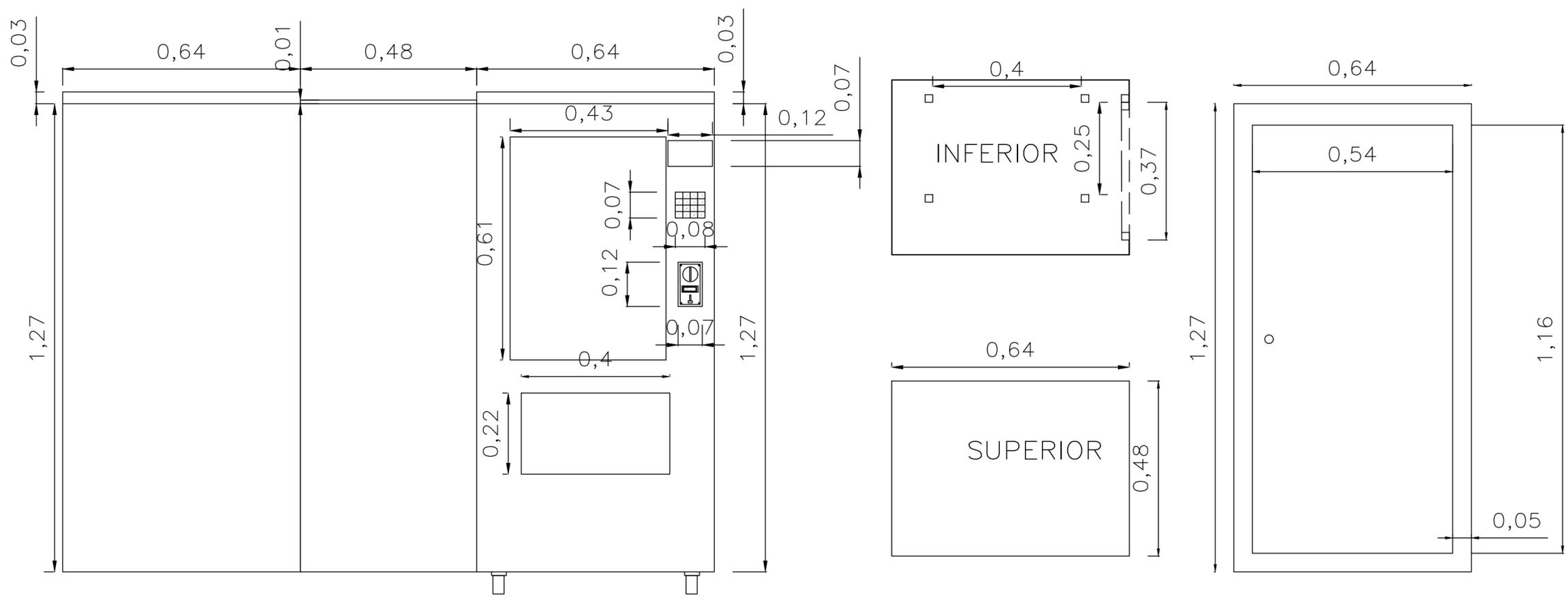


CONVERTIDOR VGA - RCA



MONITOR LED

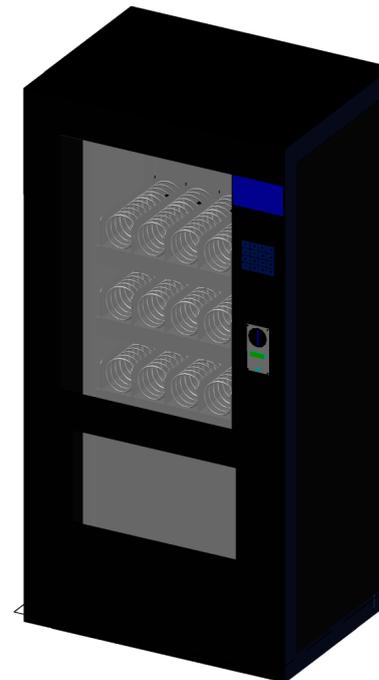
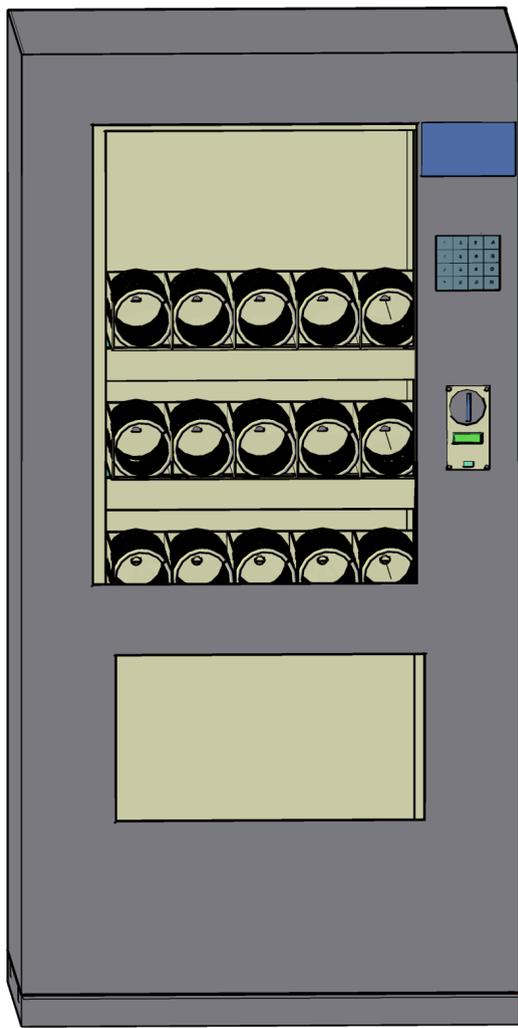





**Universidad Nacional de Ingeniería**  
 Proyecto

REV.	ZONA	DESCRIPCION	REVISADO:	APROBADO
4		Modificación de diagrama		
3		Modificación de diagrama		
2		Modificación de diagrama		
1		Actualización de diagrama		

TITULO:		MAQUINA EXPENDEDORA UNI		HOJA:	1
FECHA:	12/05/2023	DIBUJO:		COD. PLANO:	GP-E010-005.2A
ESCALA:	SIN ESC	REVISADO:			




**Universidad Nacional de Ingeniería**  
 Proyecto

						TITULO:	HOJA:
						MAQUINA EXPENDEDORA UNI	1
						FECHA: 12/05/2023	REV:
						ESCALA: SIN ESC	A
						DIBUJO:	
						REVISADO:	
						COD. PLANO:	
						GP-E010-005.2A	
REV:	ZONA	DESCRIPCION	REVISADO:	APROBADO			

						TITULO:	HOJA:
						MAQUINA EXPENDEDORA UNI	1
						FECHA: 12/05/2023	REV:
						ESCALA: SIN ESC	A
						DIBUJO:	
						REVISADO:	
						COD. PLANO:	
						GP-E010-005.2A	
REV:	ZONA	DESCRIPCION	REVISADO:	APROBADO			

# TRAZOS Y ARMADO DE ESTRUCTURA

