

# Entorno Académico

Revista de divulgación científica  
del Instituto Tecnológico Superior de Cajeme





# Editorial

Cerramos el año 2018 presentando una colección de cuatro artículos relacionados directamente con tópicos de la ingeniería mecatrónica, la cual se define como la sinergia entre las áreas de la ingeniería mecánica, electrónica, eléctrica y los sistemas computacionales. Iniciamos con un análisis del balanceo de la capacidad entre las líneas de producción y los hornos utilizados en un proceso industrial, el cual genera una propuesta de mejora mediante el control de los tiempos de horneado y la eliminación de pérdidas en el área de platinado. Continuando con un artículo que desarrolla el diseño y concluye con la construcción de un vehículo transportador de cargas el cual posee un sistema mecatrónico estabilizador, mismo que permite el movilizar el material utilizado en el proceso de producción de una empresa de la región, este artículo se deriva directamente de un proyecto de titulación de un estudiante de la

maestría en mecatrónica, la cual se encuentra en el programa nacional de posgrados de calidad del CONACYT. También como parte de los proyectos generados en la maestría en mecatrónica nos encontramos con el diseño de un controlador difuso para mejorar el proceso de seguimiento de trayectorias de una celda solar, utilizando el fuzzy toolbox de Matlab. Por último se concluye la edición de la revista con el desarrollo de un prototipo de hardware y software de un CPU didáctico de 8 bits, lo cual permite la transición que se hizo en el tecnológico para pasar de la modalidad teórica de la enseñanza de la materia de arquitectura de computadoras a una práctica, lo que permitirá la formación adecuada, y con las competencias requeridas por la industria, de nuestros estudiantes.

Respetuosamente,  
“Casa Abierta al Tiempo y las Ideas”

---

Lic. Gabriel Baldenebro Patrón  
Director General del Instituto Tecnológico Superior de Cajeme

# Directorio

## DIRECTORIO GENERAL

- Mtro. Aurelio Nuño Mayer**  
Secretario de Educación Pública
- Mtro. Efrén Rojas Dávila**  
Subsecretario de Educación Superior
- Mtro. Juan Manuel Cantú Vázquez**  
Director General de Educación Superior Tecnológica
- M.C. Manuel Chávez Sáenz**  
Director de Institutos Tecnológicos Descentralizados
- Mtro. Ernesto De Lucas Hopkins**  
Secretario de Educación y Cultura del Gobierno del Estado de Sonora
- Mtro. Onésimo Mariscales Delgadillo**  
Subsecretario de Educación Media Superior y Superior de la SEC del Estado de Sonora
- Mtro. Sergio Duarte Escoboza**  
Subsecretario de Planeación y Administración de la SEC del Estado de Sonora
- Mtro. Jose Victor Guerrero Gonzalez**  
Encargado de la Subsecretaría de Educación Básica de la SEC en el Estado de Sonora

## DIRECTORIO INSTITUCIONAL

- Lic. Gabriel Baldenebro Patrón**  
Dirección
- Mtra. Marcela Haydée Osuna Perez**  
Dirección Académica
- Lic. Obed Valenzuela Fraijo**  
Dirección de Vinculación
- Mtra. Olga Margarita Araux Sánchez**  
Subdirección Académica
- Dr. Martín Villa Ibarra**  
Subdirección de Posgrado e Investigación
- Lic. Lucía Avilés Castillo**  
Subdirección de Vinculación
- Ing. Mario Alejandro Gutiérrez De Vore**  
Subdirección de Servicios Administrativos
- Ing. Francisco Leopoldo Lugo Quirarte**  
Subdirección de Planeación
- Mtra. Lilia Beatriz Navarro Fragoso**  
Departamento de Desarrollo Académico
- Mtro. Jesús D. Huicoy Duarte**  
Dpto. de Tecnologías de la Información y Comunicación
- Mtra. Nora Iveth Torres Salazar**  
Departamento de Planeación y Programación
- C.P.C. Hiram Omar Romero Márquez**  
Departamento de Personal
- Mtra. Guadalupe Vásquez Chávez**  
Departamento de Calidad
- Lic. Doris Janeth Rivera Rivera**  
Departamento de Operación y Control Escolar
- Mtra. Fabiola Morales Ortega**  
Departamento de Recursos Financieros
- C. Zayda Vivian Villegas Elías**  
Departamento de Vinculación
- Ing. Lino Noriega Panduro**  
Departamento de Recursos Materiales y Servicios
- Arq. Lorenzo García Gámez**  
División de Arquitectura
- Ing. Bernardo Morales Cervantes**  
División de Ingeniería Ambiental
- Mtro. José Lionso Salazar Huerta**  
División de Ingeniería Electrónica
- Mtro. Hiram Álvarez Velázquez**  
División de Ingeniería en Sistemas Computacionales
- Mtra. Norma Aideé Ríos Lugo**  
División de Ingeniería Industrial
- Mtra. Teresita Burgos Ochoa**  
División de Ingeniería Mecánica
- Mtro. Alberto Limón Valencia**  
División de Licenciatura en Administración
- Mtra. Rocío Grajeda Caballero**  
División de Ingeniería en Gestión Empresarial
- Dra. Socorro del Rivero Jiménez**  
División de Ciencias Básicas
- Mtro. Juan Eduardo Aguilar Ángeles**  
Coordinación de Maestría en Arquitectura Sostenible y Gestión Urbana
- Dr. Alberto Ramírez Treviño**  
Coordinación de Maestría en Ing. Mecatrónica
- Mtra. Carla Olimpia Zapuche Moreno**  
Coordinación de Maestría en Administración
- Dr. Bruno Pablos Lugo**  
Coordinación de Maestría en Educación
- Lic. Adela Machado Acosta**  
Coordinación de Idiomas
- Lic. Paulina Tautimer Delgado**  
Coordinación de Servicios Estudiantiles
- Mtra. Mariela Rubí Navarro Valdez**  
Coordinación de Educación a Distancia
- Mtro. Ricardo Alonso Hernández**  
Coordinación de Cultura

## COMITÉ EDITORIAL

- Dr. Martín Villa**  
**Dr. Mirko Marzadro**  
**Dr. Juan Enrique Palomares**  
**Mtra. Margarita Araux**  
**IBQ. Claudia Irene Rivera**

## CONSEJO DE PUBLICACIONES

- Lic. Gabriel Baldenebro Patrón**  
Consejero Presidente
- Mtra. Marcela Haydee Osuna Perez**  
Consejero Secretario
- Mtro. Ricardo Alonso Hernández**  
Consejero de Docencia

## COMITÉ DE PRODUCCIÓN

- Dr. Juan Enrique Palomares**  
Editor responsable
- Ing. Claudia Irene Rivera Castro**  
Editor Ejecutivo
- Lic. Jonathan Alberto Monteverde López**  
Responsable de Diseño y Producción Digital

## COLABORADORES

- Rubén Navarro García**  
**Ana Mayra Luna Rodríguez**  
**Flavio Muñoz Beltrán**  
**Pablo Esaú Montoya Gómez**  
**Leobardo Rodríguez Contreras**  
**Juan Enrique Palomares Ruiz**  
**José Guadalupe Castro Lugo**  
**José Eulalio Erro Quiñonez**  
**José Efrén Ruelas Ruiz**  
**Alberto Ramírez Treviño**  
**Flavio Muñoz Beltrán**  
**José Luis Beltrán Márquez**  
**Cesar Leonel Zazueta Villavicencio**  
**Ana Cecilia Ruiz Calvillo**

ENTORNO ACADÉMICO, año 15, No. 21, Agosto 2018, es una publicación semestral editada por el Instituto Tecnológico Superior de Cajeme, Carretera Internacional a Nogales Km. 2, Col. Amaneceres 2, Cajeme, Sonora, C.P. 85024, Tel. 01 64 44 10 86 50, [www.itesca.edu.mx](http://www.itesca.edu.mx), [jepalomares@itesca.edu.mx](mailto:jepalomares@itesca.edu.mx). Editor responsable: Dr. Juan Enrique Palomares ISSN: 2448-7635, otorgado por el Instituto Nacional del Derecho de Autor. Responsable de la última actualización de este número, Lic. Martha Vázquez Amaya, Comunicación Social del Instituto Tecnológico Superior de Cajeme, Carretera Internacional a Nogales Km. 2, Col. Amaneceres 2, Cajeme, Sonora, C.P. 85024, fecha de última modificación, mayo de 2016. Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Tecnológico Superior de Cajeme.

# Índice:

Balanceo de capacidad entre línea de producción y horno, mediante el diseño y desarrollo de mejora para control de tiempos de horneado, eliminación de pérdidas y reprocesos, en el área de platinado	5
Vehículo transportador con estabilizador de carga	13
Diseño de un controlador difuso para un seguidor solar utilizando fuzzy toolbox de MATLAB	24
Prototipo de hardware y software de un CPU didáctico de 8 bits	31

# Balaceo de capacidad entre línea de producción y horno, mediante el diseño y desarrollo de mejora para control de tiempos de horneado, eliminación de pérdidas y reprocesos, en el área de platinado

R. Navarro-García<sup>1</sup>, A.M. Luna-Rodriguez<sup>1</sup>, F. Muñoz-Beltrán<sup>2</sup>

<sup>1</sup>Ingeniería en electrónica, Instituto Tecnológico Superior de Cajeme, Sonora, México

<sup>2</sup>Maestría en ingeniería mecatrónica, Instituto Tecnológico Superior de Cajeme, Sonora, México

\* e-mail: rnavarro@itesca.edu.mx

**Resumen:** *Este trabajo presenta una mejora al proceso manual de platinado de conectores en una empresa la cual ha presentado problemas en su producción por depender totalmente del factor humano. Se concentra principalmente en la etapa de horneado, donde el operador manipula la introducción y extracción de las piezas, así como la toma de tiempos del mismo, mediante la ayuda de un despliegue visual de las curvas de temperatura/tiempo, llamado Yokogawa. La propuesta consiste en un sistema con la modalidad de control manual y remoto; para este último se utiliza una Tablet con sistema operativo Android, el software de diseño electrónico Proteus, de programación para PIC MPLAB y de programación de aplicaciones Android Google MIT App Inventor. Al prototipo se le realizaron las pruebas para determinar su correcto funcionamiento y posteriormente se instaló en el horno para su trabajo en la línea; ya instalado se realizaron las pruebas en sitio, obteniendo resultados de mejoras en el proceso.*

**Palabras clave:** Proceso de platinado, etapa de horneado, control remoto.

**Abstract:** *This paper shows an improvement to the connectors plating manual process in a company which has presented problems in its production because it depends totally on the human factor. This problem is mainly concentrated in the furnacing stage, where the operator manipulates the pieces introduction and extraction, and measurement of time, by means of visual display of the temperature / time curves. The improvement proposal is a system with the manual and remote control mode; where a Tablet with Android operating system is used for remote mode is used, with Proteus electronic design software, programming software for PIC MPLAB and Google MIT App Inventor. First, the prototype is tested to verify correct operation off-line, then it is installed in the furnace for its work on line; finally it is tested on site to verify results of improvements in the process.*

**Keywords:** Plating process, stage furnaced, remote control mode.

## 1. Introducción

Dentro de los procesos de maquilado de arneses es muy común la fabricación de componentes de interconexión de alta fiabilidad para numerosas áreas, incluyendo aeroespacial, de defensa, industrial, médica, automotriz y telecomunicaciones.

Se trabajó con una empresa en donde la producción de conectores implica que las piezas pasen por varias etapas, tales como maquinado, platinado y horneado. Esta última etapa, es la que otorga las propiedades de conducción de corriente y la resistividad mediante la regulación de

los materiales que se depositan en la pieza. El platinado es uno de los acabados fundamentales que le aportan a la pieza las propiedades necesarias para cumplir con los estándares de control y calidad que maneja la empresa.

El proceso de horneado se realiza en un horno de acabado final para plásticos con recubrimientos especiales, donde el operador es el encargado de manipular la introducción y extracción de las piezas, y la toma de tiempos de horneado, mediante la ayuda de un despliegue visual de las curvas de temperatura/tiempo, llamado Yokogawa, las cuales indican el momento en que se introdujo el lote de piezas, la hora en que el horno alcanzó

la temperatura requerida para el horneado y la finalización del proceso. Mediante la observación, el operador puede determinar el tiempo de horneado a partir de que se alcanzó la temperatura deseada, para acudir directamente al horno y programar el día y la hora de finalización en el temporizador del mismo.

Cada pieza requiere una temperatura de horneado de  $200^{\circ}\text{C} \pm 2^{\circ}\text{C}$  de error. Por otra parte, este acabado que se le da a la pieza debe llevarse a cabo durante una hora y treinta minutos exactos a partir de que el horno ha alcanzado la temperatura indicada anteriormente.

El área reporta constantemente piezas dañadas por reproceso, que terminan en desperdicios o reprocesado, lo que conlleva gasto y tiempo perdido. Esto da lugar a los siguientes problemas:

- Tiempos de proceso inexactos por error humano en los cálculos.
- Reproceso por olvido de las piezas en el horno, ya que la temperatura tarda mucho en disminuir y las piezas deben ser extraídas rápidamente.
- Pérdidas de piezas y de tiempo para llevar a cabo el reproceso.
- Desbalanceo de la línea de producción, por efecto cuello de botella. Esto es, se acumulan piezas en espera del horneado mientras el horno sigue ocupado.
- Pérdida de tiempo de trabajo del operador en sus otras actividades, ya que constantemente deja lo que está realizando para ir a verificar el sistema Yokogawa.

Cuando las piezas resultan dañadas por el reproceso, esto puede generar distintas problemáticas dependiendo del tipo de pieza que sea, esto es:

- Pasar la pieza a un proceso de extracción de materiales depositados para dejarla en su estado inicial y así poder procesarla nuevamente, o
- Puede ser una pieza que no tiene una solución probable y es reportada como producto de deshecho

En este trabajo se documenta el sistema que automatiza el proceso mencionado, con el fin de evitar el error humano al que se expone el sistema al depender de la observación y pericia del trabajador.

## 2. Antecedentes

Según Silva (2011) [1, 2] uno de los problemas más importantes que se tiene dentro de los procesos de manufactura, es el de asegurar un flujo continuo y uniforme de los productos a través de los diferentes procesos dentro de la planta.

Rother y Harris (2001) [3–6] dicen que el Balanceo de Líneas de producción consiste en agrupar actividades u operaciones que cumplan con un tiempo de ciclo y con el fin de que cada línea de producción tenga continuidad, es decir que en cada estación o centro de trabajo, se cuente con un tiempo de proceso uniforme o balanceado, de esta manera las líneas de producción pueden ser continuas y no tener cuellos de botella.

Silva (2011) dice que los principales elementos para un balanceo de línea son:

- a) Tiempo de las operaciones, para determinar el número de operadores necesarios para cada operación.
- b) Tiempo de ciclo del proceso.
- c) Número de estaciones de trabajo.

Silva (2011) señala que una de las herramientas más utilizadas en la filosofía de Manufactura Esbelta o Lean Manufacturing, es el Yamazumi. Esto es, una gráfica de barras, que muestra el balance de cargas de trabajo y el tiempo de ciclo entre un número de operarios de una línea de producción o conjunto de células, de tal forma que se puede saber cuáles son las actividades que realiza cada operario y el tiempo de duración de cada una de ellas. Las ventajas del Yamazumi son:

- 1) Es una herramienta visual: Hace más fácil el identificar dónde se tienen desperdicios o dificultades de operación
- 2) Sencillez: Fácil de interpretar
- 3) Es ineludible: Es un aviso constante y perpetuo para la mejora continua, o Kaizen, ya que muestra claramente dónde se presentan las falencias dentro del proceso
- 4) Es público: No afecta la privacidad de los procesos de la empresa, ya que no especifica detalles del proceso
- 5) Identificación de oportunidades: debido a que se pueden tener visualmente las principales limitaciones y obstáculos clave, permite ampliar la eficiencia del proceso.

Por otro lado, existe una técnica gráfica llamada Value Stream Map (VSM) [5, 6], que permite visualizar todo un proceso, facilita detallar y entender completamente el flujo tanto de información como de materiales necesarios para que un producto o servicio llegue al cliente, con esta técnica se identifican las actividades que no agregan valor al proceso para posteriormente iniciar las acciones necesarias para eliminarlas. La figura 1 muestra el proceso para la realización de un VSM. ([www.leansolutions.com](http://www.leansolutions.com))

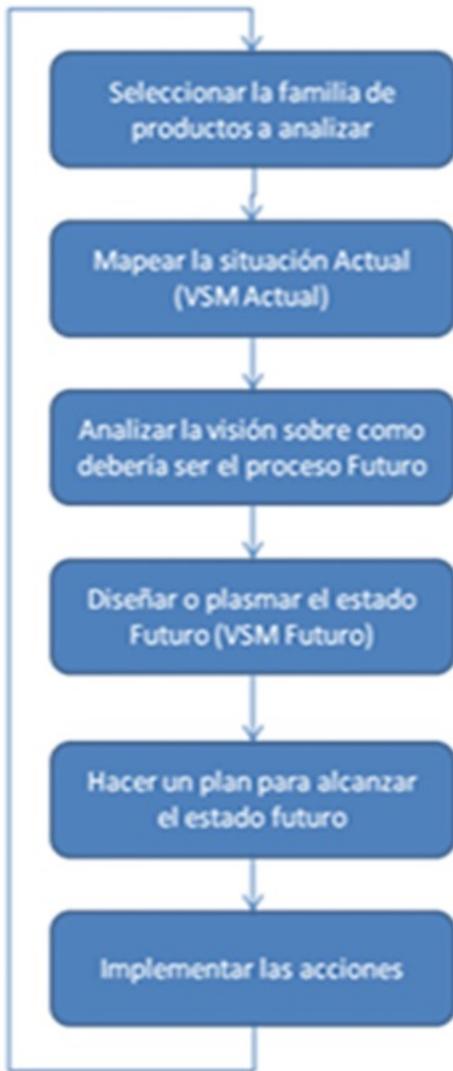


Figura 1: Implementación de un VSM.

Como parte del proceso de automatización, se debe contar con una forma de controlar dicho proceso, para lo cual es muy común el uso de microcontroladores, los cuales están conformados por un circuito integrado programable que integra en un solo chip las unidades de memoria para el almacenamiento de datos, aritmética – lógica para el cálculo de operaciones, las unidades de entrada y salida para comunicación con otros periféricos, temporizadores

y el controlador de interrupciones [7, 8].

Según CEDUVIRT, los parámetros más importantes en un microcontrolador son:

- 1) Bus de datos: 8, 16, 32 bits.
- 2) Capacidad de memoria: Tamaño de la memoria RAM y de la memoria EEPROM en kilobytes KB.
- 3) Velocidad: Número de instrucciones a ejecutar por segundo. Depende de la frecuencia del oscilador del micro.
- 4) Puertos: Puertos de entrada salida de forma paralela y serial para comunicación externa.
- 5) Módulos: Para conversión A/D, D/A, PWM, USB, CAN, I2C, SPI, UART, USART.

Para la adquisición de señales que serán enviadas al microcontrolador se usan sensores, en este caso, sensor de temperatura. Según Aragonés (2003) [9, 10], los termopares son los sensores de temperatura eléctricos más utilizados en la industria. Un termopar se hace con dos alambres de distinto material unidos en un extremo, al aplicar temperatura en la unión de los metales se genera un voltaje muy pequeño, del orden de los mili-volts el cual aumenta con la temperatura.

Como se muestra en la figura 2, los cables compensados tienen una polaridad de conexión (+) y (-) que al conectarse con el termopar se debe respetar. Es importantísimo que estos dos cables compensados sean para el tipo de termopar que se está usando y además estén conectados con la polaridad correcta (+) con (+) y (-) con (-).

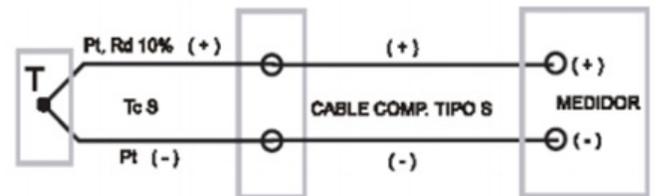


Figura 2: Termopar tipo J conectado con cable compensado.

### 3. Metodología

El presente trabajo es una aplicación tecnológica, utilizando el “Método de Ingeniería”, ya que se basa principalmente en la observación del proceso y se analiza

la problemática, para posteriormente planear en forma eficaz la solución, la cual se diseña bajo la óptica de la ingeniería en cuestión. El trabajo inicia con un análisis para identificar la distribución de actividades y tiempo del operador.

Posteriormente, se elaboró la propuesta de mejora, y se determinó la forma en que iba a trabajar el sistema, definiendo que se utilizaría un control manual y remoto, este último mediante una Tablet con sistema operativo Android. Después se utilizaron el software de diseño electrónico Proteus, de programación para PIC MPLAB, y de programación de aplicaciones Android Google MIT App Inventor.

Por último, se realizaron pruebas en un prototipo, detectándose los errores a tiempo, concluyendo que el sistema funcionaba correctamente. Posteriormente, se realizó la instalación del hardware y software en el horno.

## 4. Resultados

Como parte del diagnóstico de la problemática, en la figura 3 se muestran los tiempos de ciclo del proceso de platinado, aluminio de color azul y plástico de color rojo, donde se observa la inestabilidad del proceso. Esto puede ser causado por:

- Los paros de producción en línea por mantenimiento o descompostura de maquinaria, entre otras fallas.
- Un flujo inadecuado de piezas debido a un mal balance.
- Actividades o estaciones de trabajo mal distribuidas.

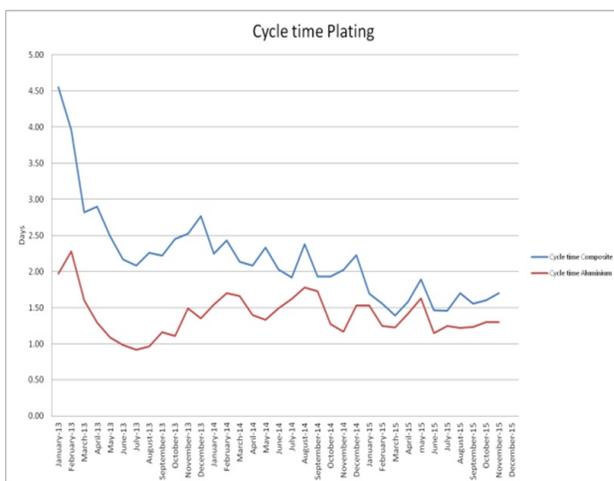


Figura 3: Tiempos de ciclo de platinado

Durante el análisis del proceso se detectaron los siguientes puntos de mejora:

- Temporizador preciso en función de horas sin tomar importancia al día de la semana.
- Implementación de base de datos con las especificaciones para disminución de error de mezcla de los mismos con distintas especificaciones.
- Graficar el comportamiento del horno en base a temperatura/tiempo, especificando los puntos clave que necesita documentar el operador, ayudando a agilizar la obtención de datos.

### 4.1. Elaboración de propuesta de mejora

En base a las necesidades de mejora del proceso, se propuso un sistema de control manual y remoto, con monitoreo portátil de los datos del proceso de horneado en tiempo real, con las propiedades mencionadas en la tabla 1

Cuadro 1: Propiedades de la propuesta de mejora

Propiedades	Viabilidad
* Instalación de tablero de control manual. * Instalación de sensores de temperatura independientes a los existentes, para evitar ruido en la obtención de señales. * Menú de opciones. * Control remoto mediante aplicación para dispositivo Android, interconectado mediante un módulo Bluetooth. * Control manual y remoto sincronizado. * Emisión de alerta de proceso terminado al trabajador directamente en el dispositivo. * Paro automático del horno a partir de la alarma. * Alto alcance. * Antena para incrementar el alcance.	* Exactitud en control de tiempos de proceso. * Notifica al operador sobre el finalizado del proceso. * Evita que las piezas queden por más tiempo del indicado en el horno. * Incrementa el flujo de piezas procesadas. * Cuenta con buen alcance incrementable mediante antena.

#### 4.1.1. Especificaciones del sistema.

En la figura 4 se detallan las señales que recibirá y emitirá el microcontrolador PIC, donde se observa que recibirá la señal de los sensores de temperatura (ya promediadas y amplificadas), recibirá la señal del control manual y remoto, y realizará operaciones de comparación

en temperatura y tiempo, para posteriormente tomar decisiones en cuanto al envío de señales al control remoto y al horno.

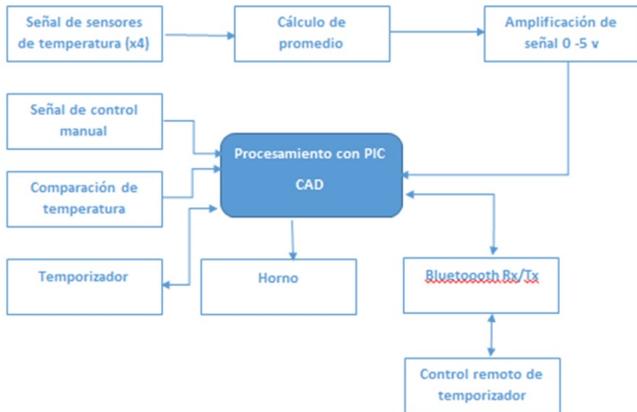


Figura 4: Diagrama del sistema

Operaciones a detalle:

- Temperatura: la señal proveniente de los sensores de temperatura, colocados dentro del horno, se promediarán para la obtención de un dato más preciso, mismo que se amplificará en un rango de 0 a 5 Volts para poder ser procesada en el PIC.
- Etapa comparativa: se comparan la señal de temperatura actual del horno y la temperatura deseada, enviando una señal binaria (0 o 1); al resultar en igualdad enviará un 1 al PIC, indicándole así que a partir de ese momento comienza el horneado de la pieza.
- Temporizador: Se manipula por alguno de los controles (manual o remoto). El PIC es quien recibe la información y lo controla; cuando el temporizador llega al final, el PIC envía una señal binaria 0 al horno, para apagarlo.
- Programación manual: El hardware contará con un menú para seleccionar el tiempo que se desea procesar la pieza. Esto se deja como medio de seguridad en caso de falla del dispositivo de control remoto.
- Programación remota: se desarrolló una aplicación Android para un dispositivo, Tablet o Smartphone; dicha aplicación, desplegará el estado de funcionamiento del horno, un temporizador auto programable y emitirá señales de alerta cuando el proceso de horneado haya finalizado. Esto es posible mediante el módulo Bluetooth

#### 4.1.2. Prototipo de pruebas

En la figura 5 se observa el prototipo que se construyó con el fin de realizar pruebas de hardware y software, mediante la visualización de señales con LED's.

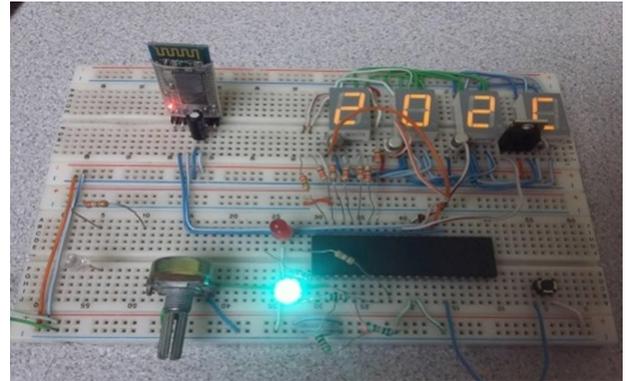


Figura 5: Prototipo de pruebas

El tiempo fue manipulado mediante un botón y la temperatura fue simulada con un potenciómetro. El encendido del horno fue representado con el led azul, y el temporizador activado fue simulado con el led rojo.

#### 4.1.3. Diseño de software

Como se observa en la figura 6, al iniciar el sistema, se realizará el sensado de temperatura y su despliegue, al programar el tiempo y la temperatura que se desea en el proceso el software comenzará a comparar la señal de temperatura actual con la deseada y hará una toma de decisión, es decir:

- Si (Temperatura actual = Temperatura programada) entonces activa el temporizador.
- Si (Temperatura actual  $\neq$  Temperatura programada) entonces continúa comparando.

Cuando el temporizador llega a cero, se ejecutan un paro de horno y la emisión de señal de alerta. La transmisión y recepción de datos se efectúa desde que el sistema comienza a funcionar. A partir de que la alerta ha sido emitida y se atiende, el sistema se reinicia.

Para configurar algunos registros se tiene que usar la instrucción BANKSEL para saltar de banco de memoria y configurarlo, al terminar de configurar volver a usar dicha instrucción para retornar al BANCO0 donde es que se ejecuta continuamente el programa.

Dichos registros funcionan de tal manera que configura la entrada analógica de un cierto pin al ponerlo a 1, en este caso se configuraron los pines del puerto

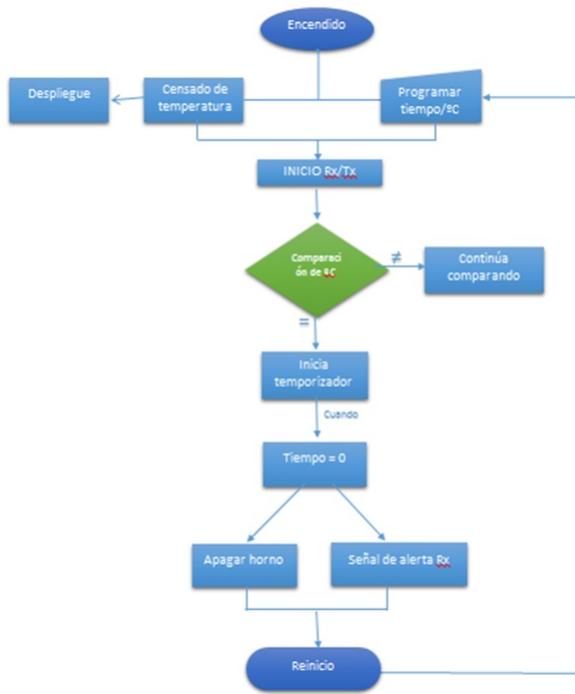


Figura 6: Estructura del software del control manual con PIC

A, RA0 y RA3 como entradas analógicas. RA0 está encargada de monitorear la temperatura del horno para su posterior encendido. RA3 está encargada de monitorear la temperatura interna en el circuito para encender el ventilador que controlará dicha temperatura para tener un óptimo desempeño del mismo, dichos registros se presentan en la figura 7.

Los registros ANSEL y ANSELH se utilizan para configurar el modo de entrada de un pin de E/S como analógico o como digital.

ANSEL	R/W (1) ANS7	R/W (1) ANS6	R/W (1) ANS5	R/W (1) ANS4	R/W (1) ANS3	R/W (1) ANS2	R/W (1) ANS1	R/W (1) ANS0	Características Nombre de bit
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

ANSELH	R/W (1) ANS13	R/W (1) ANS12	R/W (1) ANS11	R/W (1) ANS10	R/W (1) ANS9	R/W (1) ANS8	Características Nombre de bit	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figura 7: Registros ANSEL y ANSELH.

Las entradas analógicas previamente configuradas están situadas en el registro ANSEL y el registro ANSELH es puesto a cero ya que no se utilizarán las entradas analógicas que contiene.

Para configurar las entradas y salidas de dicho microcontrolador es necesario manipular los registros TRIS en los cuales se escribió un cero para configurarlo como salida (Output) y un uno para configurarlo como entrada (Input). De esta manera quedaron configurados los puertos I/O del microcontrolador.

Los registros SPBRGH y SPBRG controlan la

velocidad de transmisión y recepción (TX, RX) del microcontrolador que está dada en baudios, y utiliza su cristal interno de 4MHz, la cual se ha calculado con la fórmula:

$$\text{Baudios} = \frac{\text{frecuencia de oscilación}}{(16)(9600)} - 1 = 25$$

Para configurar el convertidor AD, se usa el registro ADCON1 que se presenta en la figura 8, el cual cuenta con 3 bits los cuales se encargan de:

- ADFM: bit de selección del formato del resultado de la conversión A/D
- VCFG1: bit de configuración de voltaje de referencia, selecciona la fuente de voltaje de referencia bajo que se necesita para el funcionamiento del convertidor A/D.
- VCFG0: bit de configuración de voltaje de referencia, selecciona la fuente de voltaje de referencia alto que se necesita para el funcionamiento del convertidor A/D.

ADCON1	R/W (0) ADFM	R/W (0) VCFG1	R/W (0) VCFG0	Features Bit name				
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figura 8: Registro ADCON1.

Algunas de las partes más importantes del programa se describen a continuación:

**Despliegue de tiempo.** Para el despliegue de información se utilizaron displays de 7 segmentos en los cuales se realiza un barrido de información de tiempo de tal forma que la información está siendo continuamente multiplexada, esto se hace a una velocidad de alrededor de 10 microsegundos entre cada uno de los displays.

**Conversión ADC.** En la conversión analógica-digital se usa un byte, es decir 8 bits con el cual se podrá hacer una conversión de 0 a 255, que en este caso se estará monitoreando una señal hasta 200°C lo cual se ajusta correctamente a los rangos de la medición.

**Restas sucesivas.** Recordando que en lenguaje ensamblador sólo se cuenta con sumas y restas para hacer de las operaciones aritméticas y no se pueden realizar directamente divisiones, se realizan restas sucesivas para poder hacer dichas divisiones.

**Despliegue de temperatura.** Para el despliegue de TEMPERATURA al igual que en el de TIEMPO se utilizaron displays de 7 segmentos, en los cuales se realiza un barrido de los valores de temperaturas del

convertidor BCD, de tal forma que la información está siendo continuamente multiplexada por el mismo puerto donde se despliega la temperatura, esto se hace a una velocidad de alrededor de 10 microsegundos entre cada uno de los displays.

**Encendido del horno.** Para encender el horno se tiene una etiqueta, una vez situado en esta parte verificará si el horno se encuentra encendido para evitar ejecutar las demás instrucciones y tener una pérdida de tiempo. Si el horno se encuentra apagado se hará un salto de línea, ejecutando así una suma de D'255' (dato decimal) con lo que se encuentre en el registro donde están guardadas las decenas de horas, si las decenas de horas son mayores a 0 como por ejemplo igual a 1, la suma realizada será igual a 256 que en binario sería 1,00000000, dando así en uno el bit de acarreo que será reflejado en un registro. Al tener en 1 el acarreo se hará un salto de línea en el cual encenderá el bit 0 del puerto que está encargado de encender y apagar el horno. En caso de que la suma no sea mayor a D'255', no realizará el salto y ejecutará la instrucción que verifica las unidades de horas donde realizará las instrucciones anteriores con el fin de saber si existe tiempo, en caso de que no exista tiempo se volverán a realizar dichas instrucciones con las decenas de minutos y segundos para verificar si hay tiempo en el temporizador, de lo contrario el horno nunca encenderá. Esto se hizo con el fin de que no se pueda encender el horno al menos de que exista tiempo para evitar problemas de error de duración de tiempos.

**Apagado del Horno.** Al momento de encender el horno comenzará a ejecutarse una instrucción que contiene una rutina en donde se estará verificando el tiempo del temporizador; funciona de tal manera que si alguno de los registros de tiempo es mayor a "1" se regresa y no apaga el horno. De no ser así, verifica cada uno de los registros de tiempo para confirmar que estén en cero y apagar el horno. Al encender el horno la temperatura comenzará a incrementarse y el tiempo en el temporizador no se reducirá hasta que llegue a su temperatura programada, esto es posible gracias a que se creó una rutina de comparación de temperatura la cual consta de un registro en el cual se programará la temperatura a la que se quiere que comience el decremento del temporizador.

#### 4.1.4. Diseño de Hardware

El hardware utilizado en el diseño propuesto fue probado en el software Proteus, que permite hacer diagramas eléctricos y sus análisis, además de los diseños

del circuito impreso en PCB, en vistas 3D, entre otras.

Como se observa en la figura 9, se realizó el diseño sencillo del despliegue de tiempo y temperatura mediante ocho displays de 7 segmentos, con los transistores necesarios, dos LED como segundos, botones para la selección de parámetros y tiras de conectores hembra, en el que se haría la conexión con el controlador mediante cable de bus plano.

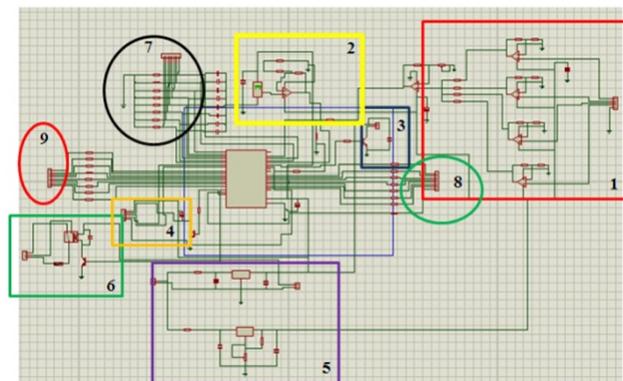


Figura 9: Diagrama eléctrico del controlador.

Este circuito se muestra seccionado en distintas etapas, las cuales son clasificadas por figuras, colores y número para su fácil distinción, como lo son:

1. Etapa de potencia para la amplificación de la señal del termopar tipo J, con su salida conectada a un sumador y un posterior divisor mediante resistencias, esto último para hacer un promedio. Esto debió ser así, para tener una mayor precisión en las lecturas de la temperatura, ya que los sensores son colocados a distintas alturas y miden la concentración de calor según su posición.
2. Etapa de potencia que amplifica la señal del sensor de temperatura LM35, el cual mide la temperatura del circuito, se procesa con el PIC y al superar una temperatura de 35°C el abanico es encendido para enfriar el sistema.
3. Salida del PIC conectada a un transistor para la activación del abanico utilizado para enfriar el sistema.
4. Salida del PIC que controla el encendido y apagado del módulo Bluetooth HC-06.
5. Etapa de reguladores, conectados desde la fuente de voltaje.
6. Relevador para el encendido y apagado del horno, controlado mediante el PIC
7. Conexión hacia los botones, para la interpretación de su señal en el PIC.
8. Conexión de tierras para los "display".
9. Conexión de activación para los 7 segmentos del "display"

## 5. Conclusiones

El proyecto tuvo el objetivo principal de reducir tiempos de producción, impactando directamente en sus tiempos de inventario (WIP), mediante el desarrollo de un temporizador controlado manual y remotamente, y que controle el horno, otorgándole en tiempo y forma la información al operador y que el supervisor pudiera recopilar esta información en el momento deseado.

Como diagnóstico inicial se realizaron análisis Yamazumi al proceso de horno y desmontado, lo cual otorgó mayor conocimiento sobre las áreas problemáticas y de oportunidad de mejora con que se contaba, ya que en un principio el problema planteado era únicamente hacer más práctico el uso del temporizador, debido a que el actual contaba con capacidad sobrada para el proceso y causaba problemas en la programación de tiempos. Posterior al análisis Yamazumi, se detectó que se podían eficientar las actividades del operador mediante la reducción de sus trayectos para verificación de datos y programación de tiempos.

Para resolver las problemáticas encontradas, se optó por modificar las actividades del operador, quitando aquellas que no agregaban valor al producto y modificando su modo de manejar el horno por medio de un control remoto, el cual le otorga la información que necesita en el momento preciso, sin tener que realizar recorridos largos para buscarla, además de un control manual directo en el horno.

Al realizar el diseño del software, tanto del control remoto y del control manual, se presentaron problemas de comunicación, debido a que al principio no se podían sincronizar los datos a transmitir y para ello se tuvo que crear una rutina en la cual se tiene que preguntar por la información deseada para responder con ella. Para esto, se diseñó un sistema de transmisión y recepción de datos que codificara la información requerida, y con las funciones de comunicación de MIT App Inventor se realizó con éxito.

## Referencias

- [1] Pons Achell, Juan Felipe, et al. The flow building® lean game: una dinámica innovadora creada en España para la enseñanza de lean construction. 2018.
- [2] Alberto Villaseñor Contreras y Edber Galindo Cota. Manual de lean manufacturing, guía básica. *Editorial Limusa. México*, 2007.
- [3] Mike Rother y Rick Harris. *Creando flujo continuo: una guía de acción para gerentes, ingenieros y*

*asociados de producción*. Lean Enterprise Institute, 2001.

- [4] Mike Rother y John Shook. *Learning to see: value stream mapping to add value and eliminate muda*. Lean Enterprise Institute, 2003.
- [5] Peter Hines y Nick Rich. The seven value stream mapping tools. *International journal of operations & production management*, 17(1):46–64, 1997.
- [6] Fawaz A Abdulmalek y Jayant Rajgopal. Analyzing the benefits of lean manufacturing and value stream mapping via simulation: A process sector case study. *International Journal of production economics*, 107(1):223–236, 2007.
- [7] José Ma Angulo Usategui, Ignacio Angulo Martínez, et al. *Microcontroladores PIC: Diseño práctico de aplicaciones*. 2003.
- [8] Enrique Mandado, L Menéndez, L Ferreira, y E López. Micro controladores pic. *Sistema integrado para el autoaprendizaje*. Barcelona: Marcombo, 2007.
- [9] Bong-Jun Kim, Yong Wook Lee, Byung-Gyu Chae, Sun Jin Yun, Soo-Young Oh, Hyun-Tak Kim, y Yong-Sik Lim. Temperature dependence of the first-order metal-insulator transition in VO<sub>2</sub> and programmable critical temperature sensor. *Applied physics letters*, 90(2):023515, 2007.
- [10] Yves Moser y Martin AM Gijss. Miniaturized flexible temperature sensor. *Journal of Microelectromechanical Systems*, 16(6):1349–1354, 2007.

# Vehículo transportador con estabilizador de carga

P.E. Montoya-Gomez<sup>1</sup>, L. Rodriguez-Contreras<sup>2</sup>, J. Palomares-Ruiz<sup>1,\*</sup>, J.G. Castro-Lugo<sup>1</sup>

<sup>1</sup>Maestría en ingeniería mecatrónica, Instituto Tecnológico Superior de Cajeme, Sonora, México

<sup>2</sup>Ingeniería en sistemas computacionales, Instituto Tecnológico Superior de Cajeme, Sonora, México

\*e-mail: jepalomares@itesca.edu.mx

**Resumen:** *El transporte de materiales o artículos dentro de una empresa con espacios físicamente delimitados puede resultar complicado e inclusive requerir una cantidad considerable de trabajo físico y detenimiento de maquinaria y obreros. En este artículo se abordará el tema de la construcción de un vehículo que permitirá facilitar el transporte de diversos materiales y equipos necesarios en el proceso de producción. Se iniciará dando una explicación de las partes que componen el dispositivo y la teoría que soporta su funcionamiento; seguido de una descripción detallada del proceso de construcción desde su inicio hasta la culminación del mismo y se finalizará mostrando los resultados de su funcionamiento.*

**Palabras clave:** Mecatrónica, estabilización de vehículo, control.

**Abstract:** *Moving materials from one place to another inside a building with limited space can be complicated or require a great deal of physical labor. This article will engage in the construction of a vehicle that will facilitate the transportation of materials. First, there will be an explanation of its components as well as the theory behind them; it will be followed by a detailed description of the construction process. lastly, it will show the results of its functioning.*

**Keywords:** Mechatronics, vehicle stabilization, control.

## 1. Antecedentes

Uno de los principales medios de transporte de productos es el terrestre, ya sea mediante tráiler, camión, camioneta o vehículo con adaptaciones para transportar cargas. En la mayoría de las ocasiones resulta necesario tomar ciertas medidas para lograr un transporte seguro de la carga unos ejemplos son: amarrar el cargamento, una o varias personas cuidando el producto, etc. Algunos factores que pueden causar el daño de las cargas son camino en mal estado, desnivelado, subidas, bajadas, por mencionar algunas [1].

Este es también un problema específico que se presenta en la empresa Edgewell planta Obregón, donde resulta sumamente común el que se presente la necesidad de trasladar un producto de un lugar a otro, ya sean piezas, herramienta especializada, materia prima, etc. Al realizar dicha operación resulta necesario tomar ciertas medidas precautorias con la finalidad de evitar algún tipo accidente, ya que dentro de las instalaciones de la empresa existen desniveles entre las áreas, vueltas, desperfectos en el piso, espacios reducidos, etc. Por lo que es necesario prestar especial cuidado en la posición

del cargamento al momento de transportarlo ya que el mismo podría caer y lastimarse o dañar a algún empleado. Lo anterior puede evitarse si se cuenta con un medio de transporte automatizado diseñado especialmente para satisfacer dicha tarea.

Por lo tanto con la finalidad de poder solucionar esta problemática es que se propone el diseño mecatrónico y la construcción de un vehículo transportador automatizado capaz de nivelar la superficie donde se encuentra depositada la carga para asegurar que la misma no pierda su posición original y por ende no se corra el riesgo de que esta pueda caerse o que el dispositivo pueda chocar con algo que se encuentre en sus alrededores al realizar la trayectoria solicitada o sobre la misma superficie.

Para lograr el equilibrio de la plataforma de carga se diseñó un mecanismo nivelador haciendo uso de un estabilizador giroscópico. Este es un dispositivo electrónico que mantendrá la carga horizontalmente nivelada en todo momento de la trayectoria, compensando las situaciones de desnivel y cambios abruptos de altura o movimiento. Existen diferentes aplicaciones para poder manipular dichos estabilizadores, esta tecnología se ha vuelto cada vez más accesible debido a que últimamente viene incorporada en muchos dispositivos.

El funcionamiento depende de tres partes básicas, todas con la misma importancia:

- Sensor. (Giroscopio/acelerómetro)
- Sistema de control.
- Actuador o motor.

El giroscopio mide la velocidad angular, significa que te dice cuántos grados ha rotado en 1 segundo y si el movimiento es en el sentido del reloj o al revés. Generalmente arroja la salida en la forma de 10 grados por segundo o -2 grados por segundo, con esta información el sistema de control sabrá que hacer para nivelar la carga.

Por su parte el acelerómetro es un dispositivo electrónico que mide las fuerzas de aceleración. Estas fuerzas pueden ser estáticas, como la fuerza de gravedad; o dinámicas, causada por el movimiento. Los acelerómetros no tienen el elemento del tiempo como los giroscopios y aunque su función principal no es necesariamente detectar el ángulo de inclinación, trabaja indirectamente ayudando como sensor secundario.

El sistema de control combinará la información recibida de los dos sensores para darse una idea precisa de lo que está ocurriendo con la carga, en tiempo real. Después calculará lo que necesita hacer para corregir la posición de la carga. La velocidad y la fuerza de la reacción necesarias también serán calculadas en este proceso para mantener la nivelación de la carga lo más fluida posible.

En resumen un estabilizador giroscópico recolectará datos de los sensores acerca del ángulo de inclinación y responderá con una respuesta matemática para los cambios recientes en el ángulo y usará los motores para corregirlo lo más rápido posible pero manteniendo la velocidad adecuada para asegurar un movimiento fluido y mantener la carga horizontal. Todo este proceso sucede varias veces por segundo y nunca se detiene mientras el control esté encendido.

Lo anterior irá montado sobre un carro impulsado por dos motores y se hará uso de un teléfono celular conectado por medio de bluetooth para maniobrar dicho carro.

## 1.1. Planteamiento del Problema

Dentro de una empresa siempre es necesario llevar artículos u objetos de un lado a otro. Al hacer esto se deben tomar ciertas medidas para evitar que se caigan; ya sea que alguien vaya cuidando la carga, que el carro donde se transporta tenga una canasta que permita depositar la carga, amarrarlo, etc. Lo anterior

es causado, mayormente, por desniveles en el camino, subidas, bajadas, irregularidades, entre otros.

## 1.2. Justificación

Este vehículo busca resolver un problema que se presenta en Edgewell Personal Care, planta Obregón, donde artículos son dañados debido al mal manejo durante su transportación dentro de las instalaciones de la empresa. Lo cual tiene un costo monetario ya sea directo, el producto se daña; o indirecto, retraso en la producción debido a que el artículo no llega a tiempo. El artículo puede ser producto terminado, piezas de máquina, materia prima, insumos, etc.

## 1.3. Objetivos

### 1.3.1. Objetivo general

Construir un vehículo prototipo capaz de nivelar la carga transportada que asegure la posición sobre la superficie.

### 1.3.2. Objetivos específicos

- Definir peso máximo y tipo de volumen de carga
- Establecer la velocidad ideal
- Definir modo de operación
- Seleccionar motor que nos proporcione la potencia necesaria para nivelar la plataforma
- Seleccionar materiales para construir el vehículo (chasis, llantas)
- Programar los sistemas
- Hacer pruebas

## 2. Antecedentes

El propósito de este capítulo es dar una idea de lo que ya existe con respecto al tema de mover cargas de un lugar a otro [2]. El tipo de vehículo a utilizar depende de las condiciones y necesidades propias de cada situación; tipo de terreno, espacio disponible, tamaño de carga, tipo de carga, entre otras [3].

Se va a clasificar dependiendo del tipo de manejo de los vehículos: manual, eléctrico y automatizado. [4]

## 2.1. Tipos de vehículos

### 2.1.1. Vehículos manuales

En este rubro entrarán todos los vehículos que dependen de la fuerza de una persona para moverlos.

Ventajas:

- ✓ Costo bajo.
- ✓ Se pueden hacer justo a las necesidades de la tarea. Usando tuberías de creform.
- ✓ Muy comunes.
- ✓ Facilidad de uso.

Desventajas:

- × Esfuerzo físico.
- × La carga está limitada por la fuerza de la persona que lo utiliza.
- × Es necesario cuidar la carga.

Ejemplos:

- Diablitos.
- Transpaletas.
- Tecles.
- Carros y carretillas de transporte.

### 2.1.2. Vehículos motorizados

Son la variante motorizada de los anteriores. Pueden ser eléctricos o de combustión interna.

Ventajas:

- ✓ Pueden llevar cargas grandes sin esfuerzo alguno.
- ✓ Rapidez.
- ✓ Más opciones de manejo de cargas.

Desventajas:

- × Alto costo
- × Las cargas son propensas a sufrir accidentes si no se tiene el cuidado apropiado.
- × Áreas de uso limitadas.

- × Se requiere capacitar al operador.

Ejemplos:

- Montacargas
- Transpaletas eléctricas.
- Carro eléctrico con adecuaciones para llevar cargas.

### 2.1.3. Vehículos automatizados

Son vehículos previamente programados para seguir una ruta o una matriz de acciones.

Ventajas:

- ✓ Pueden llevar grandes cargas sin esfuerzo alguno.
- ✓ Rapidez.
- ✓ Contiene sensores para monitorear la carga.
- ✓ No es necesario que una persona lo controle.

Desventajas:

- × Alto costo.
- × Dependiendo del tipo puede estar limitado a la ruta o tarea programada.

Ejemplos:

- SWISSLOG TransCar
- Kiva robot
- Toyota BT con piloto automático.

## 2.2. Estabilizador Giroscópico

El objetivo principal de la tecnología es facilitar las tareas diarias de la vida, ya sea en la casa, en el trabajo, en la calle o en cualquier parte. Se pensó en un vehículo con estabilizador de carga ya que surgió una problemática de tiempo y esfuerzo al mover materiales a través de los pasillos de una empresa.

Se buscará hacer uso de técnicas existentes para solucionar el problema antes mencionado. Este trabajo se basará en diseñar y construir un vehículo utilizando el principio de un nivelador giroscópico y un vehículo que se controlará remotamente.

El nivelador giroscópico consta de tres partes: giroscopio/acelerómetro, sistema de control y por lo menos dos servomotores como sistema actuador. A continuación se dará una explicación del funcionamiento de cada uno y finalmente como se integrarán en el proyecto.

### 2.2.1. Giroscopio

En esta sección se echará un vistazo al giroscopio como un sensor y algunos giroscopios MEMS que han sido fabricados con anterioridad. Este es un dispositivo para medir el radio de rotación de un objeto, su funcionamiento está basado en el principio de conservación del momento angular. Las siguientes ecuaciones y figura 1 ilustran dicho principio y como es usado para medir el radio de rotación ( $\Omega$ ) de un cuerpo [5,6].

$$L_a = mv \times r = m\Omega r^2$$

$$\frac{\partial L_a}{\partial t} = 2m\Omega \times r \frac{\partial r}{\partial t} = F \times r$$

$$F = 2m\Omega \times v$$

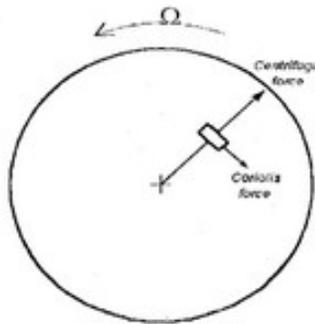


Figura 1: Principio de conservación angular

La fuerza coriolis es una fuerza ficticia ejercida sobre un cuerpo cuando se mueve en un marco de referencia rotatorio. Es una fuerza ficticia porque es un subproducto de medir coordenadas con respecto a un sistema de coordenadas rotatorio opuesto a un marco de referencia fijo. La velocidad del cuerpo “ $v$ ” es con respecto a la referencia fija del plano sobre el giroscopio.

### 2.2.2. Historia del giroscopio

En tiempos antiguos, la gente se dio cuenta que el trompo tenía la habilidad de balancearse en posición

vertical mientras rotaba rápidamente. Las sociedades romana y griega lo usaban como entretenimiento. No fue hasta finales del siglo 18 e inicios del 19 que los científicos y marineros empezaron a intentar usarlo como herramienta científica. En aquellos tiempos los marineros dependían de los sextantes para navegar, medían el ángulo entre el horizonte y ciertas estrellas. Esto tenía muchas desventajas cuando el clima era malo o estaba nublado. Un capitán inglés llamado John Serson se dio cuenta, en la década de 1740, que el trompo tenía la tendencia de mantenerse nivelado incluso cuando la superficie se inclinaba. Esto lo inspiró para inventar el espejo giratorio, el cual es el precursor de los giroscopios. El primer giroscopio moderno fue diseñado en 1817 por Johann Gottlieb Friedrich von Bohnenberger, profesor de la universidad de Tuebingen en Alemania. Fue simplemente llamado “la máquina” y estaba hecho con una bola pesada en lugar de rueda, pero como no tuvo aplicación científica se quedó en el olvido. En 1851 el científico francés León Foucault hizo unos experimentos usando un péndulo largo suspendido libremente para observar la rotación de la tierra. Corroboró esta observación usando un trompo de la misma manera. Situó una rueda, rotando a gran velocidad, soportado en un anillo de tal manera que el eje de la rueda se pudiera mover de manera independiente del anillo. De hecho, el anillo se movió a través del curso del día, como si estuviera conectado a la superficie de la tierra rotando. El eje de la rueda se mantuvo apuntando a su dirección original, confirmando que la tierra rotaba en un periodo de 24 horas. Foucault llamó a su rueda giratoria “giroscopio”, por las palabras griegas “gyros” (revolución) y “skopein” (ver).

Con la llegada de los MEMS a mediados del siglo 20, los científicos alrededor del mundo trataron de reproducir sensores miniaturizados en silicio usando tecnología de micro fabricación. Los primeros giroscopios vibratorios micro maquinados fueron hechos exitosamente en cuarzo. Como el proceso de fabricación de cuarzo no es compatible con la fabricación de circuitos integrados se hicieron esfuerzos para reproducir los resultados en silicio. El laboratorio Charles Draper fue el primero en demostrar el giroscopio vibratorio de silicio en 1991 [7–9].

### 2.2.3. Giroscopio MEMS

Casi todos los giroscopios micro maquinados usan elementos vibratorios mecánicos para medir la rotación. Ellos no cuentan con partes rotativas que requieran bearings, debido a esto pueden ser fácilmente

miniaturizados y fabricados en lotes usando técnicas de micro maquinado. Todos los giroscopios vibratorios están basados en la transferencia de energía entre dos modos de vibración de una estructura causada por la fuerza Coriolis.

Un giroscopio MEMS, también llamado de estado sólido, está formado por un cuerpo que presenta simetría en su rotación; en este caso, supóngase que la masa ( $m$ ) se desplaza dentro del chip a una velocidad  $V$ ; cuando al chip se le aplica un momento de fuerza, este girara a una velocidad angular  $\Omega$ . Esta combinación de movimientos rotacional y lineal genera la llamada fuerza de coriolis, que será perpendicular al eje de movimiento lineal [10].

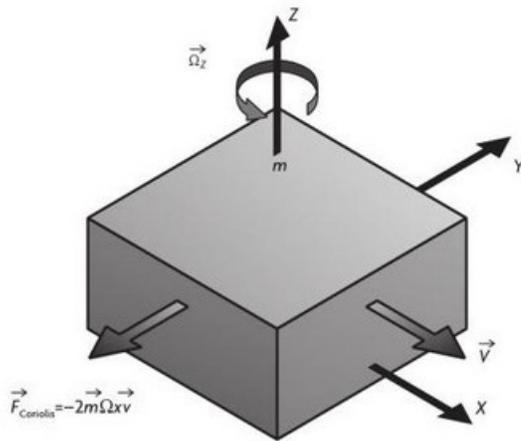


Figura 2: Fuerza coriolis

En un giroscopio de estado sólido. En general se utiliza el desplazamiento en la masa móvil provocado por el efecto de la fuerza de coriolis, para convertirlo en variación de capacitancia mediante un principio de transducción capacitivo, en un cambio de resistencia mediante un principio de transducción piezoresistivo y en un cambio en voltaje mediante un transductor piezoeléctrico.

La arquitectura que más se utiliza para este tipo de sensores es un par de masas con movimiento lineal de vaivén en sintonía. Esta arquitectura es muy utilizada debido a su gran facilidad para ser integrada en sistemas con principio de transducción de capacitancia variable. De igual forma que los acelerómetros con principio de transducción de capacitancia variable, las masas móviles constituyen los electrodos de los capacitores, la variación de desplazamiento  $y$ , por tanto, de capacitancia, lo cual se logra debido a que la fuerza de coriolis presente en cada masa empujará a cada una en direcciones opuestas, gracias a que la dirección de su velocidad es opuesta. Esta variación de capacitancia será proporcional a la velocidad angular aplicada al chip.

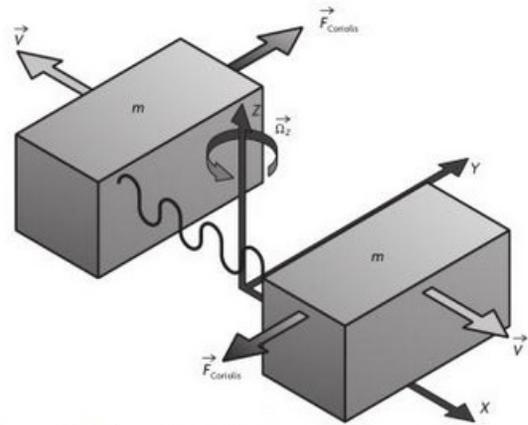


Figura 3: Giroscopio de estado sólido

El es una estructura parecida a una horquilla que está en constante movimiento hacia adelante y hacia atrás. Se sostiene usando cristales piezo eléctricos. Cuando se inclina este arreglo, los cristales experimentan una fuerza en dirección de la inclinación. Esto es el resultado de la inercia del movimiento de la horquilla. Los cristales producen una corriente en consenso con el efecto piezo eléctrico y la corriente es amplificada. Finalmente los valores son refinados por un microcontrolador.

Todas las aplicaciones prácticas del giroscopio se basan en dos características fundamentales: inercia giroscópica y precesión. La inercia giroscópica es la tendencia de un cuerpo que gira a conservar su plano de rotación, o sea, que si al giroscopio se le cambia de posición o se le desplaza, el plano de rotación, eje  $AA'$ , mantiene inalterable su dirección original.

La utilidad del giroscopio reside en su capacidad de facilitar una línea de referencia que no resulte afectada por fuerzas perturbadoras y que sirve para gobernar o vigilar uno o varios ejes o como referencia fija o direccional.

### 2.3. Acelerómetro

Un acelerómetro es un dispositivo electromecánico que mide fuerzas de aceleración. Estas fuerzas pueden ser estáticas, como la fuerza gravitacional constante que nos jala hacia nuestros pies, o pueden ser dinámicas causadas por movimiento o vibración del acelerómetro. La gran mayoría están basados en cristales piezo eléctricos, pero son muy grandes y torpes. La gente trató de desarrollar algo más pequeño, que pudiera incrementar su aplicabilidad y empezaron a buscar en el campo de la microelectrónica, este fue el nacimiento de los acelerómetros MEMS.

El primer acelerómetro micro maquinado fue diseñado

en 1979 en la Universidad Stanford, pero tuvieron que pasar 15 años para que estos dispositivos fueran aceptados para grandes volúmenes de aplicaciones. Son dispositivos con un gran potencial comercial, son compactos, proveen bajo poder y son robustos a momento de medir. A menudo varios sensores son combinados para proveer medición de múltiples ejes y datos más precisos [11,12].

### 2.3.1. Aplicaciones

Los acelerómetros están siendo incorporados cada vez en dispositivos electrónicos teléfonos celulares, reproductores de música, automóviles, etc. Son usados para contar pasos, cambiar modo vertical u horizontal en los teléfonos, cambiar de canción tocando el teléfono, detectar choques y desplegar las bolsas de aire en los autos. Estos son solo algunos ejemplos.

Un acelerómetro trabaja con el principio del efecto piezo eléctrico. Imagine un cubo con una pequeña pelota dentro. Las paredes de esta caja están hechas de cristales piezo eléctricos. Cuando la caja es inclinada, la pelota es forzada a moverse en dirección de la inclinación, debido a la gravedad. La pared con la que la pelota choca, crea pequeñas corrientes piezo eléctricas. En total hay tres pares de paredes opuestas dentro del cubo, cada par corresponde a un eje en un espacio tridimensional: X, Y y Z. Dependiendo de la corriente producida en las paredes se puede determinar la dirección de la inclinación y si magnitud.

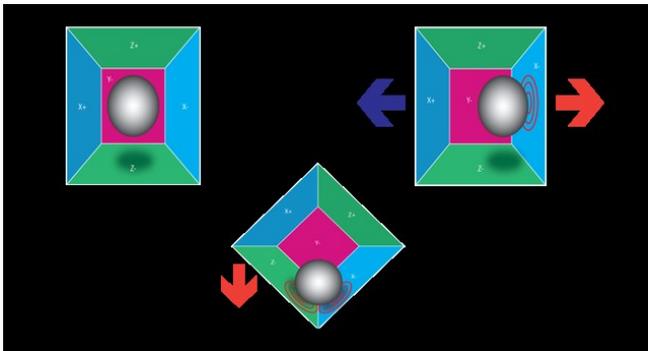


Figura 4: Acelerómetro

## 2.4. Servo motores

Los servomotores pueden ser utilizados en diversas aplicaciones industriales que requieren de una exigencia elevada en dinámica, precisión de posicionamiento y velocidad, además, de un control confiable y funcionalmente fácil de manejar. En su definición más básica, es un motor que puede ser controlado en su

velocidad de funcionamiento y en la posición dentro de un rango de operación para ejecutar la actividad requerida. Este control es realizado mediante un dispositivo llamado encoder, que mediante una señal electrónicamente codificada, indica las acciones de velocidad y movimiento a ejecutar.

Los primeros servomotores utilizaban un sistema de funcionamiento con corriente continua, en la que los electrones generadores de corriente se mueven en un solo sentido. En la actualidad los servomotores utilizados son de corriente alterna, en estos los electrones cambian de sentido en todo momento (alternan), realizando la transformación de energía mecánica a eléctrica. Estos últimos admiten voltajes más altos.

### 2.4.1. Características principales:

- Prestaciones y par elevado.
- Fiabilidad de funcionamiento.
- Bajo mantenimiento.
- Gran exactitud en el control de velocidad y posición.
- Capacidad de velocidades muy alta.
- Pérdidas en el rotor muy bajas.
- Rotor con poca inercia.
- Construcción cerrada, útil para trabajar en ambientes sucios.
- Alta gama de potencias (de 100 w a 300Kw).

### 2.4.2. Funcionamiento

El sistema servo se comunica mediante pulsos eléctricos a través de un circuito de control para determinar el ángulo de posición del motor, “el servo espera recibir un pulso cada 20 milisegundos (0.02 segundos). La longitud del pulso determinará los giros de motor; un pulso de 1.5 ms. por ejemplo, hará que el motor vaya a una posición de 90 grados (posición neutra). Si el pulso es menor se acercará a los 0 grados. Si el pulso es mayor de 1.5 ms. el eje se moverá acercándose a los 180 grados.

El hecho de que el tamaño de los servomotores sea más reducido no incide en su potencia, puesto que, precisamente, una característica importante de estos equipos es la capacidad de torque que tienen con una estructura física reducida, lo que implica un menor peso. En fuerza y potencia, los servomotores igualan a los motores mecánicos e hidráulicos, puesto

que tienen variadas posibilidades. Gracias a estas habilidades, los servos se usan en aplicaciones como corte, impresión, etiquetado, empaçado, manipulación de alimentos, robótica y automatización de fábricas.

## 2.5. Arduino

Arduino está compuesto de hardware y software. Es un circuito impreso específicamente diseñado para usar un microcontrolador así como otras entradas y salidas. También contiene otros componentes electrónicos que son necesarios para que el microcontrolador funcione con todas sus capacidades. Los microcontroladores son computadoras pequeñas contenidas dentro de un circuito integrado o chip de computadora, y son excelentes para programar y controlar dispositivos electrónicos.

Se escribe código en el software de Arduino para decirle al microcontrolador que hacer. Por ejemplo, con una línea de código se le puede decir al LED que se prenda y apague. Si conectas un botón y agregas otra línea de código, le puedes decir al LED que solo se prenda cuando el botón es presionado. De esta manera, puedes programar el comportamiento de un sistema rápidamente, lo cual sería más difícil de lograr sin un microcontrolador.

Similar a una computadora convencional, Arduino puede realizar una multitud de funciones, pero no es de mucho uso por sí solo. Requiere de entradas y salidas para ser útil. Estas entradas y salidas le permiten interactuar con el mundo.

## 3. Carro

Un carro es un vehículo o armazón con ruedas utilizado para transportar objetos diversos. En este caso servirá de base para el estabilizador giroscópico. En este caso será un chasis con 4 ruedas, dos de ellas contarán con un motor. Dichos motores serán controlados por un arduino.

### 3.1. Dirección del vehículo

Para la dirección del vehículo se utilizará el método de conducción diferencial, el cual consiste en dos llantas montadas sobre el mismo eje y cada una se puede mover independientemente ya sea hacia adelante o hacia atrás.

El arreglo de las llantas es el siguiente:

- 2 llantas motorizadas
- 2 ruedas giratorias

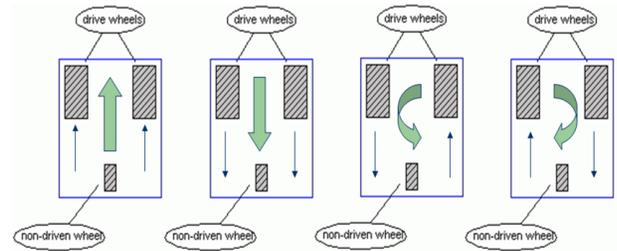


Figura 5: Dirección diferencial

### 3.2. Motor de corriente directa

un motor de corriente directa convierte energía eléctrica directa en energía mecánica. Produce una acción rotacional mecánica en el eje el cual está físicamente acoplado a una máquina o dispositivo mecánico que hace algún tipo de trabajo.

Los motores DC están equipados para muchas aplicaciones industriales. Por ejemplo, estos motores son usados cuando se requiere un control preciso de velocidad o posición de una carga y se puede acelerar o desacelerar rápida y fluidamente. Además, la dirección puede ser fácilmente revertida.

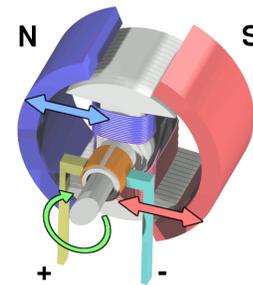


Figura 6: Motor DC

## 4. Método

En este capítulo, se describirá la construcción del proyecto, así como los componentes que lo conforman, el software usado y dificultades que surgieron durante este proceso.

El Proyecto básicamente está compuesto en dos partes:

- La parte inferior que incluye la base que soportará la carga en su totalidad y todo lo necesario para que el vehículo avance y retroceda.
- La parte superior que tendrá la plataforma donde irá la carga, así como los sensores y motores necesarios

para que esta se nivele según la inclinación a la que sea sometida.

#### 4.1. Parte inferior

Se utilizó la parte inferior de una andadera rollator como chasis del carro. Se tomó esta decisión ya que cuenta con un tamaño adecuado y está comprobado que puede soportar cargas mayores a 50 kg.



Figura 7: Andadera rollator

Un par de adecuaciones fueron necesarias para lograr el carro deseado. Se le instalaron dos llantas de bicicleta, de medidas 12 1/2 X 2 1/4 y a cada una de ellas se le instaló un motor Feber de 12 V. Las llantas se unieron con una barra de acero para brindar una mayor estabilidad.

Debido a la forma tan particular de la transmisión de estos motores se tuvo que idear un adaptador para unirlos a las llantas. Se soldó un tramo de PTR de 3 cm al centro de las llantas y se le dio la forma deseada con la ayuda de una prensa.



Figura 8: Llanta, Adaptador, Motor

El control de los motores y la dirección del vehículo se hizo a través de una batería Feber, dos puentes H BTS7960 (uno para cada motor) y un arduino.



Figura 9: Puente H. BTS7960

Los comandos son enviados desde un teléfono celular con la ayuda de la aplicación “Bluetooth RC car” y un módulo bluetooth de arduino. Dicha aplicación requiere el módulo bluetooth HC-05, este modelo ya está discontinuado, y fue un poco difícil encontrarlo.

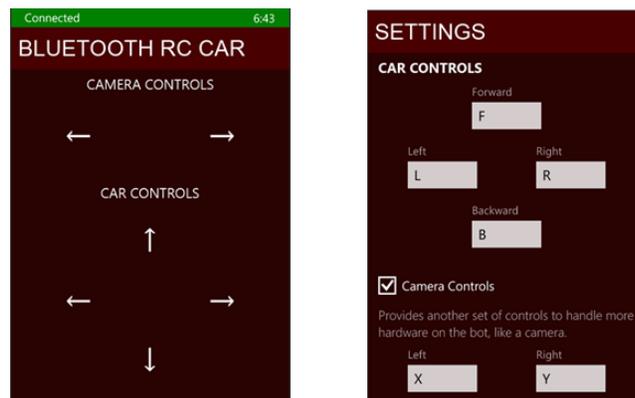


Figura 10: Aplicación móvil

Después de unir todo lo anterior surgió un problema, la estructura resultó ser muy débil y no soportaba la potencia de los motores. También se tuvo que tomar en cuenta que en la parte de arriba estaría la plataforma con la carga, lo cual conlleva un peso extra. Para solucionarlo se soldó un marco de aluminio en la parte superior de los postes de las llantas. Esto proporcionó la solidez necesaria para soportar la carga y movimiento para lo que el vehículo fue diseñado.

Sobre esta plataforma se encuentra una base de acrílico. La decisión de usar acrílico es debido a que es resistente y ligero. Estas dos características se tomaron en cuenta desde el inicio del diseño ya que el peso repercutiría directamente en el desempeño de los motores.

A continuación, se muestra el diagrama de las conexiones necesarias para la movilidad del vehículo.

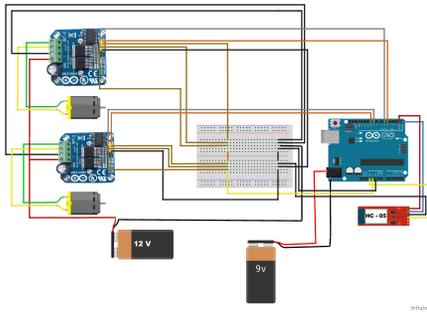


Figura 11: Diagrama de conexiones

Finalmente se desarrolló un programa en arduino para activar los motores. Lo primero que se hizo fue programar las opciones en la aplicación del teléfono, esta aplicación envía letras al módulo bluetooth. Una vez definido esto se procedió a hacer el programa el cual activa los motores dependiendo las letras que se envían. De tal manera que si se quiere avanzar se activan los dos motores hacia adelante con la letra “A”, para retroceder se activan los motores en sentido contrario con la letra “E”, para girar a la izquierda se activa únicamente el motor de la derecha con la letra “B” y finalmente para girar a la derecha se activa el motor de la izquierda con la letra “D”. Este es el código:

```
int izqAdelante = 5;
int izqAtras = 6;
int derAdelante = 9;
int derAtras = 10;
int vel = 50; // Velocidad de los motores (0-255)
int estado = 'g'; // inicia detenido
void setup() {
  Serial.begin(9600); // inicia el puerto serial para
  comunicación con el Bluetooth
  pinMode(derAtras, OUTPUT);
  pinMode(derAtras, OUTPUT);
  pinMode(izqAdelante, OUTPUT);
  pinMode(izqAtras, OUTPUT);
}
void loop() {
  if(Serial.available() > 0){ // lee el bluetooth y
  almacena el estado
    estado = Serial.read();
  }
  if(estado=='A'){ // Adelante
    Serial.println(estado);
    analogWrite(derAtras, 0);
    analogWrite(izqAtras, 0);
    analogWrite(derAdelante, vel);
    analogWrite(izqAdelante, vel);
  }
  if(estado=='D'){ // Derecha
    Serial.println(estado);
    analogWrite(derAtras, 0);
    analogWrite(izqAtras, 0);
    analogWrite(derAdelante, 0);
    analogWrite(izqAdelante, vel);
  }
  if(estado=='C'){ // Alto
    Serial.println(estado);
    analogWrite(derAtras, 0);
    analogWrite(izqAtras, 0);
    analogWrite(derAdelante, 0);
    analogWrite(izqAdelante, 0);
  }
  if(estado=='B'){ // Izquierda
    Serial.println(estado);
    analogWrite(derAtras, 0);
    analogWrite(izqAtras, 0);
  }
}
```

```
analogWrite(izqAdelante, 0);
analogWrite(derAdelante, vel);
}
if(estado=='E'){ // Reversa
  Serial.println(estado);
  analogWrite(derAdelante, 0);
  analogWrite(izqAdelante, 0);
  analogWrite(derAtras, vel);
  analogWrite(izqAtras, vel);
}
}
```

## 4.2. Parte superior

Esta plataforma debe nivelarse automáticamente tomando como referencia el ángulo de la base de acrílico del punto anterior. Para hacer esto se usó el acelerómetro MPU6050, el cual fue previamente fijado a la base.



Figura 12: MPU6050

Este sensor detecta la inclinación del carro y la manda al arduino. Este a su vez lo procesa usando el filtro de kalman, el filtro es esencial ya que ayuda a que el movimiento de los motores sea fluido. Una vez que se obtuvo la información de la inclinación de los ejes X y Y se manda a los servomotores lo cuales se encargarán de corregir la inclinación de la superficie de carga con una respuesta en tiempo de real para mantenerla en posición horizontal en todo momento.

Esta información es enviada a los servomotores, Rován 2050D, los cuales suben o bajan dependiendo de la lectura. Los motores están conectados a la plataforma usando una bisagra tipo “T”. estas bisagras permiten que cada motor mueva un eje de la plataforma.



Figura 13: Servo motor(Rován 2050D) y Bisagra

El poste central de la base superior, que es el que sostiene la mayor parte del peso de la carga esta formado

por una junta universal y una barra de extensión. Este poste cuenta con una base circular en cada extremo, las cuales están atornilladas a ambos acrílicos y proporcionan la estabilidad necesaria. La combinación de las bisagras y la junta universal permiten que la base de carga se pueda inclinar en todas direcciones.



Figura 14: Junta universal y Barra de extensión

A continuación se muestra el diagrama de conexiones de la base de carga:

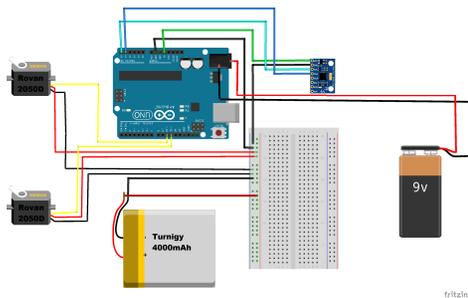


Figura 15: Diagrama de conexiones

Programa de arduino para nivelar la base:

```
void processAccelGyro()
{
  // Obtiene el estado del sensor
  mpuIntStatus = mpu.getIntStatus();
  // Obtiene la cuenta de FIFO del sensor
  fifoCount = mpu.getFIFOCount();
  // Revisa que no haya desborde (No debe de pasar)
  if ((mpuIntStatus & 0x10) == fifoCount == 1024)
  {
    // Reinicia la cuenta
    mpu.resetFIFO();
    Serial.println(F("FIFO Desborde!"));
    return;
  }
  if (mpuIntStatus & 0x02)
  {
    if (fifoCount > packetSize)
    {
      return; // fifoCount = mpu.getFIFOCount();
      // Lee la informacion de FIFO
      mpu.getFIFOBytes(fifoBuffer, packetSize);
      fifoCount -= packetSize;
      // Se reinicia el buffer para prevenir desborde
      mpu.resetFIFO();
      // Despliega los angulos en grados
      mpu.dmpGetQuaternion(&q, fifoBuffer);
      mpu.dmpGetGravity(&gravity, &q);
      mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
      mpuPitch = ypr[PITCH] * 180 / M_PI;
      mpuRoll = ypr[ROLL] * 180 / M_PI;
    }
  }
}
```

```
mpuYaw = ypr[YAW] * 180 / M_PI;
mpu.resetFIFO();
blinkState = !blinkState;
digitalWrite(LED_PIN, blinkState);
mpu.resetFIFO();
//Se manda la información a los servos
Servo1.write(-mpuPitch + 90);
Servo2.write(mpuRoll + 90);
//delay(10);
mpu.resetFIFO();
}
```

## 5. Resultados y conclusiones

La finalidad de este trabajo consistió en el desarrollo de la metodología para la construcción de un vehículo capaz de nivelar la base de carga de manera automática. En las secciones anteriores se habló acerca de las partes que lo componen, un poco de teoría acerca de ellos y como se construyó. A continuación se mostrarán los resultados de todo lo anterior.

Componentes del vehículo

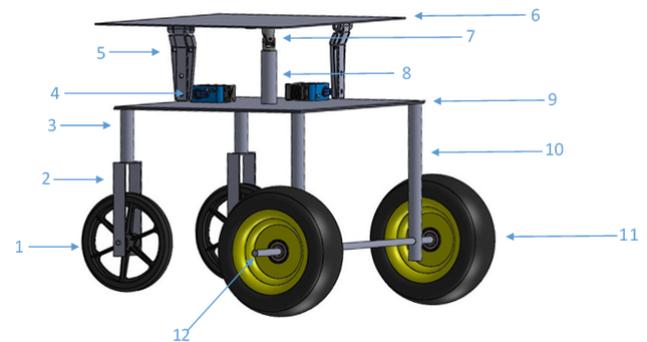


Figura 16: Diseño del vehículo

1. 2 Ruedas de 8"
2. 2 Horquillas 2.5"
3. 2 Postes delantero 4" largo X 1" de diámetro.
4. 2 Servo motores Rovon 2050D.
5. 2 Bisagra tipo T.
6. 1 Base de acrílico.
7. 1 Junta universal 3/4".
8. 1 Barra de extensión.
9. 1 Base de acrílico.
10. 2 Postes trasero 12" largo X 1" de diámetro.
11. 2 Llanta de 12".
12. 1 Eje 34" largo X 1/2" de diámetro.

La velocidad que se decidió utilizar es de 1 km/hora. Esto es debido a que se debe pensar en la carga y que puede haber obstáculos en el camino, personas u objetos. Al momento de detenerse no se detiene de golpe, sino que lo hace paulatinamente para evitar que la carga se deslice.

La carga máxima es de 40 Kg. Esto para evitar forzar los motores, tanto del carro como los de la plataforma de carga.

El modo de operación será a través de una aplicación móvil. En este trabajo se usó la aplicación “Bluetooth RC car” pero realmente se puede usar cualquier aplicación que envíe letras y se pueda configurar.

Se utilizaron los motores con mayor torque que se pudieron encontrar, Rovon RC-2050D. Estos Servo motores ofrecen un torque de 50Kg/in, pero si se toma en cuenta que la carga estará situada a 2 pulgadas del centro entonces el torque se reduce a 25Kg.

## Referencias

- [1] Carme Miralles-Guasch. Transporte y territorio urbano: del paradigma de la causalidad al de la dialéctica. *Documents d’analisi geogràfica*, (41):0107–120, 2002.
- [2] Augie Widyotriatmo, Keum-Shik Hong, y Lafin H. Prayudhi. Robust stabilization of a wheeled vehicle: Hybrid feedback control design and experimental validation. *Journal of Mechanical Science and Technology*, 24(2):513–520, 2010.
- [3] R. Attia, R. Orjuela, y M. Basset. Coupled longitudinal and lateral control strategy improving lateral stability for autonomous vehicle. In *2012 American Control Conference (ACC)*, volume 3, pages 6509–6514, June 2012.
- [4] C. E. Beal y J. C. Gerdes. Model predictive control for vehicle stabilization at the limits of handling. *IEEE Transactions on Control Systems Technology*, 21(4):1258–1269, July 2013.
- [5] Ruth E Mayagoitia, Anand V Nene, y Peter H Veltink. Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems. *Journal of biomechanics*, 35(4):537–542, 2002.
- [6] Julien Favre, BM Jolles, O Siegrist, y Kamiar Aminian. Quaternion-based fusion of gyroscopes and accelerometers to improve 3d angle measurement. *Electronics Letters*, 42(11):612–614, 2006.
- [7] N Barbour, J Connelly, J Gilmore, P Greiff, A Kourepenis, y M Weinberg. Micromechanical silicon instrument and systems development at draper laboratory. In *Guidance, Navigation, and Control Conference*, page 3709, 1996.
- [8] Charles S Draper y Roger B Woodbury. Gyroscopic apparatus, July 3 1956. US Patent 2,752,792.
- [9] JH Connelly, JP Gilmore, y MS Weinberg. Micro-electromechanical instrument and systems development at the charles stark draper laboratory. 1995.
- [10] A Kourepenis, J Borenstein, J Connelly, R Elliott, P Ward, y M Weinberg. Performance of mems inertial sensors. In *Position Location and Navigation Symposium, IEEE 1998*, pages 1–8. IEEE, 1998.
- [11] Chin-Woo Tan y Sungsu Park. Design of accelerometer-based inertial navigation systems. *IEEE Transactions on Instrumentation and Measurement*, 54(6):2520–2530, 2005.
- [12] Benedict B O’Brien, Brent E Burns, y John A Geen. Micromachined accelerometer gyroscope, February 28 1995. US Patent 5,392,650.

# Diseño de un controlador difuso para un seguidor solar utilizando fuzzy toolbox de MATLAB

J. Erro-Quiñonez<sup>1</sup>, J. Ruelas-Ruiz<sup>2,\*</sup>, A. Ramirez-Treviño<sup>2</sup>, F. Muñoz-Beltrán<sup>2</sup>

<sup>1</sup>Egresado de la maestría en ingeniería mecatrónica, Instituto Tecnológico Superior de Cajeme, Sonora, México

<sup>2</sup>Maestría en ingeniería mecatrónica, Instituto Tecnológico Superior de Cajeme, Sonora, México

\*e-mail: eruelas@itesca.edu.mx

**Resumen:** En las siguientes páginas se entrega una propuesta de metodología que usan sus servidores para realizar proyectos la cual se aplicó para diseñar un controlador con lógica difusa que nos permita controlar el movimiento de un actuador de un seguidor solar. Se presenta una breve descripción de cada módulo y su funcionamiento. Sobre la lógica difusa se encontrará una breve introducción. el programa completo en toolbox fuzzy y la interfaz con Simulink de MATLAB.

**Palabras clave:** Bias, MATLAB, SolidWorks, Proteus.

**Abstract:** En las siguientes páginas se entrega una propuesta de metodología que usan sus servidores para realizar proyectos la cual se aplicó para diseñar un controlador con lógica difusa que nos permita controlar el movimiento de un actuador de un seguidor solar. Se presenta una breve descripción de cada módulo y su funcionamiento. Sobre la lógica difusa se encontrará una breve introducción. el programa completo en toolbox fuzzy y la interfaz con Simulink de MATLAB.

**Keywords:** Bias, MATLAB, SolidWorks, Proteus.

## 1. Introducción

El presente trabajo propone una alternativa de control a través del uso de la lógica difusa aplicada a un seguidor solar. Con la finalidad de orientar las celdas fotovoltaicas en un ángulo aproximado a noventa grados con respecto a los rayos solares, buscando acercarse al punto de máxima potencia del panel fotovoltaico.

Lo anterior puede lograrse aprovechando la experiencia de un experto en seguimiento solar en conjunto con el fuzzy toolbox de MATLAB® que nos permite traducir las expresiones lingüísticas a un valor que pueda ser entendido por el microcontrolador. Para desarrollar el modelo del controlador se usó una metodología modular que separa el sistema en módulos separados que se interconectan entre sí, en este caso fueron el módulo de sensores, módulo lógico o de programación, módulo de electrónica de control, módulo de electrónica de potencia, módulo de alimentación y modulo mecánico, en el desarrollo se observaron a detalle cada uno, se incluye una parte donde se presentan las pruebas realizadas con el modelo, así como los resultados obtenidos.

## 2. Desarrollo

### 2.1. Metodología modular

Esta metodología propone tomar al sistema que se desea obtener como una caja negra identificando sus entradas y salidas, para posteriormente identificar qué tipo de modulo se requiere a la entrada, durante el proceso y la salida. Cada módulo debe tratarse por separado sin olvidar que debe ajustarse al posterior o anterior.

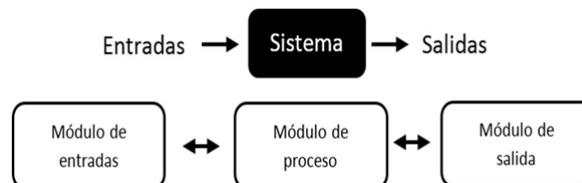


Figura 1: Metodología modular.

#### 2.1.1. Descripción del proceso

El sistema debe tener la capacidad de posicionar el panel fotovoltaico en un ángulo próximo a los 90° con

respecto a los rayos solares siguiendo la trayectoria solar. Como se muestra en la siguiente figura 2.

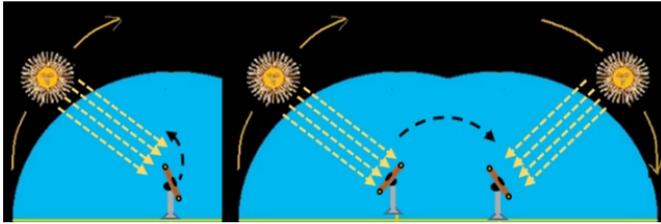


Figura 2: Seguimiento solar.

Para lograr el objetivo el sistema se separó en los siguientes módulos:

### 2.1.2. Módulo de entrada

Su función es captar la intensidad de luminosidad de los rayos solares y convertirlos en un voltaje que va de los 0 a los 5 VDC. Para lo cual se propone un arreglo de seis LDR (fotoresistor) [1] con una resistencia de  $10K\Omega$ , cuyo punto de medición es el divisor de voltaje y se alimentará con 5 VDC, el arreglo, configuración y circuito simulado en Proteus se aprecia en las siguientes figura 3 y 4.

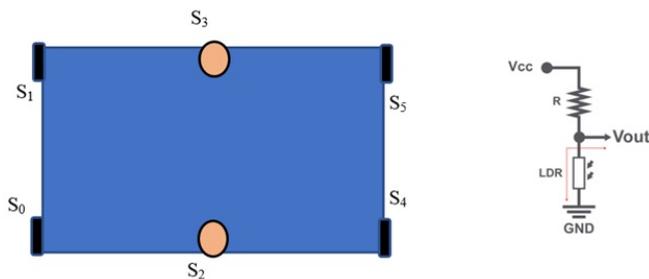


Figura 3: Módulo de entrada y punto de medición(Vout).

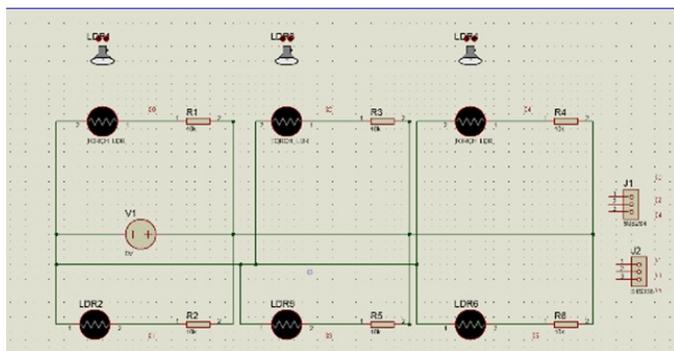


Figura 4: Circuito simulado en Proteus.

### 2.1.3. Módulo de proceso

El cual se puede dividir en la electrónica y la lógica de control.

### 2.1.4. Electrónica de control

Se propone el uso de una placa de desarrollo Arduino ya que cuenta con entradas/salidas analógicas, entradas/salidas digitales, puede ser programado fácilmente, así como puede interconectarse a una PC e interactuar con programas.

El modelo Arduino Mega2560 [2] se muestra en la siguiente figura 5.



Figura 5: Circuito simulado en Proteus.

### 2.1.5. Lógica difusa.

La lógica de control a utilizar es del tipo difusa, la cual es una rama de la inteligencia artificial que le permite a una computadora analizar información del mundo real entre una escala de lo falso y lo verdadero.

Dado que una computadora entiende el mundo en base a ceros y unos, trabaja con datos discretos que pueden tomar solo un valor dado y no aceptan ambigüedades, no entiende que es poco, medio, mucho, más o menos y otras sentencias que no arrojan un dato preciso, sino que depende de una valoración dada por alguien a quien se denomina experto [3].

Por ello se hace uso de la lógica difusa que permite darle un valor numérico dentro de un rango a tales sentencias ya que poco podría ser de 0 a 4, medio de 3 a 7 y mucho de 6 a 1 y se les conoce como variables lingüísticas. De esta manera si el dato analizar es 2 se puede decir que es poco, sin embargo, se puede dar el caso de que algunos valores no estén tan claros en que ponderación caben y es aquí donde se introduce el término del grado pertenencia a través de funciones que lo determinan. Estas funciones se denominan de pertenencia y se denotan por el símbolo  $\mu$

las cuales tienen un valor de 0 a 1 en el eje de las ordenadas y puede tomar valores en el eje de las abscisas que están determinados por los elementos totales a los cuales se hace referencia y lo cual se conoce como universo discurso.

Diseño del controlador difuso. – Para lograr el objetivo deseado a través lógica difusa se siguió el siguiente ciclo para su diseño:

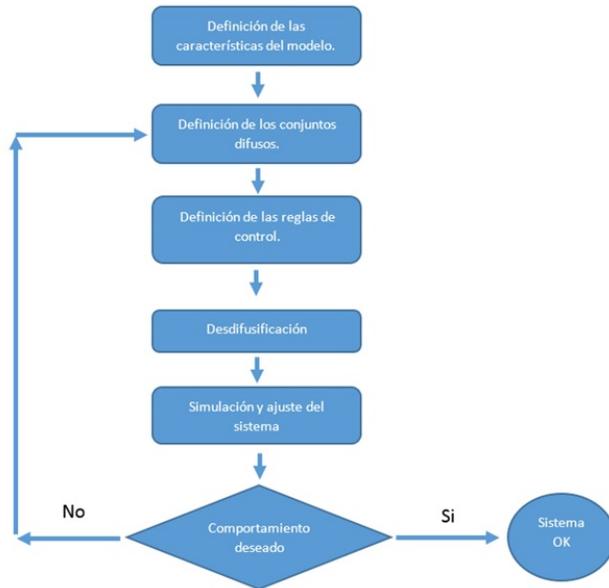


Figura 6: Ciclo de diseño del controlador difuso.

### 2.1.6. Definición de características del modelo

En esta etapa se definen las características funcionales y operacionales del modelo el cual es del tipo Mamdani, aquí la tarea del diseñador consiste en definir (aunque se carezca de un modelo matemático del sistema):

- Los datos de entrada al sistema.
- Las transformaciones básicas que se aplicarán a los datos.
- Los datos de salida del sistema.

### 2.1.7. Definición de conjuntos difusos

Para el proceso de definición de los conjuntos difusos en sistemas de control, primero se identifican, se nombran las variables de entrada, así como de salida y se establecen sus rangos,

### 2.1.8. Definición de las reglas

las reglas de control engloban el conocimiento del sistema y los objetivos de control. Cada regla tiene un estado del

sistema en su premisa y una acción de control sugerida en su conclusión.

Las reglas de control difusas conectan los valores de entrada con las propiedades de la salida del modelo. Están expresadas como proposiciones condicionales:

$$\text{SI}[\text{ESTADO DEL PROCESO}] \\ \text{ENTONCES}[\text{ACCIÓN DE CONTROL}]$$

Cuyo número de reglas se obtiene multiplicando el número de términos difusos de las variables de entrada. Y su base de reglas utiliza una matriz que cubre todas las posibles combinaciones de las entradas [4,5].

### 2.1.9. Desdifusificación

Es el proceso de transformar los valores difusos en “crisp” (que son aquellos que puede entender el computador). Se utilizan para ello métodos matemáticos simples como el del centroide ecuación (1), el cual calcula el centro de gravedad del polígono que se crea con el método de composición de membresía del máximo mínimo ecuación (2).

$$y = \frac{\sum_i b_i \int \mathbf{u}(\mathbf{i})}{\sum_i \int \mathbf{u}(\mathbf{i})} \quad (1)$$

$$\mu_{P \cdot Q} = \max \min [\mu_P, \mu_Q] \quad (2)$$

### 2.1.10. Simulación y ajuste del sistema

Cuando el modelo difuso ha sido construido, el proceso de simulación y desarrollo del modelo comienza. La cual puede realizarse en varios paquetes computacionales disponibles en el mercado.

Cuando los resultados de la simulación o las pruebas físicas no son satisfactorios se realizan ajustes en las descripciones de los conjuntos difusos o en las acciones de control sugeridas por las reglas, hasta afinar el desempeño del control.

### 2.1.11. Programación del controlador difuso

Se inicia identificando los siguientes elementos:

Entradas. – La variable que se mide es la intensidad luminosa que se realiza a través del arreglo de sensores ya descrito.

Universo discurso de la entrada. – El nivel de intensidad real va de 0

Funciones de membresía de la entrada. – El universo discurso se ha dividido en las siguientes funciones de membresía:

- 1.- Función trapezoidal minmax. – Su rango de valores es de  $[-0.2 -0.2 0 0.2]$ .
- 2.- Función triangular min. – Su rango de valores es de  $[0.1 0.3 0.5]$ .
- 3.- Función triangular medio. – Su rango de valores es de  $[0.3 0.5 0.7]$ .
- 4.- Función triangular max. – Su rango de valores es de  $[0.5 0.7 0.9]$ .
- 5.- Función trapezoidal maxmax. – Su rango de valores es de  $[0.8 1 1.2 1.2]$ .

Universo discurso de la salida. – Su rango de valores va de -1 a 1. Funciones de membresía de la salida. – El universo discurso se ha dividido en las siguientes funciones de membresía:

- 1.- Función trapezoidal maxverde. – Su rango de valores es de  $[-1 -1 -0.9 -0.7]$
- 2.- Función triangular altaverde. – Su rango de valores es de  $[-0.8 -0.6 -0.4]$
- 3.- Función triangular BAJAVERDE. – Su rango de valores es de  $[-0.5 -0.3 -0.1]$
- 4.- Función triangular velcero. – Su rango de valores es  $[-0.1902 0.008457 0.2085]$
- 5.- Función triangular BAJAAZUL. – Su rango de valores es  $[0.1 0.3 0.5]$
- 6.- Función triangular altaazul. – Su rango de valores es  $[-0.1902 0.008457 0.2085]$
- 7.- Función trapezoidal maxvelazul. – Su rango de valores es de  $[0.7 0.9 1 1]$

### 2.1.12. Variables lingüísticas

Las reglas si  $evento_1$  y  $evento_2$  y ...  $evento_n$  entonces acción, algunas utilizadas se enlistan a continuación

Si  $S_0 = \text{maxmax}$  y  $S_1 = \text{maxmax}$  y  $S_2 = \text{min}$  y  $S_3 = \text{min}$  y  $S_4 = \text{min}$  y  $S_5 = \text{min}$  entonces velmaxverde

Si  $S_0 = \text{max}$  y  $S_1 = \text{max}$  y  $S_2 = \text{max}$  y  $S_3 = \text{min}$  y  $S_4 = \text{min}$  y  $S_5 = \text{min}$  entonces BAJAVERDE

Si  $S_0 = \text{min}$  y  $S_1 = \text{min}$  y  $S_2 = \text{maxmax}$  y  $S_3 = \text{maxmax}$  y  $S_4 = \text{min}$  y  $S_5 = \text{min}$  entonces velcero

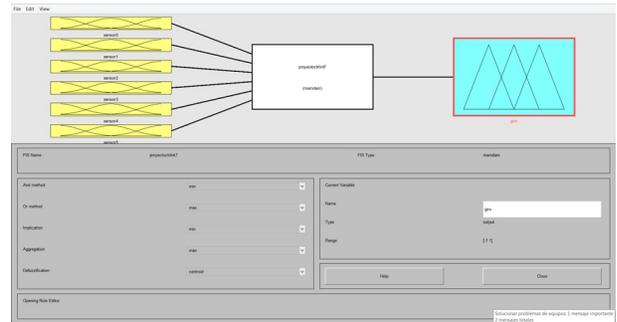


Figura 7: Ventana del toolbox de lógica difusa.

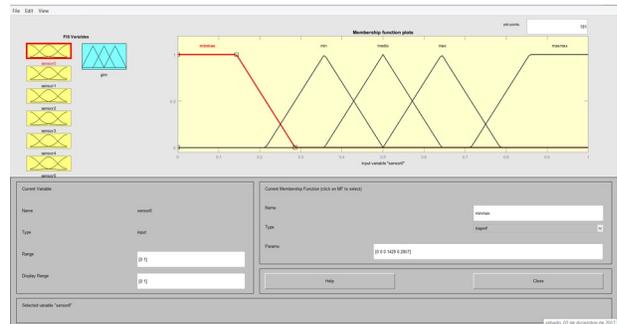


Figura 8: Funciones de membresía de las entradas.

Para su implementación se hizo uso del toolbox de lógica difusa [6] de MATLAB, ver figura 7 y 8.

En cada una de las entradas se agregaron las cinco funciones de membresía y sus rangos antes descritos:

En la salida se agregaron las siete funciones de membresía y sus rangos antes descritos, figura 9:

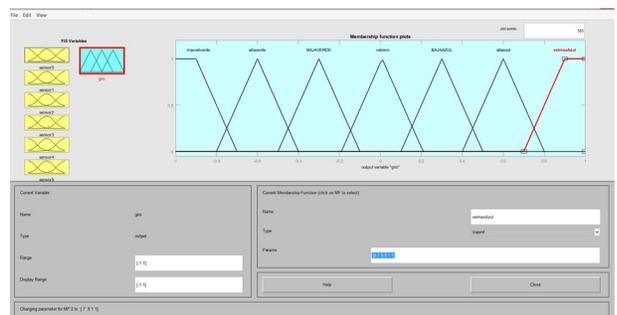


Figura 9: Funciones de membresía de la salida.

En el editor de reglas se agregaron 63 reglas como se aprecia en la figura 10.

La comprobación y validación de las reglas se realizaron en la opción visor de reglas, esto se muestra en la figura 11.

Si nos desplazamos hacia abajo se puede observar gráficamente la salida del centroide como se aprecia en la figura 12, que corresponde al ejemplo de la anterior.

También se puede observar gráficamente el

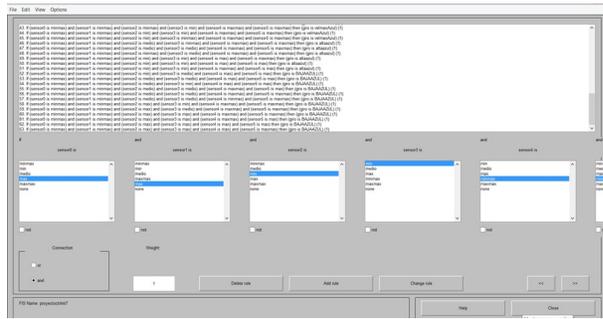


Figura 10: Editor de reglas.

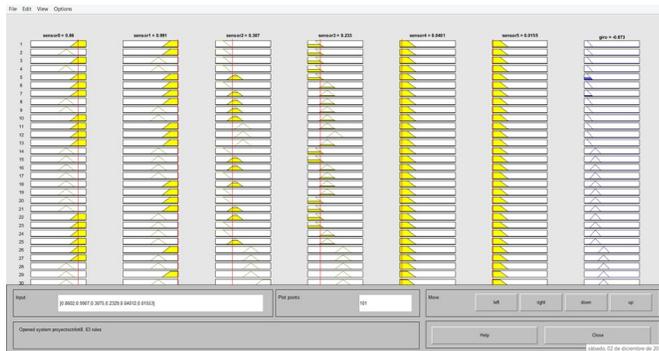


Figura 11: Vista del visor de reglas.

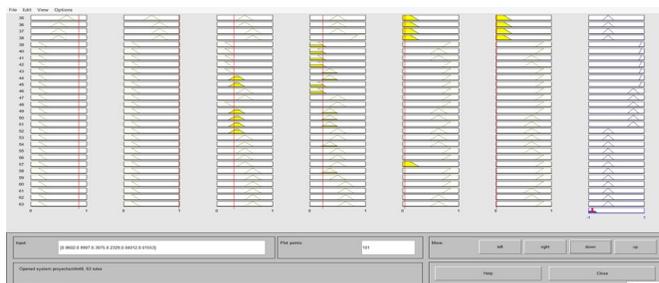


Figura 12: Vista del centroide en el visor de reglas.

comportamiento de las entradas contra la salida en la opción *Surface* como se muestra en la figura a continuación:

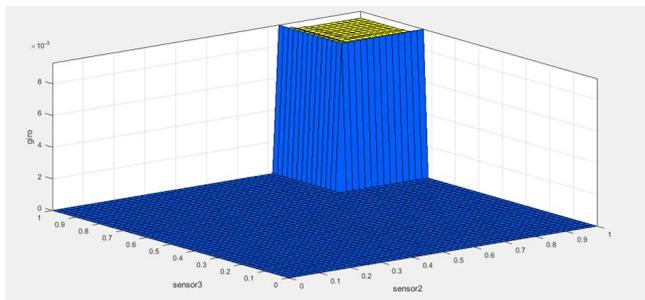


Figura 13: Vista grafica del comportamiento de las entradas contra la salida.

Para poder hacer uso del controlador diseñado se realizó en Simulink el siguiente modelo, ver figura 14:

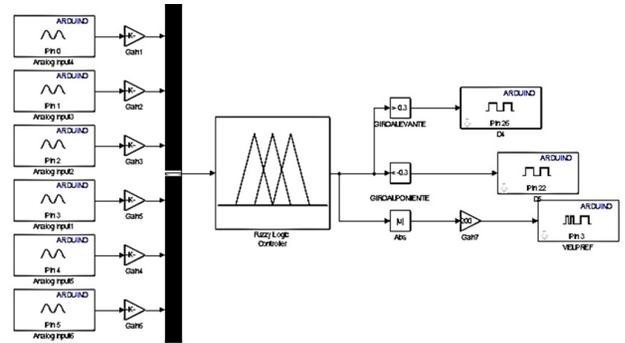


Figura 14: Modelo del control del seguidor en Simulink.

### 2.1.13. Módulo de electrónica de potencia

El actuador a controlar es del tipo motoreductor se necesita de un puente H para controlarlo por lo cual se utilizó el módulo LM298 [2] dada su facilidad de uso con la placa de desarrollo Arduino. El cual se muestra en la figura 15 con sus principales características

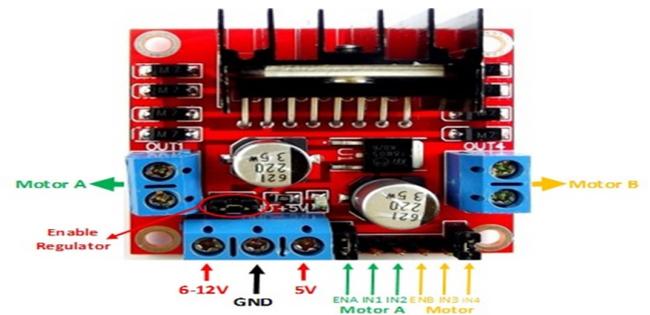


Figura 15: Módulo LM298 puente H.

### 2.1.14. Módulo de alimentación

Para este módulo se propone utilizar la fuente de alimentación del actuador que es de 12 VDC para el módulo del puente H el cual a su vez cuando el jumper "Enable Regulator" se encuentra activo, se tendrá un voltaje de 5 VDC en la clema marcada con esa tensión, de esta manera se podrá alimentar al módulo de electrónica de control y sensado.

### 2.1.15. Módulo mecánico

En este módulo se incluye los mecanismos utilizados, así como las estructuras usadas. Los cuales serían la carcasa protectora de los módulos y sus soportes al seguidor [7].

Para resguardar el módulo de sensado se diseñó en SolidWorks una caja que se puede apreciar en la figura 16:

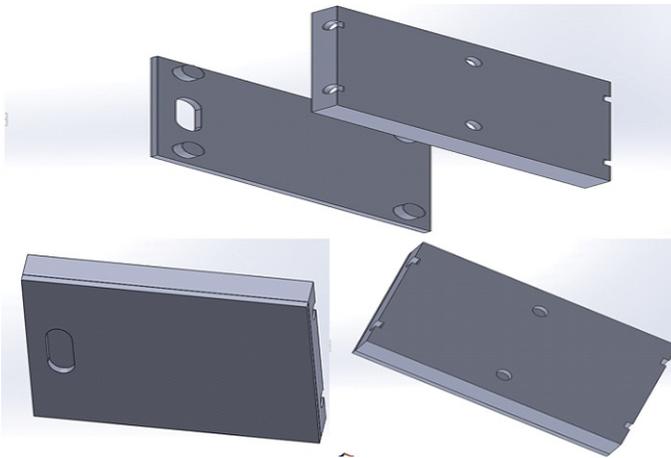


Figura 16: Vistas de la caja del módulo de entrada, en la parte inferior y vista explosionada de la caja del módulo de entrada en la parte superior.

Quedando pendiente el gabinete que alojara a los módulos de control y potencia.

### 2.1.16. Interconexión de módulos

Para continuar con la validación de las reglas si... entonces... se procedió a ensamblar los módulos como se aprecia a continuación, en la figura 17.

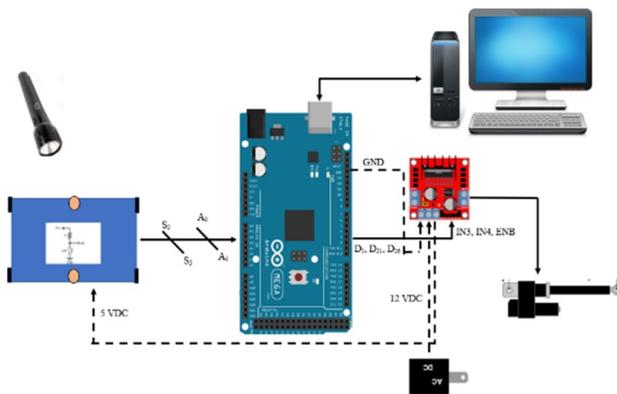


Figura 17: Interconexión de los módulos.

## 3. Pruebas y resultados del modelo

Para realizar pruebas, y poder validar el modelo presentado, se realizó un prototipo del sistema el cual se muestra en la figura 18.

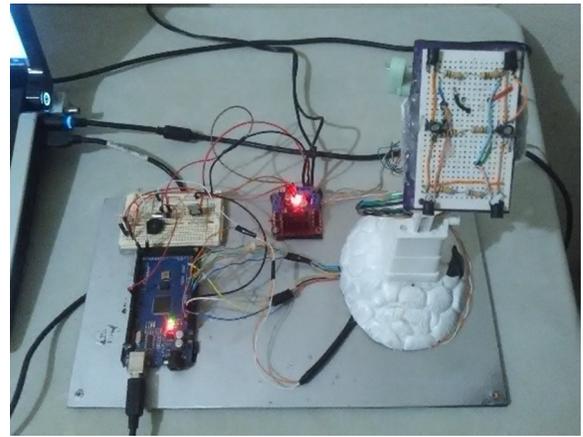


Figura 18: Modelo para pruebas.

Durante la experimentación al incidir luz en el módulo de sensado e ir observando el vector de valores de entrada contra la salida en algunos casos no realizo lo esperado por que hubo la necesidad de un ajuste de las reglas si... entonces... al incluir y borrar algunas.

Se comprobó que es necesario realizar un nuevo ajuste de las reglas si... entonces... en el sitio donde quedara instalado el dispositivo para mejorar su eficiencia.

En las siguientes graficas se observa el comportamiento de los sensores, la salida del controlador difuso y la velocidad del actuador que se va reduciendo conforme se acerca el dispositivo a la posición de noventa grados con respecto al Sol, ver figura 19.

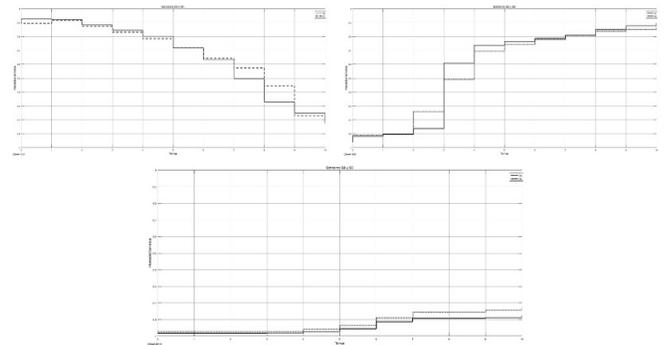


Figura 19: Comportamiento de los sensores al incidir la luz en S0/ S1, S2/ S3 y S4/ S5.

Como se puede constatar en las gráficas el controlador se moverá desde un punto inicial en alguno de sus dos extremos y se moverá variando su velocidad de máximo, la cual se irá reduciendo hasta que los sensores que tiene en medio alcancen un valor máximo entonces será de cero.

Un punto importante a resaltar es que cuando se utiliza el toolbox fuzzy junto con simulink de MATLAB y la placa de desarrollo que se utilizara para programar y/o

usarse como tarjeta de adquisición de datos, el código que se genera puede sobrepasar la capacidad de memoria del dispositivo. Se recomendaría que este no sobrepasara el 80 %, ver figura 20 y 21.

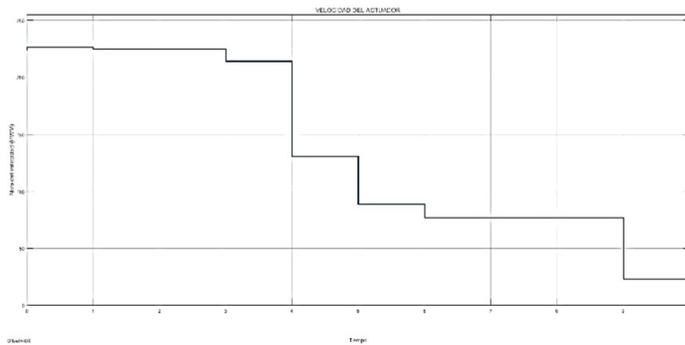


Figura 20: Comportamiento de la velocidad del actuador.

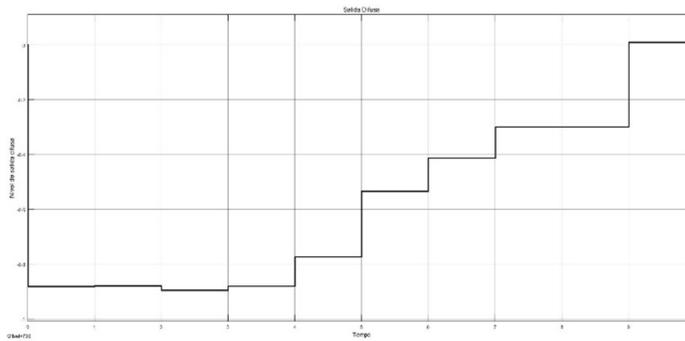


Figura 21: Comportamiento de la salida del actuador.

## 4. Conclusiones

El controlador difuso logró exitosamente seguir la trayectoria de la fuente luminosa colocando la base sobre actuador en un ángulo cercano a 90° con respecto a la incidencia de los rayos de luz.

La experiencia del experto expresada en las reglas “si...entonces...” es vital para la eficiencia del controlador difuso.

La velocidad del actuador es un punto importante ya que debe ajustarse para que no oscile el dispositivo, debido a que no se estabiliza la medición si es demasiado rápida.

El toolbox fuzzy en conjunto con Simulink de MATLAB son herramientas bastante simples que permiten implementar controladores difusos autónomos con relativa facilidad.

Se debe de tomar en consideración las capacidades del hardware a utilizar, principalmente el que se va a

programar ya que el uso del toolbox fuzzy consume muchos recursos de memoria SRAM y FLASH.

## Referencias

- [1] Robert L Boylestad y Louis Nashelsky. *Electrónica: teoría de circuitos y dispositivos electrónicos*. Pearson educación, 2003.
- [2] J Pomares. Manual de arduino. *Grupo de Innovación Educativa en Automática. Universidad de Alicante*, 2009.
- [3] Ebrahim H Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. In *Proceedings of the institution of electrical engineers*, volume 121, pages 1585–1588. IET, 1974.
- [4] Petrus J Braspenning y Frank Thuijsman. *Artificial neural networks: an introduction to ANN theory and practice*, volume 931. Springer Science & Business Media, 1995.
- [5] Maurizio Bielli, Giorgio Ambrosino, y Marco Boero. *Artificial intelligence applications to traffic engineering*. Vsp, 1994.
- [6] Amos Gilat. *MATLAB: An introduction with applications*. Number Sirsi) i9780470108772. Wiley New York:, 2008.
- [7] S González Gómez y España El gran libro de Solidworks. Marcombo, 2014.

# Prototipo de hardware y software de un CPU didáctico de 8 bits

J. L. Beltrán-Márquez<sup>1,\*</sup>, C.L. Zazueta-Villavicencio<sup>1</sup>, A. C. Ruiz-Calvillo<sup>1</sup>

<sup>1</sup>Ingeniería en sistemas computacionales, Instituto Tecnológico Superior de Cajeme, Sonora, México

\*e-mail: lbeltran@itesca.edu.mx

**Resumen:** *El presente trabajo explica la transición que se hizo en ITESCA para pasar de la modalidad teórica de la asignatura Arquitectura de Computadoras a su impartición en forma práctica; para lograr lo anterior hubo que aprovechar los avances en computación en cuanto a simuladores y lenguajes de programación. Primeramente se tuvo que diseñar un modelo simple de un procesador de 8 bits, y el set de instrucciones necesarios, así como la técnica para automatizar dichos procesos. Se hace uso del simulador de sistemas digitales (logisim) para crear el diseño digital del ALU y de la unidad de control del CPU. Por último, un alumno de la institución desarrolló en sus residencias una aplicación para lograr unir el modelo del CPU con logisim y tener entonces una interfaz gráfica que permita al estudiante programar en ensamblador su diseño de CPU y probar su funcionalidad.*

**Palabras clave:** CPU, ITESCA, Logisim, Arquitectura de Computadora.

**Abstract:** *This paper explains the transition that was made in ITESCA to move from the theoretical modality of the Computer Architecture subject to its practical delivery; to achieve the above, it was necessary to take advantage of advances in computing in terms of simulators and programming languages. First, it was necessary to design a simple model of an 8-bit processor, and the set of instructions needed, as well as the technique to automate these processes. The digital systems simulator (logisim) is used to create the digital design of the ALU and the CPU control unit. Finally, a student of the institution developed in their residences an application to achieve unite the CPU model with logisim and then have a graphical interface that allows the student to assemble their CPU design and test its functionality.*

**Keywords:** CPU, ITESCA, Logisim, Computer Architecture.

## 1. Antecedentes

La Arquitectura de Computadoras son un conjunto de reglas y métodos que describen la funcionalidad, organización e implementación de sistemas computacionales; en otras palabras, La Arquitectura de Computadoras describe la organización interna de un computadora de manera abstracta, es decir, define las capacidades de una computadora y su modelo de programación. [1, 2]

En la carrera de Ingeniería en Sistemas Computacionales de ITESCA se imparte la materia de Arquitectura de Computadoras la cual considera el siguiente contenido:

- Arquitectura de los componentes del CPU (ALU, Unidad de Control, Registros y Buses Internos)
- Memoria
- Ciclos de Instrucción

## 2. Problemática

El enfoque propuesto por el TecNM para abordar el contenido de la asignatura consiste en: clasificar, identificar, buscar y seleccionar información, y por último exponer; es decir, abordar el contenido de la materia con un enfoque teórico.

El enfoque modificado de ITESCA es que el alumno descubra la experiencia de diseñar una computadora, repitiendo el diseño de los desarrolladores originales, usando sus conocimientos previos de electrónica y sistemas digitales; sin embargo, hacerlo físicamente sería prácticamente imposible, por lo que usamos un software de distribución libre, para simular el diseño del CPU, en este caso Logisim [3-6].

Es necesario también contar con un prototipo simple que sirva de guía para que el alumno se apegue a ese modelo y pueda construir el suyo en el transcurso del semestre, por lo que además es necesario tener un prototipo funcional y simple.

Por último es necesario verificar si el prototipo funcionó, para lo cual hay que contar con un software que lo simule; sin embargo, no encontramos software disponible, por lo que fue necesario diseñar y crear un programa para simular en lenguaje ensamblador el set de instrucciones con que se construyó el prototipo.

### 3. Aspectos Teóricos

Se tiene que repasar la técnica con la cual los alumnos hacen las operaciones aritméticas y lógicas en el sistema binario, debido a que se tienen que automatizar. [7]

Enseguida se muestran como ejemplo, por medio de tablas la operación de suma y resta:

#### 3.1. Suma

La tabla 1 muestra la operación de suma considerando el acarreo previo, mostrado en la figura 1:

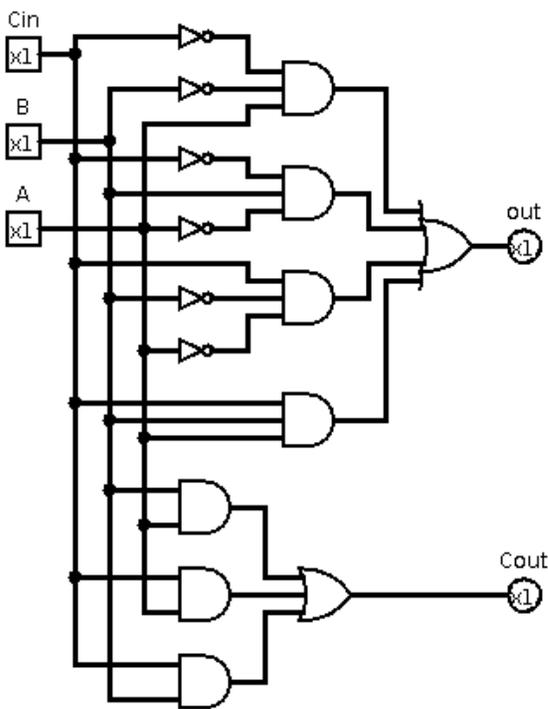


Figura 1: Circuito que suma una columna

misma que genera las siguientes ecuaciones:

$$\text{Suma: } \overline{\text{CarryIn}} \cdot \overline{B} \cdot A + \overline{\text{CarryIn}} \cdot B \cdot \overline{A} + \text{CarryIn} \cdot \overline{B} \cdot \overline{A} + \text{CarryIn} \cdot B \cdot A$$

$$\text{CarryOut: } B \cdot A + \text{CarryIn} \cdot A + \text{CarryIn} \cdot B$$

Tabla 1: Operación de suma

CarryIn	A	B	CarryOut	Suma
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

#### 3.2. Resta

Al igual que en la operación de suma, la tabla 2 que muestra la resta A-B considerando el préstamo previo mostrado en la figura 2:

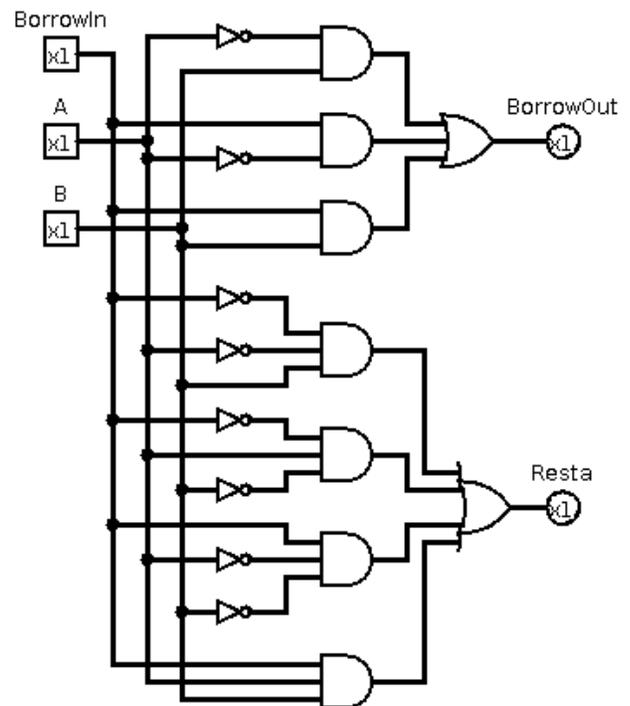


Figura 2: Circuito que Resta una columna

Que nos da las siguientes ecuaciones:

$$\text{Resta: } \overline{\text{BorrowIn}} \cdot \overline{A} \cdot B + \overline{\text{BorrowIn}} \cdot A \cdot \overline{B} + \text{BorrowIn} \cdot \overline{A} \cdot \overline{B} + \text{BorrowIn} \cdot A \cdot B$$

$$\text{BorrowOut: } \overline{A} \cdot B + \text{BorrowIn} \cdot \overline{A} + \text{BorrowIn} \cdot B$$

Tabla 2: Resta

Borrow in	A	B	Borrow out	Resta
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

## 4. Prototipo en Hardware

El prototipo en hardware es un cpu didáctico de 8 bits que en realidad solo tiene las funciones más elementales de un procesador, ya que si se seleccionan más instrucciones sería muy difícil terminar el prototipo en un semestre.

Primeramente se diseña un ALU capaz de realizar las instrucciones aritméticas y lógicas, enseguida el CPU que usa al ALU, pero que además agrega la funcionalidad de las instrucciones de transferencia, de control de flujo y las que manejan la pila y llamado de funciones. Por último se muestra un resumen de las microinstrucciones que en realidad son los pasos que tiene que hacer el CPU para completar dichas instrucciones.

### 4.1. (ALU) Unidad Aritmética y Lógica

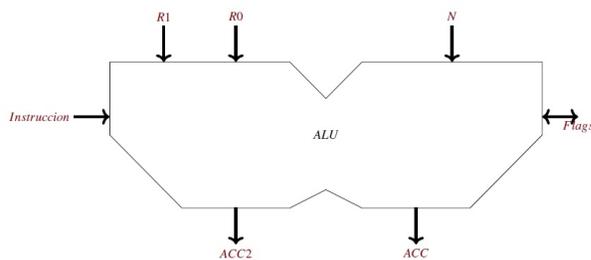


Figura 3: Unidad Aritmética y Lógica

#### 4.1.1. Entradas y Salidas

Las entradas y salidas a considerar son las siguientes,

**N** Entrada de 8 bits

**R0** Entrada de 8 bits

**R1** Entrada de 8 bits

**ACC** Primer Salida de 8 bits

**ACC2** Segunda Salida de 8 bits

**Número de Instrucción** Instrucción a ejecutar

**Banderas (Flags)** Estado lógico de la última operación realizada

**Overflow**

**Carry**

**Sign**

**Zero**

y su posterior operación se muestra en las tablas 3 y 4.

Tabla 3: Primera parte

Número	Nom bre	Sinta xis	Operación	O	S	Z	C
00	XCHG	XCHG	$ACC \leftarrow R1, ACC2 \leftarrow R0$				
01	CBW	CBW	$(ACC2 : ACC) \leftarrow N$		±	±	
02	ADD	ADD N	$ACC \leftarrow R0 + N$	±	±	±	±
03	ADC	ADC N	$ACC \leftarrow R0 + N + C$		±	±	±
04	SUB	SUB N	$ACC \leftarrow R0 - N$	±	±	±	±
05	SBB	SBB N	$ACC \leftarrow R0 - N - C$	±	±	±	±
06	IDIV	IDIV N	$(ACC = Cociente, ACC2 = Residuo) \leftarrow \frac{R1:R0}{N}$			±	±
07	IMUL	IMUL N	$(ACC2 : ACC) \leftarrow R0 \cdot N$		±	±	
08	INC	INC N	$ACC \leftarrow N + 1$	±	±	±	±
09	DEC	DEC N	$ACC \leftarrow N - 1$	±	±	±	±
0A	NEG	NEG N	$ACC \leftarrow -N$		±	±	
0B	SAL	SAL N	$ACC \leftarrow sal(N)$		±	±	±

Tabla 4: Segunda parte

Número	Nombre	Sintaxis	Operación	O	S	Z	C
0C	SAR	SAR N	$ACC \leftarrow sar(N)$		±	±	±
0D	NOT	NOT N	$ACC \leftarrow \sim N$		±	±	
0E	AND	AND N	$ACC \leftarrow R0 \text{ and } N$		±	±	
0F	OR	OR N	$ACC \leftarrow R0 \text{ or } N$		±	±	
10	XOR	XOR N	$ACC \leftarrow R0 \text{ xor } N$		±	±	
11	SHL	SHL N	$ACC \leftarrow shl(N)$		±	±	±
12	SHR	SHR N	$ACC \leftarrow shr(N)$		±	±	±
13	RCL	RCL N	$ACC \leftarrow rcl(N)$		±	±	±
14	RCR	RCR N	$ACC \leftarrow rcr(N)$		±	±	±
15	ROL	ROL N	$ACC \leftarrow rol(N)$		±	±	±
16	ROR	ROR N	$ACC \leftarrow ror(N)$		±	±	±

#### 4.1.2. Valores de las banderas para las operaciones de suma y resta

Tabla 5: Tabla que muestra los valores de las banderas para la suma

$t = a + b$	O	S	Z	C
Si hay carry				1
$t == 0$			1	
$t < 0$		1		
$[(a > 0) \text{ and } (b > 0) \text{ and } (t < 0)] \text{ or } [(a < 0) \text{ and } (b < 0) \text{ and } (t \geq 0)]$	1			

Tabla 6: Tabla que muestra los valores de las banderas para la resta o la comparación

$t = a - b$	O	S	Z	C
Si hay borrow				1
$t == 0$			1	
$t < 0$		1		
$[(a > 0) \text{ and } (b < 0) \text{ and } (t < 0)] \text{ or } [(a < 0) \text{ and } (b > 0) \text{ and } (t > 0)]$	1			

## 4.2. CPU Didáctico Simple de 8 bits

### 4.2.1. Registros usados en el procesador simple

**MAR** Memory Address Register

**MBR** Memory Buffer Register

**PC** Program Counter

**IR** Instruction Register

**DS** Data Segment

**SP** Stack Pointer

**R0** Registro 0

**R1** Registro 1

**Flags (COSZ)** Banderas Carry, Overflow, Sign, Zero

### 4.2.2. Instrucciones de Transferencia

En la tabla 7 se muestran las instrucciones de transferencia utilizadas en el proyecto,

Tabla 7: Instrucciones de Transferencia

Código	Tipo	Nombre	Sintaxis	Operación	O	S	Z	C
00	A	STOP	STOP	Alto				
01	B	MOV	MOV [N],R0	$RAM[N] \leftarrow R0$				
02	C	MOV	MOV R0,N	$R0 \leftarrow N$				
03	D	MOV	MOV R0,[N]	$R0 \leftarrow RAM[N]$				
04	E	MOV	MOV R0,[DS]	$R0 \leftarrow RAM[DS]$				
05	F	MOV	MOV R0,R1	$R0 \leftarrow R1$				
06	G	MOV	MOV R1,R0	$R1 \leftarrow R0$				
07	H	MOV	MOV DS,R0	$DS \leftarrow R0$				

### 4.2.3. Instrucciones Aritméticas

A continuación mediante el uso de la tabla 8, se ilustra de forma detallada las instrucciones aritméticas utilizadas en el desarrollo del proyecto,

Tabla 8: Instrucciones Aritméticas

Código	Tipo	Nombre	Sintaxis	Operación	O	S	Z	C
08	I	XCHG	XCHG	$R0 \leftarrow R1, R1 \leftarrow R0$				
09	I	CBW	CBW	$R1 : R0 \leftarrow R0$				
0A	J	ADD	ADD R0,N	$R0 \leftarrow R0 + N$	±	±	±	±
0B	K	ADD	ADD R0,[N]	$R0 \leftarrow R0 + RAM[N]$	±	±	±	±
0C	J	ADC	ADC R0,N	$R0 \leftarrow R0 + N + C$	±	±	±	±
0D	K	ADC	ADC R0,[N]	$R0 \leftarrow R0 + RAM[N] + C$	±	±	±	±
0E	J	SUB	SUB R0,N	$R0 \leftarrow R0 - N$	±	±	±	±
0F	K	SUB	SUB R0,[N]	$R0 \leftarrow R0 - RAM[N]$	±	±	±	±
10	J	SBB	SBB R0,N	$R0 \leftarrow R0 - N - C$	±	±	±	±
11	K	SBB	SBB R0,[N]	$R0 \leftarrow R0 - RAM[N] - C$	±	±	±	±
12	L	IDIV	IDIV N	$(R0 = Cociente, R1 = Residuo) \leftarrow \frac{R1:R0}{N}$			±	
13	M	IDIV	IDIV [N]	$(R0 = Cociente, R1 = Residuo) \leftarrow \frac{R1:R0}{[N]}$			±	
14	L	IMUL	IMUL N	$(R1 : R0) \leftarrow R0 \cdot N$			±	
15	M	IMUL	IMUL [N]	$(R1 : R0) \leftarrow R0 \cdot [N]$			±	
16	N	INC	INC R0	$R0 \leftarrow R0 + 1$	±	±	±	±
17	O	INC	INC DS	$DS \leftarrow DS + 1$	±	±	±	±
18	P	INC	INC [N]	$RAM[N] \leftarrow RAM[N] + 1$	±	±	±	±
19	N	DEC	DEC R0	$R0 \leftarrow R0 - 1$	±	±	±	±
1A	O	DEC	DEC DS	$DS \leftarrow DS - 1$	±	±	±	±
1B	P	DEC	DEC [N]	$RAM[N] \leftarrow RAM[N] - 1$	±	±	±	±
1C	N	NEG	NEG R0	$R0 \leftarrow -R0$		±	±	
1D	P	NEG	NEG [N]	$RAM[N] \leftarrow -RAM[N]$		±	±	
1E	N	SAL	SAL R0	$R0 \leftarrow sal(R0)$		±	±	±
1F	P	SAL	SAL [N]	$RAM[N] \leftarrow sal(RAM[N])$		±	±	±
20	N	SAR	SAR R0	$R0 \leftarrow sar(R0)$		±	±	±
21	P	SAR	SAR [N]	$RAM[N] \leftarrow sar(RAM[N])$		±	±	±

#### 4.2.4. Instrucciones Lógicas

De igual forma se presenta una serie de tablas 9, 10 y 11 donde se muestran las instrucciones lógicas que son empleadas en el proyecto.

Tabla 9: Instrucciones Lógicas, parte 1

Código	Tipo	Nombre	Sintaxis	Operación	O	S	Z	C
22	N	NOT	NOT R0	$R0 \leftarrow \sim R0$		±	±	
23	P	NOT	NOT [N]	$RAM[N] \leftarrow \sim RAM[N]$		±	±	
24	J	AND	AND R0 and N	$R0 \leftarrow R0 \text{ and } N$		±	±	

Tabla 10: Instrucciones Lógicas, parte 2

Código	Tipo	Nombre	Sintaxis	Operación	O	S	Z	C
25	K	AND	AND [N]	$R0 \leftarrow R0 \text{ and } RAM[N]$		±	±	
26	J	OR	OR R0 or N	$R0 \leftarrow R0 \text{ or } N$		±	±	
27	K	OR	OR [N]	$R0 \leftarrow R0 \text{ or } RAM[N]$		±	±	
28	J	XOR	XOR R0 xor N	$R0 \leftarrow R0 \text{ xor } N$		±	±	
29	K	XOR	XOR [N]	$R0 \leftarrow R0 \text{ xor } RAM[N]$		±	±	

Tabla 11: Instrucciones Lógicas, parte 3

Código	Tipo	Nombre	Sintaxis	Operación	O	S	Z	C
2A	N	SHL	SHL R0	$R0 \leftarrow shl(R0)$		±	±	±
2B	P	SHL	SHL [N]	$RAM[N] \leftarrow shl(RAM[N])$		±	±	±
2C	N	SHR	SHR R0	$R0 \leftarrow shr(R0)$		±	±	±
2D	P	SHR	SHR [N]	$RAM[N] \leftarrow shr(RAM[N])$		±	±	±
2E	N	RCL	RCL R0	$R0 \leftarrow rcl(R0)$		±	±	±
2F	P	RCL	RCL [N]	$RAM[N] \leftarrow rcl(RAM[N])$		±	±	±
30	N	RCR	RCR R0	$R0 \leftarrow rcr(R0)$		±	±	±
31	P	RCR	RCR [N]	$RAM[N] \leftarrow rcr(RAM[N])$		±	±	±
32	N	ROL	ROL R0	$R0 \leftarrow rol(R0)$		±	±	±
33	P	ROL	ROL [N]	$RAM[N] \leftarrow rol(RAM[N])$		±	±	±
34	N	ROR	ROR R0	$R0 \leftarrow ror(R0)$		±	±	±
35	P	ROR	ROR [N]	$RAM[N] \leftarrow ror(RAM[N])$		±	±	±

4.2.5. Instrucciones de Control de Flujo

En la tabla 12, 13 y 14, se muestran de forma detallada las instrucciones del control de flujo.

Tabla 12: Instrucciones de Control de Flujo

Código	Tipo	Nombre	Sintaxis	Operación	O	S	Z	C
36	Q	CMP	CMP N	Compara R0 con N $a= R0, b= N$	±	±	±	±

Tabla 13: Instrucciones de Control de Flujo

Código	Tipo	Nombre	Sintaxis	Operación	O	S	Z	C
37	R	CMP	CMP [N]	Compara R0 con RAM[N] $a= R0, b= RAM[N]$	±	±	±	±
38	S	CMP	CMP [DS]	Compara R0 con RAM[DS] $a= R0, b= RAM[DS]$	±	±	±	±
39	X	JE	JE N	Si $(a = b)$ salta a la dirección N las banderas consideradas son: $(Z)$				
3A	X	JNE	JNE N	Si $(a \neq b)$ salta a la dirección N las banderas consideradas son: $(\sim Z)$				
3B	X	JA	JA N	Si $(a > b)$ salta a la dirección N las banderas consideradas son: $(\sim C \& \sim Z)$ (los números se consideran sin signo)				
3C	X	JAЕ	JAЕ N	Si $(a \geq b)$ salta a la dirección N las banderas consideradas son: $(\sim C)$ (los números se consideran sin signo)				
3D	X	JB	JB N	Si $(a < b)$ salta a la dirección N las banderas consideradas son: $(C)$ (los números se consideran sin signo)				

En la tabla 15 se muestran las condiciones de salto.

Tabla 14: Instrucciones de Control de Flujo

Código	Tipo	NOMBRE	Sintaxis	Operación	O	S	Z	C
3E	X	JBE	JBE N	Si $(a \leq b)$ salta a la dirección N las banderas consideradas son: $(C   Z)$ (los números se consideran sin signo)				
3F	X	JG	JG N	Si $(a > b)$ salta a la dirección N las banderas consideradas son: $(\sim(S^O) \& \sim Z)$ (los números se consideran con signo)				
40	X	JGE	JGE N	Si $(a \geq b)$ salta a la dirección N las banderas consideradas son: $(\sim(S^O))$ (los números se consideran con signo)				
41	X	JL	JL N	Si $(a < b)$ salta a la dirección N las banderas consideradas son: $(S^O)$ (los números se consideran con signo)				
42	X	JLE	JLE N	Si $(a \leq b)$ salta a la dirección N las banderas consideradas son: $((S^O)   Z)$ (los números se consideran con signo)				
43	X	JMP	JMP N	salta a la dirección N				

Tabla 15: Condiciones de saltos utilizados.

Condición	Mnemonic	Símbolo	Condición de banderas	Descripción
salto	jmp			salto incondicional
$a = b$	e	z	Z	Equal / Zero
$a \neq b$	ne	nz	$\sim Z$	Not Equal / Not Zero
$a > b$	a	nbe	$\sim C \& \sim Z$	Above (unsigned)
$a \geq b$	ae	nb	$\sim C$	Above ó Equal (unsigned)
$a < b$	b	nae	C	Below (unsigned)
$a \leq b$	be	na	$C   Z$	Below ó Equal (unsigned)
$a > b$	g	nle	$\sim(S^O) \& \sim Z$	Greater (signed)
$a \geq b$	ge	nl	$\sim(S^O)$	Greater ó Equal (signed)
$a < b$	l	nge	$S^O$	Less (signed)
$a \leq b$	le	ng	$(S^O)   Z$	Less ó Equal (signed)

#### 4.2.6. Instrucciones de Control de Funciones, y la Pila

En la tabla 16 se presentan las instrucciones del control de funciones.

Tabla 16: Instrucciones de Control de Funciones

Código	Tipo	NOMBRE	Sintaxis	Operación
44	T	CALL	CALL N	Metete a la pila el valor del Registro PC y salta a la dirección N
45	U	RET	RET	Saca el último valor de la pila y salta a esa dirección
46	V	PUSH	PUSH	Metete a la pila el valor de R0
47	W	POP	POP	Saca el último valor de la pila y lo metete a R0

Tabla 17: Micro-Instrucciones en el Procesador Simple de 8 bits, Micro Instrucciones A-H, I-P y Q-X

#	A	B	C	D	E	F	G	H
0	MAR<PC							
1	MBR<RAM							
2	IR<MBR							
3	FIN	PC+=2	PC+=2	PC+=2	PC++	PC++	PC++	PC++
4		MAR++	MAR++	MAR++	MAR++	R0<R1	R1<R0	DS<R0
5		MBR<RAM	MBR<RAM	MBR<RAM	MAR<DS	FIN	FIN	FIN
6		MAR<MBR	R0<MBR	MAR<MBR	MBR<RAM			
7		MBR<R0	FIN	MBR<RAM	R0<MBR			
8		RAM<MBR		R0<MBR	FIN			
9		FIN		FIN				
#	I	J	K	L	M	N	O	P
0	MAR<PC							
1	MBR<RAM							
2	IR<MBR							
3	PC++	PC+=2	PC+=2	PC++	PC+=2	PC++	PC++	PC+=2
4	ALU	MAR++	MAR++	MAR++	MAR++	ALU	ALU<DS	MAR++
5	R0<ACC	MBR<RAM	MBR<RAM	MBR<RAM	MBR<RAM	R0<ACC	DS<ACC	MBR<RAM
6	R1<ACC2	ALU<MBR	MAR<MBR	ALU<MBR	MAR<MBR	FIN	FIN	MAR<MBR
7	FIN	R0<ACC	MBR<RAM	R0<ACC	MBR<RAM			MBR<RAM
8		FIN	ALU<MBR	R1<ACC2	ALU<MBR			ALU<MBR
9			R0<ACC	FIN	R0<ACC			MBR<ACC
10			FIN		R1<ACC2			RAM<MBR
11					FIN			FIN
#	Q	R	S	T	U	V	W	X
0	MAR<PC							
1	MBR<RAM							
2	IR<MBR							
3	PC+=2	PC+=2	PC++	PC+=2	SP++	PC++	PC++	PC+=2
4	MAR++	MAR++	ALU<DS	MAR<SP	MAR<SP	MAR<SP	SP++	MAR++
5	MBR<RAM	MBR<RAM	FIN	MBR<PC	MBR<RAM	MBR<R0	MAR<SP	MBR<RAM
6	ALU<MBR	MAR<MBR		RAM<MBR	PC<MBR	RAM<MBR	MBR<RAM	Salto<MBR
7	FIN	MBR<RAM		SP- -	FIN	SP- -	R0<MBR	PC<Salto
8		ALU<MBR		MAR<PC		FIN	FIN	FIN
9		FIN		MAR- -				
10				MBR<RAM				
11				PC<MBR				
12				FIN				
13								
14								
15								

## 5. Prototipo en Software

El siguiente emulador usado en la materia de Arquitectura de Computadoras:

<https://emucompu8.mjroman.com/>

es utilizado como herramienta del alumno, en el diseño de un CPU didáctico de 8 bits.

El emulador está basado en el diseño de un mini CPU simplificado al máximo, para alcanzar a ver su funcionamiento completo en el transcurso de una materia con duración de un semestre.

Para el diseño del CPU, primeramente se diseñó un ALU igualmente de 8 bits.

Las tablas que resumen el funcionamiento del ALU y del CPU, así como ejemplos de los mismos se incluyen en este documento.

Dichas tablas fueron proporcionadas al alumno: *Manuel de Jesús Román*, quién desarrolló una aplicación web basada en java, que emula el funcionamiento del miniCPU de 8 bits, en sus Residencias Profesionales en ITESCA.

### 5.1. Desarrollo del Intérprete de código ensamblador

Con los elementos visuales acomodados en el lugar que debían estar, se procedió con el desarrollo del módulo que interpretaría el código introducido en el editor de texto; se utilizaron expresiones regulares para descomponer el texto presente en el editor y encontrar las partes que se emparejarán con las instrucciones de código ensamblador. Una vez que el módulo reconoció todas las instrucciones del set establecido se procedió a desarrollar el sub módulo

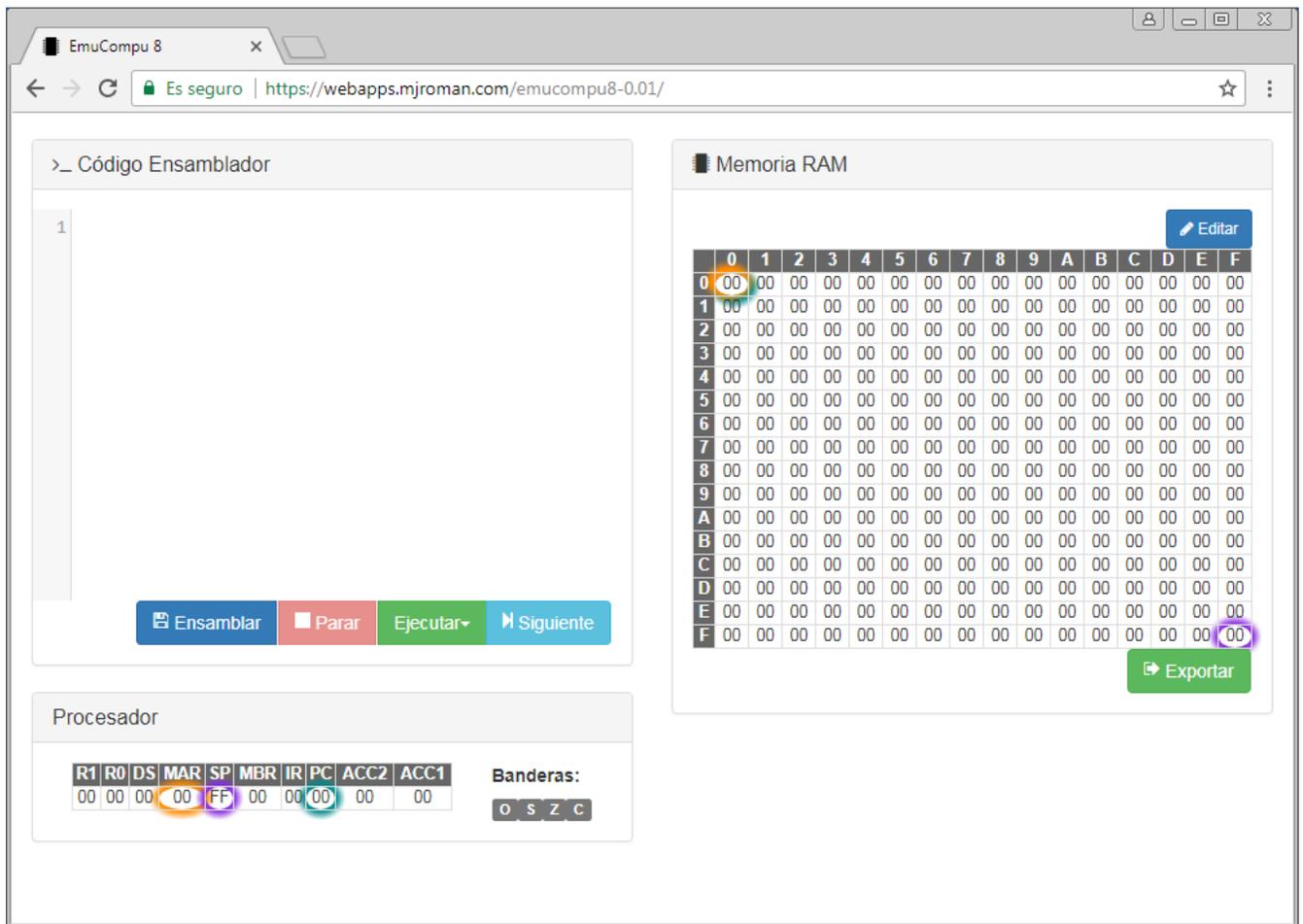


Figura 4: Vista de la aplicación

que las convertiría a código máquina y dejar esta salida lista para la siguiente etapa. Una vez terminado el módulo se realiza la conexión con su correspondiente elemento gráfico: el editor de texto y el botón que generará la salida de código máquina. [8]

## 5.2. Interfaz Gráfica

El sistema cuenta con una sola interfaz donde se muestran todos los componentes necesarios para el funcionamiento del mismo. Es un sistema web responsivo, para que todas las animaciones se aprecien el sistema ha de ejecutarse en una resolución horizontal mínima de 992 píxeles. Como se puede apreciar en la figura 4, todos los componentes de la aplicación son perfectamente visibles con la resolución horizontal mínima recomendada.

El intérprete de código ensamblador se activa al dar clic en el botón “Ensamblar”; analiza el código introducido por el usuario, reconoce instrucciones y argumentos, ignora los comentarios, reconoce etiquetas, marca errores donde los encuentra y, si no los hubiera, traduce las

instrucciones reconocidas en código máquina y las escribe en memoria RAM. [8]

## 6. Conclusiones

Ofrecer la asignatura de arquitectura de computadoras con un enfoque práctico tiene el beneficio que el alumno crea su propia versión de una computadora, en lugar de solo estudiarla; lo que le permite construir nuevos conocimientos de manera más sólida.

El software **emuCompu8** genera un archivo en formato de texto plano, el cual puede ser importado desde logisim y ejecutar el mismo programa del emulador; validando así que lo teórico funciona en la práctica.

Construir en físico el diseño de un CPU propuesto por los alumnos, es inviable por cuestiones de tiempo y de economía. Por lo tanto, el utilizar las tecnologías que nos permiten simular el prototipo del alumno en lugar de construirlo es una gran ventaja porque los alumnos podrán ver funcionando su prototipo sin los

costos onerosos de construirlo físicamente.

NOTA: Si alguien está interesado en el programa y en el modelo del procesador, nos lo puede solicitar por email a cualquiera de los autores.

## Referencias

- [1] Alan Clements. *Principles of Computer Hardware*. OUP Oxford, Oxford ; New York, edición: 4 edition, February 2006.
- [2] Alan Clements. Arms for the poor: Selecting a processor for teaching computer architecture. In *2010 IEEE Frontiers in Education Conference (FIE)*, pages T3E–1. IEEE, 2010.
- [3] Carl Burch. The Guide to Being a Logisim User. Saint John’s University (Collegeville, Minnesota, USA).
- [4] Carl Burch. Logisim: a graphical system for logic circuit design and simulation. *Journal on Educational Resources in Computing (JERIC)*, 2(1):5–16, 2002.
- [5] Carl Burch. Logisim, a graphical tool for designing and simulating logic circuits, 2010.
- [6] Carl Burch y Lynn Ziegler. Science of computing suite (socs): resources for a breadth-first introduction. *ACM SIGCSE Bulletin*, 36(1):437–441, 2004.
- [7] M. Morris Mano. *Diseño digital*. Pearson Educación, 2003. Google-Books-ID: 8WhBtfnaenkC.
- [8] Manuel Román. *MEMORIA DE RESIDENCIA PROFESIONAL: Simulador de procesador de 8 bits*. México (ITESCA), 2017.