



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

Propuesta e implementación de  
instalación experimental para la  
enseñanza teórica práctica de la  
polarización de la luz en el laboratorio de  
Dispositivos Ópticos

**MATERIAL DIDÁCTICO**

para obtener el título de

**Ingeniero en Telecomunicaciones**

**P R E S E N T A (N)**

Tomás Alvarado Chávez

**DIRECTORA DEL MATERIAL DE  
APOYO**

Dra. Selene Pérez García



Ciudad Universitaria, Cd. Mx., 2016

## Agradecimientos y dedicatorias

---

Primero que nada quiero agradecer a mis padres por todo el apoyo brindado durante mi trayectoria escolar, ellos han sido parte fundamental en cada paso que doy y siempre están para mí en los buenos y malos momentos, brindándome unas palabras de aliento para seguir adelante y para así completar mis metas y objetivos.

A mi hermana Ana Paola que siempre ha sido mi referencia y guía tanto en la escuela cómo en la vida diaria, la que siempre me brinda una mano cuando estoy en apuros, la que me ha motivado a seguir sus pasos para seguir creciendo.

A mi tutora Dr. Selene por todo el conocimiento adquirido durante este tiempo que hemos trabajado juntos en este proyecto, así como el apoyo y la paciencia necesaria para el desarrollo éste, también quiero agradecer al Dr. Sergei por su aportación, colaboración y transferencia de conocimiento para la culminación de la presente tesis.

Al profesor Mauricio Enrique Reyes Castillo del centro de investigación en diseño industrial quien me ayudó a la creación y desarrollo de piezas para la parte mecánica de este proyecto.

A mis amigos que siempre me han apoyado a lo largo de mi vida, desde mis amistades más antiguas hasta las amistades más recientes que han aportado algo en mi vida para seguir mejorando, tanto en lo personal como en lo profesional.

Al Fondo de Colaboración del Instituto de Ingeniería y la Facultad de Ingeniería de la UNAM por apoyo en forma del proyecto “Sistema de detección temprano de riesgos en edificios históricos”

A la DGAPA PAPIIT de la UNAM por el apoyo en forma del Proyecto de investigación científica “Implementación de un nuevo método de monitoreo del comportamiento de estructuras hasta su falla de elementos de concreto y materiales compuestos, con arreglos de fibras ópticas” con número de referencia IT102515 y PAPIME “Nuevas prácticas de laboratorio con actividades y experimentos virtuales y reales para el mejoramiento de la enseñanza y aprendizaje en dos asignaturas de la carrera de Ingeniería en Telecomunicaciones” con número de referencia PE101616

## Contenido

---

Resumen	4
Objetivos	5
CAPITULO 1. Antecedentes	
1.1. Algunos de los experimentos realizados con aplicaciones en campo	5
1.2. Antecedentes Teóricos	8
1.2.1. Estados de polarización de la luz; Lineal, circular, elíptica	9
1.2.2. Tipos de polarizadores y prismas de polarización.	15
2.1. Conclusiones	18
CAPITULO 2. Desarrollo Experimental	
2.1. Subsistema Mecánico	23
2.1.1. Diseño de engranes	23
2.1.2. Diseño de cremalleras	24
2.1.3. Diseño de camisas	25
2.1.4. Diseño de soportes	26
2.2. Subsistema de Control	27
2.2.1. Introducción a lo servomotores	27
2.2.2. Programación del Arduino para movimiento de servomotores	29
2.2.3. Diseño de potencia para servomotores	29
2.3. Interfaz de usuario	31
2.3.1. Desarrollo de interfaz GUI en MATLAB	31
2.4. Conclusiones	32
CAPITULO 3. Resultados	
3.1. Subsistema Mecánico	33
3.2. Subsistema de Control	34
3.3. Subsistema Óptico	36
3.4. Pruebas de Funcionamiento de la Instalación	38
3.5. Conclusiones	41
Propuesta de un fabricante.	41
Conclusiones Generales	45
Referencias	46
Anexos	48

## Resumen

---

En el presente trabajo se ha diseñado e implementado una instalación experimental para una práctica de laboratorio con fines docente de la asignatura “Dispositivos Ópticos”, sobre la polarización de la luz.

Este trabajo describe brevemente el conocimiento teórico sobre polarización de la luz, estados de polarización, polarizadores, divisores de polarización, engranes y cremalleras, y el desarrollo de una instalación experimental para que los estudiantes de dicha asignatura puedan observar, entender y aprender los efectos de la polarización de la luz, a través de una práctica de laboratorio significativa y fácil de implementar.

Como parte del proyecto se desarrolló una interfaz GUI (interfaz gráfica de usuario) hecha en Matlab, para que el estudiante pueda manipular los elementos ópticos sin tener contacto directo con ellos.

Se implementó una unidad de control que dirige la instalación experimental a través de un microcontrolador Arduino para interconexión entre el programa desarrollado en la interfaz gráfica de usuario y la instalación experimental. Ésta cumplirá con las necesidades de la práctica sobre la polarización de la luz.

Por último se presenta una comparación entre la instalación experimental desarrollada con otros sistemas comerciales.

## Objetivos

---

Desarrollar e implementar una instalación experimental para una práctica de laboratorio con fines docente de la asignatura “Dispositivos Ópticos”, sobre la polarización de la luz, que se propone lograr los siguientes objetivos de aprendizaje.

Donde los objetivos son los siguientes:

- 1.- El alumno comprenderá, analizará y evaluará el estado de la polarización de la luz, esto a través de una práctica de laboratorio que sea significativa y al mismo tiempo fácil de implementar.
- 2.- Para que el estudiante pueda manipular los elementos ópticos sin tener contacto directo con ellos, los elementos ópticos han sido desplazados a través de una estructura mecánica.
- 3.- El alumno controlará el sistema utilizando un programa con presentación amigable, claro y fácil de operar.

## CAPITULO 1 Antecedentes

---

Actualmente muchas instituciones o universidades han diseñado sus propios sistemas con la finalidad de implementar prácticas de laboratorio para realizar pruebas de polarización de la luz, dependiendo de sus objetivos con afán de que los estudiantes mejoren la experiencia académica de aprender sobre la teoría de la luz. A manera de comparación y para darnos una idea de lo que se tiene en otras universidades se ha desarrollado en este capítulo una descripción de dos breves prácticas para sistemas desarrollados anteriormente en universidades donde han implementado prácticas de polarización de la luz.

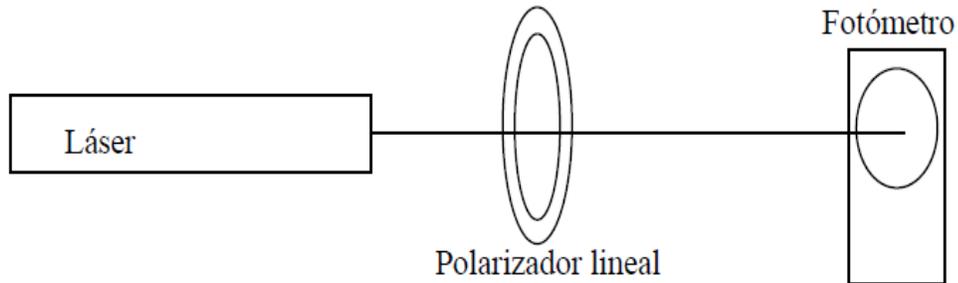
### 1.1 Algunos de los experimentos realizados con aplicación en campo

#### i. Polarización lineal, verificación experimental de la ley de Malus.

Este sistema fue introducido en el curso del 2007 de la Universidad de Castilla-La Mancha, donde fue aplicado para la materia de óptica, por el profesor titular de *Física Aplicada*; Antonio J. Barbero García.

El sistema estudió la transmisión de un haz polarizado a través de un filtro polarizador. Es una verificación experimental de la ley de *Malus*, es decir, se midió la intensidad de un haz de luz polarizado linealmente a través de un fotómetro. El sistema implementado como se muestra en la figura 1.1, es un láser como fuente de

luz polarizada, un polarizador lineal montado sobre un disco graduado en ángulos y un fotómetro para medir la luz transmitida. El diagrama propuesto para este sistema es:



**Figura 1.1** Montaje Experimental de la práctica "polarización lineal, verificación experimental de la ley de Malus" [8].

Ventajas del sistema:

- Con el fotómetro obtendremos datos de medición para hacer una representación gráfica de los datos experimentales tomados en la práctica.
- Con el filtro polarizador lineal graduado, es fácil determinar cuantitativamente la relación entre la intensidad de luz transmitida y el ángulo que forma la dirección de transmisión del polarizador con la dirección del campo eléctrico.

Desventaja del sistema:

- El filtro se mueve de forma irregular debido a que no hay un soporte que regule los grados de rotación del filtro.
- No hay una estructura o soporte de las piezas, por lo que el alumno mueve el material y eso puede ocasionar errores a la hora de tomar los datos experimentales.

## ii. **Polarización de ondas electromagnéticas planas**

Este sistema fue publicado en el curso del 2013-2014 de la Universidad de Almería, donde fue aplicado para la materia de Electromagnetismo y Óptica, por la profesora Dra. María José García Salinas titular de "Física Aplicada".

El sistema estudia las características de la luz polarizada, para comprobar experimentalmente las leyes de *Brewster* y *Malus*. En este sistema mostrado en la figura 1.2, se estudia el comportamiento de la luz a partir de luz polarizada. La luz blanca origen, o el láser utilizado en el laboratorio no están polarizados, un espejo realiza la reflexión con un cierto ángulo llamado de *Brewster* donde éste refleja un haz de luz polarizado y con un analizador se puede determinar la dirección de polarización y por último se determina la dirección de polarización a través de una pantalla.

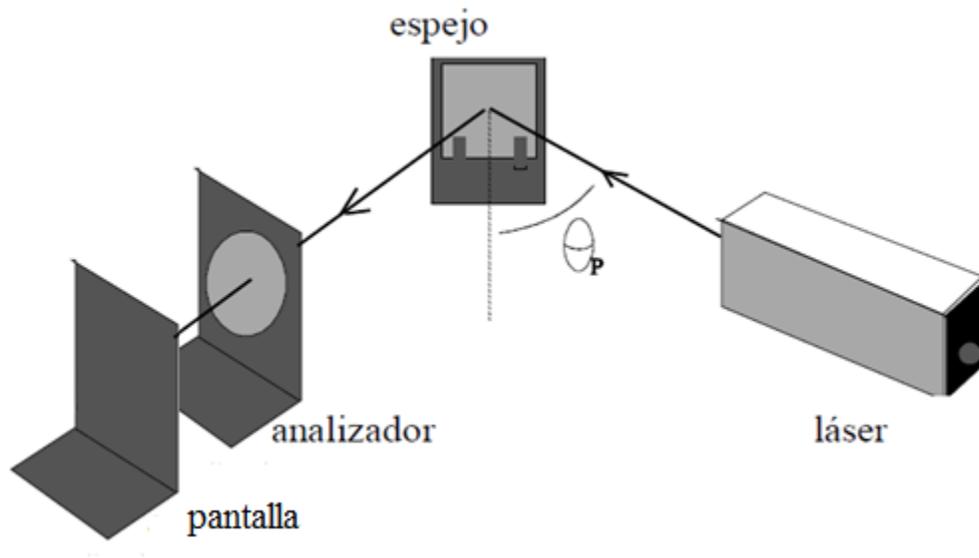


Figura 1.2. Montaje experimental para estudiar la polarización de ondas electromagnéticas [9].

Ventajas del sistema:

- En este montaje puede verse el ángulo de reflexión, si el ángulo de incidencia es  $\theta_p$ , la luz reflejada está totalmente polarizada.
- El alumno puede dibujar el esquema de polarización en un plano (componente X, componente Y), indicando la dirección del eje de transmisión.

Desventaja del sistema:

- El giro del láser es manual y el alumno debe obtener los resultados en el analizador, esto puede ser tardado y a la vez con un alto porcentaje de error.

## 1.2 Antecedentes Teóricos

La polarización óptica fue estudiada por E. L. Malus en 1808. En el siglo diecinueve y principios del veintiuno en física teórica, fue el desarrollo de la teoría electromagnética y la demostración de que la polarización óptica es completamente descrita por ella. Esta teoría es fenomenológica en el sentido de que en vez de intentar explicar por qué los materiales tienen ciertas características fundamentales, se concentra en las propiedades resultantes que mostrará cualquier material con esas características. En el caso óptico, la polarización y todas las otras propiedades ópticas de un material están determinadas por dos o más parámetros fenomenológicos llamados constantes ópticas. La teoría electromagnética tiene poco o nada que decir acerca de por qué un material debe tener esas constantes ópticas particulares o cómo están relacionadas con su carácter atómico. Éste problema ha sido extensivamente investigado en la física de estado sólido del siglo veinte y únicamente se entiende parcialmente. Queda claro, sin embargo, que las constantes ópticas son una función no sólo de la naturaleza atómica del material, es decir, de su posición en la tabla periódica, sino que también son muy sensibles a cómo es preparado. Quizá “parámetros ópticos” serían un mejor término en lugar de constantes ópticas. De cualquier modo, el concepto de las constantes ópticas es extremadamente útil y permite hacer posible, predecir cuantitativamente el comportamiento óptico del material.

Dado que las constantes ópticas son tan fundamentales, las diferencias en su definición son particularmente desafortunadas. La más dañina de dichas diferencias surge de una ambigüedad en la derivación inicial. Las ecuaciones de Maxwell, que forman la base de la teoría electromagnética, resultan en la ecuación de onda, que en unidades mks es:

$$\nabla^2 E = \frac{\epsilon}{c^2} \frac{\partial^2 E}{\partial t^2} + \frac{4\pi\sigma}{c^2} \frac{\partial E}{\partial t} \quad ec(1.1)$$

En donde:

$\nabla^2$  = Operador laplaciano

$E$  = Vector de campo eléctrico de la onda viajante

$t$  = Tiempo

$C$  = Velocidad de la luz

$\sigma$  = Conductividad del material a la frecuencia del movimiento de la onda

$\epsilon$  = Constante dieléctrica del material a la frecuencia de movimiento de la onda

Un concepto cómo la polarización óptica puede ayudar a comprender efectos como la dispersión por modo de polarización (PMD) o pérdidas dependientes de polarización (PDL), dónde éstos efectos pueden ocurrir en los enlaces de fibra óptica. Por ejemplo, una dispersión por modo de polarización (PMD) es una forma de dispersión en donde dos polarizaciones diferentes de luz en una guía de onda (por ejemplo un núcleo de una fibra), viajan a la misma velocidad, pero debido a las imperfecciones y asimetrías al azar viajan a diferentes velocidades, a menos que se compensen. Como consecuencia limita la velocidad a la que los datos pueden ser transmitidos a través de una fibra. En general se presenta los efectos de dispersión PMD y PDL en redes de fibra óptica monomodo, dónde se presenta un estado de polarización menos atenuada en la presencia de ambos PMD y PDL. Los dos estados de polarización no son ortogonales lo que lleva a interferencias que producen anomalías en impulsos propagados.

### 1.2.1 Estados de polarización de luz: Lineal, circular, elíptica

Se entiende mejor a la polarización considerando inicialmente solo estados puros de polarización, y sólo a una onda senoidal coherente a una frecuencia óptica. En la figura 1.3 podemos observar una onda electromagnética descompuesta en su campo eléctrico (de color rojo en el eje  $y$ ) y su campo magnético (de color azul en el eje  $x$ ), donde es un ejemplo de una oscilación del campo electromagnético emitido por un solo láser.

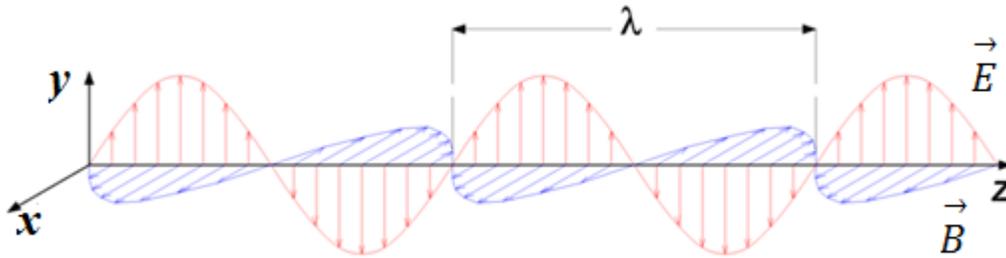


Figura 1.3. Una onda electromagnética de longitud de onda  $\lambda$  donde el campo eléctrico (eje  $y$ ) oscilante en la dirección vertical y el campo magnético (eje  $x$ ) oscilante en ángulo recto al campo eléctrico y ambas son perpendiculares a la dirección de la propagación (eje  $z$ ) [10]

El campo oscila en el plano  $x$ - $y$  mostrado en la figura 1.3, con la onda propagándose en la dirección del eje  $z$ , perpendicular al plano  $x$ - $y$ . En las figuras 1.4 a y b mostrada abajo se trazan el vector del campo eléctrico (vector de Jones) sobre un ciclo completo para una polarización lineal en dos diferentes orientaciones; esas son ambas consideradas un distinto estado de polarización (SOP) por sus siglas en inglés. Observe que la polarización lineal a 45 grados también puede verse como la adición de una onda polarizada horizontalmente

alineada (como en la figura 1.4 a) y una onda verticalmente polarizada de la misma amplitud en la misma fase.

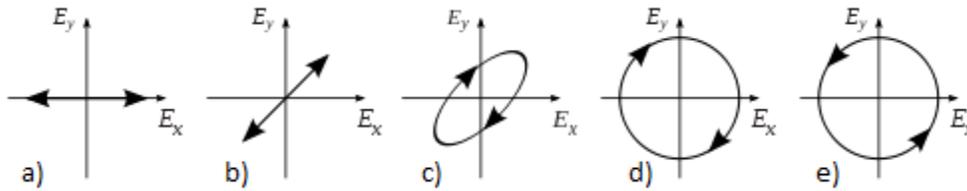


Figura 1.4 Estados de Polarización (SOP) a) vector de campo eléctrico, b) vector del campo eléctrico c) polarización elíptica d) polarización circular y e) polarización circular [10]

Una onda polarizada circularmente es la suma de las dos componentes linealmente polarizadas, con un desfase de 90 grados. Ahora, si introdujéramos un cambio de fase entre esos dos componentes de polarización horizontal y vertical, podríamos generalmente obtener polarización elíptica como se muestra en la figura 1.4 c). Cuando el cambio de fase es exactamente  $\pm 90$  grados, entonces la polarización circular es producida (figuras 1.4 d y e). Así se implementa la polarización circular en la práctica, comenzando con luz linealmente polarizada y empleando una placa retardadora de cuarto de onda para introducir un cambio de fase.

En la figura 1.5 se muestran algunos ejemplos de las diferentes variaciones de la dirección del vector campo eléctrico (senoidal de color azul en el eje vertical), con sus componentes en el plano  $x$ - $y$  mostrados en la figura 1.5 (la senoidales están de color rojo y verde respectivamente). Cada uno de los tres ejemplos corresponde a un tipo de polarización.

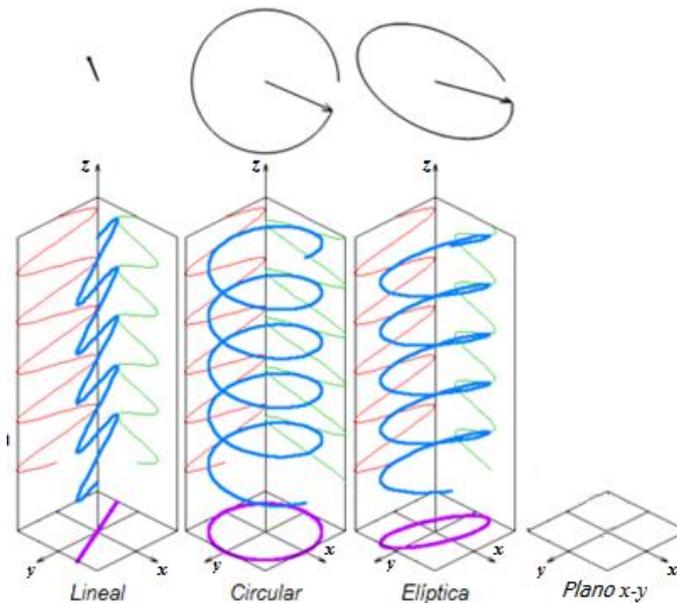


Figura 1.5 Tipos de polarización [10].

## Polarización Lineal

Determinamos cómo una luz linealmente polarizada cuándo el plano  $x$ - $y$  mostrado en la figura 1.3, la componente  $y$  del vector del campo eléctrico está en fase, manteniendo su dirección y solo modificando (de forma senoidal en tiempo y espacio) su magnitud y sentido.

Suponiendo que tenemos dos ondas de luz linealmente polarizadas en la misma frecuencia, desplazándose a través del mismo espacio con la misma dirección, siendo su vector de campo eléctrico colineales. Las ondas superpuestas se combinarán para formar una onda resultante linealmente polarizada. Si observáramos las oscilaciones del campo eléctrico en un haz de luz linealmente polarizada, dónde el movimiento descrito sería lineal o una recta.

Retomando el plano  $x$ - $z$  cómo una referencia de la figura 1.3, consideramos al campo eléctrico ( $E$ ) en un plano como una onda armónica simple, la cual se propaga a lo largo del eje  $z$ . El resultado de la luz será una onda electromagnética donde el campo eléctrico oscilará en el eje  $x$ , perpendicularmente al eje  $z$  y a una determinada frecuencia.

Si tomáramos como referencia el plano  $y$ - $z$  de la figura 1.3, el campo eléctrico se considera de la misma manera en ese plano como una onda armónica simple, que también se propaga en dirección del eje  $z$ , y cuyas oscilaciones se dan en el eje  $y$  perpendicularmente a eje  $z$ .

La onda en plano  $x$ - $z$  y la onda en plano  $y$ - $z$  mostradas en la figura 1.3 pueden se descritas, matemáticamente, por las siguientes ecuaciones:

$$\vec{E}_x(z, t) = \hat{e}_x E_{0x} \cos(kz - \omega t) \text{ ec(1.2)}$$

$$\vec{E}_y(z, t) = \hat{e}_y E_{0y} \cos(kz - \omega t + \phi) \text{ ec(1.3)}$$

En donde:

- $\vec{E}$ : Vector del campo eléctrico
- $E_0$ : Amplitud de la onda
- $\hat{e}$ : Vector de dirección unitario de la onda
- $\phi$ : Diferencia de fases entre las ondas
- $k$ : Vector de onda
- $z$ : Dirección de la onda que está viajando
- $\omega$ : Frecuencia angular de la onda
- $t$ : Tiempo

Las diferencias de fase entre las ondas que viajan en dirección del eje z (figura 1.3), donde la amplitud de dichas ondas pudiera variar, esta diferencia solo determina la dirección de la línea recta que traza el vector del campo eléctrico mientras se propaga.

Si sumamos los vectores de sus componentes del campo eléctrico, esto nos dará como resultado  $\vec{E}$ :

$$\vec{E}(x, y, z, t) = \vec{E}_x(z, t) + \vec{E}_y(z, t) \text{ ec(1.4)}$$

Si el resultando de la diferencia de fases entre ondas ( $\epsilon$ ) es cero, o un múltiplo entero de  $\pm 2\pi$ , ambas componentes se encontrara en fase, es decir la suma vectorial de ambas sería:

$$\vec{E} = (\hat{e}_x \vec{E}_{0x} + \hat{e}_y \vec{E}_{0x}) \cos(kz - \omega t) \text{ ec(1.5)}$$

La superposición de las ondas  $\vec{E}_x$  y  $\vec{E}_y$  en fase da como resultado de la ecuación con una amplitud fija a:

$$(\hat{e}_x \vec{E}_{0x} + \hat{e}_y \vec{E}_{0x}) \text{ ec(1.6)}$$

Para tener una mejor idea de cómo se describe el movimiento lineal, en este tipo de polarización, recordemos que, matemáticamente, los componentes del vector en el eje x y el eje y mostrado en la figura 1.6 se ven así:

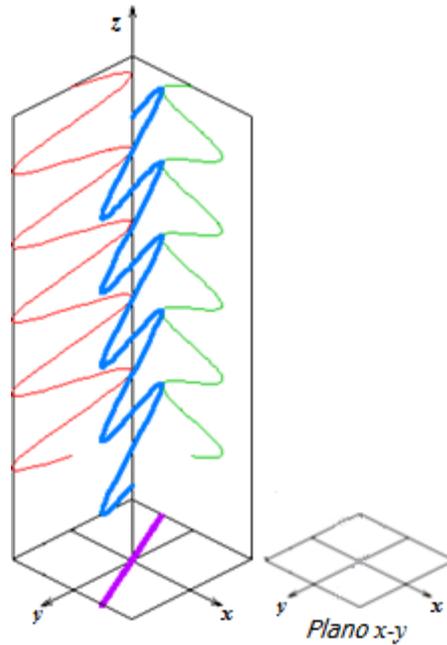


Figura 1.6 Tipo de polarización lineal donde el  $E_x$  es de color verde y el  $E_y$  es de color rojo. [10].

## Polarización Circular

Determinamos cómo una luz es linealmente polarizada cuándo la onda del campo eléctrico en el eje  $x$  y el eje  $y$  de la figura 1.7 deben estar en fase ( $e=0$ ). Es decir, que las fases estén desfasada 90 grados y la amplitud de ambas este exactamente igual, entonces obtendremos una polarización circular. Donde las oscilaciones del campo eléctrico en un haz de luz linealmente polarizada en un plano  $x-y$  (figura 1.7), el resultado sería un movimiento circular.

Si describimos las ondas en el eje  $x$  y el eje  $y$  que se muestra en la figura 1,7 de este tipo de polarización pueden representarse matemáticamente, por las siguientes ecuaciones:

$$\vec{E}_x(z, t) = \hat{e}_x E_0 \cos(kz - \omega t) \text{ ec(1.7)}$$

$$\vec{E}_y(z, t) = \hat{e}_y E_0 \sin(kz - \omega t) \text{ ec(1.8)}$$

En donde:

$\vec{E}$ : Vector del campo eléctrico

$E_0$ : Amplitud de la onda

$\hat{e}$ : Plano de la onda

$k$ : Vector de onda

$Z$ : Dirección de la onda que está viajando

$\omega$ : Frecuencia angular de la onda

$t$ : Tiempo

Donde la amplitud vector del campo eléctrico en el eje  $x$  y en el eje  $y$  mostrado en la figura 1.7, es la misma ( $E_{0x} = E_{0y} = E_0$ ). Es decir que existe un desfase de 90 grados, el campo en el eje  $y$  cambia  $E_0 \cos(kz - \omega t + \varepsilon)$  a  $E_0 \sin(kz - \omega t)$  por lo que la fase  $\varepsilon = -\pi/2 + 2m\pi$  (con  $m = 0, \pm 1, \pm 2, \dots$ ).

Sumando vectorialmente las componentes  $\vec{E}_x$  y  $\vec{E}_y$  se obtiene:

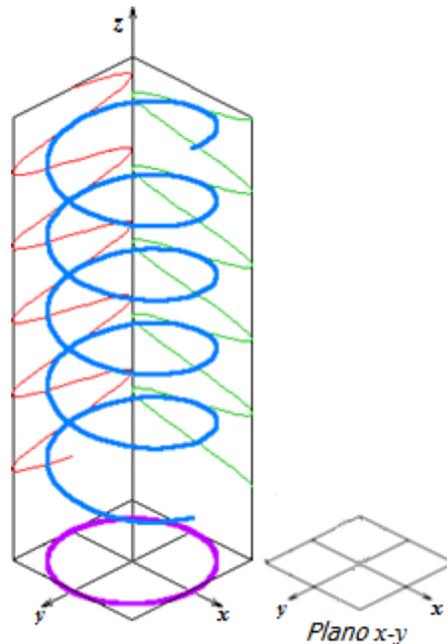


Figura 1.7 Tipo de polarización circular donde el  $E_x$  es de color verde y el  $E_y$  es de color rojo [10].

## Polarización Elíptica

La polarización elíptica se presenta cuando los campos magnéticos ( $\vec{E}_x$  y  $\vec{E}_y$ ), la diferencia de fase entre las ondas ( $\varepsilon$ ) y la amplitud son distintas.

Un valor aleatorio puede ser cualquier valor donde  $\varepsilon$  puede tomar el valor de 0 o  $\pi/2$  (o múltiplos de éste), también se puede presentar que la amplitud sea la misma como el caso de la polarización circular o una diferencia de fases entre ondas ( $\varepsilon$ ) es cero, o un múltiplo

entero de  $\pm 2\pi$  como en el caso de la polarización lineal, se podría decir que estas dos polarizaciones son un “caso espacial” de la polarización elíptica.

Por los valores diferentes que presenta al emitir la luz, la mayoría de las veces veríamos luz elípticamente polarizada (si la viéramos de frente en un plano).

Para tener una mejor idea de cómo se describe el movimiento elíptico en este tipo de polarización, recordemos que, matemáticamente, los componentes del vector en el eje  $x$  y el eje  $y$  como se muestra en la figura 1.8 se ven así:

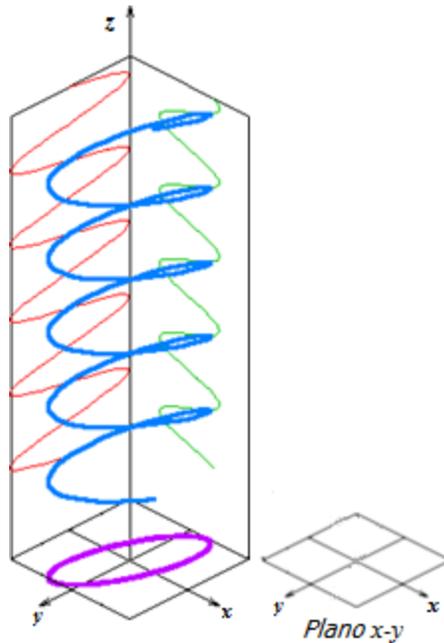


Figura 1.8 Tipo de polarización elipse donde el  $E_x$  es de color verde y el  $E_y$  es de color rojo [10].

Ley de Malus. La ley de Malus describe la diferencia de variación de la intensidad de la luz trasmite, es decir, si sobre un polarizador lineal se hace incidir luz linealmente polarizada, la intensidad de la luz que lo atraviesa ira disminuyendo progresivamente a medida que el ángulo ( $\theta$ ) vaya aumentando.

### 1.2.2 Tipos de polarizadores y prismas de polarización.

Un filtro polarizador es un material con la característica de separar las dos componentes ortogonales del campo eléctrico de la onda y solo dejar pasar una componente de la onda electromagnética.

En los prismas de polarización están hechos de muchos minerales, específicamente aquellos que no tienen una estructura cristalina cúbica, son birrefringentes, casi todos los prismas polarizadores usados en el visible, ultravioleta cercano, y en el infrarrojo cercano regiones

del espectro están hechos de vidrio calcita óptica, que exhibe fuerte birrefringencia en un amplio intervalo de longitudes de onda.

## **Tipos de polarizadores.**

Para poder definir cuáles son los tipos de polarización de la luz es indispensable definir que la luz es una radiación electromagnética transversal, en otras palabras, la oscilación de los campos electromagnéticos son perpendiculares a la dirección de propagación, esto puede cambiar con diferentes dispositivos físicos que permitan filtrar una sólo oscilación, por lo que en este estado la luz está polarizada.

Cada uno de esos tipos de luz polarizada (lineal, circular y elíptico), puede ser representado como una rendija (filtro polarizador) que permite el paso únicamente de la luz que oscila en el plano paralelo al vector normal a la superficie de la rendija. La luz transmitida resultante de la rendija se considera luz polarizada.

En nuestro caso mostrado en la figura 1.9, el filtro polarizador comercial consiste en una cadena de polímero estirada al límite, de modo que las moléculas actúan como una rendija, al ocurrir este efecto el filtro polarizado absorbe fuertemente una oscilación de un campo electromagnético en cambio para la otra oscilación del campo electromagnético es muy transparente, como describe la ley de Malus.

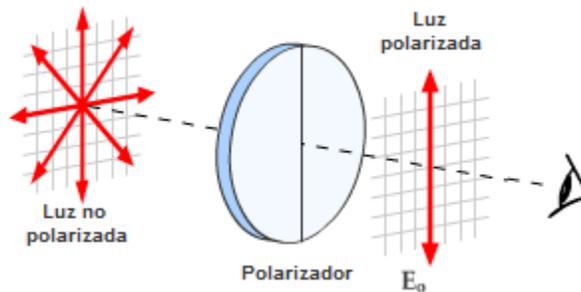


Figura 1.9 Luz polarizada verticalmente [11].

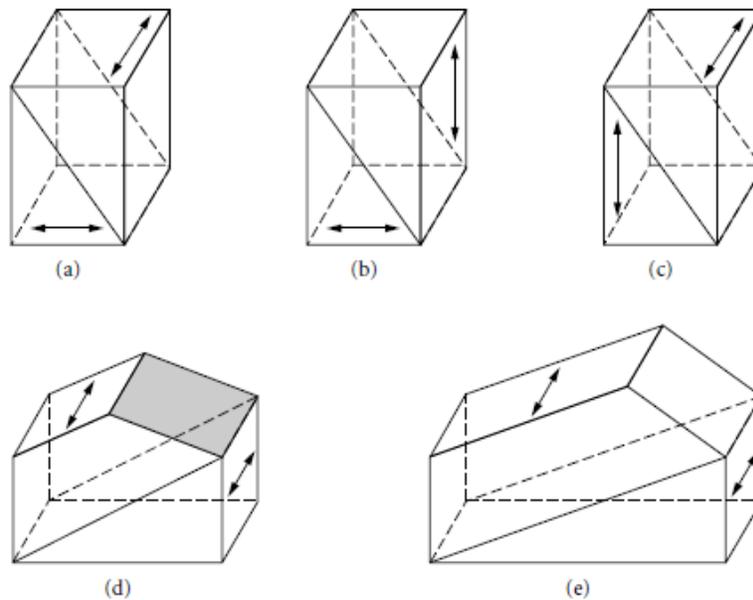
## **Prismas de polarización.**

Para describir un prisma polarizado se debe encontrar alguna manera de separar los dos haces polarizados. En regiones de longitud de onda donde la calcita está absorbiendo (y por tanto, solo un volumen mínimo de calcita puede ser usado), esta separación ha sido hecha simplemente usando un corte muy delgado de calcita. El haz de luz que incide queda

restringido a un haz angosto. En regiones de longitud de onda más favorables, donde la cantidad de calcita a través de la cual pasa la luz no es tan crítica, usualmente se ocupan diseños más sofisticados. Dichos prismas pueden ser divididos en dos categorías principales, prismas polarizadores convencionales y prismas polarizadores corta-haces, y una tercera categoría, prismas Feussner.

En prismas polarizadores convencionales, solo la luz polarizada en una dirección es transmitida. Esto se logra al cortar y cementar las dos mitades del prisma juntas, de manera que el otro haz experimenta una reflexión interna total en la union. Normalmente rebota hacia un lado, donde es absorbido por un recubrimiento que contiene materiales como negro de humo (pigmento negro). Dado que el rayo ordinario, que tiene el índice más alto, es el que normalmente rebota, el negro de humo es comúnmente mezclado en un aglutinante de índice alto apropiado, cuando se usan láseres de alta potencia, el recubrimiento es omitido para evitar sobrecalentar el prisma, y la luz se absorbe externamente.

Hay muchos prismas de polarización, pero para esta tesis nos interesan los prismas de polarización llamados BEAM-SPLITTER (divisor de haz), la figura 1.10 muestra 5 tipos diferentes de prismas BEAM-SPLITTER.



**Figura 1.10** Vistas tridimensionales de varios tipos de prismas divisores de haz de polarización : ( a ) Rochon , ( b ) Sénarmont ; ( c ) Wollaston , ( d ) Crianza ( cara sombreada está recubierta con plata ) , y ( e ) divisor de haz Glan- Thompson [4].

Los tres prismas polarizadores clásicos separadores de haz, son los Rochon, Senarmont y Wollaston, mostrados en la figura 1.10a a 1.10c, en vista lateral en las figuras 1.10a a 1.10c. Además, cualquier prisma polarizador puede usarse como separador de rayos cambiando la forma de un lado y retirando la cubierta absorbente de su superficie. Dos

ejemplos de dichos prismas son el prisma Foster, en el cual los rayos ordinarios y extraordinarios emergen en ángulos rectos entre sí, y el prisma de rayo-separante de Glan-Thompson, en el cual el rayo ordinario emerge normal hacia un lado (figura 1.10d y e, figura 1.10d y e).

En prismas polarizadores, los ejes ópticos son siempre paralelos a cada uno en las dos mitades del prisma. En contraste (figura 1.11), los ejes ópticos en las dos mitades de los prismas polarizadores rayo-separantes Rochon, Sénarmont, y Wollaston están en ángulos rectos entre sí. El cristal de cuarzo es usado con frecuencia para hacer prismas rayo-separantes, y dichos prismas pueden usarse hasta el ultravioleta. En aplicaciones que no requieren longitudes de onda tan cortas, la calcita es preferible pues provee una separación angular mayor de los rayos (típicamente 10 grados en comparación a los 0.5 del cuarzo) y no produce rotación óptica.

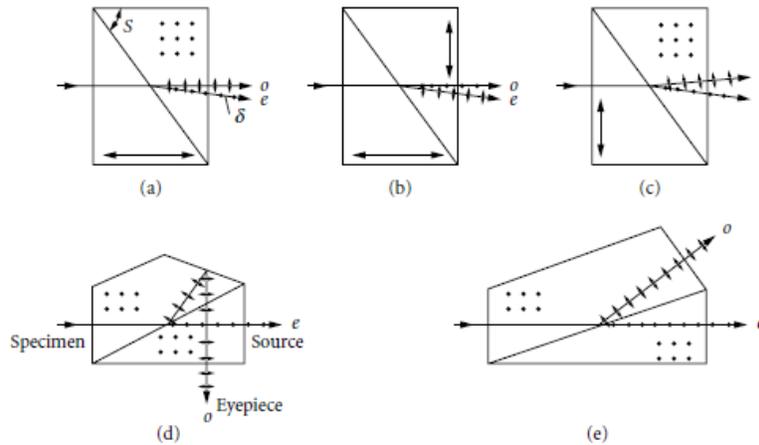


Figura 1.11 Vistas laterales de prismas polarizadores rayo-separantes de la figura 1.10 Las direcciones de los ejes ópticos están indicadas por los puntos y las flechas robustas de punta doble. El ángulo del corte para el prisma Rochon es S, donde los prismas son: ( a ) Rochon , ( b ) Sénarmont ; ( c ) Wollaston , ( d ) Crianza ( cara sombreada está recubierta con plata ) , y ( e ) divisor de haz Glan- Thompson [4].

### 1.3 Conclusiones

En este capítulo se describió cómo otras universidades emplearon mecanismos para realizar prácticas de polarización de la luz dependiendo el objetivo de cada una de ellas, pero el mayor problema de ellas es que el alumno gira las piezas o filtros, esto ocasiona que el alumno toque el material con riesgo a descomponer o deteriorarlo rápidamente.

Otro punto importante de mencionar fue el porcentaje de error de estas prácticas, ya que los alumnos manipulan las piezas y puede ocasionar un gran grado de error o en el peor de los casos, que el objetivo de las prácticas no se cumple o se aproxime a conseguir resultados no óptimos para el aprendizaje del alumno.

También obtuvimos nociones fundamentales de la luz polarizada, que acabamos de explicar, confirmando una de las teorías de la luz, a saber, que la luz tiene la característica de propagarse por un medio a través de ondas transversales. La luz polarizada es una herramienta eficaz para los científicos que ocupan en el análisis de sistemas ópticos de luz y el estudio de las estructuras de los cristales.

Fue importante contar con buenos conocimientos básicos sobre el comportamiento de la polarización de la luz así como del prisma polarizador, para reafirmar el conocimiento en una práctica simple pero significativa del estado de polarización de la luz.

## CAPITULO 2 Desarrollo Experimental

La práctica desarrollada está compuesta por dos partes:

En la primera parte el objetivo es que el estudiante observe los efectos de los filtros polarizadores lineales sobre la luz no polarizada y consiste en hacer incidir un haz de luz no polarizada por tres filtros de polarización de luz lineales, el primer filtro de polarización esta fijo, es decir se posiciona el primer polarizador en forma vertical u horizontal en un soporte fijo, no se mueve y solo deja pasar un estado de polarización (vertical u horizontal) dependiendo de la posición del filtro, los dos filtros de polarización restantes fueron colocados en piezas móviles, los cuales pueden ser rotados para que los filtros tuvieran movimientos hacia la derecha o izquierda dependiendo de lo cambios realizados por el alumno a través del programa realizado.

En la figura 2.1 se muestra el esquema para ejemplificar lo descrito anteriormente:

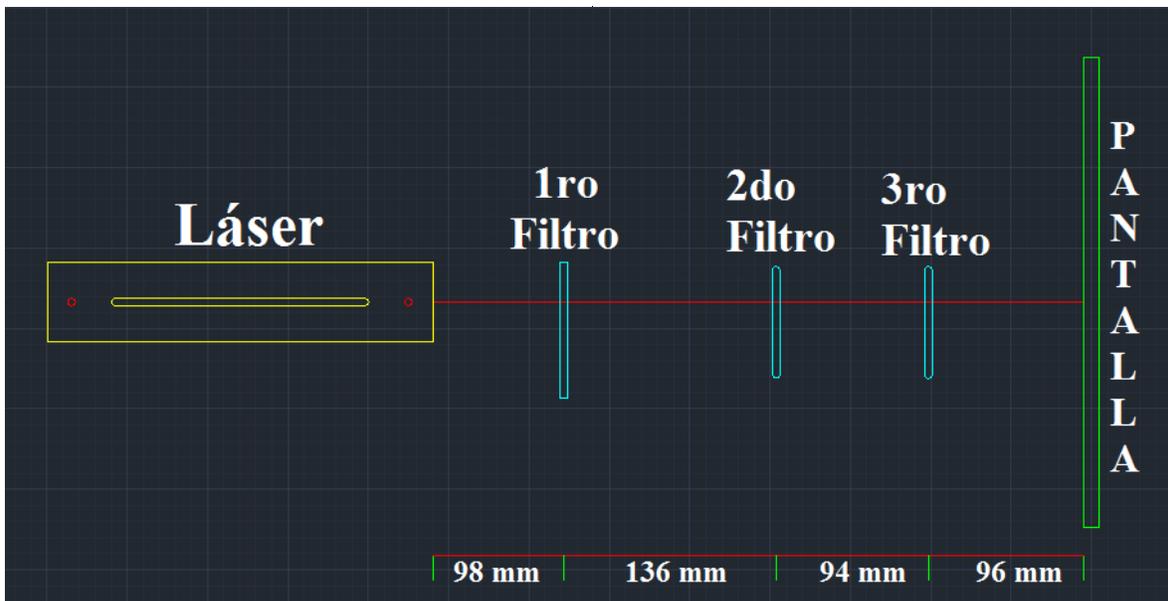


Figura 2.1 Esquema de la polarización de la luz (primera parte) [13].

Es decir el láser posicionado en la parte izquierda emitirá un haz de luz no polarizada, al pasar por el primer filtro polarizador, la onda electromagnética en el plano  $x$ - $y$  (figura 2.2) se propagará perpendicularmente al plano, esté solo dejará pasar una sólo componente del plano  $x$ - $y$  (figura 2.2) dependiendo de la posición del filtro que podría ser vertical u horizontal.

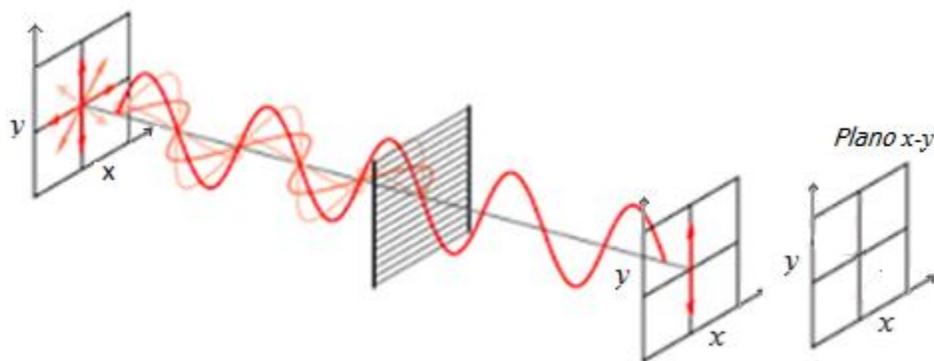


Figura 2.2 luz no polarizada emitida por un láser a través de un filtro polarizador (primera parte) [11].

Una vez que el haz de luz ha atravesado el primer filtro de polarización (fig. 2.1) solo tendremos una componente del plano  $x$ - $y$  (fig. 2.2) original donde el siguiente filtro ha sido rotado por el alumno para dejar pasar la componente del haz de luz o no, dependiendo la posición en la que el alumno colocó el filtro polarizador de la luz. Lo mismo pasará con el tercer y último filtro, si el alumno colocó en la misma posición los tres filtros dejan pasar la componente resultante del primer filtro y el haz de luz se mostrará en la pantalla final, de lo contrario si algún filtro está puesto ortogonalmente al primer filtro de luz, ese filtro no dejará pasar la componente por lo cual no se verá nada en el siguiente filtro o en su defecto en la pantalla final.

En la segunda parte el objetivo es que el estudiante compruebe con un prisma divisor de la polarización que la luz no polarizada puede ser descompuesta en dos componentes de polarización lineal ortogonales entre sí. Este consta de un divisor de polarización llamado prisma Wollaston, el haz de luz del láser incide sobre el prisma Wollaston, el cual dividirá el haz en sus dos componentes (una es la horizontal y la otra es la vertical, donde cada haz de luz es dirigido a un filtro polarizador, los filtros se rotan en forma vertical u horizontal de forma independiente).

La figura 2.3 muestra el filtrado de los estados de polarización por un prisma de Wollaston:

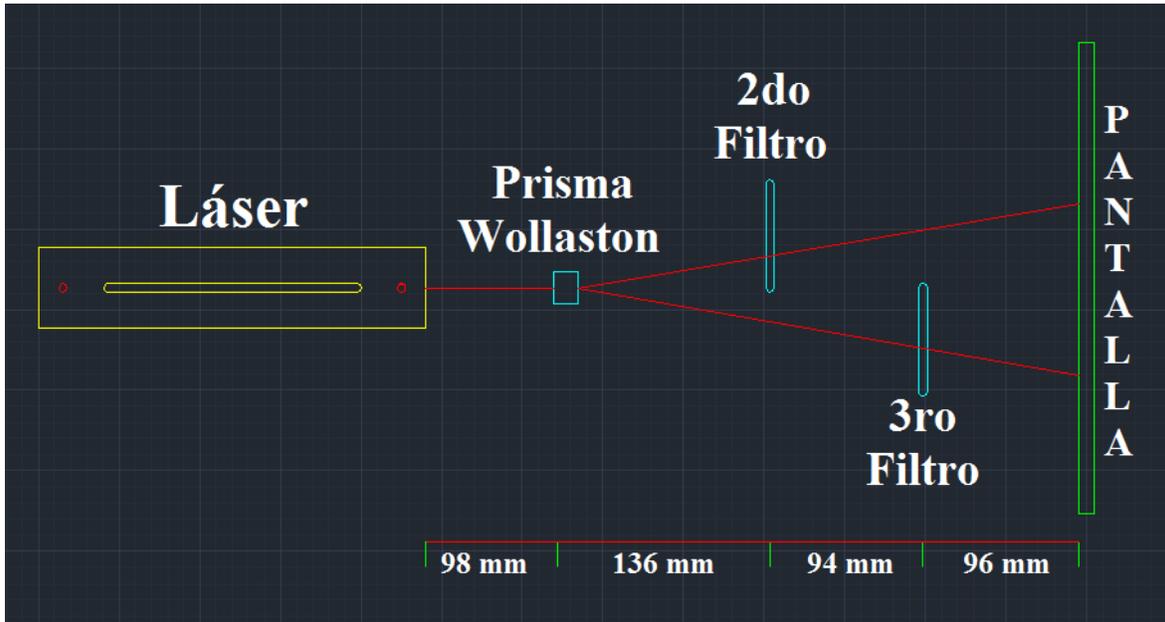


Figura 2.3 Esquema del montaje para el filtrado de la luz utilizando un prisma Wollaston [13].

El láser posicionado en la parte izquierda emitirá un haz de luz no polarizada, lo cual genera una onda electromagnética en el plano  $x$ - $y$  (figura 2.4) propagándose en el eje  $z$ . Al pasar el divisor de polarización, este dividirá las componentes de la onda, el resultado será emitir dos haces de luz polarizados linealmente ortogonales entre sí, (fig. 2.4), los dos haces de luz tendrán una abertura de 20 grados debido a que es el rango que nos brinda el fabricante para el prisma Wollaston que utilizamos.

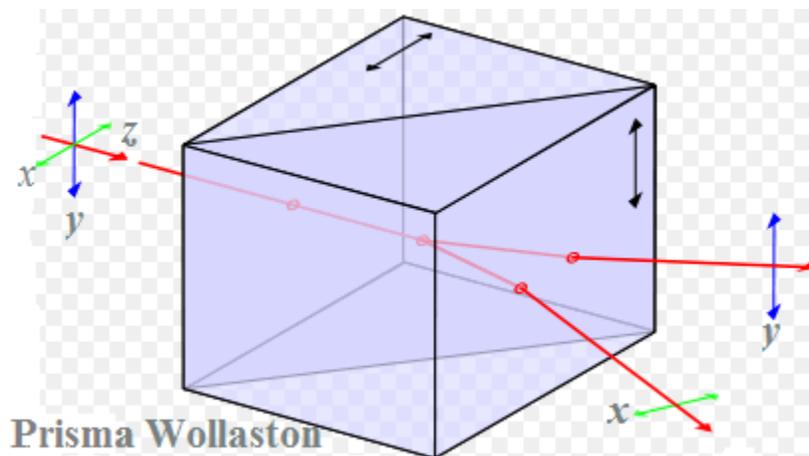


Figura 2.4 Esquema de la polarización de la luz (segunda parte) [11].

Una vez que el haz de luz ha atravesado el divisor de polarización (fig.2.4) tendremos dos componentes una en el eje  $x$  y otra en el eje  $y$ , cada componente atravesara un filtro de polarización de la luz respectivamente donde el alumno debe manipular los filtros para

analizar el estado de polarización de cada componente y el resultado se verá en una pantalla al final de cada filtro.

A continuación se describe el desarrollo de cada uno de los subsistemas que conforman el esquema experimental, dentro de las partes del sistema son: el diseño mecánico, el subsistema de control y por último la interfaz de usuario. Donde se describió cómo fueron diseñadas y fabricadas cada una de estas piezas, cómo se desarrolló la implementación del control, por último, la realización de una interfaz de usuario para que resulte fácil la manipulación del equipo y el alumno pueda operar el equipo experimental de manera óptima y adecuada.

## **2.1 Subsistema Mecánico**

En este capítulo se describe el diseño de las partes mecánicas que componen la instalación experimental que son: engranes, cremalleras, camisas, soportes y que herramientas fueron implementadas para realizar la elaboración de las piezas.

### **2.1.1 Diseño de Engranes**

Para el diseño de engranes el primer paso es situar los centros y dibujar las circunferencia (fig.2.5), donde normalmente cuando se refiere a un engrane que tiene  $x$  centímetros de diámetro estos se refieren al diámetro de la circunferencia primitiva. Si se designan los radios de las dos ruedas por  $r_1$  y  $r_2$  respectivamente, y las velocidades angulares por  $w_1$  y  $w_2$ , por lo que teniendo dos circunferencias primitivas la velocidad lineal de la línea primitiva es:

$$V = w_1 r_1 = w_2 r_2$$

Donde ponemos la ecuación uno en relación entre los radios y las velocidades angulares

$$\frac{w_1}{r_1} = \frac{w_2}{r_2}$$

El siguiente paso para estudiar los engranes (fig. 2.5) es dibujar las líneas  $ab$  por el punto primitivo P perpendicular a la línea de los centros.

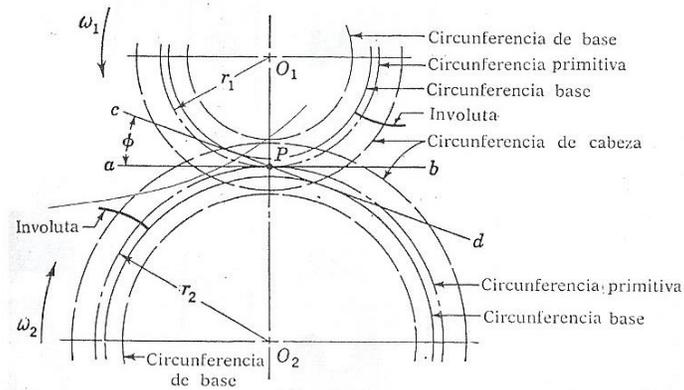


Figura 2.5 Dibujo de un engrane [5].

Esto ha permitido que los dos engranes giren libremente sin que esté genere algún problema o atasco entre ellos, ya que el punto “P” es el punto de inflexión entre los dos círculos primitivos. El siguiente cálculo ha sido el de las circunferencia de base y la circunferencia de cabeza que determinaron la altura de los dientes de cada engrane así como las profundidades del mismo, por último el grosor de cada diente de cada engrane lo determina la circunferencia base, uniendo todas las circunferencias con los cálculos respectivos, pudimos generar engranes funcionales que sean capaz de embonar y girar libremente.

El cálculo de cada uno de los engranes implementados se describe en el Anexo A para detallar más a fondo de los cálculos de los dos tipos de engranes implementados en esta práctica.

### 2.1.2 Diseño de cremalleras

En mecánica, una cremallera (figura 2.6) es un prisma rectangular con una de sus caras laterales tallada con dientes. Estos pueden ser rectos o curvados y estar dispuestos en posición transversal u oblicua.

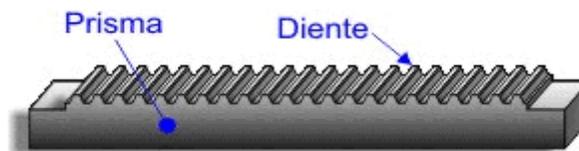


Figura 2.6 Dibujo de una cremallera [12].

Desde el punto de vista tecnológico podemos considerarla como un caso particular de la rueda dentada, pues puede suponerse que es un engranaje de radio infinito.

Se emplea (figura 2.7), junto con un engranaje (piñón), para convertir un movimiento giratorio en longitudinal o viceversa. Tiene gran aplicación en apertura y cierre de puertas

automáticas de corredera, desplazamiento de órganos de algunas máquinas herramientas (taladros, tornos, fresadoras...), cerraduras, microscopios, gatos de coche.

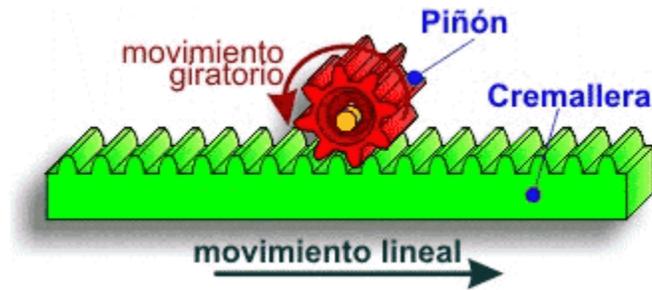


Figura 2.7 Dibujo de una cremallera con un piñón [12].

Para el cálculo de cada uno de las cremalleras, se describe en el Anexo B para detallar más a fondo los cálculos de la cremallera desarrollada en esta práctica.

### 5.1.3 Diseño de “camisas”

Las “camisas”, son el soporte de los engranes, la cual tiene un diseño para no afectar el engrane principal (anexo d, figura d.3), el que contendrá el filtro polarizador de luz, se le llaman “camisas” porque de una manera protegen a los engranes, ya que hay una camisa delantera y otra trasera como podemos observar en la figura (fig.2.8):

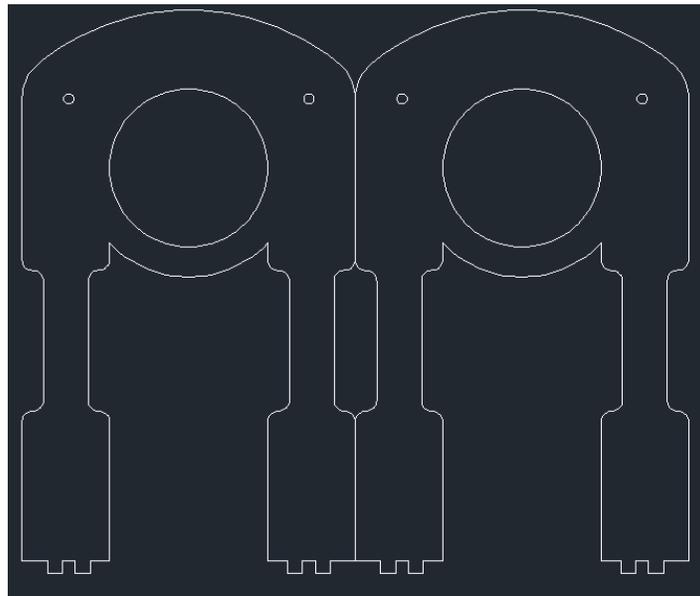


Figura 2.8 Diagrama de “camisas” en AutoCAD [13].

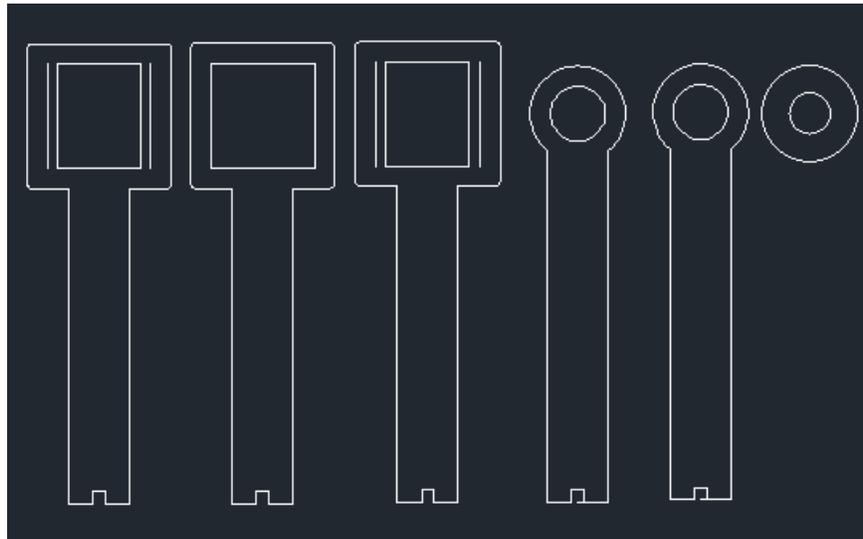
### 2.1.4 Diseño de soportes

los soportes, son un conjunto de piezas que soportan diferentes piezas importantes en la instalación como son:

- El servomotor
- La cremallera y piñón
- Engranajes y filtro

En cada uno de estos puntos hay piezas para soportar las piezas como se muestran a continuación.

En esta figura 2.9 se muestran los soportes del filtro polarizador y el divisor de polarización, las primeras tres piezas son las del primer filtro de polarización de la luz donde esas tres piezas van unidas y las últimas tres piezas son las del divisor de polarización.



**Figura 2.9** Diagrama de una cremallera con un piñón [13].

En la figura 2.10 se muestran tres soportes, el soporte a) es donde se desliza la cremallera, el dispositivo b) es el soporte del filtro polarizador de luz y el divisor de polarización, para desplazarla dependiendo, es decir, si se ha requerido el filtro polarizador de la luz o en divisor de estado de la polarización de la luz y por último, el soporte c) que es del servomotor.

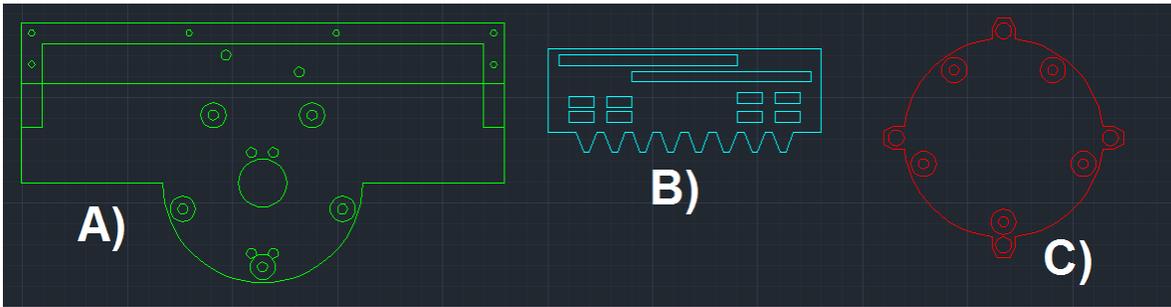


Figura 2.10 Diagrama de una cremallera con un piñón [13].

En la figura 2.11 se muestran cuatro soportes, la figura b) es una cremallera que sostiene las camisas para poder desplazarla dependiendo si está en el filtro polarizador de la luz o en el divisor de polarización de la luz, la figura a) es un soporte donde se desliza la cremallera (fig. b), el soporte c) es la base de la cremallera. Por último, el soporte d) es la pieza que soportan el servomotor.

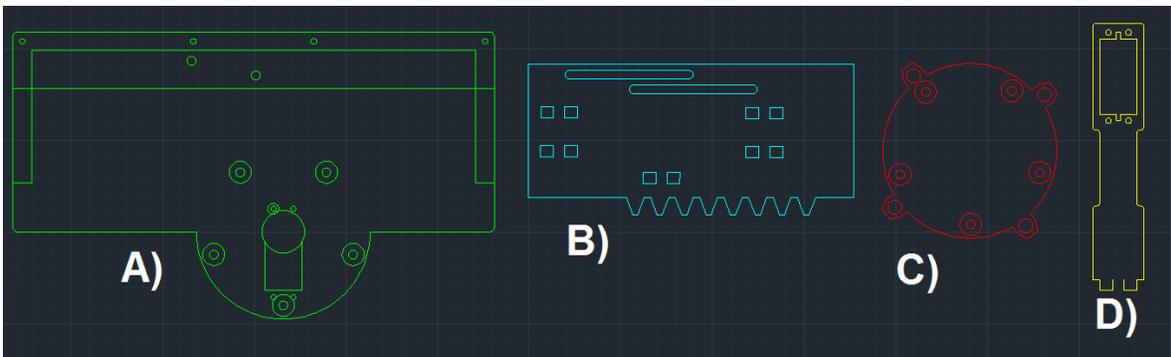


Figura 2.11 diagrama de algunos soportes, a) soporte de la cremallera, b) cremallera que soporta la pieza del servomotor y camisas, c) soporte de la figura a), d) figura que sostiene el servomotor [13].

## 2.2 Subsistema de control

En este punto describimos la comunicación entre la interfaz gráfica de usuario y el movimiento de las piezas con los servomotores, la comunicación entre esas piezas fue realizada con una tarjeta de Arduino para la manipulación adecuada para los servomotores y una etapa de potencia para darle la corriente adecuada a los motores.

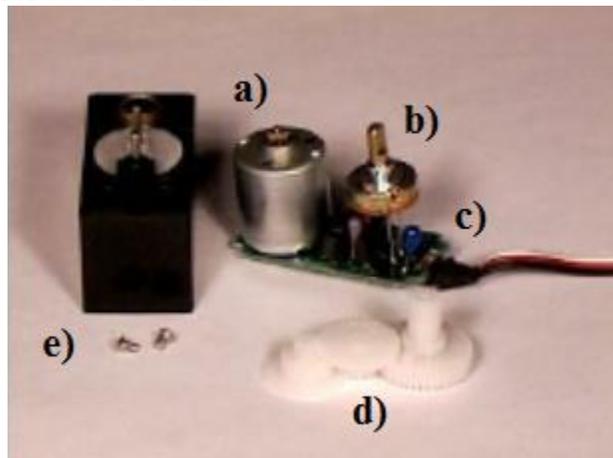
### 2.2.1 Implementación de los servomotores

Los servomotores son dispositivos pequeños que a través de engranes pueden colocar una posición angular específica enviando una señal codificada, esta señal codificada manda un

dato de entrada y el dispositivo mantendrá dicha posición angular, cuando la señal codificada cambia, la posición angular de los engranes también cambia de posición.

Un servomotor está compuesto de un motor pequeño con un conjunto de engranes para generar el torque necesario, este motor contiene internamente un circuito de control que es el traductor de la señal codificada a posición angular. Una muestra de un servomotor internamente se muestra en la figura 2.12, donde podemos ver el circuito de control, el motor, un conjunto de engranes, y la caja que contiene todo el mecanismo, en la misma figura (2.12) se observan 3 alambres de conexión externa, que son:

- Alimentación Vcc (+5 volts) de color rojo.
- Conexión de tierra (GND) de color negro.
- Cable de control de color blanco.



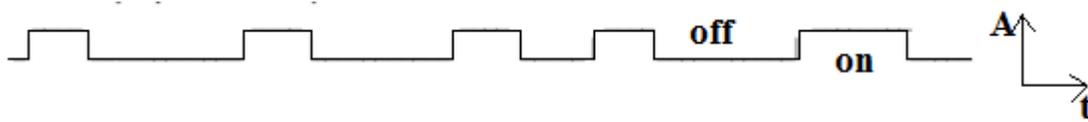
**Figura 2.12** Partes de un servomotor, a) motor de DC, b) potenciómetro, c) circuito de control, d) engranes de precisión, e) caja para proteger al servomotor [18].

Para que un servomotor pueda funcionar adecuadamente, éste debe contener un circuito de control y un potenciómetro donde el potenciómetro permite que el circuito de control pueda regular la posición angular del servomotor. Si la señal codificada cambia su valor el circuito de control y el potenciómetro pondrán al motor en la posición angular deseada. Un servomotor normalmente se usa para controlar un movimiento angular entre 0 y 180.

Si nosotros introducimos una señal codificada indicando el ángulo deseado, en un rango de 0 grados hasta 180 grados. La señal de entrada codificada enviará una serie de pulsos. En un tiempo de encendido el pulso indica el ángulo al que debe posicionarse como se muestra en la figura 2.13.

Si el pulso codificado es igual a 1 [ms] eso equivale a una posición de 0 grados, por otro lado si tenemos un pulso codificado de 2.0 [ms] esto es igual al grado máximo del servomotor. Los límites que recomienda el fabricante es entre 1 ~ 2 [ms], por lo general se

usa un rango mayor de 1.5 [ms] para obtener un ángulo mayor. El que regula y limita el giro del servomotor es el potenciómetro y los límites mecánicos construidos en el servomotor. Un sonido de zumbido normalmente indica que se está forzando por encima al servomotor, entonces debe disminuir un poco.



**Figura 2.13** Ejemplo de la señal codificada de un servomotor [18].

El tiempo de apagado en el servomotor puede estar alrededor de los 20ms. Esto puede ser de una manera distinta, es decir, puede variar de un pulso a otro. Los pulsos que ocurren frecuentemente en el tiempo de OFF y podría escucharse un sonido de zumbido o alguna vibración en el eje. Si el espacio del pulso es mayor de 50ms (depende del fabricante), entonces el servo podría estar en modo “reposo” entre los pulsos. Entraría a funcionar en pasos pequeños y el rendimiento no sería el óptimo.

### **2.2.2 Programación del Arduino para movimiento de servomotores**

Una tarjeta Arduino puede controlar servomotores a través de un microcontrolador programado, donde fueron configurados todos los parámetros de entrada y salidas. Es decir, que puertos de salida están disponibles para mandar la señal codificada (el tren de pulsos) para convertir la señal de entrada al servomotor para que esta señal pase por el proceso interno del servomotor y coloque el ángulo en posición.

Los servomotores tienen tres hilos: alimentación, tierra, y señal. El cable de alimentación de color rojo de cada servomotor fue conectado a un circuito eléctrico para el sistema de potencia de los servomotores. El cable a tierra de color negro también fue conectado a un circuito eléctrico. El cable de señal de color blanco de cada servomotor fue conectado a cada pin digital en la placa Arduino. Teniendo en cuenta que los servomotores consumen energía considerablemente, porque implementamos el circuito eléctrico para el sistema de potencia (es decir, el pin +5 V en la placa Arduino).

### **2.2.3 Diseño de Encendido/Apagado para servomotores**

Se diseñó un circuito eléctrico para el sistema de encendido/apagado de los servomotores y del láser implementado donde también se controla su estado (encendido/apagado).

En la figura 2.14 se muestran 2 entradas y una salida las cuales están conectadas a dos relevadores (respectivamente cada entrada está conectada a un dispositivo diferente, una de ellas al láser y la otra a los servomotores), los relevadores son los que dejarán pasar la corriente o no a través de su mecanismo de bobina que se han accionado a través del

Arduino mandándole un pulso en la entrada 2 y 3 de la parte de conexión digital para el Arduino PWN.

Las conexiones digitales para el Arduino (PWM) correspondientes a los puntos 7-11, se conectan al Arduino en las entradas de PWN del 7-11 para mandar los pulsos correspondientes a los motores, esto se manipulará por medio de la interfaz gráfica de usuario, donde el alumno controlará el movimiento de cada filtro.

Por último tenemos el módulo de motores donde requerimos una fuente de 5 [V] para alimentar los motores con 2 [A] de corriente para su óptimo funcionamiento.

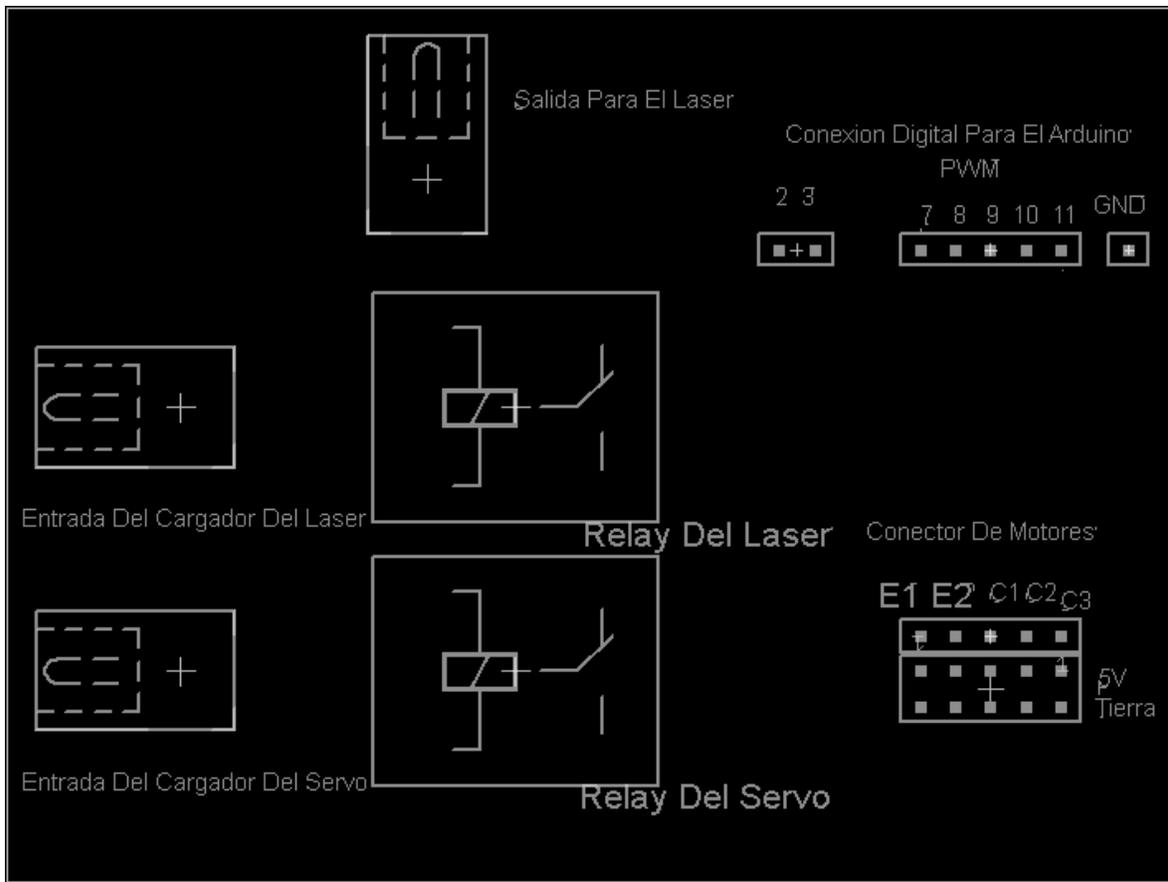


Figura 2.14 Esquema del circuito del sistema de encendido y apagado del láser y de los servomotores [16].

## 2.3 Interfaz gráfica de usuario

En informática, la interfaz gráfica de usuario (GUI), es un tipo de interfaz de usuario que permite a los usuarios interactuar con los dispositivos electrónicos a través de iconos gráficos utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

Las acciones en GUI se realizan generalmente a través de la manipulación directa de los elementos gráficos. Además de las computadoras, la interfaces gráficas de usuario se pueden encontrar en los dispositivos portátiles, como reproductores de MP3, reproductores de medios portátiles, dispositivos de juegos, electrodomésticos, oficina y equipo industrial.

### 2.3.1 Desarrollo de interfaz GUI en MATLAB

En la figura 2.15 se muestra la interfaz de usuario, donde del lado superior izquierdo tenemos dos botones los cuales encienden o apagan el láser y los servomotores, en la parte inferior izquierda se muestra el botón para posicionar qué práctica desea realizar el alumno. La práctica número 1, que es la de filtros polarizadores de luz o la práctica número 2 que es la de divisor de polarización de la luz, en la parte derecha colocamos el movimiento de los filtros polarizadores colocado en los dos engranes grandes.

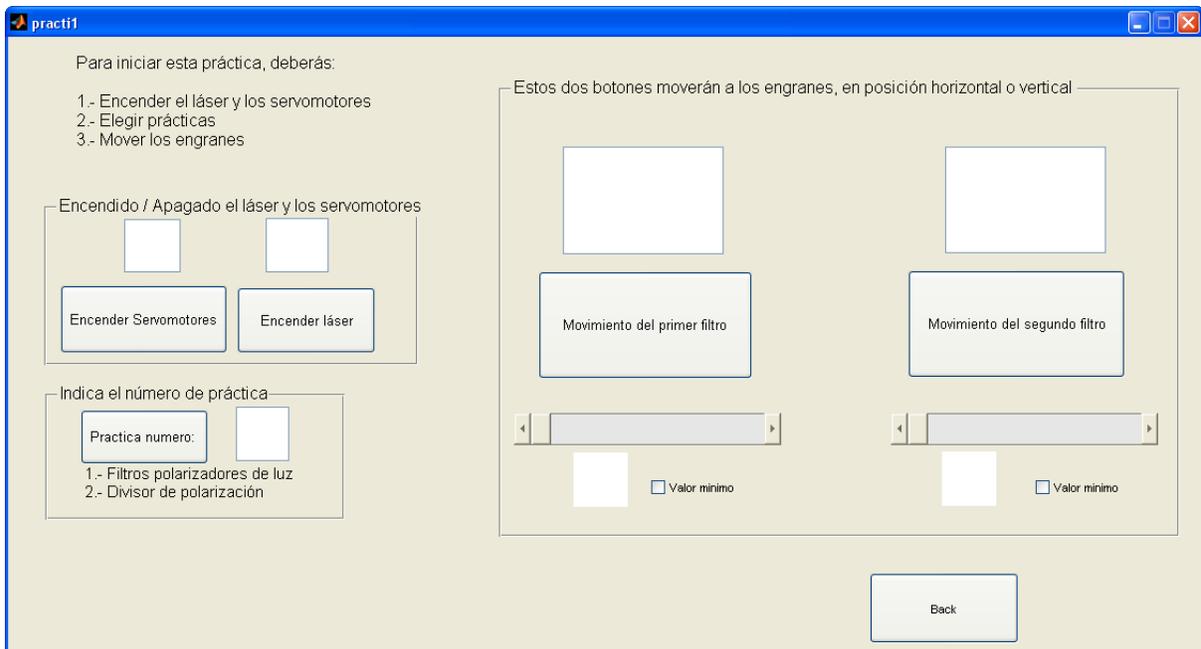


Figura 2.15 Interfaz gráfica de usuario que es fácil de operar para realizar la práctica de "Dispositivos ópticos" [15] .

## **2.4 Conclusiones.**

En esta parte hemos realizado todo el diseño de la instalación experimental donde el cual consistió en tres partes: la parte mecánica, la parte de control y por último la interfaz gráfica de usuario.

En el desarrollo del subsistema mecánico escribimos que es un engrane y se realizaron los cálculos pertinente para poder realizarlos en los programas AutoCAD y SolidWorks, fue importante investigar los cálculos pertinentes para el perfecto giro de los engranes, ya que si no se hubiera investigado correríamos el riesgo de que no giren bien o podrían existir complicaciones a la hora de la rotación (que no embonen por la diferencia de los dientes o en el giro debido a un mal planteamiento de circunferencias primitivas (figura 2.5), no permita la correcta rotación de los engranes en la trayectoria de la recta ab). En el caso de las cremalleras es similar ya que es considerada un engrane con una circunferencia infinita eso facilita los cálculos matemáticos y debemos considerar los mismos riesgos que el engrane normal con sus respectivas excepciones previas.

Por último realizamos el diseño de camisas y soportes, estas dan gran utilidad ya que son la estructura que mantiene cada engrane o cremallera en su lugar, fueron calculadas para el modelo, específicamente para esta instalación experimental.

En el desarrollo del subsistema de control mencionamos que es un servomotor y como está compuesto, donde para poderlo ocupar se utilizó una etapa de potencia para alimentar a los servomotores. Para el movimiento de los servomotores fue requerido un Arduino para la manipulación de los servos y darles la posición exacta a través de un PWM y así poder girar los engranes a la posición requerida.

En el desarrollo de la interfaz de usuario fue basada en MATLAB, donde ha mostrado una interfaz de usuario final (Alumno) y sea desde aquí donde finalmente opere la instalación experimental, el programa de Matlab manda los valores necesarios al Arduino para que este pueda girar las piezas en la posición por el alumno.

## CAPITULO 3 Resultados

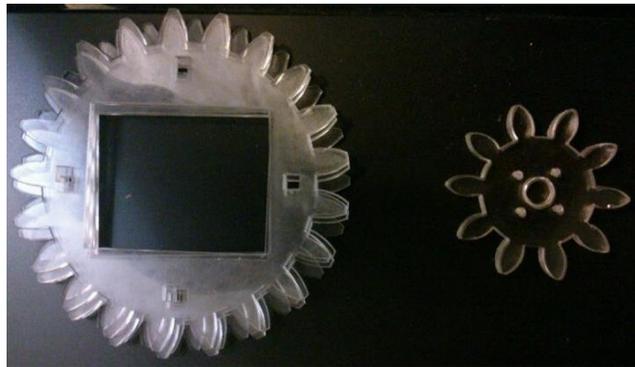
---

En este capítulo se describe el diseño final de las piezas, una vez conjuntado la instalación experimental y analizaremos el comportamiento de cada una de las piezas diseñadas. También mostraremos imágenes de las piezas finales (físicas), de las etapas de control y de la interfaz de usuario, para hacer pruebas finales.

### 3.1 Subsistema mecánico

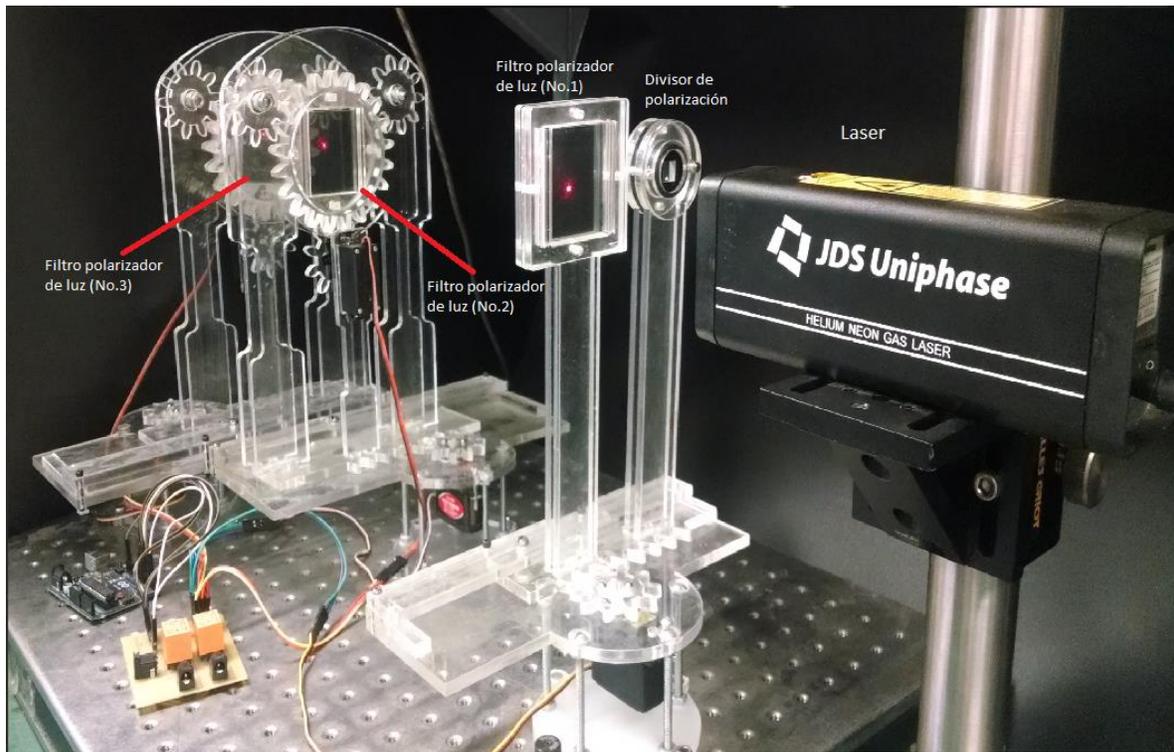
En el subsistema mecánico mostramos todas las partes juntas para el armado del sistema, empezamos por los engranes.

Los engranes fueron hechos con los planos mostrados en el anexo C en piezas de acrílico de 6 [mm] de grosor para el engrane pequeño y para el engrane grande fueron 3 piezas de 2[mm] para ensamblar un engrane de 6 [mm] de grosor en total, estos fueron hechos con una maquina cortadora a láser.



**Figura 3.1** Engranes cortados con láser.

En la figura 3.1 se muestran los primeros engranes para la estructura mecánica, así mismo las demás piezas fueron hechas a corte a láser con sus respectivos cálculos, a continuación las piezas finales a láser y una foto final de la instalación mostrada en la figura 3.2.

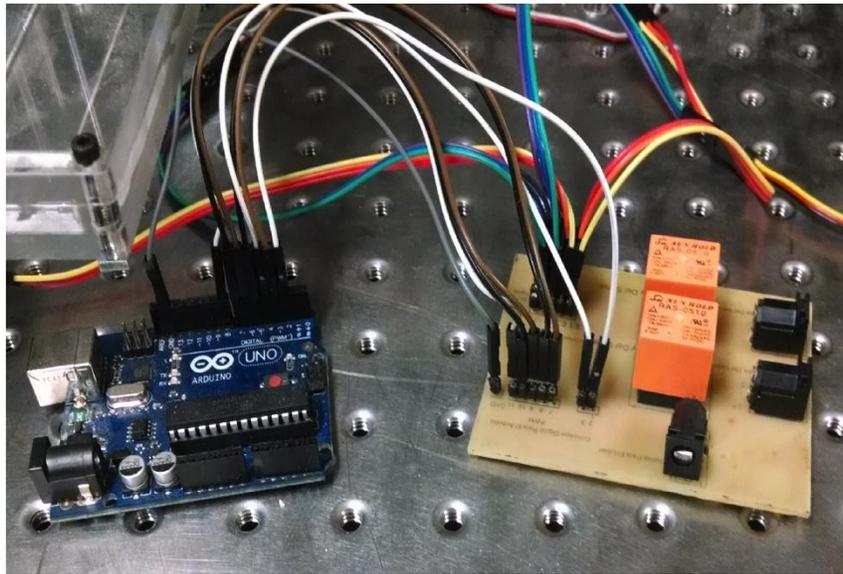


**Figura 3.2** Imagen del sistema experimental. En la figura el láser emite un haz de luz no polarizado a través del primer filtro polarizador de luz, la luz resultante que atravesó el primer filtro incidirá el segundo y el tercer filtro.

## 3.2 Subsistema de control

El subsistema de control se divide en la etapa de potencia, el control de servomotores, el encendido y apagado del láser, la programación del microprocesador Arduino Uno y por último la programación de la interfaz gráfica de usuario.

Para la parte del control de servos el encendido y apagado del láser se realizó un circuito en fenólica para tener concentrado toda la etapa de control mostrada en la figura 3.3.



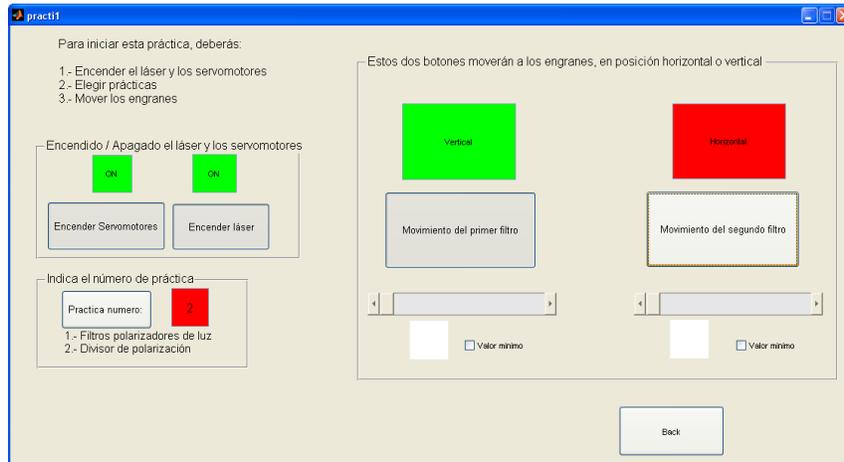
**Figura 3.3** Arduino y Tarjeta de control de los servomotores y laser.

También podemos observar la tarjeta Arduino UNO (figura 3.4), ocupada para la programación de los servomotores y para el encendido /apagado del láser a través del PWN, donde las señales de pulsos de diferente tamaños por la tarjeta del Arduino.



**Figura 3.4** Arduino UNO[17].

La figura 3.5 muestra la interfaz gráfica de usuario con la que se manipulará todo el sistema mecánico y el sistema óptico a través el subsistema de control.



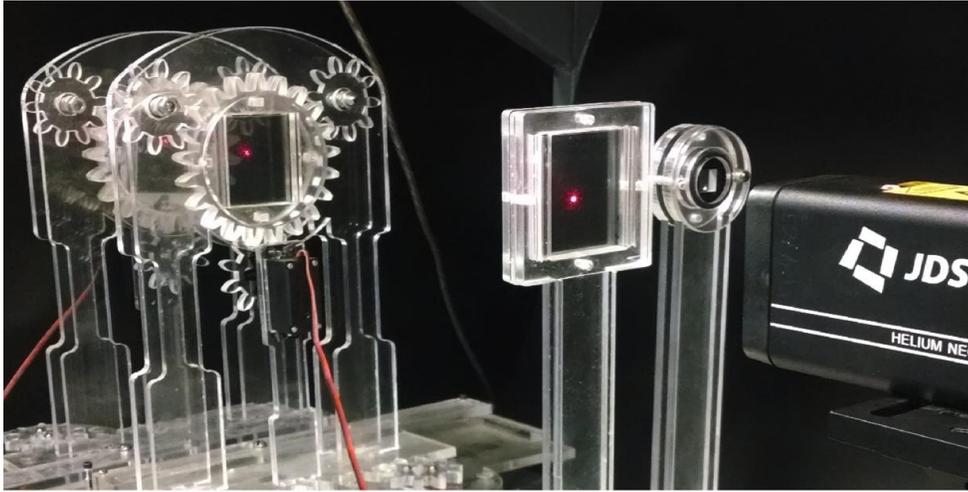
**Figura 3.5** interfaz gráfica de usuario en operación [15].

Por último esta la interfaz gráfica de usuario realizada en Matlab, que ha facilitado la interacción del usuario y la instalación. Donde se ha modificado a través de los botones el estado de polarización de la luz.

### **3.3 Subsistema Óptico**

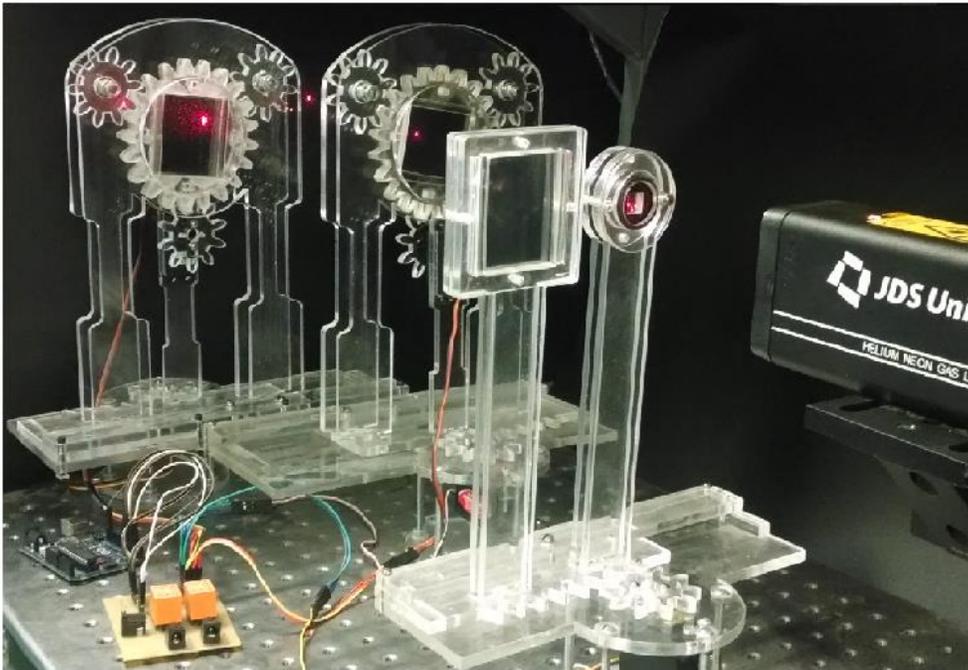
El sistema óptico está dado por un láser de haz no polarizado marca JDSU de gas HeNe modelo 34-891 que inciden en un punto, puede ser en un polarizador de haz de luz marca ThorLabs modelo LPVISE2x2 o en un divisor de polarización marca ThorLabs modelo Wollaston - WP10.

Cuando incide en un polarizador de haz de luz, las estructuras del engrane uno y el engrane dos están alineados en una posición de línea recta donde el alumno girará de acuerdo a las instrucciones del profesor los engranes 1 y 2 (donde cada engrane contiene un filtro polarizador de la luz), observando los efectos dependiendo el estados de polarización como se muestra en la figura 3.6.



**Figura 3.6** un haz de luz no polarizada fue incidida en tres filtro polarizador lineal donde el primer filtro es fijo y el segundo y el tercero pueden ser rotados a través de los engranes.

Cuando incide en un divisor de polarización de la luz separa la componente electromagnética en dos haces en un ángulo de 20 grados, las estructura del engrane uno y dos se alinean en posición abierta (se deslizan las cremalleras para que el haz de luz pase por los filtros polarizadores que están contenidos en el engrane uno y dos respectivamente, donde existe una distancia de 90 [mm] entre ellos), observando los efectos dependiendo los estados de polarización. Una representación gráfica la podemos observar en la figura 3.7.



**Figura 3.7** Haz de luz no polarizada fue incidida en el divisor de polarización y a la salida del divisor cada componente de luz fue incidida en un filtros de polarización de luz respectivamente para dejar pasar el haz de luz o no.

### 3.4 Pruebas de Funcionamiento de la Instalación

Las pruebas del funcionamiento fueron realizadas en el Laboratorio de Láseres del Departamento de Telecomunicaciones. La estructura se ha montado en una mesa óptica para ver el funcionamiento de la práctica y se comprobó el comportamiento de todas las piezas, junto con la parte de control y la interfaz de usuario.

En la primera parte observamos los efectos de los filtros polarizadores lineales sobre la luz no polarizada que consistió en hacer incidir un haz de luz no polarizada por tres filtros de polarización de luz lineales, el primer filtro de polarización esta fijo y de forma vertical en el soporte fijo, este filtro solo dejo pasar un estado de polarización (vertical). Los dos filtros de polarización restantes fueron colocados en piezas móviles, los cuales fueron rotados para que los filtros tuvieran movimientos hacia la derecha o izquierda dependiendo de los cambios realizados en el laboratorio a través del programa realizado.

En la figura 3.8 se muestra el esquema para ejemplificar lo descrito anteriormente:

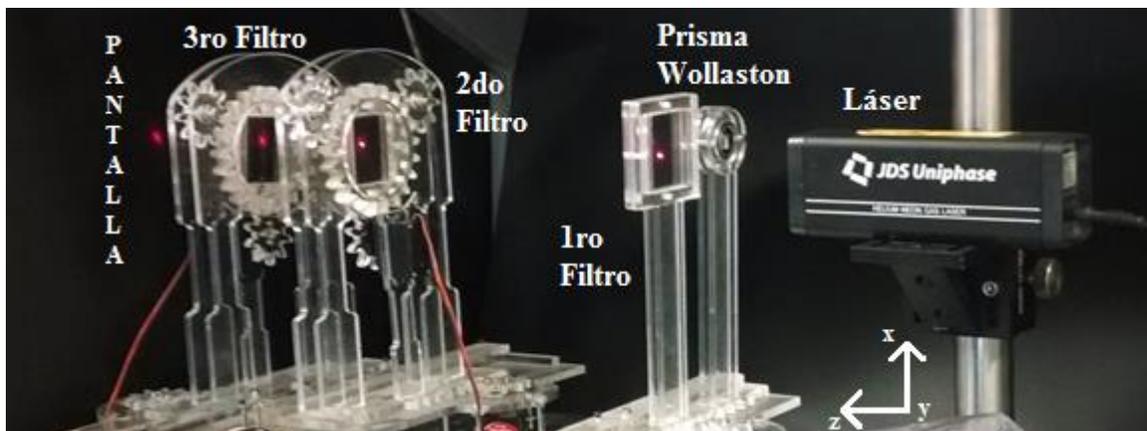


Figura 3.8 Un haz de luz emitido por un laser inside sobre tres filtros polarizadores de luz y la resultante es mostrada sobre una pantalla.

Una vez que el haz de luz atravesó el primer filtro de polarización (fig. 3.8) solo tendremos una componente vertical, el haz resultante del primer filtro atravesó el segundo y tercer filtro que estaba colocados también en forma vertical y así el resultado se ve en la pantalla final.

En la segunda parte se comprobó con un prisma divisor de la polarización que la luz no polarizada puede ser descompuesta en dos componentes de polarización lineal ortogonales entre sí. Esta consta de un divisor de polarización llamado prisma Wollaston, donde el haz de luz del láser incidió sobre el prisma Wollaston, el cual dividió el haz en sus dos componentes (una es la horizontal y la otra es la vertical), donde cada haz de luz es dirigido

a un filtro polarizador, los filtros se rotan en forma vertical u horizontal de forma independiente.

La figura 3.9 muestra el filtrado de los estados de polarización por un prisma de Wollaston:

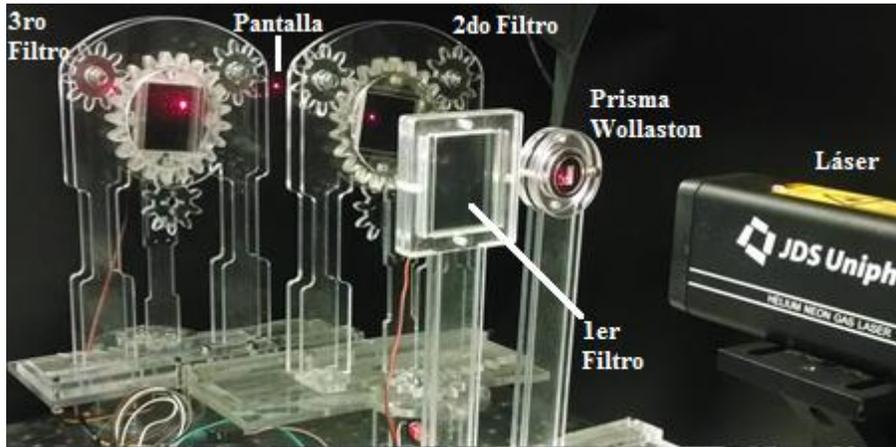


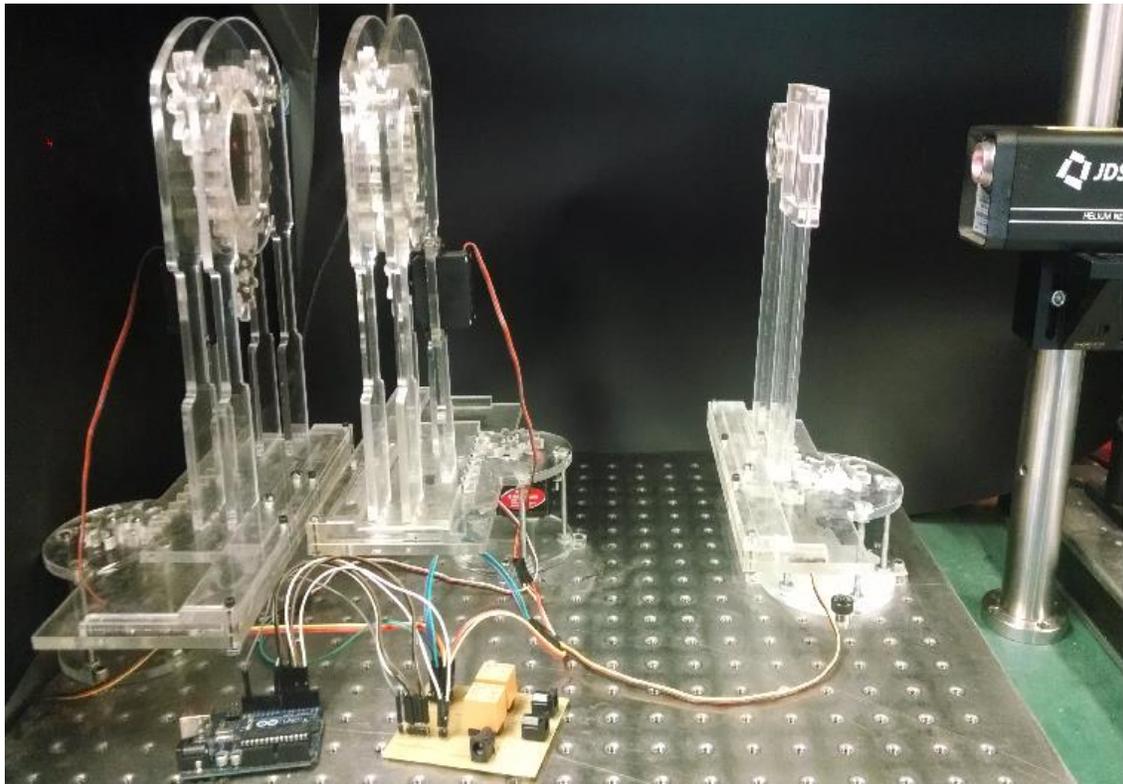
Figura 3.9 Un haz de luz no polarizado incidio sobre un prisma Wollaston, sus resultantes fueron incididas sobre un filtro polarización respectivamente y mostradas en una pantalla [13].

El láser posicionado en la parte derecha emitió un haz de luz no polarizada y al pasar el prisma Wollaston, este dividió las componentes de la onda, el resultado fueron dos haces de luz polarizados linealmente ortogonales entre sí, los dos haces de luz tuvieron una abertura de 20 grados debido a que es el rango que nos brinda el prisma Wollaston que utilizamos.

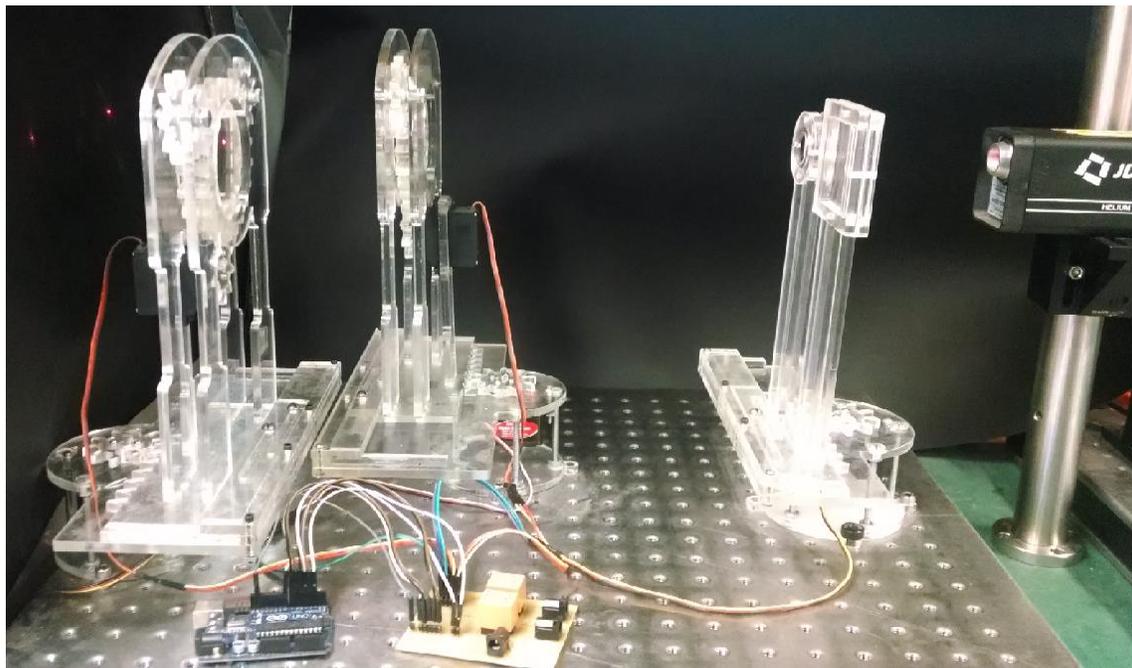
Donde observamos que al mover los engranes hay una oportunidad de mejora debido a que son pocos dientes, ya que los engranes tienen un error de 6 a 8 grados al mover un engrane a otro, esto se vuelve impreciso al realizar el movimiento de los engranes grado por grado a través de los servomotores, sin embargo las posiciones 0, 45 y 90 grados son óptimas y cumple con la práctica, es decir, logrando dejar pasar o no el haz de luz del láser

El encendido del láser a través de la interfaz gráfica de usuario funciona bien al mandarle el pulso de encendido o apagado respectivamente.

El giro de los engranes para las cremalleras corren óptimamente, deslizando las camisas correctamente. Podemos observar esta descripción en la figura 3.10 y 3.11



**Figura 3.10** Un haz de luz no polarizada fue incidida en tres filtro polarizador lineal donde el primer filtro es fijo y el segundo y el tercero pueden ser rotados a través de los engranes.



**Figura 3.11** haz de luz no polarizada fue incidida en el divisor de polarización y a la salida del divisor cada componente de luz fue incidida en un filtros de polarización de luz respectivamente para dejar pasar el haz de luz o no.

### **3.5 Conclusiones**

En este capítulo se mostraron los resultados finales de las piezas en acrílico y el armado de cada subsistema para completar el sistema y hacer las pruebas de manipulación de engranes cremalleras y del láser.

En esta parte fue la más complicada ya que al juntar todas las piezas se ajustaron o modificaron muchas cosas debido a que la programación no hacía lo que deseaba para la posición de los engranes o de las cremalleras, también hacer girar grado por grado los engranes fue otro reto interesante ya que los servos demandaban más corriente, por eso fue necesario ponerle una etapa de potencia.

Fue necesario poner dos baleros en los ejes de soporte de los engranes mostrados en la anexo C figura c.1, para el soporte del engrane uno y dos, para que girara mejor y no se atorara al girar los engranes (utilice también un poco de aceite para los baleros para lubricarlos igual que en la base de las cremalleras para que resbalaran mucho mejor y así evitar fricción)

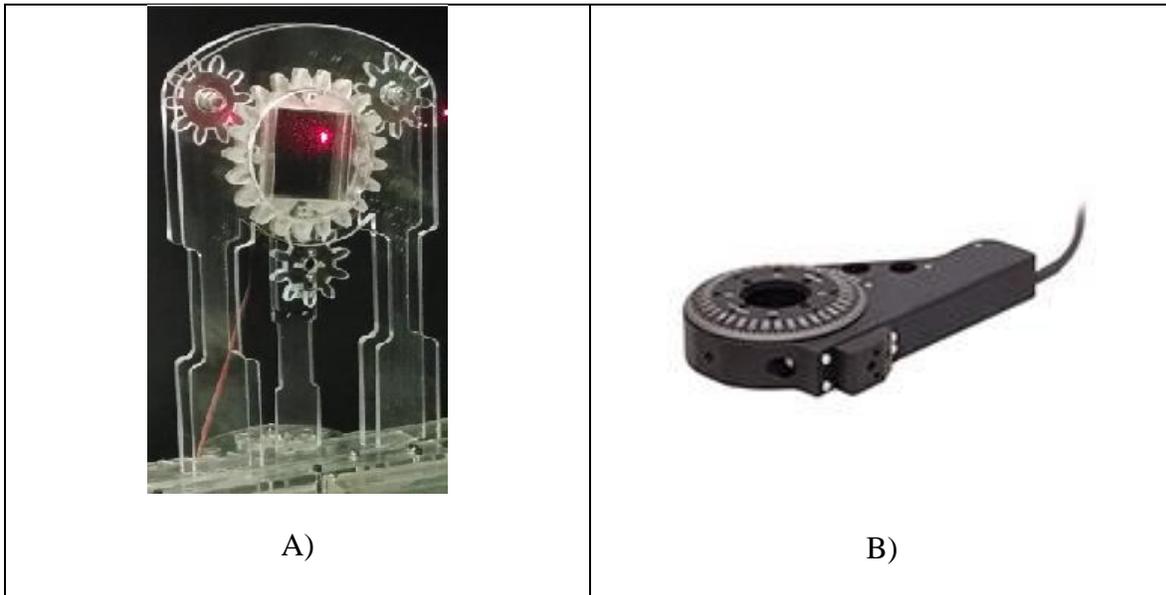
### **Propuesta de mejora**

---

La instalación experimental propuesta anteriormente nos sirvió para comprobar el funcionamiento de la práctica, sin embargo tiene sus limitaciones, entre ellas el movimiento de los filtros soportados en los engranes grado por grado. Además durante la implementación de la práctica se observó que el acrílico no es el mejor material para el subsistema mecánico ya que durante las pruebas de la práctica los soportes se rompieron, los engranes se salían de los soportes entre otras fallas. Por lo anterior se propone que el subsistema mecánico sea sustituido por piezas de materiales más resistentes y comerciales.

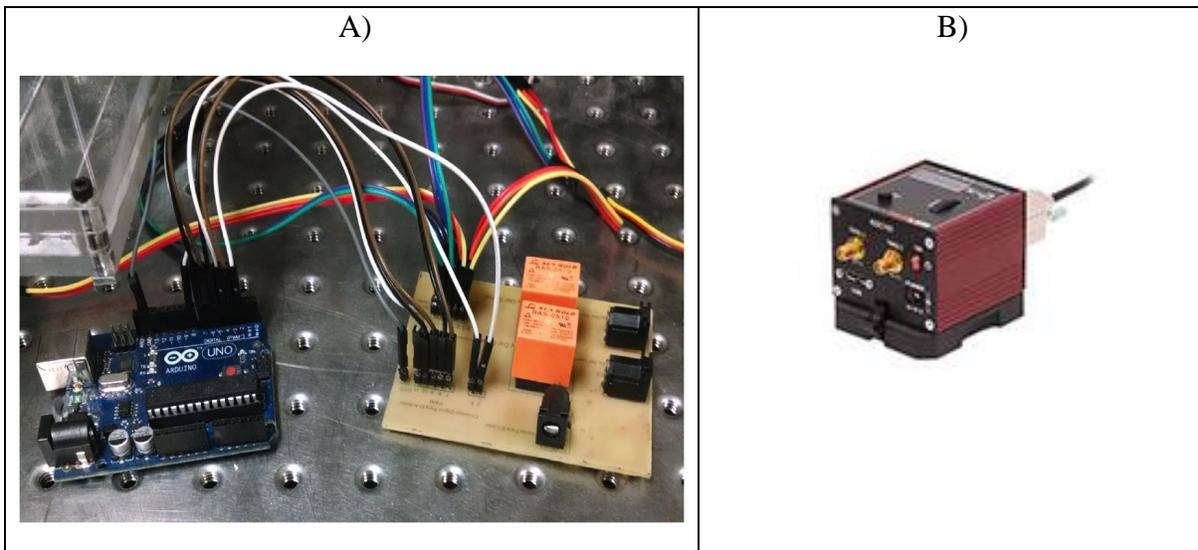
El producto es del fabricante Thorlabs. Es un fabricante con herramientas y sistemas fotónicos, que tiene productos que van desde óptica y componentes de posicionamiento opto-mecánico a los sistemas de imágenes ópticas sofisticadas.

Usando piezas del fabricante Thorlabs seleccionamos un par de productos para reemplazar las piezas desarrolladas anteriormente. La primera pieza reemplazada está mostrada en la (figura 4,1 A); será la pieza que sostiene el filtro polarizador de luz que puede ser girada a través de los engranes por una pieza motorizada (figura 4.1 B) que contiene un filtro polarizado en el centro girada por un motor de precisión.



**Figura 4.1** A) pieza que sostiene un filtro polarizador de luz con engranes y camisas de soporte , B) modelo del producto es PRM1Z8 del fabricante Thorlabs[20].

El segundo producto que puede ser sustituido mostrado en la figura 4.2, es la parte de potencia de los motores junto con la codificación el Arduino que realice el control de los motores para cambiar la posición de los engranes que contiene los filtro polarizadores de luz por el productor llamado KPRM1E que contiene la potencia para poder girar el dispositivo mostrado en al figura 4.1 B), que lleva una señal codificada para poder girar el filtro polarizador del centro del dispositivo.



**Figura 4.2** A) Arduino y circuito de potencia que controlan, corriente y señal codificada de lo servomotores , B) producto con nombre KPRM1E del fabricante Thorlabs[20].

Para entrar un poco más en detalle del dispositivo que sustituirá las piezas desarrolladas anteriormente éstas son:

KPRM1E – Modelo motorizado de precisión con etapa de rotación incluida y controladores de servomotor (DC) y fuente de poder mostrada en la figura 4.3.



**Figura 4.3** producto PRM1Z8 que contiene un filtro polarizador de luz y un servomotor (DC) y producto KPRM1E que contiene una fuente de poder y una tarjeta para controlar a los servomotores [20].

## Producto KPRM1E

### Características

- Movimiento gradual de  $1^\circ$
- Velocidad de rotación de 25 deg/sec
- Compact Design is Only 23 mm Thick
- Contiene un limite de cambio de inicio
- Base montable
- Compatible con otros accesorios (SM1)

Recomendación de controladores: [KDC101 DC Servo Controller](#)

La PRM1z8 mostrado en la figura 4.4, es pequeño, compacto, con un servomotor de corriente directa que tiene una rotación 360 grados y soporta movimientos de grado por grado donde el alumno medirá el desplazamiento angular con las marcas de graduación que están marcadas en la placa que gira en incrementos de 1 grado. Esta etapa de rotación o montaje está también equipado con un servomotor para facilitar la rotación automática, lo que permite x posicionamiento angular a partir del servomotor y permitir la rotación continua, a través de múltiples ciclos.



**Figura 4.4** producto PRM1Z8 que contiene filtro paralizador en el centro. [21].

El producto KPRM1E mostrado en la figura 4.5 incluye un manejador para el dispositivo de rotaciones PRM1Z8 monitoreado con un controlador. Este controlador es un compañero ideal para un movimiento suave y continuo que puede ser automatizado a través de la interfaz del software. Los productos contiene un paquete completo con una fuente de alimentación, que incluye un adaptador específico.



**Figura 4.5** modelo del producto KPRM1E, muestra la fuente de poder para alimentar de corriente y señal para la pieza PRM1Z8. [22]

## Conclusiones Generales

---

Como resultado de la investigación y en un primer acercamiento para una práctica experimental del estudio de la polarización de la luz. Se desarrolló e implemento una instalación experimental para una práctica de laboratorio de la asignatura “Dispositivos Ópticos”.

En general el desarrollo de esta práctica cumplió las necesidades básicas para el aprendizaje de la polarización de la luz. Trabajando con algún material que contaba el laboratorio y ajustándose a las necesidad de la práctica, se desarrollaron a bajo costo pero de manera funcional piezas que facilitarán la manipulación, cómo engranes, cremalleras y soportes, acompañados de servomotores y circuitos de potencia, para que los filtros polarizadores de luz sea rotados a través de la estructura mecánica sin que los alumnos tengan que tocarlos. Otro punto importante de rescatar es la interfaz de usuario que se encarga de mover los filtros, ya que facilitó mucho el manejo de la práctica para que el alumno pueda modificar la posición de los filtros y del divisor de polarización.

Comparando esta instalación con una instalación del mercado, es mucho más económica, debido a que los materiales ocupados son de bajo costo (acrílico, Arduino, cargador, servomotores) y otros materiales. Algunos de estos materiales ya existían en el laboratorio de “Dispositivos Ópticos”, como los filtros de polarización de la luz, el divisor de polarización, el láser de haz no polarizada, pero no es tan preciso como las piezas que están el mercado. A pesar que no es tan preciso cumple con las necesidades básicas del proyecto para que el alumno pueda aprender el conocimiento básico de la práctica desarrollada en el laboratorio.

## Referencias

---

### a) Libros

- [1] Eugene Hecht, *Optics*, 4th ed. Addison Welseley, 2002, pp. 325-328.
- [2] David S. Kliger, James W. Lewis and Cora E., *Polarized Light in Optics and Spectioscopy*, Rondall, Academic Press, 1990.
- [3] Stephen Gulmand and David Elmore, *Polarizing views*, Oemagazine, Mayo 2004.
- [4] Michael Bass, Virendra N. Mahajan, *Geometrical and Optic, Polarized Light componants and Instruments*, third edition, McGraw Hill, 2010.
- [5] Richard G. Budynas y J. Keith Nisbett, *Diseño en ingeniería mecánica*, octava edición McGraw Hill.
- [6] Robert L. Mott, P.E, *Diseño de elementos mecánicos I*, cuarta edición, Pearson Educación, 2006.
- [7] Joseph Edward Shigley, John Joseph Uicker, Jr *Teoría de máquinas y mecanismos*, octava edición McGraw Hill.

### b) Manuales de Practicas

- [8] Practica de física Recreativa, Universidad de Castilla-La Macha, *Polarización lineal, Verificación experimental de la ley de Malus*, 2007 [OnLine]. Available: <https://www.uclm.es/profesorado/ajbarbero/CursoAB2007/Malus2003.pdf>
- [9] Practicas de Electromagnetismo y Óptica, Universidad de Almería, *Polarización de Onda Electromagnéticas planas*, curso 2013-2014 [OnLine]. Available: <http://www.ual.es/~migarca/polarizacion.pdf>

### c) Internet

- [10] Esta imagen fue desarrollada con Lnkcape, editada en el programa de código abierto Lnscape, february 2007.
- [11] Medicion de intensidad de luz polarizada, Valencia, *Intensidad de Luz polarización*, [OnLine]. Available: <http://intercentres.edu.gva.es/iesleonardodavinci/Fisica/Luz/Polarizacion/polarizacion.htm>

[12] Intituto Nacional de tecnologías eductavias y de información del profesorado, *Mecanismos*, 2006.

**d) Programas**

[13] Esta imagen fue desarrollada con AutoCad LT 2013, Derechos Reservados por Autodesk, Inc. All rights reserved.

[14] Esta Imagen fue desarrollada con Solid PREMIUM 2012, derechos reservados del 1995-2011 Dasault Systems.

[15] Esta Imagen fue desarrollada con MATLAB 7.10.0 (R2010a), derechos reservados del 1984-2010 The MathWorks, INC.

[16] Esta Imagen fue desarrollada con EAGLE, derechos reservados del 1988-2010 CadSoft All rights reserved worldwide.

**e) Mesografía**

[17] Construcción de un brazo robótico, <http://brazorobotipn.blogspot.mx/>

[18] Configuración de un servomotor en Arduino, robótico, [http://electrongeeks.blogspot.mx/2014\\_02\\_01\\_archive.html](http://electrongeeks.blogspot.mx/2014_02_01_archive.html)

[19] Características de la pieza PRM1E, Thorlabs, [http://www.thorlabs.com/newgrouppage9.cfm?objectgroup\\_id=2875&pn=KPRM1E](http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2875&pn=KPRM1E)

[20] Características de la pieza PRM1Z8, Thorlabs, <http://www.thorlabs.com/thorproduct.cfm?partnumber=PRM1Z8>

[21] Características de la pieza KDC101, Thorlabs, <http://www.thorlabs.com/thorproduct.cfm?partnumber=KDC101>



Engrane Grande	Engrane Pequeño
$M = \frac{D}{N} = \frac{89.25}{21} = 4.25$	$M = \frac{D}{N} = \frac{42.5}{10} = 4.25$
$\Phi = 20^\circ$	$\Phi = 20^\circ$

La altura de cabeza o addendum es la distancia que sobresale el diente a partir de la circunferencia primitiva. La altura de la base o addendum es la distancia desde la circunferencia primitiva a la base o pie del diente y es igual a la altura de cabeza más una cantidad extra para el “juego”. Estas distancias corrientemente están normalizadas y depende del sistema de dientes que se utilice. Se pueden dibujar para cada engranaje las circunferencias de cabeza y de base o addendum y dedendum (fig. 5.3)

El último paso es marcar los anchos del hueco y espesores de dientes sobre cada circunferencia primitiva. Se construyen iguales entre si y su valor será la mitad del paso circunferencial. Utilizados plantillas que se ajusten a las curvas y con centro en  $O_1$  y  $O_2$ , pueden construirse ya los perfiles de los dientes.

El radio de acuerdo en la base del diente es generalmente igual al juego o ligeramente mayor.

Para calcular la circunferencia de cabeza o circunferencia exteriores es de la siguiente manera:

$$D_E = D + (2 * M)$$

En donde  $D_E$  = diámetro de cabeza, en [mm]

$M$  = módulo

$D$  = diámetro de la circunferencia primitiva, en [mm]

Engrane Grande	Engrane Pequeño

$D_E = D + (2 * M)$ $D_E = 89.25 + (2 * 4.25)$ $D_E = 89.25 + 8.5$ $D_E = 97.75$	$D_E = D + (2 * M)$ $D_E = 42.5 + (2 * 4.25)$ $D_E = 42.5 + 8.5$ $D_E = 51.0$
--	---

Para calcular la circunferencia de Base o circunferencia Interior se utiliza la siguiente ecuación:

$$D_i = D - (2 * 1.25 * M)$$

En donde  $D_i$  = diámetro de base, en [mm]

$M$  = módulo

$D$  = diámetro de la circunferencia primitiva, en [mm]

Engrane Grande	Engrane Pequeño
$D_i = D - (2 * 1.25 * M)$ $D_i = 89.25 - (2 * 1.25 * 4.25)$ $D_i = 89.25 - 10.625$ $D_i = 78.625$	$D_i = D - (2 * 1.25 * M)$ $D_i = 42.5 - (2 * 1.25 * 4.25)$ $D_i = 42.5 - 10.625$ $D_i = 31.875$

Para calcular la circunferencia Base es de la siguiente manera:

$$D_B = D * \text{Cos}\Phi$$

En donde  $D_B$  = diámetro base, en [mm]

$\Phi$  = ángulo

$D$  = diámetro de la circunferencia primitiva, en [mm]

Engrane Grande	Engrane Pequeño

$D_B = D * \text{Cos}\Phi$ $D_B = 89.25 * \text{Cos}(20^\circ)$ $D_B = 83.86756642$	$D_B = D * \text{Cos}\Phi$ $D_B = 42.55 * \text{Cos}(20^\circ)$ $D_B = 39.98392101$
---	---

Para calcular la altura del diente, cabeza del diente o addendum y pie del diente o dedendum es:

$$A_d = 2.25 * M * 2$$

En donde  $A_d$  = Altura de diente, en [mm]

$M$  = módulo

Engrane Grande	Engrane Pequeño
Altura de diente	Altura de diente
$A_d = 2.25 * M * 2$ $A_d = 2.25 * 4.25 * 2$ $A_d = 19.125$	$A_d = 2.25 * M * 2$ $A_d = 2.25 * 4.25 * 2$ $A_d = 19.125$
Cabeza del diente (addendum)	Cabeza del diente (addendum)
$D_E - D = 8.5$	$D_E - D = 8.5$
Pie del diente (dedendum)	Pie del diente (dedendum)
$D - D_i = 10.625$	$D - D_i = 10.625$

### b) Diseño de cremalleras

El modulo es, el número de milímetros que representan cada diente en el diámetro primitivo, es el paso medio sobre el diámetro y por ella es llama paso diametral.

$$M = \frac{D}{N}$$

En donde: M = módulo

D = diámetro de la circunferencia primitiva, en [mm]

N = números de dientes

Cremallera	Piñón
$M = \frac{D}{N} = \frac{38.25}{9} = 4.25$	$M = \frac{D}{N} = \frac{42.5}{10} = 4.25$
$\Phi = 20^\circ$	$\Phi = 20^\circ$

Para la implementación de la cremallera la tomaremos como un engrane infinito partiendo de ese punto, la cremallera se calculara como un engrane normal para determinar los parámetros de:

- Diámetro primitivo
- Diámetro externo
- Diámetro interior
- Circulo base
- Altura del diente (addendum)
- Cabeza del diente (dedendum)

Usando la ecuación de arriba :

Cremallera	Piñón
$D_E = D + (2 * M)$	$D_E = D + (2 * M)$
$D_E = 38.25 + (2 * 4.25)$	$D_E = 42.5 + (2 * 4.25)$
$D_E = 38.25 + 8.5$	$D_E = 42.5 + 8.5$
$D_E = 46.75$	$D_E = 51.0$

Para calcular la circunferencia de Base o circunferencia Interior es:

$$D_i = D - (2 * 1.25 * M)$$

En donde  $D_i$  = diámetro de base, en [mm]

$M$  = módulo

$D$  = diámetro de la circunferencia primitiva, en [mm]

Cremallera	Piñón
$D_i = D - (2 * 1.25 * M)$	$D_i = D - (2 * 1.25 * M)$
$D_i = 38.25 - (2 * 1.25 * 4.25)$	$D_i = 42.5 - (2 * 1.25 * 4.25)$
$D_i = 38.25 - 10.625$	$D_i = 42.5 - 10.625$
$D_i = 27.625$	$D_i = 31.875$

Para calcular el circunferencia Base es:

$$D_B = D * \text{Cos}\Phi$$

En donde  $D_B$  = diámetro base, en [mm]

$\Phi$  = ángulo

$D$  = diámetro de la circunferencia primitiva, en [mm]

Cremallera	Piñón
$D_B = D * \text{Cos}\Phi$	$D_B = D * \text{Cos}\Phi$
$D_B = 38.25 * \text{Cos}(20^\circ)$	$D_B = 42.55 * \text{Cos}(20^\circ)$
$D_B = 35.94324275$	$D_B = 39.98392101$

Para calcular el Altura de diente, cabeza del diente o addendum y pie del diente o dedendum:

$$A_d = 2.25 * M * 2$$

En donde  $A_d$  = Altura de diente, en [mm]

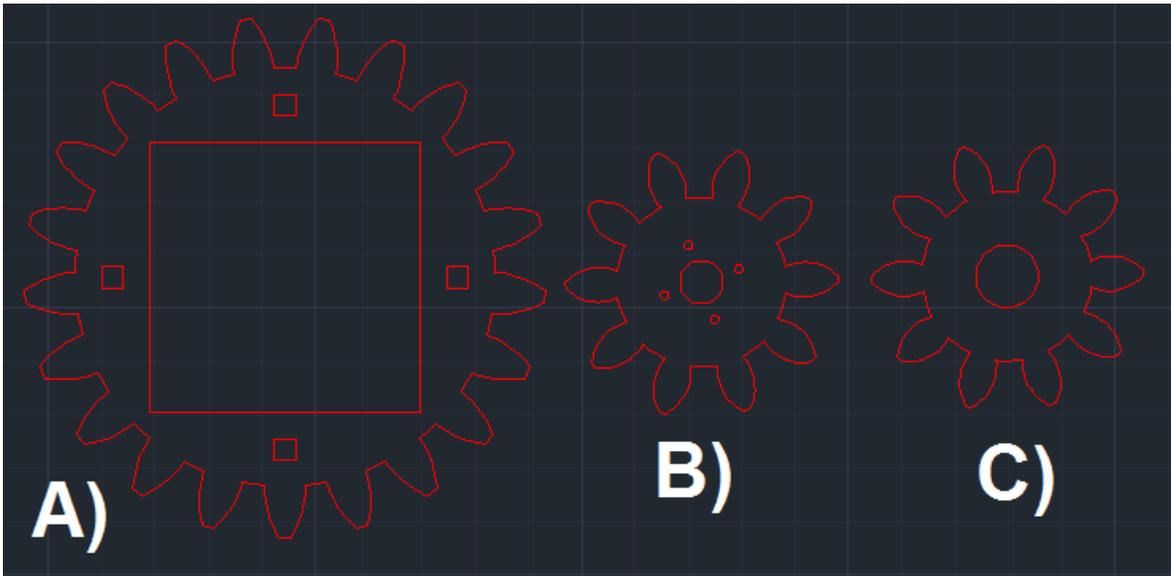
M = módulo

Cremallera	Piñón
Altura de diente	Altura de diente
$A_d = 2.25 * M * 2$ $A_d = 2.25 * 4.25 * 2$ $A_d = 19.125$	$A_d = 2.25 * M * 2$ $A_d = 2.25 * 4.25 * 2$ $A_d = 19.125$
Cabeza del diente (addendum)	Cabeza del diente (addendum)
$D_E - D = 8.5$	$D_E - D = 8.5$
Pie del diente (dedendum)	Pie del diente (dedendum)
$D - D_i = 10.625$	$D - D_i = 10.625$

### c) Desarrollo de piezas de acrílico

Para el diseño de las piezas de acrílico fue necesario ocupar dos programas de computación llamados AutoCAD y SolidWorks, estas herramientas fueron útiles para la elaboración y diseño de las piezas.

Para la realización de los engranes y cremalleras se ocupó el programa SolidWorks, en este programa se realizaron introducimos todos los cálculos teóricos para obtener los engranes y cremalleras correspondientes, cómo podemos ver en la figura c.1, vemos cómo vamos trazando las circunferencias requeridas para la obtención de los dientes y que la rotación entre los engranes sea adecuada y la línea de presión o recta de presión no se atore o genere un atasco o no embone.

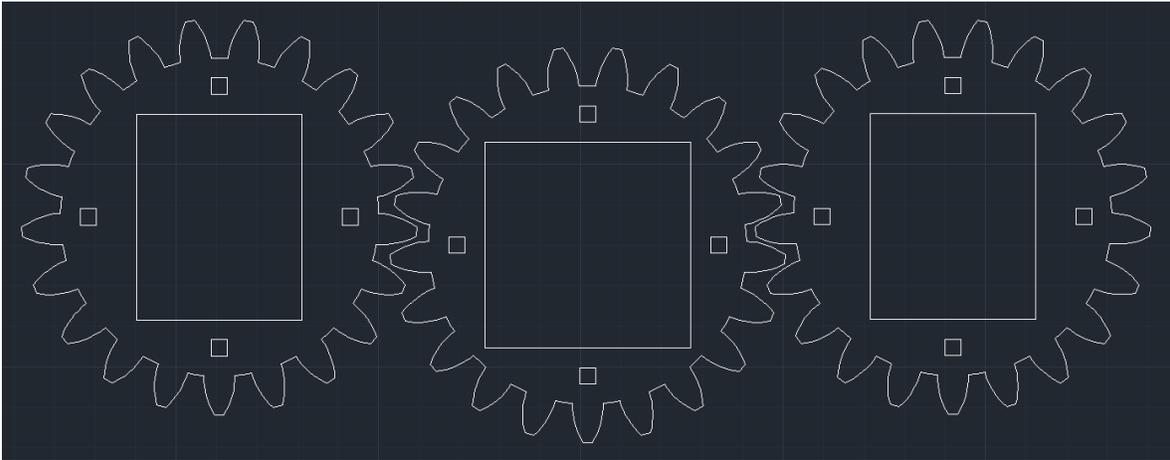


**Figura c.1** a) Engrane grande que contiene el filtro b) Engrane pequeño colocado en el servomotor c) engrane pequeño colocado en posición de piñón realizados en SolidWorks [14].

Una vez obtenidos los engranes los pasamos a un archivo de AutoCAD para hacerle modificaciones pertinentes, ya que se implementaron para funciones de diferentes tipos, para esto se desglosó en 3 engranes diferentes:

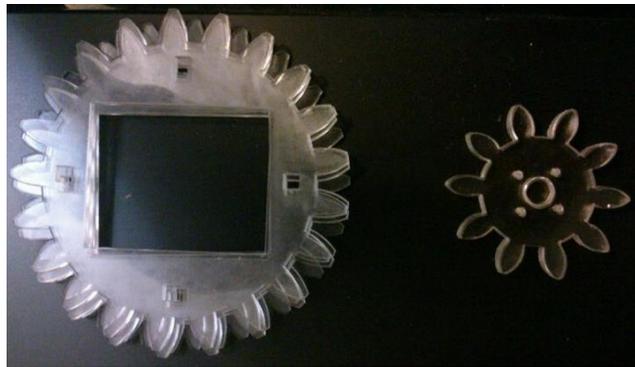
- Engrane Grande ( figura c.1 a)
- Engrane pequeño (figura c.1 b)
- Soporte de engranes (figura c.1 b)

Para el engrane grande (figura c.1 a), este contiene el filtro polarizador de luz, por lo tanto se copiará tres veces y dentro de cada uno de los engranes realizaremos un corte, un engrane se realizará un corte del tamaño del filtro del polarizador de luz posicionado en el centro del engrane y los otros dos engranes se recortarán 5 cm menos de cada lado como se muestra en la figura c.2.



**Figura c.2** Engrande realizado en AutoCAD [13].

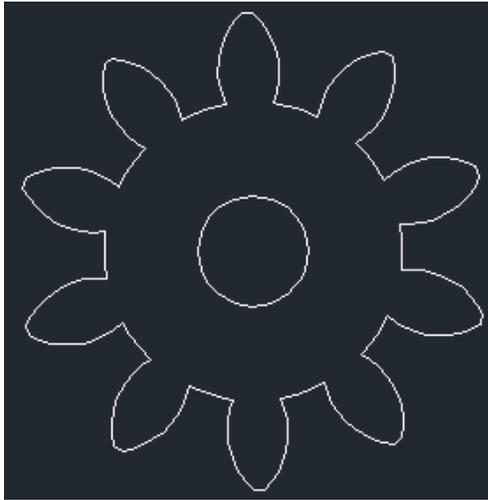
Como se ve en la figura a.2 el propósito de realizar de estos tres engranes es conjuntar uno solo pegando los tres engranes, esto se genera a través de una maquina cortadora a láser de acrílico donde cada uno de estos engranes tendrá un grueso de 2 [mm] para tener un total de 6[mm] como se muestra en la siguiente figura c.3



**Figura c.3** Engrande físico cortado a máquina laser [15].

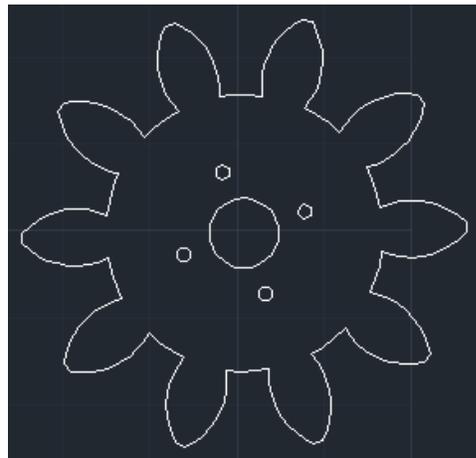
Para el engrane pequeño, este se hicieron dos pequeñas modificaciones en el interior del engrane dependiendo su función.

1. Engrane pequeño (figura c.1 c).- este engrane es de apoyo para el giro del engrane grande, en este se realizó un corte cilíndrico en el centro del engrane para colocar un balero en ese orificio, esto para facilitar el giro del engrane grande impulsado por el engrane pequeño 2, este engrane es cortado a láser en un acrílico de 6 [mm].



**Figura a.4** Engrande Diseñado en el programa AutoCAD [13].

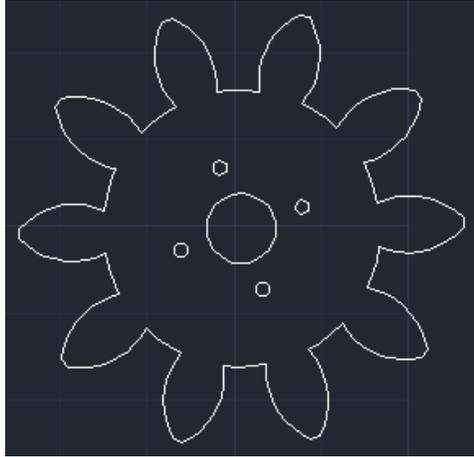
2. Engrane pequeño (figura c.1 b).- este engrane es girado por un servomotor, en este engrane se realizaron 5 cortes cilíndricos en el centro del engrane para colocar tornillos para sujetar el engrane al servomotor, para que no se que no existan inconvenientes a de rotar el engrane grande, este engrane es cortado a laser en un acrílico de 6 [mm].



**Figura c.5** Engrande Diseñado en el programa AutoCAD [13].

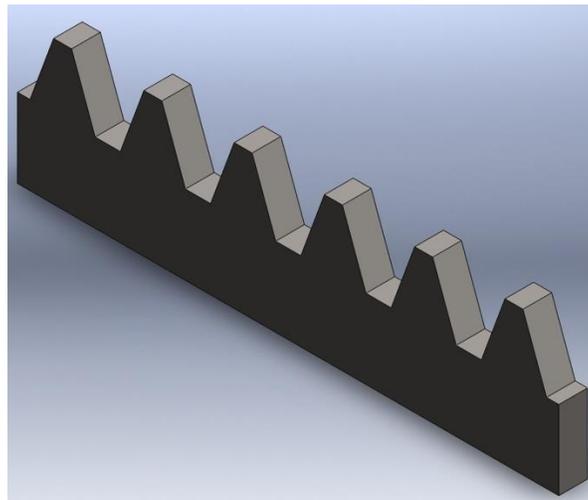
En el piñón, este engrane es girado por un servomotor, en este engrane se realizaron 5 cortes cilíndricos en el centro del engrane para colocar tornillos para sujetar el engrane al

servomotor, para que no existan inconvenientes a de rotar el engrande grande (figura c.1 a), este engrane es cortado a láser en un acrílico de 6 [mm].



**Figura c.6** Engrande Diseñado en el programa AutoCAD [13].

En la figura c.7 es el diseño de cremallera generado en SolidWorks, utilizando datos teóricos antes calculados para generar la cremallera.



**Figura c.7** cremallera realizada en SolidWorks [14].

Una vez generado la cremallera, exportamos la cremallera generada en SolidWorks a AutoCad para generar modificaciones para ser la pieza de soporte en las camisas y en las parte de soporte del primer filtro polarizador de haz de luz y divisor de polarización de haz de luz, en la figura c.8 donde la pieza amarilla es para el soporte del primer filtro polarizador de haz de luz y divisor de polarización de haz de luz, la blanca es para es para el soporte de las camisas vista en la imágenes del capítulo 5 respectivamente

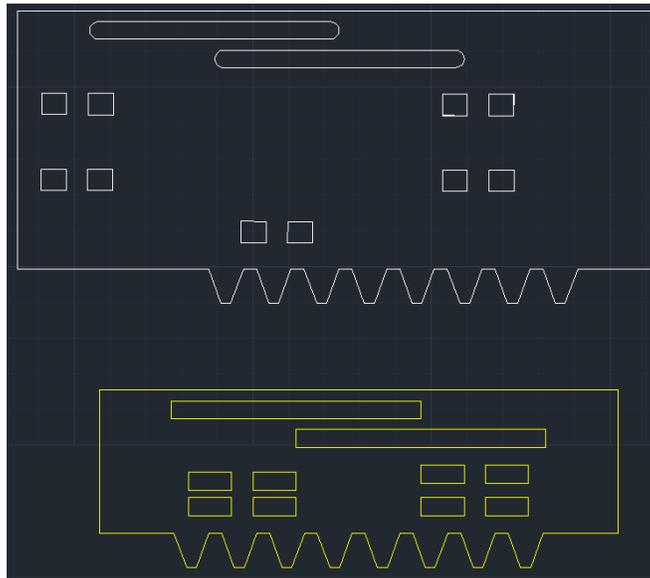


Figura c.8 cremalleras realizadas en SolidWorks modificadas en AutoCAD [13].

#### d) Medidas de las piezas desarrolladas en Matlab

Las siguientes imágenes son las medidas de todas las piezas de acrílico y están dadas en milímetros:

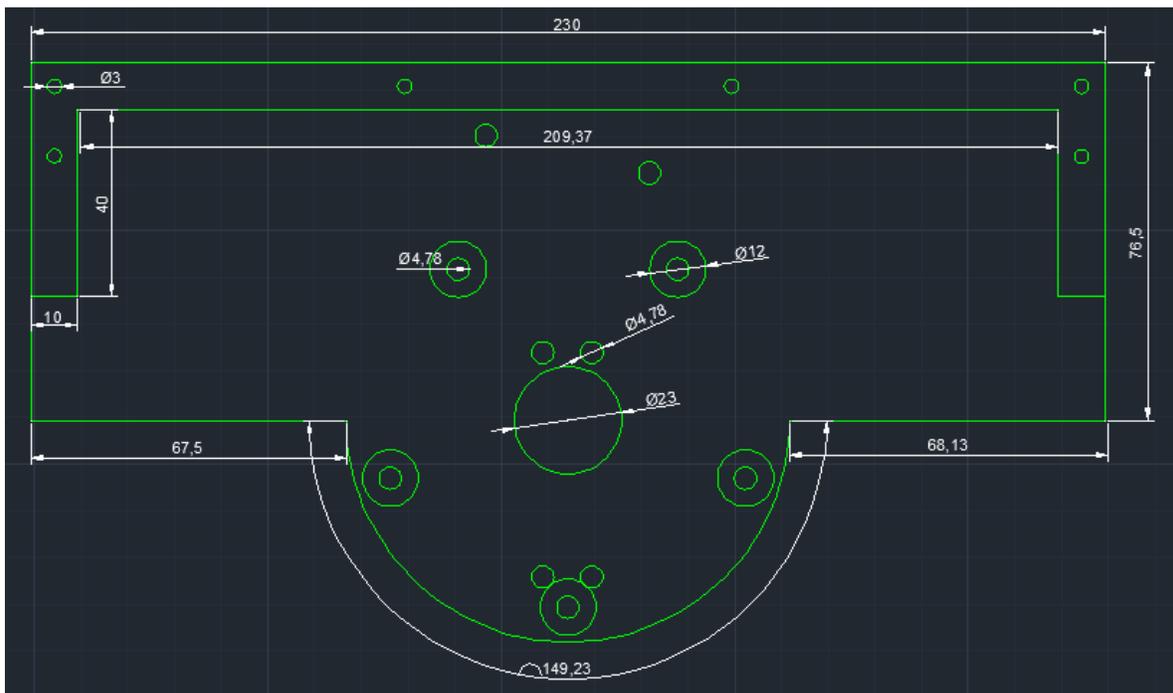


Figura d.1 base de cremallera realizado en AutoCAD [13].

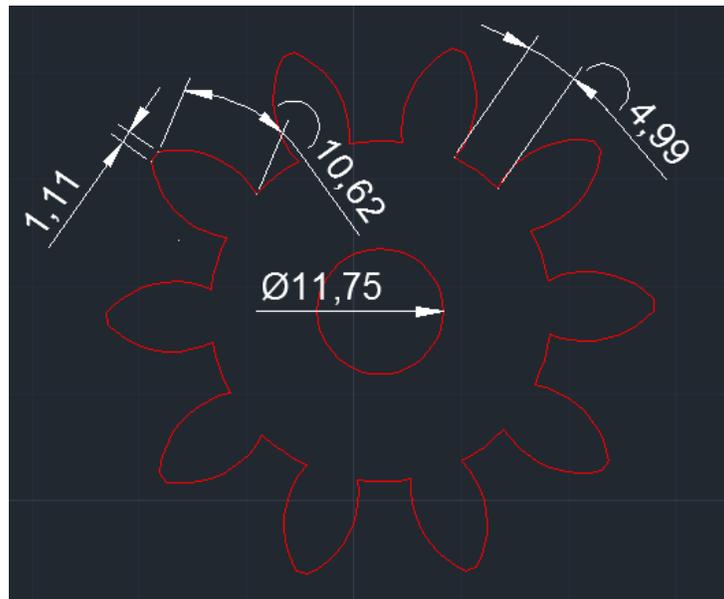


Figura d.2 engrane de apoyo realizado en AutoCAD [13].

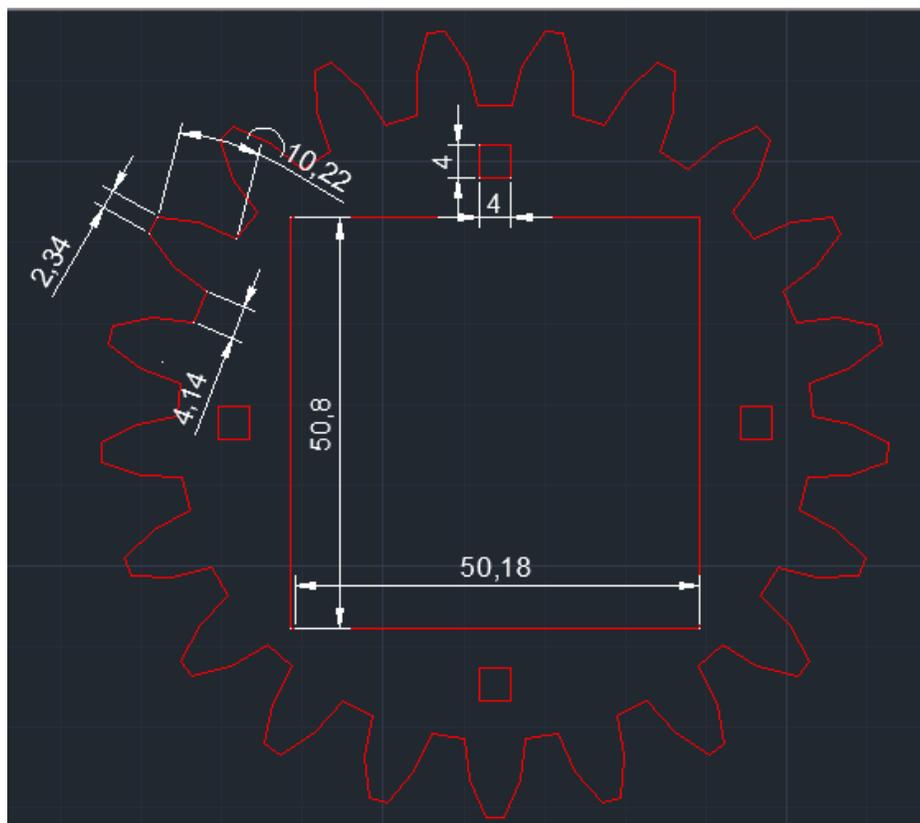


Figura d.3 engrane de apoyo realizado en AutoCAD [13].

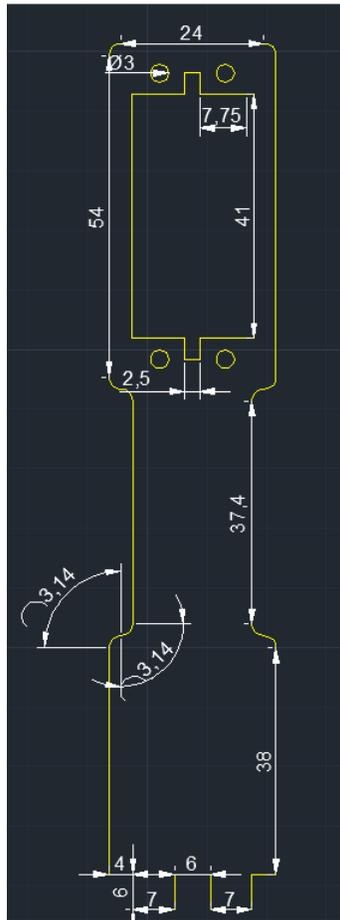


Figura d.4 soporte del servomotor que esta ensamblado a la cremallera realizado en AutoCAD [13].

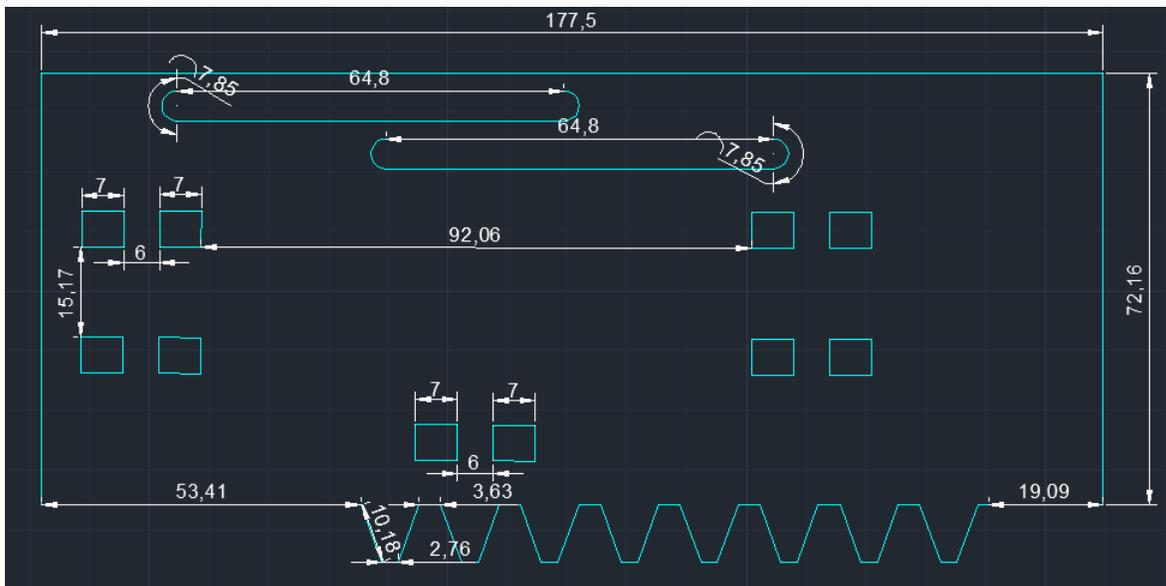


Figura d.5 cremallera que sostiene soporte de servomotor y las camisas de soporte realizado en AutoCAD [13].

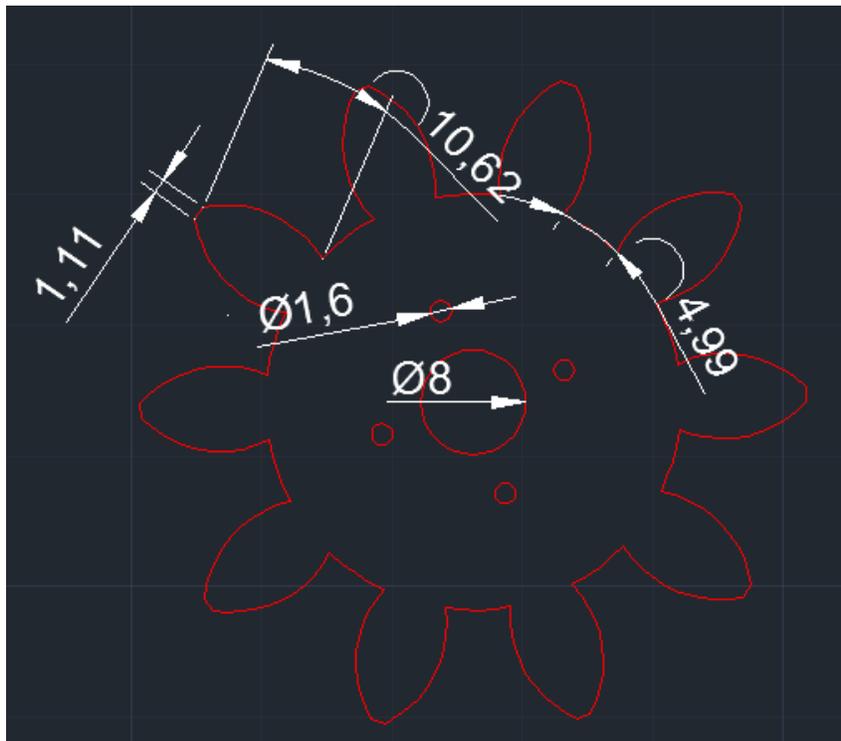


Figura d.6 engrane que hace rotar el servomotor realizado en AutoCAD [13].

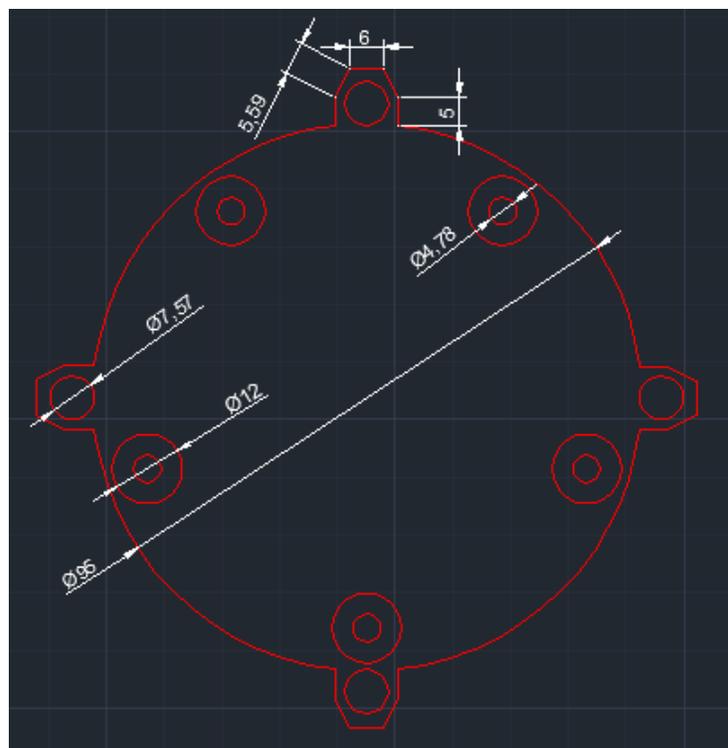


Figura d.7 base del soporte de la cremallera realizado en AutoCAD [13].

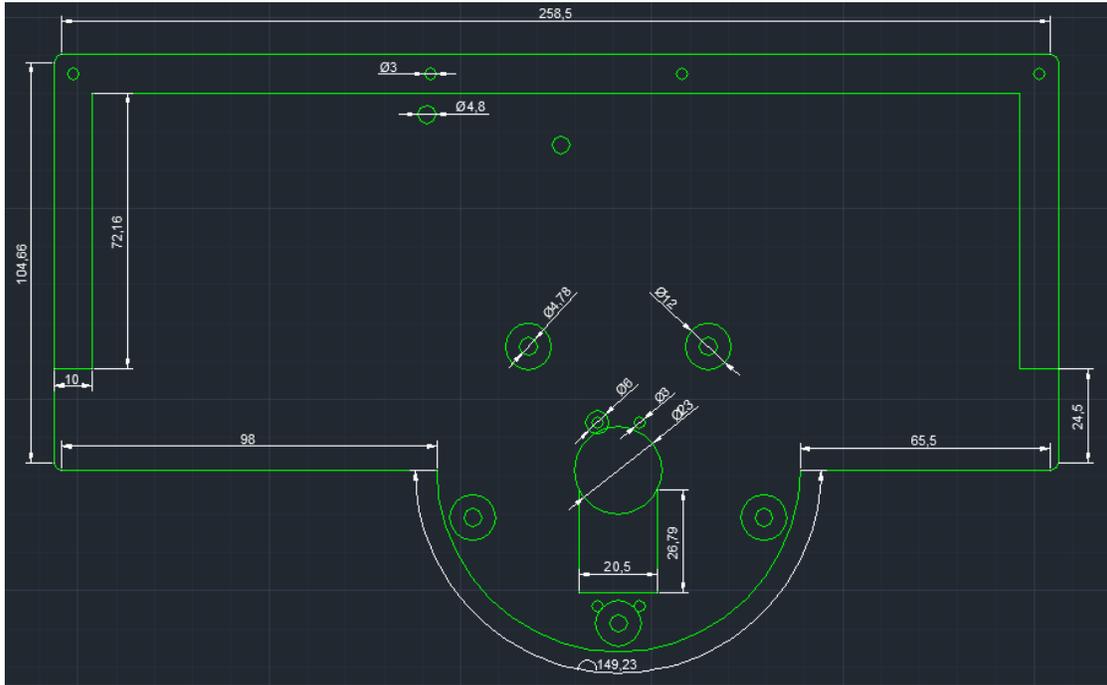


Figura d.8 soporte de la cremallera realizado en AutoCAD [13].

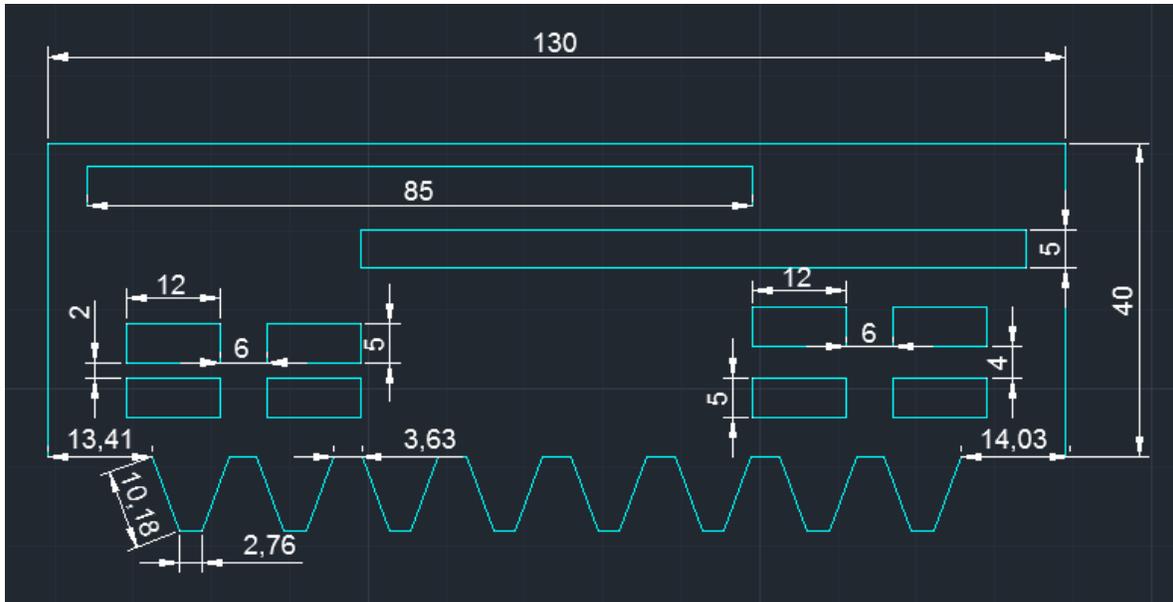


Figura d.9 cremallera que soporta el filtro de polarización y divisor de polarización realizado en AutoCAD [13].



### **e) Código de Arduino**

```
a.servoAttach(7);
a.servoStatus;
a.servoAttach(8);
a.servoStatus;
a.servoAttach(9);
a.servoStatus;
a.servoAttach(10);
a.servoStatus;
a.servoAttach(11);
a.servoStatus;
handles.a = a;

% Choose default command line output for practi1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes practi1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = practi1_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in mot_led_1.
function mot_led_1_Callback(hObject, eventdata, handles)
% hObject handle to mot_led_1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of mot_led_1
button_state = get(hObject, 'value');

if button_state == get(hObject, 'Max');
    set(handles.txt_led_1, 'BackgroundColor', 'g');
```

```
set(handles.txt_led_1,'String', 'Vertical' );
%handles.a.digitalWrite(2,1);
handles.a.servoWrite(7,1);

else button_state == get(hObject,'Min');
set(handles.txt_led_1, 'BackgroundColor', [1 0 0]);
set(handles.txt_led_1,'String', 'Horizontal' );
%handles.a.digitalWrite(2,0);
handles.a.servoWrite(7,180);
end

% --- Executes on button press in mot_led_2.
function mot_led_2_Callback(hObject, eventdata, handles)
% hObject handle to mot_led_2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of mot_led_2
button_state = get(hObject, 'value');

if button_state == get(hObject, 'Max');
set(handles.txt_led_2, 'BackgroundColor', 'g');
set(handles.txt_led_2,'String', 'Vertical' );
%handles.a.digitalWrite(3,1);
handles.a.servoWrite(8,1);

else button_state == get(hObject,'Min');
set(handles.txt_led_2, 'BackgroundColor', [1 0 0]);
set(handles.txt_led_2,'String', 'Horizontal' );
%handles.a.digitalWrite(3,0);
handles.a.servoWrite(8,180);
end
```

**f) Código de matlab**

```
function varargout = practi1(varargin)
% PRACTI1 M-file for practi1.fig
% PRACTI1, by itself, creates a new PRACTI1 or raises the existing
% singleton*.
%
% H = PRACTI1 returns the handle to a new PRACTI1 or the handle to
% the existing singleton*.
%
% PRACTI1('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in PRACTI1.M with the given input arguments.
```

```
%
% PRACTI1('Property','Value',...) creates a new PRACTI1 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before practi1_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to practi1_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help practi1

% Last Modified by GUIDE v2.5 13-Sep-2013 19:58:16

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @practi1_OpeningFcn, ...
                  'gui_OutputFcn', @practi1_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before practi1 is made visible.
function practi1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to practi1 (see VARARGIN)

%CREACION DEL OBJETO ARDUINO PARA OBTENER LA CONECCION ENTRE ARDUINO Y
%MATLAB
a = arduino('COM9');
```

```
a.pinMode(2, 'output');
a.pinMode(3, 'output');
%a.pinMode(4, 'output');
%a.pinMode(5, 'output');
a.servoAttach(7);
a.servoStatus;
a.servoAttach(8);
a.servoStatus;
a.servoAttach(9);
a.servoStatus;
a.servoAttach(10);
a.servoStatus;
a.servoAttach(11);
a.servoStatus;
handles.a = a;

% Choose default command line output for practi1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes practi1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = practi1_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in mot_led_1.
function mot_led_1_Callback(hObject, eventdata, handles)
% hObject handle to mot_led_1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of mot_led_1
button_state = get(hObject, 'value');
```

```
if button_state == get(hObject, 'Max');
    set(handles.txt_led_1, 'BackgroundColor', 'g');
    set(handles.txt_led_1, 'String', 'Vertical' );
    %handles.a.digitalWrite(2,1);
    handles.a.servoWrite(7,1);

else button_state == get(hObject, 'Min');
    set(handles.txt_led_1, 'BackgroundColor', [1 0 0]);
    set(handles.txt_led_1, 'String', 'Horizontal' );
    %handles.a.digitalWrite(2,0);
    handles.a.servoWrite(7,180);
end

% --- Executes on button press in mot_led_2.
function mot_led_2_Callback(hObject, eventdata, handles)
% hObject    handle to mot_led_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of mot_led_2
button_state = get(hObject, 'value');

if button_state == get(hObject, 'Max');
    set(handles.txt_led_2, 'BackgroundColor', 'g');
    set(handles.txt_led_2, 'String', 'Vertical' );
    %handles.a.digitalWrite(3,1);
    handles.a.servoWrite(8,1);

else button_state == get(hObject, 'Min');
    set(handles.txt_led_2, 'BackgroundColor', [1 0 0]);
    set(handles.txt_led_2, 'String', 'Horizontal' );
    %handles.a.digitalWrite(3,0);
    handles.a.servoWrite(8,180);
end

function txt_led_1_Callback(hObject, eventdata, handles)
% hObject    handle to txt_led_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of txt_led_1 as text
%       str2double(get(hObject,'String')) returns contents of txt_led_1 as a double

% --- Executes during object creation, after setting all properties.
```

```
function txt_led_1_CreateFcn(hObject, eventdata, handles)
% hObject handle to txt_led_1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function txt_led_2_Callback(hObject, eventdata, handles)
% hObject handle to txt_led_2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of txt_led_2 as text
% str2double(get(hObject,'String')) returns contents of txt_led_2 as a double
```

```
% --- Executes during object creation, after setting all properties.
function txt_led_2_CreateFcn(hObject, eventdata, handles)
% hObject handle to txt_led_2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in regresar.
function regresar_Callback(hObject, eventdata, handles)
% hObject handle to regresar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
menu
close practi1
```

```
% --- Executes on button press in mot_led_3.
function mot_led_3_Callback(hObject, eventdata, handles)
```

```
% hObject handle to mot_led_3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of mot_led_3
button_state = get(hObject, 'value');

if button_state == get(hObject, 'Max');
    set(handles.txt_led_3, 'BackgroundColor', 'b');
    set(handles.txt_led_3, 'String', '1' );
    %handles.a.digitalWrite(4,1);
    handles.a.servoWrite(9,180);
    handles.a.servoWrite(10,180);
    handles.a.servoWrite(11,1);

else button_state == get(hObject, 'Min');
    set(handles.txt_led_3, 'BackgroundColor', [1 0 0]);
    set(handles.txt_led_3, 'String', '2' );
    %handles.a.digitalWrite(4,0);
    handles.a.servoWrite(9,1);
    handles.a.servoWrite(10,0);
    handles.a.servoWrite(11,180);

end

function txt_led_3_Callback(hObject, eventdata, handles)
% hObject handle to txt_led_3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of txt_led_3 as text
% str2double(get(hObject,'String')) returns contents of txt_led_3 as a double

% --- Executes during object creation, after setting all properties.
function txt_led_3_CreateFcn(hObject, eventdata, handles)
% hObject handle to txt_led_3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on slider movement.
function niveleng1_Callback(hObject, eventdata, handles)
% hObject   handle to niveleng1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%   get(hObject,'Min') and get(hObject,'Max') to determine range of slider
vv=get(handles.niveleng1,'Value');
set (handles.valoreng1,'String',fix(vv));

if vv<1 || vv>179
    if vv == 0
        set(handles.nota,'value',1);
        set(handles.nota,'string','VALOR MINIMO');
        handles.a.servoWrite(7,(vv));
    end
    if vv == 180
        set(handles.nota,'value',1);
        set(handles.nota,'string','VALOR MAXIMO');
        handles.a.servoWrite(7,vv);
    end
else
    set(handles.nota,'value',0);
    set(handles.nota,'string','normal');
    v=fix(vv);
    handles.a.servoWrite(7,v);
end

% --- Executes during object creation, after setting all properties.
function niveleng1_CreateFcn(hObject, eventdata, handles)
% hObject   handle to niveleng1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes during object creation, after setting all properties.
```

```
function valorengr1_CreateFcn(hObject, eventdata, handles)
% hObject handle to valorengr1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% --- Executes on button press in enc_laser.
function enc_laser_Callback(hObject, eventdata, handles)
% hObject handle to enc_laser (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of enc_laser
button_state = get(hObject, 'value');

if button_state == get(hObject, 'Max');
    set(handles.txt_led_4, 'BackgroundColor', 'g');
    set(handles.txt_led_4, 'String', 'ON' );
    handles.a.digitalWrite(3,1);

else button_state == get(hObject,'Min');
    set(handles.txt_led_4, 'BackgroundColor', [1 0 0]);
    set(handles.txt_led_4, 'String', 'OFF' );
    handles.a.digitalWrite(3,0);

end

function txt_led_4_Callback(hObject, eventdata, handles)
% hObject handle to txt_led_4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of txt_led_4 as text
% str2double(get(hObject,'String')) returns contents of txt_led_4 as a double

% --- Executes during object creation, after setting all properties.
function txt_led_4_CreateFcn(hObject, eventdata, handles)
% hObject handle to txt_led_4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in nota.
function nota_Callback(hObject, eventdata, handles)
% hObject    handle to nota (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of nota

% --- Executes on button press in enc_mot.
function enc_mot_Callback(hObject, eventdata, handles)
% hObject    handle to enc_mot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of enc_mot
button_state = get(hObject, 'value');

if button_state == get(hObject, 'Max');
    set(handles.txt_led_5, 'BackgroundColor', 'g');
    set(handles.txt_led_5, 'String', 'ON' );
    handles.a.digitalWrite(2,1);

else button_state == get(hObject, 'Min');
    set(handles.txt_led_5, 'BackgroundColor', [1 0 0]);
    set(handles.txt_led_5, 'String', 'OFF' );
    handles.a.digitalWrite(2,0);

end

function txt_led_5_Callback(hObject, eventdata, handles)
% hObject    handle to txt_led_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of txt_led_5 as text
%       str2double(get(hObject,'String')) returns contents of txt_led_5 as a double

% --- Executes during object creation, after setting all properties.
function txt_led_5_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to txt_led_5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function niveleng2_Callback(hObject, eventdata, handles)
% hObject handle to niveleng2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

%vv=get(handles.niveleng2,'Value');
%set (handles.valoreng2,'String',fix(vv));

%if vv<1 || vv>179
% if vv == 0
% set(handles.nota2,'value',1);
% set(handles.nota2,'string','VALOR MINIMO');
% handles.a.servoWrite(8,(vv));
% end
% if vv == 180
% set(handles.nota2,'value',1);
% set(handles.nota2,'string','VALOR MAXIMO');
% handles.a.servoWrite(8,vv);
% end
%else
% set(handles.nota2,'value',0);
% set(handles.nota2,'string','normal');
% v=fix(vv);
% handles.a.servoWrite(8,v);
%end

% --- Executes during object creation, after setting all properties.
function niveleng2_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to niveleng2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in nota2.
function nota2_Callback(hObject, eventdata, handles)
% hObject handle to nota2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of nota2
```

### g) Datasheet

En la practica desarrollado se ocupado servomotores TowerPro modelo MG996R

[Servo Database](#) > [TowerPro Servos](#) > [MG996R](#)

## TowerPro MG996R - Standard Servo

### Basic Information

Modulation:	Analog
Torque:	<b>4.8V:</b> 130.5 oz-in (9.40 kg-cm) <b>6.0V:</b> 152.8 oz-in (11.00 kg-cm)
Speed:	<b>4.8V:</b> 0.17 sec/60° <b>6.0V:</b> 0.14 sec/60°
Weight:	1.94 oz (55.0 g)
Dimensions:	Length: 1.60 in (40.6 mm) Width: 0.78 in (19.8 mm) Height: 1.69 in (42.9 mm)
Motor Type:	? (add)
Gear Type:	Metal
Rotation/Support:	Dual Bearings

### Additional Specifications

Rotational Range:	? (add)
Pulse Cycle:	? (add)
Pulse Width:	? (add)
Connector Type:	? (add)



Brand:	<a href="#">Tower pro</a>
Product Number:	? (add)
Suggested Retail:	? (add)
Street Price:	11.95 USD
Compare:	<a href="#">add</a>

### User Reviews

Number of Reviews:	3
Average Rating:	3.0 / 5.0

Fuente:

[http://www.electronicoscaldas.com/datasheet/MG995\\_Tower-Pro.pdf](http://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf)