

# Proyecto final

## Diego Isaac Hurtado Miguel, Miguel Pablo Juárez Sánchez, Ricardo Rodríguez, Ramírez.

**Resumen** - En este documento se presentará todo el desarrollo de un robot cartesiano. Desde su programación hasta su diseño físico.

**Abstract** - This document contains the development of a cartesian robot. From its programming to its physical prototype.

### Introducción

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

MATLAB es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

La topología cartesiana es la más extendida en los robots utilizados en la industria. El robot cartesiano es el robot más simple, pudiendo ser fácilmente comprendido por el usuario dado que, como su nombre indica, el esquema de su movimiento se basa en el sistema de ejes cartesianos X-Y-Z tan bien conocido. Por otra parte, también es el robot más fácil de concebir y fabricar porque su producción es tan sencilla como ir acoplando múltiples actuadores lineales unos sobre otros perpendicularmente. Indudablemente también es una de las construcciones más económicas.

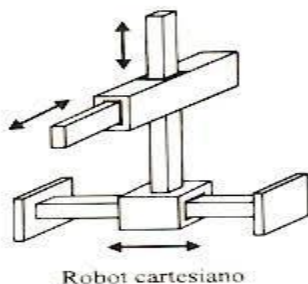


Figura 1. Concepto de un robot cartesiano.

### Objetivo:

Programar, diseñar en CAD los elementos mecánicos, y simular en MATLAB la cinemática de un robot de tipo cartesiano.

### Estado del arte:

Actualmente, los fabricantes más famosos de robots cartesianos son Shibaura Machine, Shibaura es una compañía americana que antes era japonesa de nombre Toshiba Machine, quienes los fabrican para la industria desde 1986. Sus robots están diseñados para tener gran confiabilidad, eficiencia y simplicidad para trabajar con ellos. Otros fabricantes de robots cartesianos son Omron y Yamaha, aunque en menor medida.

Una aplicación interesante es el implementado en 2018 durante un torneo de ajedrez. La cual consistió en jugar una partida utilizando un robot KUKA

### Marco Teórico

En el estudio de la Robótica, existe un proceso denominado algoritmo de Denavit–Hartenberg, que nos ayuda a establecer los sistemas de referencia para cada uno de los eslabones del robot.

En este algoritmo a cada articulación se le asigna un Sistema de Referencia Local con origen en un punto  $Q_i$  y ejes ortonormales  $\{X_i Y_i Z_i\}$ , comenzando con un primer S.R fijo e inmóvil dado por los ejes  $\{X Y Z\}$ , anclado a un punto fijo  $Q_0$  de la Base sobre la que está montada toda la estructura de la cadena. Este Sistema de Referencia no tiene por qué ser el Universal con origen en  $(0,0,0)$  y la Base canónica.

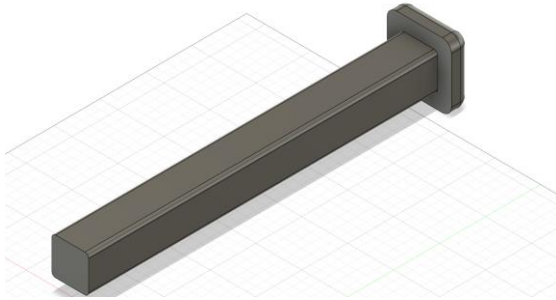
### Planteamiento del problema:

Se desea realizar una simulación sobre el funcionamiento de un robot cartesiano. Se deberá de poder simular el cambio de posiciones mediante Matlab, para implementarlo de manera exitosa en un tablero de ajedrez.

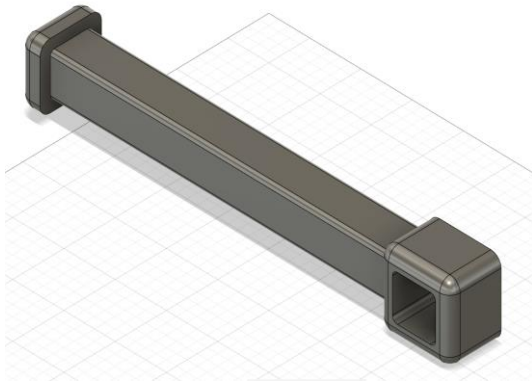
### Desarrollo:

Para el desarrollo seguiremos los siguientes pasos:

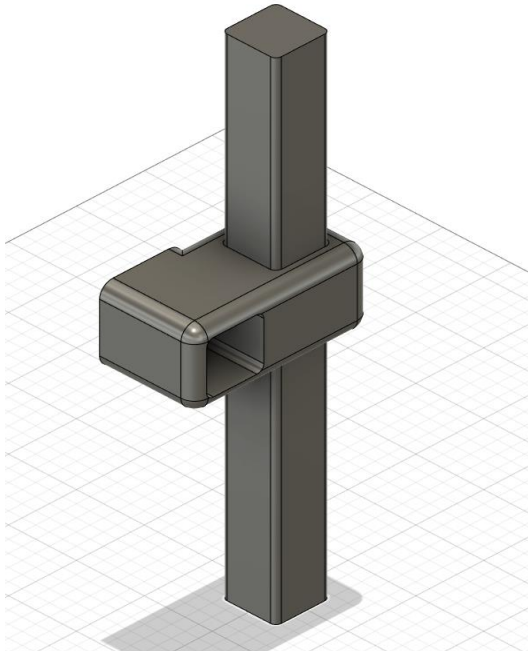
1. Diseño en CAD.



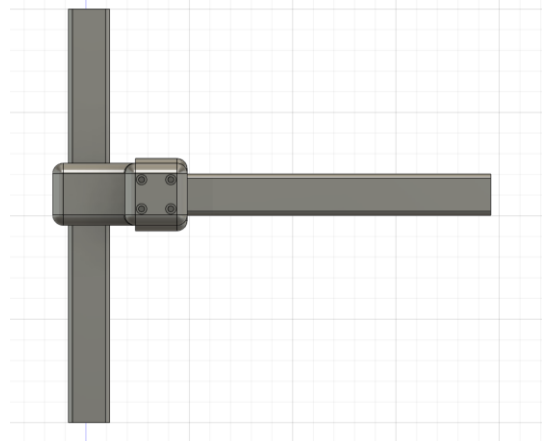
*Figura 2. Eje X*



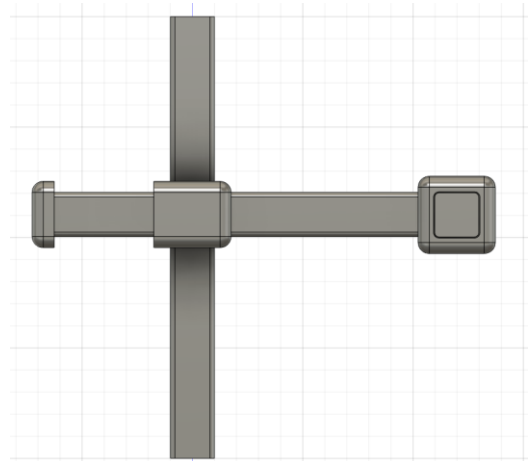
*Figura 3. Eje Y*



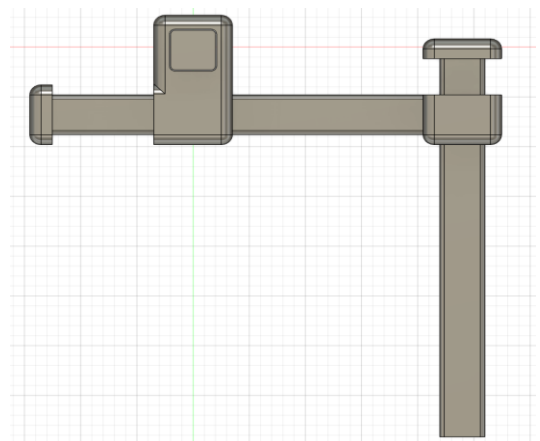
*Figura 4. Eje Z*



*Figura 5. Vista 1.*



*Figura 6. Vista 2.*



*Figura 7. Vista 3*

Ensamble:

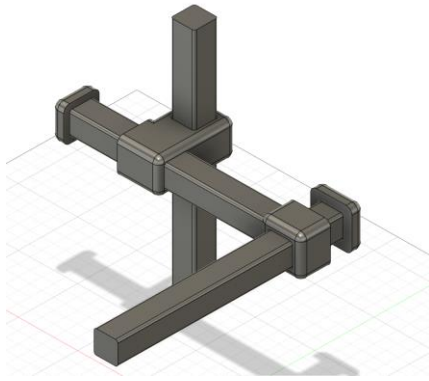


Figura 8. Vista isométrica.

## 2. Electrónica.

- L298N: El L298 es un puente H que soporta hasta 4A de corriente continua y hasta 35V. Usualmente es utilizado para controlar dos motores dc, pero también se puede controlar un a pasos si se alimenta con las dos salidas de motor dual, por lo que necesitaremos tres de ellos para controlar las tres articulaciones. Se utilizará el siguiente breakout board para L298, disponible en <link>:<https://sandorobotics.com/producto/hr0112/> :

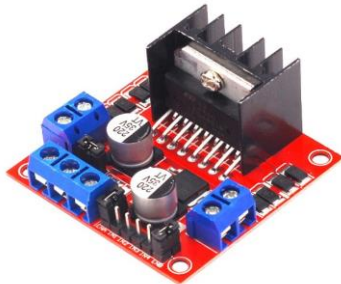


Figura 9. L298N

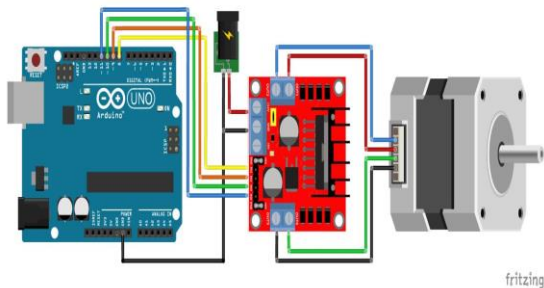


Figura 10. Conexión del L298N con un motor a pasos.

- Motor a Pasos: Se pretende utilizar, para las tres articulaciones, tres motores a pasos Pololu, bipolares de 2A por fase y 3.6V a tensión nomina, de tamaño NEMA23 y eje tipo D, disponible

en <link>:<https://sandorobotics.com/producto/1473/> :



Figura 11. Motor a pasos.

- Arduino MEGA: La parte de control, se llevará a cabo con ayuda de un Arduino MEGA, el cual tiene puertos digitales suficientes para controlar los tres motores a pasos, con la posibilidad de agregar encoders, para retroalimentar el sistema, ya que también cuenta con puertos analógicos y con protocolo de comunicación  $I^2C$ , que pueden ser de gran ayuda, disponible en <link>:<https://sandorobotics.com/producto/sd-a067/> :

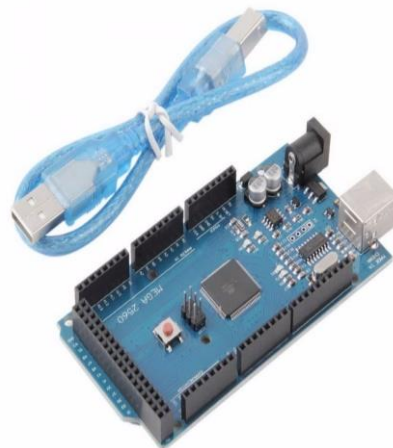


Figura 12. Arduino MEGA

- Se pretende utilizar una impresora 3D, y con ayuda de Fusion 360 se diseñará y se imprimirán las piezas del robot.



Figura 13. Software FUSION 360.

- Obtención de parámetros para el programa en MATLAB

Eslabón	Alpha	Theta	d	a
1	90°	90°	d1	0
2	90°	-90°	d2	0
3	0	0	d3	0

- Matriz de Denavit Hartenberg

0	1	0	d2
0	0	1	-d3
1	0	0	d1
0	0	0	1

- Simulación en MATLAB

Codigo:  
clear all  
close all  
clc

L1 = Prismatic('a', 0, 'alpha', pi/2, 'theta', pi/2, 'qlim',[0,20]);

L2 = Prismatic('a', 0, 'alpha', pi/2, 'theta', -pi/2, 'qlim',[0,20]);

L3 = Prismatic('a', 0, 'alpha', 0, 'theta', 0, 'qlim',[0,20]);

scara=SerialLink([L1,L2,L3]);

scara.plot([10, 10, 10]);

scara.teach

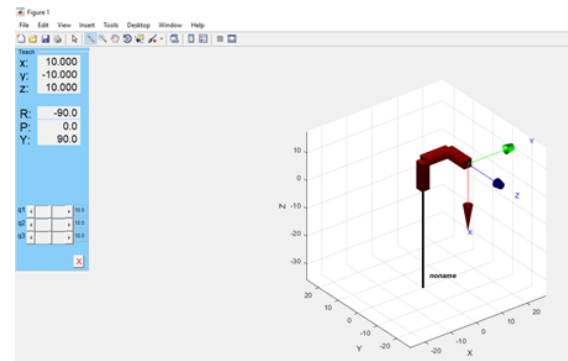


Figura 14. Simulación MATLAB

Para la simulación en MATLAB fue utilizada la caja de herramientas del profesor Corke con el Código antes presentado, en el se establecen 3 eslabones de tipo prismático, con sus respectivos ángulos de cada uno, con rangos de movimiento para cada eslabón de 0 a 20 unidades longitudinales.

- Acceso a GitHub

<https://github.com/MRDAMPC/BrazoRobotico>

## Resultados

Utilizando la herramienta TINKERCAD, se obtuvo la siguiente simulación:

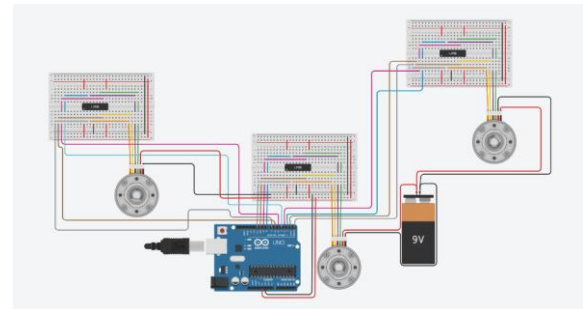


Figura 15. Simulación con TINKERCAD.

Para el diseño del circuito se utilizan 3 motores, los cuales representan cada eje de un robot de tipo cartesiano. Dentro de la carpeta de MATLAB en Github se puede apreciar la forma en que MATLAB se comunica con Arduino.

## Conclusiones

Es posible automatizar el algoritmo de Denavit Hartenberg para obtener la cinemática directa de un robot, con ayuda de MATLAB, en donde también se

puede simular al mismo y su comportamiento al aplicarle el algoritmo, si se le proporcionan coordenadas válidas. Se logró hacer uso de un sistema Git para respaldar y ordenar la documentación y progreso del proyecto, así como recopilar información suficiente para la construcción de un brazo robótico de tipo cartesiano.

## Referencias

1. Robótica: Algoritmo de Denavit–Hartenberg. Caso de estudio SSRMS – Mafer's Tech Holdings industries. (2021). Retrieved 4 January 2021, from <https://maferstech.com/robotica-algoritmo-de-denavit-hartenberg-caso-de-estudio-ssrms/#:~:text=En%20el%20estudio%20de%20la,los%20que%20cuenta%20el%20robot>
2. cartesianos, R. (2021). Robots cartesianos | Larraioz Elektronika. Retrieved 4 January 2021, from <https://larraioz.com/iai/productos/robots-cartesianos>
3. (2021). Retrieved 4 January 2021, from [https://personal.us.es/jcortes/Material/Material\\_archivos/Articulos%20PDF/RepresentDH.pdf](https://personal.us.es/jcortes/Material/Material_archivos/Articulos%20PDF/RepresentDH.pdf)
4. MATLAB. (2021). Retrieved 8 January 2021, from <https://es.wikipedia.org/wiki/MATLAB>
5. Qué es Arduino, cómo funciona y qué puedes hacer con uno. Retrieved 8 January 2021, from <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
6. Serial Link, Consultado el 08 de enero del 2021 <https://www.petercorke.com/RTB/r9/html/SerialLink.html>