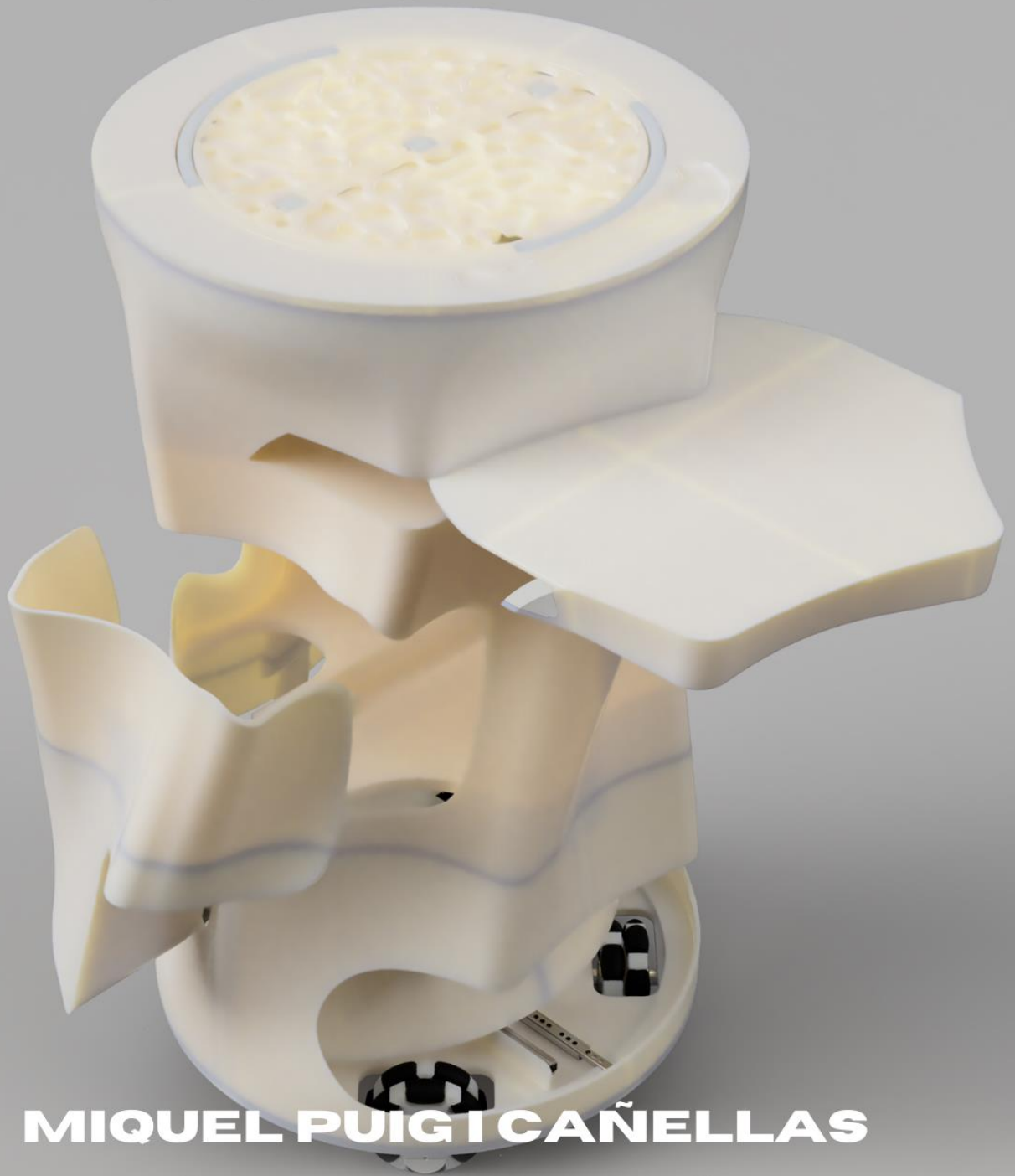


# SMART TABLE ASSISTANT



**MIQUEL PUIG I CAÑELLAS**

**TUTORA: MÒNICA MASCORT**

INSTITUT VIDRERES



# **ABSTRACT O RESUM DEL TREBALL DE RECERCA:**

## **RESUMEN:**

El Smart Table Assistant es un robot que ha sido creado con el objetivo de ayudar a la gente con incapacidad en las extremidades inferiores, capaz de adaptarse al usuario y facilitarle las tareas en las comidas del día.

Primero se ha estudiado el movimiento omnidireccional de las ruedas, seguidamente se ha diseñado para una mejor ergonomía en el producto, después se han programado las instrucciones de la mesa, se ha subido el programa a la placa Arduino y finalmente se ha construido el robot y sus mecanismos para que funcione todo correctamente. La mayoría de la estructura de este proyecto ha sido posible gracias al método de la impresión 3D.

A partir de todo el trabajo realizado, se han podido observar todas las etapas de la creación de un proyecto de estas dimensiones y su complejidad, el desarrollo de un producto en el mercado y sus costes y se ha podido cumplir con todas las expectativas que se tuvieron en un inicio.

## **ABSTRACT:**

The Smart Table Assistant is a robot that has been created with the aim of helping people with disabilities in the lower extremities, capable of adapting to the user and facilitating the tasks in the meals of the day.

First, the omnidirectional movement of the wheels has been studied, then it has been designed for better ergonomics in the product, then the instructions of the table have been programmed, then it has been applied to the elements through Arduino and finally the mechanisms have been made to make it work properly. Most of the structure of this project has been possible thanks to the 3D printing method.

From all the work carried out, it has been possible to observe all the stages of the creation of a project of these dimensions and its complexity, the development of a product in the market and its costs, and it has been possible to comply with all the expectations that were established in the beginning.



# ÍNDEX

1.- INTRODUCCIÓ DE L´S.T.A .....	1
- Objectius	
- Motivació	
2.- NECESSITAT I DESTINATARIS .....	2
3.- FUNCIONALITATS I ÚS .....	3
4.- PLATAFORMES PER DUR A TERME LA PROGRAMACIÓ .....	4
- Arduino	
- Blynk	
5.- MOVIMENT OMNIDIRECCIONAL .....	8
6.- INSPIRACIÓ DEL DISSENY .....	10
7.- DISSENY .....	11
8.- MECANISMES .....	27
9.- ELECTRÒNICA .....	30
10.- PROGRAMACIÓ .....	35
11.- CONSTRUCCIÓ .....	45
12.- RESULTAT I FOTOGRAFIES DE L´S.T.A .....	46
13.- COST I ECONOMIA DEL PROJECTE .....	47
14.- CONCLUSIONS I ASSOLIMENTS .....	48
15.- AGRAÏMENTS .....	49
16.- INFOGRAFIA .....	50



# 1.- INTRODUCCIÓ DEL S.T.A

## OBJECTIU:

Aquest projecte té com a objectiu construir una maqueta d'un robot destinat a persones amb discapacitat a les extremitats inferiors. Serà capaç de memoritzar el recorregut des del punt d'inici, anar fins al destinatari i mantenir-se en la posició, fins a donar l'ordre de tornar a la base. Podrà arribar a moure's en diverses direccions, gràcies a les rodes omnidireccionals situades a la plataforma de la "taula intel·ligent". L'aparell contindrà un comandament a distància, format per una pantalla, una palanca de comandament i quatre botons, on el robot rebrà la informació per realitzar certes funcions, com ara, la disponibilitat d'una taula extraïble, una bodega, activar la retroil·luminació i fer possible l'espai pel reposapeus.

## MOTIVACIÓ:

La principal idea que em va motivar a dur a terme aquest projecte va ser quan la meua mare va patir una immobilitat motriu durant un mes. Vaig estar rumiant en com a ajudar-la i aprofitant el treball de recerca, volia aplicar els meus coneixements per trobar una solució a les seves dificultats.

## **2.- NECESSITAT I DESTINATARIS**

El projecte Smart Table Assistant va sorgir de la necessitat d'ajudar les persones afectades per la paraplegia, un concepte que s'utilitza per descriure la pèrdua corporal de moviment del sistema nerviós i que majoritàriament involucra la paràlisi completa o parcial de les cames i en alguns casos el tronc, però no els braços.

Seguidament, em vaig qüestionar com podria ajudar les persones que patissin aquesta immobilitat, i considerant-ho molt i amb les tecnologies que tenia a mà, vaig concloure que el món de la impressió 3D permetria personalitzar el mobiliari per ajudar en certes patologies.

Ja és hora que els mobles s'automatitzin i siguin capaços de transformar-se per tal d'adaptar-se a totes les necessitats i comoditats. No ens hem de conformar en veure les pel·lícules futuristes quan ja és possible gaudir-ho en el present.

L'S.T.A vol obrir un univers de possibilitats de mercat, adreçat a satisfer l'usuari i auxiliar les seves discapacitats per tal de proporcionar comoditat.



**S M A R T   T A B L E   A S S I S T A N T**

IMATGE 1. LOGO SMART TABLE ASSISTANT (PRÒPIA)



### **3.- FUNCIONALITATS I ÚS**

L´Smart Table Assistant té diverses funcionalitats que s´activen pel comandament a distància, certes accions a partir de la pantalla tàctil i d´altres per diferents components inclosos en el mateix radiocontrol:

- Gràcies a un *joystick* incorporat en el comandament, des d´un inici es pot controlar l´S.T.A en totes les direccions.
- És possible gravar un recorregut, perquè a posteriori el robot pugui anar al seu destinatari accionant el botó de reproduir, i quan es premi el polsador de tornar, fer el mateix recorregut a la inversa fins a arribar al punt de partida.
- Es desplaça part de l'estructura del cos per formar un espai on encabir els peus juntament amb el reposapeus de la cadira de rodes, per així, poder menjar més còmode i a prop de la taula.
- Conté un buit no visible per col·locar plats i coberts al centre del robot. D´allà mateix se n´extreu un ampoller, per facilitar l´accés a la beguda i tenir-ho a mà. Altrament, té la capacitat de conservar les temperatures dels plats, ja que és una zona estanca.
- A la part superior del producte, activant la funció de la taula, es mou un conjunt per així tenir la disponibilitat d´una taula unipersonal d´accés directe.
- Situat a dalt de tot, hi ha un cubell per dipositar els plats bruts, coberts i gots per quan ja s´ha acabat l´àpat, simplificar la feina a l´hora de recollir i posar-los al rentaplats directament.
- Per cobrir el forat del cubell, té una coberta simulada amb un mosaic. Aquest es fa servir com a nansa per agafar-lo. També té un relleu en el disseny que serveix per posar el plat principal, el plat petit i un got; així tenir a l´abast el menjar quan estàs assegut.
- Il·luminació amb colors a l´interior de l´S.T.A.



IMATGE 2, 3, 4 I 5. FUNCIONS DE L'S.T.A (PRÒPIA)

## 4.- PLATAFORMES PER DUR A TERME L´S.T.A

Per poder programar i fer possible el funcionament del robot, he hagut d'utilitzar dos programes de codi obert.

### ARDUINO

L'Arduino és una plataforma destinada a l'aprenentatge i creació d'electrònica de codi obert, basada en l'aplicació de diferents elements en hardware i software lliure, ampli i fàcil pels desenvolupadors. Utilitza una variació del llenguatge C++, que està pensada per poder controlar les seves pròpies plaques.



IMATGE 6 I 7. LOGO CODI OBERT C++ I ARDUINO

· Quins tipus de plaques existeixen? Hi ha molta varietat de plaques, però les que se solen utilitzar més són les següents:

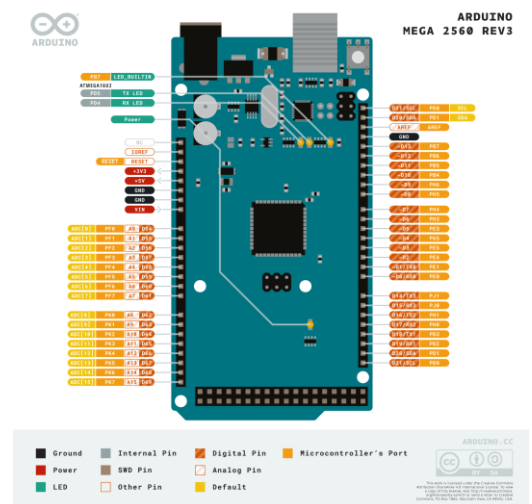
- **Arduino UNO**, la placa original on la major part de "shields" es dissenyen per aquesta, la més habitual.
- **Arduino Mega 2560**, la usada per aquest projecte.
- **Arduino Nano**, com l'Arduino clàssica, però en unes mides molt més petites, conté un port Micro USB.
- **Arduino Micro**, dissenyada per ser utilitzada en protoboards, conté comunicació USB permetent processadors secundaris.
- **Arduino Mini**, dissenyada per ser muntada i utilitzada en protoboard, però no té cap port USB.

- Per què he escollit l'Arduino Mega 2560 per aquest treball?

La Mega 2560 és la placa amb la qual vaig introduir-me en el món de l'arduino, és amb la que estic més familiaritzat i és la placa que disposa de més varietat i quantitat de pins. N'he utilitzat tres d'elles, un parell per la plataforma del robot, on reben senyals i són subordinades de la principal, i la que emet informació al control a distància.

#### · Característiques de l'Arduino Mega 2560:

- Microcontrolador: ATmega2560
- Voltatge d'operació: 5V
- Voltatge d'entrada recomanat: 7 - 12V
- Voltatge d'entrada mínim i màxim: 6 - 20V
- Pins digitals: 54 on d'ells 15 són sortida PWM
- Pins d'entrada analògica: 16
- Corrent CC per cada pin: 20 mA
- Corrent CC per el pin de 3.3V: 50mA
- SRAM: 8KB
- EEPROM: 4KB
- Longitud: 101.52mm
- Amplitud: 53.3mm
- Pes: 37g



IMATGE 8. ARDUINO MEGA 2560

#### · Funcions especialitzades d'alguns pins:

- **Serial:** els pins 0 (RX) i 1 (TX) del port Serial 0, els 19 i 18 (RX, TX) del port Serial 1, els 17 i 16 (RX, TX) del port Serial 2; i els pins 15 (RX) i 14 (TX) del port serial 3.
- **LED:** pin 13 pel LED incorporat.
- **SPI:** pins 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) utilitzats amb llibreria SPI.
- **PWM:** dels pins 2 a 13 i 44 a 46, amb la funció *analogWrite()*.

## **BLYNK**

La plataforma Blynk és una eina que permet controlar fàcilment qualsevol projecte Arduino amb un dispositiu amb sistema iOS o Android. Combina una plataforma en el núvol amb una aplicació per crear prototips d'apps que es connecten remotament amb dispositius electrònics. Per aquest motiu he escollit aquesta plataforma, ja que m'ha ajudat molt al comunicar el mòbil mitjançant Bluetooth a la placa Arduino de la taula.

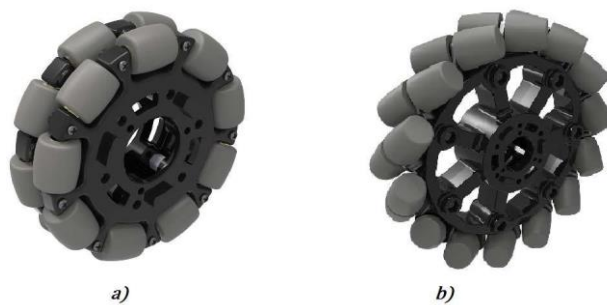


IMATGE 9. LOGO BLYNK

## 5.- MOVIMENT OMNIDIRECCIONAL

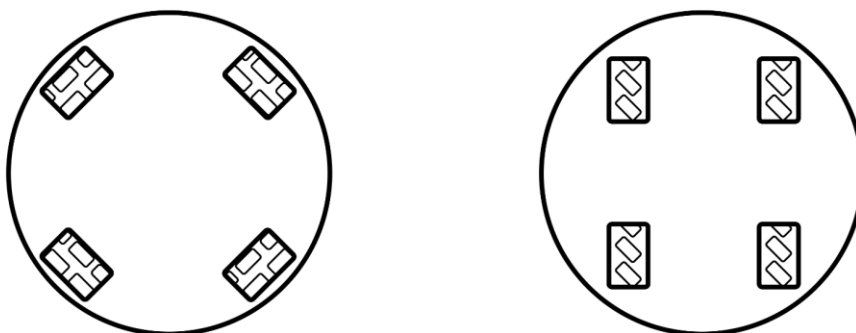
Tots estem acostumats al moviment de les rodes d'un automòbil, el qual és vectorial. Amb les rodes omnidireccionals, es pot desplaçar l'objecte en totes direccions: endavant, endarrere, esquerra, dreta, endavant-dreta, endavant-esquerra, endarrere-dreta i endarrere-esquerra.

Aquest efecte s'aconsegueix mitjançant la introducció de diferents rodes en comú, de manera que moltes rodes es col·loquen dins d'una altra més gran. Aquesta combinació de diverses rodes fan que s'anul·lin els esforços oblics, podent-se desplaçar en cada direcció.



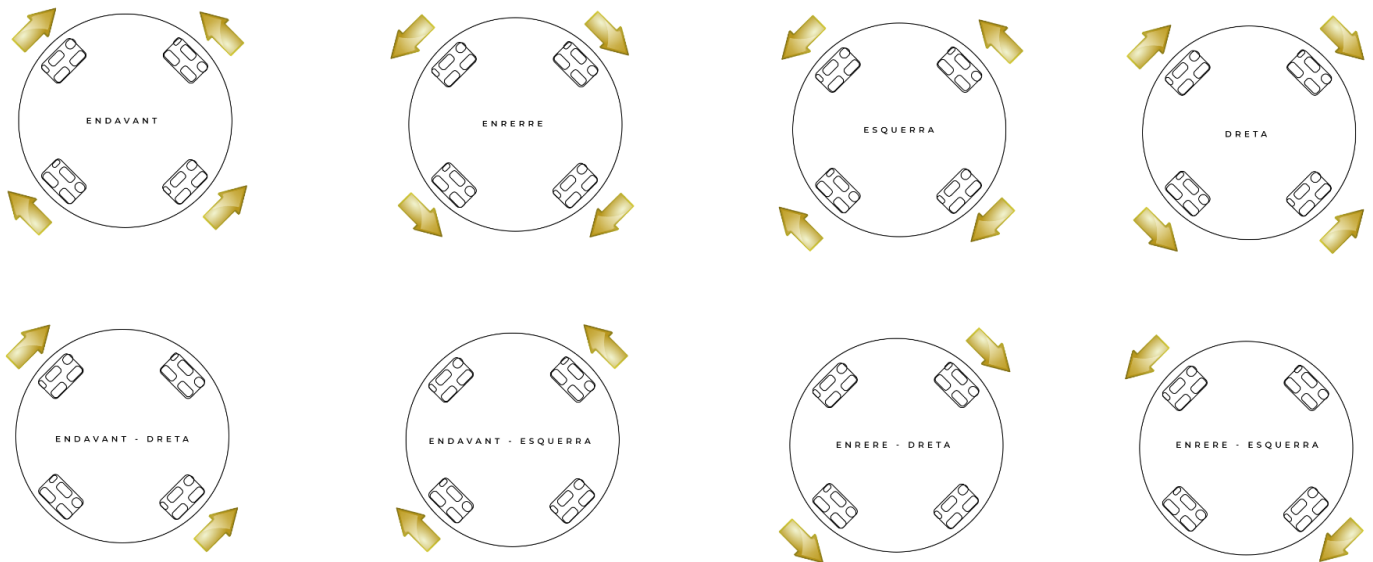
IMATGE 10. TIPUS RODES OMNIDIRECCIONALS

Per poder fer aquests moviments, segons el model de roda que triïs (a o b) tindràs diverses opcions per posicionar les teves rodes.



IMATGE 11. POSICIONS RODES OMNIDIRECCIONALS (PRÒPIA)

El moviment omnidireccional de l'Smart Table Assistant ha estat pensat de la següent manera per a desplaçar-se, per així, accionar el màxim de servomotors alhora.



SMART TABLE ASSISTANT

IMATGE 12. ACCIONAMENT RODES OMNIDIRECCIONALS (PRÒPIA)



## 6.- INSPIRACIÓ DEL DISSENY



IMATGE 13. INSPIRACIÓ SAGRADA FAMILIA

Recordo el moment que estava rumiant en l'estètica del meu producte del treball de recerca, del robot. Estava passejant amb la meva família per l'interior de la Sagrada Família, i de sobte, observant com n'era de preciosa, em vaig fixar en dues característiques bàsiques:

- Les **columnes** tenen una peculiar forma de torsió entre elles mateixes; emeten un toc modern i ascensional. Una sola columna pot arribar a impressionar, i això és el que volia per a la meva taula, que sense estar en funcionament cridés l'atenció de qui ho mirés.
- Els **vitralls** de l'edifici es veuen reflectits en la incorporació dels leds en l'S.T.A, l'entrada de la llum en diferents colors de l'interior de la Sagrada Família, dona un toc d'inspiració, tranquil·litat i modernització el qual volia que es veiés reflectit en el projecte.



## **7.- DISSENY**

El programa que he fet servir per dissenyar l'“Smart Table Assistant” ha estat el Fusion 360, en mans de la companyia Autodesk, un software de disseny 3D fàcil i complet. El motiu pel qual he escollit aquesta aplicació és per la seva àmplia disponibilitat de serveis i la llicència de tres anys que incorpora per ser estudiant. Així mateix, tenia la possibilitat de trobar molts tutorials perquè em facilités l'aprenentatge i el seu ús.

Fusion 360, té una interfície capaç de renderitzar el teu model, simular el material i la seva combinació, fingir on s'apliquen les forces, animar moviments i realitzar esbossos del producte. Una de les funcions que m'ha ajudat molt a dur a terme els enllaços de les meves peces ha estat la simulació de moviment lineal dels objectes. Així amb la forma abstracta en què està basada, el robot poder fer talls i veure com la plataforma o altres parts hi encaixen correctament.



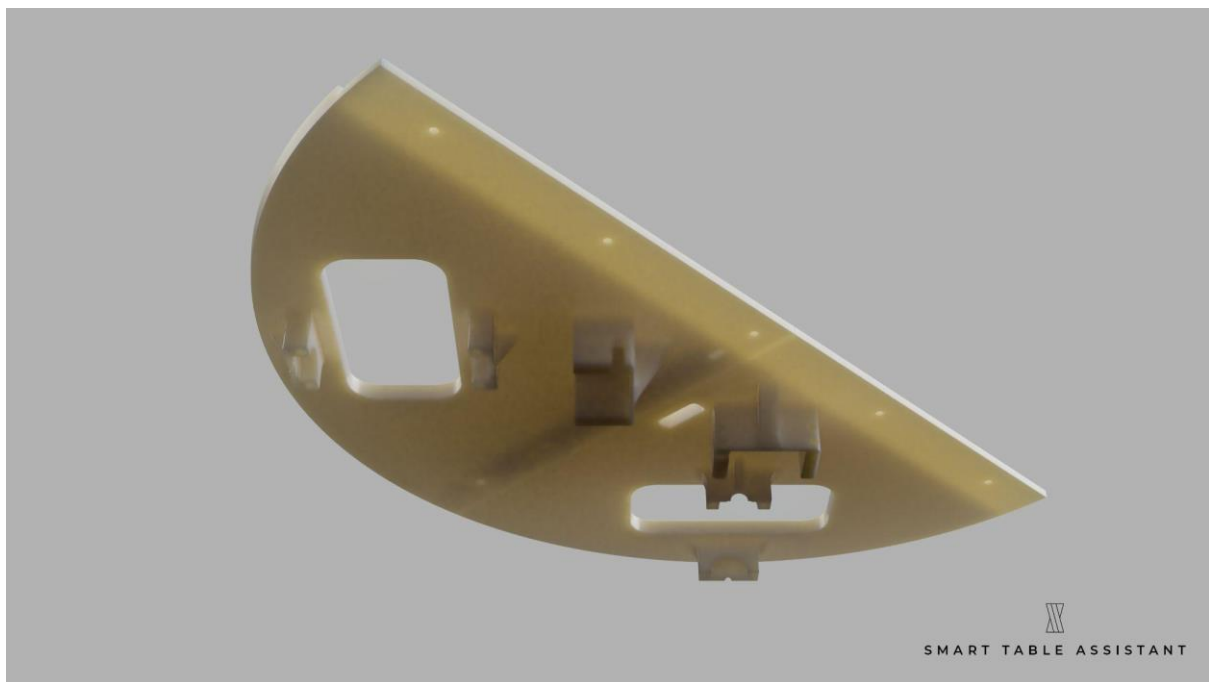
IMATGE 14. LOGO FUSION 360

L'estructura de l'S.T.A. està dividida en cinc parts que es veuen afectades per la mida òptima de les impressores 3D que tinc a mà: Artillery Sidewinder X1 i Creality cr-10S, ambdues amb una base de 300mm \* 300mm.

· **Estructura general:**

- Base del robot: La base és molt important en el robot, aguanta el pes de tota la impressió i subjecta les rodes omnidireccionals. Està formada per quatre peces i els enllaços per mantenir la unió de les rodes amb la plataforma. Els cossos que les uneixen amb la base, tenen forats per així, ancorar-les mitjançant cargols de mètrica tres.

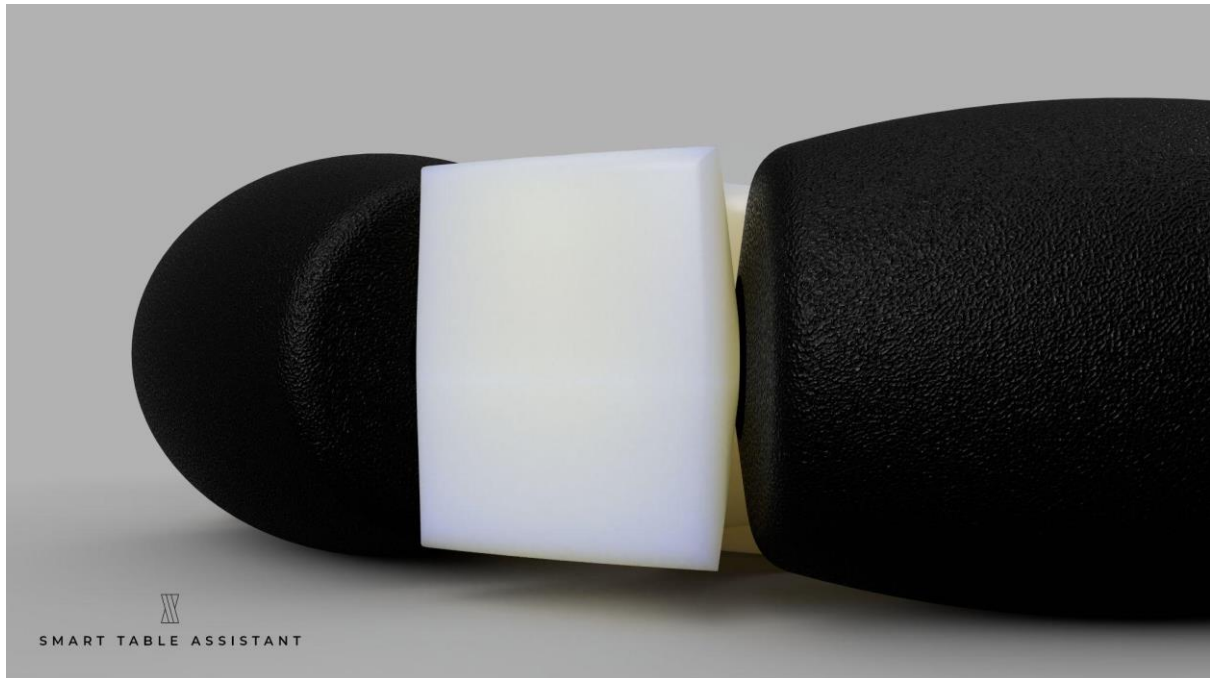
La plataforma està dividida en quatre grans cossos, entrellaçats entre ells superposant-se per formar un grup més estable. Estan units, per precaució, amb silicona calenta i cargols de mètrica quatre en aquest cas.



IMATGE 15. PLATAFORMA DIVIDIDA (PRÒPIA)

Principalment, perquè unes rodes es puguin moure en totes les direccions correctament han d'estar ben collades i les seves rodetes han de tenir facilitat per moure's. Per això, a l'hora de dissenyar les rodes omnidireccionals he pensat en el següent:

· Per evitar el desgast del material i que tingui millor rodament, cada rodeta (negre) té un angle de seguretat respecte a la part blanca.



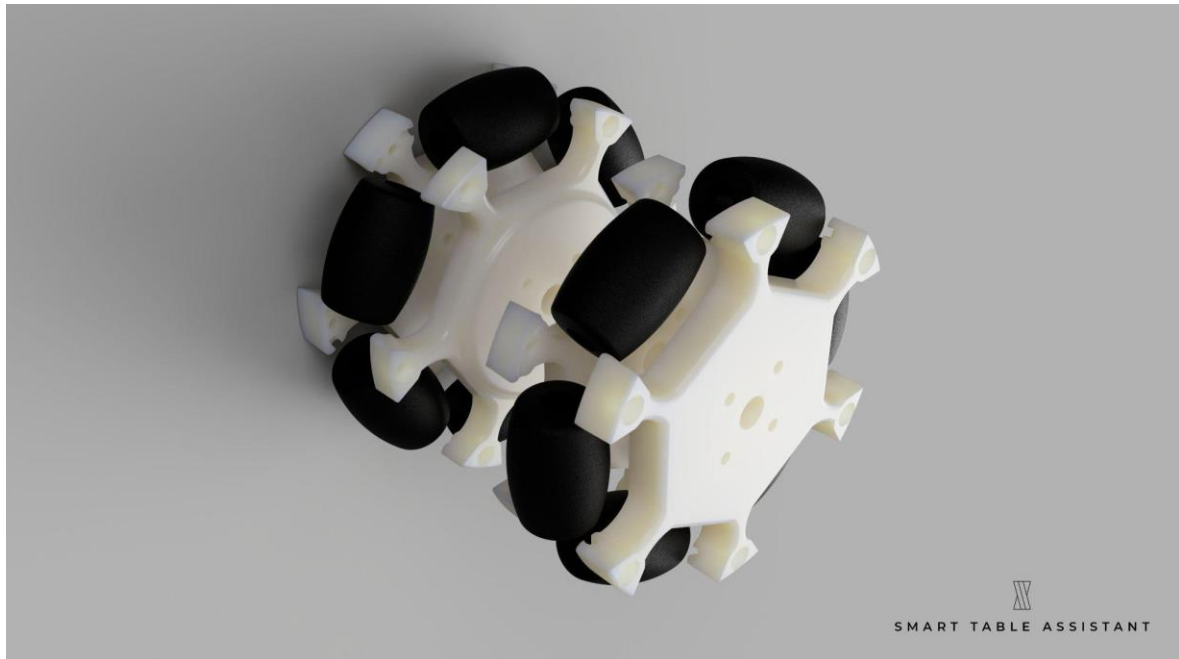
IMATGE 16. ANGLE (PRÒPIA)



IMATGE 17. ANGLE SUPERIOR (PRÒPIA)

- Col·locar dos coixinets per cada rodeta evitant el fregament entre la impressió i l'eix de subjecció del component.

· Pensat per facilitar el seu muntatge, impreses en quatre peces, ajuntant-les per formar dos conjunts motrius els quals s'uneixen amb cargols pressionats per les femelles, assegurant la seva unió, ja que és molt important que no es bellugui cap part.



IMATGE 18. QUATRE PARTS DE LES RODES (PRÒPIA)

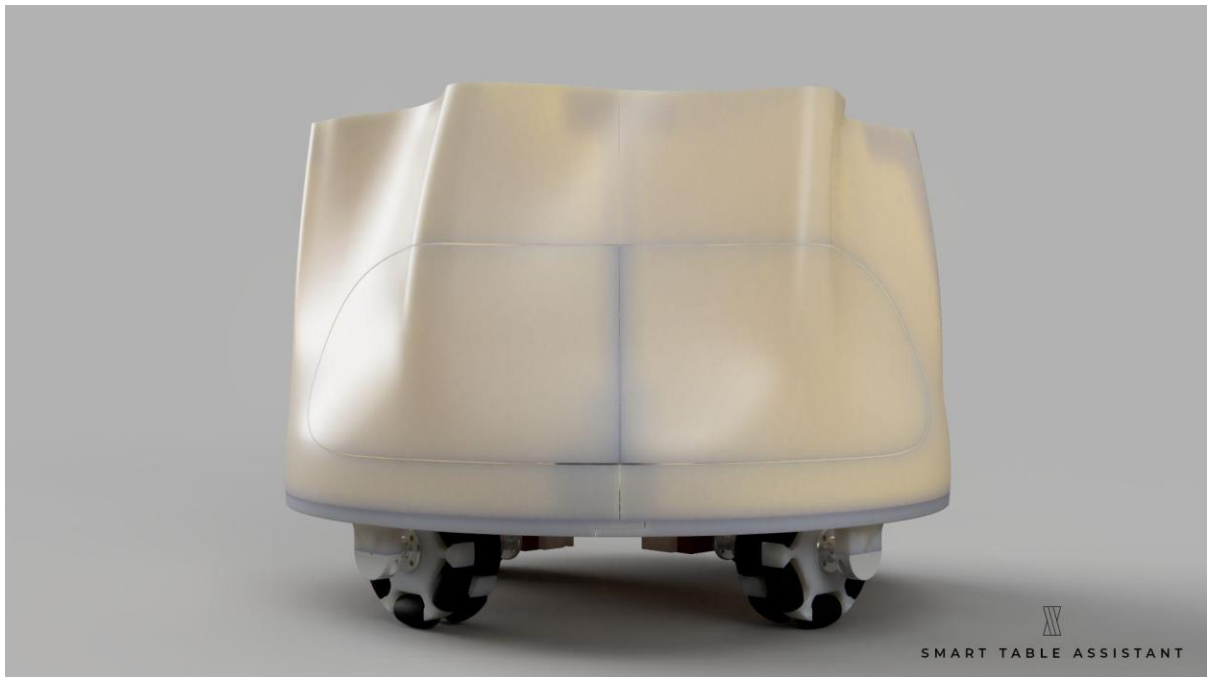
· Les dues parts directores estan diferent posicionades, una d'elles està rotada sense canviar el centre x, y perquè així, que no hi hagi cap punt en el desplaçament del robot on hi hagi un punt mort i la taula tingui joc. Per comprovar els espais entre les rodetes, s'ha fet servir una perspectiva ortogonal i com es pot observar, forma un cercle perfecte.



IMATGE 19. DIFERENT POSICIÓ (PRÒPIA)

- Motriu inferior: Està formada per la carcassa visible i per la distribució dels elements electrònics a través de caixes per organitzar tot el material, les quals, estan pensades per evitar embolics entre cables i que realitzin el menor recorregut permès.

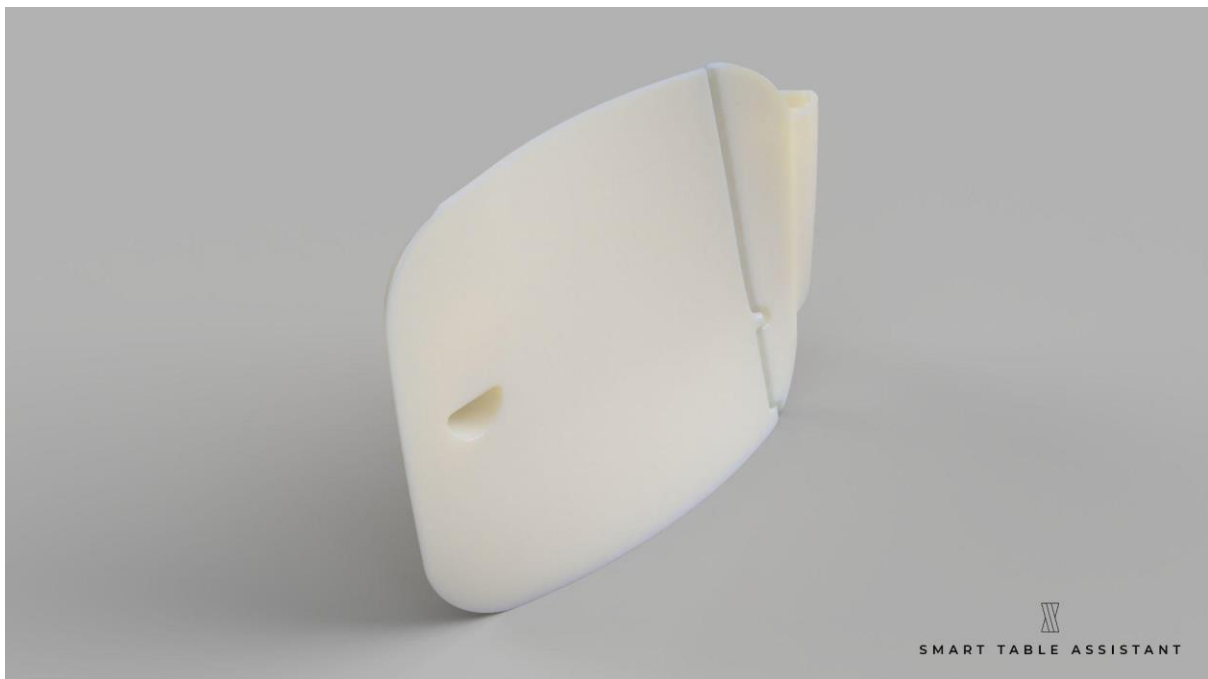
La part exterior, composta per sis cossos impresos 3D, iniciant la forma icònica del projecte, d'entrada una circumferència perfecta que a mesura que es forma el volum es veu com ella mateixa es torça. Quatre parets on dues d'elles, estan modificades per fer possible l'espai on posicionarem els peus juntament amb el reposapeus de la cadira de rodes, i les altres dues, editades per ficar-hi una tapa per accedir a l'interior del robot.



IMATGE 20. ESPAI TANCAT PEL REPOSAPEUS (PRÒPIA)

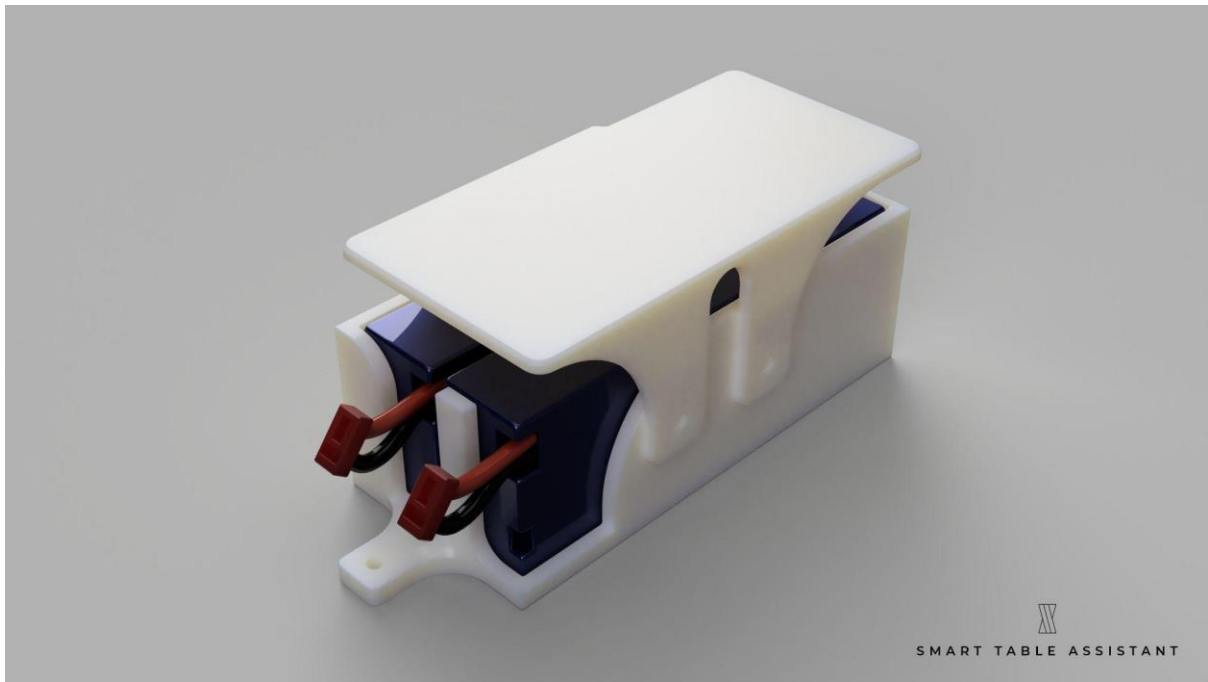


IMATGE 21. ESPAI OBERT PEL REPOSAPÉUS (PRÒPIA)

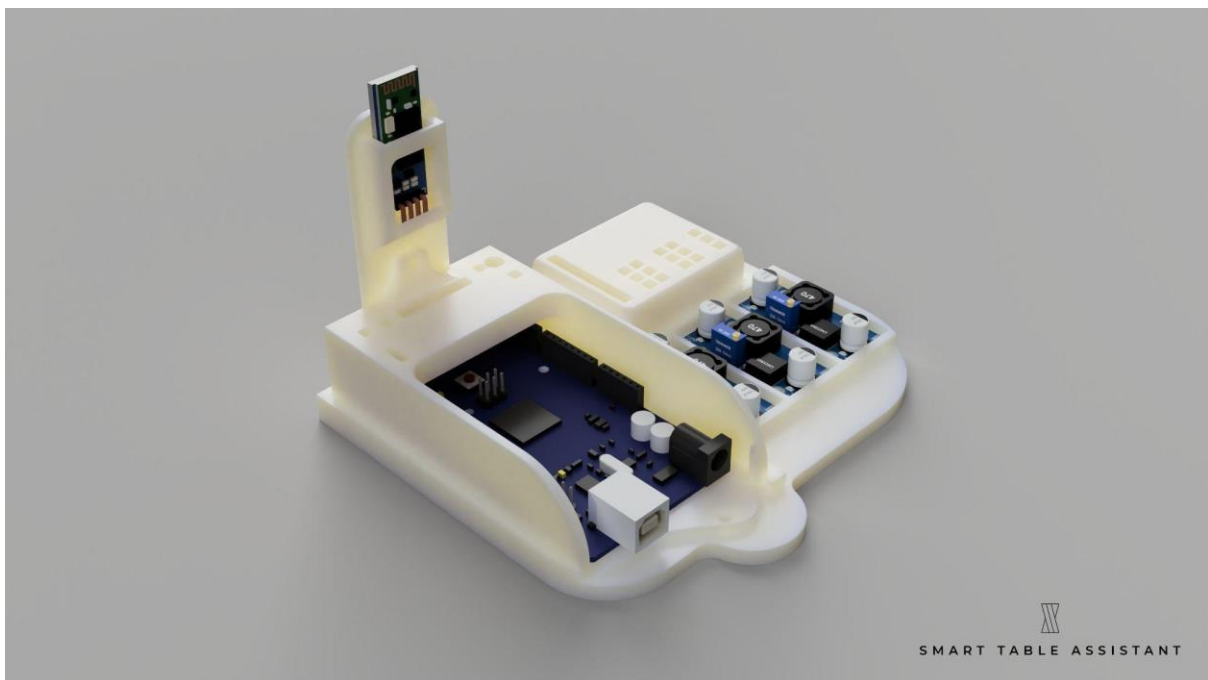


IMATGE 22. TAPA ELECTRÒNICA (PRÒPIA)

I no tan apreciable però no menys important, la base dels components electrònics i un parell de bateries per assolir la suficient potència per alimentar la taula intel·ligent. Aquesta part està pensada per poder reparar i treballar confortablement en cas d'haver de solucionar qualsevol error en el sistema. El que subjecta els components, està imprès en tres parts, on a sobre de les peces hi ha tres caixes per reposar els elements i poder treure-les individualment per la porta d'accés a l'electrònica.



IMATGE 23. CAIXA BATERIES (PRÒPIA)

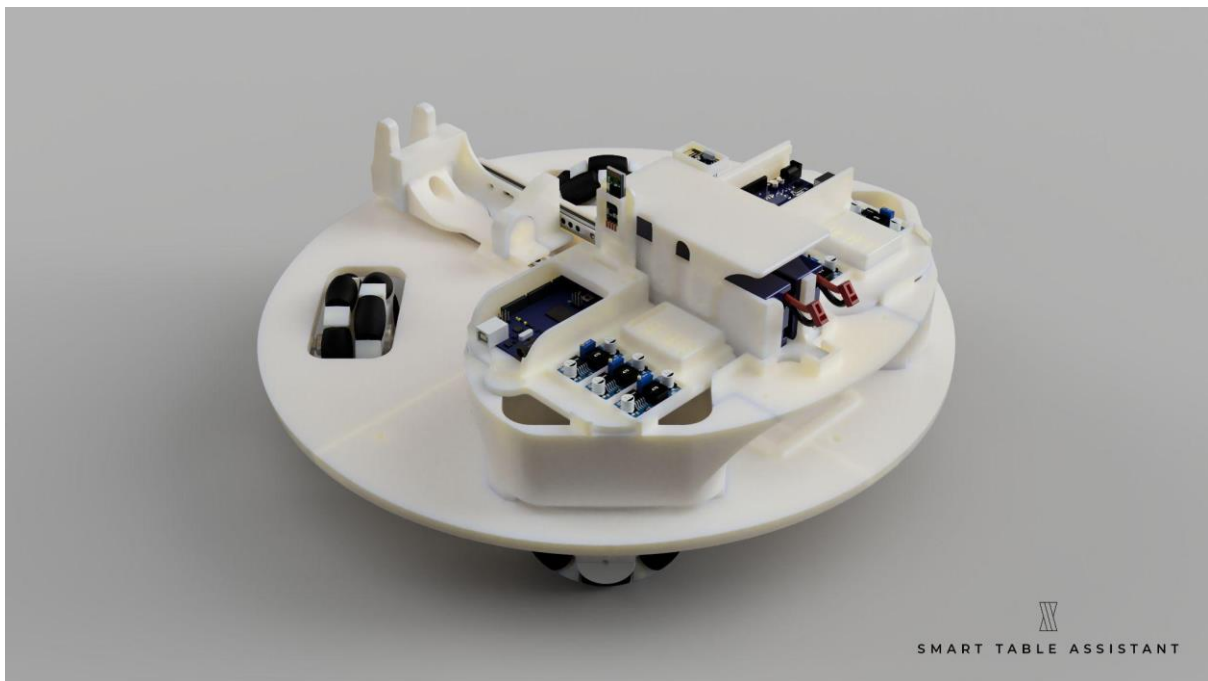


IMATGE 24. CAIXA CONTROL MOTORS PAS A PAS (PRÒPIA)





IMATGE 25. CAIXA CONTROL SERVOMOTORS (PRÒPIA)



IMATGE 26. CONJUNT DE CAIXES MOTRIU (PRÒPIA)



- Bodega (calaix del mig): Aquest conjunt segueix amb la forma dels pilars de la Sagrada Família, mantenint l'estètica del projecte. L'apartat del centre del robot, té dues funcionalitats:

· Un dels seus propòsits és la reposició de coberts, pots d'oli, vinagre, sal i espècies, postres i mantenir la temperatura dels plats elaborats.



IMATGE 27. INTERIOR BODEGA (PRÒPIA)

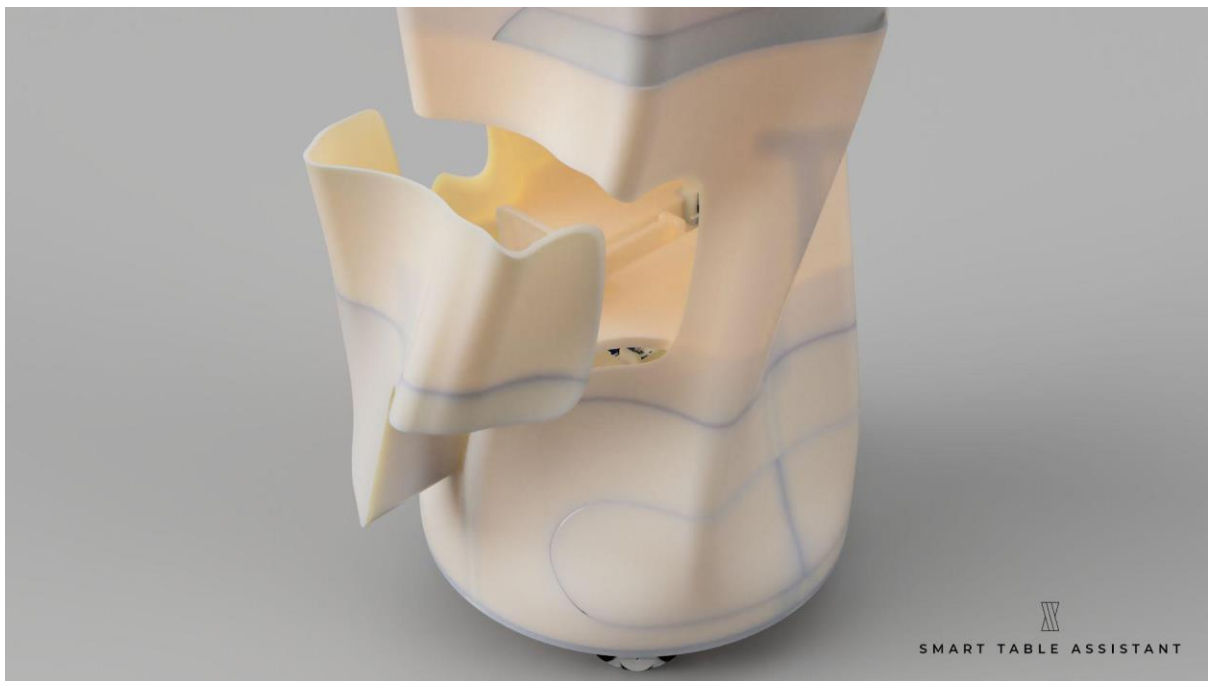
· La bodega està rumiada per poder-hi posar una ampolla de qualsevol beguda que les seves mides siguin concordants amb la preestablerta del forat. Ha estat un repte aconseguir desplaçar un cos d'aquestes dimensions en relació amb l'estètica del producte. En activar la funció de l'ampoller, es desplaça tot el conjunt cap a fora trencant la forma completament de l' Smart Table Assistant.

Per precaució i evitar problemes d'encaix a l'hora de retornar l'ampoller a la seva posició inicial, he deixat un mil·límetre de diferència en tot el perímetre de la figura.



IMATGE 28. BODEGA TANCADA (PRÒPIA)

I com a resultat podem veure que s'ha pogut dur a terme tal i com esperava.



IMATGE 29. BODEGA OBERTA (PRÒPIA)

- Taula superior: Una de les funcions principals del projecte és destinar la taula intel·ligent a persones invàlides de les extremitats inferiors, la qual cosa majoritàriament fan ús a la cadira de rodes. Per tant, en el seu domicili és preferible evitar trobar-se obstacles, i l' Smart Table Assistant proporciona estalviar-se certs mobles i generar una taula per poder-hi menjar còmodament on vulguis, adaptant-se i acomodant el client tenint-ho tot a mà.

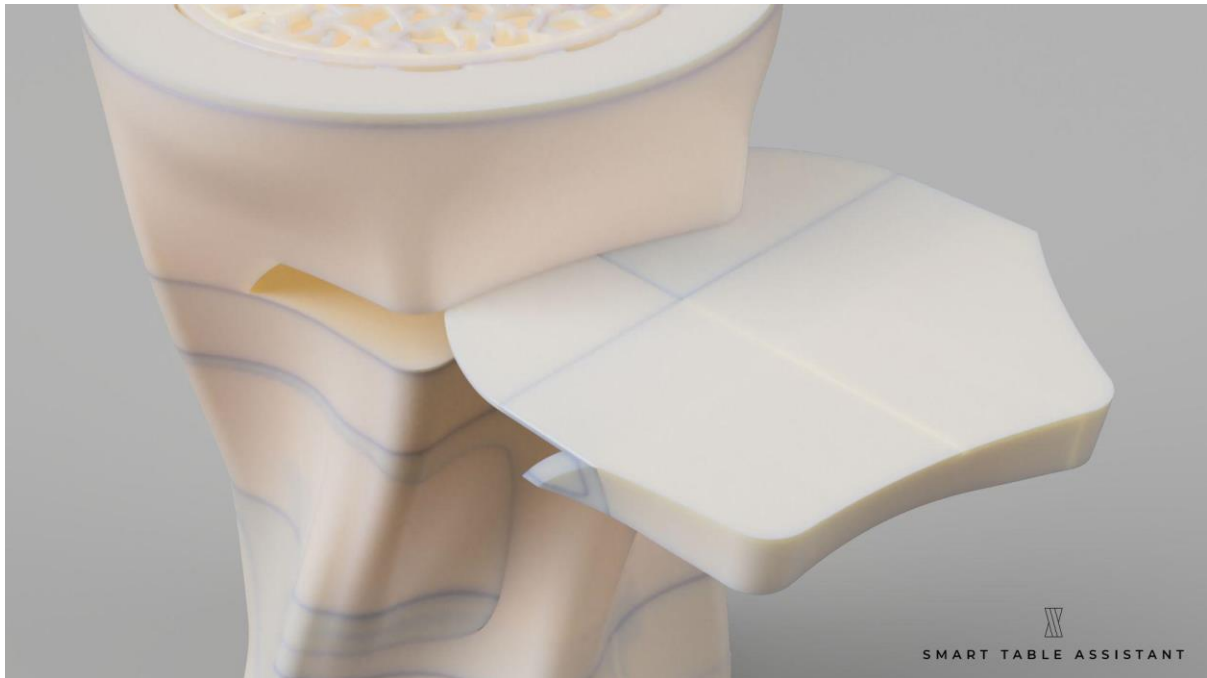
La part superior està formada per quatre peces finalitzant la torsió del pilar i tornant a la circumferència. Dues cavitats en dues d'elles, permeten el desplaçament de la taula.

Aquesta taula es retrau per facilitar la maniobrabilitat en l'accés per dins l'habitatge, reduint el seu volum.

La plataforma que s'exhibeix està pensada amb unes mides per facilitar a l'individu l'accés a la resta de funcions del projecte com la bodega, reposar correctament els peus, agafar els plats i got del mosaic i accedir als ajustos del comandament.



IMATGE 30. TAULA TANCADA (PRÒPIA)



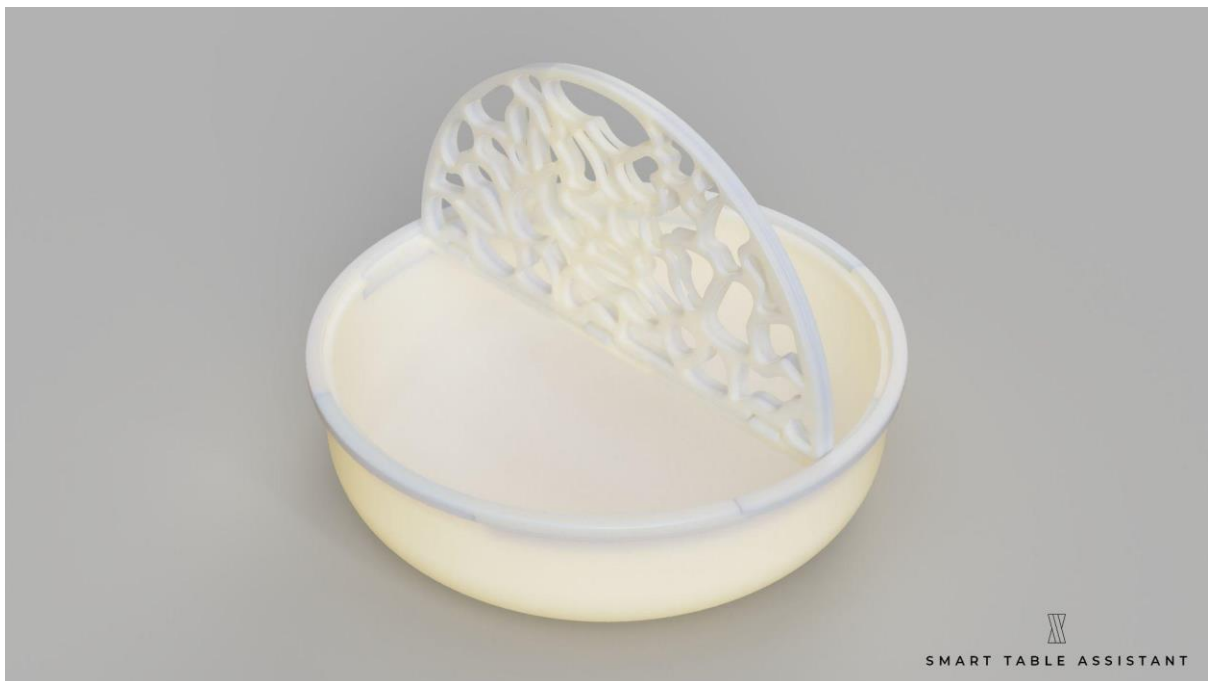
IMATGE 31. TAULA OBERTA (PRÒPIA)

- Cubell amb mosaic i nansa: La funcionalitat per la qual està pensada el cubell de la taula, és per dipositar el restant de l'àpat i facilitar la seva recollida. A diferència de la resta dels cossos, està feta de plàstic, ja que és molt possible que en algun got o plat hi quedin restes d'aigua o d'un líquid oliós amb la qual cosa el PLA es veuria afectat. També, té un altre objectiu, i aquest és que el cubell pugui ser tret agafant de la nansa que es formaria en ajuntar les dues cares del mosaic visible, per així, arribar a col·locar el cubell al rentaplats, facilitant la feina a l'usuari.

Així, com que el forat podria trencar l'estètica del producte, per cobrir-lo hi ha dues tapes articulades en forma de mosaic simètric en les dues bandes on hi ha dibuixades tres circumferències que s'han extruït per formar el relleu del seu perímetre. Així en dipositar el plat gran, el plat petit i el got evitarem l'opció que caiguin. També s'ha realitzat un rebaix amb la forma del comandament a distància per tal de millorar l'estabilitat a l'hora de deixar-lo amb repòs.



IMATGE 32. MOSAIC (PRÒPIA)



IMATGE 33. MOSAIC AMB NANSES (PRÒPIA)

## - Comandament:

El comandament de l'Smart Table Assistant està dissenyat per poder controlar el robot còmodament i tenint totes les opcions a mà. Per encendre'l es fa a partir d'un interruptor, col·locat darrere del comandament.

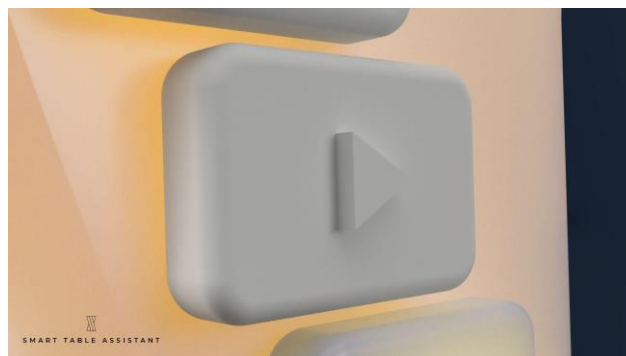
Té quatre botons a la part de davant a l'esquerra:

- El botó per gravar el recorregut que vulguis, que a diferència dels altres botons aquest s'ha de mantenir premut.



IMATGE 34. BOTÓ REC (PRÒPIA)

- El botó per iniciar el recorregut des del punt de partida fins al destinatari.



IMATGE 35. BOTÓ REPRODUIR (PRÒPIA)

- El botó per iniciar el recorregut des del destinatari fins a la seva base, és a dir, el mateix circuit però a la inversa.



IMATGE 36. BOTÓ TORNAR (PRÒPIA)

- El botó per reiniciar els recorreguts emmagatzemats a la memòria de l'Arduino, per després poder-ne grabar d'altres.



IMATGE 37. BOTÓ REINICIAR (PRÒPIA)

Al centre del control, hi ha un mòbil on en encendre'l, s'inicia l'aplicació de l' S.T.A que, a partir de l'app, podràs activar el següent:

- Els llums de colors de l'interior del robot.
- Els motors per extreure una taula automàticament, expulsar l'ampoller i tenir espai per posar els plats calents juntament amb l'aigua i, finalment, tenir lloc per posar el reposapeus de la cadira de rodes.

A la dreta del comandament, hi ha un *joystick* per poder dirigir el robot en la direcció que vulguis.



IMATGE 38. COMANDAMENT COMPLERT (PRÒPIA)



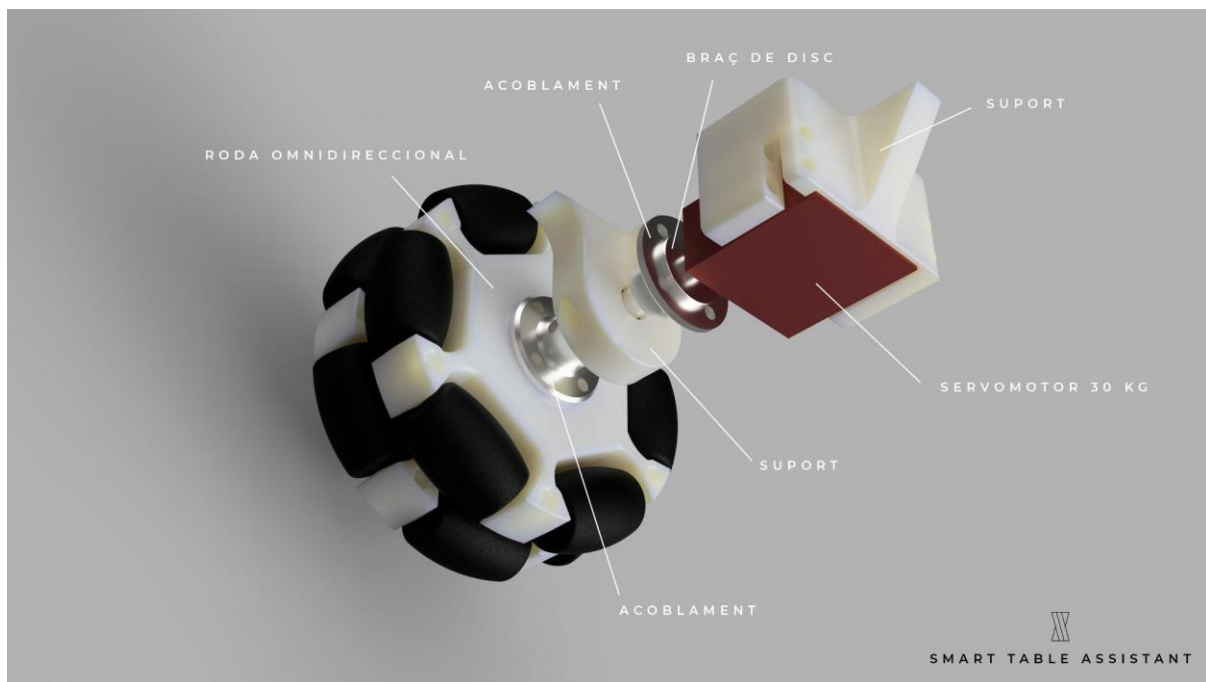
## 8.- MECANISMES

### - Estructura i moviment de les rodes omnidireccionals:

Per fer possible aquest conjunt s'han necessitat els següents materials:

- Servomotor de 360 graus capaç de suportar 30 kg amb engranatges d'alumini i un braç de 25 dents.
- Braç de disc de 25 dents.
- Acoblament per la vareta de 8mm.
- Vareta de 8mm d'alumini.
- Coixinet de forat 8mm.

Primer es posa el braç de disc encaixat a les dents del servo, i aquest s'uneix a l'acoblament que ajunta la vareta de 8mm amb el moviment del servo. Seguidament hi ha un coixinet subjectat per un suport de la plataforma impresa, per així, aguantar millor el pes del robot. Fins a arribar a la primera cara de la roda es col·loca un acoblament pensat per cargolar-lo a la mateixa roda. A l'altra banda de la roda, torna a haver-hi un acoblament per ajuntar-ho de nou, i al final de tot el sistema hi ha un coixinet que també s'aguanta per un altre suport.



IMATGE 39. MECANISME RODES OMNIDIRECCIONALS (PRÒPIA)

**· Motor pas a pas (moviment uniforme circular) a guies (moviment lineal):**

Els materials que s'han necessitat per fer aquesta transformació de moviment:

- Motor pas a pas Nema 17 bipolar.
- Guies telescòpiques de dues mides diferents.
- Acoblament de motor pas a pas a la vareta roscada.
- Rosca de coure.

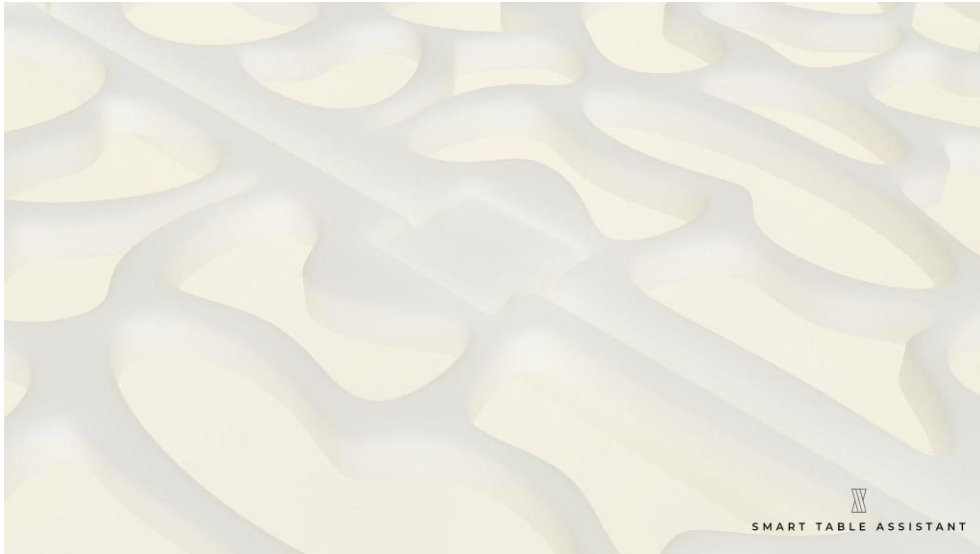
Principalment el motor pas a pas és el que fa funcionar el mecanisme. Per transformar el moviment circular en lineal s'ha necessitat posar un acoblament a l'eix del motor, per unir-lo amb la vareta. Seguidament, en ambdues bandes de la vareta de rosca, hi ha dues guies telescòpiques, cargolades a les parets perquè només es desplaci una part de la guia. Per fer moure les guies, s'han ajuntat, mitjançant una peça impresa, les dues cares de les guies amb una rosca perquè es pugui desplaçar tot el conjunt.



IMATGE 40, 41 I 42. GUIES TELESCÒPIQUES (PRÒPIA)

### • Articulacions de les nanses del cubell:

Per poder fer les nanses del cubell he dissenyat tres frontisses de llibre per després imprimir-les en 3D. Aquestes frontisses són de les més utilitzades per la seva simplicitat i funcionalitat.

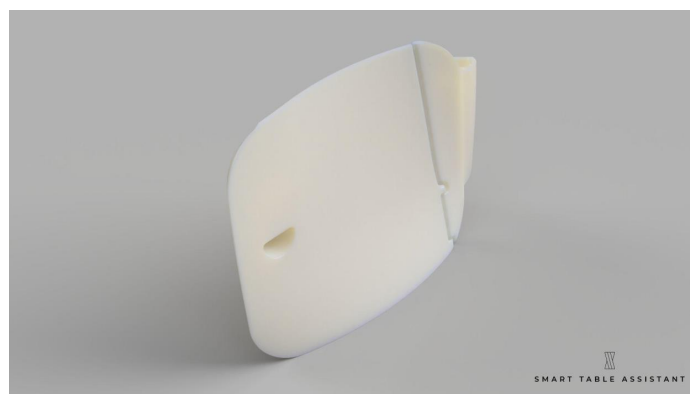


IMATGE 43. FRONTISSES (PRÒPIA)

Hi ha tres punts on se subjecten les frontisses juntament amb dues varetes d'alumini per les dues nanses i així poder-les agafar i treure tot el cubell.

### • Frontissa de la tapa:

El sistema que s'ha fet servir per poder posar una tapa a la part inferior del robot ha estat molt complicat a causa de la forma del projecte. Només s'ha necessitat una vareta fina, però era difícil poder rotar la tapa sense que toqués en algun costat, i s'ha aconseguit de la manera que es veu:



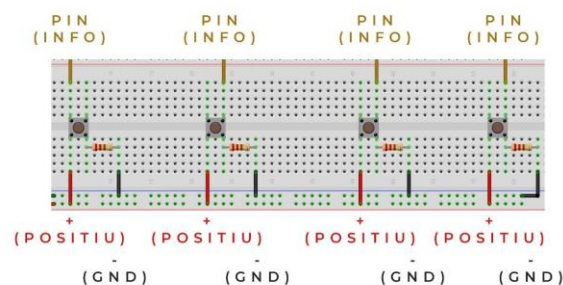
IMATGE 44. TAPA ELECTRÒNICA (PRÒPIA)

## 9.- ELECTRÒNICA

El comandament és el mestre de la comunicació remota amb relació a l' STA. Aquesta acció és possible gràcies als mòduls NRF24L01 i cadascun d'ells té una funcionalitat diferent: esclau i missatger.

### COMANDAMENT:

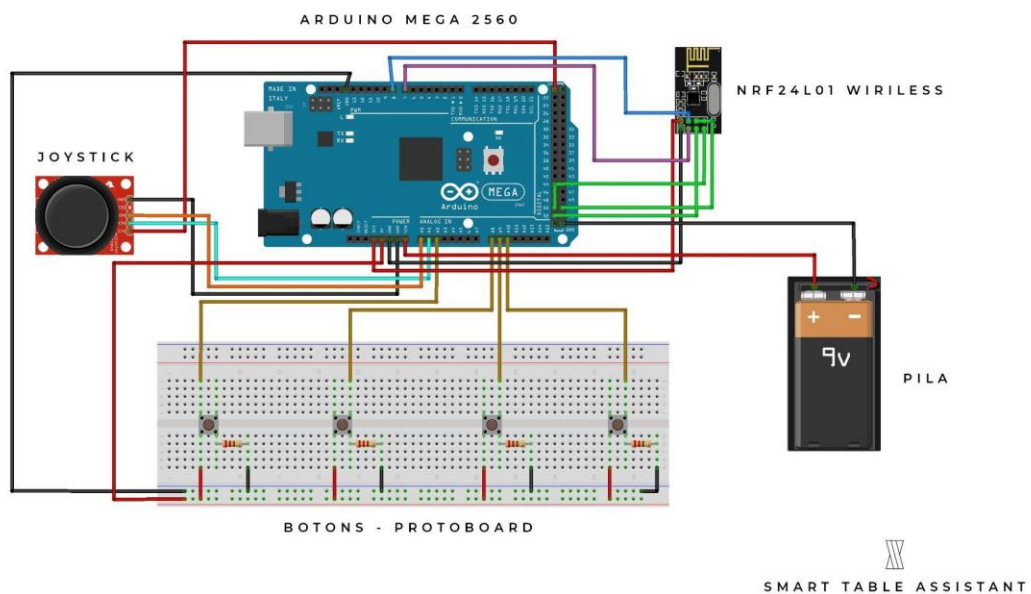
- **Joystick:** El *joystick* és la palanca de comandament de l'aparell, on depenent de quina posició estigui el receptor realitzarà les accions que hagi predeterminat. En el meu cas, he enllaçat quatre de les funcions dels pins que surten del *joystick*. El GND com bé sabem, va directament al negatiu de la placa Arduino. El pin de 5V on també, es col·loca a la Mega al pin de 5V. I els pins de les direccions x i y, estan posats als analògics A0 i al A1.
- **Protoboard amb botons:** Aquesta part per poc important que sembli, ho és molt. Qualsevol fals contacte podria enviar informació errònia. Els botons estan tots col·locats amb la seva resistència i cables corresponents. La resistència és de 10k  $\Omega$  i les potes dels botons tenen la següent estructura:



SMART TABLE ASSISTANT

IMATGE 45. SKETCH BOTONS (PRÒPIA)

- **Arduino Mega 2560:** L'Arduino Mega és un dels microcontroladors més amplis de la família de plaques Arduino. Té 54 pins digitals que funcionen com a sortida i entrada, 16 entrades analògiques, una connexió USB i una entrada per a l'alimentació de la placa entre d'altres.
- **NRF24L01 Mòdul:** És un dels mòduls de comunicació remota més utilitzats. Opera entre 2.4 GHz i 2.5 GHz, permet la comunicació simultània fins a sis dispositius i a més incorpora tota la lògica necessària per a una comunicació estable, sense errors i reenviant dades si és necessari. Destaca pel seu baix consum i baix cost.
- **Bateria:** És l'element que alimenta la placa. En el cas del comandament se n'ha utilitzat una de 9V.

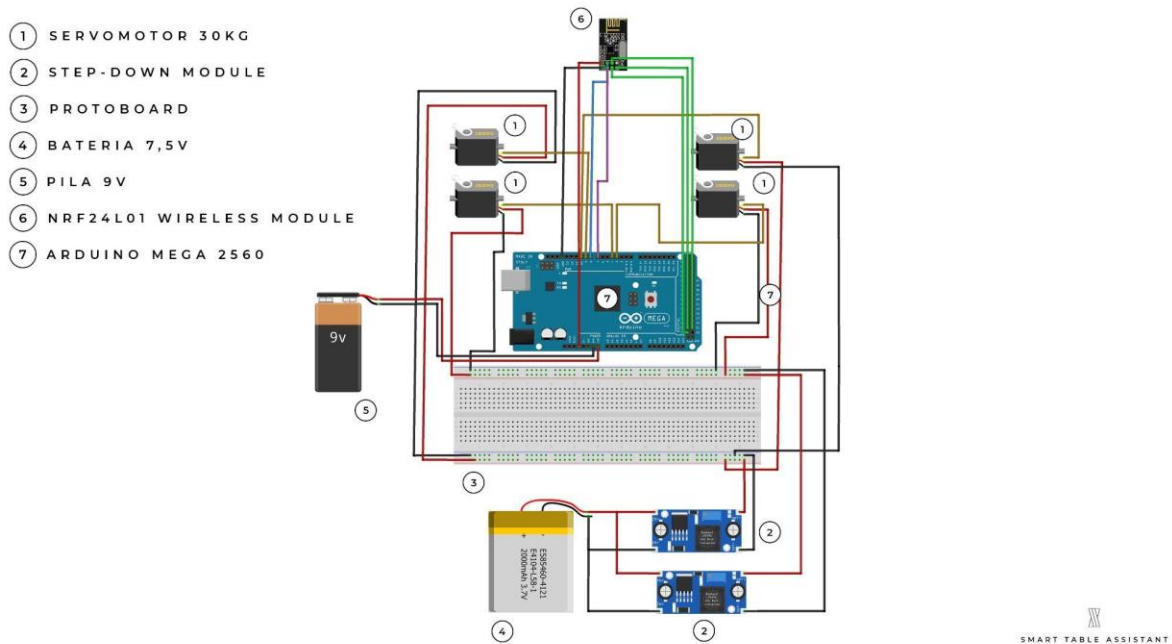


IMATGE 46. SKETCH ELECTRÒNICA COMANDAMENT (PRÒPIA)

## SMART TABLE ASSISTANT (ROBOT):

### · CAIXA SERVOMOTORS

- **Convertidors de voltatge step-down:** Tenen com a funció entregar un voltatge de sortida constant inferior al voltatge d'entrada. El voltatge de sortida se selecciona mitjançant un potenciòmetre de multivolta.
- **Convertidors de voltatge step-up:** Tenen com a funció entregar un voltatge de sortida superior al voltatge d'entrada. El voltatge de sortida es gradua mitjançant un potenciòmetre de multivolta.
- **Bateries de LiPo:** Les bateries de LiPo (Liti i Polímer) són bateries recarregables compostes per múltiples cel·les utilitzades en aplicacions que requereixen amperatge una mica elevat. Les dues bateries que alimenten l' S.T.A són de 7.5V.
- **Arduino Mega 2560:** L'Arduino Mega amb les mateixes característiques que la del comandament.
- **Servomotors:** Els servomotors utilitzats suporten 30kg per unitat, tenen resistència a l'aigua i tenen la capacitat de girar 360 graus de forma contínua. Aquests servos tenen una gran potència i són ideals per muntatges com l' Smart Table Assistant. També tenen engranatges d'alumini i un braç de 25 dents.
- **NRF24L01 Mòdul:** Aquest mòdul té les mateixes característiques que el del comandament; l'únic que canvia és la seva funció.



IMATGE 47. SKETCH ELECTRÒNICA CONTROL SERVOMOTORS (PRÒPIA)

· CAIXA MOTORS PAS A PAS:

- **Arduino CNC Shield V3:** L'Arduino CNC Shield és una petita placa que permet controlar fins a quatre motors pas a pas amb el teu propi Arduino gràcies al seu format shield. Suporta quatre controladors de potència no inclosos, i disposa de totes les connexions necessàries per connectar interruptors de final de carrera, sortides de relleu i diversos sensors. Necessita una alimentació de 12-36V.
- **Bluetooth HC-06:** Aquest mòdul ens permet connectar el projecte Arduino a un smartphone de forma remota (Bluetooth), amb la facilitat d'un port serial. La transmissió es realitza totalment de forma transparent al programador, ja que es connecta directament als pins serials del microcontrolador. Tots els paràmetres es poden configurar mitjançant els comandaments AT.
- **Arduino Mega 2560:** És la mateixa placa que s'ha utilitzat en els anteriors casos.
- **Motors pas a pas:** Els motors pas a pas que he utilitzat són els Nema 17 de tipus bipolar, capaços de dividir cada una de les seves revolucions en 200 passos, permetent una millor precisió. També són capaços de desenvolupar una força bastant considerable.

- **Convertidors de voltatge step-down:** Tenen com a funció entregar un voltatge de sortida constant inferior al voltatge d'entrada. El voltatge de sortida se selecciona mitjançant un potenciòmetre de multivolta.
- **Convertidors de voltatge step-up:** Tenen com a funció entregar un voltatge de sortida superior al voltatge d'entrada. El voltatge de sortida es gradua mitjançant un potenciòmetre de multivolta.
- **Bateries de LiPo:** Són les mateixes bateries que es fan servir en el sistema dels servomotors, per així quan s'encengui el robot, tot estigui alimentat per utilitzar-lo.
- **Drivers pels motors pas a pas:** Els drivers que s'han fet servir són els DRV8825. Aquest driver és un circuit que permet controlar els motors de corrent continu d'una manera molt simple. Permeten gestionar el voltatge i intensitat que s'està subministrant al motor per així controlar la velocitat de gir.



# 10.- PROGRAMACIÓ

Aquest és el programa que té la placa situada al robot, “l’esclava”. Aquesta placa rep la informació i les ordres donades des del comandament.

La primera part del programa és on es defineixen els pins que s'utilitzen, es preestableixen llibreries per alguns elements en concret i es posen noms per substituir als pins. Seguidament ve el void setup que és on es diu si un pin és d'entrada o de sortida, si volem que estigui encès o apagat des d'un inici i, en aquest cas, la informació necessària per a la comunicació sense fil dels mòduls NRF.

També podem veure que s'inicia el void loop, que és la part del programa que s'anirà repetint constantment. També definim les constants que es reben dels mòduls i totes les instruccions per dur a terme el funcionament del robot:

(S'indica què volen dir algunes instruccions del codi quan s'observa el següent: *// comentar*)

```
// S.T.A PROGRAMA ARDUINO OFICIAL //

#include <Servo.h> // incloum la llibreria per utilitzar els servomotors
#include <EEPROM.h> // incloum la llibreria per accedir a la memòria EEPROM

// llibreries necessàries per utilitzar el mòdul NRF24L01
#include <SPI.h>
#include <NRF24L01.h>
#include <NRF24.h>

NRF24 radio(7, 8); // CE, CSN // CE, CSN els quals són els pins per la transmissió analògica

const byte address[4] = "0001"; // el conducte del qual es comuniquen els dos mòduls
char receivedData[32] = ""; // configurem la direcció del receptor al que enviarem dades
int xAxis, yAxis, recordState, playState, resetState, toonazState; // introduim les dades que venen del receptor

int servoPin = 12; // anomenem el primer servo al pin 12
int servoPin2 = 10; // anomenem al segon servo al pin 10
int servoPin3 = 9; // anomenem al tercer servo al pin 9
int servoPin4 = 11; // anomenem al quart servo al pin 11

int addr; // introduim variable addr sense valor
int val1; // introduim variable sense valor inicial
int val2; // introduim variable 2 sense valor inicial
int val3; // introduim variable 3 sense valor inicial
int val4; // introduim variable 4 sense valor inicial

int direccio; // introduim variable per definir després la direcció

Servo myServo; // introduim el primer servo
Servo myServo2; // introduim el segon servo
Servo myServo3; // introduim el tercer servo
Servo myServo4; // introduim el quart servo

void setup() {
  Serial.begin(9600);

  myServo.attach(servoPin); // diguem que myServo sigui el que es llegeix del servoPin (12)
  myServo2.attach(servoPin2); // diguem que myServo2 sigui el que es llegeix del servoPin (10)
  myServo3.attach(servoPin3); // diguem que myServo3 sigui el que es llegeix del servoPin (9)
  myServo4.attach(servoPin4); // diguem que myServo4 sigui el que es llegeix del servoPin (11)

  radio.begin(); // es manténgui la comunicació analògica emesa
  radio.openReadingPipe(0, address); // configurem la mateixa direcció i així habilitem la comunicació entre els dos mòduls
  radio.setPwrLevel(SPI4_PA_MIN); // establim el nivell de l'amplificador de potència, en aquest cas al mínim
  radio.startListening(); // establim el mòdul com a receptor
}
```

SMART TABLE ASSISTANT

IMATGE 48. CODI 1 (PRÒPIA)

```

void loop()
{
  if (radio.available()) { // comprovem si hi ha dades per rebre
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    xAxis = atoi(&receivedData[0]); // dades del JOYSTICK X
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    yAxis = atoi(&receivedData[0]); // dades del JOYSTICK Y
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    recordState = atoi(&receivedData[0]); // dades del botó grabar (record)
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    playState = atoi(&receivedData[0]); // dades del botó anar (play)
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    resetState = atoi(&receivedData[0]); // dades del botó reset
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    tomanState = atoi(&receivedData[0]); // dades del botó tomanar
    delay(1);
  }

  if (resetState == 1023) // si el botó està premut
  {
    Serial.println("Borrando EEPROM, espere..."); // escrivim al serial el que hi ha entre ""
    // Ordres per al tema de la memòria EEPROM
    for (addr = 0; addr <= 512; addr++) {
      EEPROM.write(addr, 255);
    }
    addr = 0;

    Serial.println("EEPROM borrada"); // escrivim al serial el que hi ha entre ""
  }

  if (recordState == 1023) // quan el botó de grabar està premut
  {
    delay(270);
    grabaServos(servoPin, servoPin2, servoPin3, servoPin4, recordState); // executem l'ordre del void graba servos creat a part
  }

  else ( // qualsevol altre posició del botó que no realitzi cap acció
  )
}

```

## IMATGE 49. CODI 2 (PRÒPIA)

```

if (playState == 1023) // si el botó de reproduir es'ta premut
{
  delay(270);
  ejecutaServos(servoPin, servoPin2, servoPin3, servoPin4, playState); // executem l'ordre del void ejecutasservos creat a part
}

else ( // si no està premut
// presentació de totes les posicions possibles del joystick per realitzar el moviment omnidireccional, per cada punt establert de la palanca de comandament activarà el endavant dreta, endavant, etc
if (yAxis < 10) {
  if (xAxis < 10) {
    quita(servoPin, servoPin2, servoPin3, servoPin4);
  }
}
else {
  if (yAxis > 550) {
    if (xAxis > 550) {
      endavantdreta(servoPin, servoPin2, servoPin3, servoPin4);
      delay(1);
    }
  }
  else {
    if (xAxis < 470) {
      endavantesquerra(servoPin, servoPin2, servoPin3, servoPin4);
      delay(1);
    }
  }
  else {
    endavant(servoPin, servoPin2, servoPin3, servoPin4);
    delay(1);
  }
}
}
else {
  if (yAxis < 470) {
    if (xAxis > 550) {
      endarredreta(servoPin, servoPin2, servoPin3, servoPin4);
      delay(1);
    }
  }
  else {
    if (xAxis < 470) {
      endarreesquerra(servoPin, servoPin2, servoPin3, servoPin4);
      delay(1);
    }
  }
}
}

```

## IMATGE 50. CODI 3 (PRÒPIA)





```

case "e": // en el cas e
  val1 = 90; // el servo 1 estigui quiet
  val2 = 0; // el servo 2 rodi enrere
  val3 = 180; // el servo 3 rodi endavant
  val4 = 90; // el servo 4 estigui quiet
  Serial.println("endarrerejeta");
  break;

case "f": // en el cas f
  val1 = 180; // el servo 1 rodi endavant
  val2 = 90; // el servo 2 estigui quiet
  val3 = 90; // el servo 3 estigui quiet
  val4 = 0; // el servo 4 rodi enrere
  Serial.println("endarrerejeta");
  break;

case "g": // en el cas g
  val1 = 0; // el servo 1 rodi enrere
  val2 = 0; // el servo 2 rodi enrere
  val3 = 180; // el servo 3 rodi enrere
  val4 = 180; // el servo 4 rodi enrere
  Serial.println("dreta");
  break;

case "h": // en el cas h
  val1 = 180; // el servo 1 rodi endavant
  val2 = 180; // el servo 2 rodi endavant
  val3 = 0; // el servo 3 rodi enrere
  val4 = 0; // el servo 4 rodi enrere
  Serial.println("esquerra");
  break;

case "i": // en el cas i
  val1 = 90; // el servo 1 estigui quiet
  val2 = 90; // el servo 2 estigui quiet
  val3 = 90; // el servo 3 estigui quiet
  val4 = 90; // el servo 4 estigui quiet
  Serial.println("quieto");
  break;
}

myServo.write(val1);
myServo2.write(val2);
myServo3.write(val3);
myServo4.write(val4);

delay(150);
addr++;
if (addr == 812) break;

```

## IMATGE 55. CODI 8 (PRÒPIA)

```

Serial.println(" y acaba");
delay(100);
}

void grabaServo(int servoPin, int servoPin2, int servoPin3, int servoPin4, int recordState) { // aqui generem un void extern al loop i el setup, la qual cosa ens ha permès no haver de redactar tot més d'una vegada ja que ho resumim amb una paraula
  if (radio.available()) { // comprovem si hi ha dades per rebre
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    xAis = atoi(&receivedData[0]); // dades del JOYSTICK X
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    yAis = atoi(&receivedData[0]); // dades del JOYSTICK Y
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    recordState = atoi(&receivedData[0]); // dades del botó grabar (record)
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    playState = atoi(&receivedData[0]); // dades del botó anar (play)
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    resetState = atoi(&receivedData[0]); // dades del botó reset
    delay(1);
    radio.read(&receivedData, sizeof(receivedData)); // llegim i emmagatzem les dades rebudes
    tomasState = atoi(&receivedData[0]); // dades del botó tornar
    delay(1);
  }

  Serial.println("Grabando posiciones");

  // a l'hora de grabar necessitem que amb les mateixes direccions preestablertes guardi les dades a la memòria EEPROM en diferents lletres per què com hem vist abans es pupin dur a terme
  if (yAis < 10) {
    if (xAis < 10) {
      EEPROM.write(addr, '1');
    }
  }
  else {
  }
  if (yAis > 550) {
    if (xAis > 550) {
      EEPROM.write(addr, 'e');
    }
  }
  else {
    if (xAis < 470) {
      EEPROM.write(addr, 'd');
    }
  }
}

```

## IMATGE 56. CODI 9 (PRÒPIA)



```

        val4 = 180;
        Serial.println("endarrereesquerre");
        break;

    case 'g':
        val1 = 180;
        val2 = 180;
        val3 = 0;
        val4 = 0;
        Serial.println("dreta");
        break;

    case 'h':
        val1 = 0;
        val2 = 0;
        val3 = 180;
        val4 = 180;
        Serial.println("esquerre");
        break;

    case 'i':
        val1 = 90;
        val2 = 90;
        val3 = 90;
        val4 = 90;
        Serial.println("quieto");
        break;
}

myServo.write(val1);
myServo2.write(val2);
myServo3.write(val3);
myServo4.write(val4);

delay(540);
addr--;
Serial.println(addr);
if (addr == 0) break;
}

Serial.println(" y acaba");
delay(300);
}

```

SMART TABLE ASSISTANT

## IMATGE 59. CODI 12 (PRÒPIA)

```

void endavant(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'anar endavant, recull amb aquesta paraula totes les ordres que estan dins del void
    Serial.println("Endavant");
    myServo.write(0);
    myServo2.write(180);
    myServo3.write(0);
    myServo4.write(180);

    val1 = 0;
    val2 = 180;
    val3 = 0;
    val4 = 180;

    int direccio = 'a';
}

void endarere(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'anar endarere, recull amb aquesta paraula totes les ordres que estan dins del void
    Serial.println("Endarere");
    myServo.write(180);
    myServo2.write(0);
    myServo3.write(180);
    myServo4.write(0);

    val1 = 180;
    val2 = 0;
    val3 = 180;
    val4 = 0;

    int direccio = 'b';
}

void endavantdreta(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'anar endavantdreta, recull amb aquesta paraula totes les ordres que estan dins del void
    Serial.println("EndavantDreta");
    myServo.write(0);
    myServo2.write(90);
    myServo3.write(90);
    myServo4.write(180);

    val1 = 0;
    val2 = 90;
    val3 = 90;
    val4 = 180;

    int direccio = 'c';
}

```

SMART TABLE ASSISTANT

## IMATGE 60. CODI 13 (PRÒPIA)

```

void endavantesquerra(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'anar endavantesquerra, recull amb aquesta paraula totes les ordres que estan dins del void
  Serial.println("Endavantesquerra");
  myServo.write(90);
  myServo2.write(180);
  myServo3.write(0);
  myServo4.write(90);

  val1 = 90;
  val2 = 180;
  val3 = 0;
  val4 = 90;

  int direccio = 'd';
}

void endarredreta(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'anar endarredreta, recull amb aquesta paraula totes les ordres que estan dins del void
  Serial.println("endarredreta");
  myServo.write(90);
  myServo2.write(0);
  myServo3.write(180);
  myServo4.write(90);

  val1 = 90;
  val2 = 0;
  val3 = 180;
  val4 = 90;

  int direccio = 'e';
}

void endarreesquerra(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'anar endarreesquerra, recull amb aquesta paraula totes les ordres que estan dins del void
  Serial.println("endarreesquerra");
  myServo.write(180);
  myServo2.write(90);
  myServo3.write(90);
  myServo4.write(0);

  val1 = 180;
  val2 = 90;
  val3 = 90;
  val4 = 0;

  int direccio = 'e';
}

```

SMART TABLE ASSISTANT

## IMATGE 61. CODI 14 (PRÒPIA)

```

void dreta(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'anar a la dreta, recull amb aquesta paraula totes les ordres que estan dins del void
  Serial.println("Dreta");
  myServo.write(0);
  myServo2.write(0);
  myServo3.write(180);
  myServo4.write(180);

  val1 = 0;
  val2 = 0;
  val3 = 180;
  val4 = 180;

  int direccio = 'g';
}

void esquerra(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'anar a l'esquerra, recull amb aquesta paraula totes les ordres que estan dins del void
  Serial.println("Esquerra");
  myServo.write(180);
  myServo2.write(180);
  myServo3.write(0);
  myServo4.write(0);

  val1 = 180;
  val2 = 180;
  val3 = 0;
  val4 = 0;

  int direccio = 'h';
}

void quieto(int servoPin, int servoPin2, int servoPin3, int servoPin4) { // aquest és el void d'estar quiet, recull amb aquesta paraula totes les ordres que estan dins del void
  Serial.println("quieto");
  myServo.write(90);
  myServo2.write(90);
  myServo3.write(90);
  myServo4.write(90);

  val1 = 90;
  val2 = 90;
  val3 = 90;
  val4 = 90;

  int direccio = 'i';
}

```

SMART TABLE ASSISTANT

## IMATGE 62. CODI 15 (PRÒPIA)



Ara és el programa de la placa situada al comandament, “**la mestra**”. Aquesta placa envia la informació i les ordres a la placa situada al robot.

(Està tot explicat en els comentaris: *// comentari*)

```
//          S.T.A PROGRAMA ARDUINO OFICIAL -- MASTER (COMANDAMENT)          //

//llibries necessàries per utilitzar el mòdul NRF24L01
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 radio(7, 8); // CE, CSN els quals són els pins per la transmissió inalàmbrica

const byte address[6] = "00001"; // el conducte del qual es comuniquen els dos mòduls
char myData[32] = ""; // configurem la direcció del receptor al que enviarem dades
String xAxis, yAxis, recordState, playState, resetState, tornarState; // introduïm les dades que enviarem

int record = A2; // anomenem el botó de grabar al pin A2
int play = A8; // anomenem el botó de reproduir al pin A8
int reset = A9; // anomenem el botó de resetejar al pin A9
int tornar = A10; // anomenem el botó de tornar al pin A10

void setup() {

  Serial.begin(9600);

  radio.begin(); // es mantengui la comunicació inalàmbrica encesa
  radio.openWritingPipe(address); // configurem la mateixa direcció i així habilitem la comunicació entre els dos mòduls
  radio.setPALevel(RF24_PA_MIN); // establim el nivell de l'amplificador de potència, en aquest cas al mínim
  radio.stopListening(); // establim el mòdul com a transmissor

  pinMode(record, INPUT); // diguem que el botó de grabar sigui un pin d'entrada
  pinMode(play, INPUT); // diguem que el botó de reproduir sigui un pin d'entrada
  pinMode(reset, INPUT); // diguem que el botó de resetejar sigui un pin d'entrada
  pinMode(tornar, INPUT); // diguem que el botó de tornar sigui un pin d'entrada
}

void loop() {

  xAxis = analogRead(A0); // Llegeix l'eix x del joystick del pin A0
  yAxis = analogRead(A1); // Llegeix l'eix y del joystick del pin A1
  recordState = analogRead(record); // definim l'estat del botó de grabar llegint el seu propi pin
  playState = analogRead(play); // definim l'estat del botó de reproduir llegint el seu propi pin
  resetState = analogRead(reset); // definim l'estat del botó de resetejar llegint el seu propi pin
  tornarState = analogRead(tornar); // definim l'estat del botó de tornar llegint el seu propi pin
  // X value
  xAxis.toCharArray(myData, 5);
  radio.write(&myData, sizeof(myData)); // enviem les dades de l'eix x del joystick a l'altre mòdul NRF24L01
  yAxis.toCharArray(myData, 5);
  radio.write(&myData, sizeof(myData)); // enviem les dades de l'eix y del joystick a l'altre mòdul NRF24L01

  recordState.toCharArray(myData, 5);
  radio.write(&myData, sizeof(myData)); // enviem les dades del botó de grabar a l'altre mòdul NRF24L01

  playState.toCharArray(myData, 5);
  radio.write(&myData, sizeof(myData)); // enviem les dades del botó de reproduir a l'altre mòdul NRF24L01

  resetState.toCharArray(myData, 5);
  radio.write(&myData, sizeof(myData)); // enviem les dades del botó de resetejar a l'altre mòdul NRF24L01

  tornarState.toCharArray(myData, 5);
  radio.write(&myData, sizeof(myData)); // enviem les dades del botó de tornar a l'altre mòdul NRF24L01

  delay(20);
  Serial.print("recordState="); Serial.print(recordState); Serial.print(" playState=");
  Serial.print(playState); Serial.print(" resetState="); Serial.print(resetState);
  Serial.print(" tornarState="); Serial.println(tornarState);
}
}

```



## 11.- CONSTRUCCIÓ

El robot es va muntar en cinc parts. La primera part és la plataforma, on estan situades les rodes omnidireccionals i que està formada per quatre cossos. Es van encaixar les peces i es van posar les unions amb els cargols. Un cop estava tot unit, es va posar silicona calenta al cos per totes les escletxes. Seguidament es va procedir a instal·lar les rodes omnidireccionals, on cada roda és subjectada per un parell de suports impresos 3D col·locats en la posició que pertoca. Quan ja es van posar les rodes, es va seguir amb la part motriu inferior, el primer “calaix”. Consistia en encaixar les quatre peces noves per crear la forma del projecte, així com, tornar-les a unir amb els cargols. Després, com que encara hi havia espai, es va col·locar tota l'electrònica i la guia telescòpica per poder tirar endarrere els dos cossos que deixen espai pel reposapeus. Al principi el resultat era el correcte, però es va haver d'instal·lar una altra guia de les mateixes mides per fer més estable el moviment. També es va crear una peça a part per enfortir la unió dels dos cossos.

A continuació, es va posar el segon conjunt, amb el mateix procediment que el primer, però aquest era més complicat, ja que l'ampoller era de dimensions més grans i pesava més. Per això es van haver de posar dues guies més estables i una de relleu petita per evitar joc a l'hora de desplaçar-se. I aprofitant que encara no estava cobert, es va fer el forat pels cables de les llums i del motor pas a pas.

El tercer conjunt ja va ser més fàcil gràcies a la pràctica i experiència dels altres. Es va tornar a unir-ho tot amb els cargols i a rejuntar-ho amb la silicona calenta; així quedava tot més fort per qualsevol cop que hi pogués haver. En aquest conjunt s'hi situa la taula que s'extreu automàticament. Aquesta taula està formada per altres quatre peces, unides només amb silicona i tres barres d'alumini.

Finalment al cim del robot, hi ha el mosaic amb el cubell. Aquest cubell al ser d'un altre material (plàstic). No es va poder imprimir i es va tallar a mida per poder-lo encaixar al cercol que uneix les nanses amb ell mateix. A la part del centre, ambdues nanses tenen un forat a banda i banda per passar-hi una vareta roscada per cada una; així podran fer de nansa, que és l'objectiu.

## 12.- RESULTAT I FOTOGRAFIES DEL L´S.T.A



IMATGE 65, 66, 67 I 68. RESULTATS (PRÒPIA)

## 13.- COST DEL PROJECTE

	PREU PER UNITAT	UNITATS	COST
Arduino Mega 2560	14.99 €	x3	44.97 €
Bluetooth HC-06	5.33 €	x1	5.33 €
Joystick	2.49 €	x1	2.49 €
Push buttons	0.1 €	x4	0.4 €
NRF24L01 Mòdul	2.09 €	x2	4.18 €
Bateria LiPo	22.99 €	x2	45.98 €
Servomotor	12.07 €	x4	48.38 €
Step-down	2.25 €	x2	4.5 €
Step-up	2.89 €	x3	8.67 €
Arduino CNC Shield V3	7.49 €	x1	7.49 €
Driver DRV8825	2.59 €	x4	10.36 €
Vareta 8mm	10.99 €	x1	10.99 €
Motor pas a pas	15.74 €	x3	47.22 €
Cargols M3	10.99 €	x2 Packs	21.98 €
Cargols M4	12.99 €	x2 Packs	25.98 €
Llum	1.11 €	x6	6.66 €
Vareta 4mm	8.49 €	x1	8.49 €
Coixinet 8mm forat	1.26 €	x10	12.69 €
Coixinet 3mm forat	0.42 €	x96	40.32 €
Clau	8.99 €	x1 Pack	8.99 €
Bateria 9V	1.41 €	x1	1.41 €
Acoplament eix guia	1.33 €	x12	15.96 €
Braç de disc per servo	0.8 €	x4	3.2 €
Acoplament pas a pas	4.99 €	x3	14.98 €
Impressió del cos	15.67 €	x11	172.37 €
Guies telescòpiques	5.29 €	x9	47.61 €
Cubell de plàstic	4.29 €	x1	4.29 €
Samsung j5 segona mà	50.0 €	x1	50.0 €
Fundes termoretractils	8.49 €	x1 Pack	8.49 €
Interruptor	0.8 €	x2	1.6 €
Barres de silicona	0.13 €	x30	3.9 €
Cables	2.24 €	x4	8.99 €
Antiguspines	2.17 €	x1	2.17 €
<b>TOTAL</b>			<b>701.04 €</b>

IMATGE 69. COST (PRÒPIA)



## **14.- CONCLUSIONS I ASSOLIMENTS**

En principi, l'objectiu d'aquest projecte era crear una maqueta d'un robot capaç de poder facilitar i ajudar persones amb discapacitat a les extremitats inferiors. Un robot que pogués ser controlat a distància, pogués transportar plats, coberts i gots i tingués certes funcions com memoritzar el recorregut, reproduir-lo, i adaptar-se al client.

Haver fet l'Smart Table Assistant, un robot de grans dimensions i molta complexitat ha comportat moltes hores de feina, esforç i patiment. S'ha partit de poc coneixement en l'àmbit de la robòtica, disseny i impressió 3D i com conseqüència d'això, he hagut d'anar aprenent al llarg de fer tot el projecte. La programació ha estat tot un repte, realitzar aquest programa ha estat molt complicat, ja que a internet només trobava les bases per fer funcionar cada element per individual o un conjunt concret, i relacionar la comunicació remota, memòria EEPROM pròpia de l'Arduino, servomotors i motors pas a pas no ha estat fàcil. Haver d'imprimir totes les peces ha provocat un desgast a les dues impressores, havent de canviar certs components de l'extrusor per millorar l'acabat del producte. Durant el procés del disseny he tingut molts entrebancs, s'ha hagut de lluitar per solucionar els errors i fer possible l'encaix de tots els cossos. En resum, el procés de dur a terme l' S.T.A ha portat molta dedicació.

Poder haver creat el meu propi robot, estar en la majoria d'etapes de creació d'un projecte, m'ha ensenyat a valorar la feina que hi ha darrere del desenvolupament d'un producte al mercat. Assumir els fracassos i celebrar quan assoleixes el teu objectiu és bàsic en el camí d'aconseguir l'èxit.

Em sento orgullós, ja que he pogut fer possible tot els objectius que m'havia marcat en un principi. La gravació del recorregut va genial, el control del robot funciona perfectament i m'agrada com ha quedat el disseny de la taula. Com sempre es podrien millorar molts aspectes del robot, com posar sensors ultrasons, disminuir el soroll dels motors i millorar la seva estabilitat, però, en general em sento satisfet i crec que ha estat un èxit.

## **15.- AGRAÏMENTS**

Òbviament aquest projecte no s'hauria pogut fer sense l'ajut de certes persones. Principalment volia agrair a la meva tutora de treball de recerca, Mònica Mascort, pel seu seguiment final, constant i constructiu.

En segon lloc, agrair de tot cor a la meva família tot el seu suport emocional, econòmic i emprenedor, i sobretot al meu pare per haver estat allà en tot moment, ajudant-me en els entrebancs i seguir endavant intentant fer-ho el millor possible.

Finalment, donar gràcies a tot el professorat de l'Institut de Vidreres per facilitar-me l'aprenentatge i certa informació.

## **16.- INFOGRAFIA**

BLYNK COMMUNITY:

<https://community.blynk.cc/>

ARDUINO FÒRUM:

<https://forum.arduino.cc/>

HOW TO MECHATRONICS:

<https://howtomechatronics.com>

BITWISE AR:

<https://www.youtube.com/c/BitwiseAr>

LUIS LLAMAS:

<https://www.luisllamas.es>