



warthog
robotics

Tutorial MoveIt e Unity para Manipulação Robótica

Warthog – USP São Carlos, SP

Felipe Padula Sanches



Warthog Robotics
USP São Carlos

Tutorial MoveIt e Unity para manipulação robótica
Felipe Padula Sanches

Motivação



Motivação

- Moveit:
 - Framework de planejamento de movimento/trajetória
 - Amplamente utilizado na academia e indústria
 - Implementa diversos algoritmos
 - Integração com ROS



UR10
Universal Robots



Robonaut 2
NASA Johnson Space Center



Atlas
Boston Dynamics



KR 210
KUKA

Imagens retiradas de: <https://moveit.ros.org/robots/>



Warthog Robotics
USP São Carlos

Tutorial Moveit e Unity para manipulação robótica
Felipe Padula Sanches

Motivação

- Unity:
 - Prototipação rápida: mais de 1m de Assets na Asset store
 - Permite criar cenas com alta fidelidade visual
 - Novo solver permite melhor física
 - Integração com outras ferramentas
 - ML-Agents, OpenAI-Gym etc



Imagens retiradas de: <https://unity.com/solutions/automotive-transportation-manufacturing/robotics>



Warthog Robotics
USP São Carlos

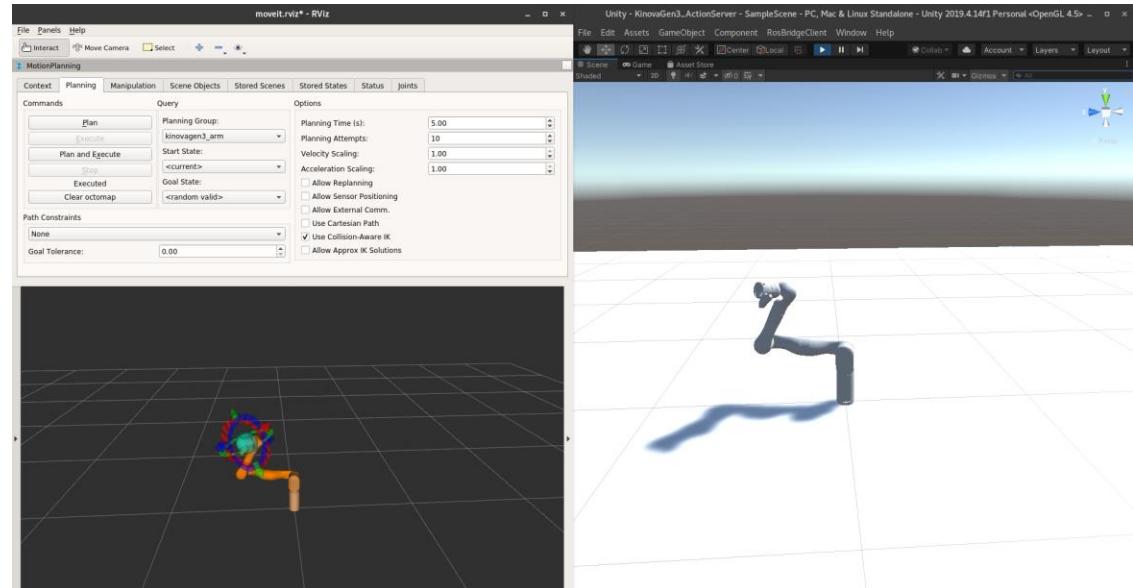
Tutorial MoveIt e Unity para manipulação robótica
Felipe Padula Sanches

Objetivo



Objetivo

- Mostrar etapas do processo de construção e configuração de uma simulação



Ementa



Ementa

1. Fundamentação teórica

2. Fusion 360

 1. Breve intro e apresentação do ambiente

 2. Criação do Mesh a partir do CAD (Kinova Gen 3)

 3. Extrair informação das juntas a partir do CAD

3. MoveIt

 1. Criação do URDF do manipulador

 2. Gerar pacote ROS através do assistente do MoveIt

 3. Configurar controlador



Ementa (cont.)

4. Unity

- 1.Carregar Meshes
- 2.Configurar juntas
- 3.Instalação do ROS#
- 4.Criação de action-server para integração com ROS e MoveIt



Fundamentação teórica



Fundamentação teórica - Manipuladores

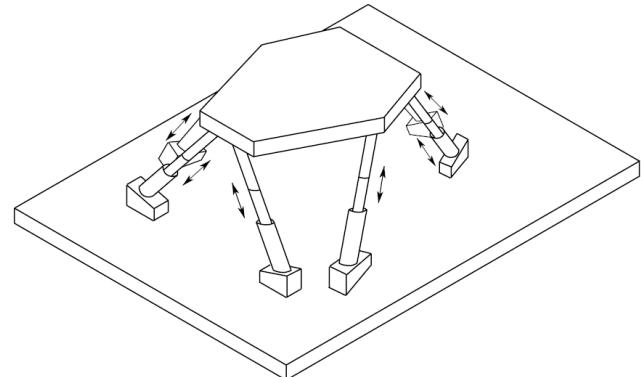
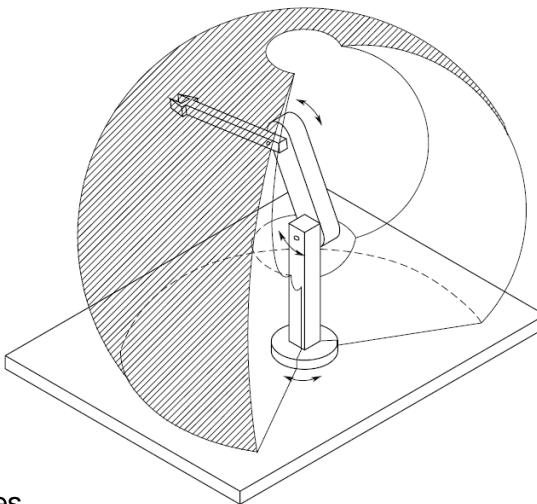
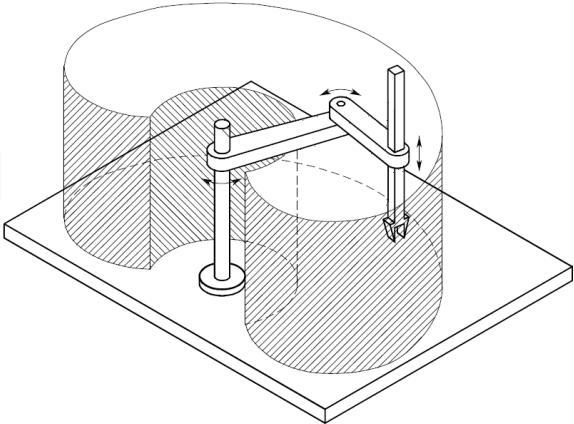


Imagen 1. Exemplos de robôs manipuladores

1.A - Robô serial tipo SCARA, um dos mais antigos exemplos de manipulador.

1.B - Robô serial tipo antropomórfico, um dos mais comum hoje em dia.

1.C - Manipulador paralelo tipo plataforma Stewart, usado para alta rigidez.

Imagens retiradas de: Siciliano, Bruno, et al. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.



Fundamentação teórica - Manipuladores

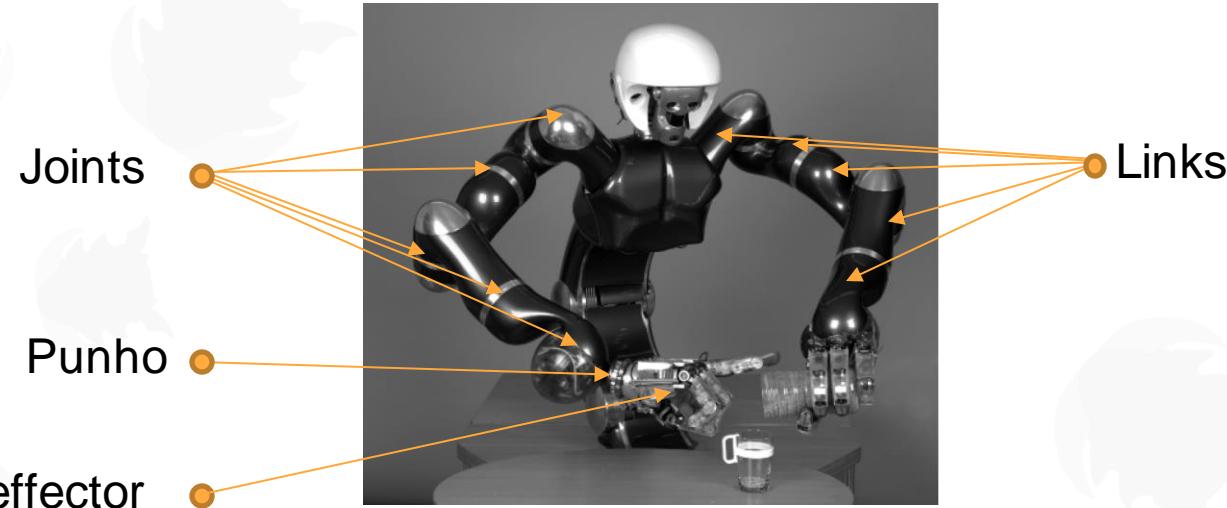


Imagen 2 - Robô humanoide Justin, desenvolvido pela DLR.

Imagens retiradas de: Siciliano, Bruno, et al. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.



Fundamentação teórica - Graus de Liberdade

Os **graus de liberdade** são variáveis livres do sistema que determinam a configuração do manipulador

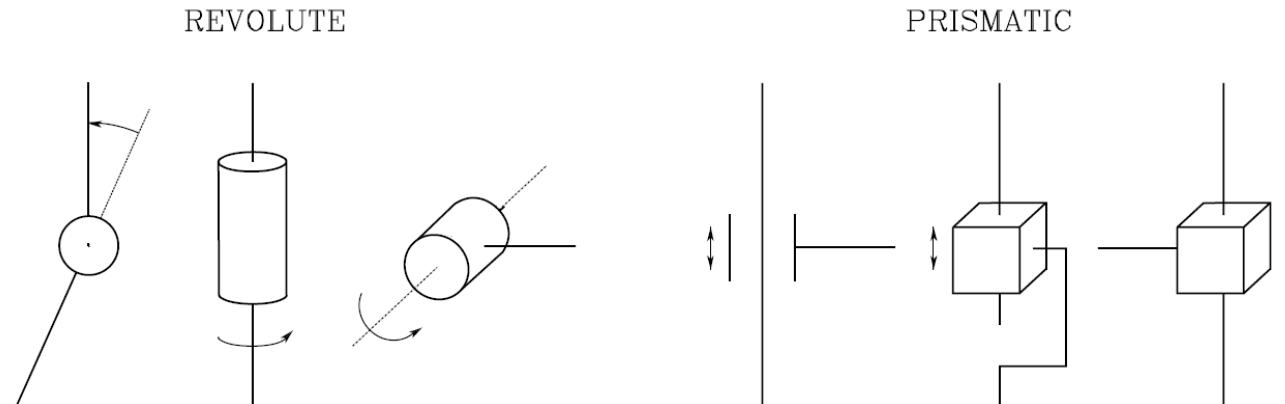


Imagen 3 - Representação típica de graus de liberdade.

Todas citações e imagem retiradas de: Siciliano, Bruno, et al. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.



Fundamentação teórica - Links

Links são estruturas que conectam as juntas

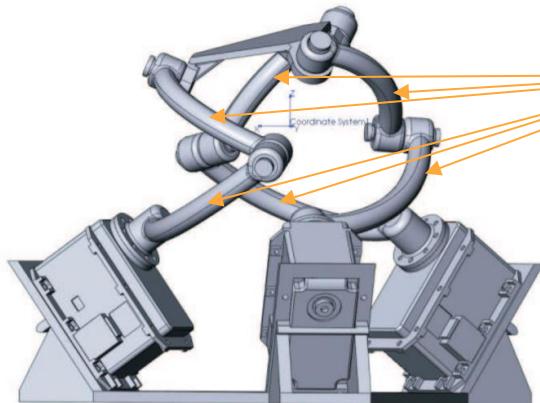


Imagen 4 - Manipulador esférico paralelo.

Niyetkaliyev, Aibek, and Almas Shintemirov. "An approach for obtaining unique kinematic solutions of a spherical parallel manipulator." 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, 2014.

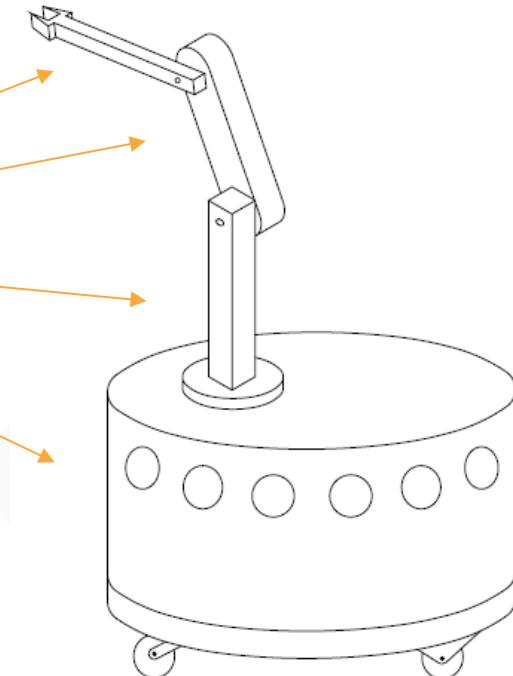


Imagen 5 - Manipulador serial em base móvel.

Siciliano, Bruno, et al. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

Fundamentação teórica - Atuadores

Atuadores são responsáveis por causar uma mudança de estado em um dos graus de liberdade (ex. motores elétricos ou pistões).

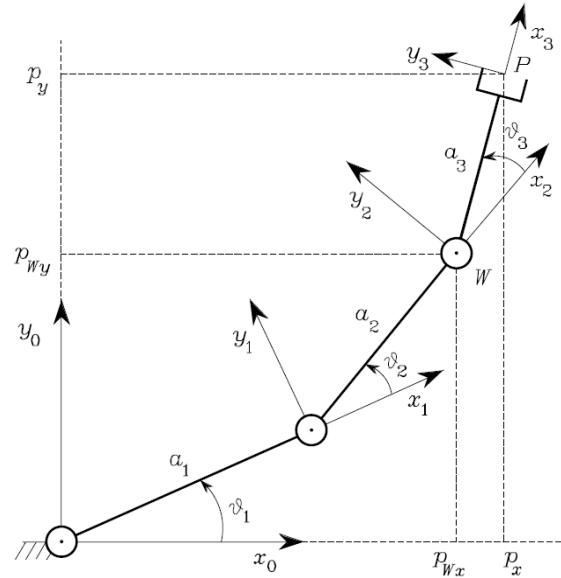
	Power density	Controllability	Actuation	Efficiency
Brushless DC	High	Medium	Hard	High
Brushed DC	Low	Low	Easy	Low
Stepper	Medium	High	Hard	Medium

Tabela 1. Comparação entre diferentes tipos de motores elétricos (atuadores)



Fundamentação teórica - Cinemática Direta

Cinemática direta é o ato de calcular a posição e orientação do "end-effector"

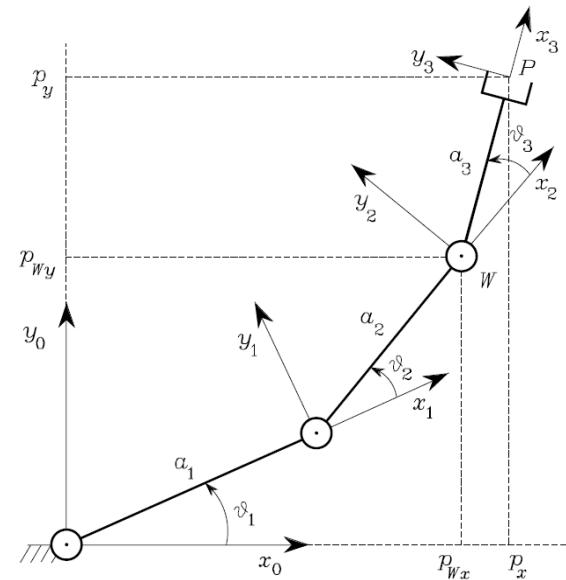


Todas citações e imagens retiradas de: Siciliano, Bruno, et al. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.



Fundamentação teórica - Cinemática Inversa

Cinemática inversa é o ato de calcular o valor de cada grau de liberdade do sistema dado a posição e ângulo do "end-effector"



Todas citações e imagens retiradas de: Siciliano, Bruno, et al. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.



Fundamentação teórica - Planejamento de Trajetória

O **Planejamento de trajetória** define o caminho que o end-effector percorrerá para sair de um ponto inicial A para um ponto B



Imagen 7 - Planejamento de trajetória discretizado



Imagen 8 - Execução da trajetória planejada por um manipulador.
Murray, S., Floyd-Jones, W., Qi, Y., Sorin, D. J., & Konidaris, G. D. (2016, June).
Robot Motion Planning on a Chip. In *Robotics: Science and Systems*.



Fusion 360



Warthog Robotics
USP São Carlos

Tutorial MoveIt e Unity para manipulação robótica
Felipe Padula Sanches

Fusion 360 - Intro

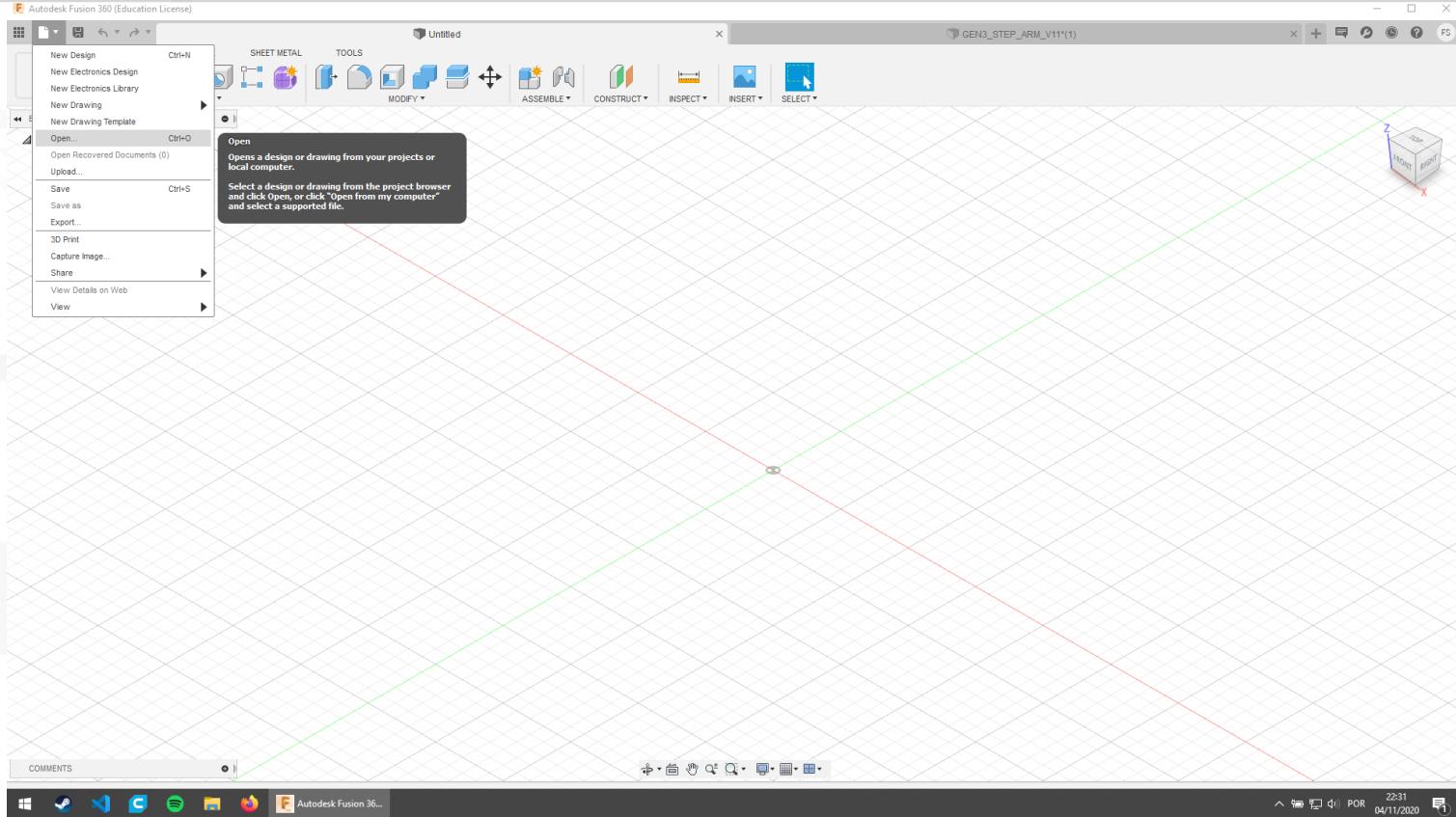


Fusion 360 - Intro

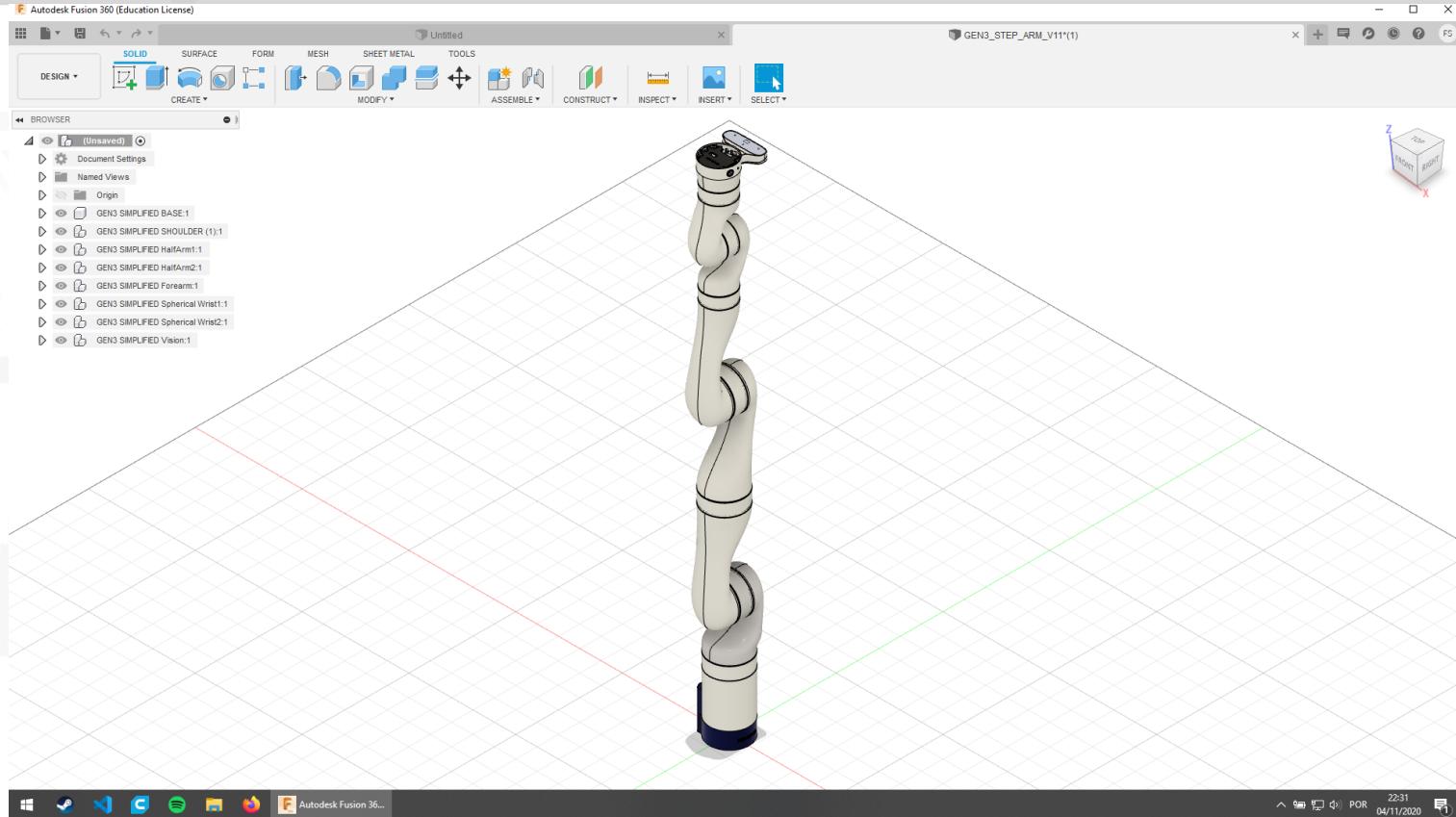
- Utilizaremos o Kinova Gen3 como caso de estudo
 - <https://www.kinovarobotics.com/en/products/gen3-robot>



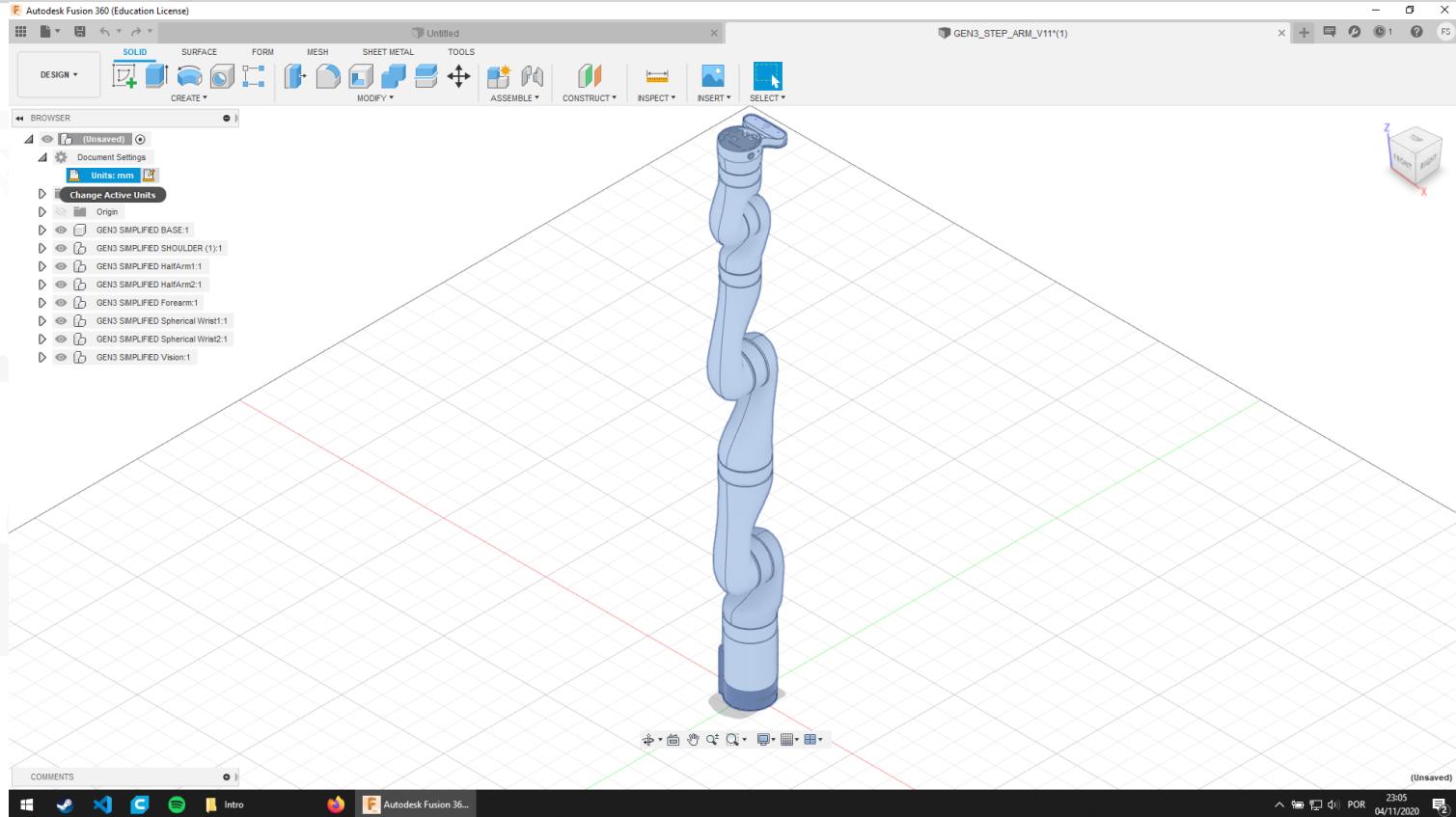
Fusion 360 - Intro



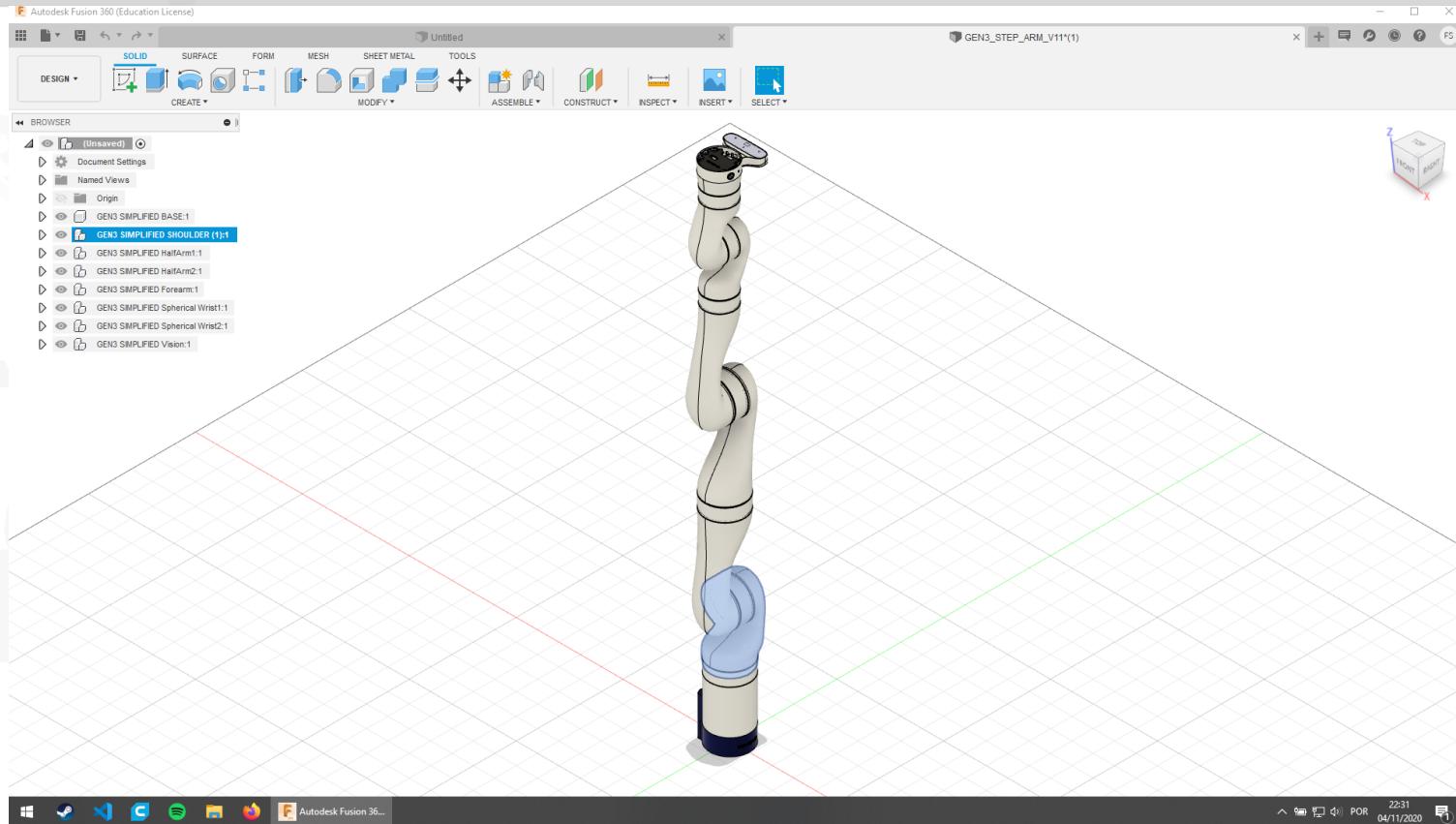
Fusion 360 - Intro



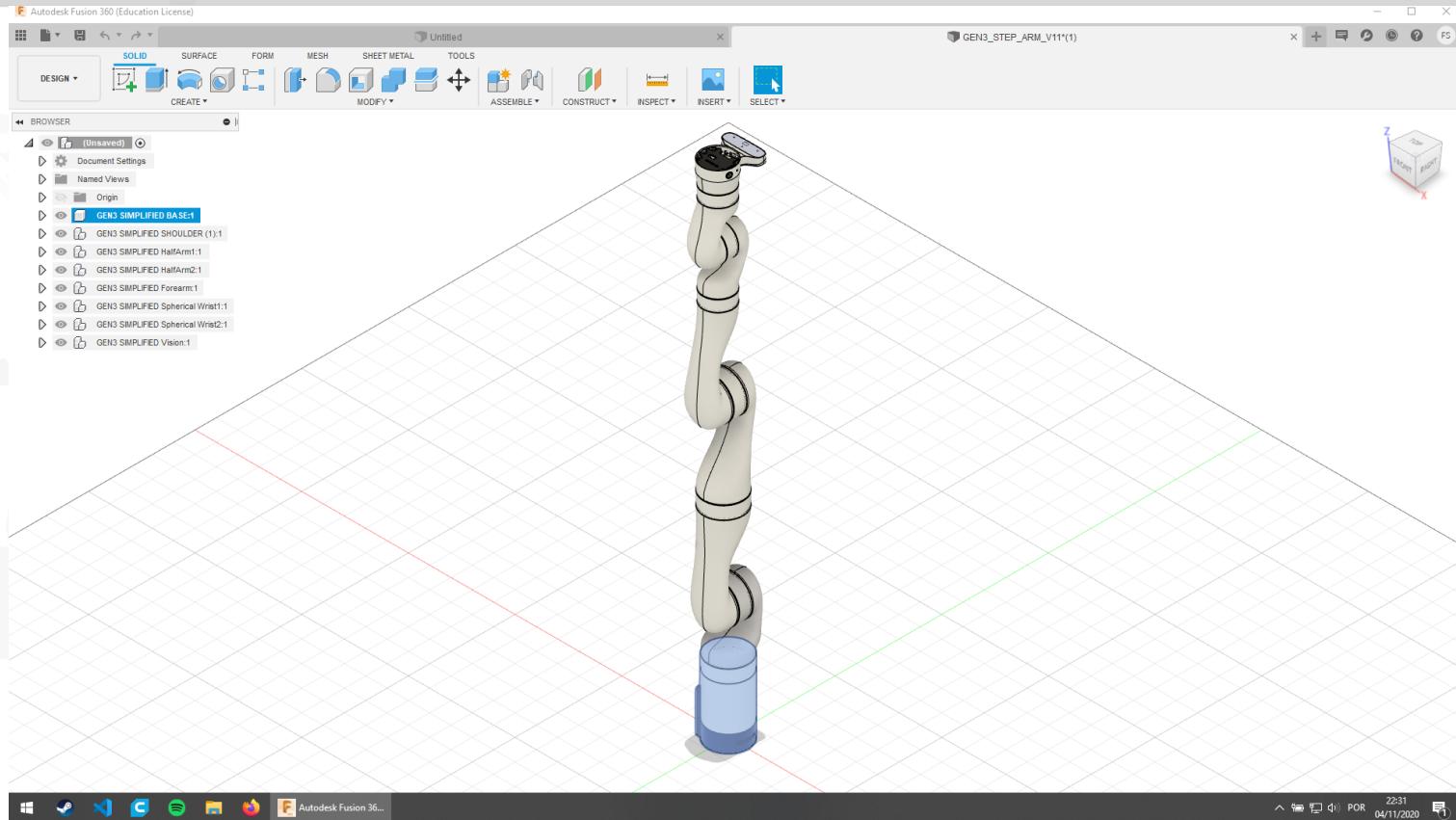
Fusion 360 - Meshes



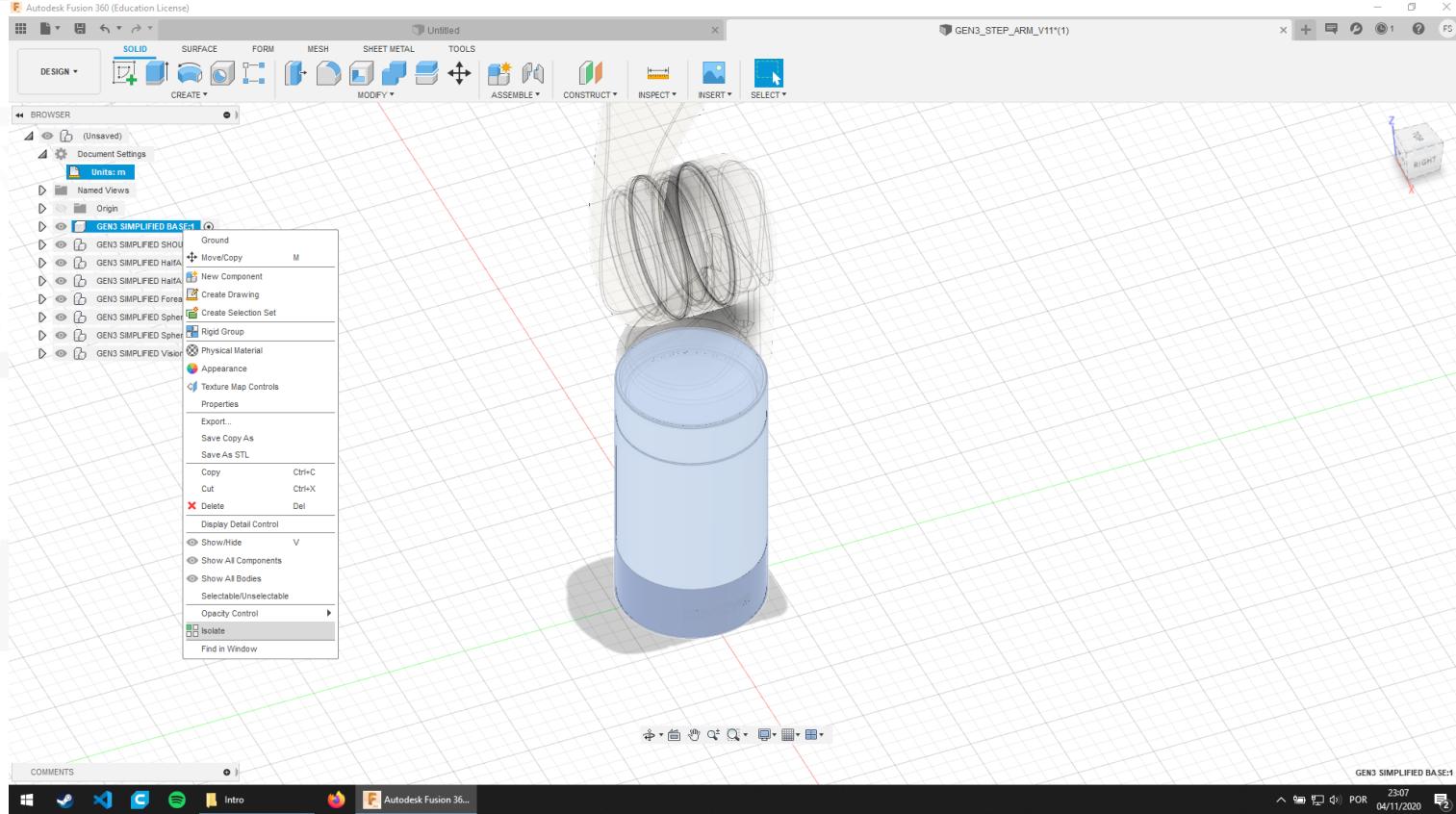
Fusion 360 - Intro



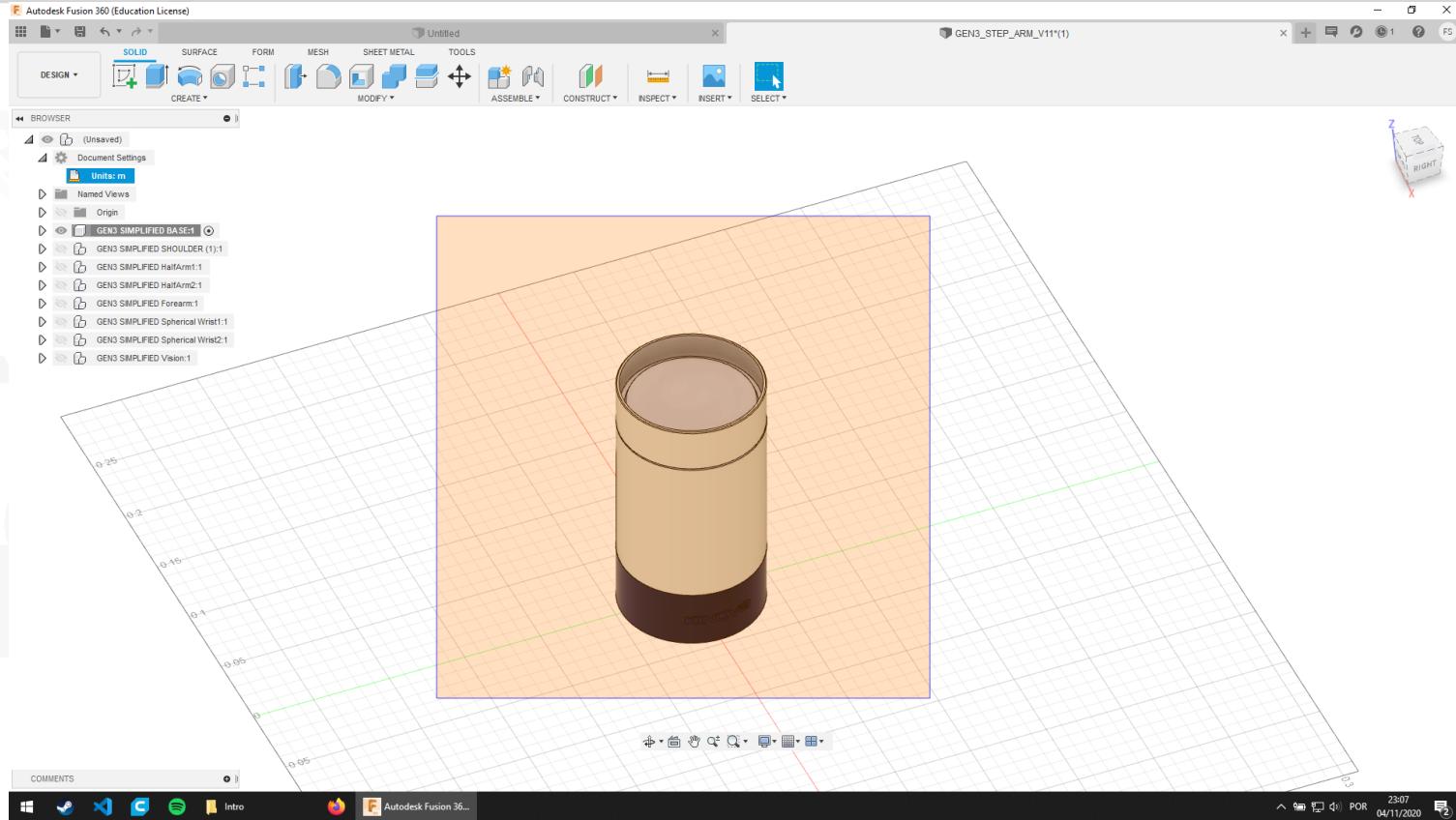
Fusion 360 - Intro



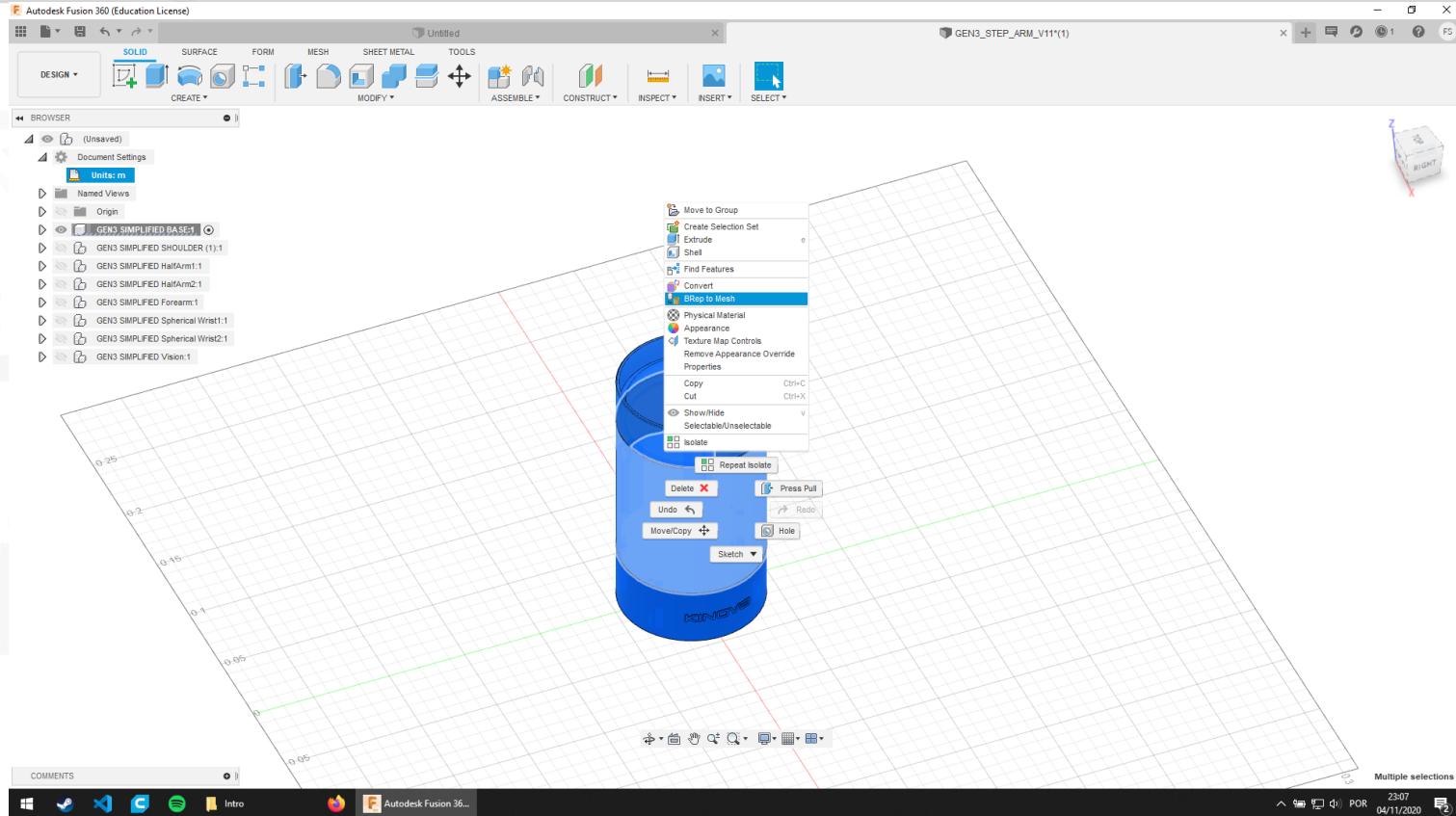
Fusion 360 - Meshes



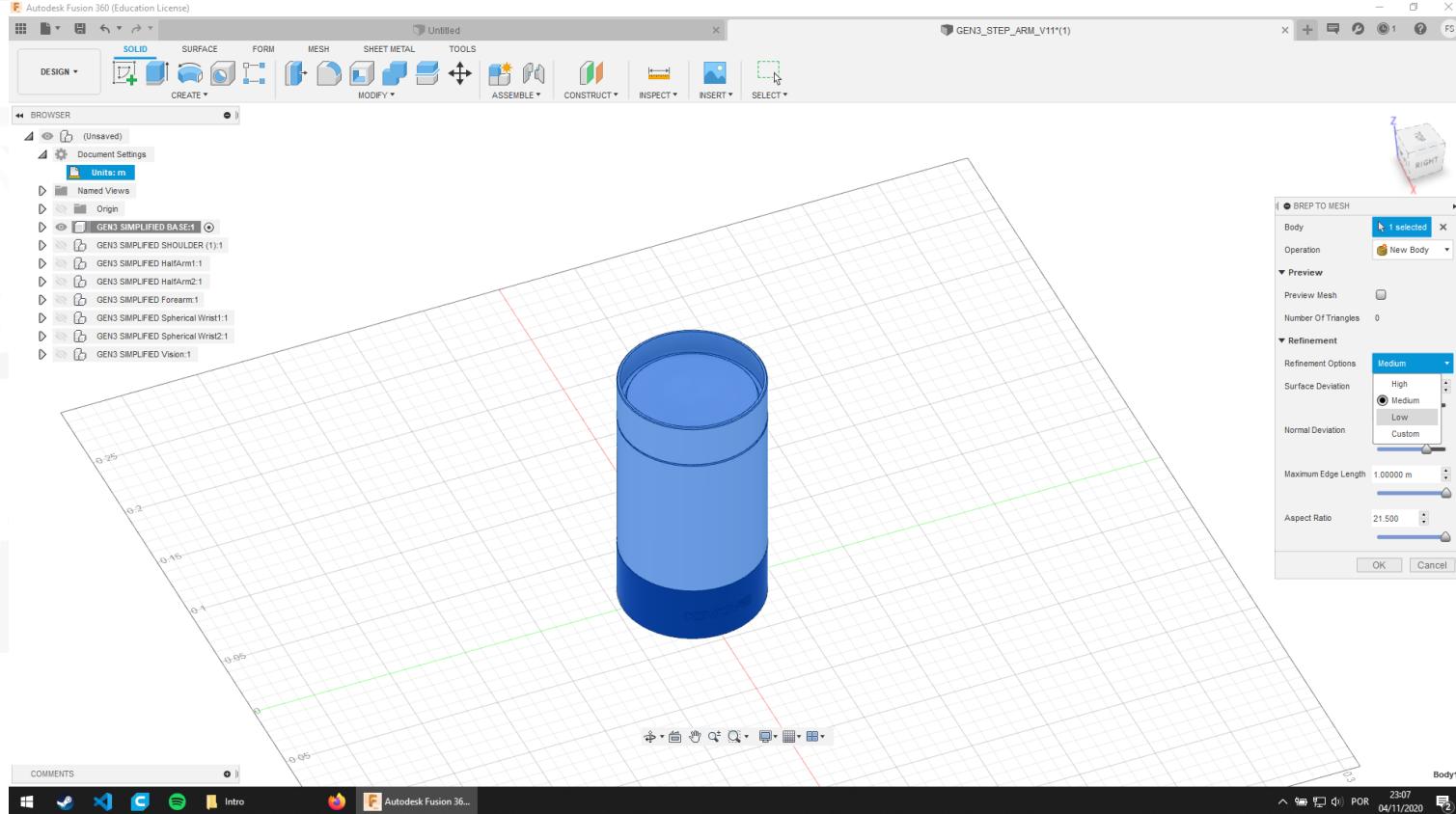
Fusion 360 - Meshes



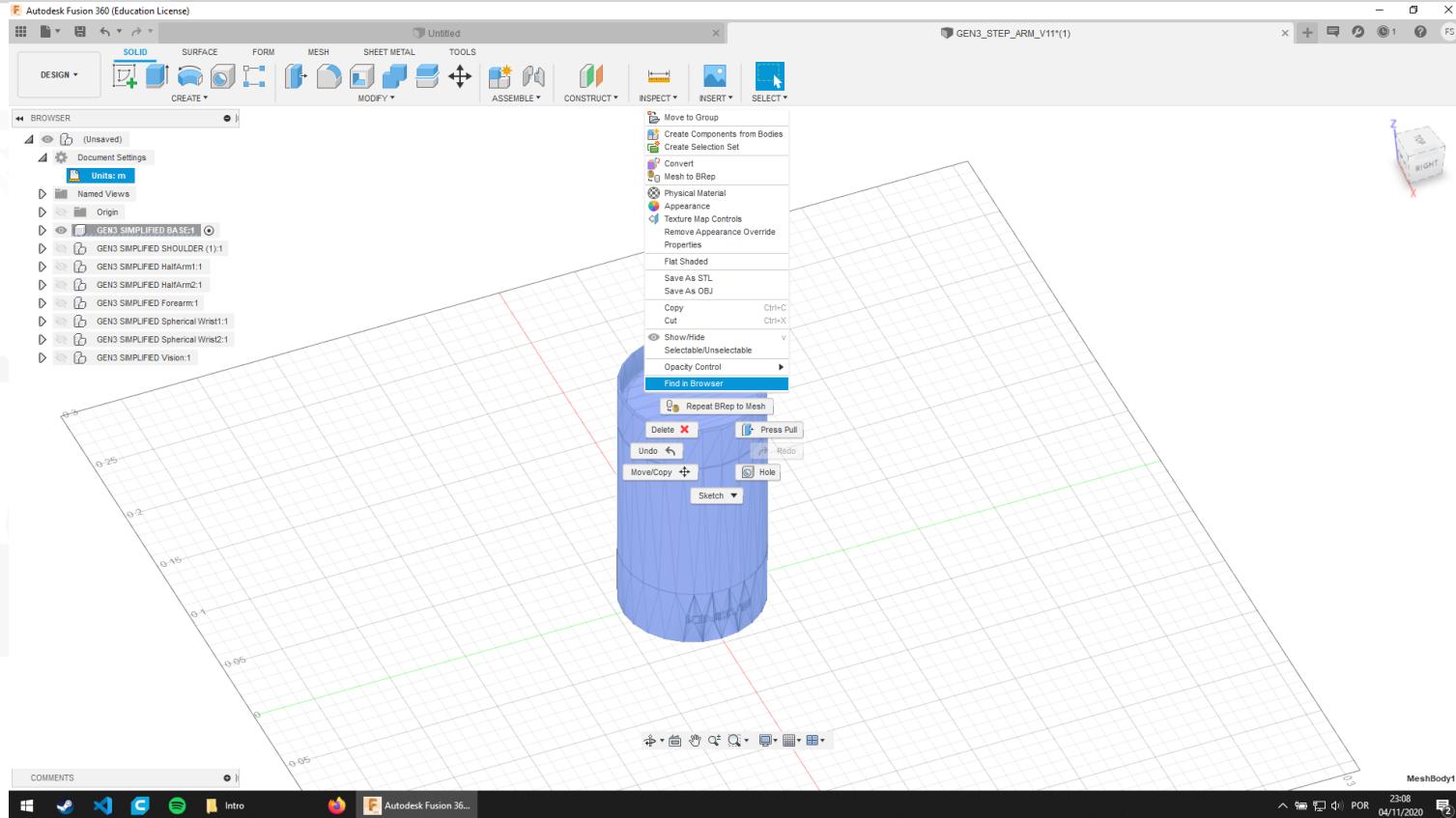
Fusion 360 - Meshes



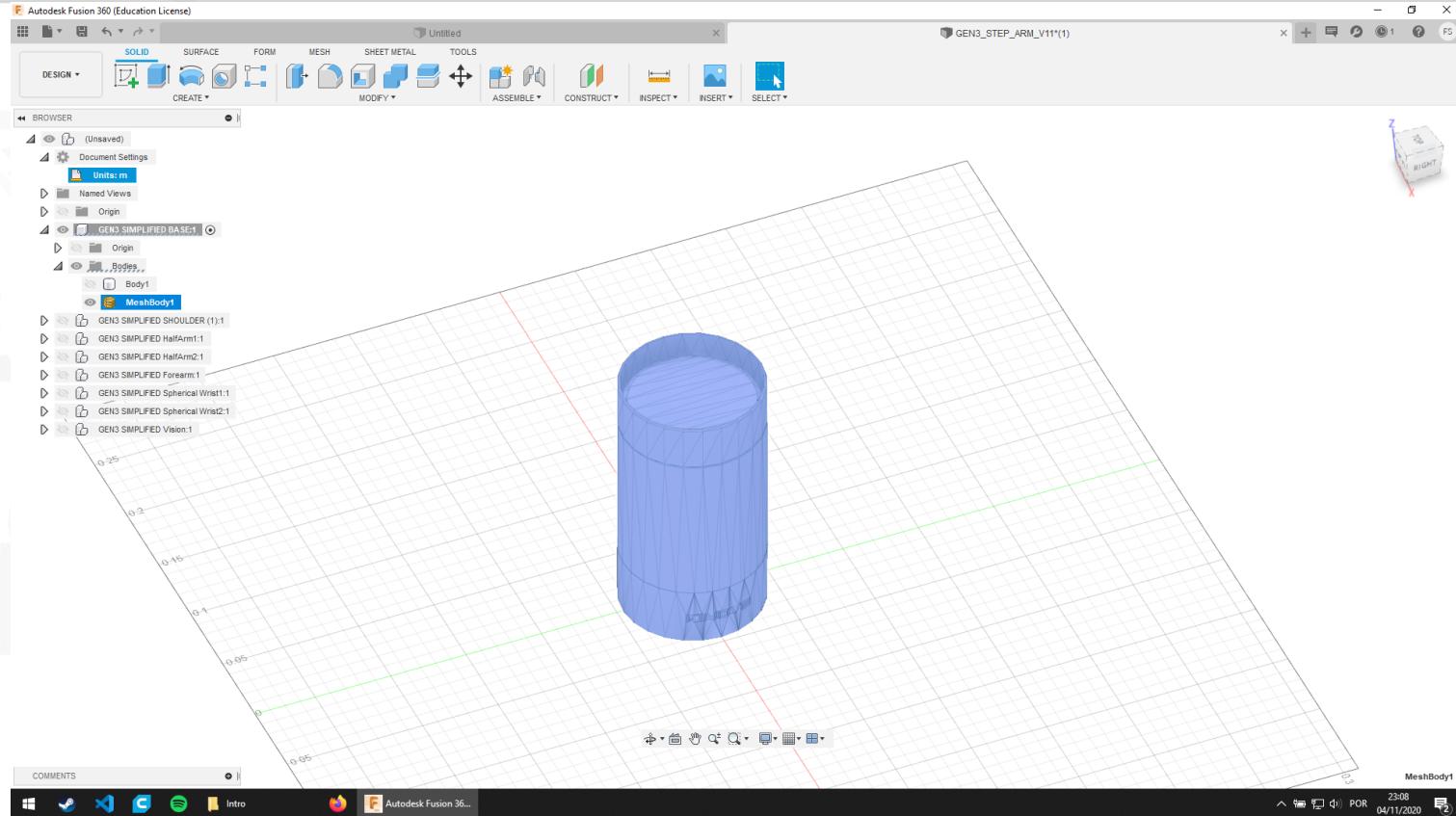
Fusion 360 - Meshes



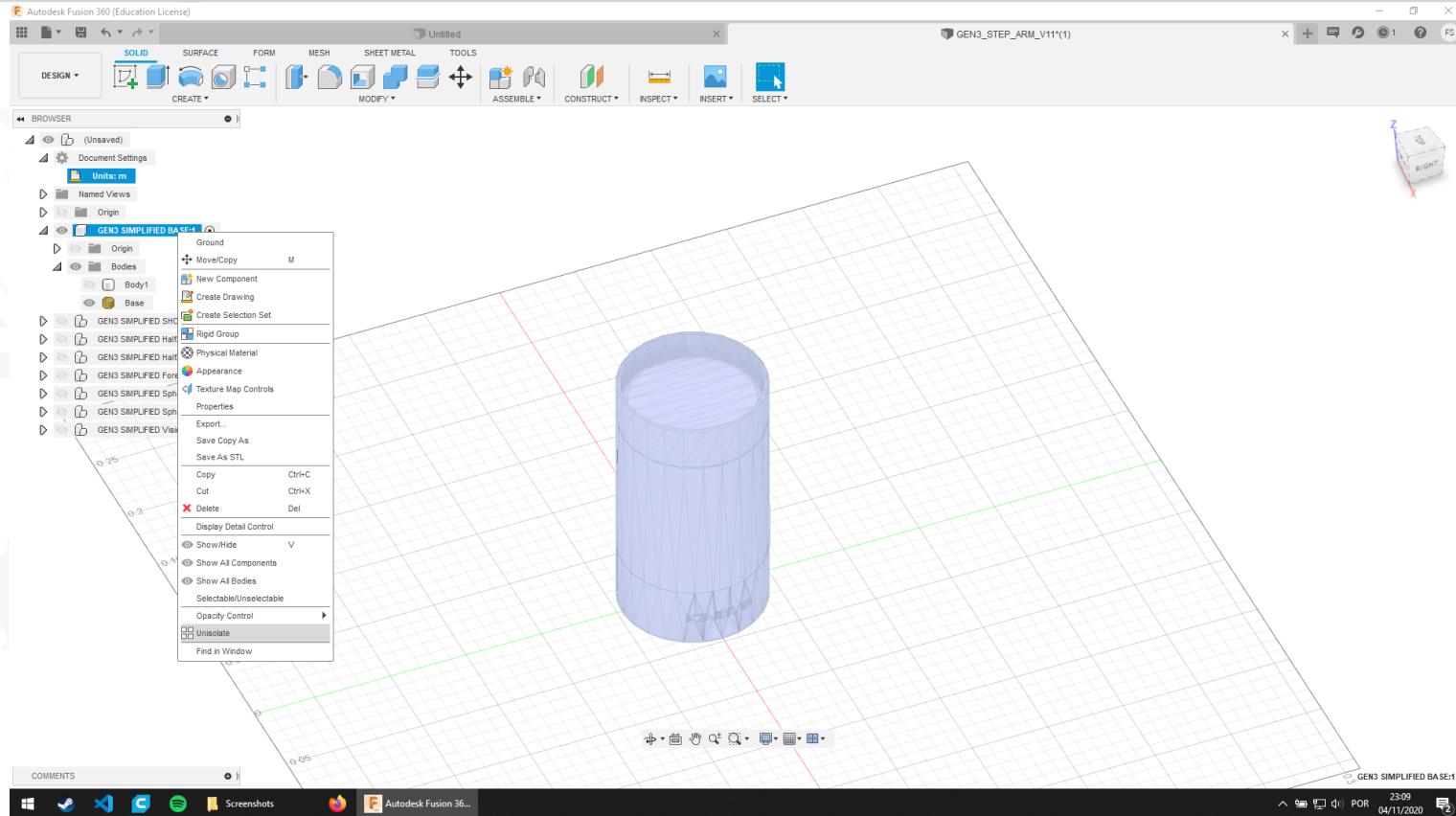
Fusion 360 - Meshes



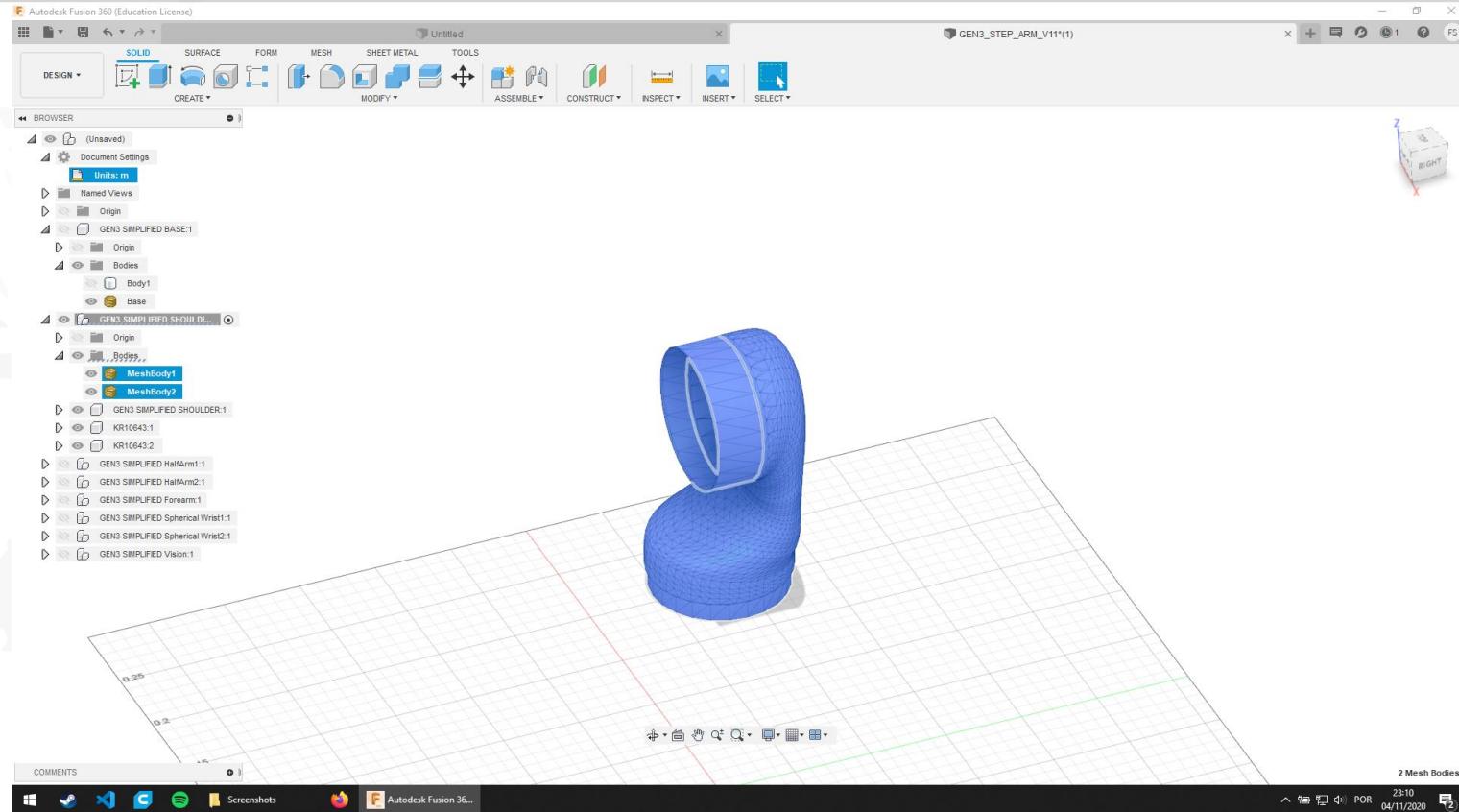
Fusion 360 - Meshes



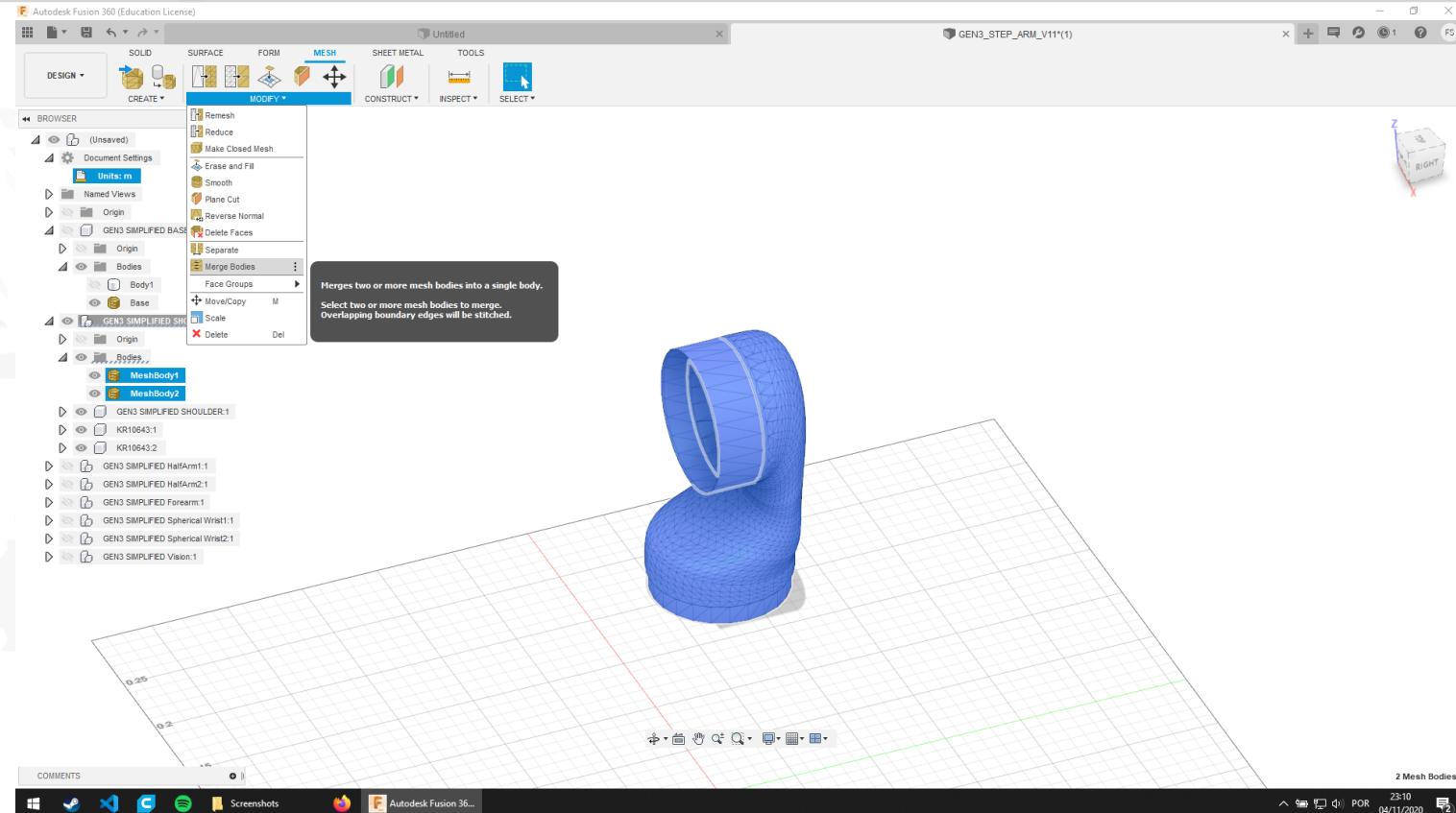
Fusion 360 - Meshes



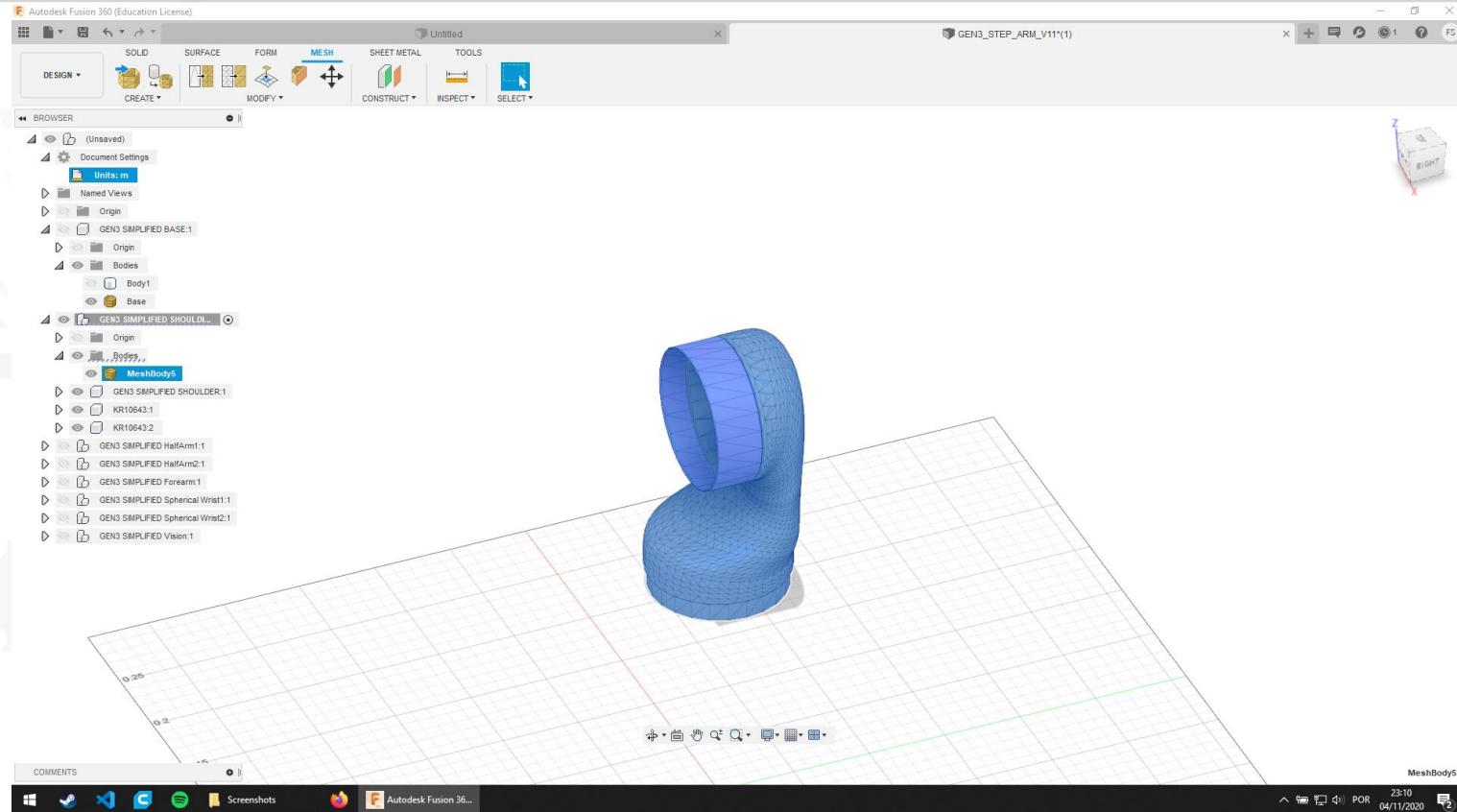
Fusion 360 - Meshes



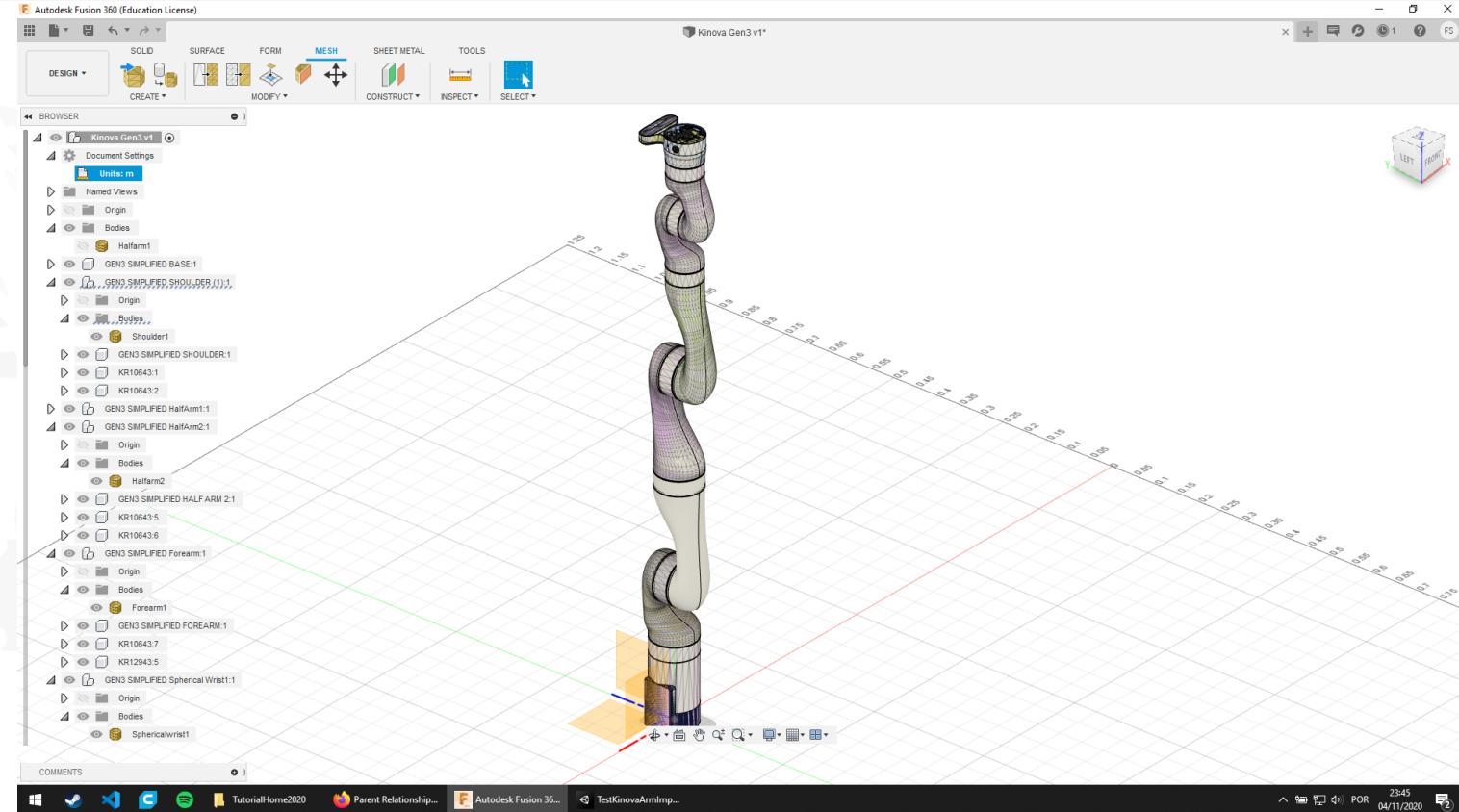
Fusion 360 - Meshes



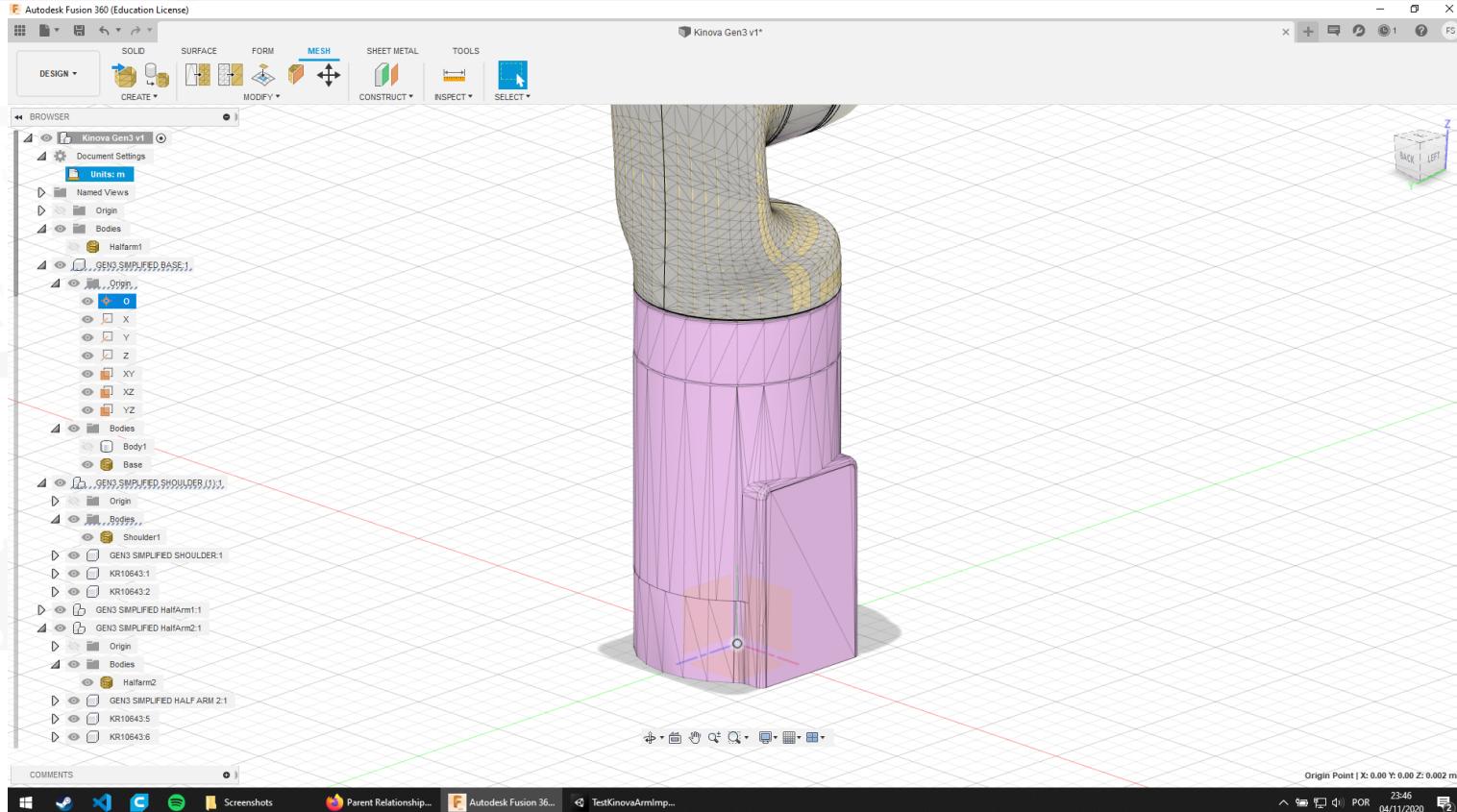
Fusion 360 - Meshes



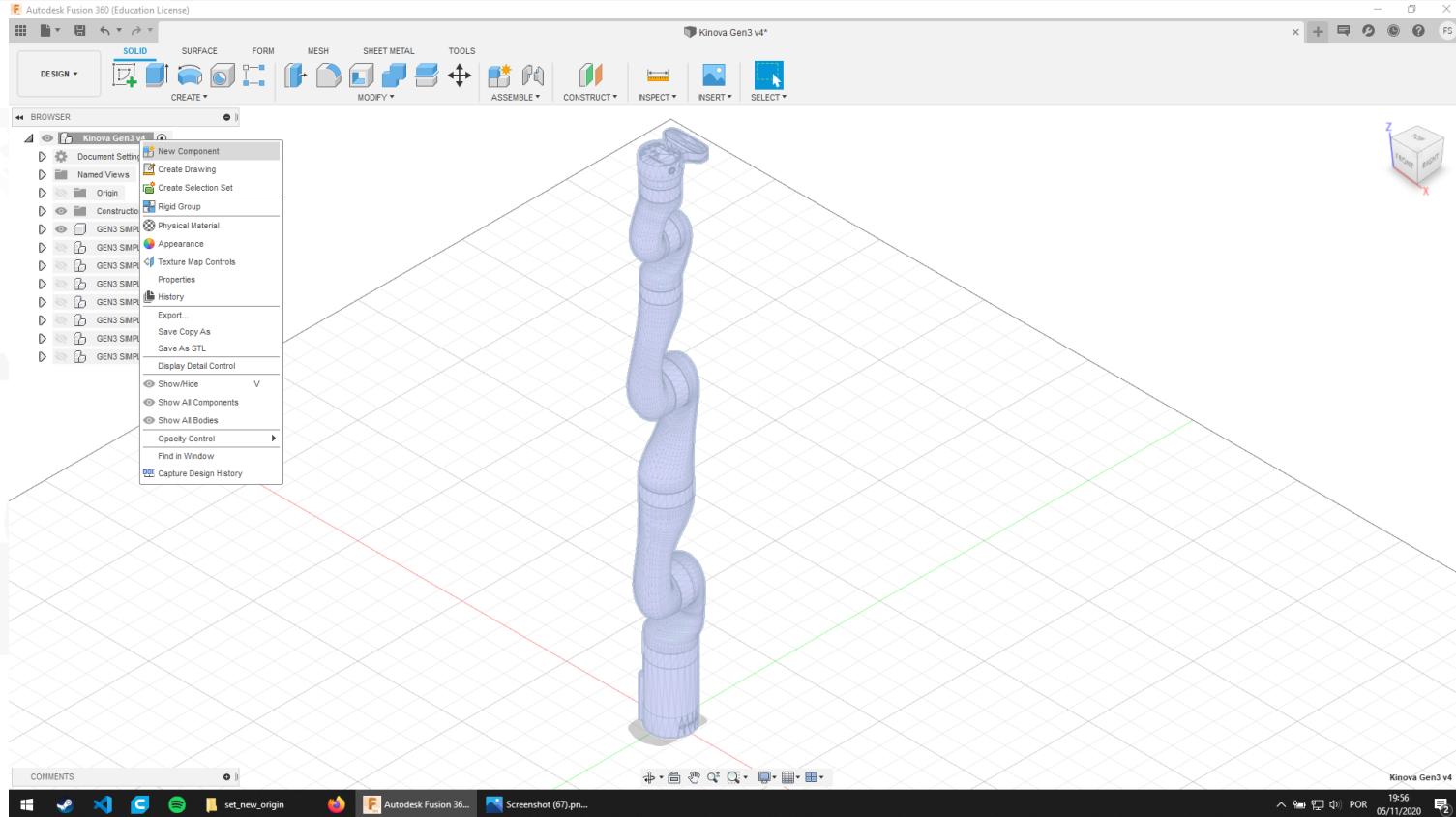
Fusion 360 - Meshes



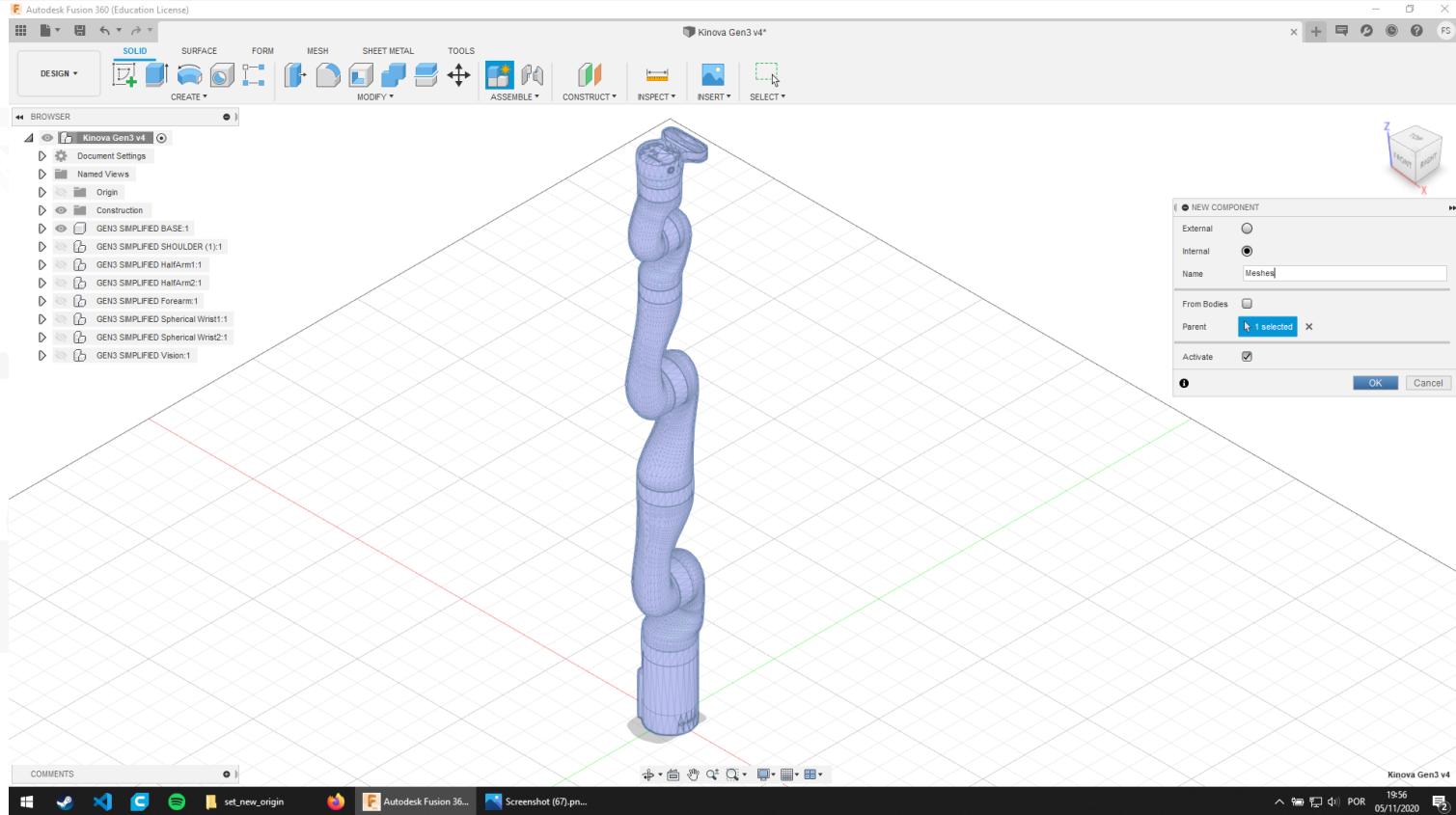
Fusion 360 - Meshes



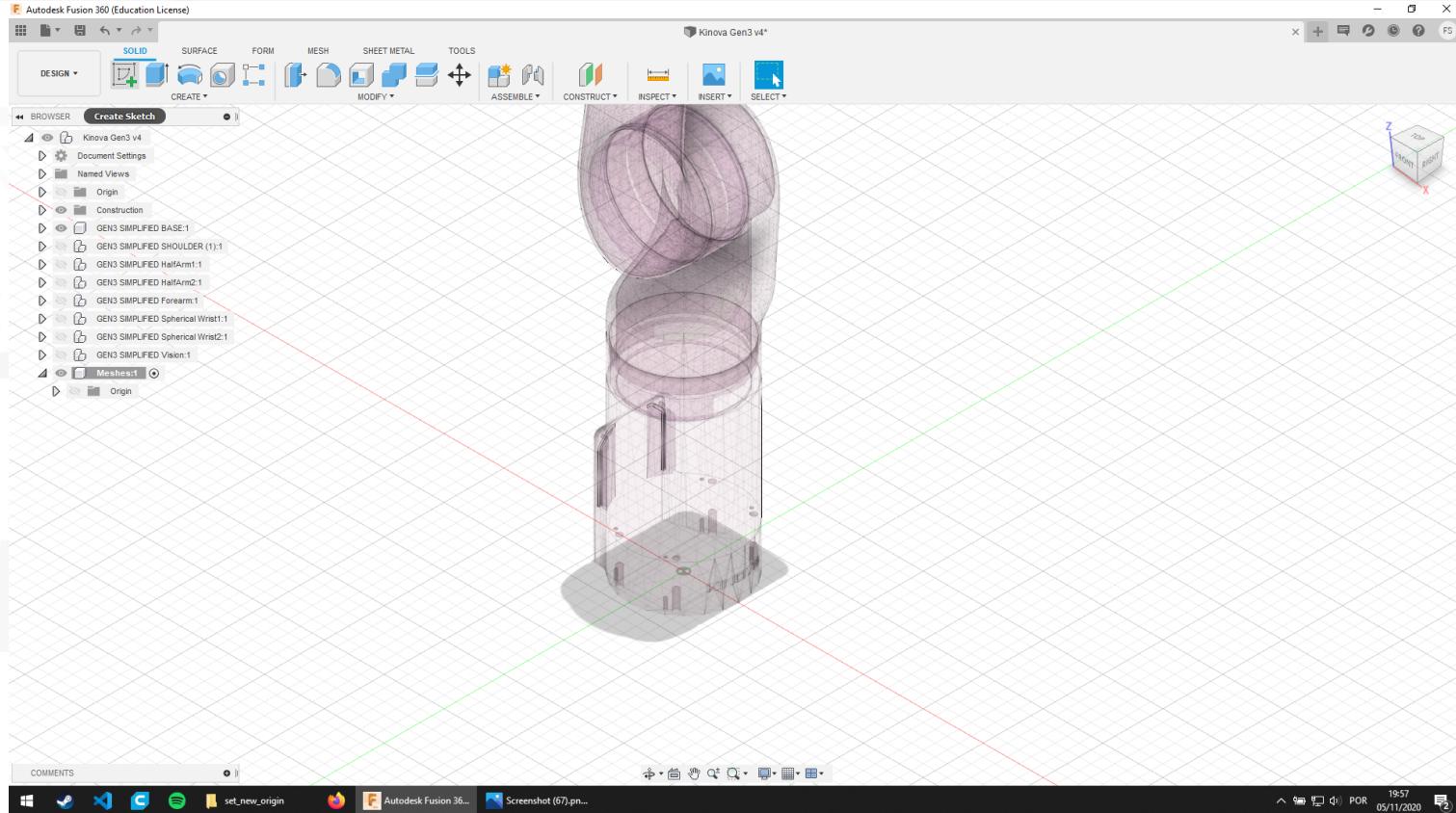
Fusion 360 - Meshes



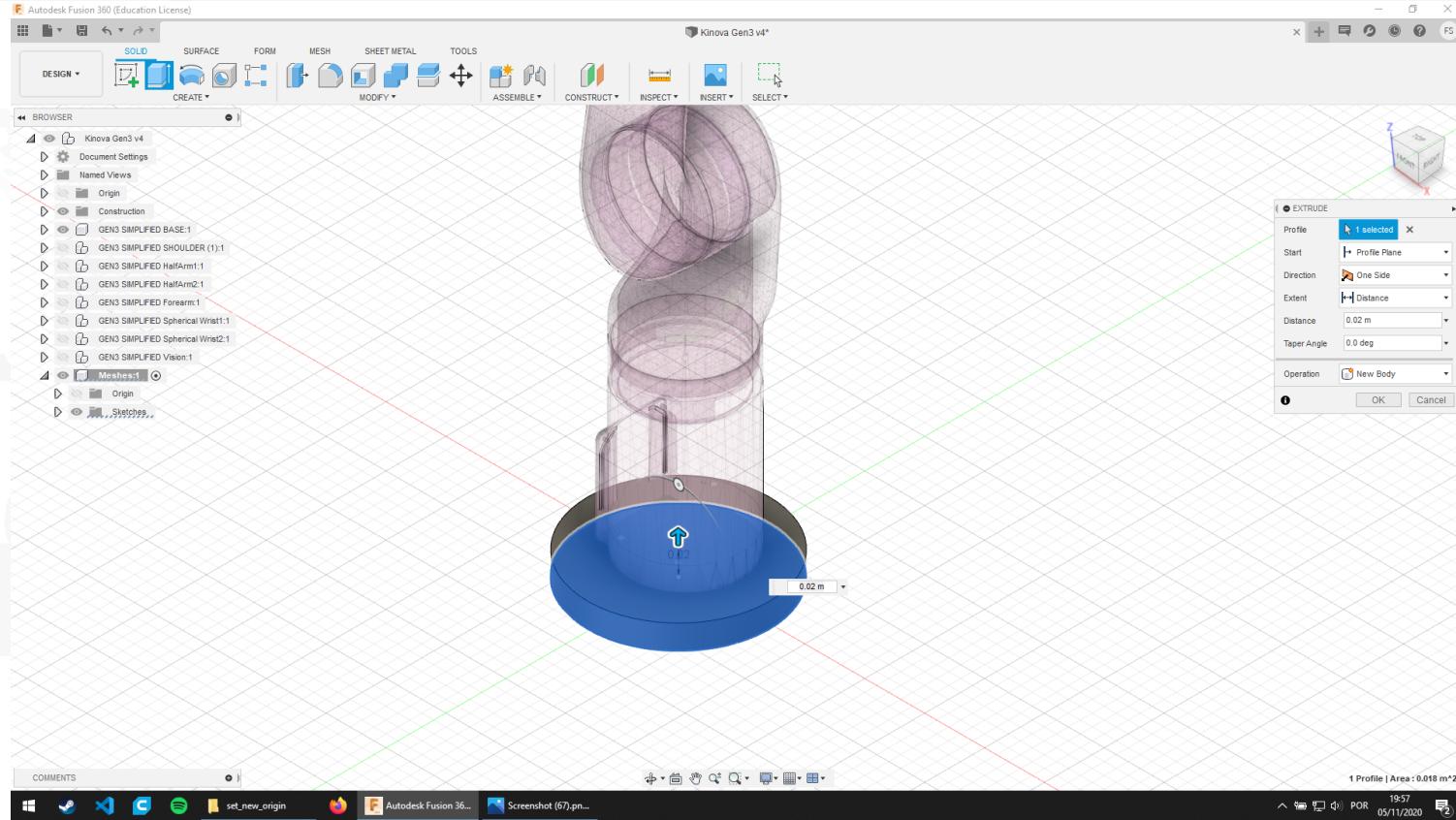
Fusion 360 - Meshes



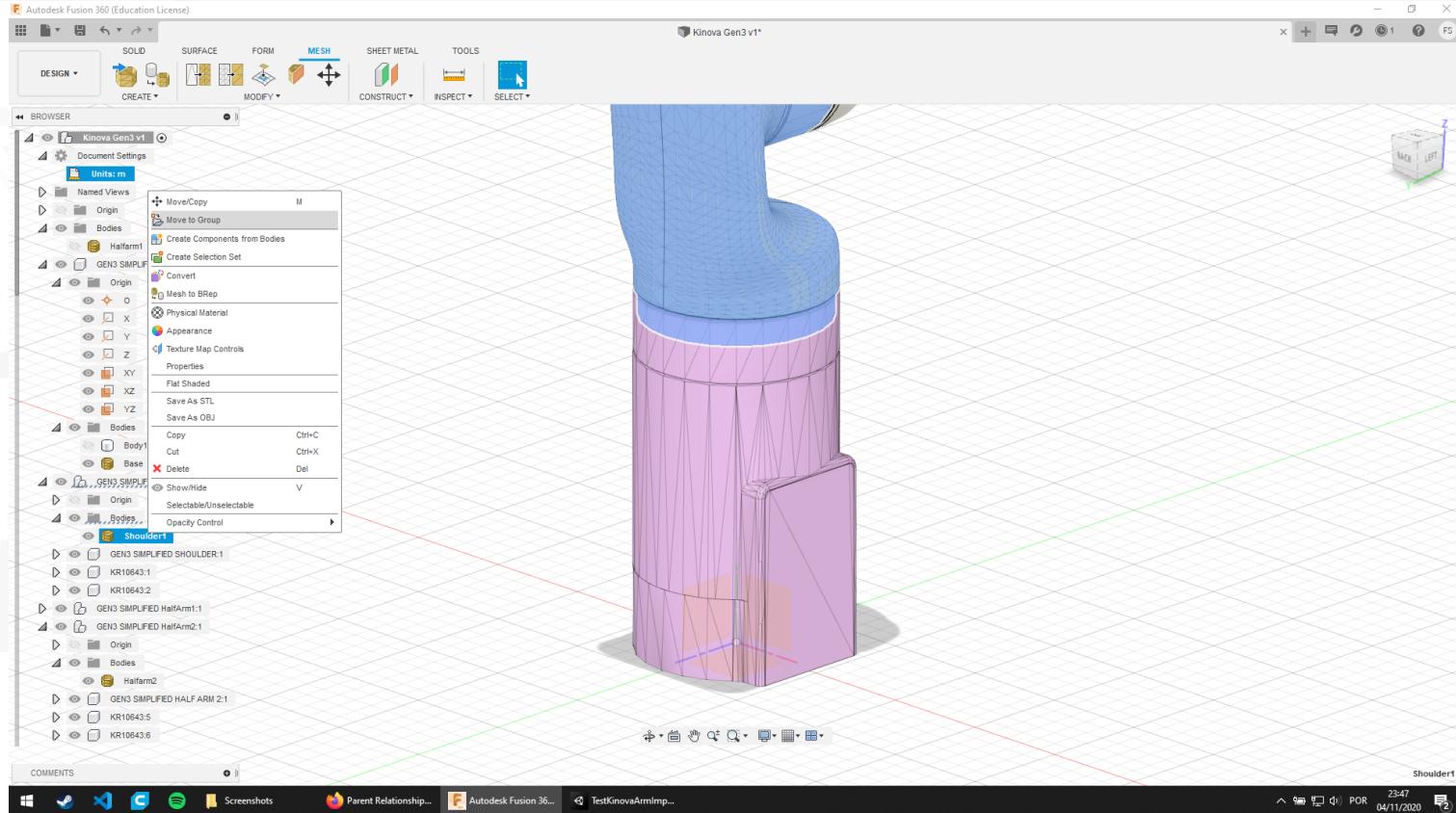
Fusion 360 - Meshes



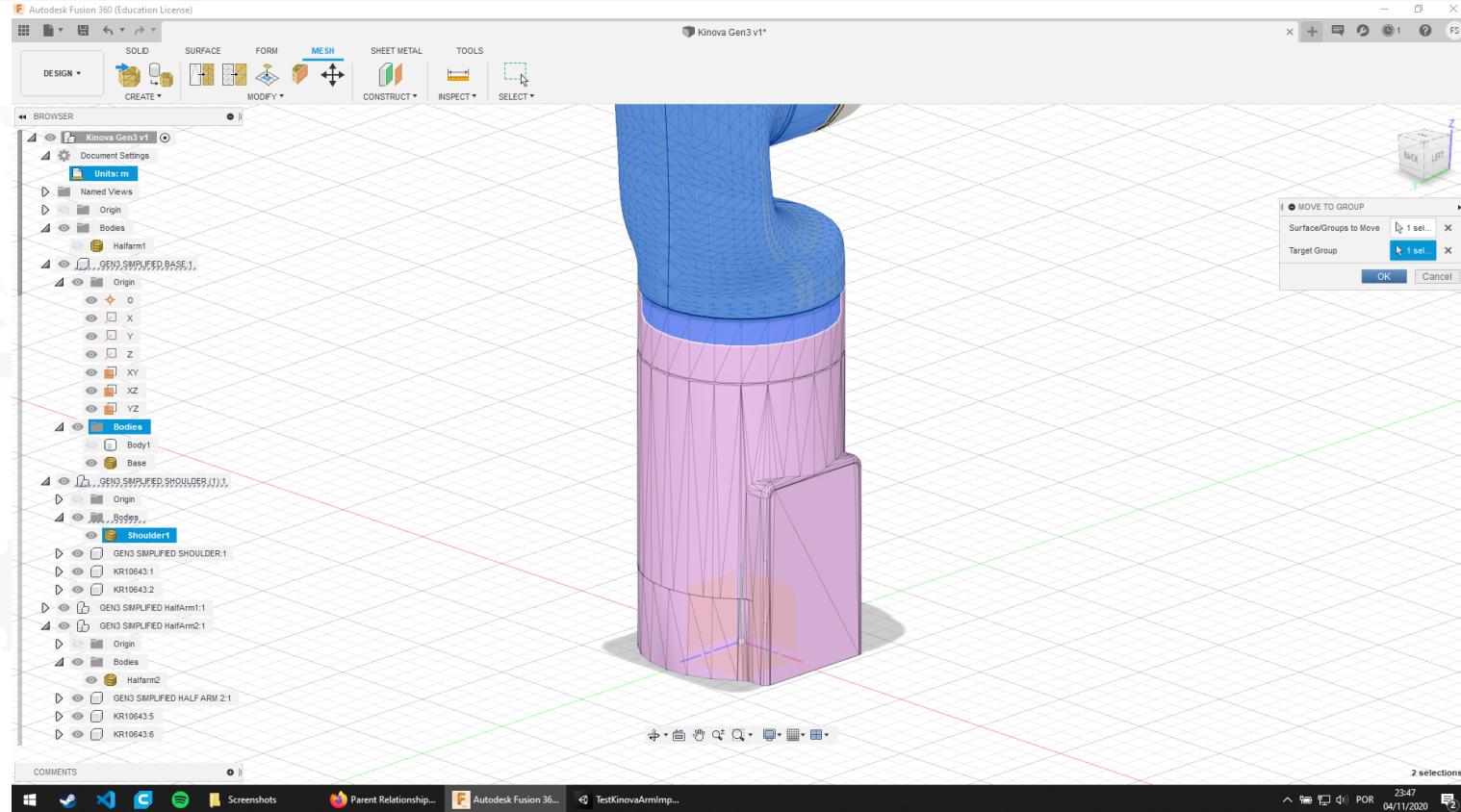
Fusion 360 - Meshes



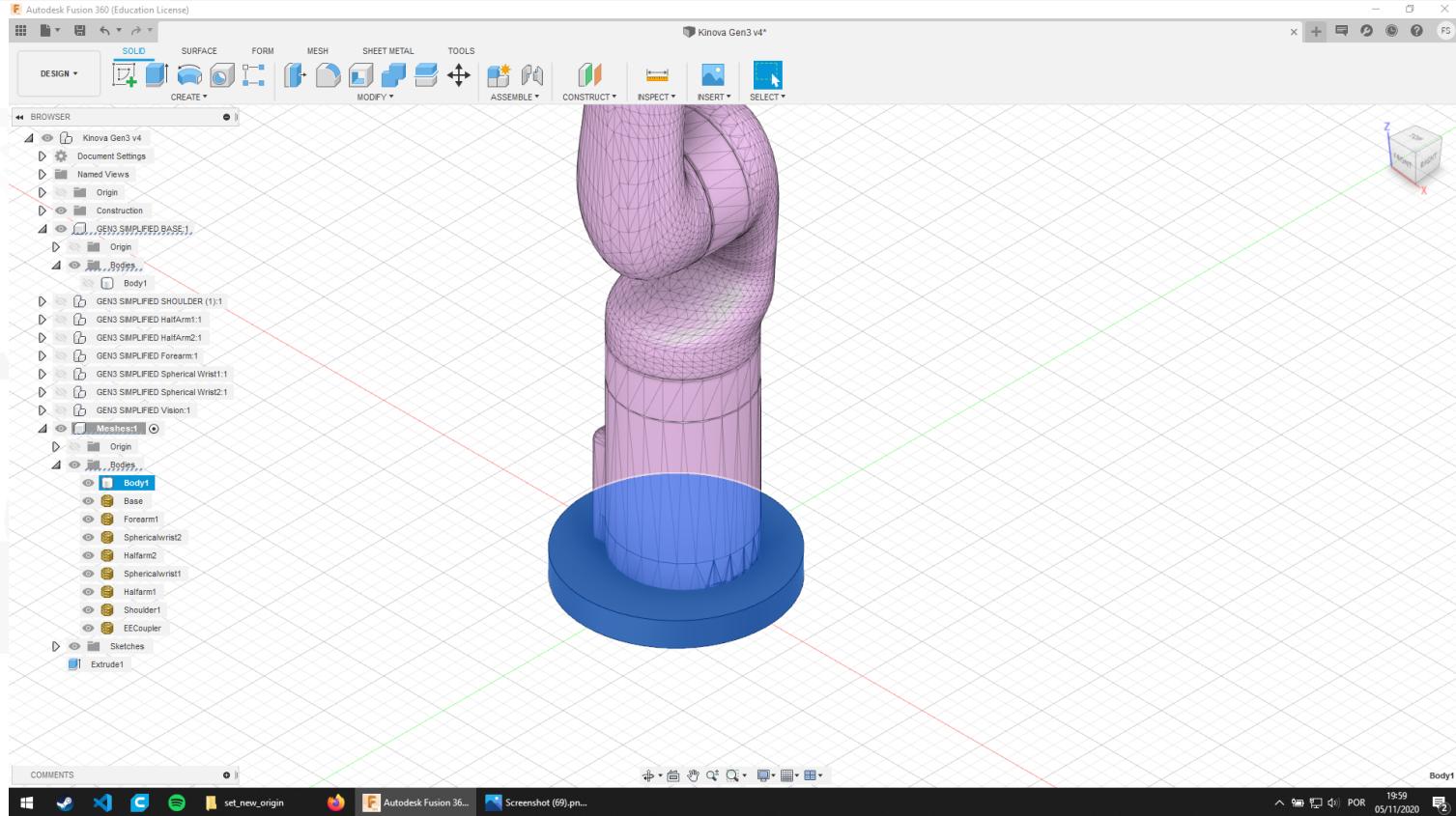
Fusion 360 - Meshes



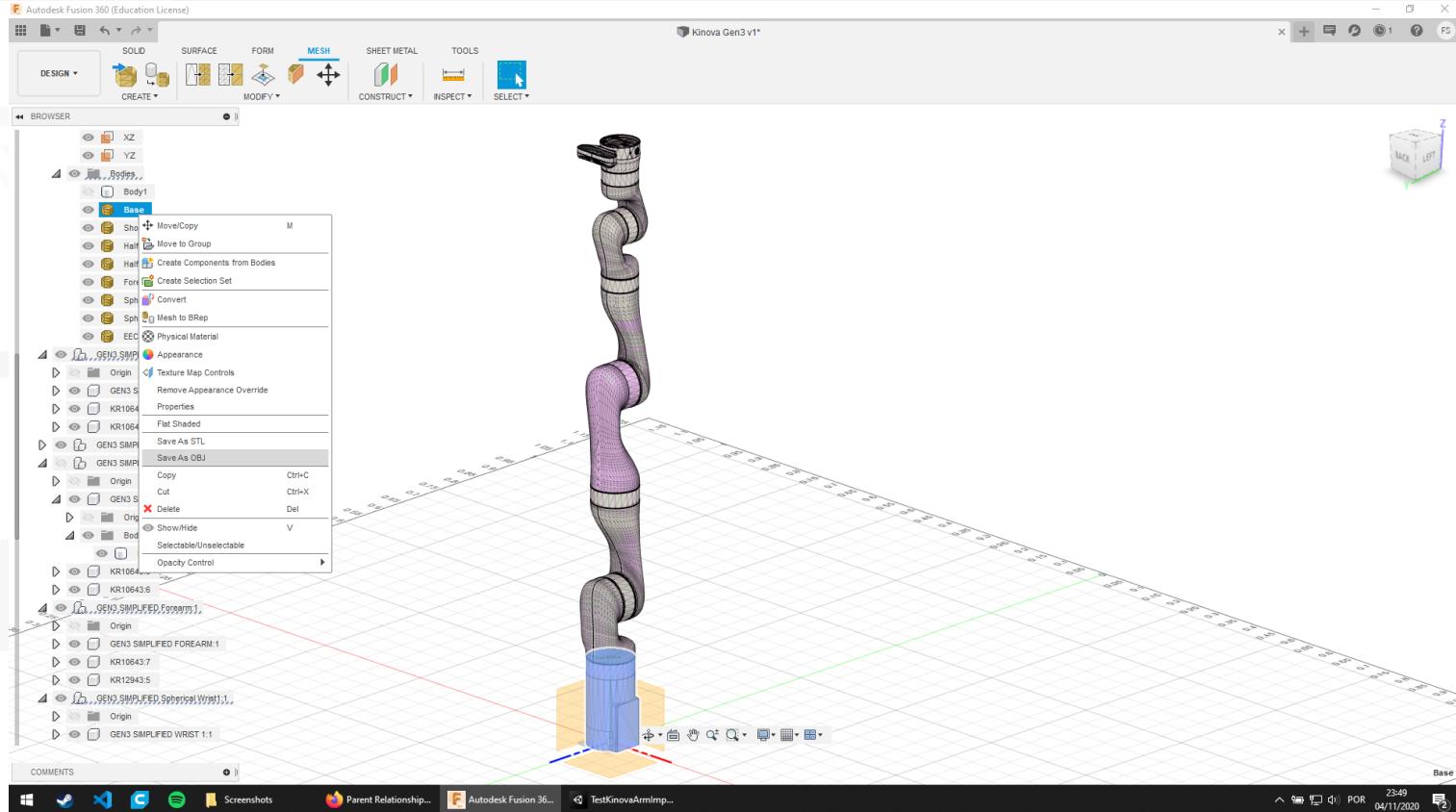
Fusion 360 - Meshes



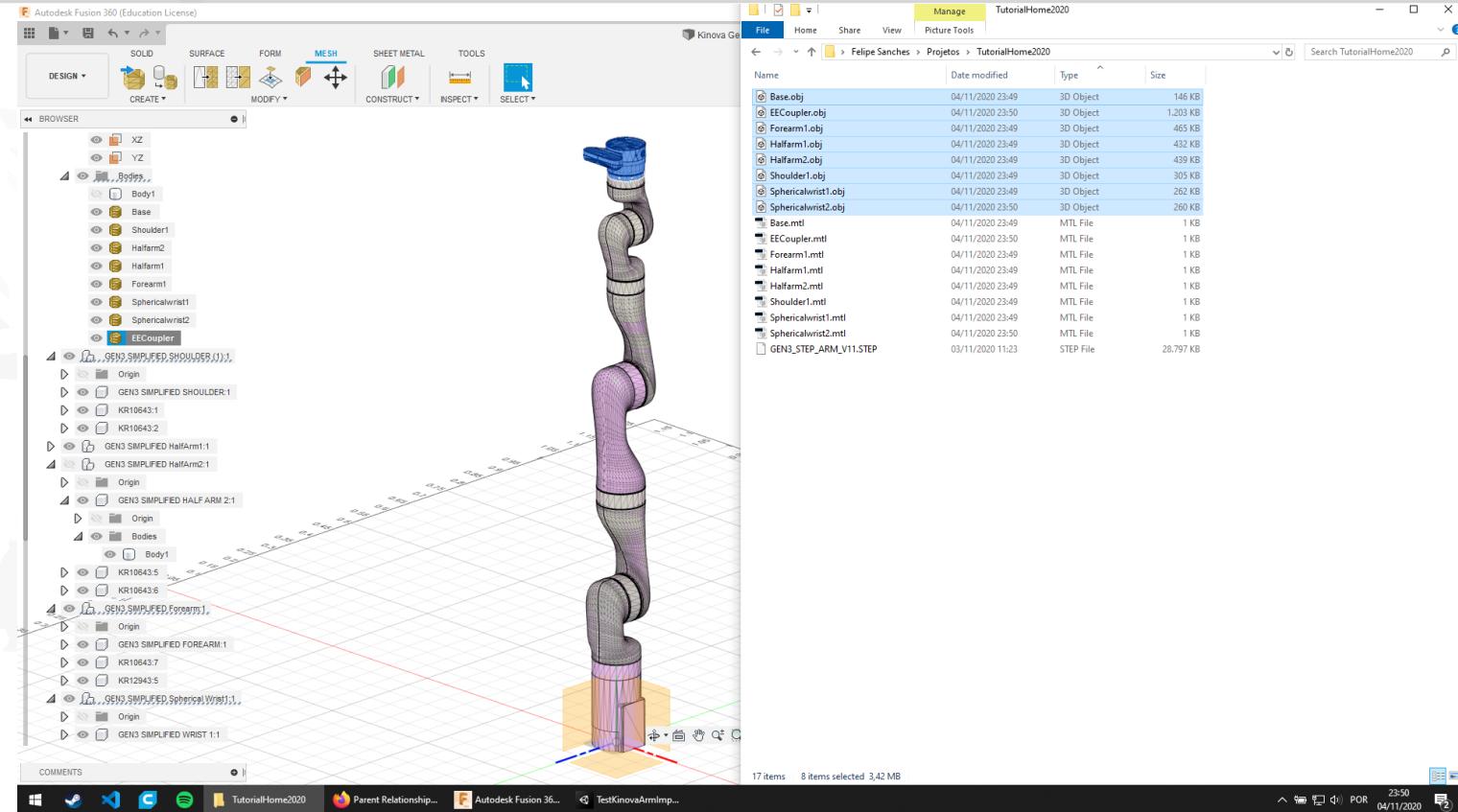
Fusion 360 - Meshes



Fusion 360 - Meshes



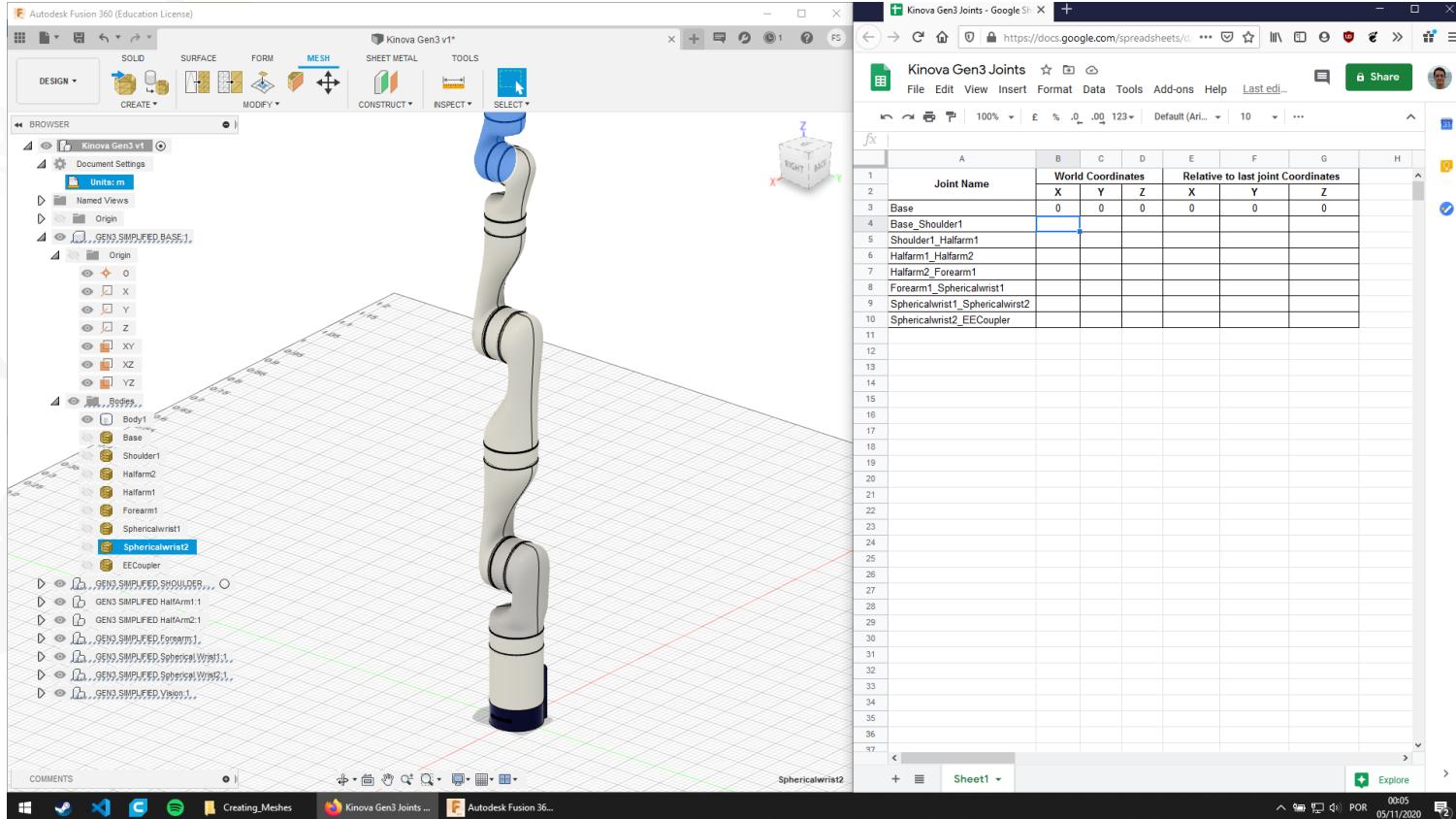
Fusion 360 - Meshes



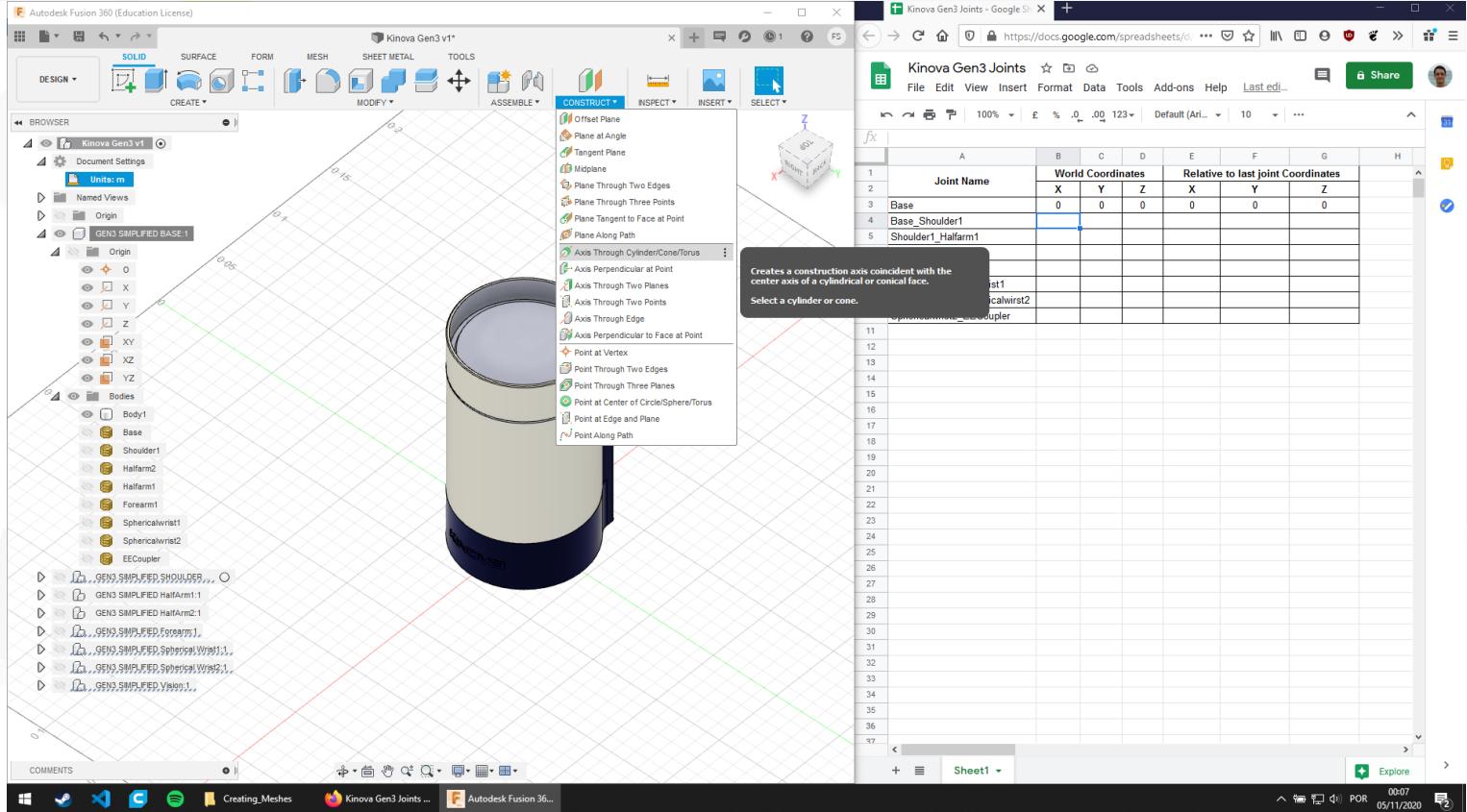
Extraindo info. das Juntas



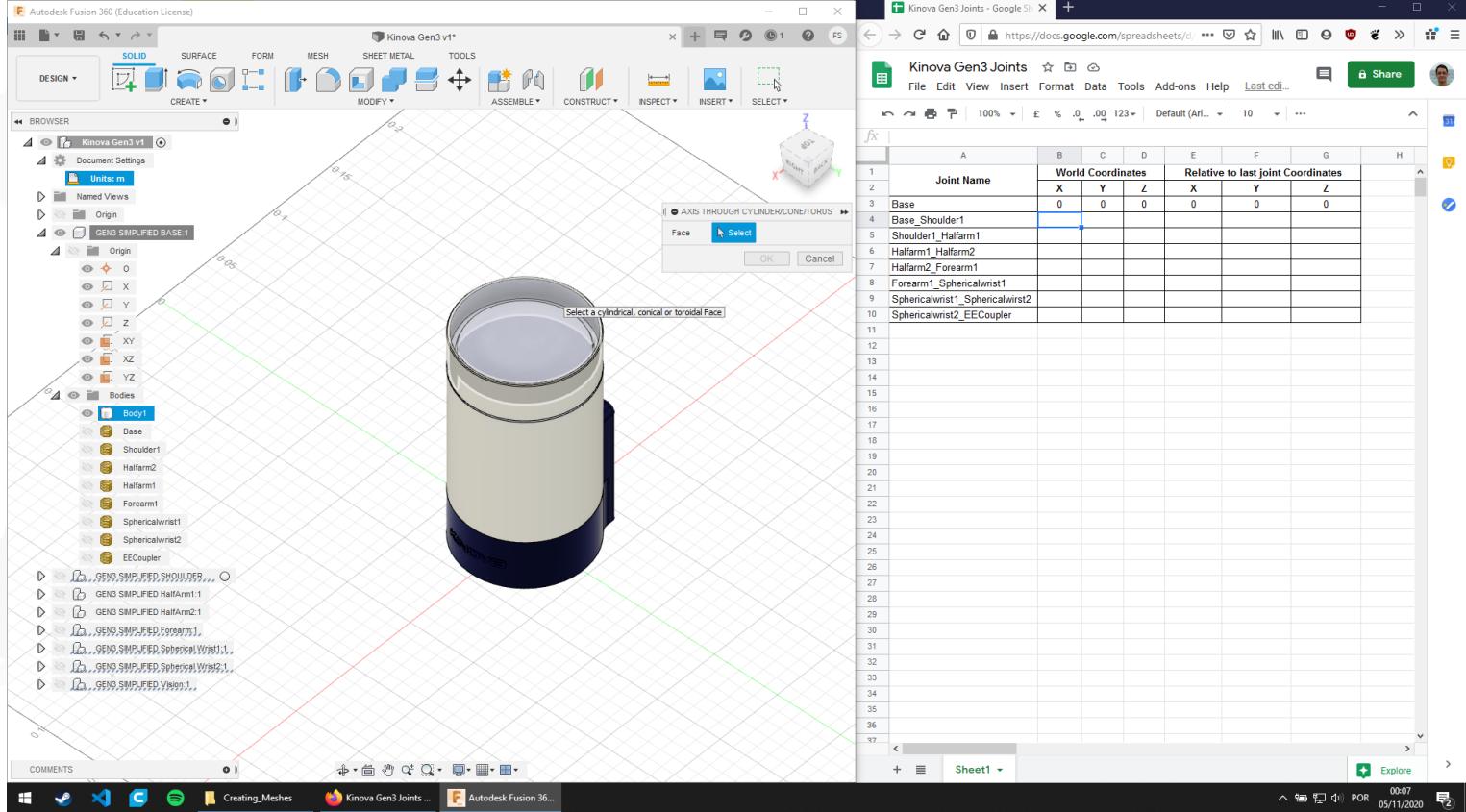
Fusion 360 - Posição das juntas



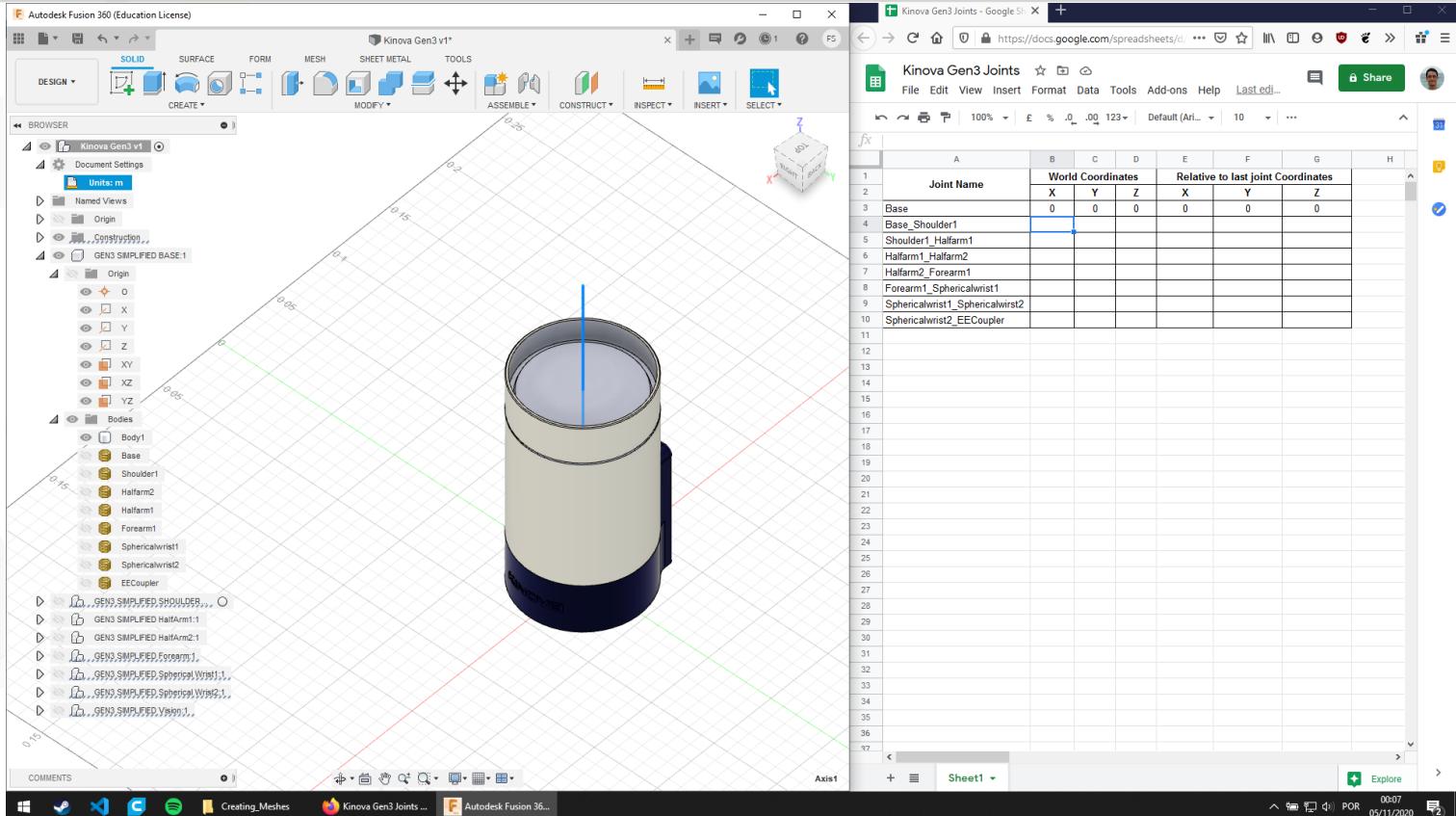
Fusion 360 - Posição das juntas



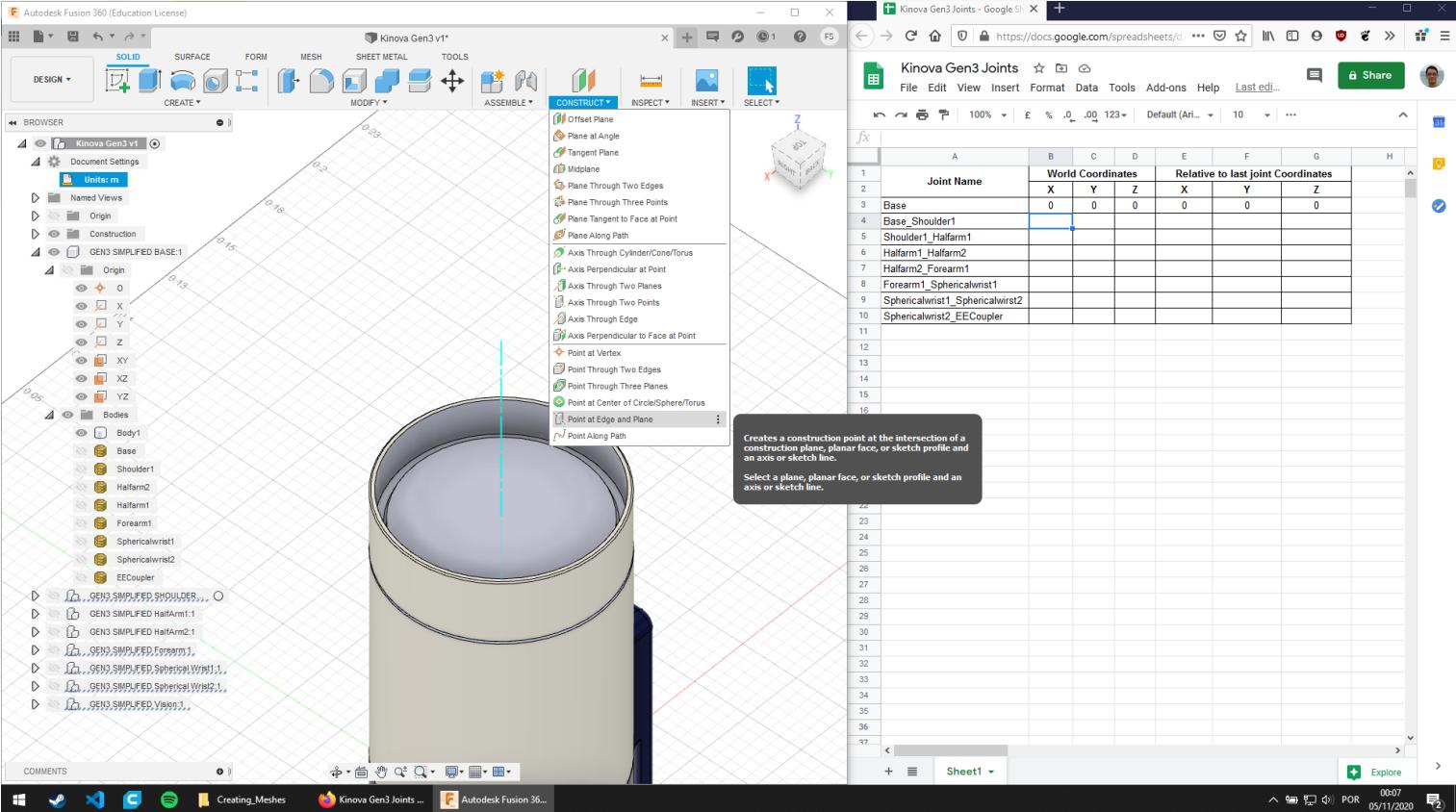
Fusion 360 - Posição das juntas



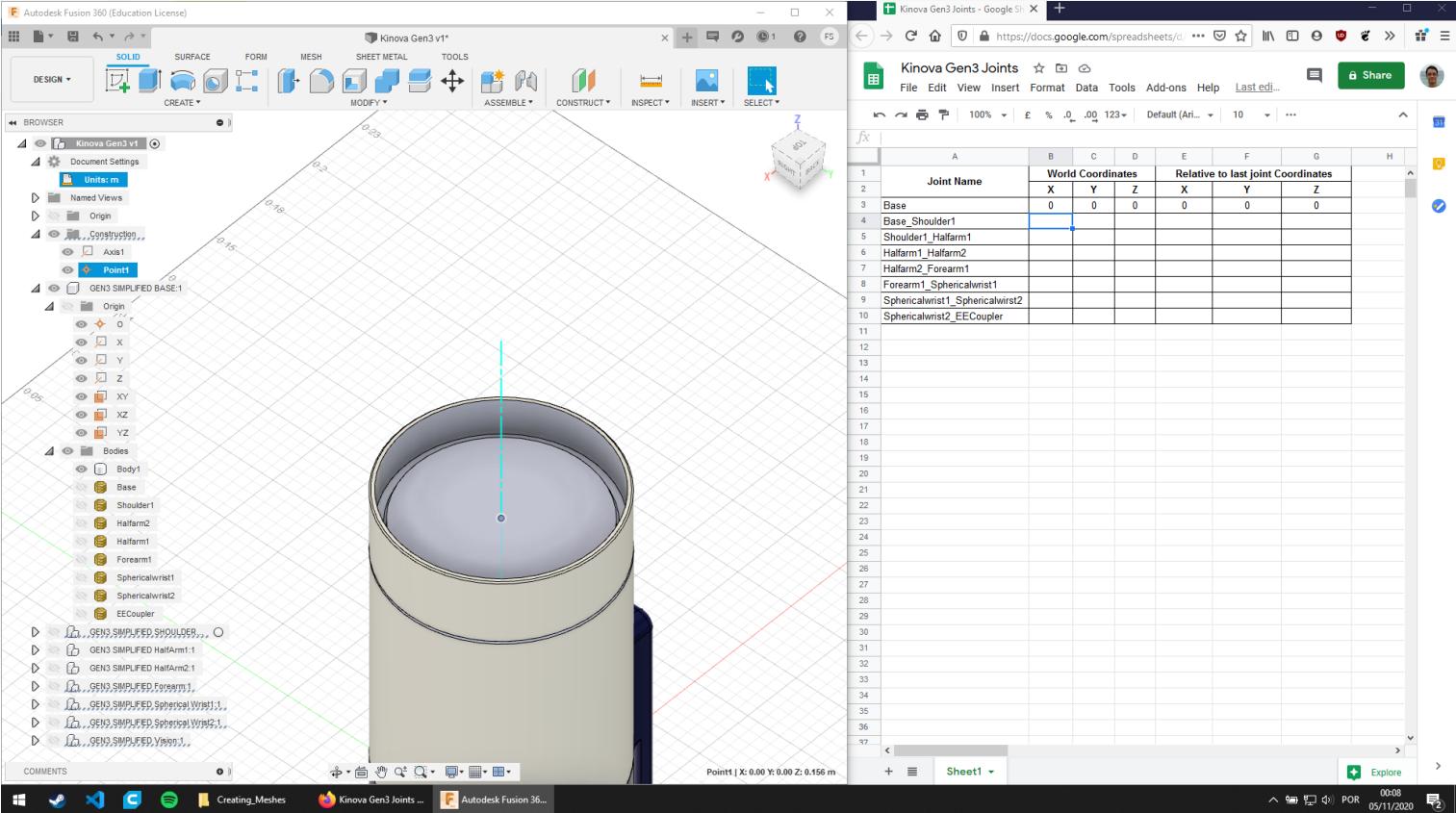
Fusion 360 - Posição das juntas



Fusion 360 - Posição das juntas



Fusion 360 - Posição das juntas



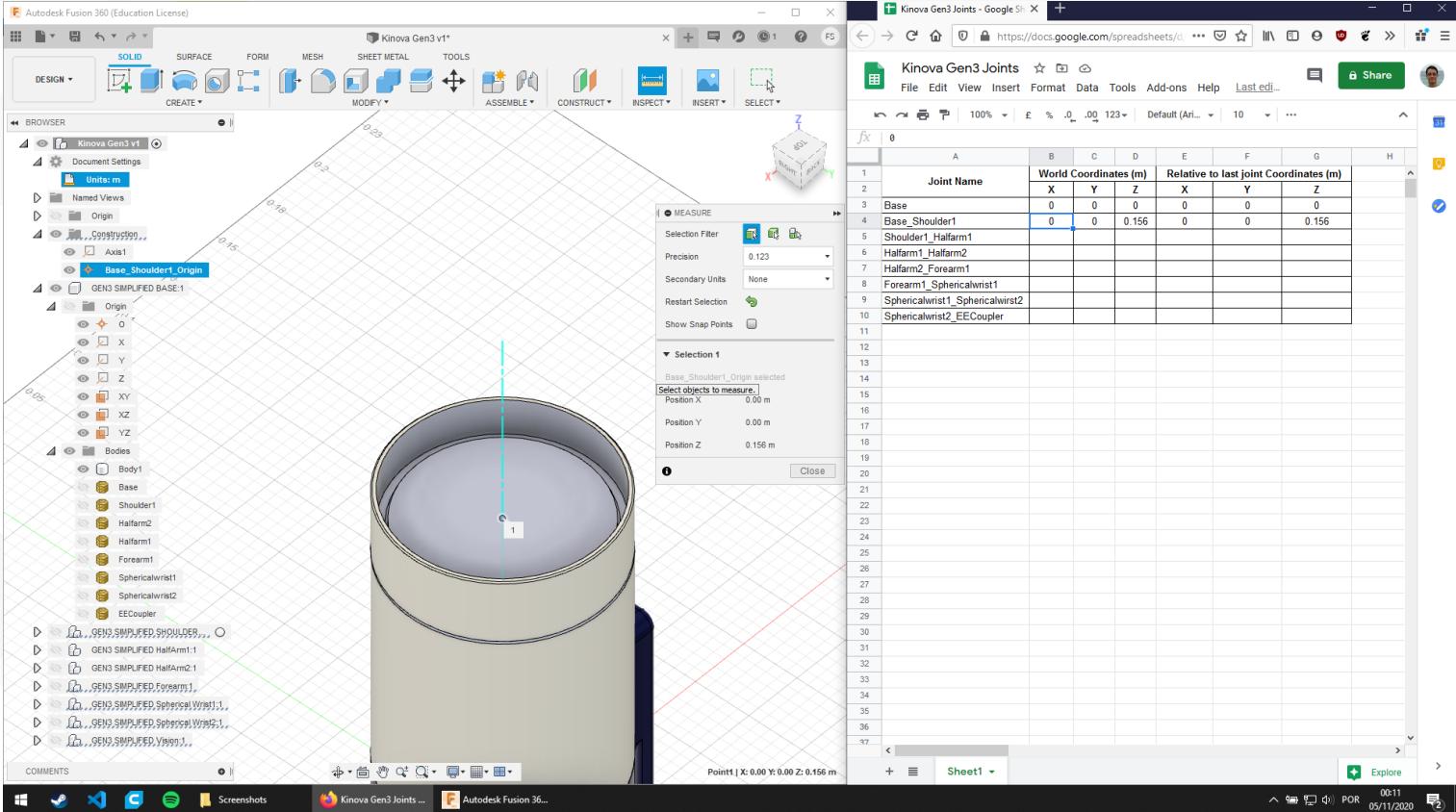
Fusion 360 - Posição das juntas

The image shows two software interfaces side-by-side. On the left is Autodesk Fusion 360, displaying a 3D model of a robotic arm (Kinova Gen3) in its simplified base configuration. The model consists of a cylindrical base and several articulated segments. The Fusion 360 interface includes a toolbar at the top, a browser panel on the left listing components like 'Base_Shoulder1_Origin' and 'Body1', and a 3D workspace in the center. On the right is a Google Sheets document titled 'Kinova Gen3 Joints'. This spreadsheet contains a table with two columns of data: 'World Coordinates' and 'Relative to last joint Coordinates'. The table lists ten joints, each with a unique name and its corresponding coordinates in both coordinate systems. The 'World Coordinates' column has the first three rows filled with data for 'Base', 'Base_Shoulder1', and 'Shoulder1_Halfarm1'. The 'Relative to last joint Coordinates' column has the first three rows filled with data for 'Base', 'Base_Shoulder1', and 'Shoulder1_Halfarm1'. The rest of the table is empty.

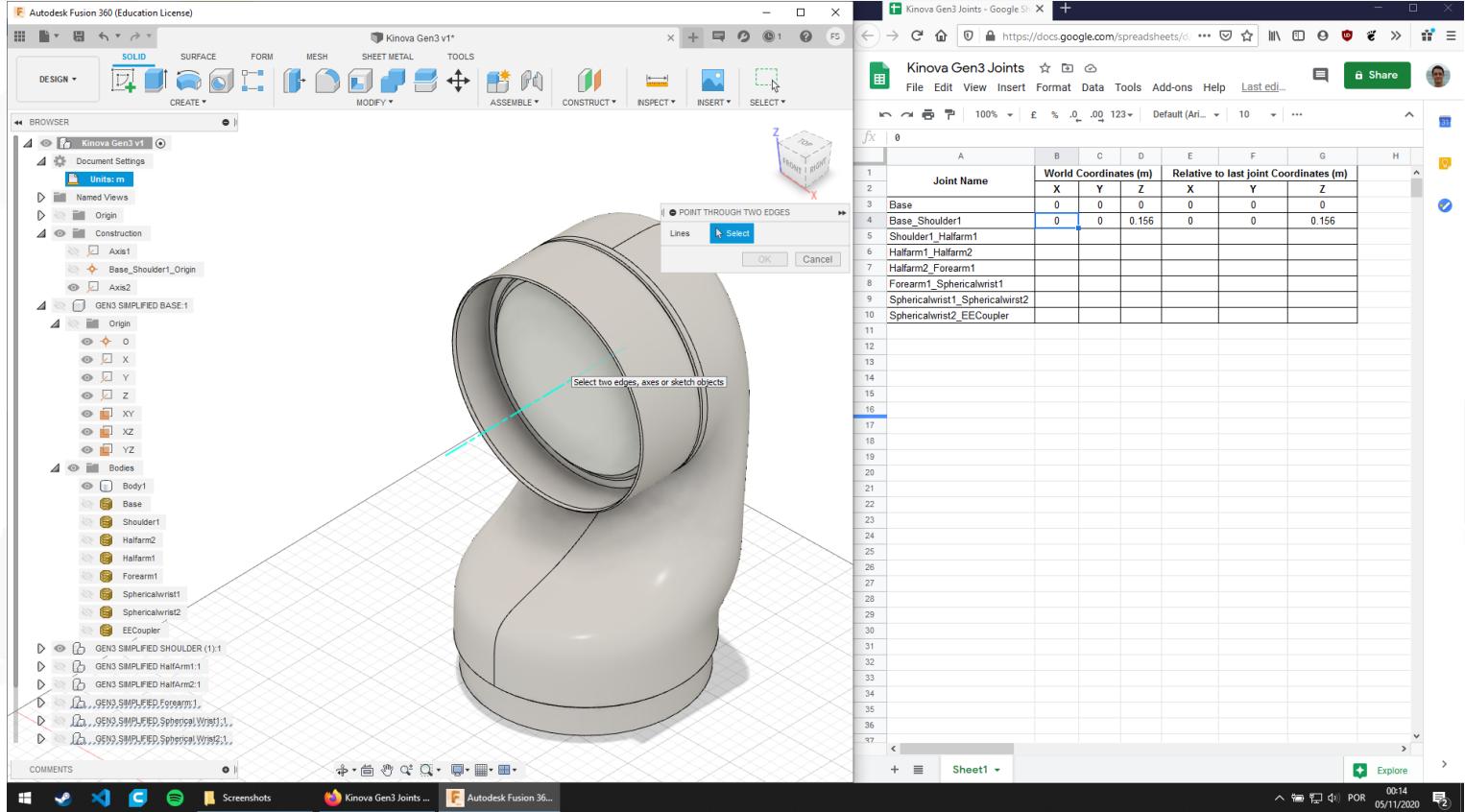
A	B	C	D	E	F	G
Joint Name	World Coordinates	Relative to last joint Coordinates				
	X	Y	Z	X	Y	Z
Base	0	0	0	0	0	0
Base_Shoulder1						
Shoulder1_Halfarm1						
Halfarm1_Halfarm2						
Halfarm2_Forearm1						
Forearm1_Sphericalwrist1						
Sphericalwrist1_Sphericalwrist2						
Sphericalwrist2_EECoupler						



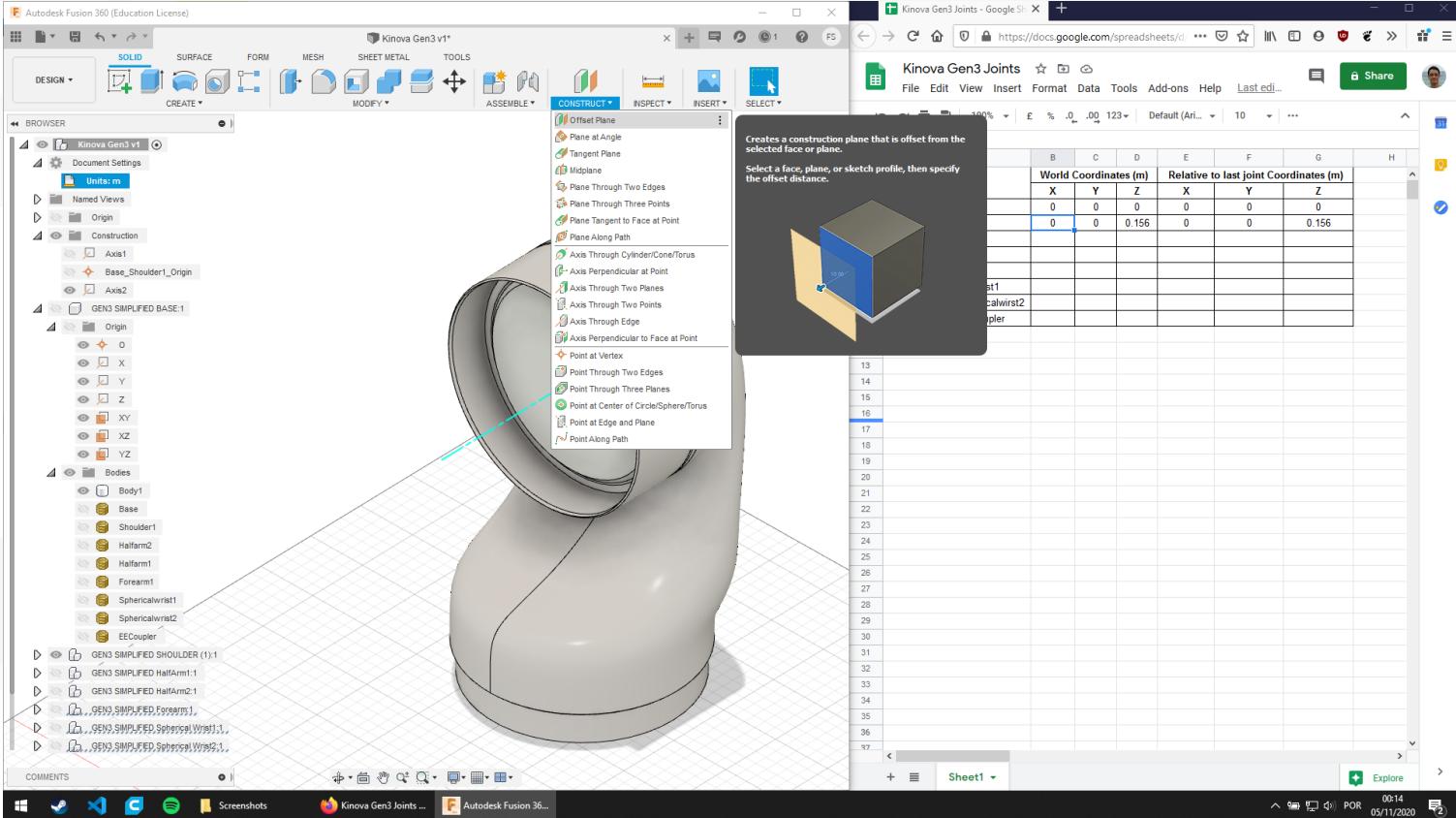
Fusion 360 - Posição das juntas



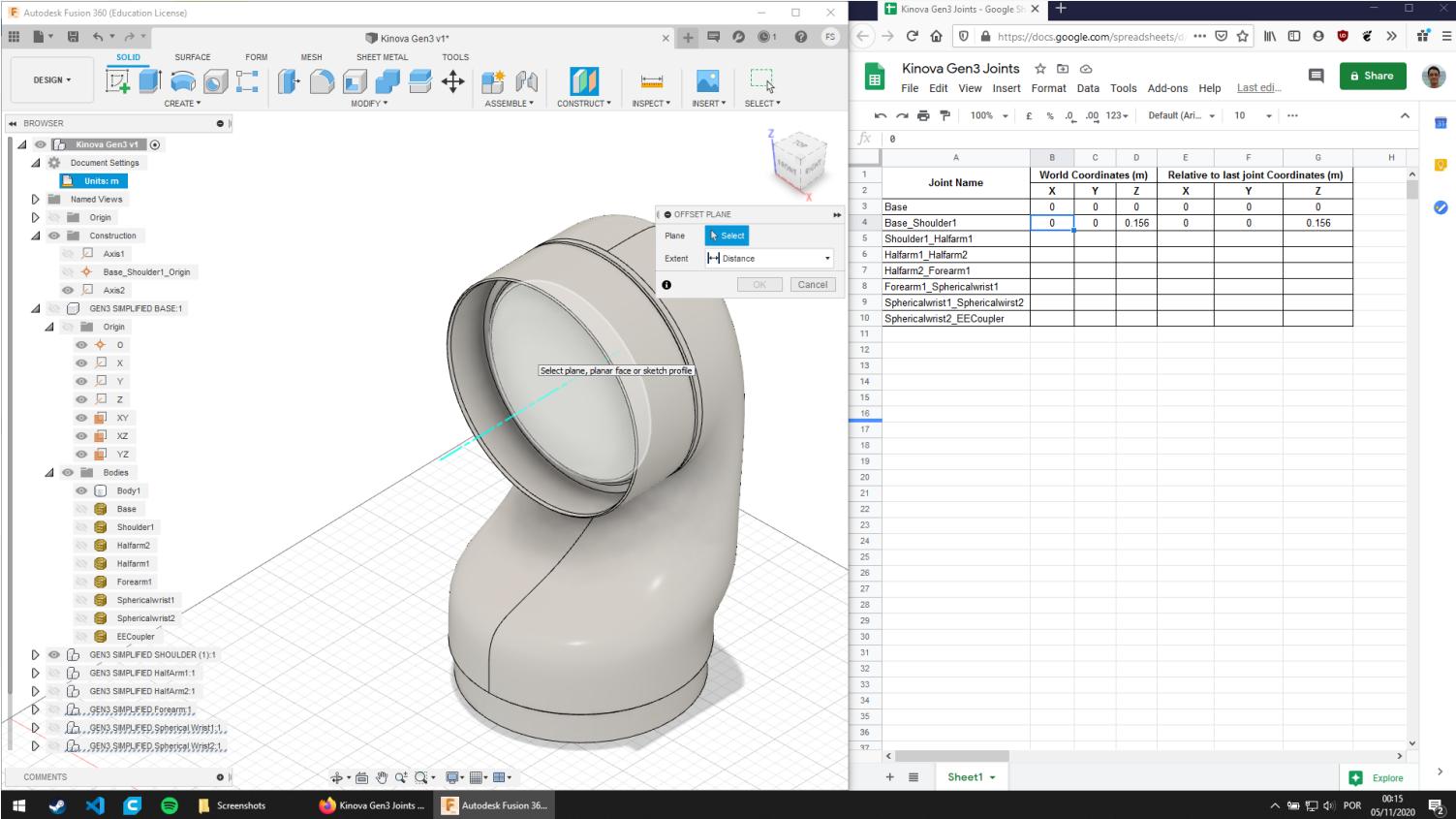
Fusion 360 - Posição das juntas



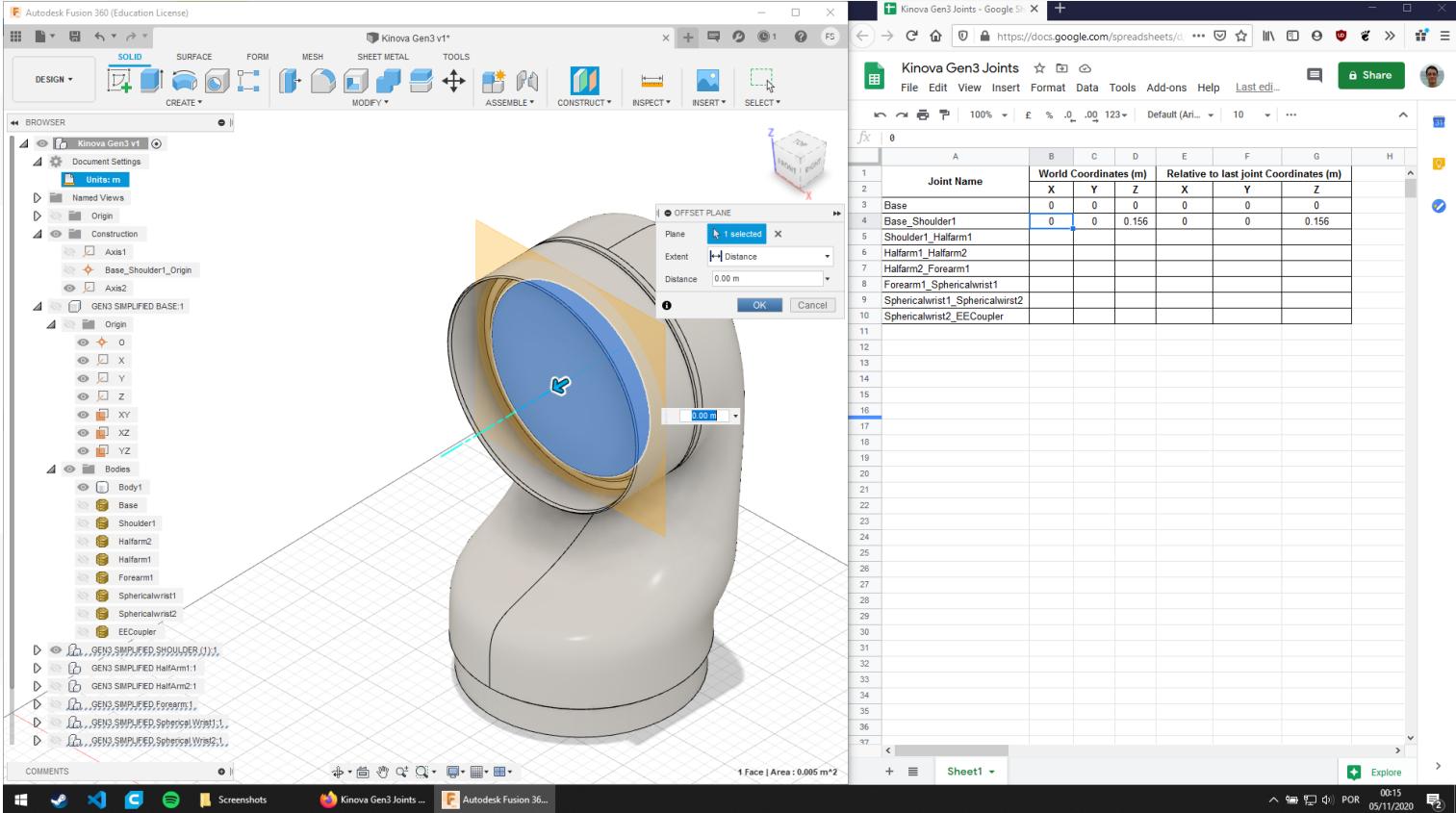
Fusion 360 - Posição das juntas



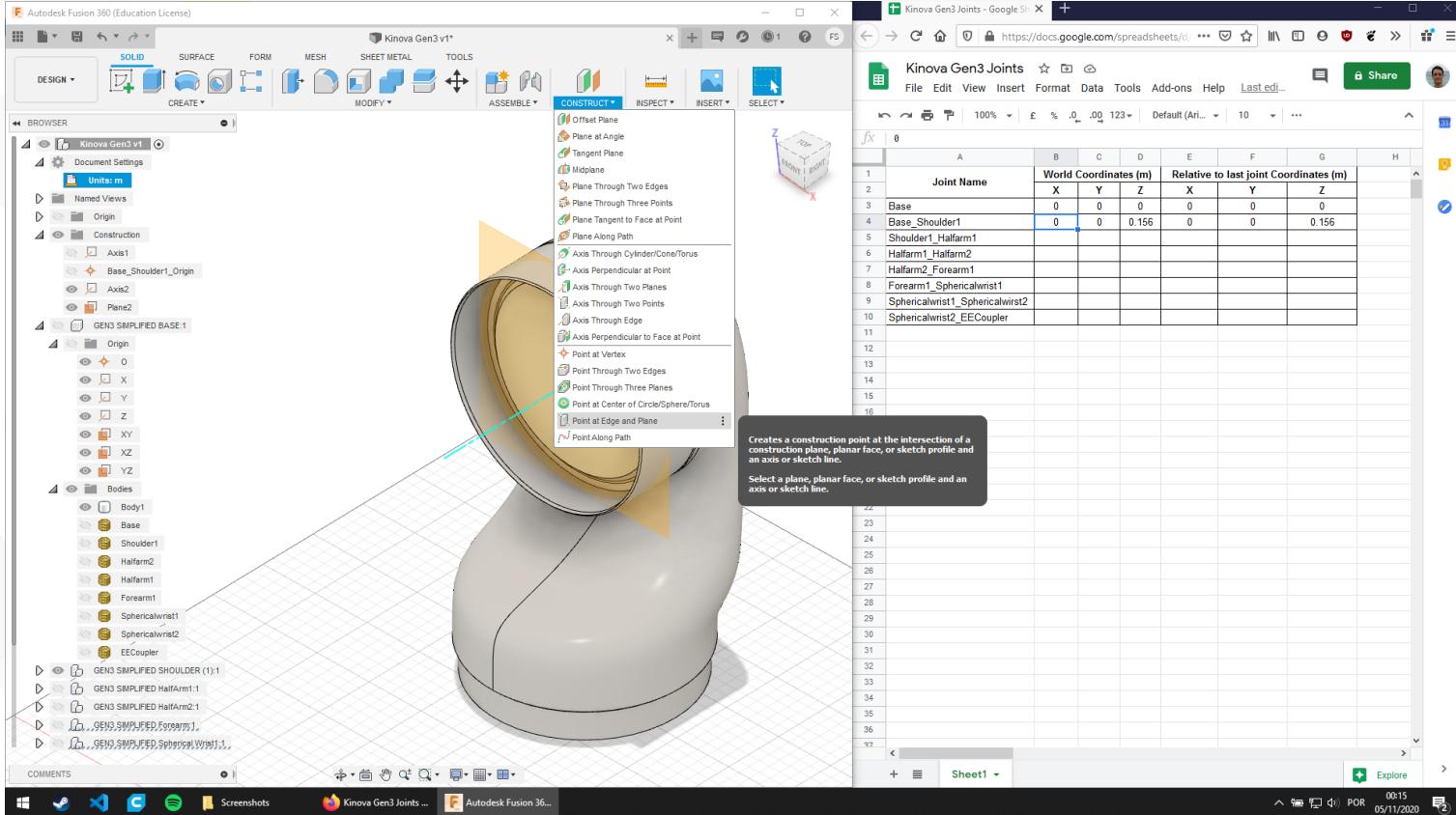
Fusion 360 - Posição das juntas



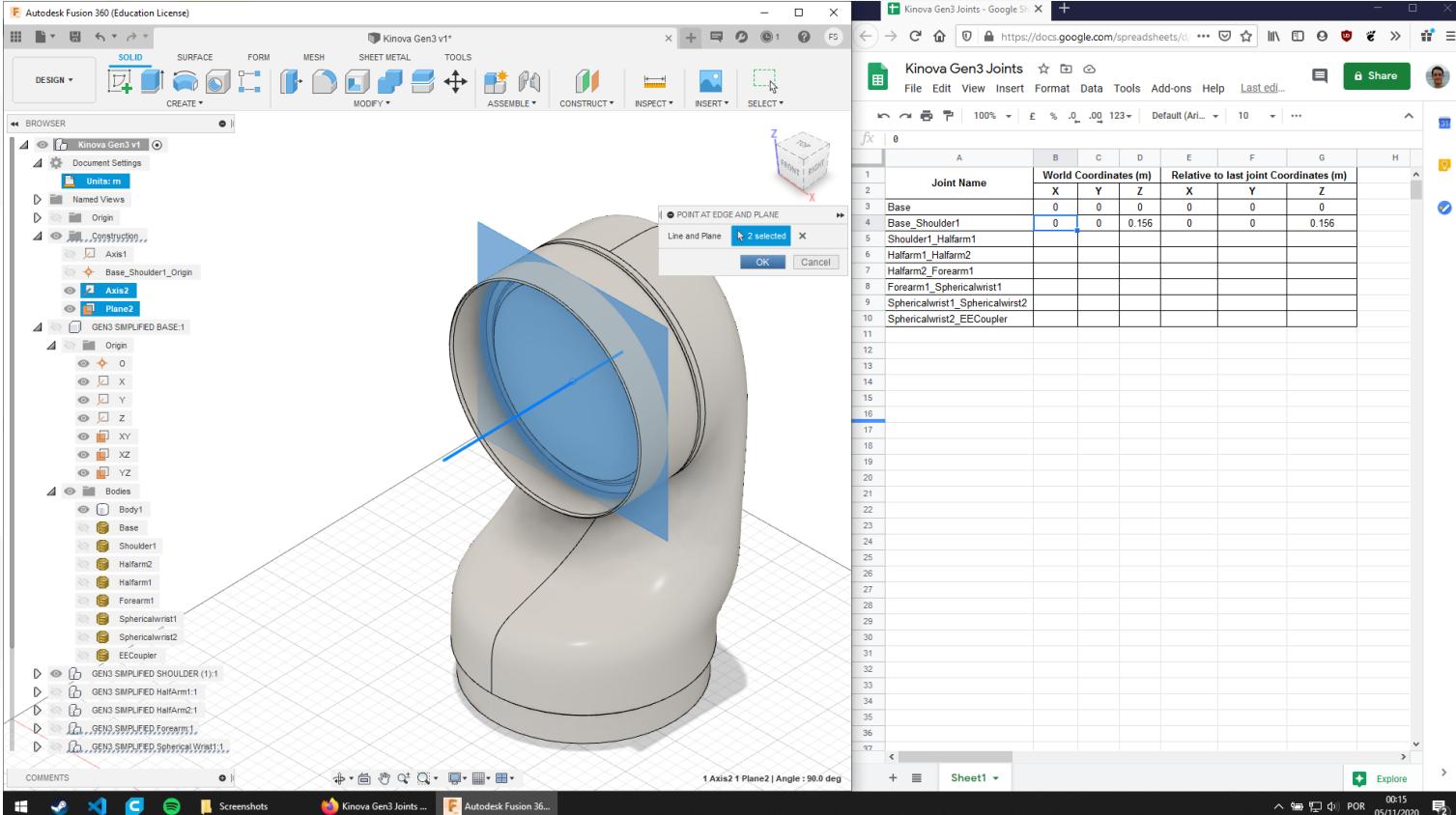
Fusion 360 - Posição das juntas



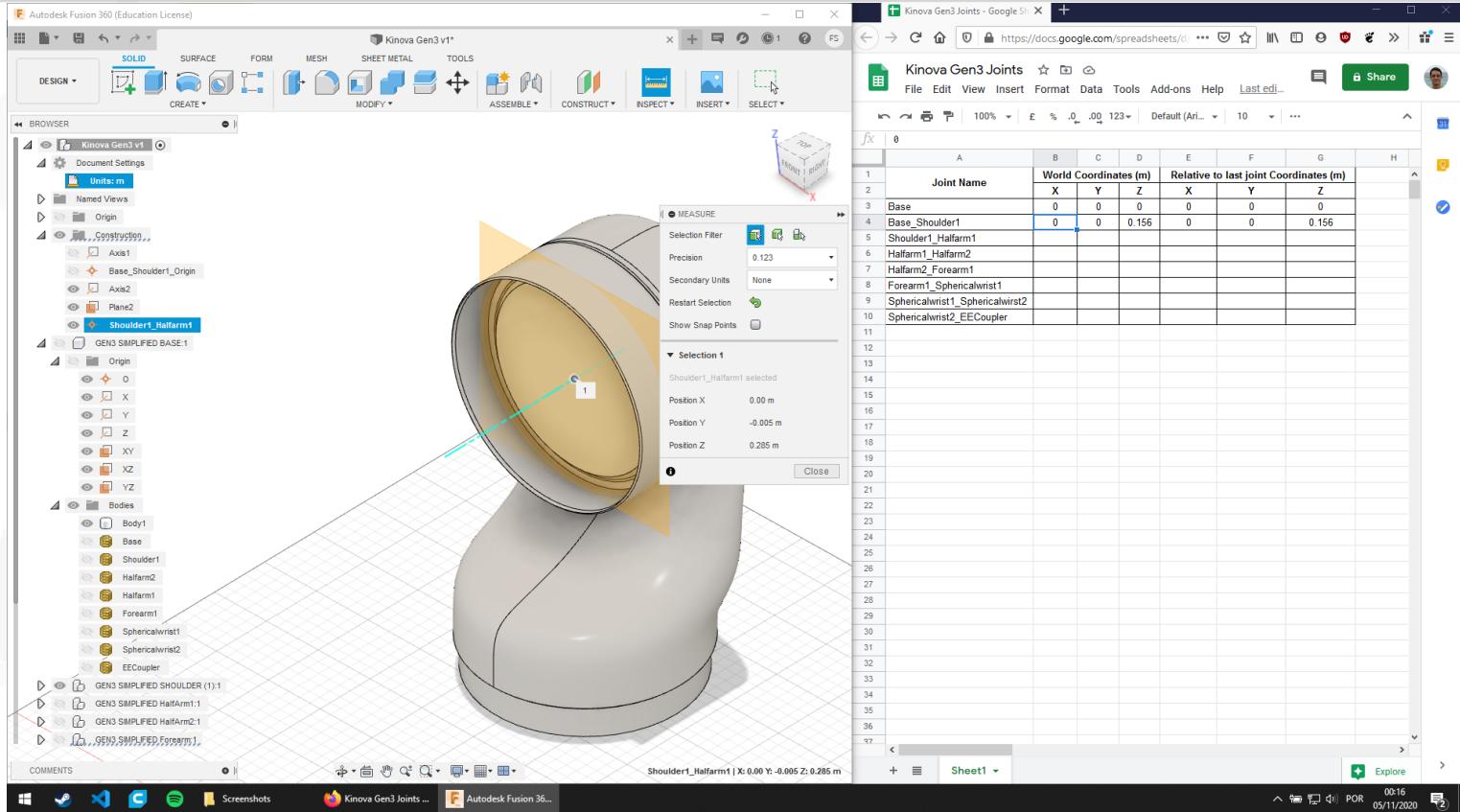
Fusion 360 - Posição das juntas



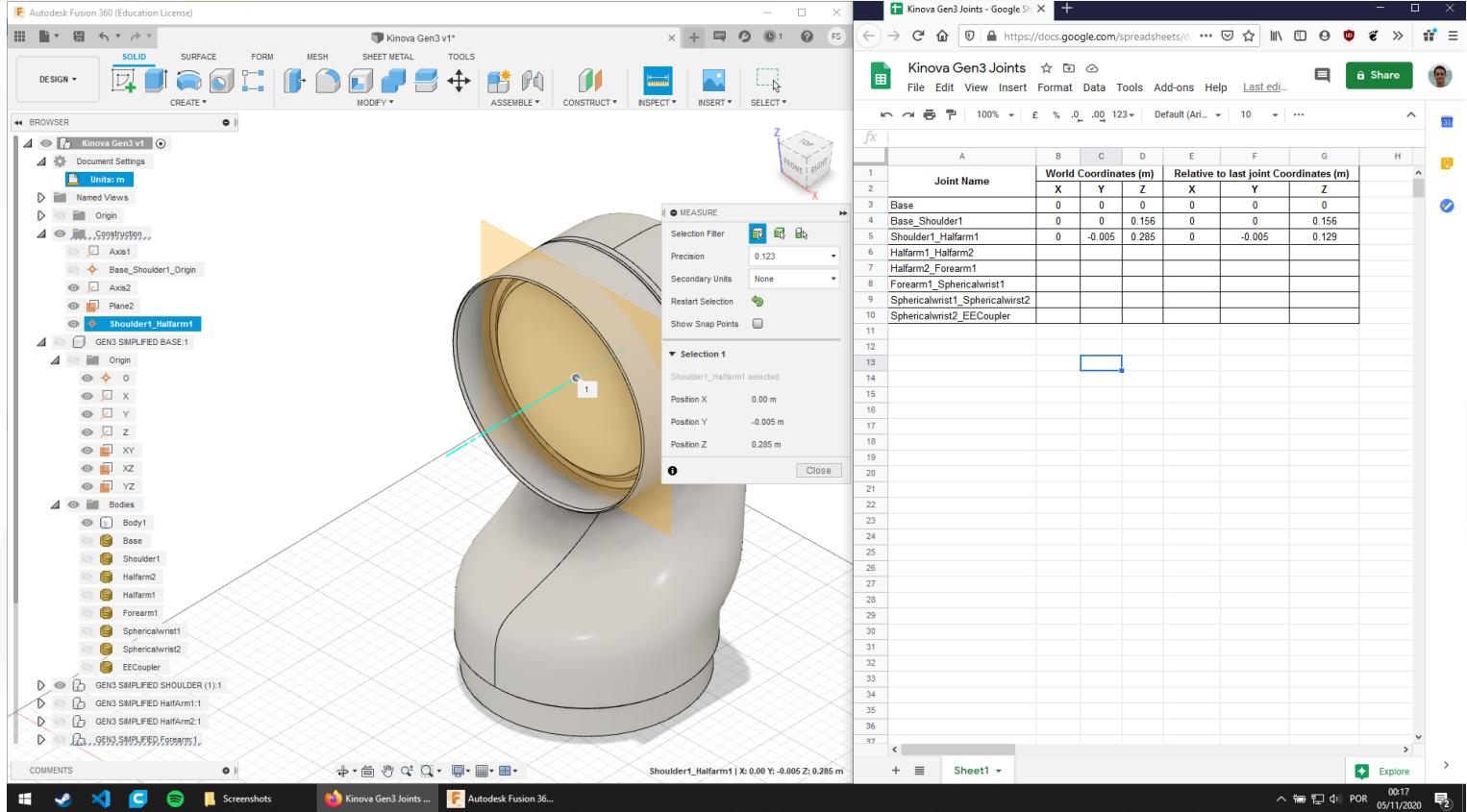
Fusion 360 - Posição das juntas



Fusion 360 - Posição das juntas



Fusion 360 - Posição das juntas



Fusion 360 - Posição das juntas

The image shows a dual-screen setup. On the left screen, the Autodesk Fusion 360 interface is displayed. It features a top toolbar with tabs like DESIGN, SOLID, SURFACE, FORM, MESH, SHEET METAL, and TOOLS. Below the toolbar is a ribbon menu with categories such as CREATE, MODIFY, ASSEMBLE, CONSTRUCT, INSPECT, INSERT, and SELECT. The main workspace shows a 3D model of a robotic arm, specifically a Kinova Gen3, in a light beige color. A coordinate system (X, Y, Z) is overlaid on the arm. To the left of the workspace is the 'BROWSER' panel, which lists various components and joints of the assembly. On the right side of the Fusion 360 window, there is a 'MEASURE' panel with settings for Selection Filter (set to 'None'), Precision (set to 0.123), Secondary Units (set to 'None'), and a 'Restart Selection' button. Below this is a 'Selection 1' section showing that 'Sphericalwrist2_EECoupler' has been selected. It displays the Position X (0.00 m), Position Y (-0.025 m), and Position Z (1.126 m). At the bottom of the Fusion 360 window, there is a toolbar with icons for file operations, search, and other tools. The status bar at the bottom right shows the part name 'Sphericalwrist2_EECoupler | X: 0.00 Y: -0.025 Z: 1.126 m'. On the right screen, a Google Sheets spreadsheet titled 'Kinova Gen3 Joints' is open. The spreadsheet has two columns: 'Joint Name' and 'World Coordinates (m)'. The 'World Coordinates (m)' column is further divided into 'X', 'Y', and 'Z' sub-columns. The data for the first ten joints is as follows:

Joint Name	X	Y	Z	X	Y	Z
Base	0	0	0	0	0	0
Base_Shoulder1	0	0	0.156	0	0	0.156
Shoulder1_Halfarm1	0	-0.005	0.285	0	-0.005	0.129
Halfarm1_Halfarm2	0	-0.012	0.495	0	-0.007	0.21
Halfarm2_Forearm1	0	-0.018	0.706	0	-0.006	0.211
Forearm1_Sphericalwrist1	0	-0.025	0.914	0	-0.007	0.208
Sphericalwrist1_Sphericalwrist2	0	-0.025	1.02	0	0	0.106
Sphericalwrist2_EECoupler	0	-0.025	1.126	0	0	0.106



MoveIt



MoveIt

- Nessa etapa iremos:
 - Criar o arquivo de descrição Xacro/URDF
 - Criar o pacote ROS do manipulador através do Moveit setup assistant
 - Criar um controlador para comunicação "baixo-nível"
 - No nosso caso não usaremos um robô real mas sim uma simulação do Unity



Movel – Xacro/URDF

- Criação do arquivo de descrição do nosso manipulador
- Formato xacro nos fornece algumas facilidades quando comparado ao URDF
 - Para o nosso caso, algumas operações aritiméticas e constantes (π)



Movelt – Xacro/URDF

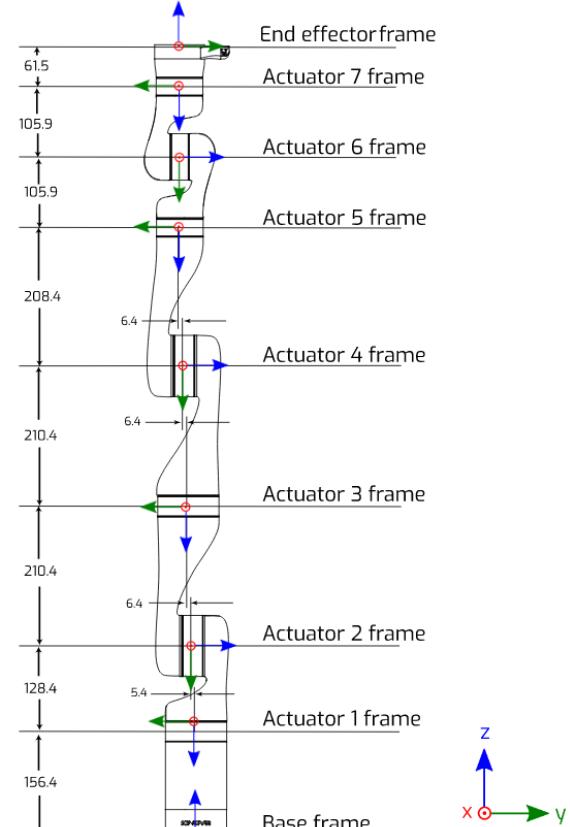


Table 10: Joint limits - 7 DoF spherical wrist

Actuator	Angular range	
	Lower limit	Upper limit
1	$-\infty$	$+\infty$
2	-126°	$+126^\circ$
3	$-\infty$	$+\infty$
4	-147°	$+147^\circ$
5	$-\infty$	$+\infty$
6	-117°	$+117^\circ$
7	$-\infty$	$+\infty$

Actuator Specifications:

- actuator speed (maximum, unloaded):
 - 25 RPM (small)
 - 17 RPM (large)
- actuator torque (small):
 - 13 N·m (nominal)
 - 34 N·m (peak)
- actuator torque (large):
 - 32 N·m (nominal)
 - 74 N·m (peak)



Moveit

- Nessa etapa iremos:
 - Criar o arquivo de descrição Xacro/URDF
 - Criar o pacote ROS do manipulador através do Moveit setup assistant
 - Criar um controlador para comunicação "baixo-nível"



Movel

- Nessa etapa iremos:
 - Criar o arquivo de descrição Xacro/URDF
 - Criar o pacote ROS do manipulador através do Moveit setup assistant
 - Criar um controlador para comunicação "baixo-nível"



Movel

- Nessa etapa iremos:
 - Criar o arquivo de descrição Xacro/URDF
 - Criar o pacote ROS do manipulador através do Moveit setup assistant
 - Criar um controlador para comunicação "baixo nível"



Unity



Unity

- Nessa etapa iremos:
 - Carregar Meshes
 - Configurar juntas
 - Instalação do ROS# e criação do joint_state publisher
 - Criação de action-server para integração ROS -> MoveIt



Unity – Configurando modelo do braço

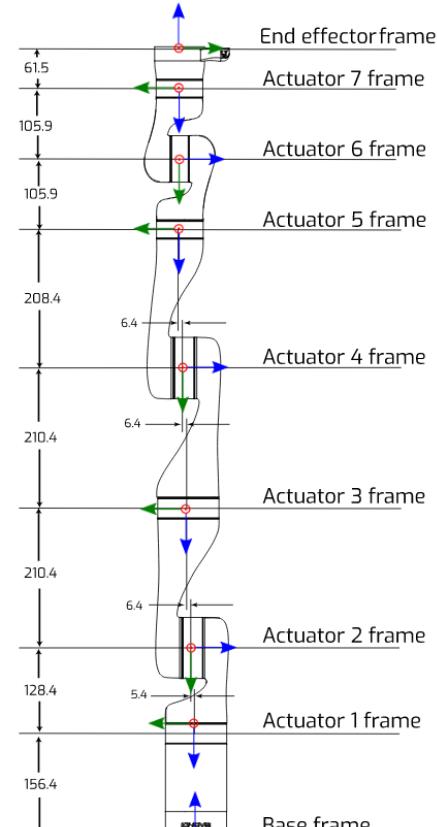


Table 10: Joint limits - 7 DoF spherical wrist

Actuator	Angular range	
	Lower limit	Upper limit
1	$-\infty$	$+\infty$
2	-126°	+126°
3	$-\infty$	$+\infty$
4	-147°	+147°
5	$-\infty$	$+\infty$
6	-117°	+117°
7	$-\infty$	$+\infty$

Table 42: Link 3

Physical quantity	Value
mass (kg)	1.163667 kg

Table 44: Link 5

Physical quantity	Value
mass (kg)	0.678106

Table 43: Link 4

Physical quantity	Value
mass (kg)	0.930287

Table 45: Link 6

Physical quantity	Value
mass (kg)	0.678106

Table 39: Base

Physical quantity	Value
mass (kg)	1.697353

Table 40: Link 1

Physical quantity	Value
mass (kg)	1.377353

Table 41: Link 2

Physical quantity	Value
mass (kg)	1.163667

Table 47: Interface & Vision Module

Physical quantity	Value
mass (kg)	0.500657



Unity

- Nessa etapa iremos:
 - ~~Carregar Meshes~~
 - Configurar juntas
 - Instalação do ROS# e criação do joint_state publisher
 - Criação de action-server para integração ROS -> MoveIt



Unity

- Nessa etapa iremos:
 - ~~Carregar Meshes~~
 - ~~Configurar juntas~~
 - Instalação do ROS# e criação do joint_state publisher
 - Criação de action-server para integração ROS -> MoveIt



Unity

- Nessa etapa iremos:
 - ~~Carregar Meshes~~
 - ~~Configurar juntas~~
 - ~~Instalação do ROS# e criação do joint_state publisher~~
 - Criação de action-server para integração ROS -> MoveIt



Unity

- Nessa etapa iremos:
 - ~~Carregar Meshes~~
 - ~~Configurar juntas~~
 - ~~Instalação do ROS# e criação do joint_state publisher~~
 - ~~Criação de action server para integração ROS -> MoveIt~~



Dúvidas ou comentários?





warthog
robotics