warthog robotics

Tutorial MoveIt e Unity para Manipulação Robótica

Warthog – USP São Carlos, SP

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

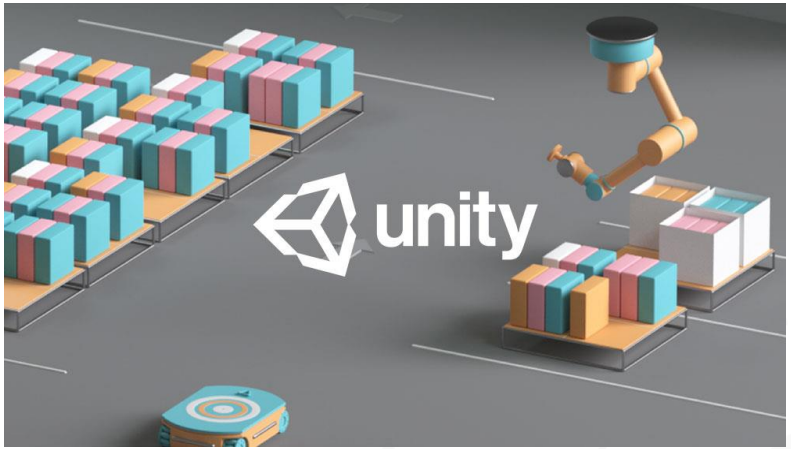
Atlas
Boston Dynamics

KR 210
KUKA

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

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

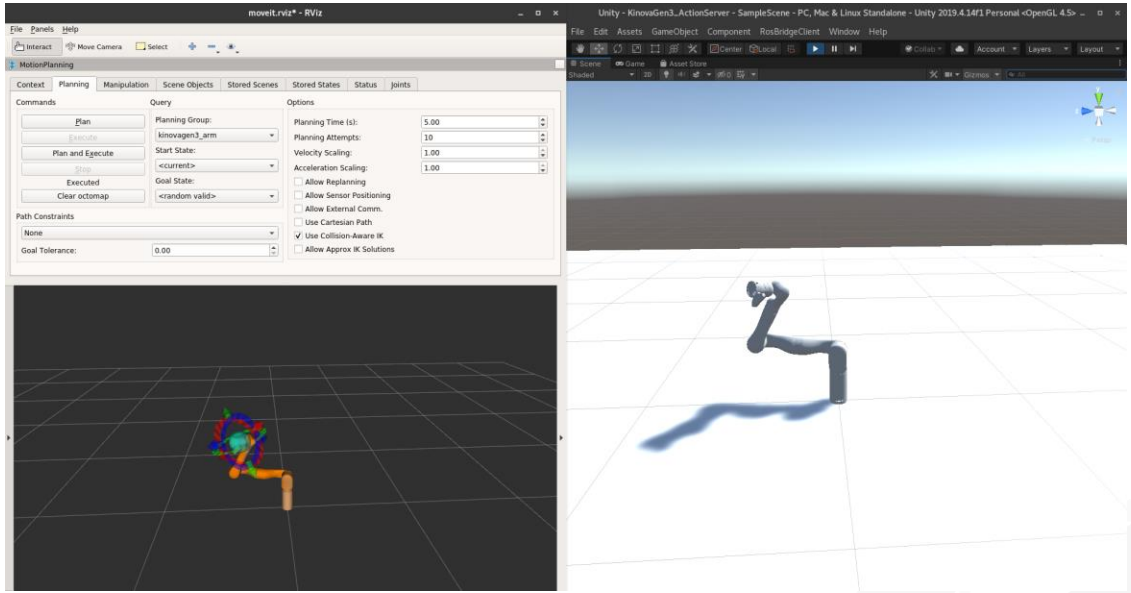


Imagens retiradas de: https://unity.com/solutions/automotive-transportation-manufacturing/robotics

Objetivo

Objetivo

- Mostrar etapas do processo de contrução 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

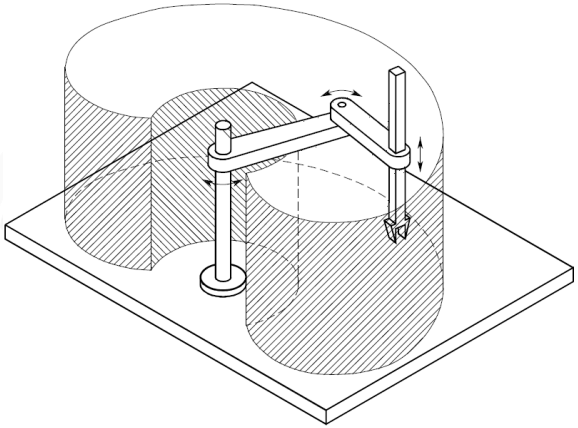
    2. Configurar juntas

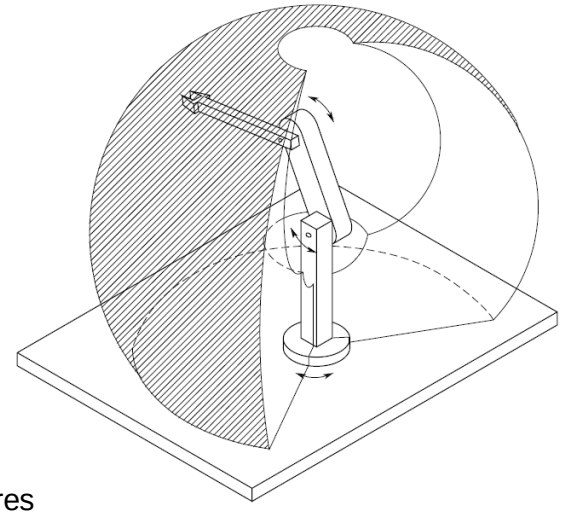
    3. Instalação do ROS#

    4. Criação de action-server para integração com ROS e MoveIt

Fundamentação teórica

Fundamentação teórica - Manipuladores





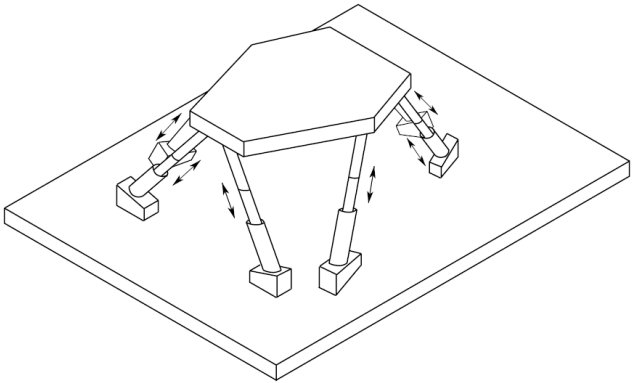


Imagem 1. Exemplos de robôs manipuladores

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

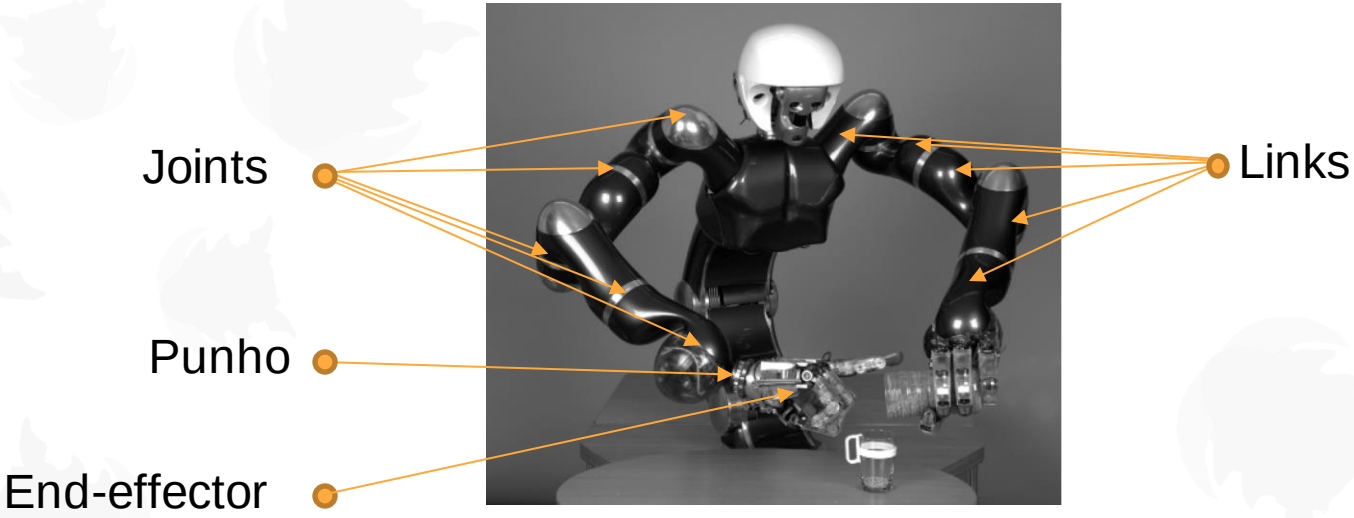


Imagem 2 - Robô humanoide Justin, desenvolvido pela DLR.

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

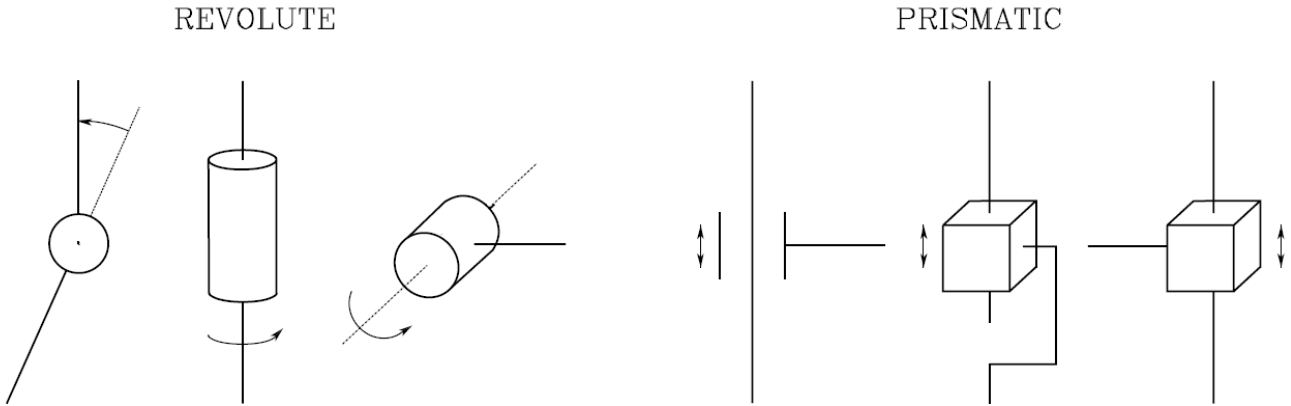
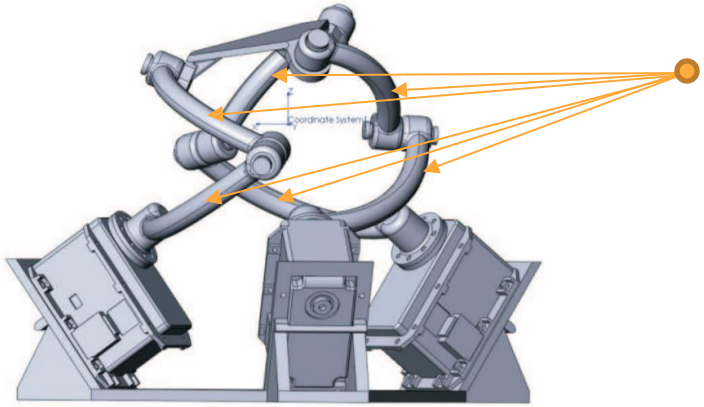


Imagem 3 - Representação típica de graus de liberdade.

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



"Links"

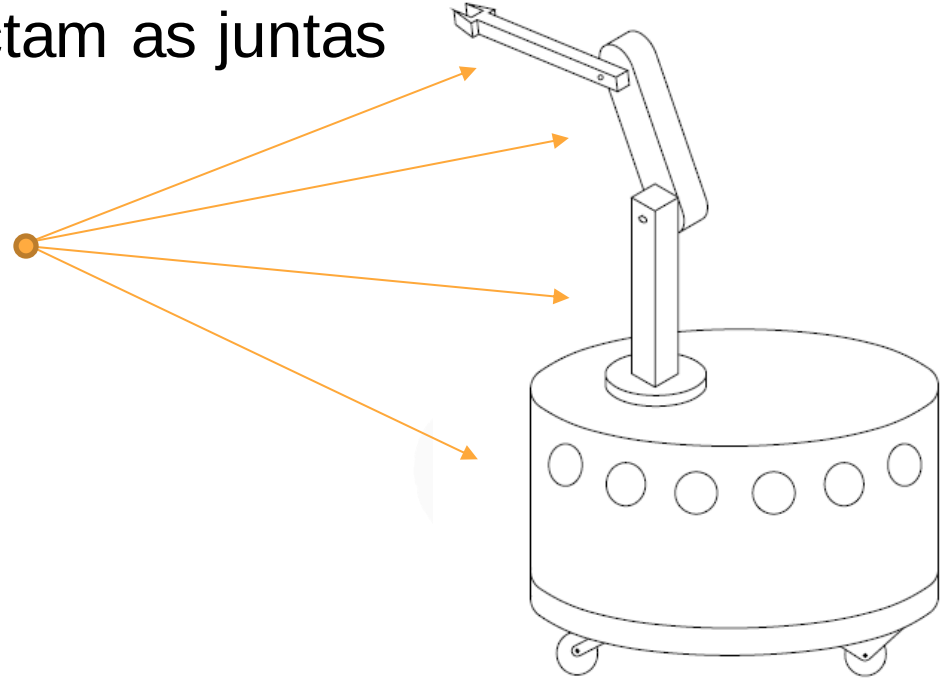


Imagem 4 - Manipulador esférico paralelo.
Niyetkaliyev, Aibek, and Almas Shintemirov. "An approach for obtaining unique kinematic solutions of a spherical parallel manipulator." *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2014.

Imagem 5 - Manipulador serial em base móvel.
Siciliano, Bruno, et al. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

Fundamentação teórica - Atuadores

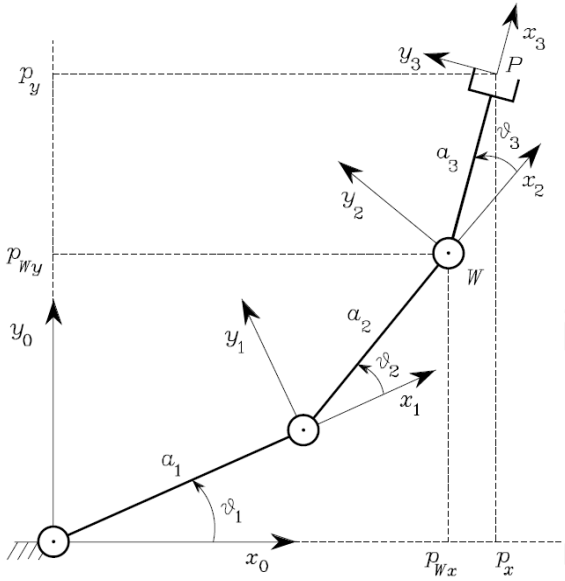
Atuadores são responsáveis por causar uma mudança de estado em um dos graus de liberdade (ex. motores elétricos ou pistões).

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

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

Fundamentação teórica - Cinemática Direta

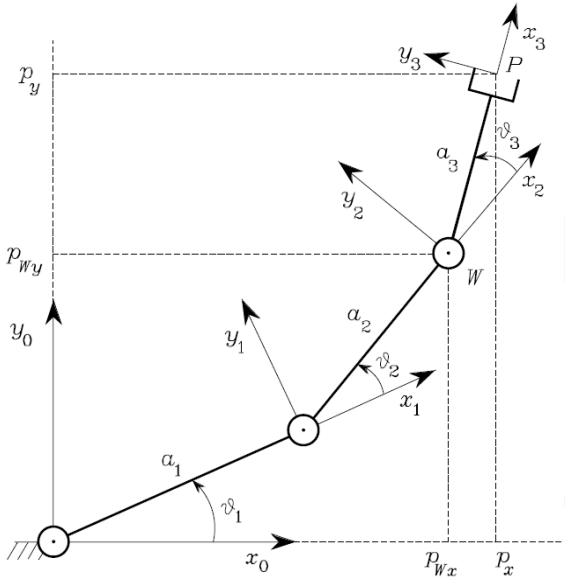
Cinemática direta é o ato de calcular a posição e orientação do "end-effector"



Todas citações e imagem 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 imagem 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

Imagem 7 - Planejamento de trajetória discretizado

Imagem 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*.

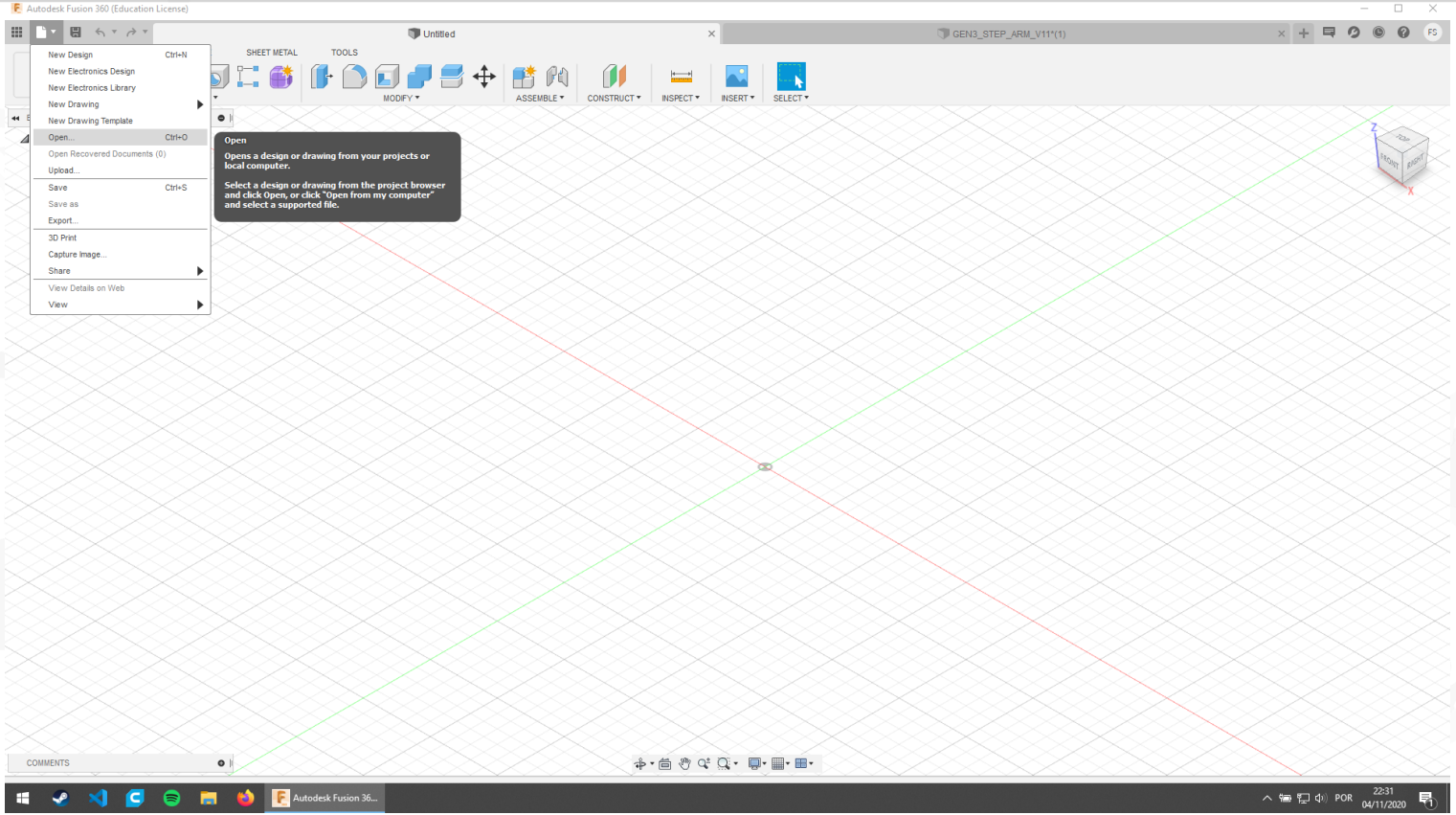
Fusion 360

Fusion 360 - Intro

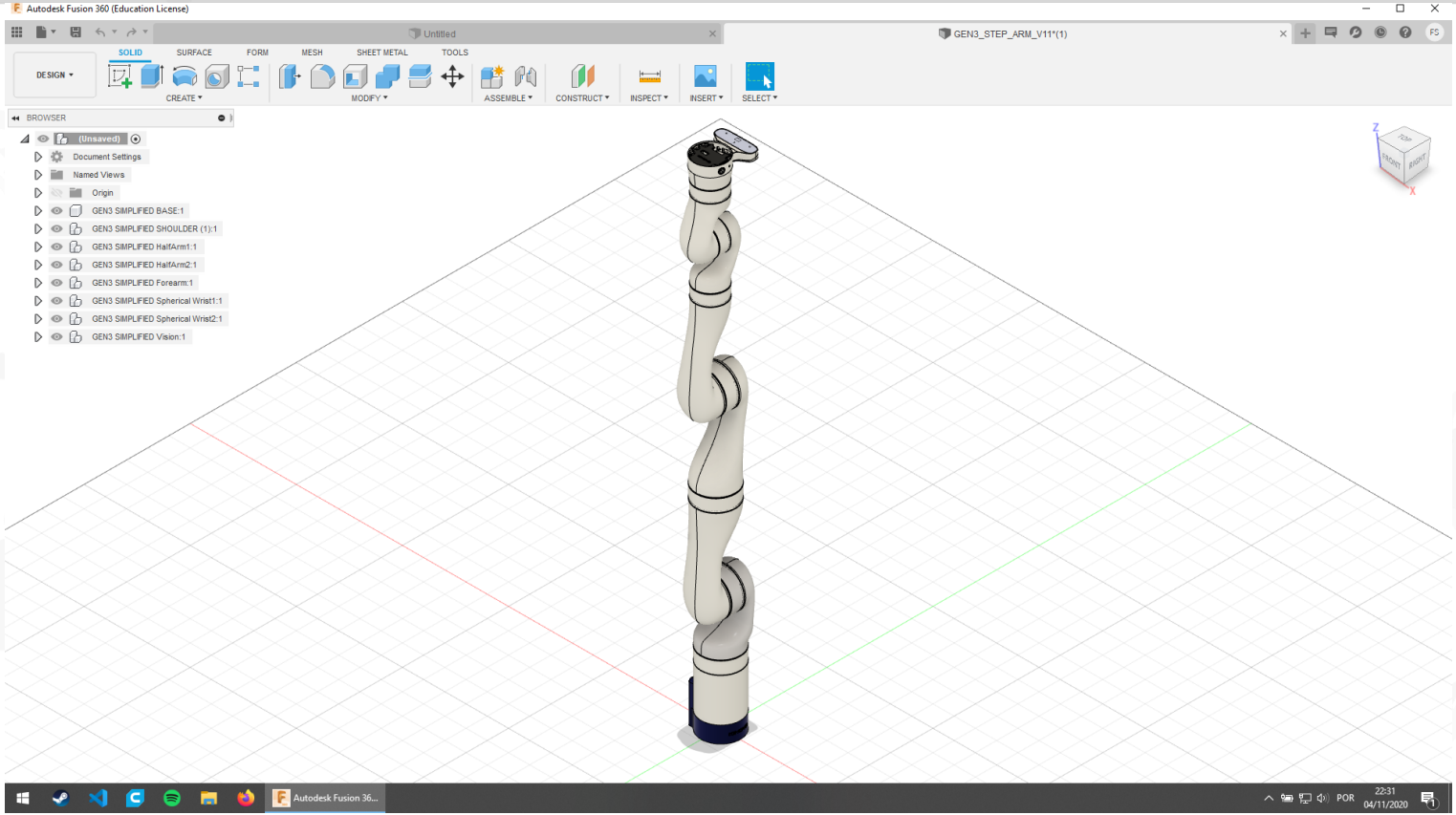
Fusion 360 - Intro

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

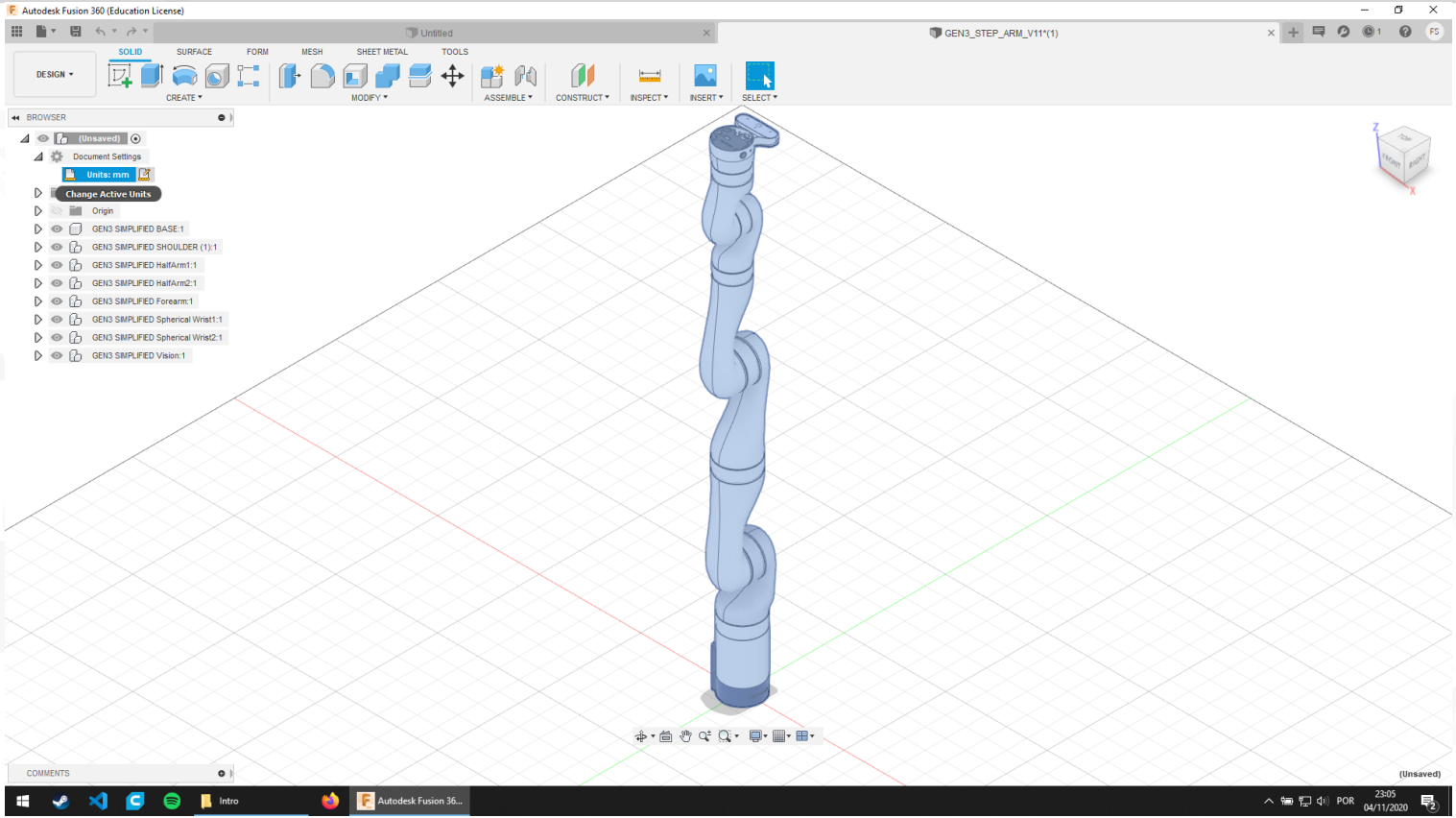
Fusion 360 - Intro



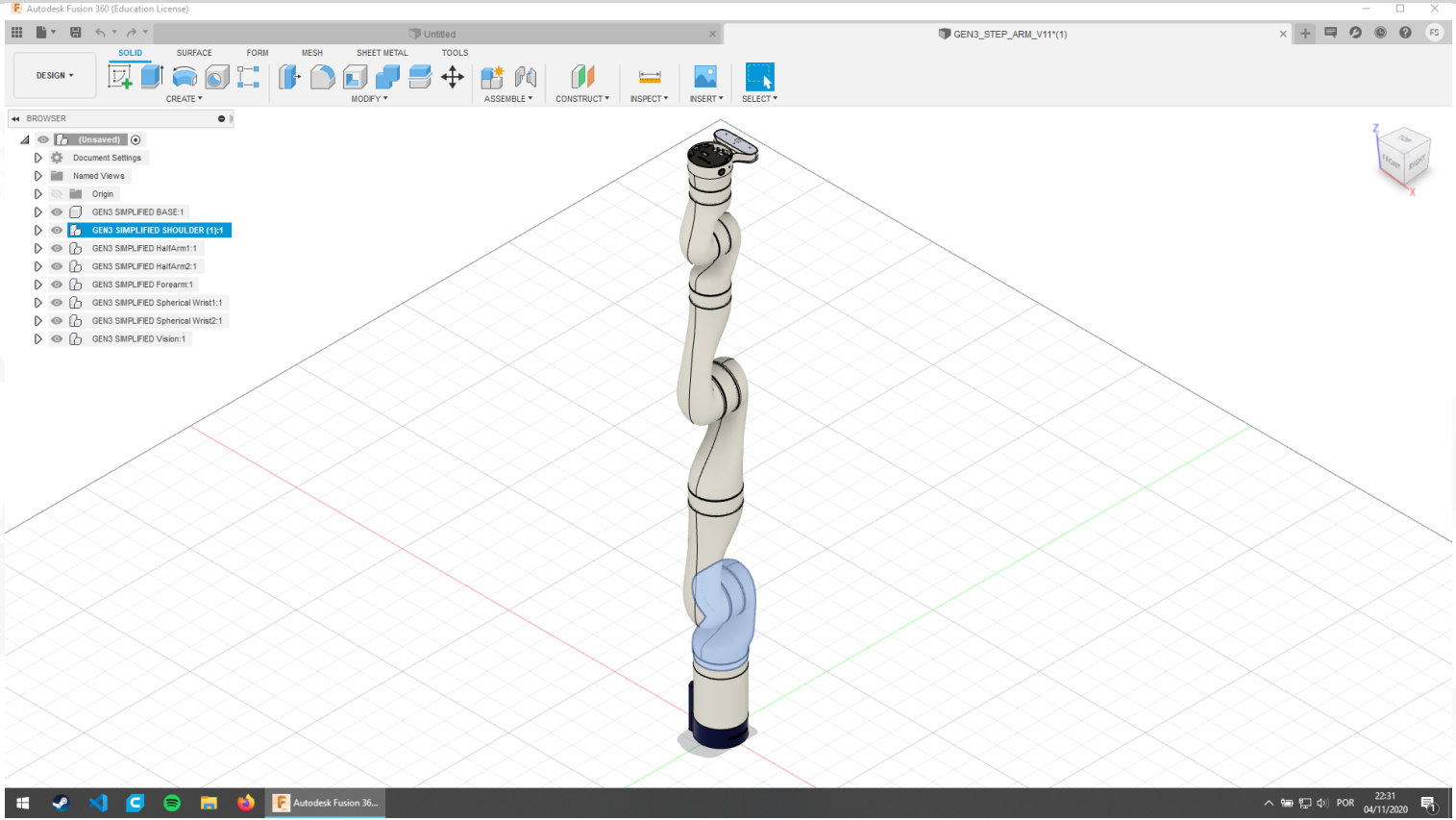
Fusion 360 - Intro



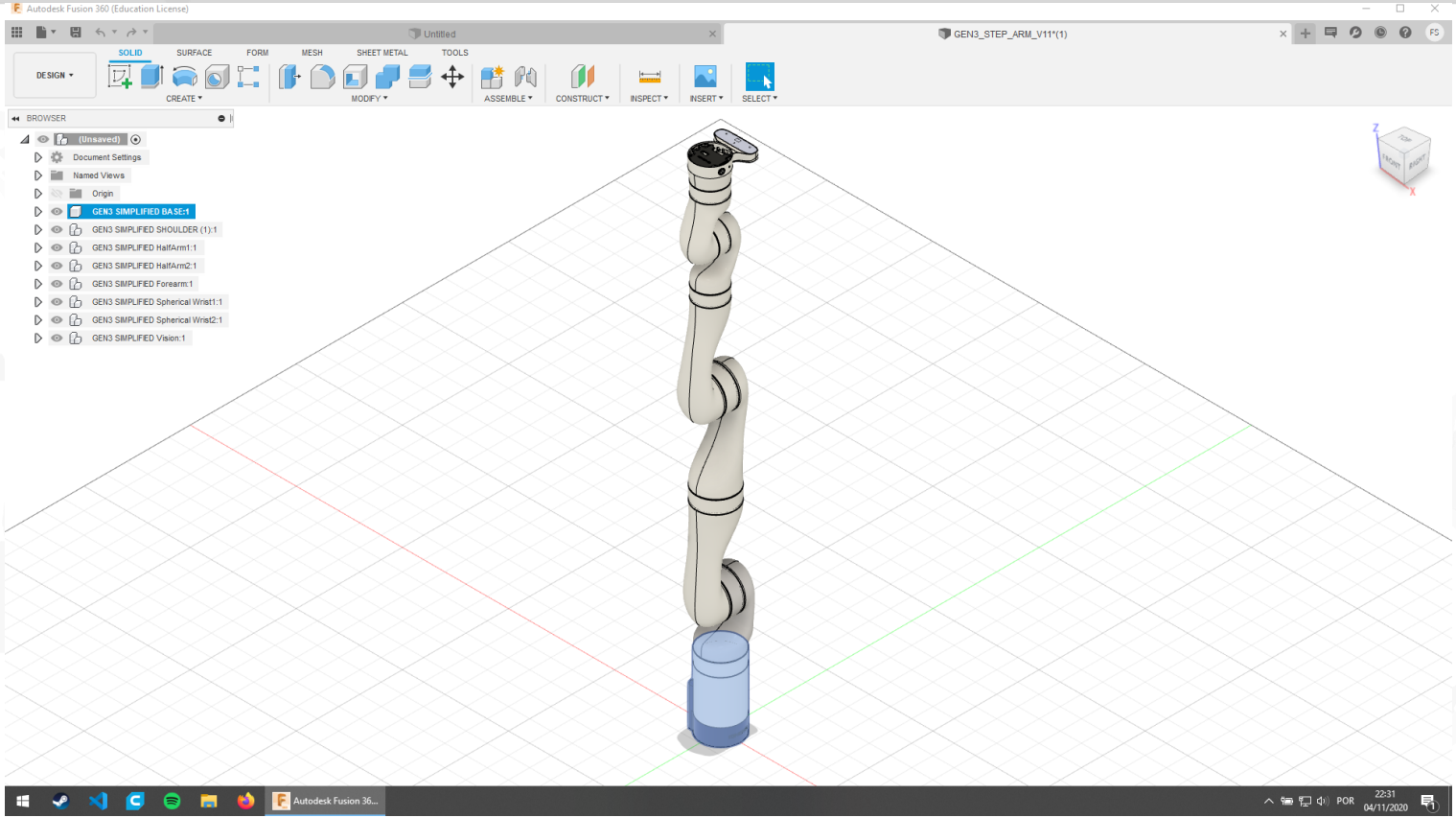
Fusion 360 - Meshes



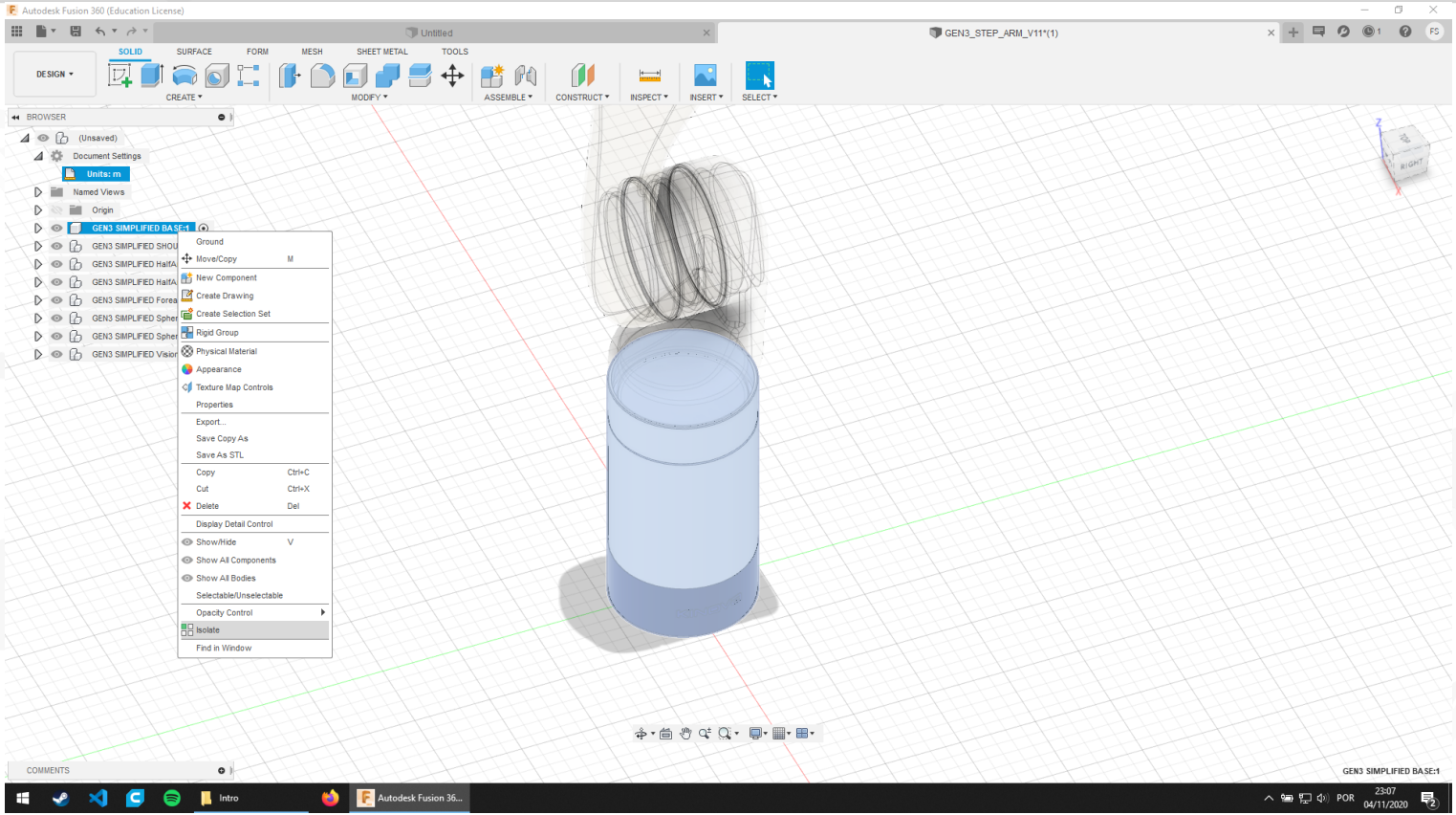
Fusion 360 - Intro



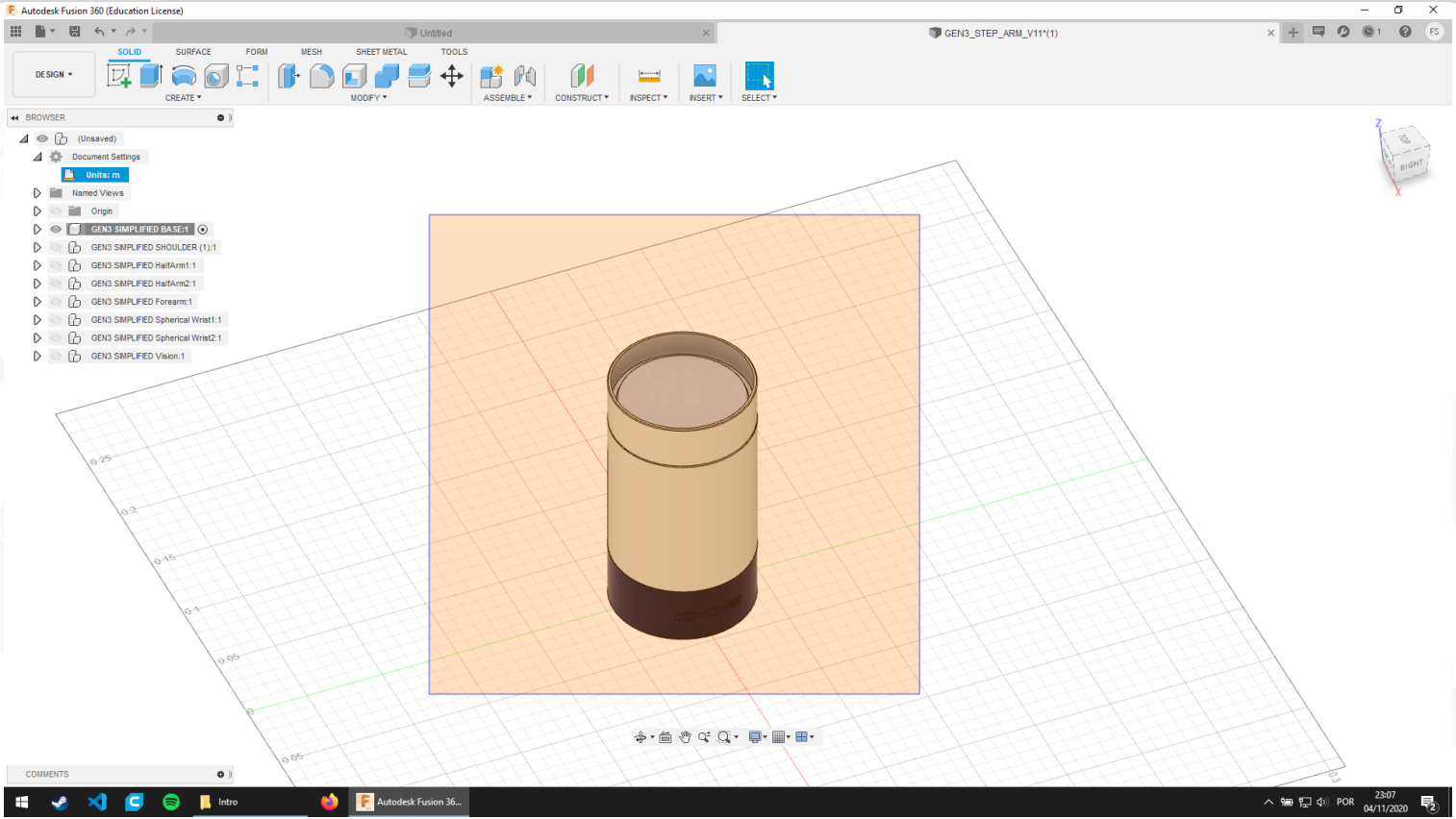
Fusion 360 - Intro



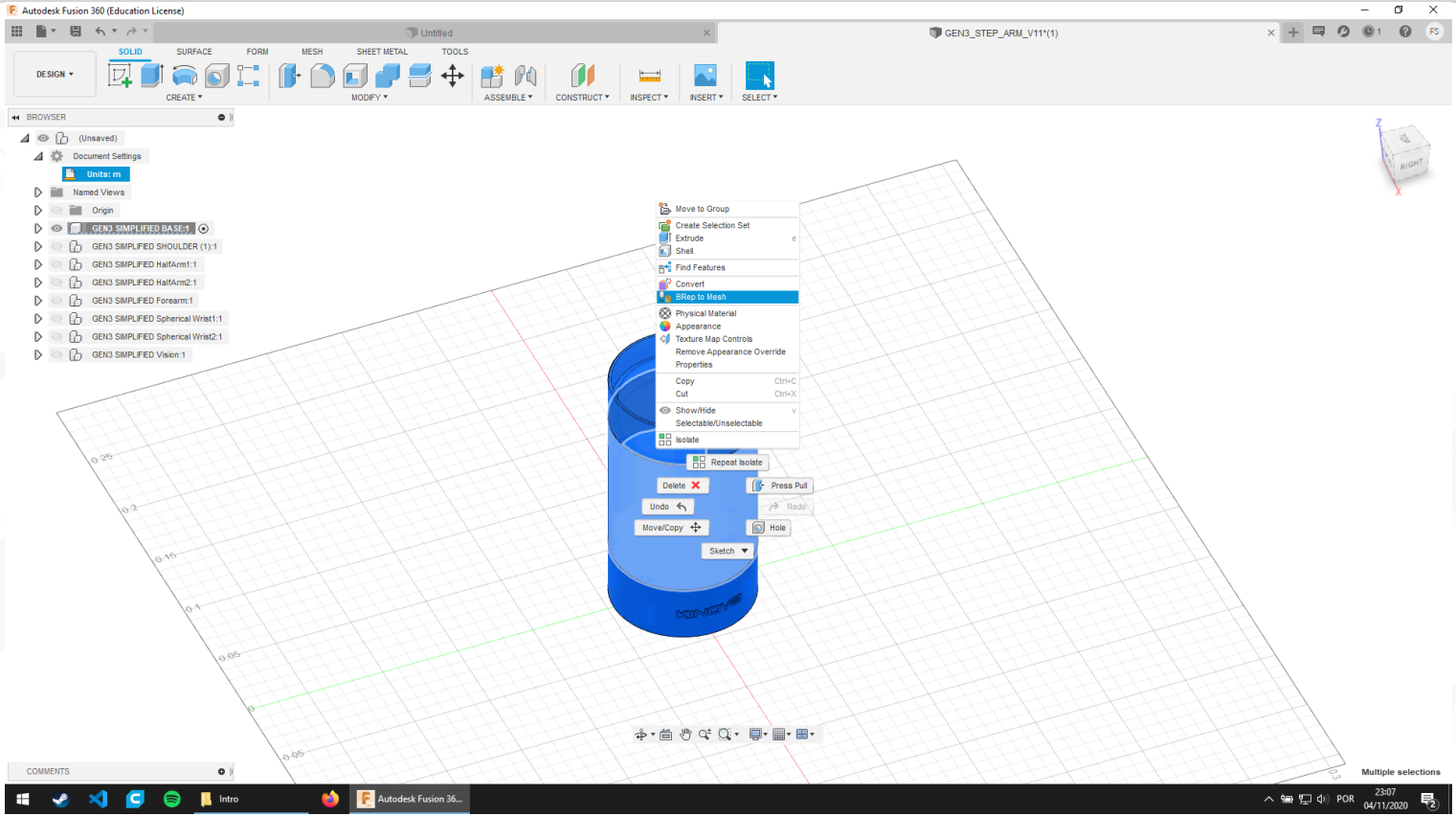
Fusion 360 - Meshes



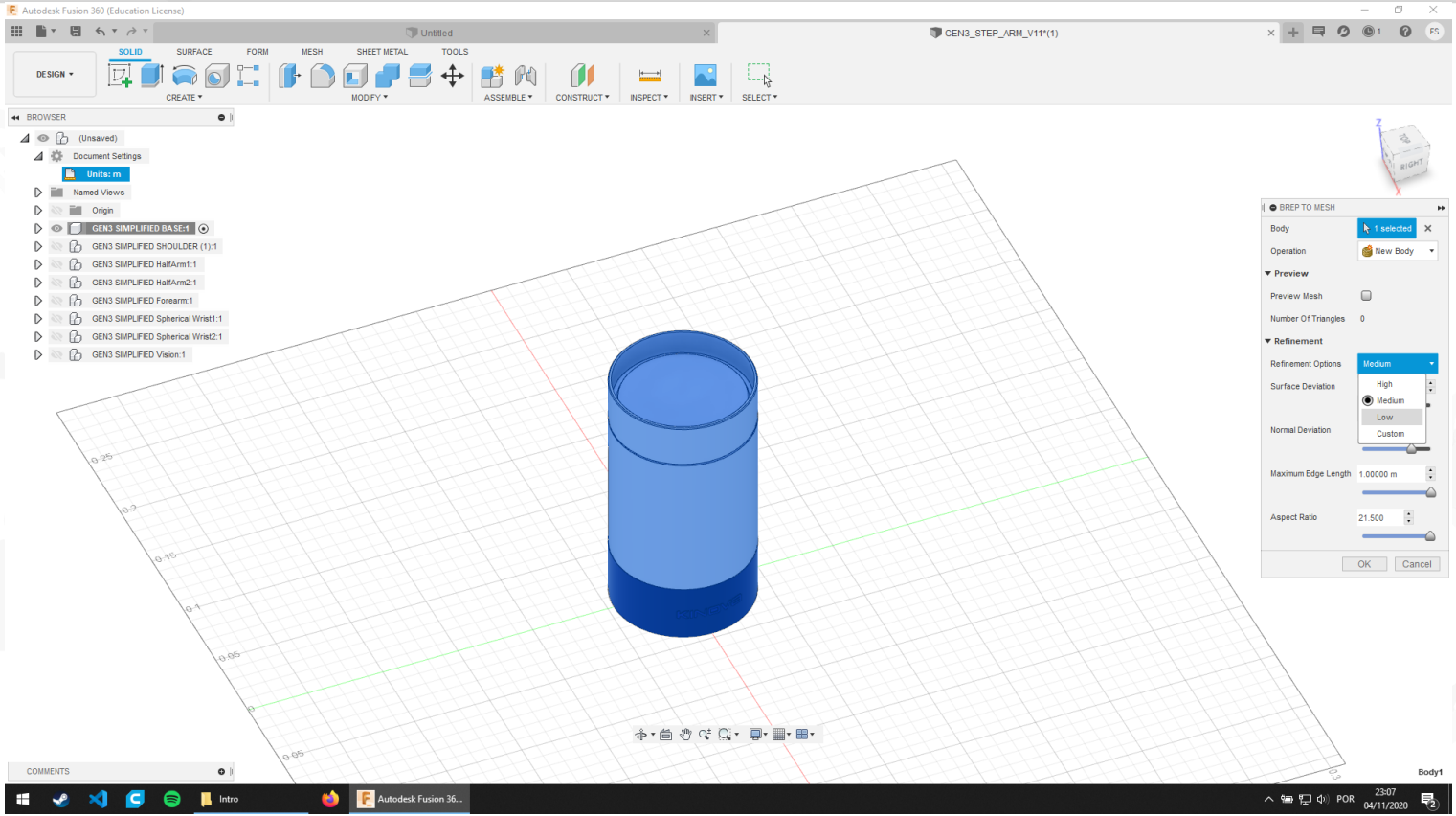
Fusion 360 - Meshes



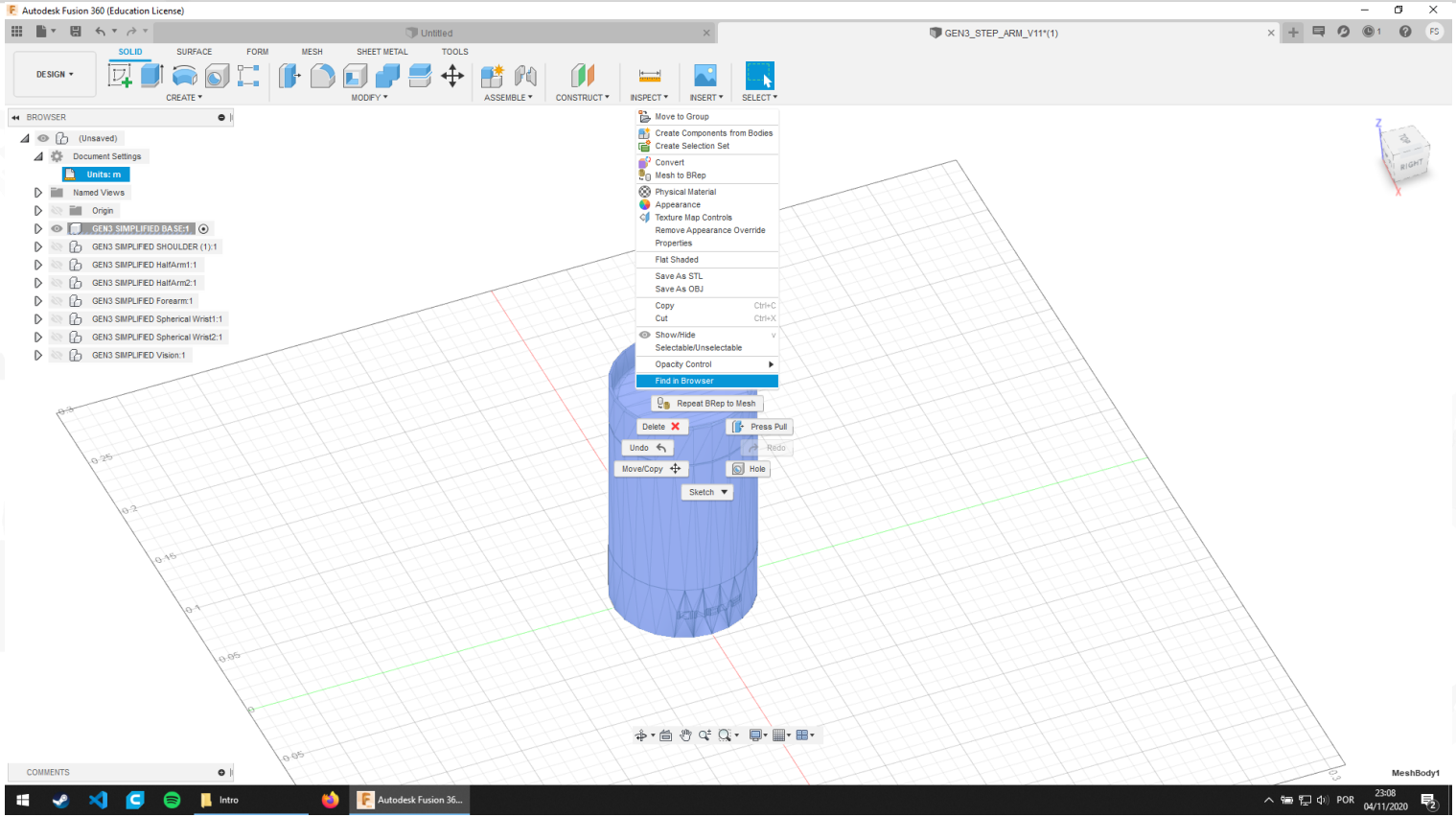
Fusion 360 - Meshes



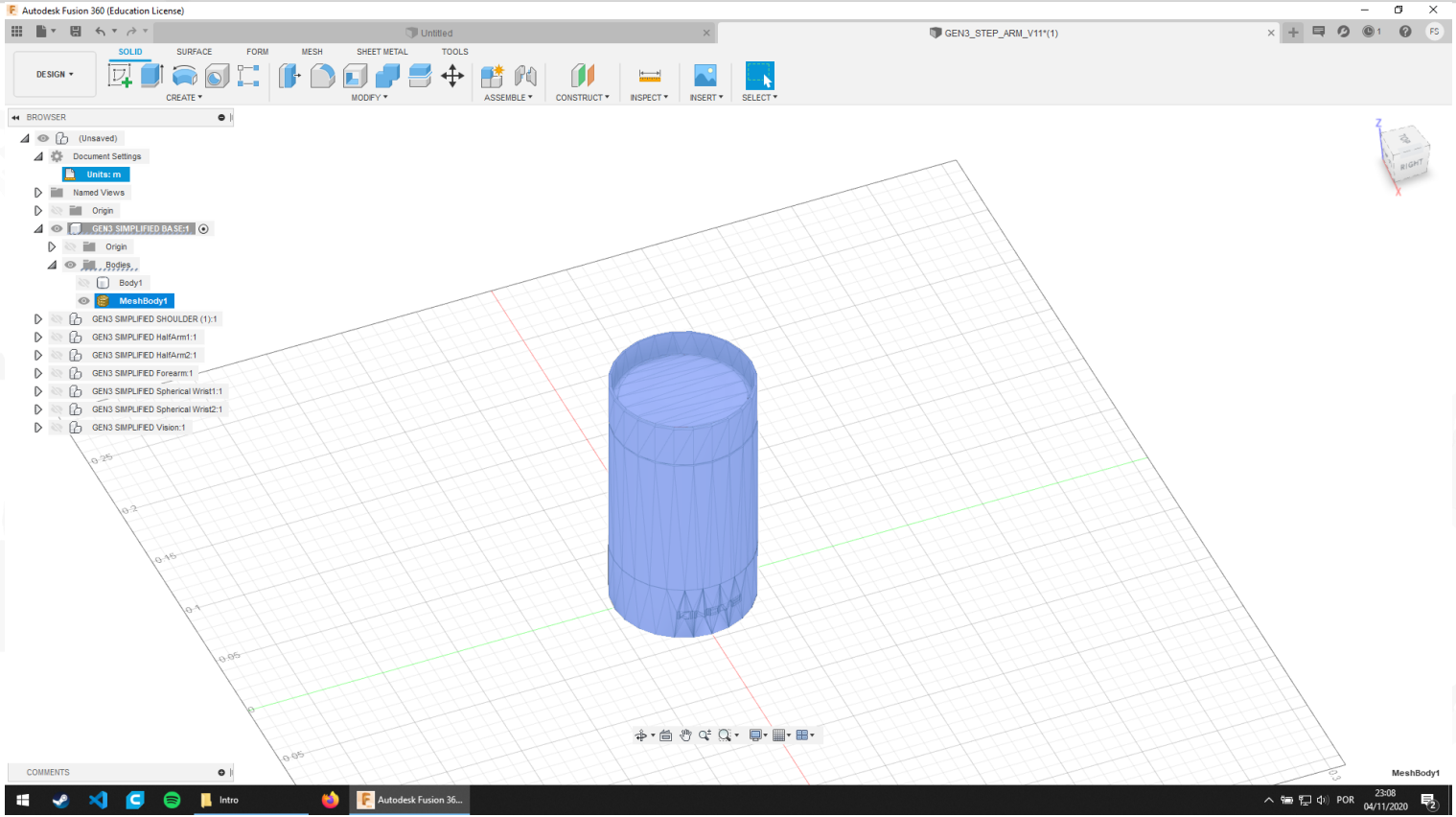
Fusion 360 - Meshes



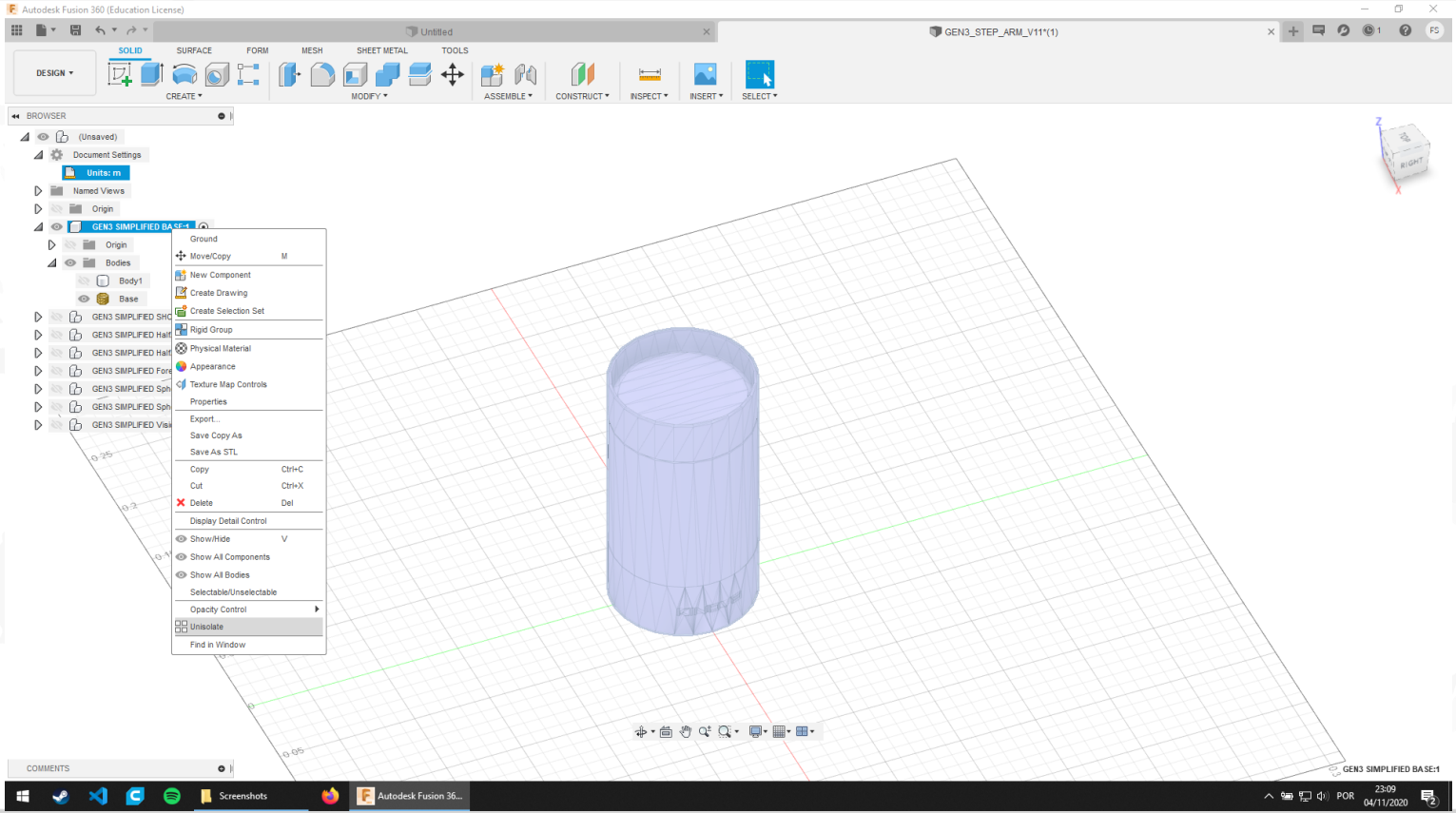
Fusion 360 - Meshes



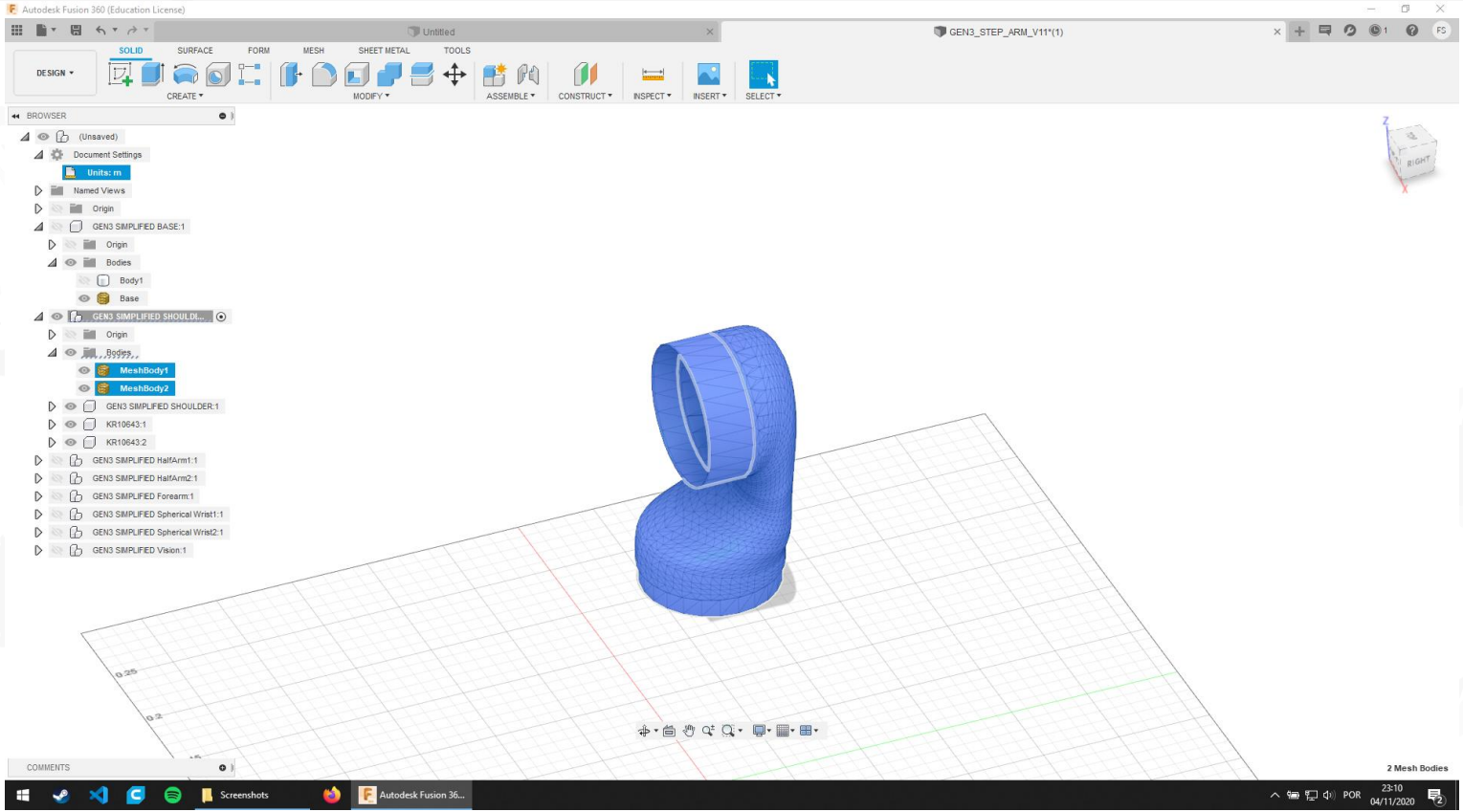
Fusion 360 - Meshes



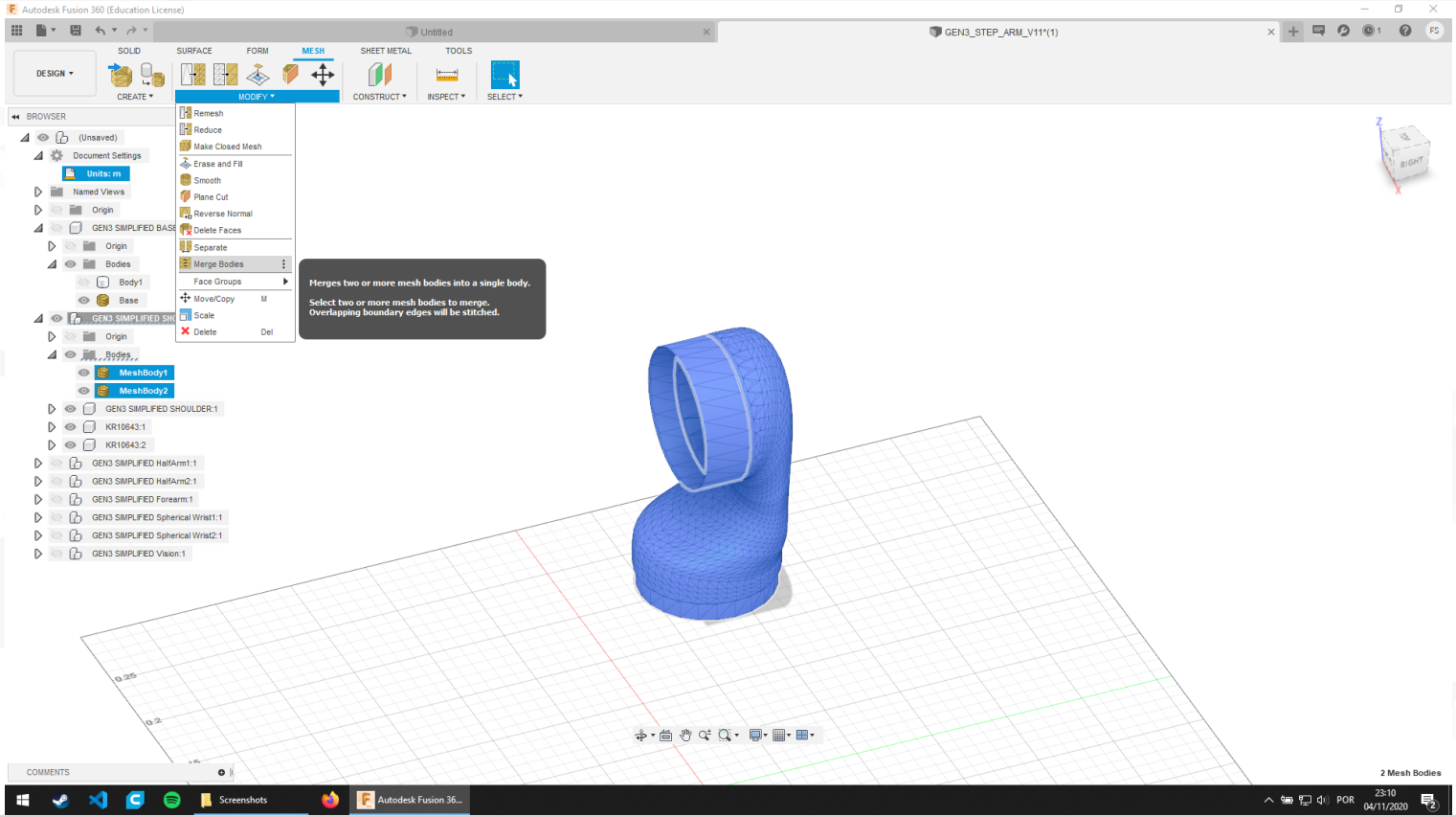
Fusion 360 - Meshes



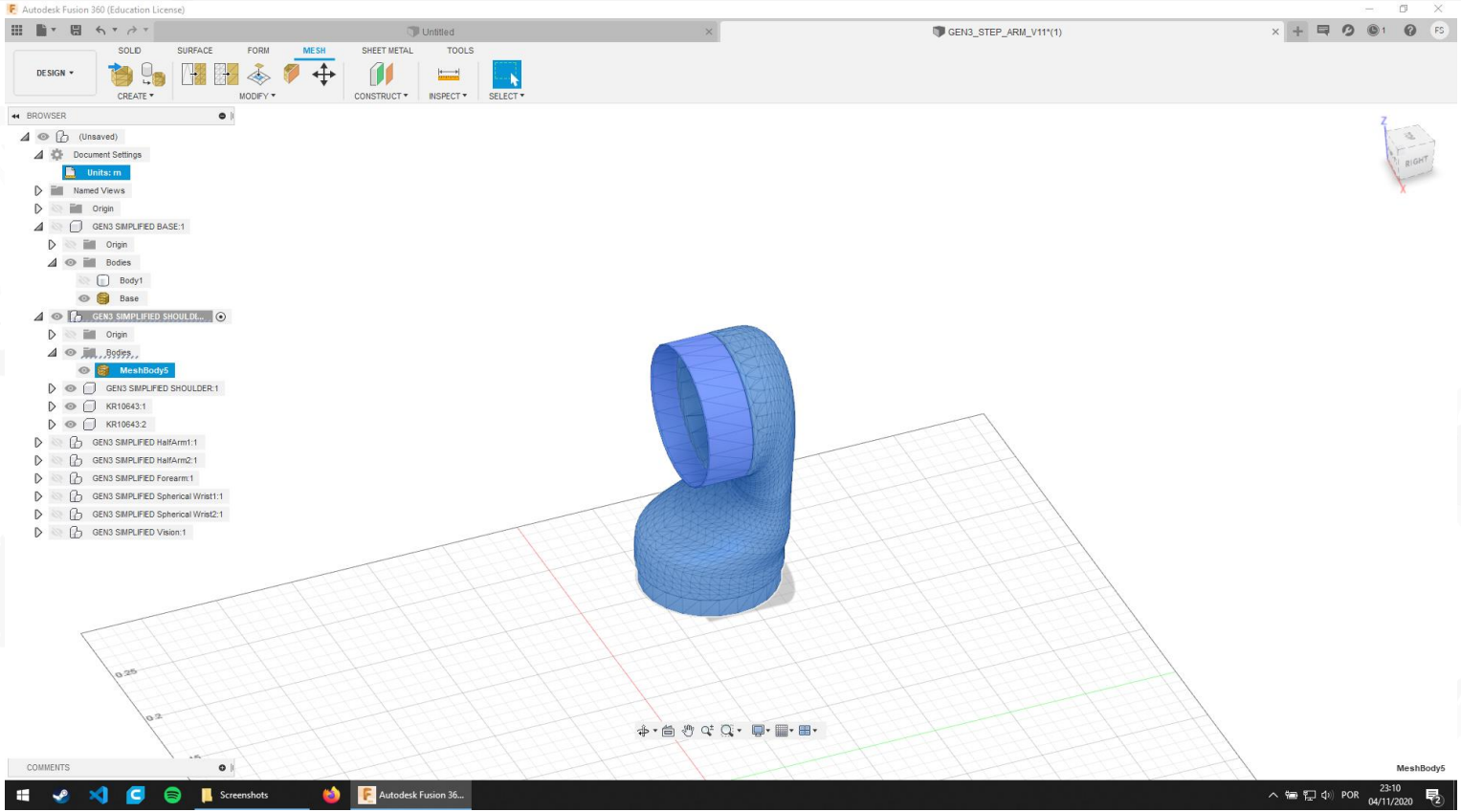
Fusion 360 - Meshes



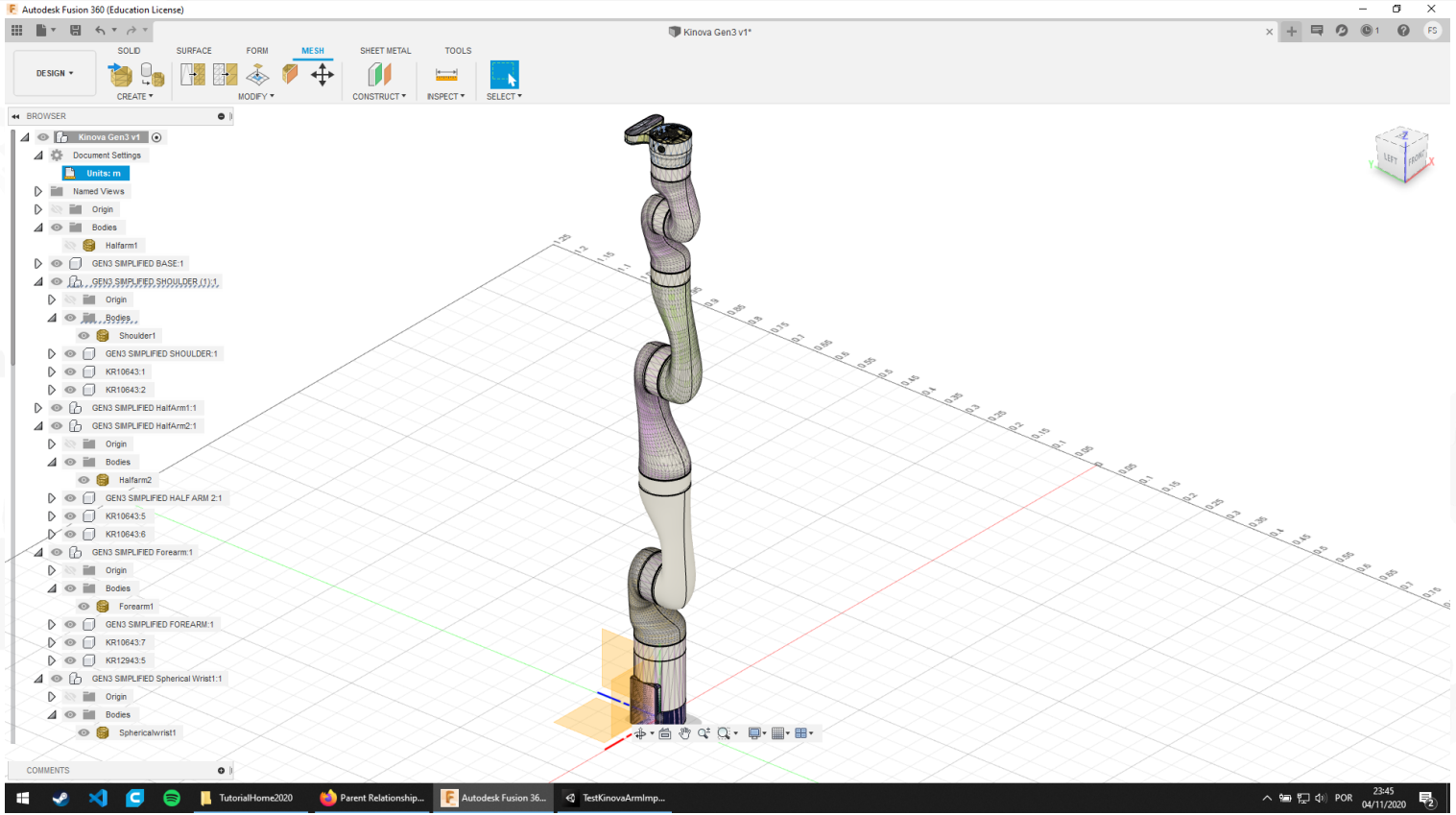
Fusion 360 - Meshes



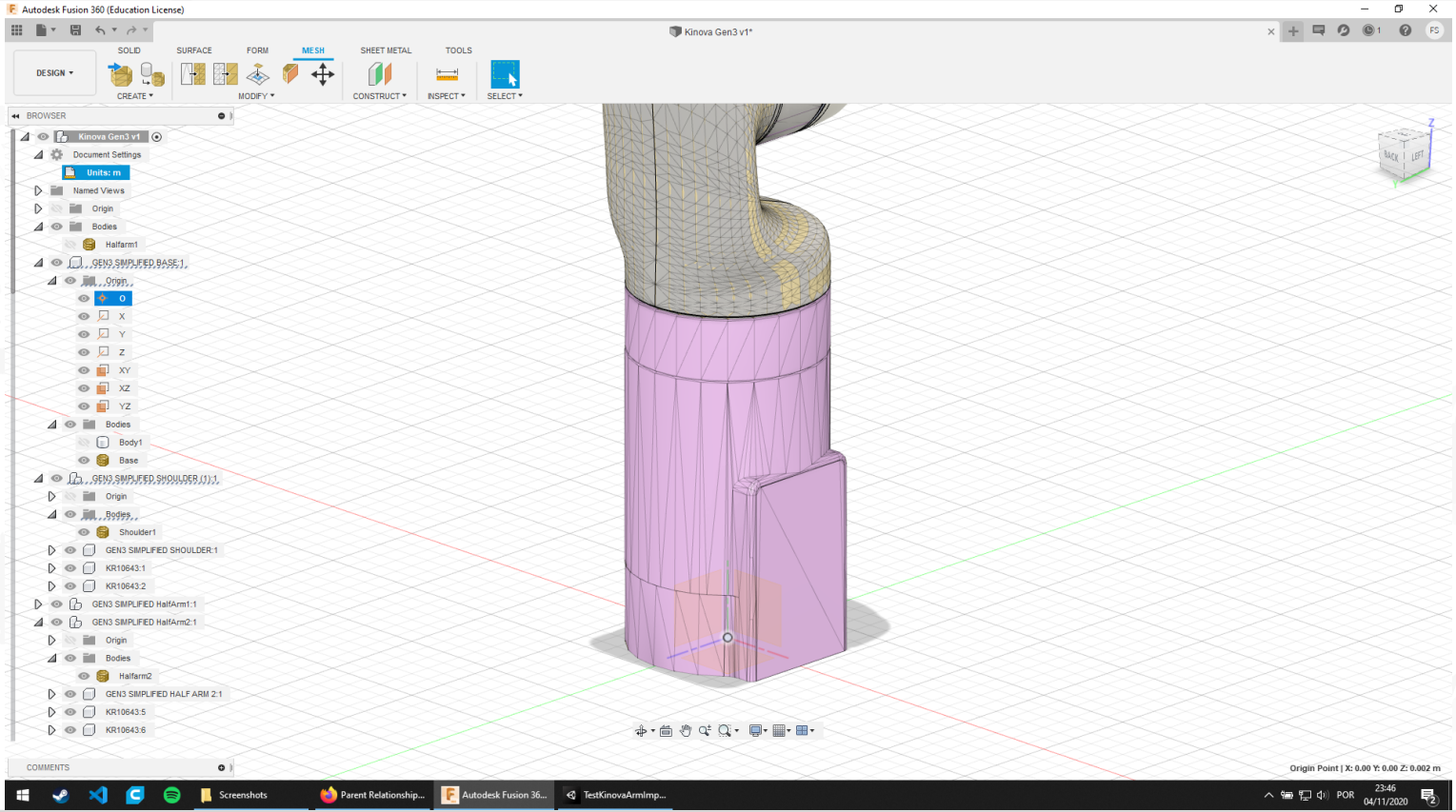
Fusion 360 - Meshes



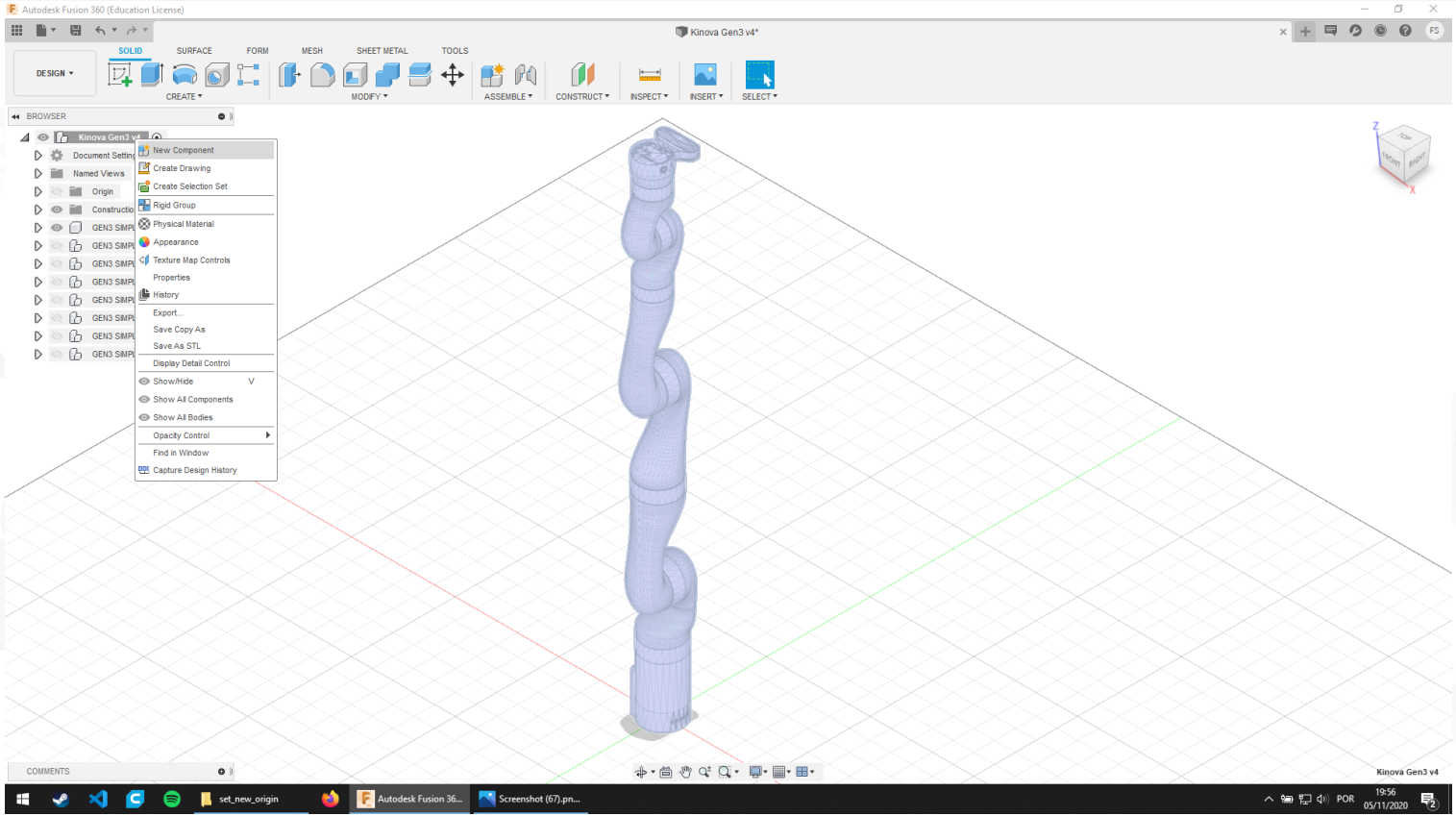
Fusion 360 - Meshes



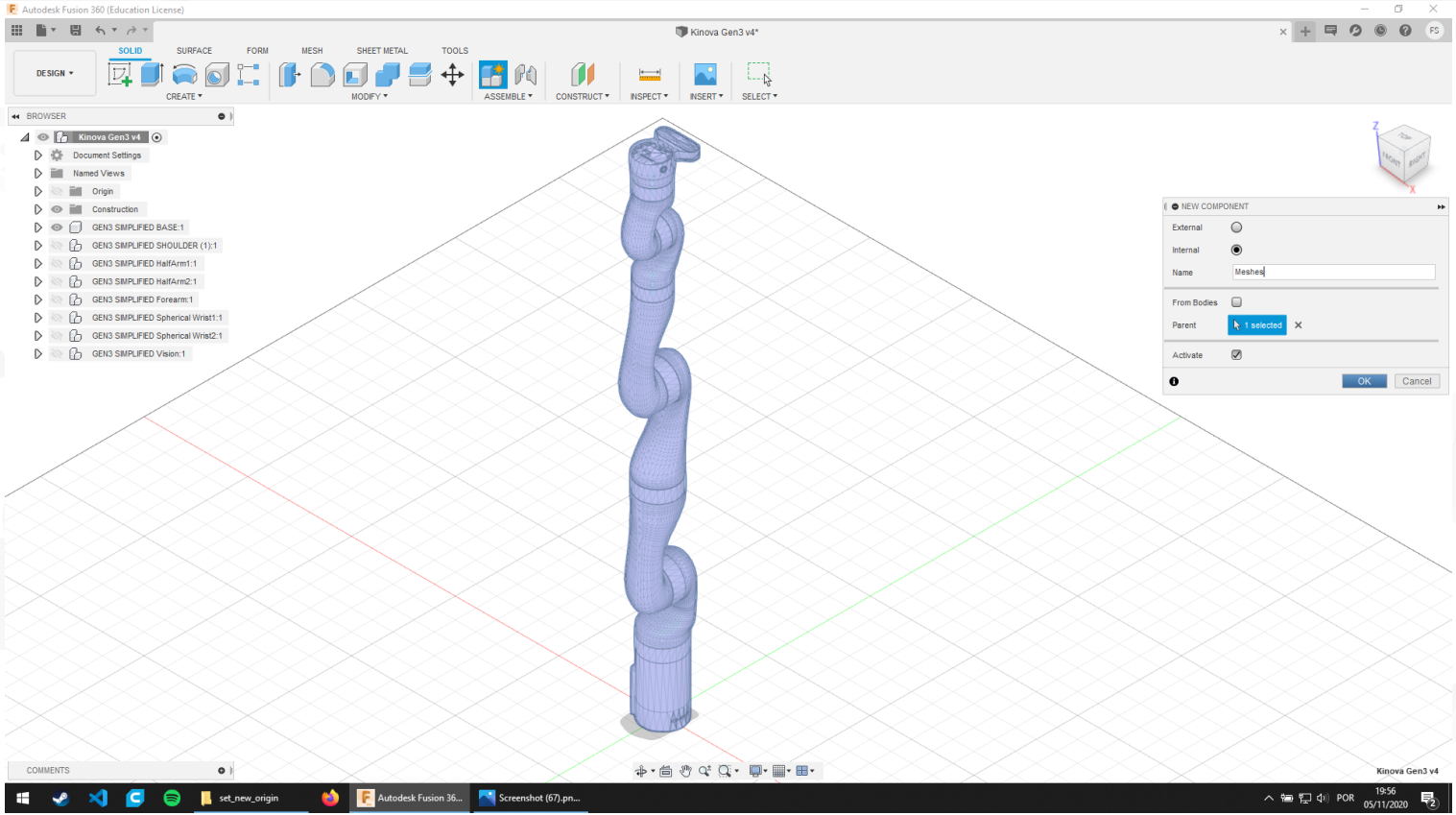
Fusion 360 - Meshes



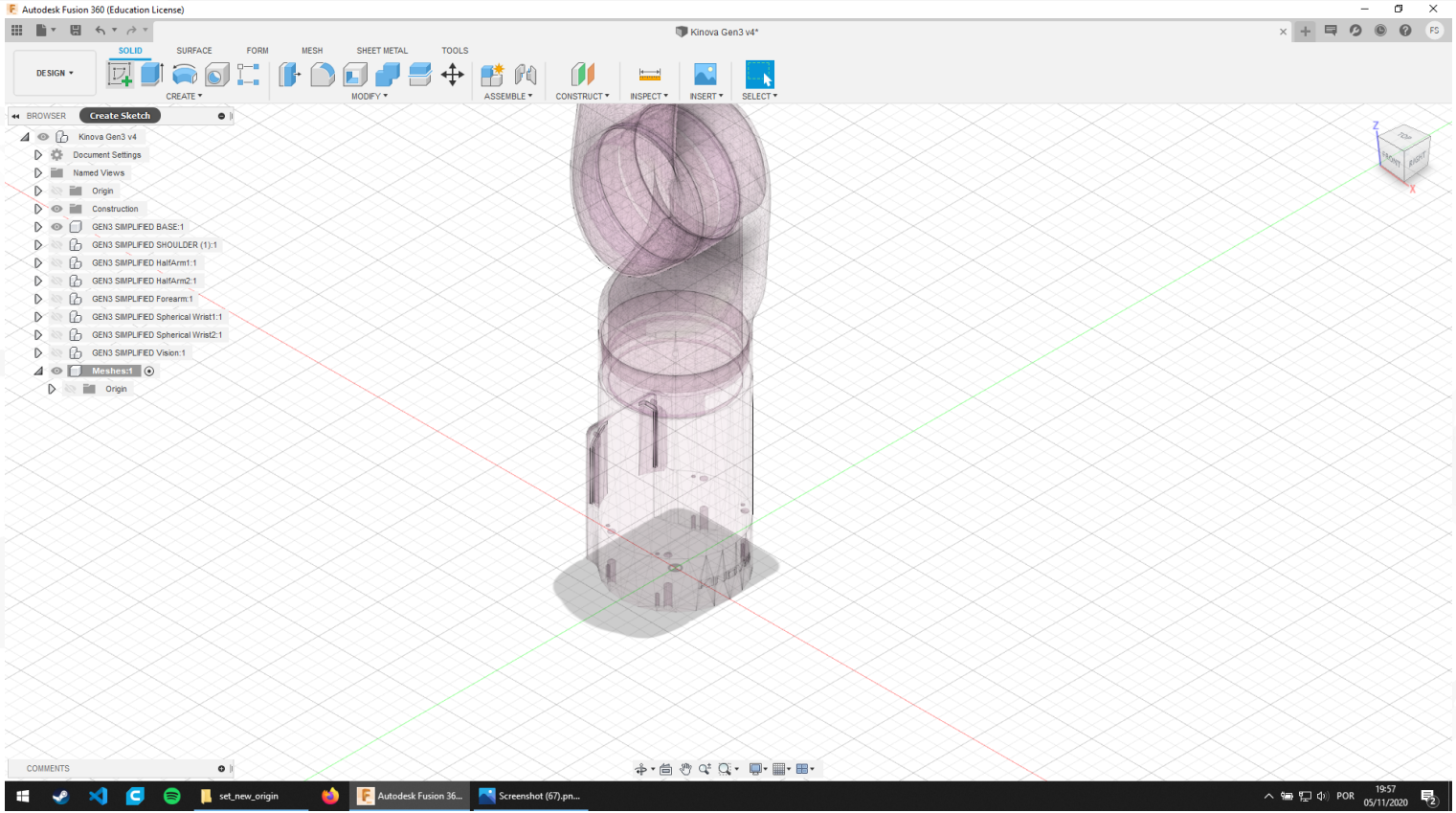
Fusion 360 - Meshes



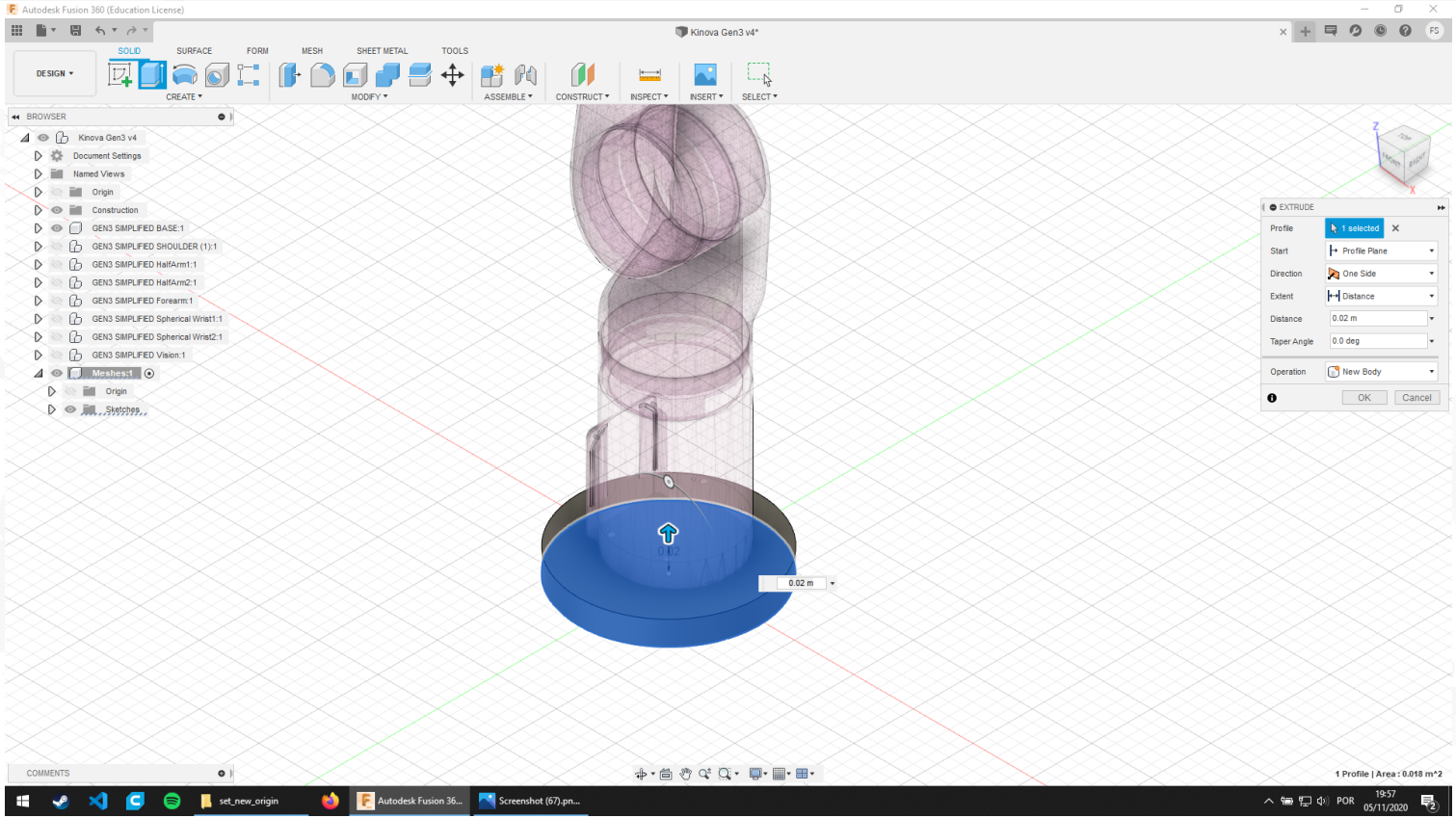
Fusion 360 - Meshes



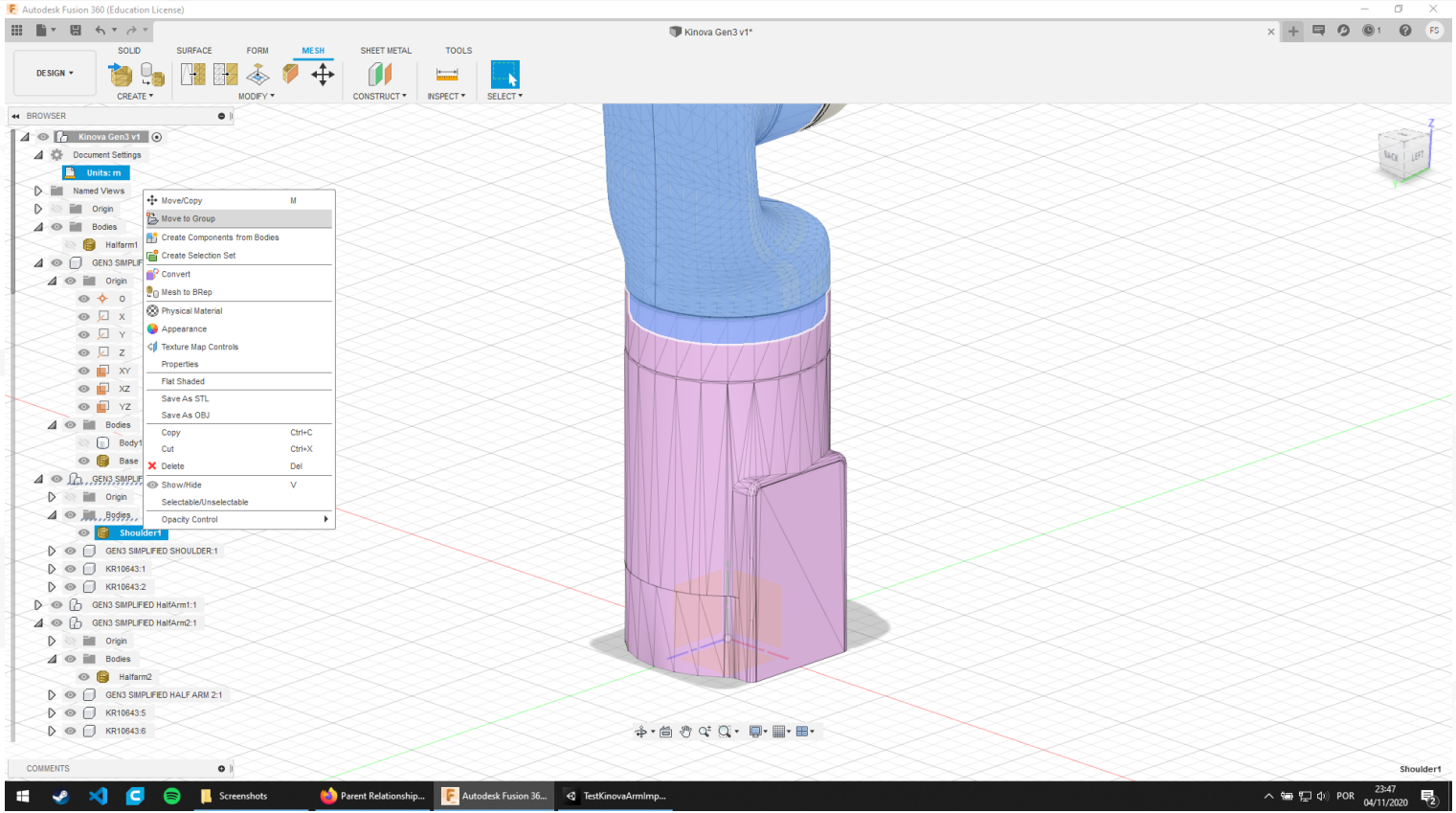
Fusion 360 - Meshes



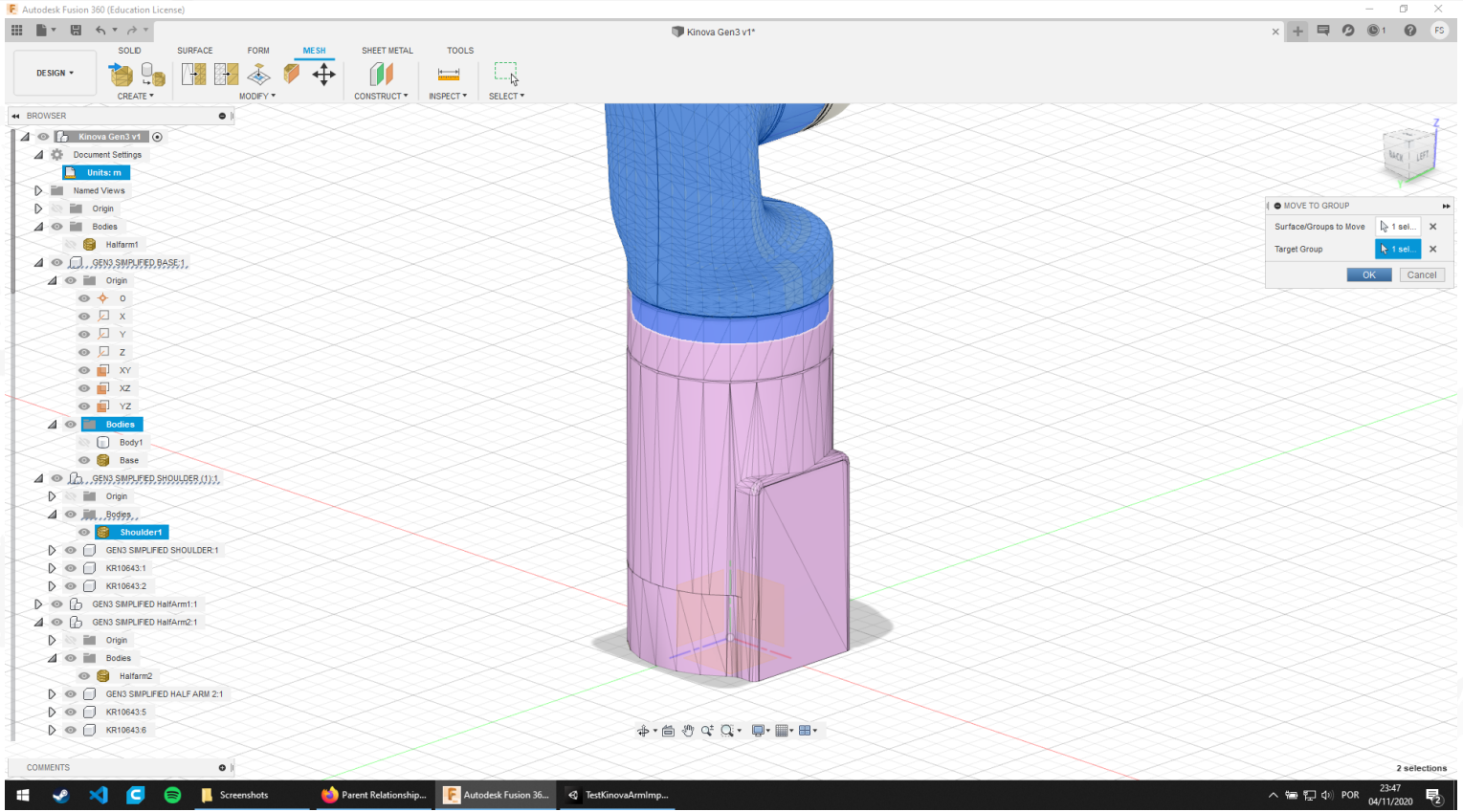
Fusion 360 - Meshes



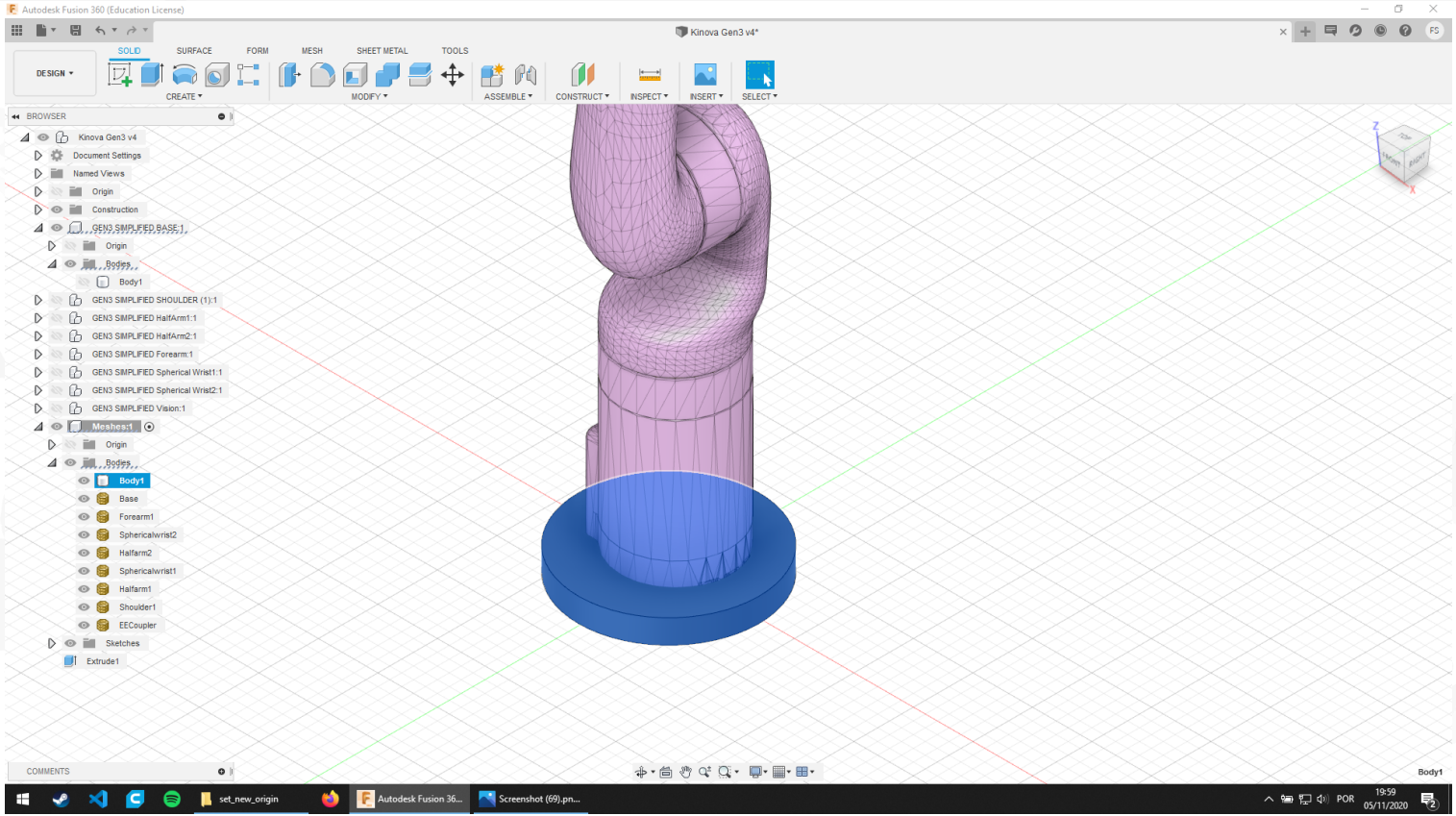
Fusion 360 - Meshes



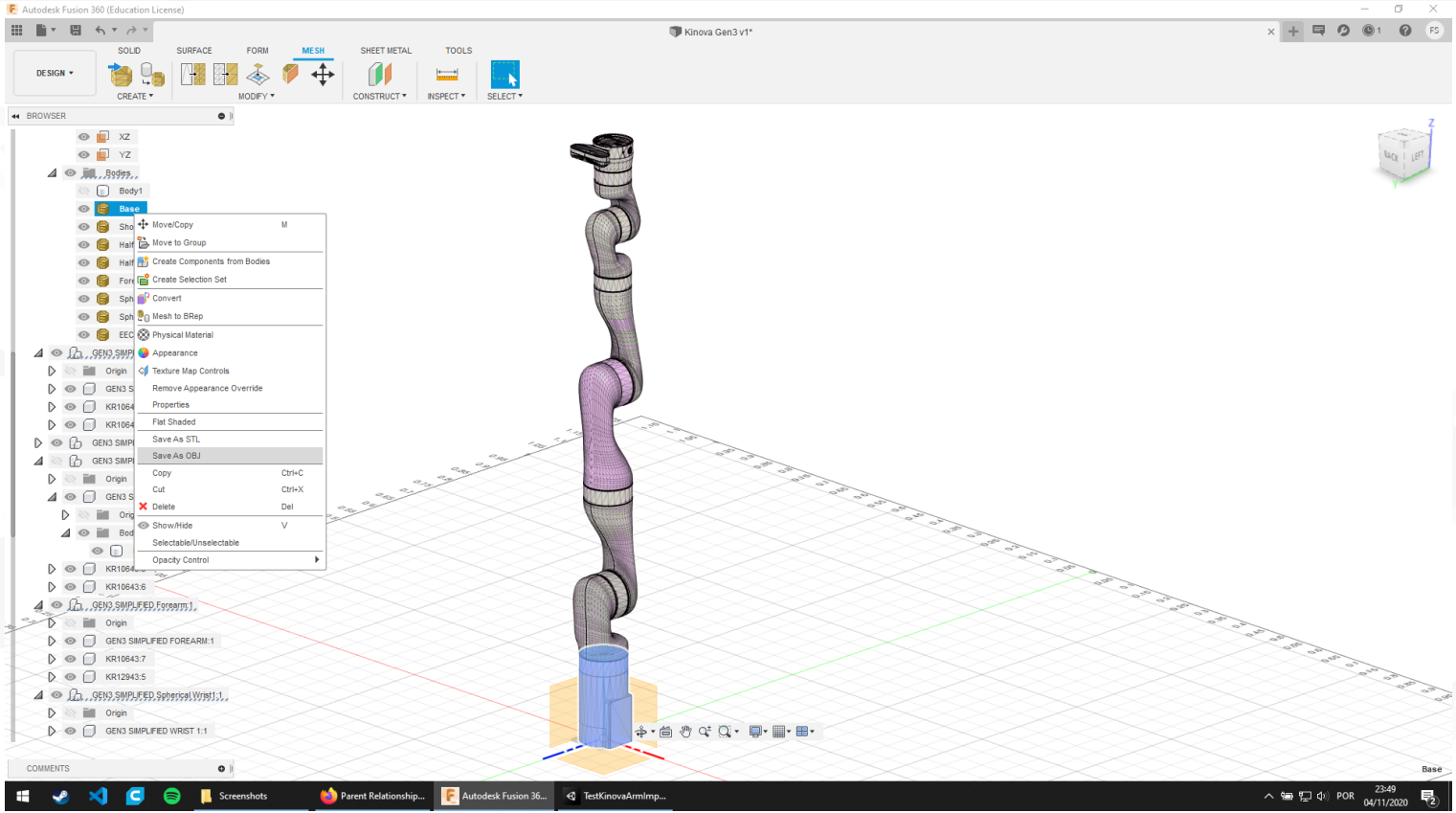
Fusion 360 - Meshes



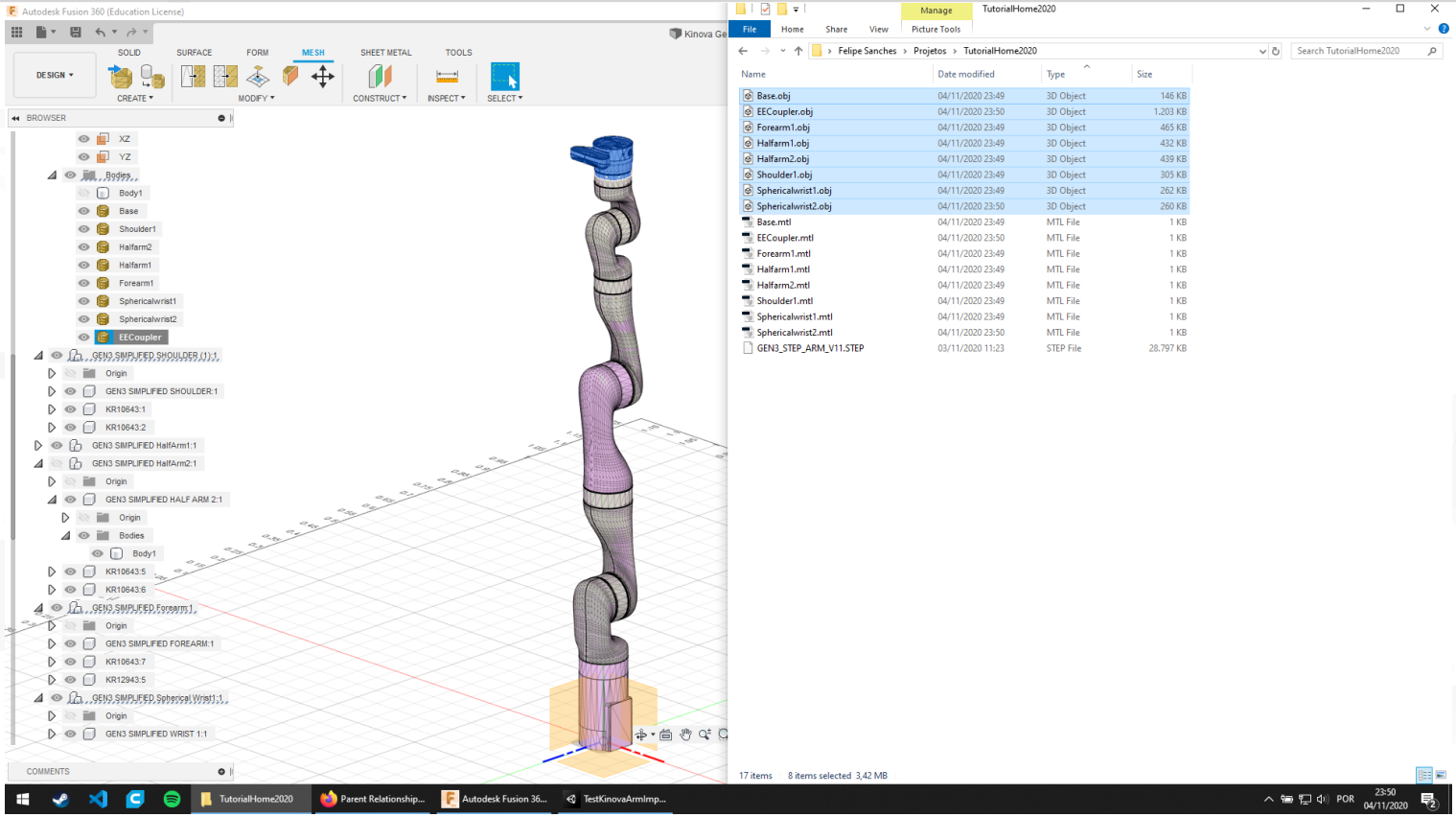
Fusion 360 - Meshes



Fusion 360 - Meshes

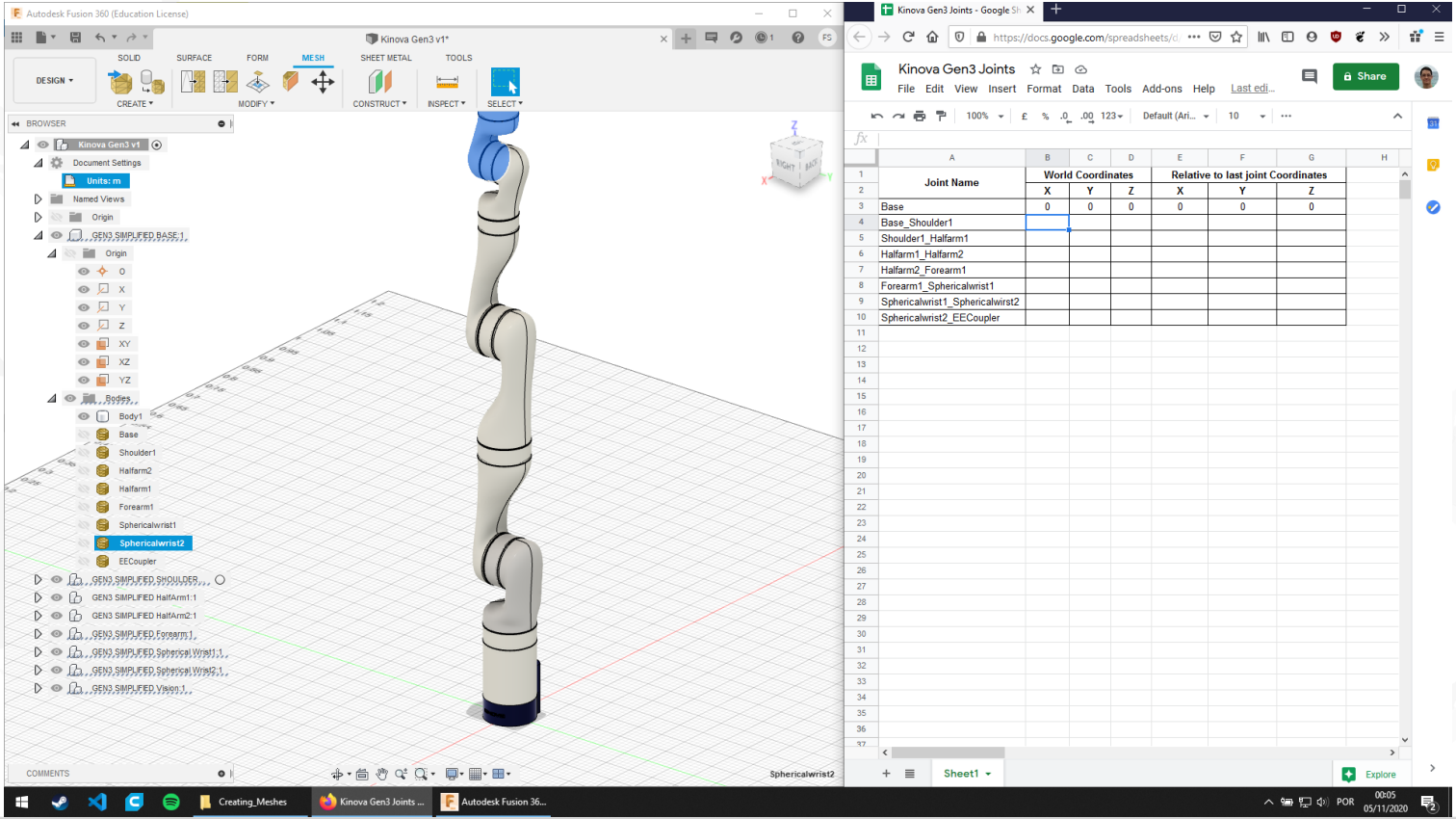


Fusion 360 - Meshes

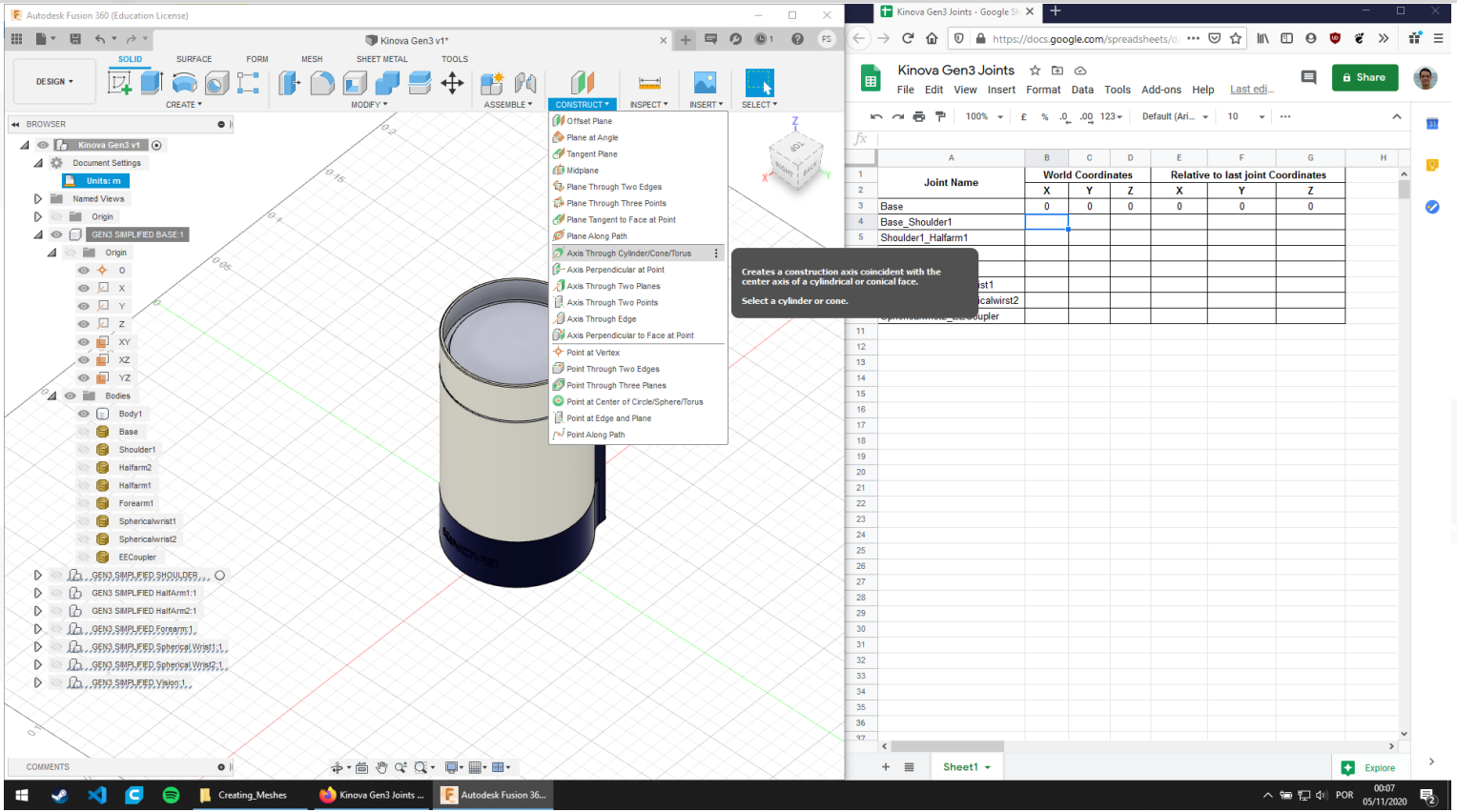


Extraindo info. das Juntas

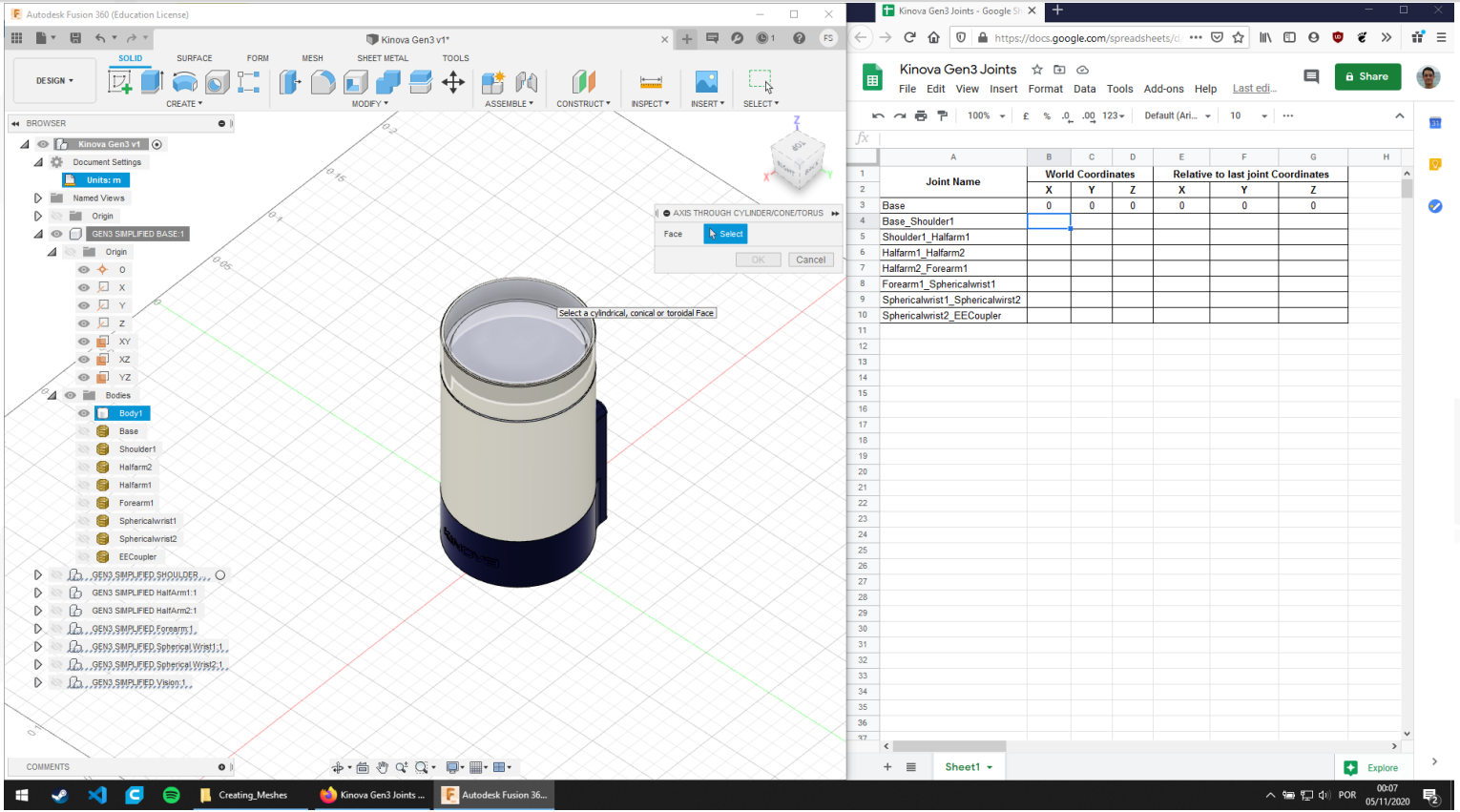
Fusion 360 - Posição das juntas



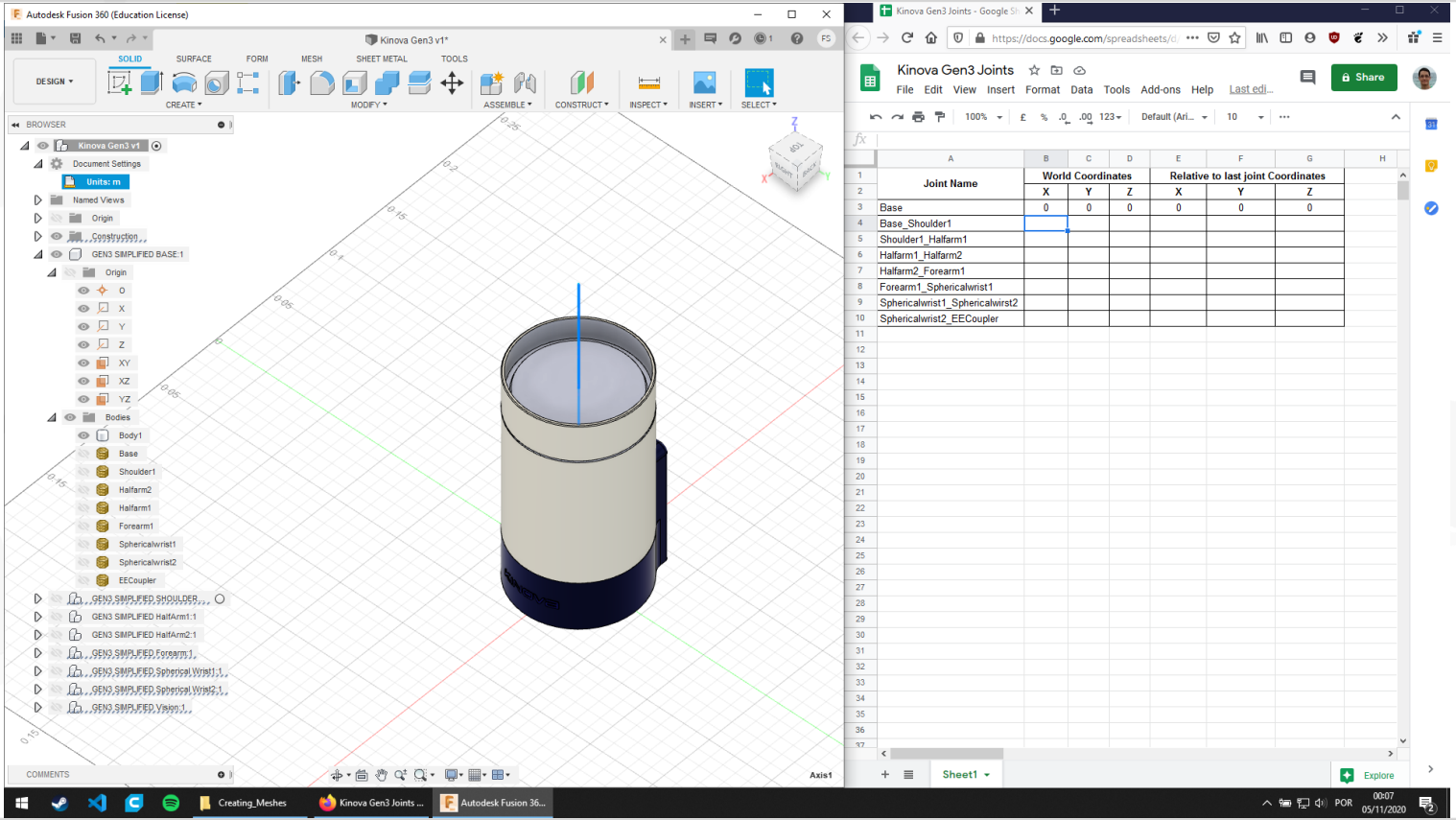
Fusion 360 - Posição das juntas



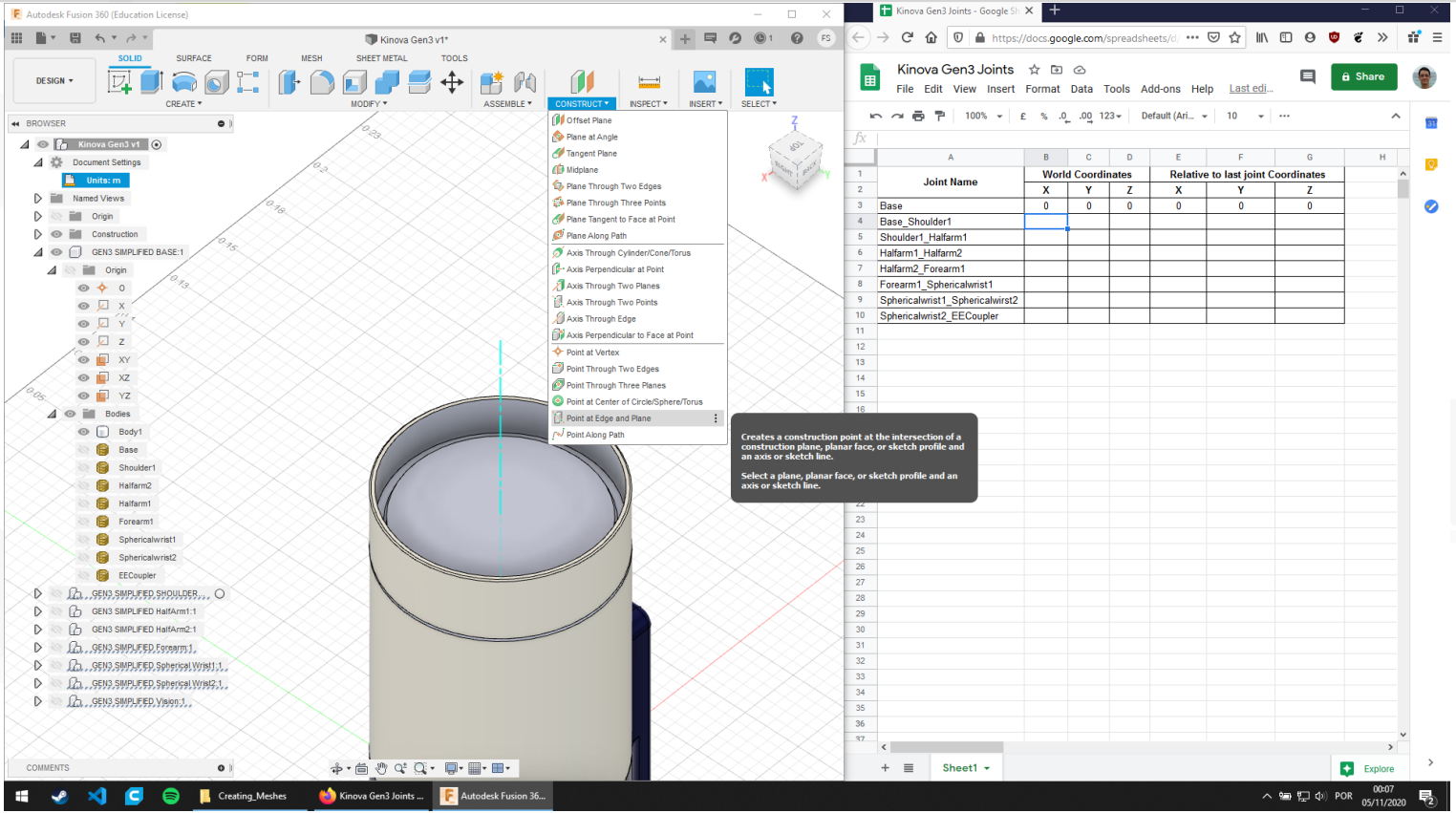
Fusion 360 - Posição das juntas



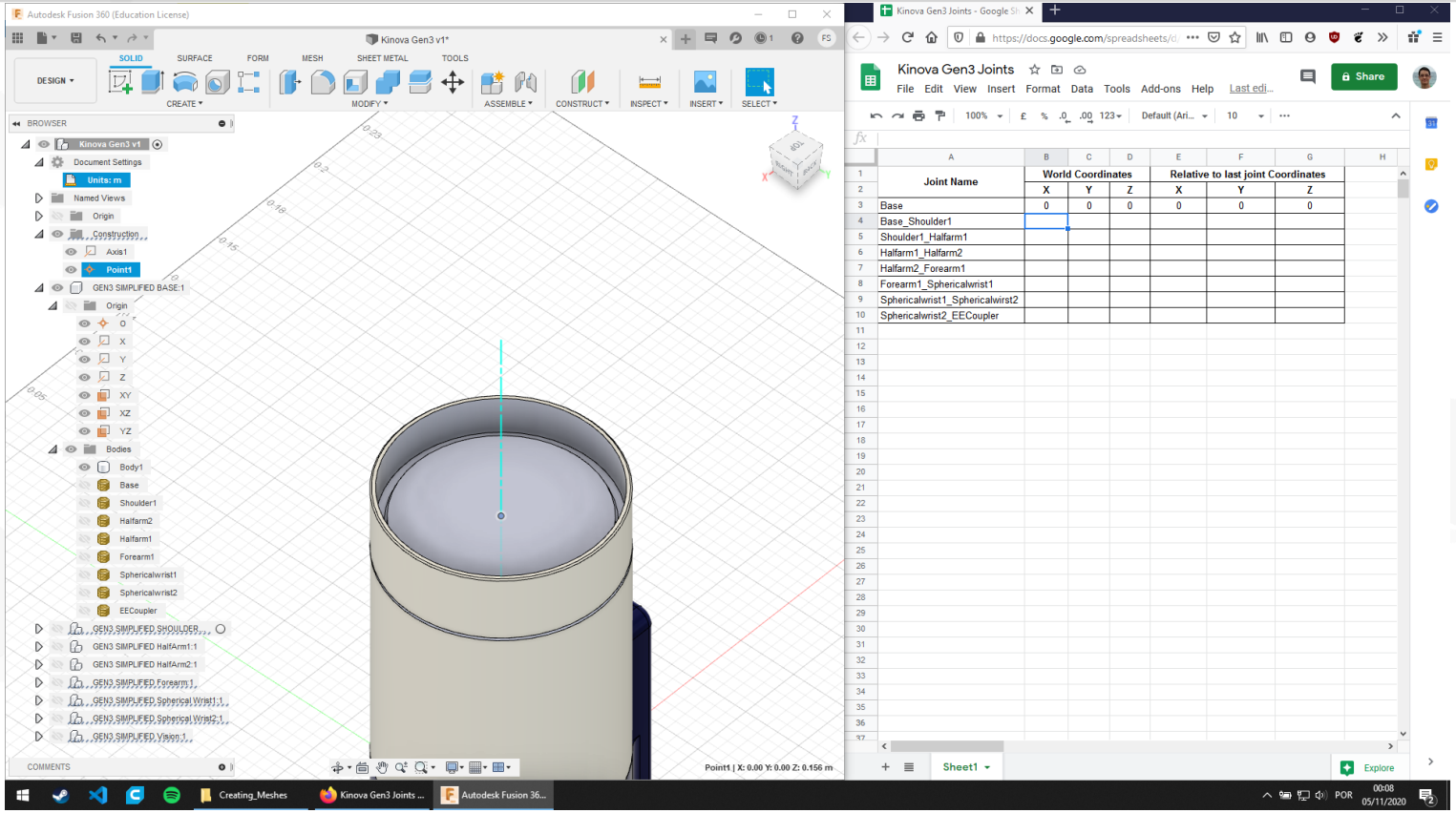
Fusion 360 - Posição das juntas



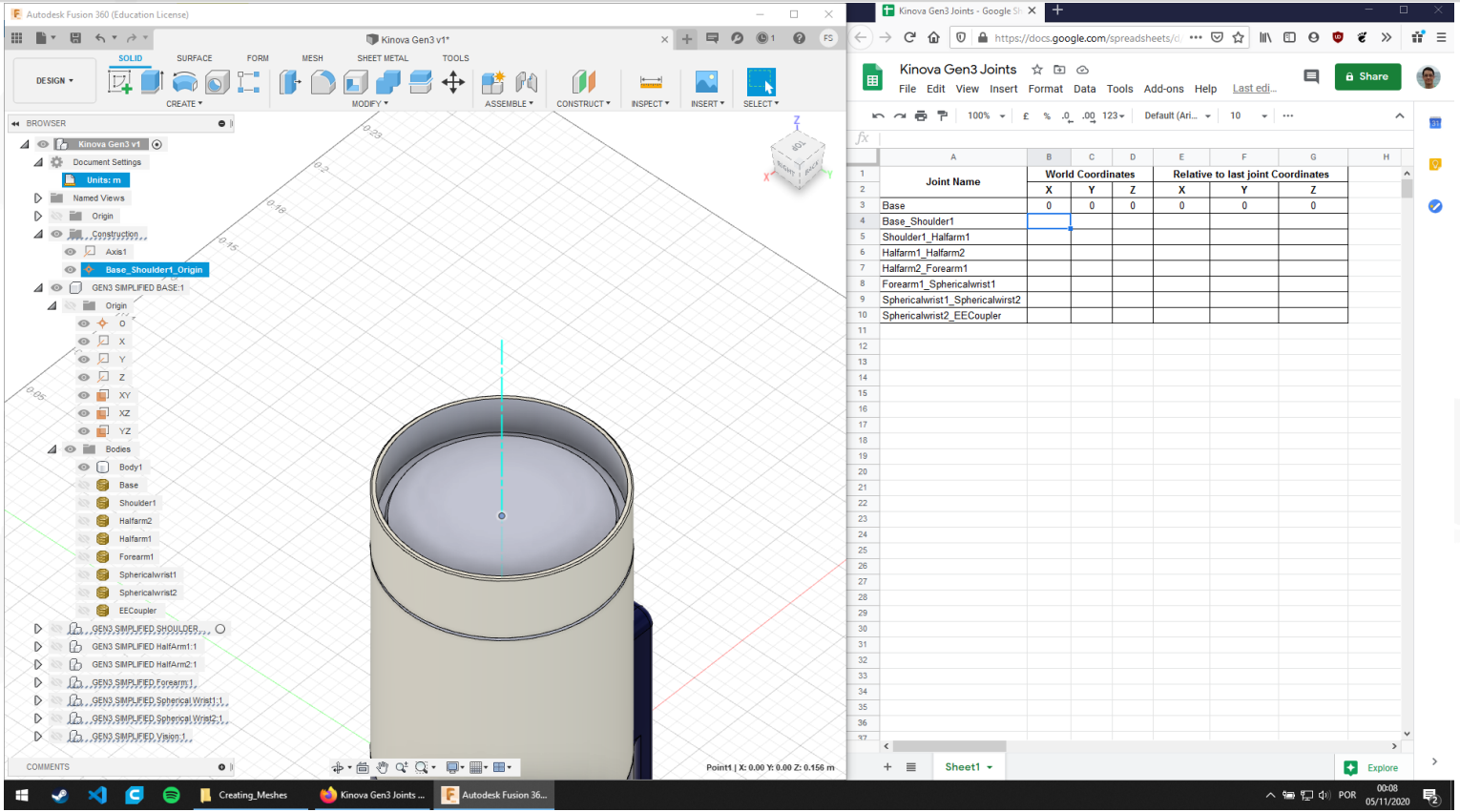
Fusion 360 - Posição das juntas



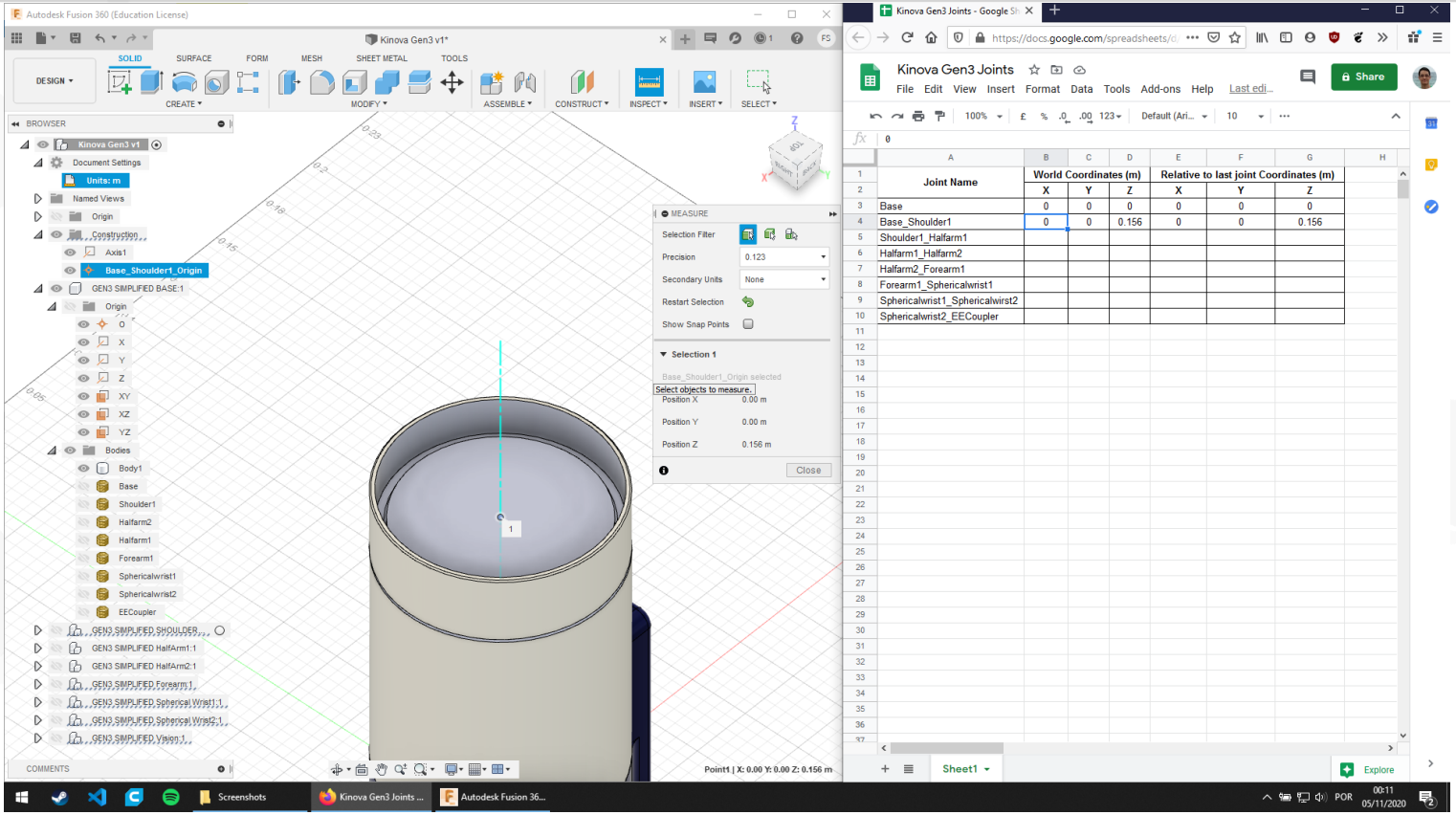
Fusion 360 - Posição das juntas



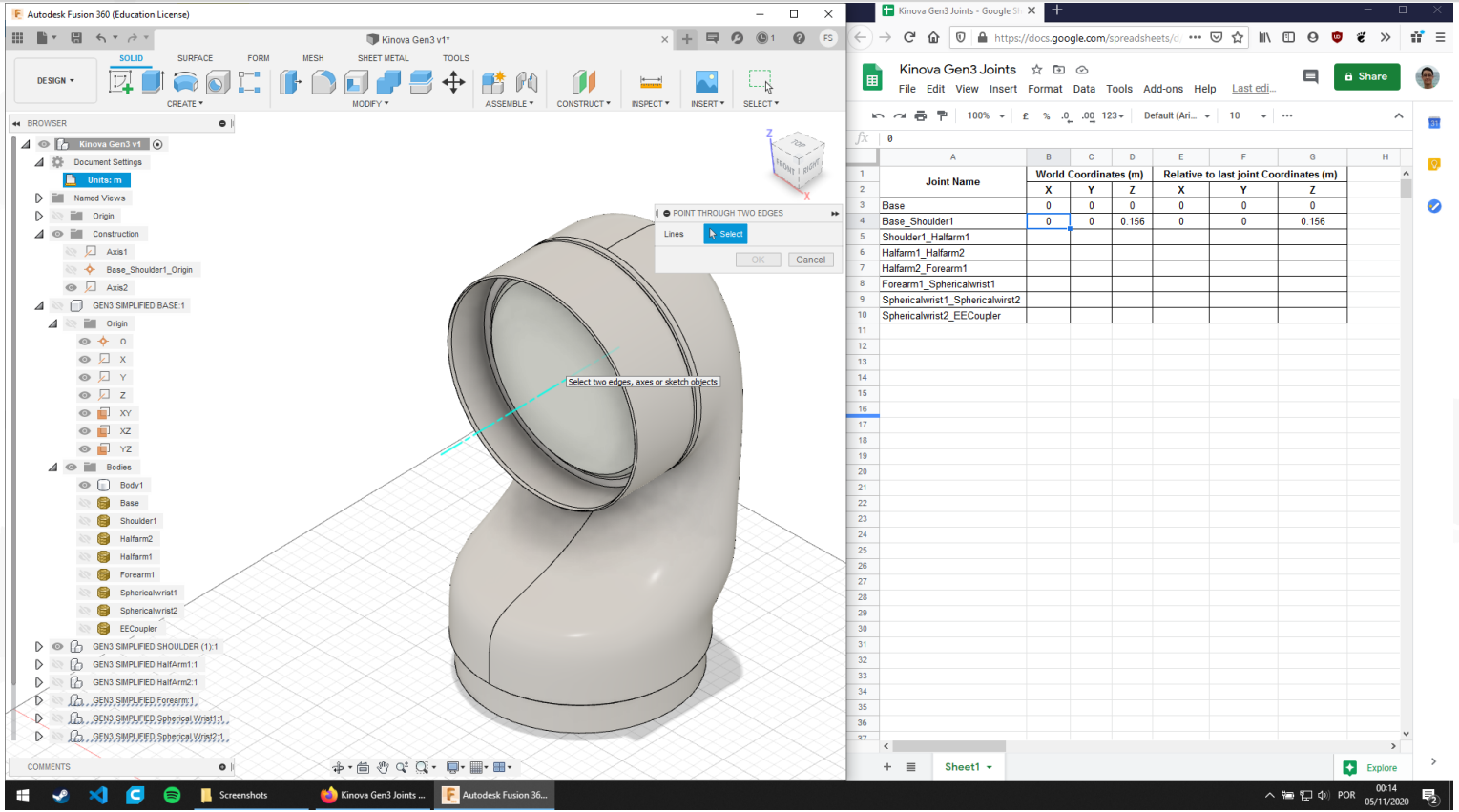
Fusion 360 - Posição das juntas



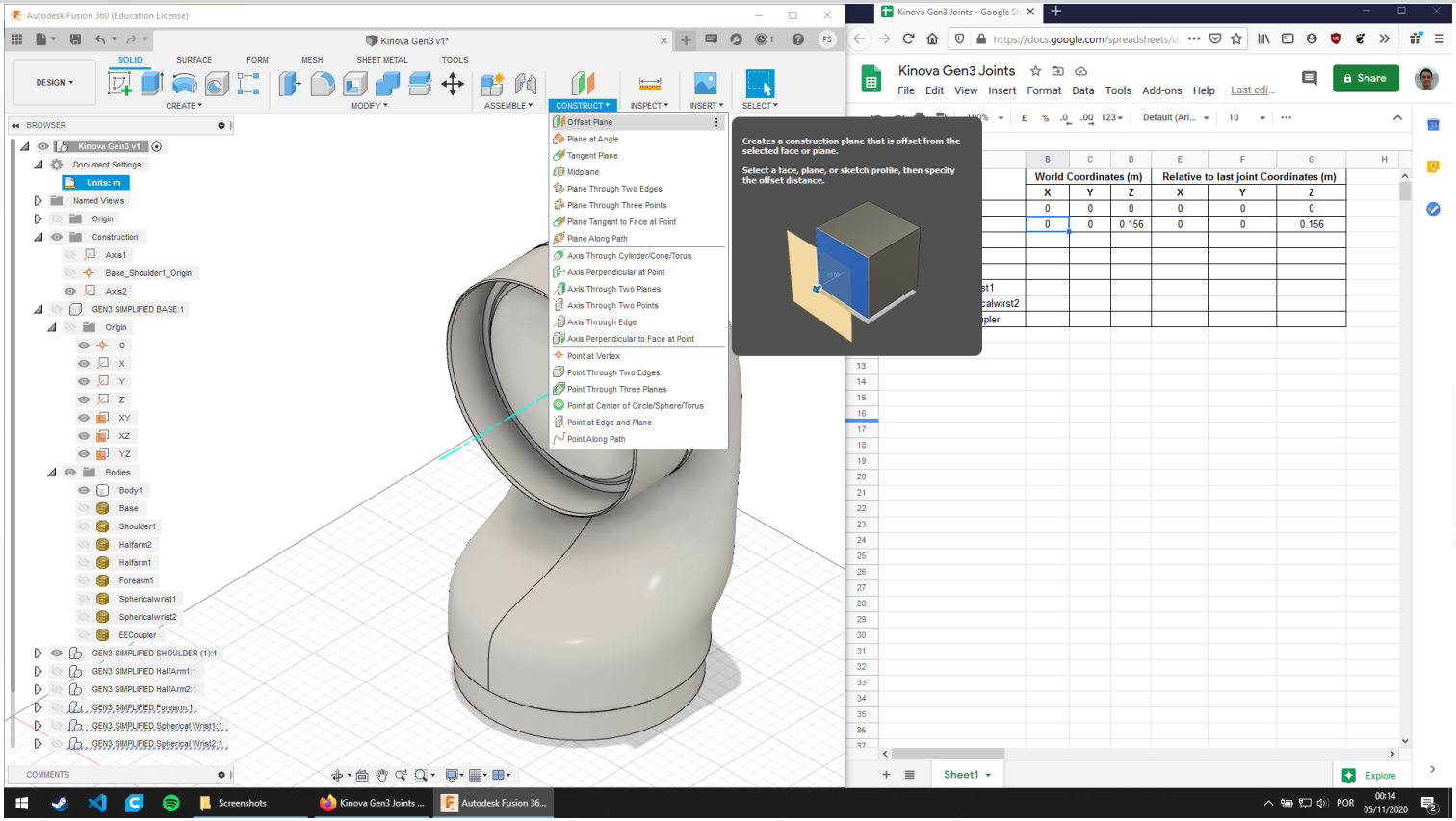
Fusion 360 - Posição das juntas



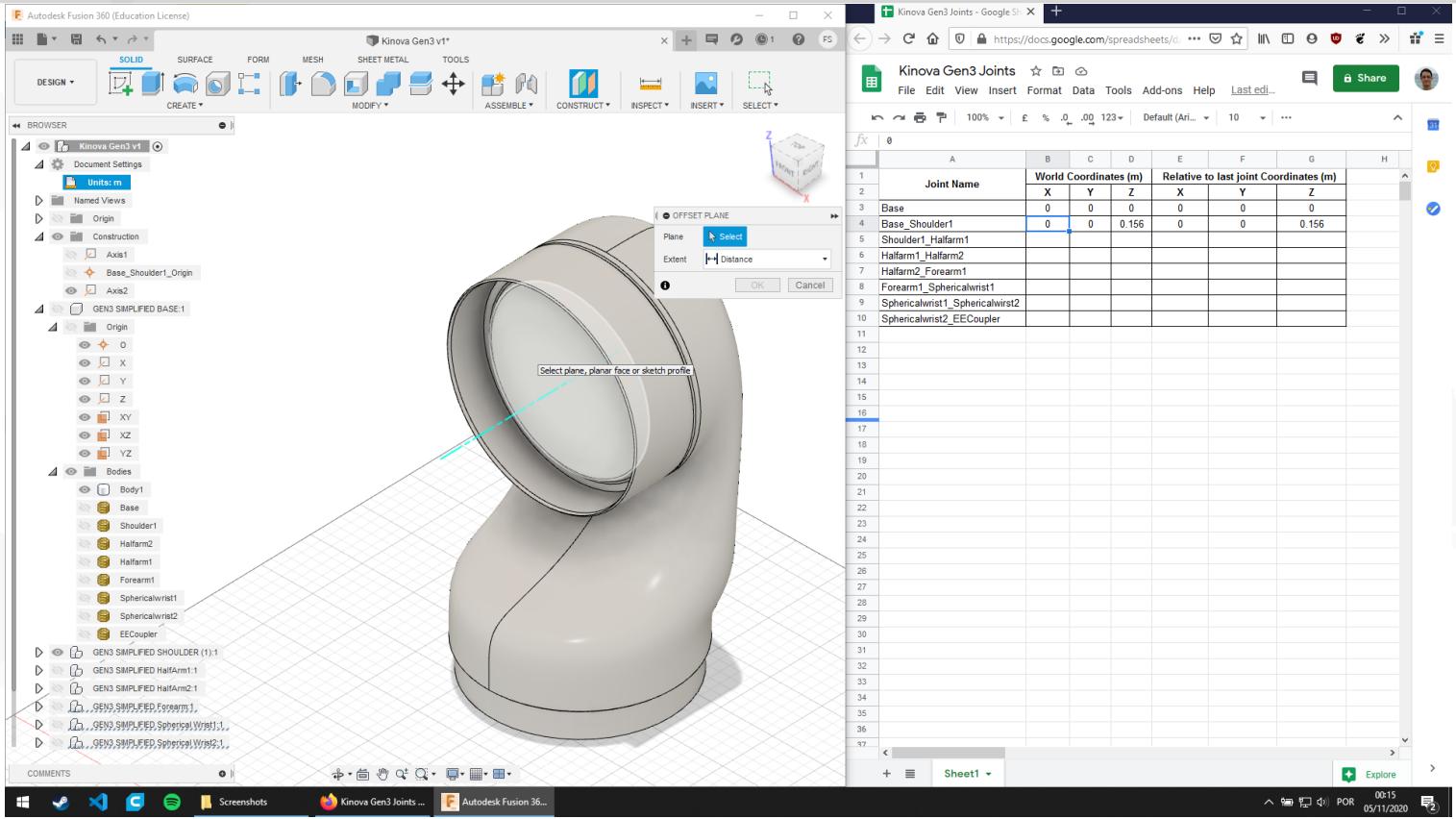
Fusion 360 - Posição das juntas



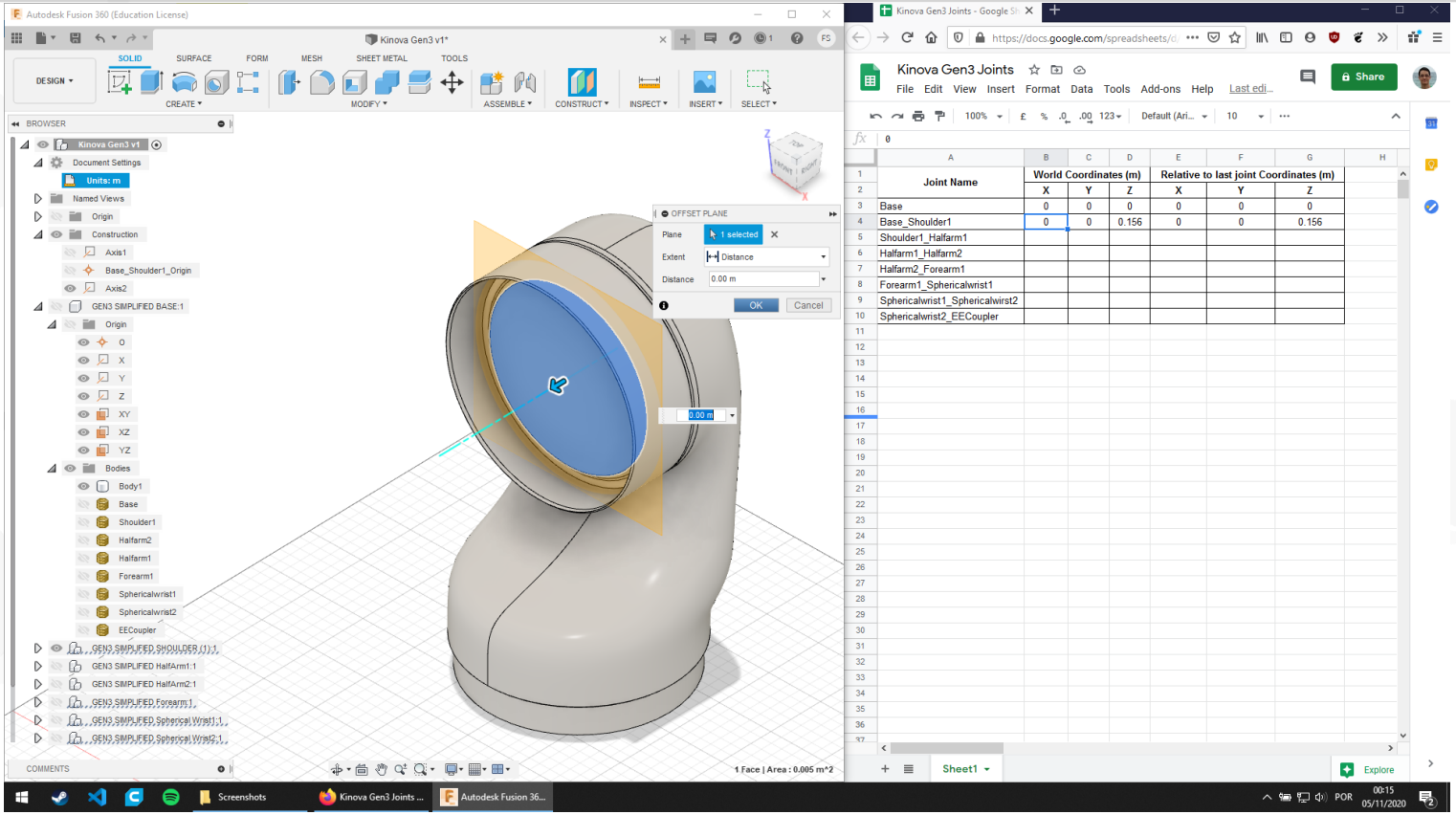
Fusion 360 - Posição das juntas



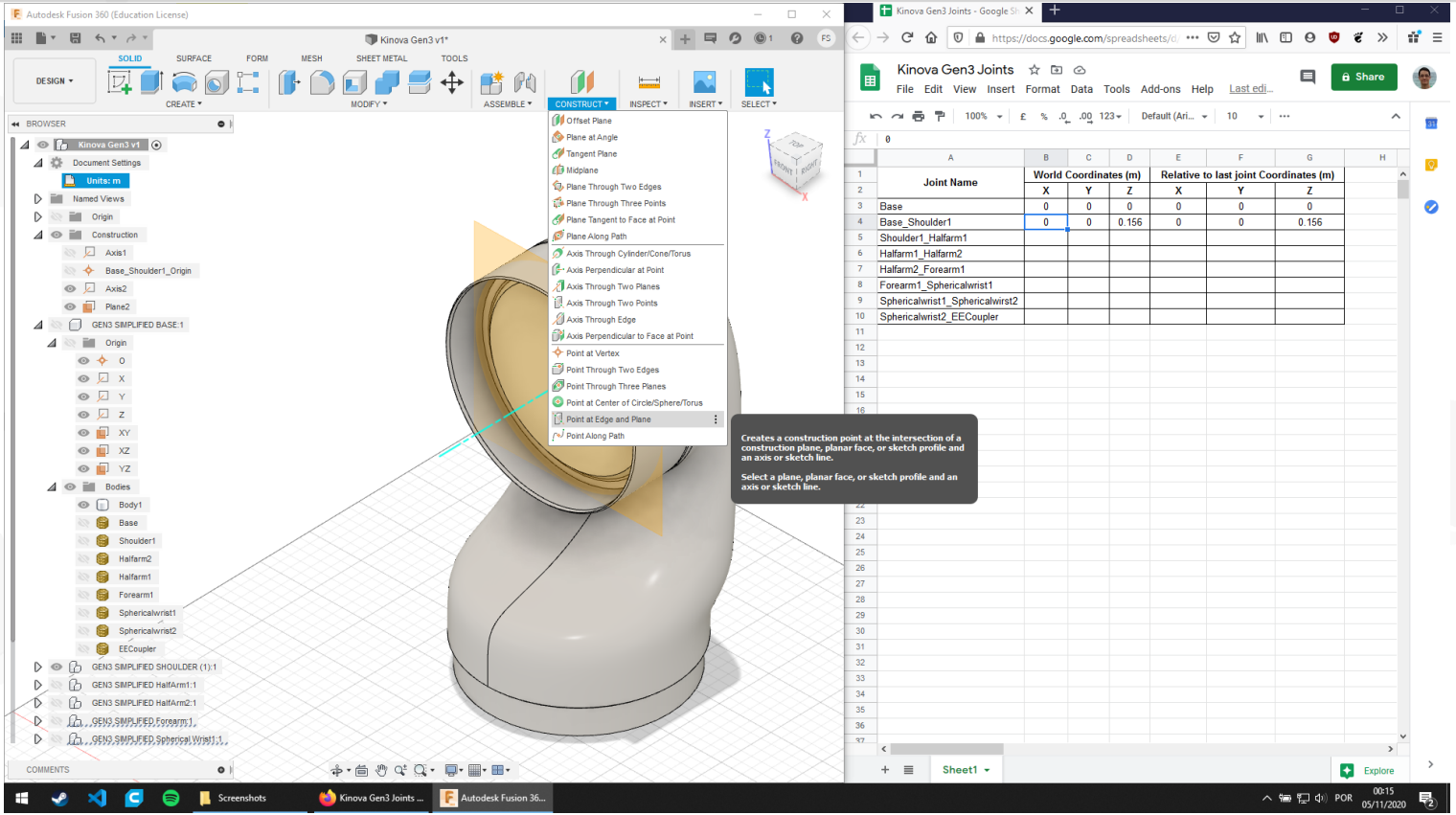
Fusion 360 - Posição das juntas



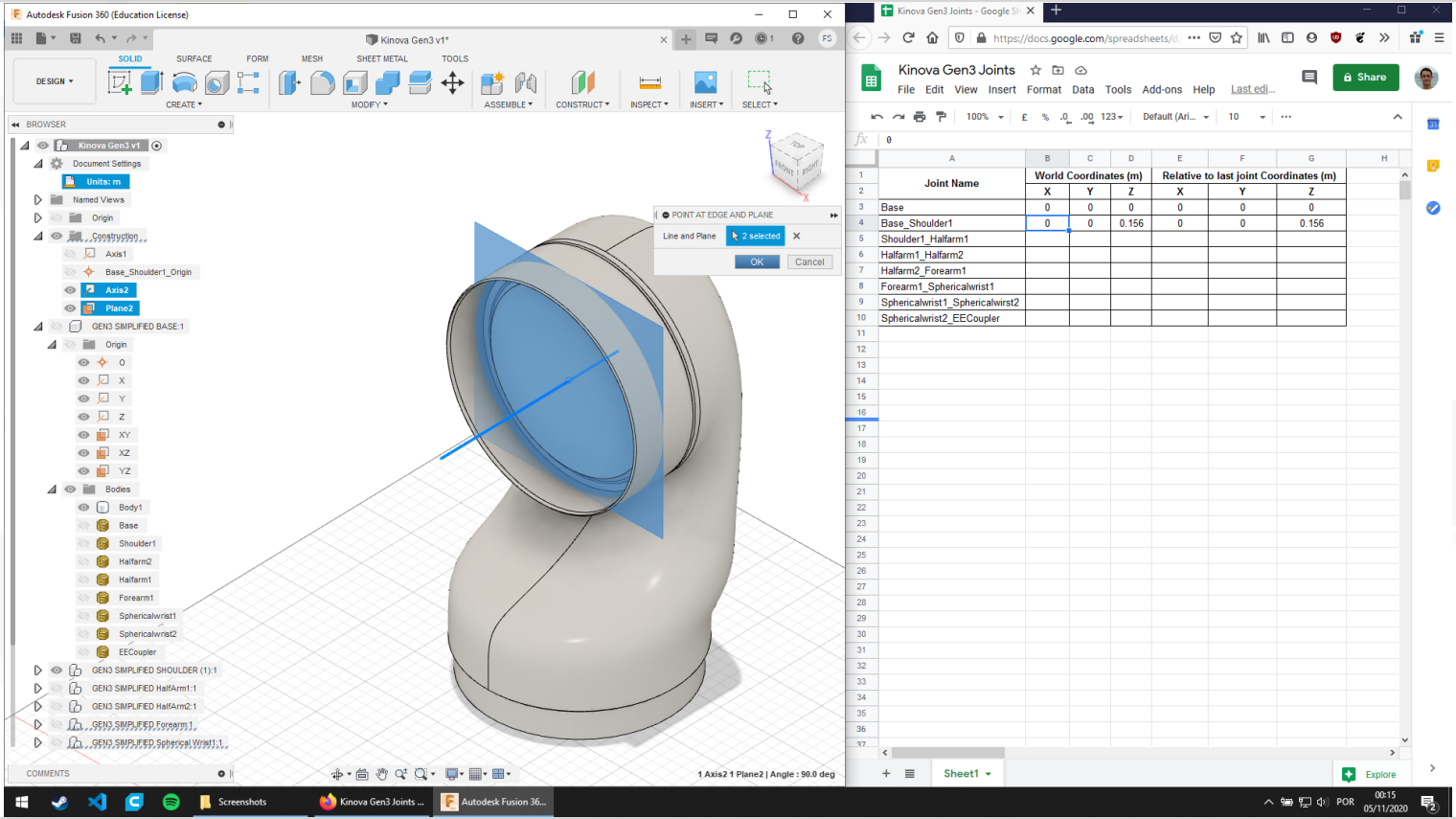
Fusion 360 - Posição das juntas



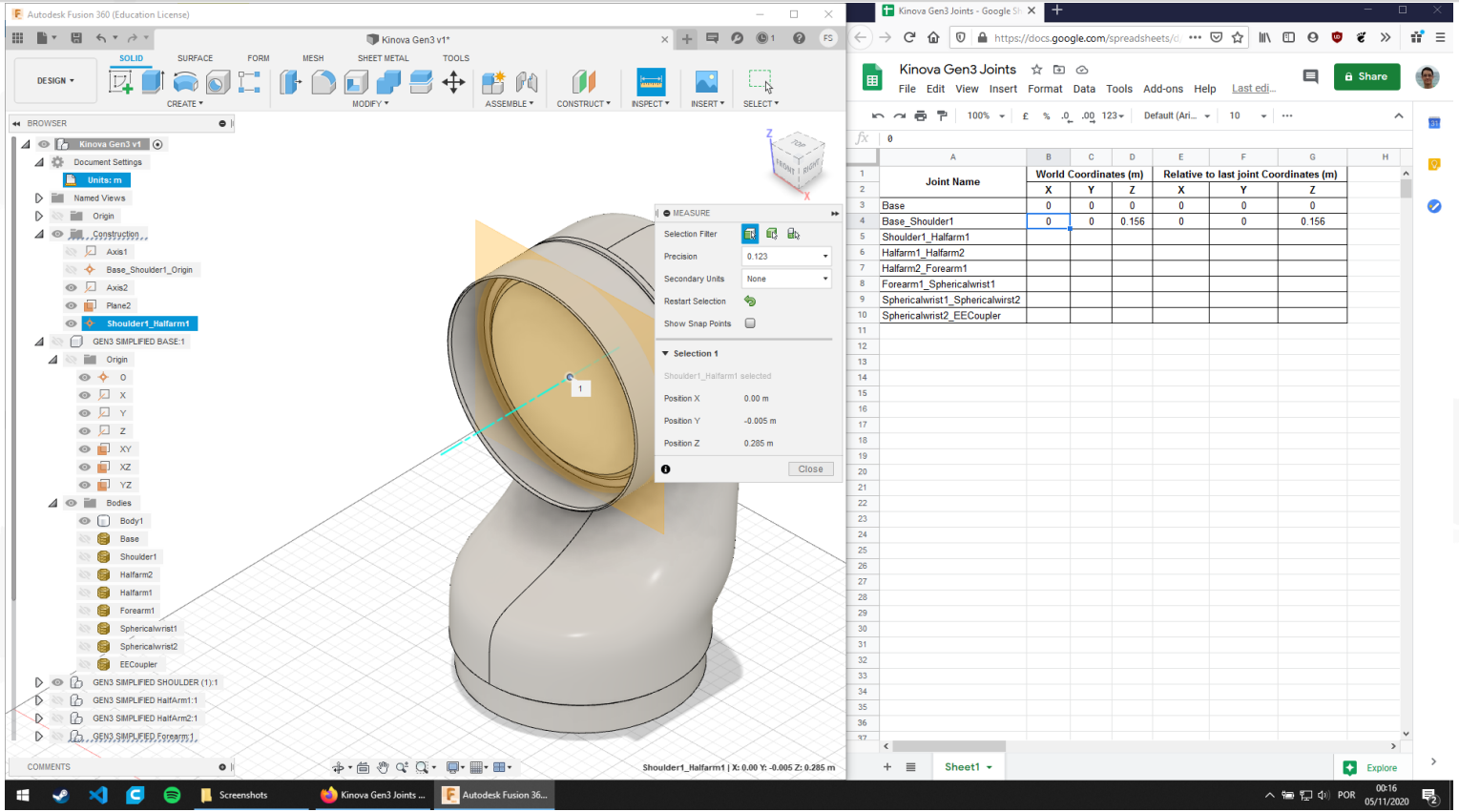
Fusion 360 - Posição das juntas



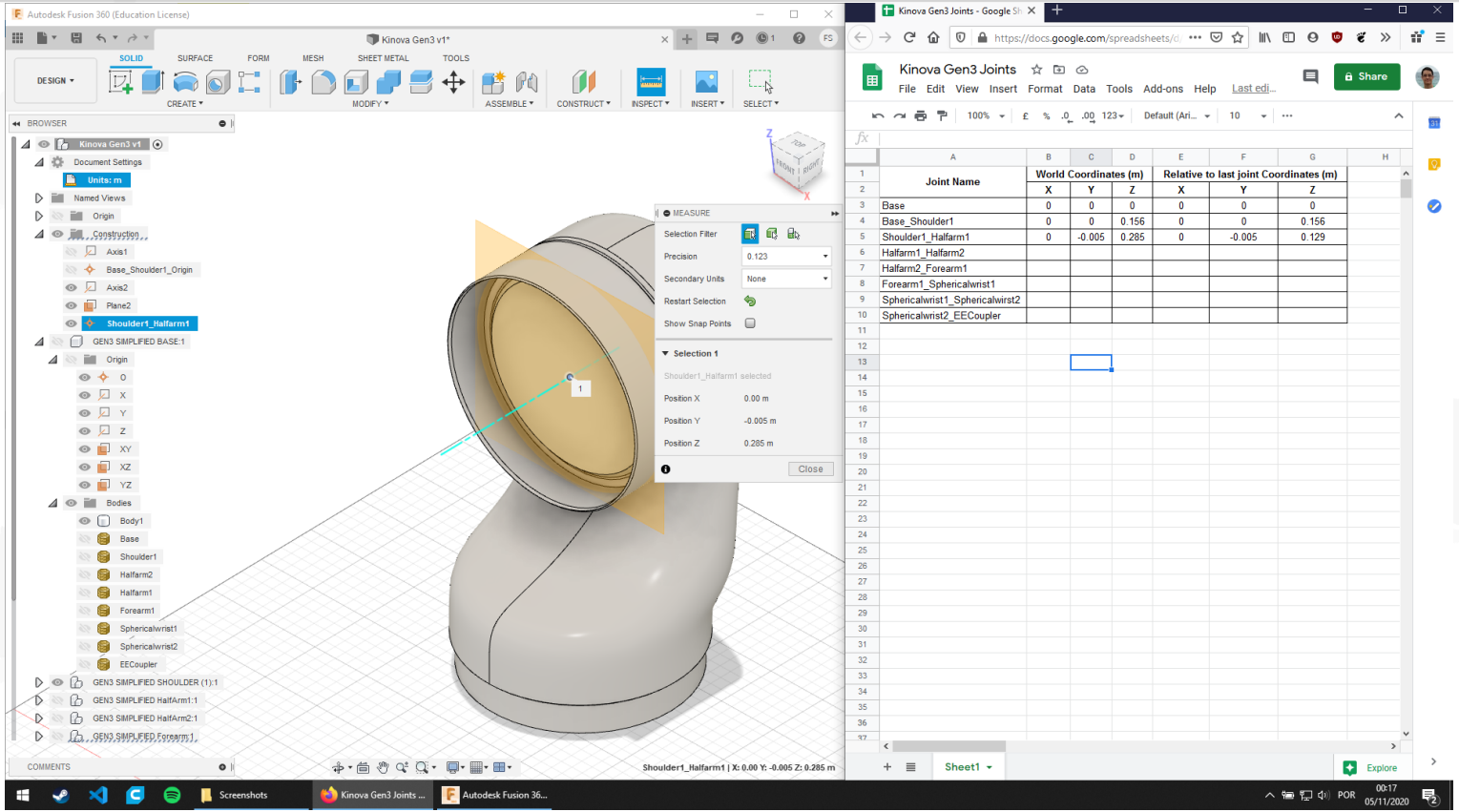
Fusion 360 - Posição das juntas



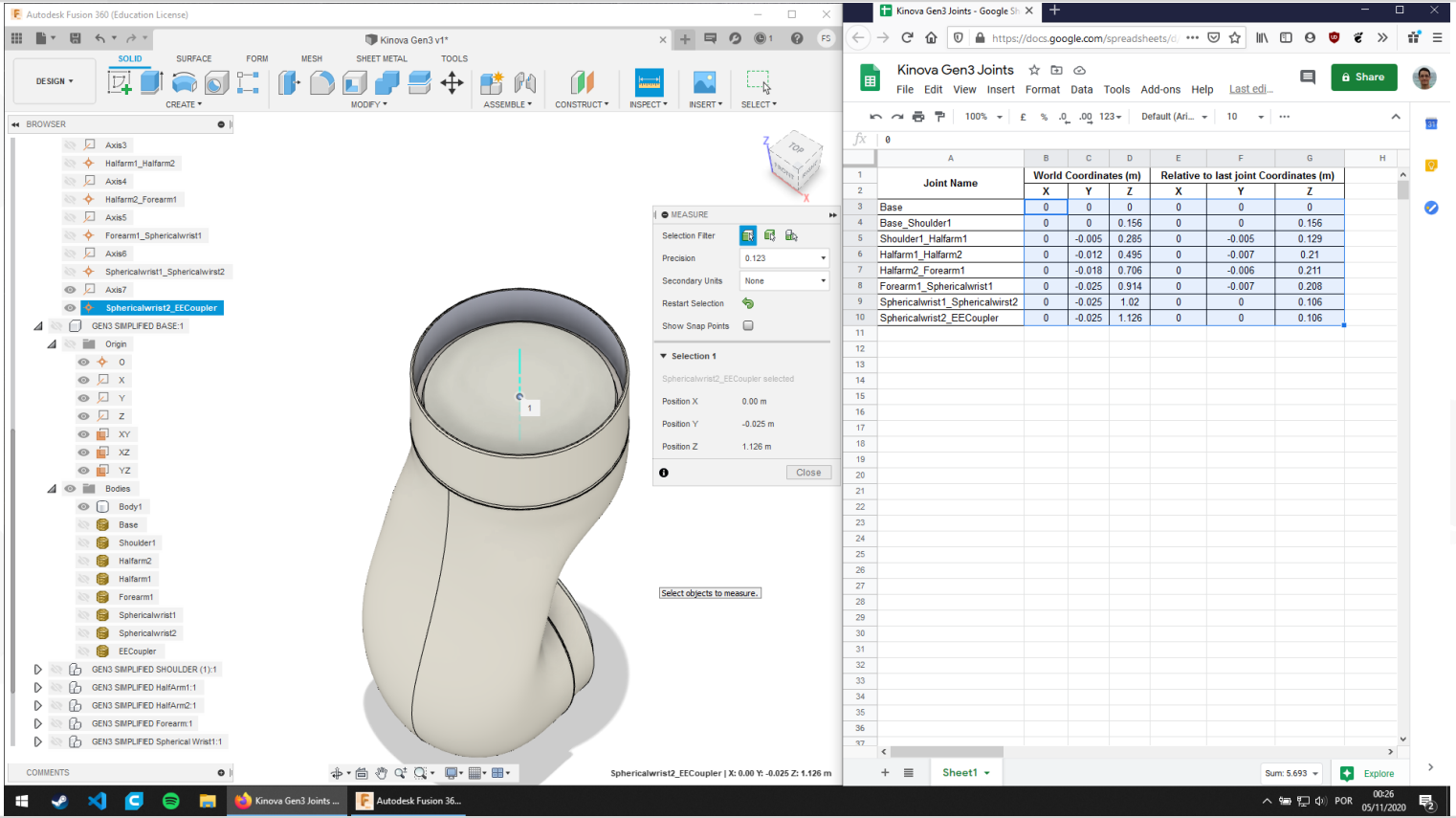
Fusion 360 - Posição das juntas



Fusion 360 - Posição das juntas



Fusion 360 - Posição das juntas



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

MoveIt – 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 (pi)

MoveIt – Xacro/URDF

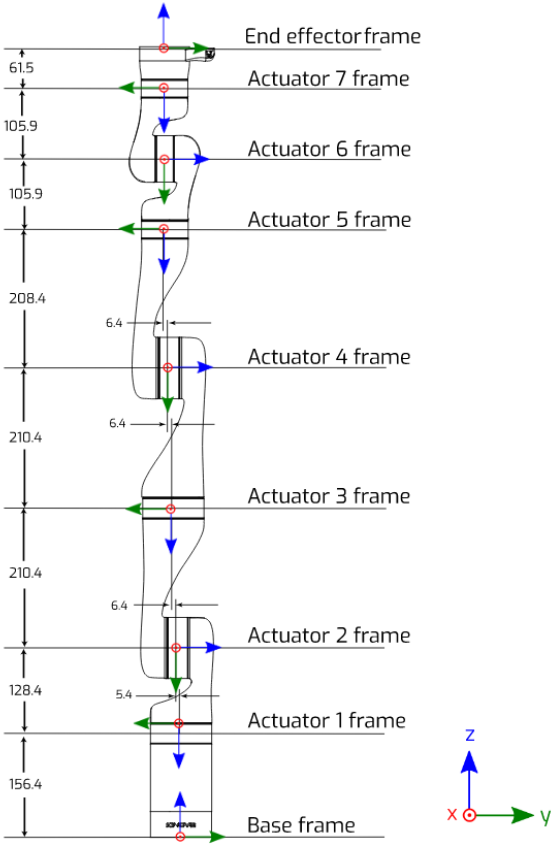


Table 10: Joint limits - 7 DoF spherical wrist

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

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"

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"

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"~~

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

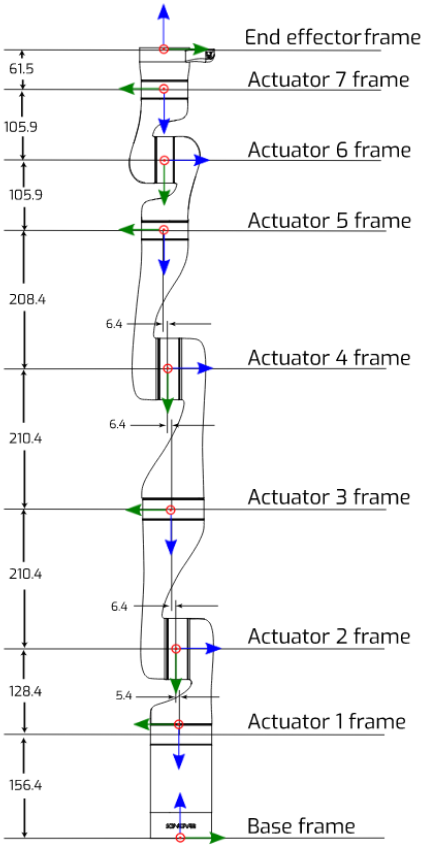


Table 10: Joint limits - 7 DoF spherical wrist

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

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 42: Link 3

Physical quantity	Value
mass (kg)	1.163667 kg

Table 43: Link 4

Physical quantity	Value
mass (kg)	0.930287

Table 44: Link 5

Physical quantity	Value
mass (kg)	0.678106

Table 45: Link 6

Physical quantity	Value
mass (kg)	0.678106

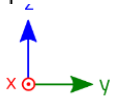


Table 47: Interface & Vision Module

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