Trabalho Prático 1 – Ferramentas e Transformações

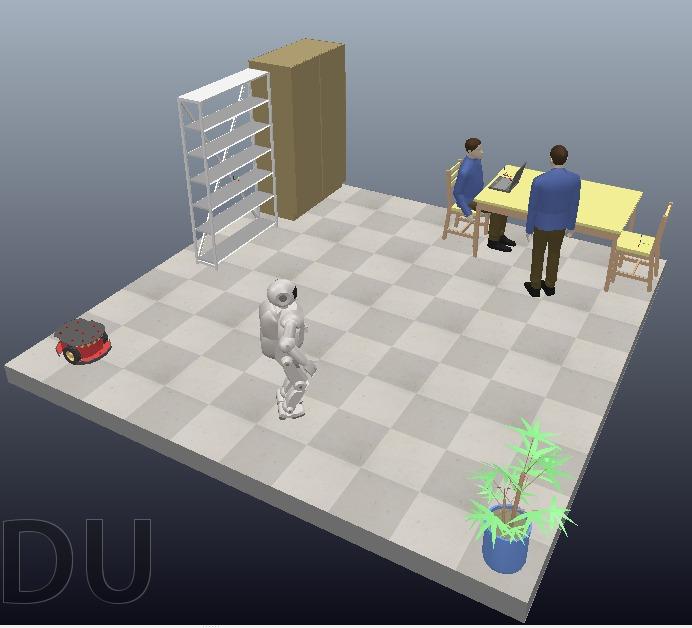
Davi Fraga -2020420575

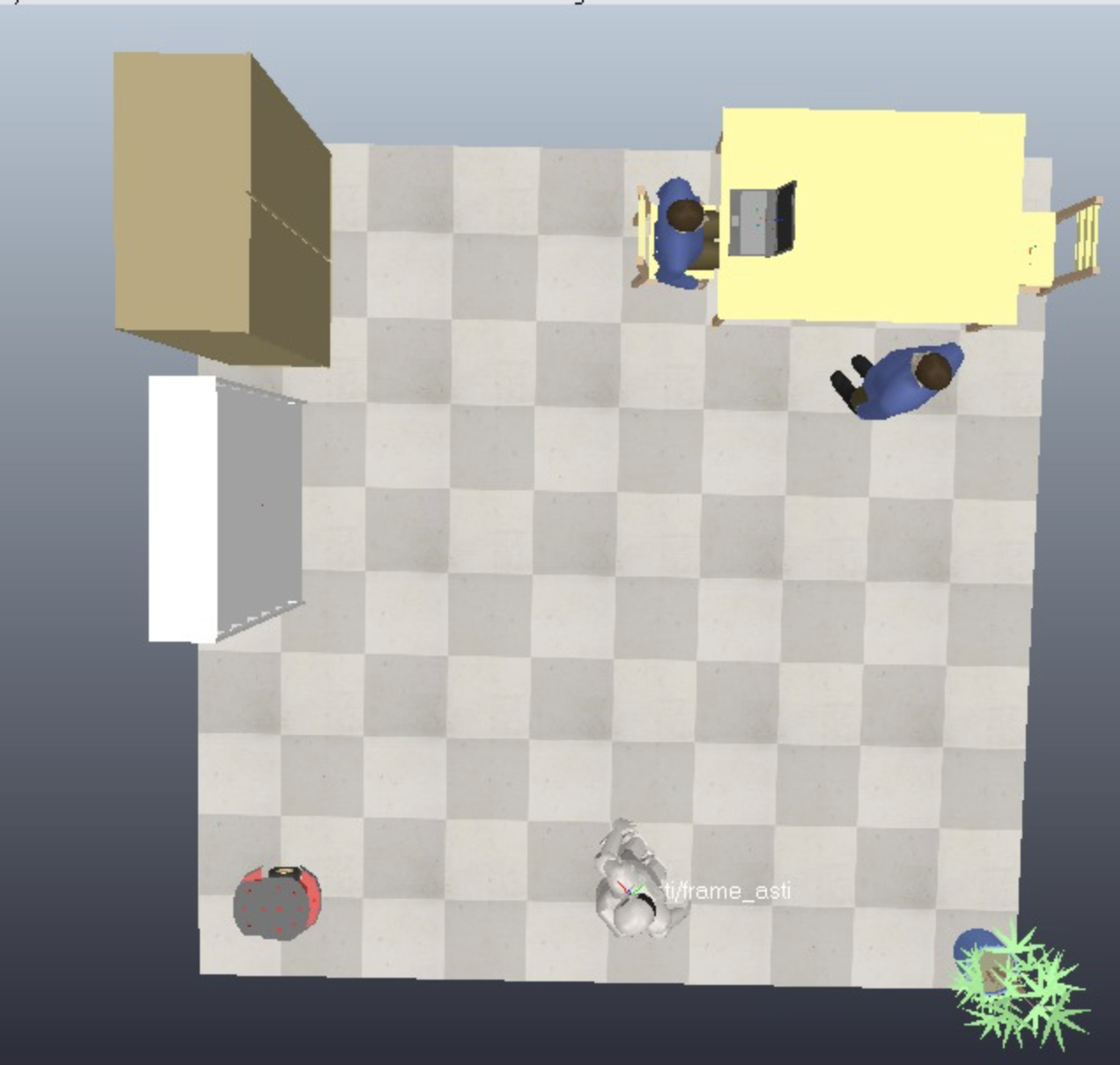
Alexis Mariz -

10 de setembro de 2025

## criar a cena com o robo pionner\_3dx e com 5 objetos

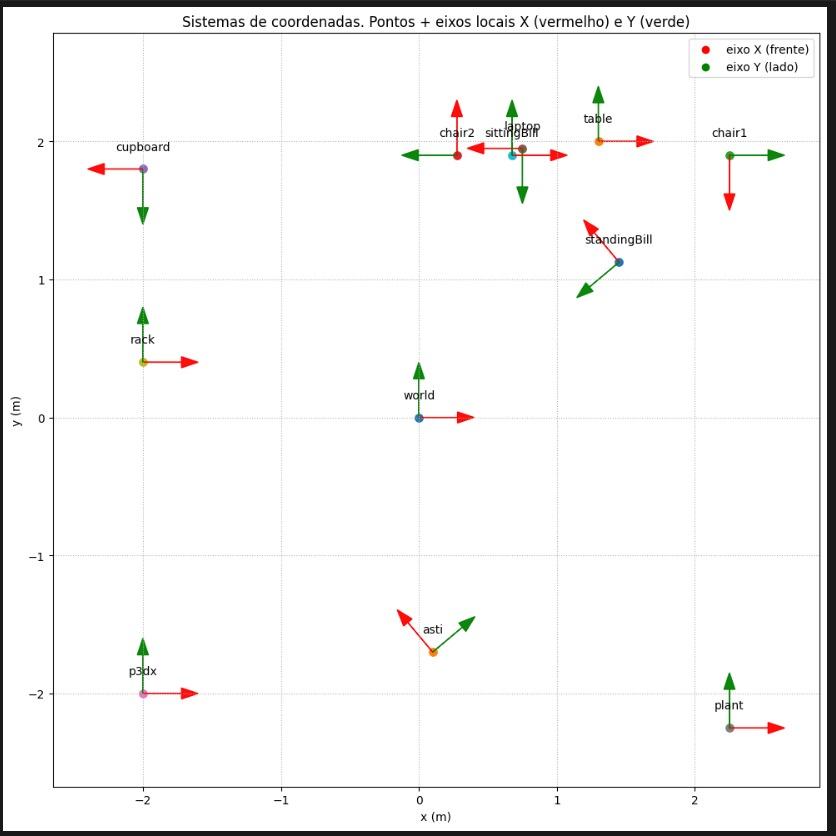
abaixo, a cena criada com o script em python que inicia a cena com os objetos via api



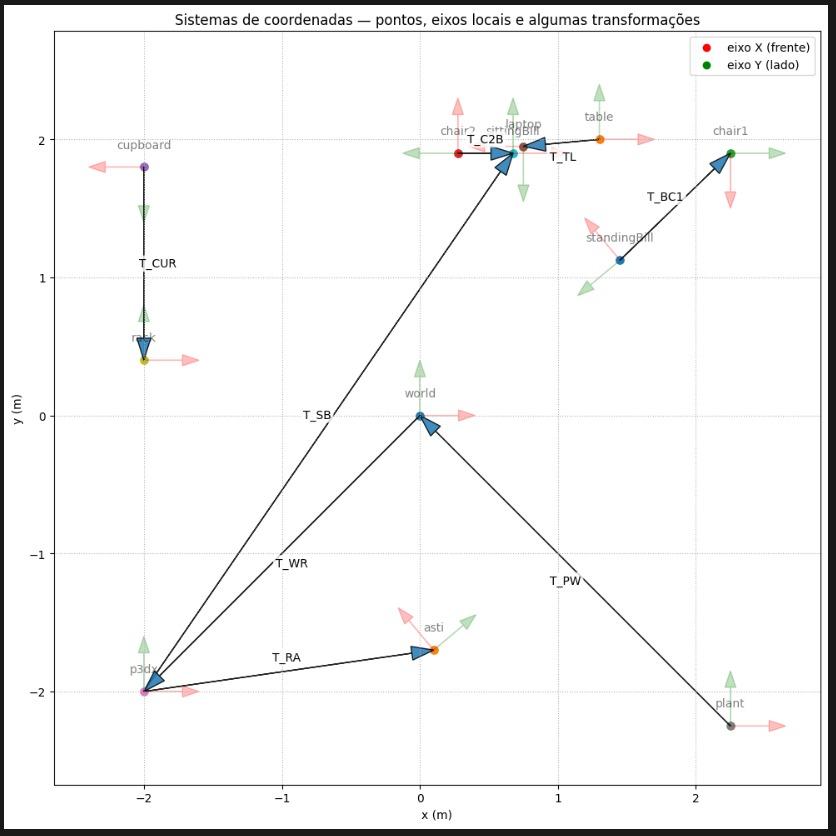


## Criar o sistema de coordenadas para todos os itens na cena e criar um diagrama

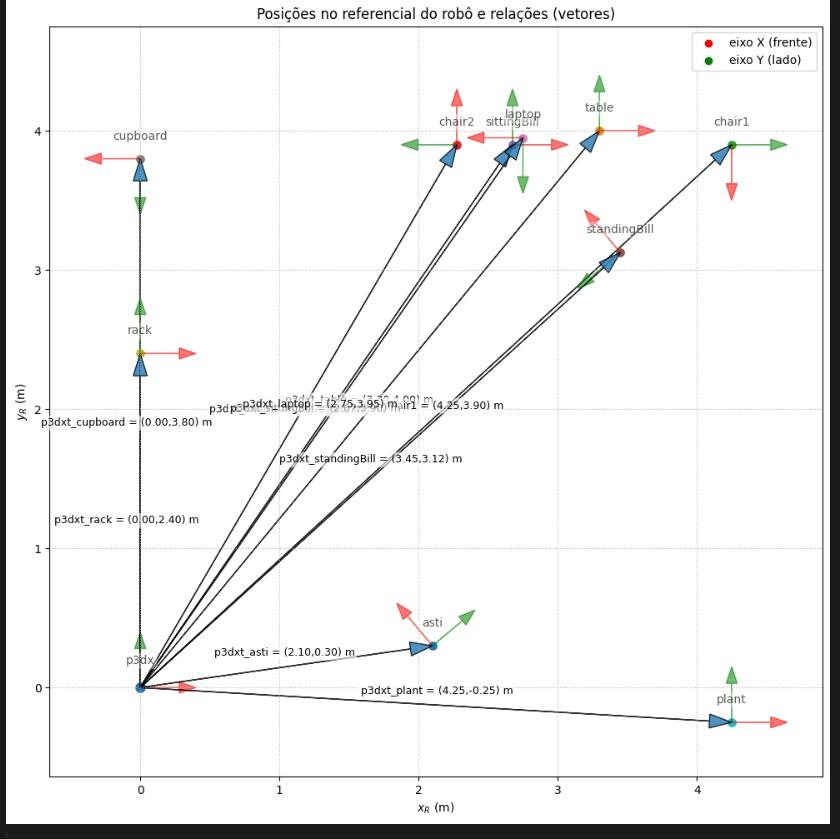
coordenadas de cada elemento com o eixo x e y indicando a direção de cada um



posições relativas dos objetos em relação ao world e ao robô principal



## matrizes de transformação homogêneas que representam as posições de todos os outros elementos da cena no referencial local do robô



as matrizes de transformacao de cada objeto da cena para o robo sao:

Transformação robô-table:

[ 0.0000 0.0000 -1.0000 | 3.3000 ]

[ 0.0000 1.0000 0.0000 | 4.0000 ]

[ 1.0000 0.0000 0.0000 | 0.2362 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-chair1:

[ 0.0000 1.0000 0.0000 | 4.2500 ]

[ -1.0000 0.0000 0.0000 | 3.9000 ]

[ 0.0000 0.0000 1.0000 | 0.3112 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-chair2:

[ 0.0000 -1.0000 0.0000 | 2.2750 ]

[ 1.0000 0.0000 0.0000 | 3.9000 ]

[ 0.0000 0.0000 1.0000 | 0.3112 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-sittingBill:

[ 1.0000 0.0000 0.0000 | 2.6750 ]

[ 0.0000 1.0000 0.0000 | 3.9000 ]

[ 0.0000 0.0000 1.0000 | 0.7612 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-standingBill:

[ -0.6428 -0.7660 0.0000 | 3.4500 ]

[ 0.7660 -0.6428 0.0000 | 3.1250 ]

[ 0.0000 0.0000 1.0000 | 0.8612 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-laptop:

[ -0.6428 0.0000 0.7660 | 2.7500 ]

[ -0.0000 -1.0000 0.0000 | 3.9500 ]

[ 0.7660 -0.0000 0.6428 | 0.6886 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-cupboard:

[ -1.0000 0.0000 0.0000 | 0.0000 ]

[ -0.0000 -1.0000 0.0000 | 3.8000 ]

[ 0.0000 0.0000 1.0000 | 0.8612 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-rack:

[ 1.0000 0.0000 0.0000 | 0.0000 ]

[ 0.0000 1.0000 0.0000 | 2.4000 ]

[ 0.0000 0.0000 1.0000 | 0.7112 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-plant:

[ 1.0000 0.0000 0.0000 | 4.2500 ]

[ 0.0000 1.0000 0.0000 | -0.2500 ]

[ 0.0000 0.0000 1.0000 | 0.0262 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

Transformação robô-asti:

[ -0.6428 0.7660 -0.0000 | 2.1000 ]

[ 0.7660 0.6428 -0.0000 | 0.3000 ]

[ -0.0000 -0.0000 -1.0000 | 0.6672 ]

[ 0.0000 0.0000 0.0000 | 1.0000 ]

## trocando o robô de posições e refazendo os plots com as transformações

## 

## trocando o robô de posições e refazendo os plots com as transformações

## conclusão e desafios

Ao final desse trabalho já temos uma familiaridade maior com o coppelia Slim e com os comandos via api para criar uma cena com um robô e captar sua interação com o mundo simulado no coppelia. A partir daqui já sabemos usar os comandos básicos e podemos partir para tarefas mais complexas a serem realizadas.  
Durante o processo de criar a cena encontramos alguns desafios que valem a pena serem ressaltados:

Primeiro: achar a referência dos arquivos de cada modelo nos arquivos do coppelia para chamar a api. Tivemos que recorrer a documentação para entender quais parâmetros usar da maneira correta para colocar os objetos de maneira adequada na cena.

Segundo: Definir as matrizes de transformação ficou um pouco confuso mas com a referencia do material de apoio <https://github.com/verlab/dcc042-robotica-movel/blob/main/jupyter-notebooks/aula05-transformacoes-homogeneas-espaco-configuracoes.ipynb>

ficou bem mais tranquilo.

Terceiro: Para usar os lasers não tinham muitas referências fora as aulas, então sem o conteúdo de apoio seria bem mais complexo de realizar esse trabalho mas não temos certeza se os plots estão de acordo com o que é lido do laser e na questão 6 não conseguimos pois dependia das lasers da 5. conseguimos fazer o robo andar mas os plots tambem nao funcionaram.