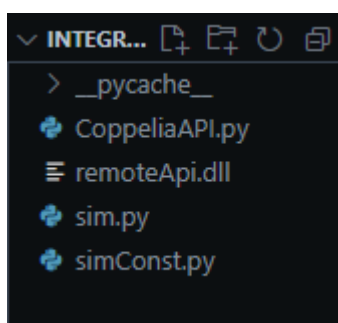


Configuração e utilização da API Remota em Python

- Passo 1: Organização de diretório

Localize os arquivos “sim.py”, “simConst.py” e “remoteApi.dll” localizados na pasta de instalação do coppelia, copie-os e cole-os em outra pasta.

Nesta mesma pasta, crie seu arquivo .py onde vamos escrever o código de conexão e utilização da API



- Passo 2: Métodos de conexão e controle de simulação

Abra o editor de texto da sua escolha, aqui estou usando o VsCode, importe os métodos do arquivo sim e a biblioteca time (será usada para fazer algumas manipulações em relação ao tempo) faça um método chamado connect(), para se conectar com o coppelia e obter o clientID referente ao seu computador e seu coppelia

```
import sim
import time

#Conectando com o coppelia
def connect():
    sim.simxFinish(-1) # Fecha todas as conexões existentes
    clientID = sim.simxStart('192.168.1.102', 19997,
                             waitUntilConnected=True, # Aguardar conexão
                             doNotReconnectOnceDisconnected=False, # Reconectar se desconectado
                             timeOutInMs=5000, # Tempo limite em milissegundos
                             commThreadCycleInMs=5) # Ciclo do thread de comunicação em milissegundos

    # Verificando se a conexão foi bem-sucedida
    if clientID != -1:
        print('Conectado ao Coppeliasim Edu!')
    else:
        print('Falha ao conectar-se ao Coppeliasim Edu!')
    return clientID
```

também o método disconnect(), será utilizado quando quisermos encerrar a conexão

```
def disconnect(clientID):
    sim.simxFinish(clientID)
    print('Desconectado do CoppeliaSim')
```

Após isso, para iniciar e interromper uma simulação, criaremos mais dois métodos, `start_simulation()` e `stop_simulation()`

```
def start_simulation(clientID):
    res = sim.simxStartSimulation(clientID, sim.simx_opmode_oneshot_wait)

    if res != sim.simx_return_ok:
        print('Falha ao iniciar a simulação')
    else:
        print('Simulação iniciada com sucesso')

def stop_simulation(clientID):
    res = sim.simxStopSimulation(clientID, sim.simx_opmode_oneshot_wait)

    if res != sim.simx_return_ok:
        print('Falha ao parar a simulação')
    else:
        print('Simulação parada com sucesso')
```

● Passo 3: Métodos de controle do robô

próximo método será para pegar o `handle` do robô, um identificador único que usaremos para controlar o robô

```
def get_robot_handle(clientID):
    # Obtenha o handle do robô
    errorCode, youbotHandle = sim.simxGetObjectHandle(clientID, 'RobotnikSummitXL', sim.simx_opmode_blocking)

    # Verifique se a operação foi bem-sucedida
    if errorCode == sim.simx_return_ok:
        print('Handle do robô obtido com sucesso')
        return youbotHandle
    else:
        print('Falha ao obter o handle do robô')
        return
```