

Trabalho Prático 1: RPC

Disponível no Google Classroom: Trabalho 1 - RPC

Neste trabalho você irá implementar uma chamada remota a procedimento usando a biblioteca RPyC de Python (<https://rpyc.readthedocs.io/en/latest/install.html>).

Para instalar a biblioteca rpyc basta executar o comando:

```
pip install rpyc
```

Este trabalho pode ser realizado em grupo de até DUAS pessoas.

O grupo deve enviar via Classroom (pasta Trabalho : RPC) um **relatório detalhado com as respostas às questões até 23 de setembro às 18:00 horas. Além das respostas, o relatório deve conter a política de divisão de tarefas, a abordagem adotada para fazer a comunicação entre máquinas diferentes, problemas encontrados e como foram solucionados, e o número de vezes em que foi rodado o código para os cálculos dos tempos de execução.**

O modelo que iremos utilizar é orientado a serviços. A seguir apresenta-se um template para um servidor oferecer os serviços de chamada remota de procedimento.

```
import rpyc
class MyService(rpyc.Service):
    def on_connect(self, conn):
        # código que é executado quando uma conexão é iniciada, caso
        seja necessário
        pass
    def on_disconnect(self, conn):
        # código que é executado quando uma conexão é finalizada,
        caso seja necessário
        pass
    def exposed_get_answer(self): # este é um método exposto
        return 42
    exposed_the_real_answer_though = 43 # este é um atributo
    exposto
    def get_question(self): # este método não é exposto
        return "Qual é a cor do cavalo branco de Napoleão?"
#Para iniciar o servidor
if __name__ == "__main__":
    from rpyc.utils.server import ThreadedServer
    t = ThreadedServer(MyService, port=18861)
    t.start()
```

Devem existir os dois métodos para conectar e desconectar uma conexão e os outros métodos podem ser definidos livremente por você. Neste caso, você pode escolher os atributos que vão ser expostos para outros processos: se o nome

começa com exposed, ele poderá ser acessado remotamente, senão ele será acessível somente localmente.

O programa cliente abaixo se conecta com o servidor e executa algumas instruções remotamente.

```
import rpyc
import sys

if len(sys.argv) < 2:
    exit("Usage {} SERVER".format(sys.argv[0]))

server = sys.argv[1]

conn = rpyc.connect(server,18861)

print(c.root)
print(c.root.get_answer())
print(c.root.the_real_answer_though)
```

Execute o servidor em uma máquina e execute o cliente na mesma máquina. Para isso basta executar o programa python do servidor sem argumentos e o do cliente com o argumento "localhost".

Questão 1: Explique o que foi impresso no cliente.

Execute o servidor em uma máquina e execute o cliente em outra máquina. Para isso basta executar o programa python do servidor sem argumentos e o do cliente em outra máquina com o argumento "nome_da_maquina_servidor".

Questão 2: Explique o que foi impresso no cliente.

Execute o programa cliente abaixo na mesma máquina.

```
import rpyc
import sys
import os

if len(sys.argv) < 2:
    exit("Usage {} SERVER".format(sys.argv[0]))

server = sys.argv[1]

conn = rpyc.connect(server,18861)

print(c.get_question)
```

Questão 3: Explique o que ocorreu.

Escreva um programa cliente que cria um vetor de n posições, onde n é definido pelo usuário, com elementos variando de 0 a n-1. Este procedimento chama um

procedimento no servidor que soma os elementos do vetor e retorna o resultado da soma. O programa cliente deve imprimir o valor de soma.

Questão 4: Mostre o código do cliente e do servidor.

Inclua no seu código do cliente e do servidor as instruções abaixo para medir o tempo de execução:

```
start = time.time()
instruções
end = time.time()
print(end-start)
```

Execute o cliente e servidor na mesma máquina para um vetor de 10000 posições.

Questão 5: Indique o tempo de execução para obter o resultado no cliente e o tempo para executar o procedimento no servidor.

Execute o cliente e servidor em máquinas diferentes para um vetor de 10000 posições.

Questão 6: Indique o tempo de execução para obter o resultado no cliente e o tempo para executar o procedimento no servidor.

Questão 7: Existe diferença nos tempos obtidos? Explique a razão de existir diferença ou de não existir diferença.

Faça uma tabela mostrando o tempo de execução do cliente e servidor executando em uma mesma máquina, com $n = \{100, 1000, 10000\}$.

Questão 8: O que você observa em relação ao tempo de execução para os três valores de n ?

Faça uma tabela mostrando o tempo de execução do cliente e servidor executando em máquinas diferentes, com $n = \{100, 1000, 10000\}$.

Questão 9: O que você observa em relação ao tempo de execução para os três valores de n ?

Questão 10: Compare as duas tabelas, indique se existem diferenças nos tempos de execução, e explique a existência ou não existência de diferenças.