

Trabalho Prático/PARTE3 – Sistemas Distribuídos – ENTREGA: 30/05/2023

Professora: Thais Regina de M. B. Silva

Semestre: 2023/01 – **Valor:** 10,0

Formato de entrega: PDF + código

O objetivo das 3 partes finais deste trabalho é mostrar aos alunos as diferenças, benefícios, desvantagens e dificuldades do desenvolvimento de sistemas distribuídos utilizando diferentes visões de elementos arquitetônicos. Em outras palavras, será uma oportunidade de construir uma experiência sobre como é o desenvolvimento de SDs com e sem o uso de *middlewares*, considerando inclusive diferentes alternativas de camadas de software como essas.

Para atingirmos este objetivo, cada grupo fará 3 implementações da rede social proposta. A primeira (Parte 3) será feita utilizando simplesmente a API de Sockets, configurando-se assim como o uso da visão de processos para os elementos arquitetônicos. A segunda implementação (Parte 4) utilizará o recurso de um *middleware* RMI, quando os elementos arquitetônicos serão vistos como objetos. Por fim, a última implementação (Parte 5) utilizará um *middleware Web Service*.

Desta forma, nesta, que é a parte 3 do trabalho, toda a implementação deverá ser feita com uso apenas da API de Sockets. Cada elemento do sistema deverá ser implementado como um processo e a comunicação deverá ser feita por fluxo TCP. Observe que, ao usar a API de Sockets, a comunicação é feita entre processos, utilizando troca de mensagens, as quais devem ser inteiramente construídas e gerenciadas pelo programador. Em outras palavras, vocês serão responsáveis por definir a estrutura das mensagens a serem trocadas e deverá programar todo o processo de construção e leitura das mesmas. O recurso de *threads* deverá ser utilizado nesta implementação de forma bastante simples: o processo servidor deverá utilizar uma *thread* para receber as requisições recebidas e outra (a principal) para processá-las. O uso de interface gráfica (simples, sem exageros) é obrigatório. Para a implementação deverá ser utilizada obrigatoriamente a linguagem Python (e somente ela).

- Requisitos:

- 1) O SD será testado no sistema operacional Linux. Sugiro fortemente o uso deste SO no desenvolvimento do trabalho, visto que não serão feitas adaptações nos códigos entregues para que os mesmos possam executar corretamente.
- 2) O SD deve possuir base de dados implementada via SQLite com arquivo[1], devendo o mesmo já conter tudo o que for necessário para que o sistema seja utilizado com sucesso (ou seja, o banco já precisa vir populado).
- 3) Utilização de arquivo makefile para facilitar a execução do SD. Você pode criar os comandos do makefile da maneira que achar mais conveniente (run, install, etc...), porém tudo terá que estar explicando do arquivo README.txt mencionado abaixo.
- 4) Utilização de ambiente virtual leve do Python (virtualenv) [2]. Para rodar seu SD, considere que terá que ser criado o virtualenv contendo todas as instalações necessárias. Desta forma, a criação deste ambiente deverá fazer parte das tarefas automatizadas pelo seu makefile.
- 5) Considerar o uso de Python versão 3.

- O que deve ser entregue:

- documentação pequena porém completa, descrevendo o projeto do sistema distribuído desenvolvido com uso da API de Sockets. Deixe bem claras todas as escolhas de implementação, em particular das mensagens trocadas entre os processos. Coloque uma conclusão em que vocês relatam as principais dificuldades percebidas com a programação via Sockets, indicando se e como as mesmas puderam ser superadas ou contornadas. A qualidade e o capricho com a documentação também serão considerados na avaliação.
- arquivo README.txt contendo o passo a passo detalhado para a correta execução do SD desenvolvido. Esta informação também poderá estar no texto da documentação.
- todo código fonte produzido.

Cada dupla submeterá, via PVANET Moodle, um arquivo compactado (.zip ou .tar.gz) contendo tudo o que foi pedido. O nome do arquivo deve ser: <nomes_alunos>_TP1P3_SD.<zip/tar.gz>

[1] <https://pythonclub.com.br/gerenciando-banco-dados-sqlite3-python-parte1.html>

[2] <https://docs.python.org/pt-br/3/library/venv.html>