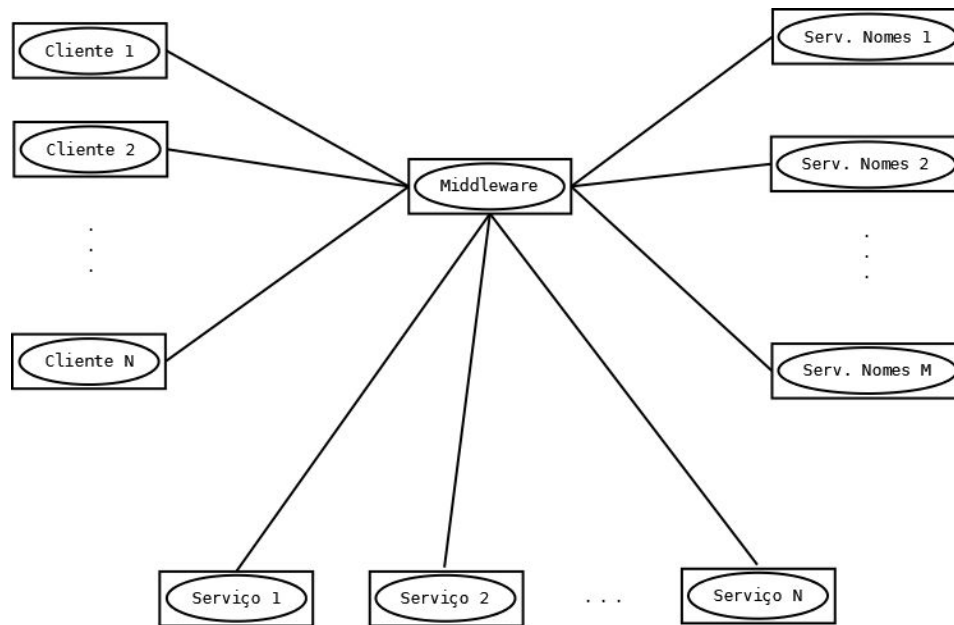
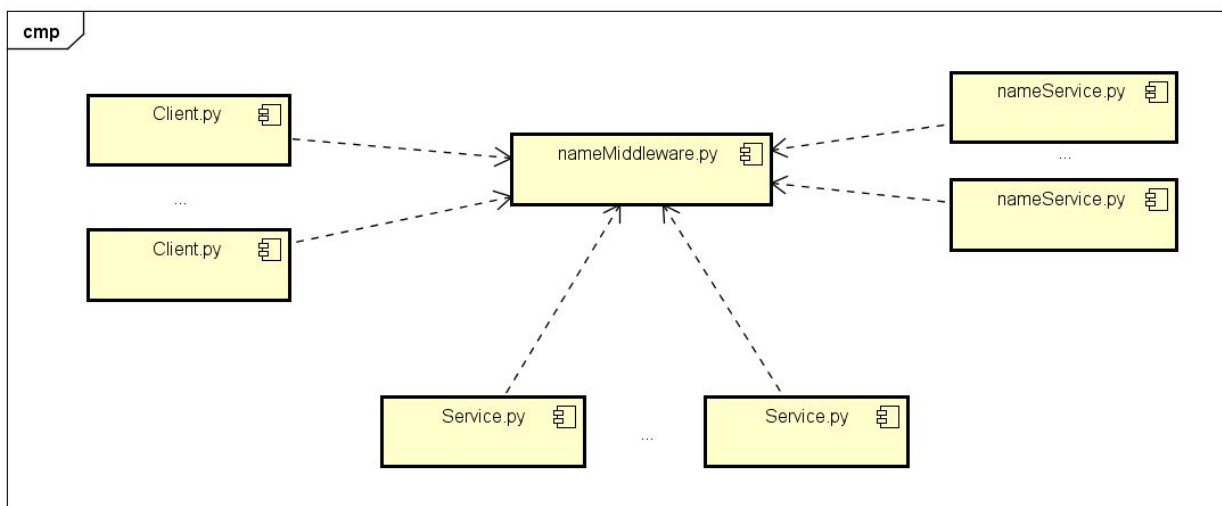


TRABALHO FINAL - SISTEMAS DISTRIBUÍDOS SERVIDOR DE NOMES EM PYTHON

- Diagrama Inicial:



- Diagrama de Componentes (UML):



- **Descrição do Projeto:**

A ideia principal é desenvolver um servidor de nomes distribuído, utilizando sockets, na linguagem python. Os clientes não “conhecem” o servidor e os serviços (não possuem o endereço ip e porta dos mesmos), conectando-se diretamente apenas à um middleware, que serve como “ponto central”: os servidores se conectam à ele, bem como os serviços e clientes.

Para conectar-se à um serviço, o cliente busca pelo nome do serviço ou por palavras-chave (características do serviço desejado), que são próprias de cada serviço (estão detalhadas segundo um “protocolo próprio” dentro de cada serviço - quando ele é iniciado ele informa essas características). Ao receber a requisição, através de troca de mensagens (sockets), o middleware busca um servidor (que esteja menos ocupado) para resolver o nome (buscar pelas características informadas), retornando o par ip e porta de volta ao cliente que, por fim, se conecta e requisita o serviço, obtendo resposta.

- **Questões Relativas à Implementação:**

Para a implementação, foi escolhida a linguagem Python devido a sua simplicidade (no que é tangente à codificação), trazendo a oportunidade de centrar os esforços no sistema e nos conceitos da disciplina envolvidos, possibilitando abstrair certos detalhes que seriam necessários com linguagens como C e Java, por exemplo.

No processo de troca de mensagens foi utilizado o conceito de sockets, justificado pela arquitetura do próprio problema em si, que necessita da troca de mensagens para que a comunicação se estabeleça e o serviço seja providenciado. Houveram tentativas de migração para o RPC ou RMI, mas por questões de projeto e por tudo o que fora mencionado, não foram continuadas.

A implementação foi composta de uma série de arquivos, sendo eles:

- ***Client.py*** : cliente, que se conecta ao middleware e faz a requisição dos serviços, utilizando as palavras-chave (ex: vídeo engraçado) ou o nome do serviço, recebe o endereço (ip e porta) do serviço e faz a requisição, esperando pelo resultado.
- ***nameService.py*** : servidor que faz a resolução dos nomes. Recebe a sinalização de um serviço, quanto este é iniciado, e busca pelas palavras-chave informadas pelo middleware (resolução em si). Retorna ao cliente o endereço do serviço que atende à requisição realizada.
- ***nameMiddleware.py*** : middleware, que recebe a sinalização dos servidores de nomes e serviços que são inicializados, bem como encaminha o ip e porta do servidor de nomes ao cliente.



- **Server.py** : serviços. Possui características, nome e palavras-chave e informa-as aos servidores de nomes disponíveis. Recebe alguma entrada que o cliente precise informar e manda o serviço para o middleware, assim que solicitado.

- **Conclusão:**

É importante, em uma disciplina como a de Sistemas Distribuídos, que possui um viés teórico acentuado, ter a possibilidade de implementar e visualizar na prática o resultado de tudo o que é visto em sala de aula. O momento de implementação se torna, portanto, oportunidade de aprendizado e de ampliação/consolidação dos conceitos abordados.

