

Alunos:

Bianca Garcia Martins 606723

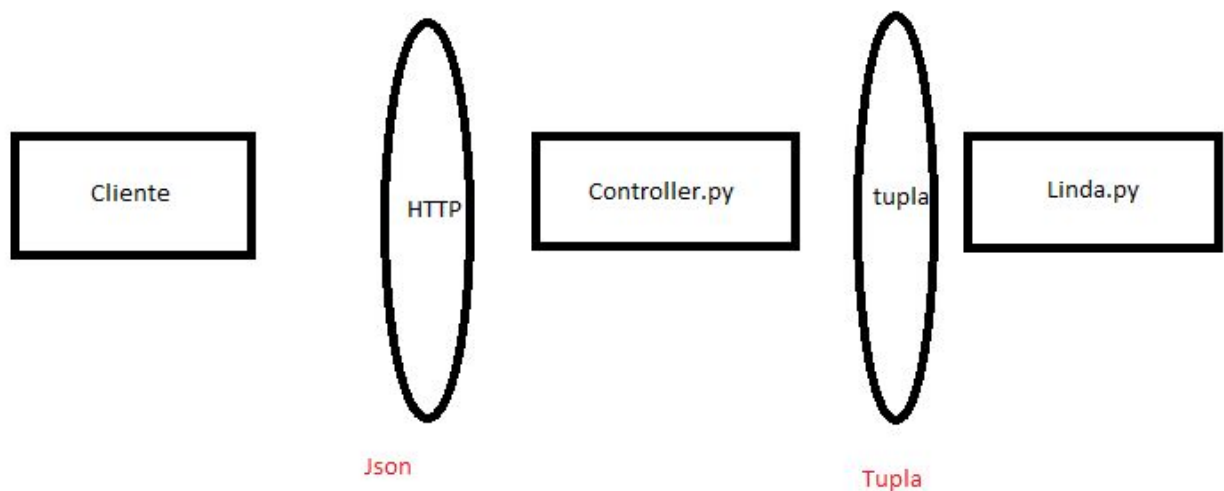
Isadora Eliziário Gallerani 726542

Esse projeto utiliza REST api e Flask no servidor. Para utilizá-lo, instale o flask e flask-restful:

```
pip install flask
```

```
pip install flask-restful
```

A arquitetura utilizada no projeto é representada abaixo:



No cliente é feita uma requisição http (*put, get* ou *delete*) dependendo do interesse do cliente (inserir, ler ou deletar uma mensagem). Essa requisição é interpretada no servidor (*server.py*), que chama as funções *add()*, *read()* e *remove()* do *Controller.py*, que por sua vez chama as funções da Linda Tuple Space para cada operador (*in, out, rd*), passando uma tupla. O *server.py* gera uma resposta no formato json, que será retornada ao cliente, de forma tratada.

No projeto utilizamos o host *localhost* e a porta *5000*, mas sinta-se a vontade para alterá-los nos arquivos *server.py* e nos arquivos de exemplos de clientes.

Para testar, basta rodar o servidor (*server.py*) em um terminal, e em outro, rodar um dos exemplos de clientes: *Alice.py*, *Bob.py*, *Chuck* ou *Remove.py*, sendo os dois primeiros exemplos da função *out*, o terceiro, exemplo da função *rd*, e o quarto, exemplo da função *in*.

Um exemplo de teste para verificar a remoção de mensagens é feita rodando o *Alice.py*, seguido do *Bob.py*, *Chuck.py*, *Remove.py* e finalmente *Chuck.py*. Assim, pode se observar mensagens sendo postadas por cada cliente, as mensagens de certos publishers existentes em determinados tópicos, e a remoção de determinadas mensagens.

Montando exemplos

Sinta-se livre para montar exemplos nos mesmos moldes dos clientes já existentes.

Para criar exemplos de remoção e leitura de mensagens, necessita-se passar os parâmetros *publisher*, *topic* e *message* na url da requisição, de acordo com seu interesse. No caso de remoção, os três parâmetros são essenciais, o que não é verdade para a leitura. Dessa forma, somente é permitido remover uma mensagem, se for informado quem publicou, o tópico e a mensagem. No caso de leitura, se o cliente quiser ler todas as mensagens publicadas por um determinado publicador, independente do tópico, basta passar o parâmetro *publisher*. Se o cliente quiser ler todas as mensagens de um determinado publicador em um tópico, basta informar o *publisher* e o *topic*. Mas se o cliente quiser ler uma mensagem específica, ele precisa informar o *publisher*, *topic* e *message*.

- *Exemplo de remoção (função _in):*

```
response=requests.delete("http://localhost:5000/blog?publisher='Alice'&topic='gtcn'&message='This graph theory stuff is not easy???'")
```

- *Exemplo de leitura (função _rd):*

```
response = requests.get("http://localhost:5000/blog?publisher='Bob'&topic='distsys'")
```

Para exemplos de inserção de mensagens, basta criar umas request onde o dado(*data*) é feito no formato json, no qual a chave é *tuple*, e os valores são os *publisher*, *topic* e *message* de seu interesse.

- *Exemplo de inserção(função _out):*

```
data = {  
    'tuple':"'Bob','gtcn','Cool book!'"  
}  
response = requests.post('http://localhost:5000/blog', data)
```