



# Seminário - Sistemas Distribuídos

Frederico Martins e Lucas Loges



# Proposta

Material didático

Técnicas de implementação

- Similaridade semântica entre a interface e a implementação
- Simplicidade

Praticidade no processo de teste

- Automação do processo de atribuição de endereços e lançamento
- Facilidade de observação dos acontecimentos durante a execução

# Exemplo de módulo

```
class FairLossPointToPointLink:
```

```
"""
Module 2.1: Interface and properties of fair-loss point-to-point links

Events
-----
Send(q, m):
    Requests to send message m to process q.
Deliver(p, m):
    Delivers message m sent by process p.

Properties
-----
FLL1: Fair-loss:
    If a correct process p infinitely often sends a message m to a
    correct process q, then q delivers m an infinite number of times.
FLL2: Finite duplication:
    If a correct process p sends a message m a finite number
    of times to process q, then m cannot be delivered an infinite number of times by q.
FLL3: No creation:
    If some process q delivers a message m with sender p, then m
    was previously sent to q by process p.
"""
```

```
def __init__(self, address, on_delivery = print):
    self.handle_delivery = on_delivery

    self.socket = socket(type=SOCK_DGRAM)
    self.socket.bind(solve_address(address))

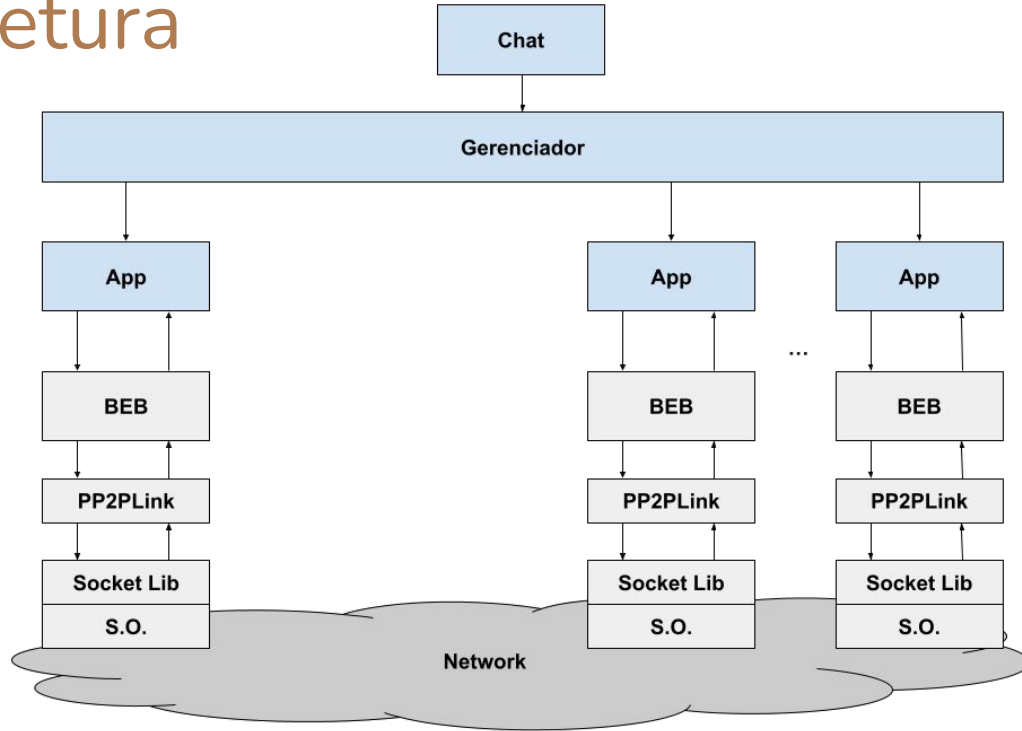
    self.threads = [Thread(target=f) for f in (self.watch,)]
    for t in self.threads:
        t.start()

def Send(self, q, m):
    # Requests to send message m to process q.
    q = solve_address(q)
    self.socket.sendto(m.encode(), q)

def Deliver(self, p, m):
    # Delivers message m sent by process p.
    p = solve_address(p)
    self.handle_delivery('.'.join(map(str, p)), m)

def watch(self):
    # Watches the link.
    while(True):
        message, address = self.socket.recvfrom(1024)
        self.Deliver(address, message.decode())
```

# Arquitetura



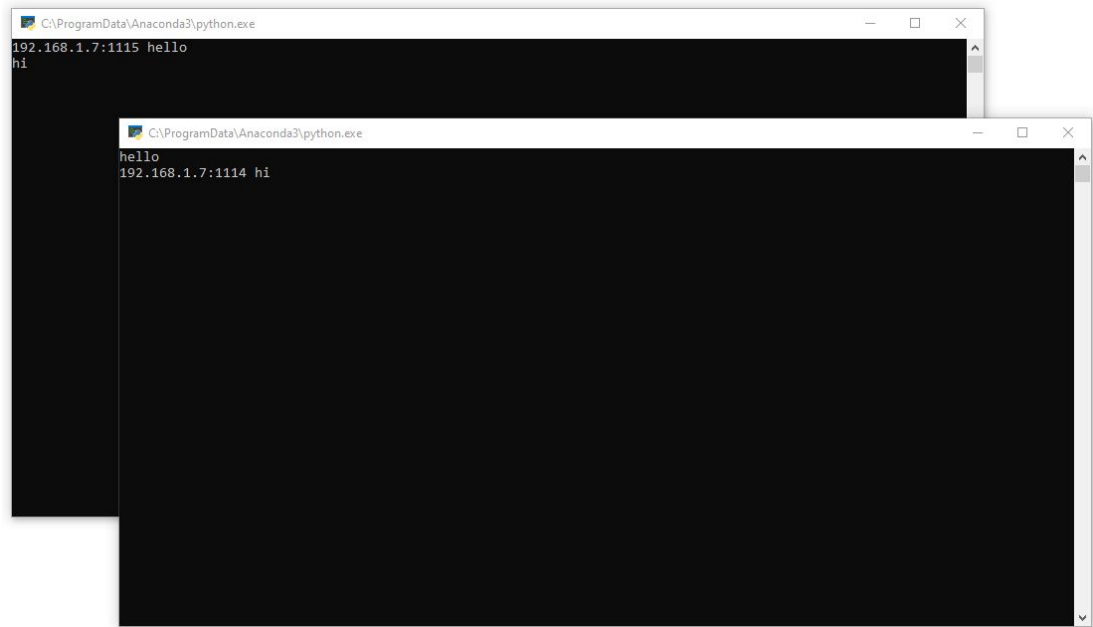
# Uso do gerenciador

O gerenciador tem o trabalho de lançar os processos distribuídos, resolvendo os endereços utilizados.

- Pode ser utilizado em uma ou múltiplas máquinas.
- Quando lançado, o gerenciador lista outros gerenciadores (em outras máquinas) que foram descobertos.
- Ao adicionar um gerenciador, é feita uma união dos grupos dos dois.
- Toda ação executada em um membro do grupo é executada em todos os membros.

# Demonstração

```
>>> from manager import Manager
>>> m = Manager('FairLossPointToPointLink', 2)
Choose an ip:
0: 192.168.56.1
1: 192.168.1.7
1
>>> m.ready()
```



The image shows two overlapping terminal windows running Python code. The top window, titled 'C:\ProgramData\Anaconda3\python.exe', displays the output of a network program: '192.168.1.7:1115 hello' followed by 'hi' on the next line. The bottom window, also titled 'C:\ProgramData\Anaconda3\python.exe', displays 'hello' followed by '192.168.1.7:1114 hi' on the next line. Both windows have a black background with white text.