

## **1) organização do código:**

### **1.1) Descrição das funcionalidades dos módulos ou classes;**

Para este trabalho temos 5 classes, são elas a classe ClientNode, MultiCastController, MulticasMessageFormat, Resource e SuperNode.

A classe SuperNode é responsável por concentrar a lógica do programa, mantendo dentro dela os MulticastControllers dos SuperNodos e ClientNodes. Basicamente ela é ciente de todos os nodos com quem ela pode se conectar e de seus recursos.

A classe ClientNode é responsável por ter recursos e se conectar a um Supernodo através de um multicast próprio.

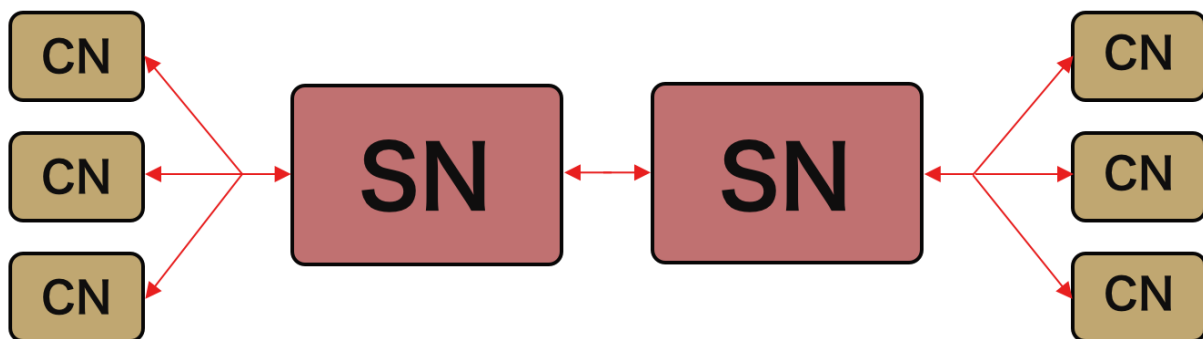
A classe MulticasController é responsável por implementar a comunicação multicast utilizada nas classes de nodo, aqui implementamos através de sockets os métodos de envio e recebimento de mensagens, bem como o método para terminar a comunicação.

A classe MulticasMessageFormat é responsável pela formatação das mensagens do Multicast, no nosso caso o formato da mensagem contempla o "request", "sender", "body" e "originalMessage".

Por último, a classe Resource é responsável pela administração dos recursos dos Nodes, ou seja, é responsável pela função hash, pela conversão dos arquivos para a comunicação multicast e por checar se nenhuma parte do recurso foi perdida.

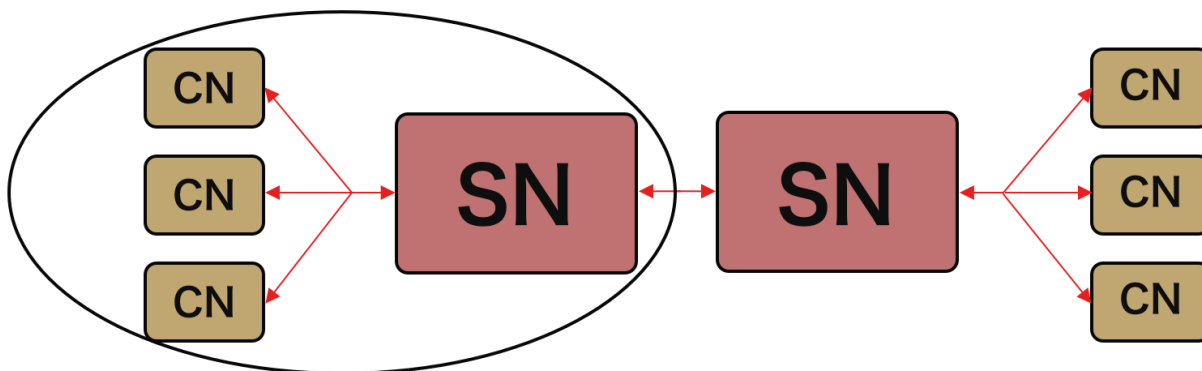
## **2) utilização dos programas:**

Basicamente os super nodos comunicam-se entre si e entre os nodos normais (ClientNodes) através da comunicação multicast. Podemos ver na imagem abaixo as possíveis comunicações do nosso programa.



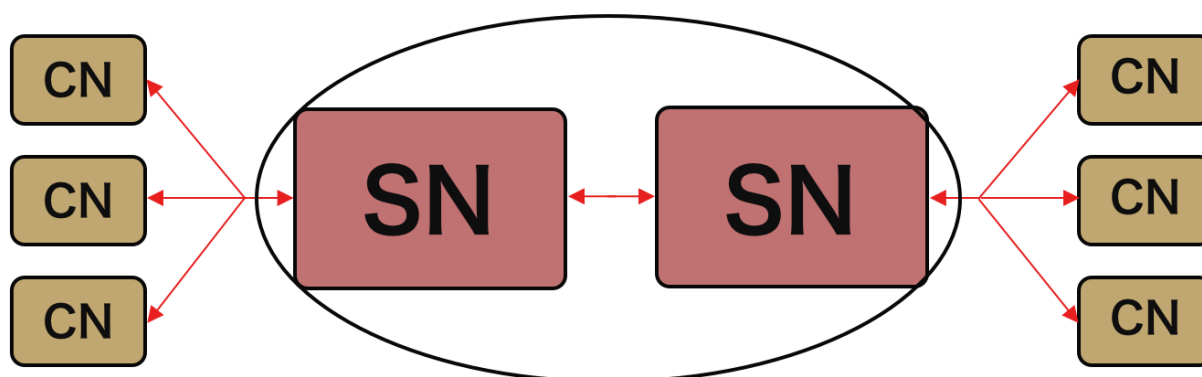
**Figura 01 - Demonstração de Nodos**

Entretanto vale salientar que não usamos a mesma comunicação multicast para todas as comunicações. As comunicações entre Super nodo e nodo normal tem uma multicas própria, bem como as comunicações entre nodo e nodo tem outra multicast própria. Podemos observar melhor nas imagens abaixo.



**Figura 02 - Multicast SN para CN**

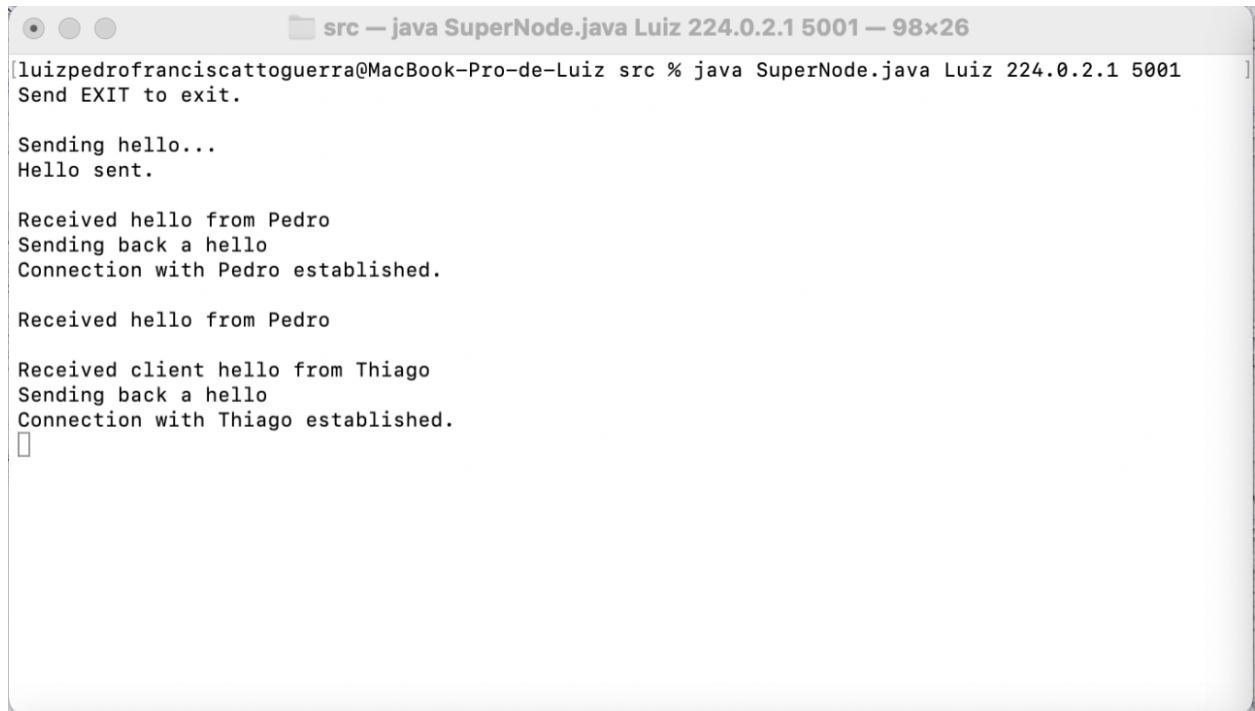
Nestas imagens podemos observar melhor que teremos multicasts diferentes para cada tipo de relação de comunicação.



**Figura 03 - Multicast SN para SN**

### **3) demonstração da implementação, apresentando casos de uso relevantes.**

Na Série de Imagens abaixo temos a demonstração do Supernodo Luiz se conectando com o Super nodo Pedro e com o nodo Thiago e o Super nodo Pedro se conectando com o Supernodo Luiz e com o nodo Fab. Vale Salientar que o número da porta difere nas conexões dos nodos justamente para diferenciar em qual Supernodo eles irão se conectar. Nas Conexões os nodos mandam mensagem entre si e se eles recebem uma mensagem de volta a conexão é estabelecida.

A terminal window titled 'src — java SuperNode.java Luiz 224.0.2.1 5001 — 98x26'. The terminal shows the execution of 'java SuperNode.java Luiz 224.0.2.1 5001'. The output includes: 'Send EXIT to exit.', 'Sending hello...', 'Hello sent.', 'Received hello from Pedro', 'Sending back a hello', 'Connection with Pedro established.', 'Received hello from Pedro', 'Received client hello from Thiago', 'Sending back a hello', 'Connection with Thiago established.', and a cursor at the end of the line.

```
src — java SuperNode.java Luiz 224.0.2.1 5001 — 98x26
[luizpedrofranciscattoguerra@MacBook-Pro-de-Luiz src % java SuperNode.java Luiz 224.0.2.1 5001
Send EXIT to exit.

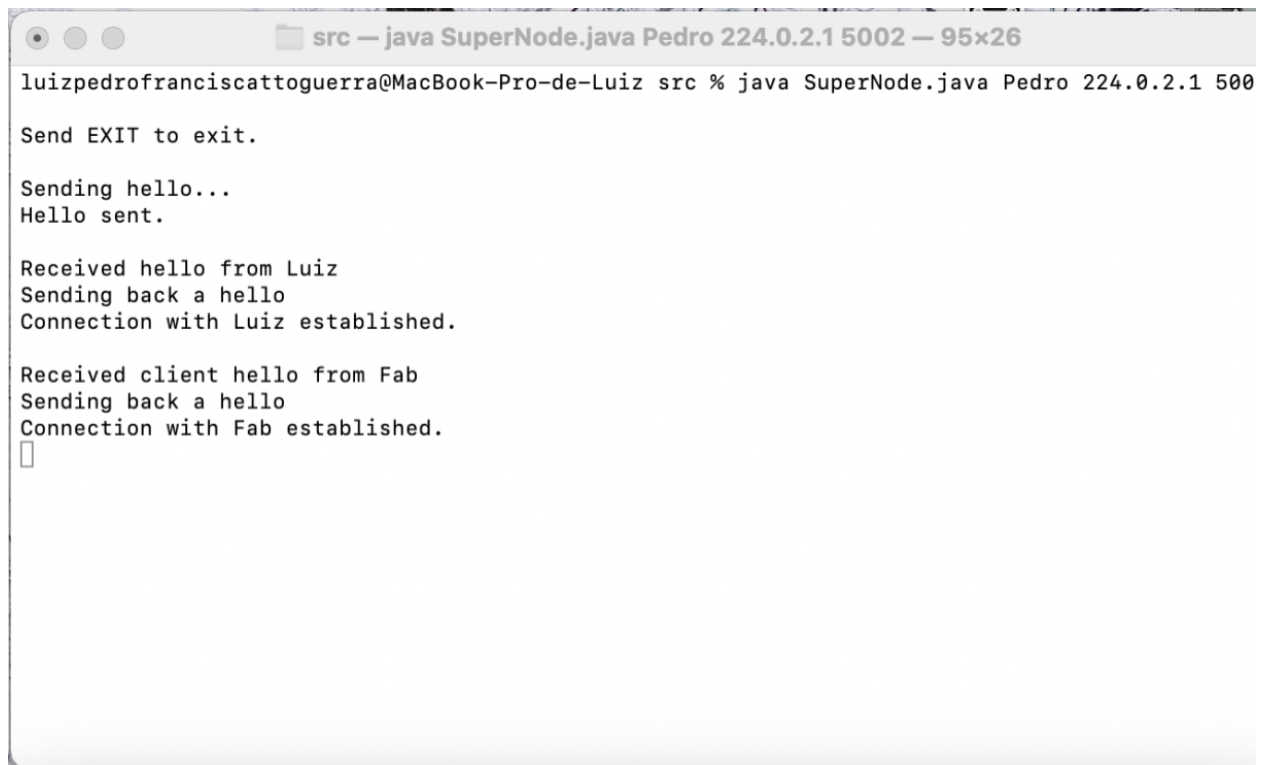
Sending hello...
Hello sent.

Received hello from Pedro
Sending back a hello
Connection with Pedro established.

Received hello from Pedro

Received client hello from Thiago
Sending back a hello
Connection with Thiago established.
█
```

**Figura 04 - Demonstração Super nodo Luiz**

A terminal window titled 'src — java SuperNode.java Pedro 224.0.2.1 5002 — 95x26'. The terminal shows the execution of 'java SuperNode.java Pedro 224.0.2.1 5002'. The output includes: 'Send EXIT to exit.', 'Sending hello...', 'Hello sent.', 'Received hello from Luiz', 'Sending back a hello', 'Connection with Luiz established.', 'Received client hello from Fab', 'Sending back a hello', 'Connection with Fab established.', and a cursor at the end of the line.

```
src — java SuperNode.java Pedro 224.0.2.1 5002 — 95x26
luizpedrofranciscattoguerra@MacBook-Pro-de-Luiz src % java SuperNode.java Pedro 224.0.2.1 5002


Send EXIT to exit.

Sending hello...
Hello sent.

Received hello from Luiz
Sending back a hello
Connection with Luiz established.

Received client hello from Fab
Sending back a hello
Connection with Fab established.
█
```

**Figura 05 - Demonstração Super nodo Pedro**



A terminal window titled "src — java ClientNode Thiago 5001 — 98x25". The prompt is "luizpedrofranciscattoguerra@MacBook-Pro-de-Luiz src %". The command executed is "java ClientNode Thiago 5001". The output shows the program's usage instructions and its initial actions.

```
[luizpedrofranciscattoguerra@MacBook-Pro-de-Luiz src % java ClientNode Thiago 5001
Usage:
Send EXIT to exit.

Send R to ask for all resources
Sending hello...
Hello sent.

]
```

**Figura 06 - Demonstração nodo Thiago**



A terminal window titled "src — java ClientNode Fab 5002 — 95x25". The prompt is "luizpedrofranciscattoguerra@MacBook-Pro-de-Luiz src %". The command executed is "java ClientNode Fab 5002". The output shows the program's usage instructions and its initial actions.

```
[luizpedrofranciscattoguerra@MacBook-Pro-de-Luiz src % java ClientNode Fab 5002
Usage:
Send EXIT to exit.

Send R to ask for all resources
Sending hello...
Hello sent.

]
```

**Figura 07 - Demonstração nodo Fab**