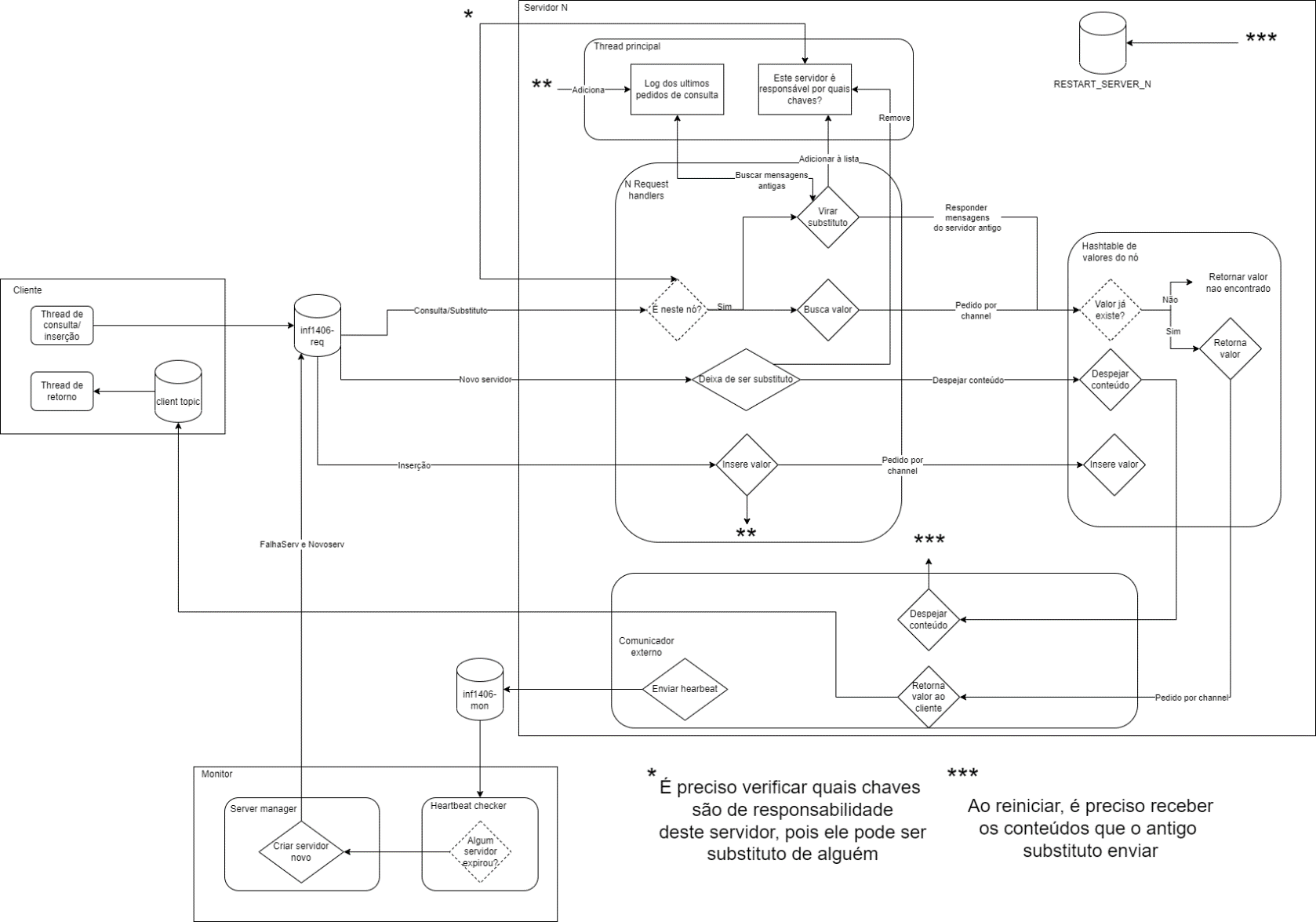
**Alunos:**

* **Fernando Bernardo dos Santos (1710160)**
* **Yuri Zoel Brasil (1710730)**

**Introdução:**

A criação dos componentes se deu seguindo o seguinte diagrama de fluxo:

Possuimos 3 componentes distintos: cliente, servidor e monitor. Os clientes podem enviar 3 tipos de requisições ao tópico de requisições: INSERT, SEARCH e DIE. Sendo o DIE para mérito de debug e testes.

Um cliente recebe os seguintes argumentos: <TIPO OPERAÇÃO> <CHAVE PARA INSERT> <VALOR> <NUM DO CLIENTE> <VALOR DE SEARCH ESPERADO> <SERVIDOR ESPERADO PARA RESPONDER SEARCH>

Um servidor recebe os seguintes argumentos: <NUM DO SERVIDOR> <NUM TOTAL DE SERVIDORES> <TIPO DE BOOT (normal ou restart)>

O monitor recebe os seguintes argumentos: <NUM TOTAL DE SERVIDORES>

**Detalhes de implementação:**

Implementamos em RUST e utilizamos o Mosquitto para ser o nosso servidor MQTT. Todos os servidores escutam no tópico “inf1406-reqs” para os tipos de mensagem. Caso receba um pedido de FALHASERV, ele verifica se ele deve assumir aquele servidor. Ele verifica se ele, ou algum dos servidores ao qual ele é atualmente substituto, deve assumir, e se for o caso ele assume o servidor como substituto.

Todos os servidores armazenam os logs dos pedidos de busca, mesmo que não sejam para eles. Caso eles virem substitutos de algum servidor, eles varrem a lista de logs, e verificam se há um pedido com o timestamp de pedido superior ao último heartbeat do servidor morto. Se for o caso, eles respondem este pedido de insert com o valor armazenado atualmente.

Ao iniciar o servidor N, há dois tipos de inicialização que podemos passar para <TIPO DE BOOT>: BOOT e RESTART. Caso utilize o RESTART, o servidor irá esperar uma mensagem no tópico RESTART\_SERVER\_N, com o dump do conteúdo atual do servidor substituto. Após realizar o dump, o substituto remove o servidor reiniciado da sua lista, e para de responder requisições dele.

É o monitor que é reponsável por inicializar e reiniciar os servidores mortos.