

## Sistemas Multinível e Distribuição

Mestrado em Engenharia Informática

Instituto Superior de Engenharia do Porto

---

# TheyChat! Um sistema de conversação fiável!

## Introdução

O objetivo deste exercício é desenvolver um sistema de mensagens simples baseado em texto chamado TheyChat! A implementação usará Erlang e será fortemente baseada nos processos de Erlang e nas características de passagem de mensagens, com foco na fiabilidade e tolerância a falhas.

## Comportamento do sistema

O TheyChat! utiliza uma arquitetura cliente/servidor e um padrão de conceção supervisor/trabalhador. Os clientes ligam-se aos servidores e podem conversar uns com os outros (utilizando o servidor). O foco deste exercício não serão as funcionalidades de conversação, mas sim a fiabilidade e a tolerância a falhas da arquitetura. Haverá três nós principais a serem implementados:

**Cliente :** Capaz de entrar/sair de servidores e enviar/receber mensagens.

**Router :** Responsável por indicar a um cliente o PID de um servidor ao qual o cliente pretende aderir e por monitorizar e resolver falhas do servidor.

**Servidor :** Este nó é onde os clientes podem conversar partilhando mensagens e também deve implementar funcionalidades de segurança.

Na figura 1 podemos ver uma proposta de organização dos nós para atingir o objetivo deste projeto. Em mais pormenor, note-se que os processos `router` e `router_monitor` se supervisionam mutuamente, o que significa que se um deles falhar o outro reinicia-o. O processo `server_monitor` supervisiona (e reinicia em caso de falha) o processo `server_chat`. O router também é responsável por reiniciar o `server_monitor` em caso de falha. É muito importante notar que um dos recursos que os processos de monitoramento devem fornecer são os backups de estado, ou seja, quando algum processo falha, seu supervisor deve reiniciá-lo com os dados de estado que ele tinha quando falhou. Por exemplo, se um processo servidor falhar, ele deve reiniciar com a lista de clientes que tinha antes de falhar, de forma que possa retomar a operação graciosamente. É conveniente ter todos os processos a correr na mesma máquina (mas em máquinas virtuais Erlang diferentes), uma vez que é fácil escalar essa solução para uma baseada em máquinas em rede diferentes. O número e a funcionalidade dos processos no sistema devem ser os adequados para atingir o objetivo e não necessariamente os descritos em

Figura 1.

O protocolo de comunicação da interação cliente/servidor não está predefinido, mas deve permitir as seguintes acções:

Juntar-se a um determinado servidor.

Abandonar um determinado servidor.

Enviar e receber mensagens de conversação.

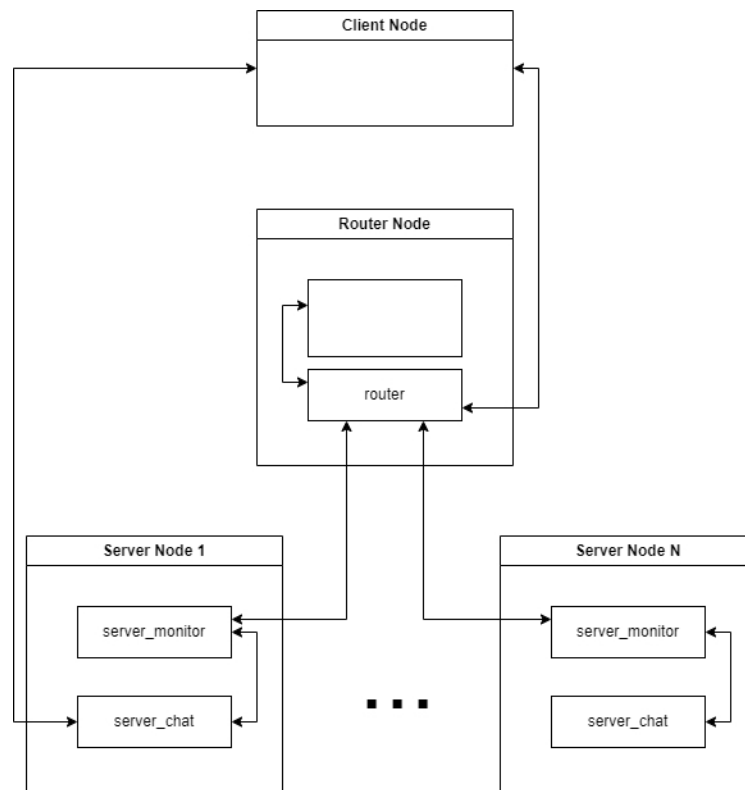


Figura 1: Nós cliente, router e servidor

## Tratamento de erros

O tratamento de erros deve ser possível quando:

- O router falha

- Falha do servidor

- Falha do cliente

O tratamento de erros neste projeto tem o significado mais lato de reiniciar os processos falhados, retomando o funcionamento do sistema com um impacto mínimo ou sem impacto para as diferentes entidades. Este não será o caso dos clientes. Quando um cliente falha, o servidor tem de o notar e atualizar a lista de clientes para deixar de enviar mensagens a esse cliente.

## Submissão e apresentação

Não é necessário qualquer relatório, sendo apenas necessário o envio dos programas cliente e servidor (a data limite é 26<sup>th</sup> de maio). A apresentação decorre de 27<sup>th</sup> de maio a 7<sup>th</sup> de junho.

## Classificação

Este trabalho deverá ser resolvido individualmente e avaliará os conhecimentos sobre

programação distribuída na linguagem funcional Erlang. Os critérios de avaliação são descritos

seguinte.

Descrição	Percentagem (%)
Vários clientes e pelo menos um servidor que comunicam através de mensagens	25
Implementação básica do router	10
Tolerância a falhas do router	10
O router supervisiona o servidor	10
Funcionalidade básica do servidor	15
Tolerância a falhas do servidor	15
Suporta diferentes servidores (para diferentes tópicos)	15

## Código de honra

No "Código de boas Práticas de Conduta" a partir de 27 de outubro de 2020, afirma-se

que os estudantes devem apresentar uma declaração, tal como descrito no artigo 8º, para cada projeto apresentado. Esta declaração constitui um compromisso assinado de originalidade do trabalho apresentado. A não apresentação desta declaração retira o trabalho da avaliação. A apresentação de um trabalho que não cumpra a declaração assinada terá consequências legais.