

# Programação concorrente e distribuída

Professor: Nelson Souto Rosa

Grupo:

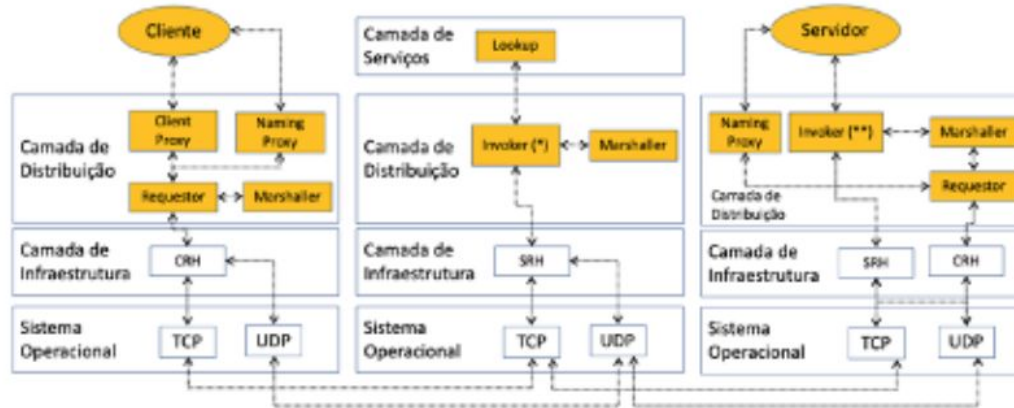
Felipe Sinésio Trajano de Arruda

Pedro Henrique Lopes da Silva

Relatório referente à atividade 6  
Implementação de Middleware  
(Camadas de distribuição e serviços)  
utilizando a camada de infraestrutura  
desenvolvida na atividade 5

# Avaliação de desempenho

- Objetivo: Implementar Pooling padrão de projeto.
- Serviço do Sistema: Cliente solicita que o servidor avalie uma expressão matemática e envie uma resposta da avaliação
- Métricas de Desempenho: “Round trip time” da solicitação





# Principais partes do código

```
1 referencia
38 private static void executeClient(Object obj)
39 {
40     LookupProxy lookupProxy = LookupProxy.GetInstance();
41     ICalculadora calculadora = new CalculadoraProxy(lookupProxy.Lookup("calculadora"));
42     for (int i = 0; i < 10000; i++)
43     {
44         var watch = new System.Diagnostics.Stopwatch();
45         watch.Start();
46         calculadora.Eval(i + "*2+1");
47         watch.Stop();
48         tempoTranscorrido += watch.ElapsedMilliseconds;
49         tempos.Add(watch.ElapsedMilliseconds);
50     }
51 }
```

# Principais partes do código

```
7  class Pool
8  {
9      public Queue<Server.Server> pool = new Queue<Server.Server>(10);
10
11     1 referência
12     private Pool()
13     {
14         for(int i = 0; i < 10; i++)
15         {
16             pool.Enqueue(new Server.Server());
17         }
18
19     private static Pool _instance;
20
21     private static readonly object _lock = new object();
22
```



# Parâmetros

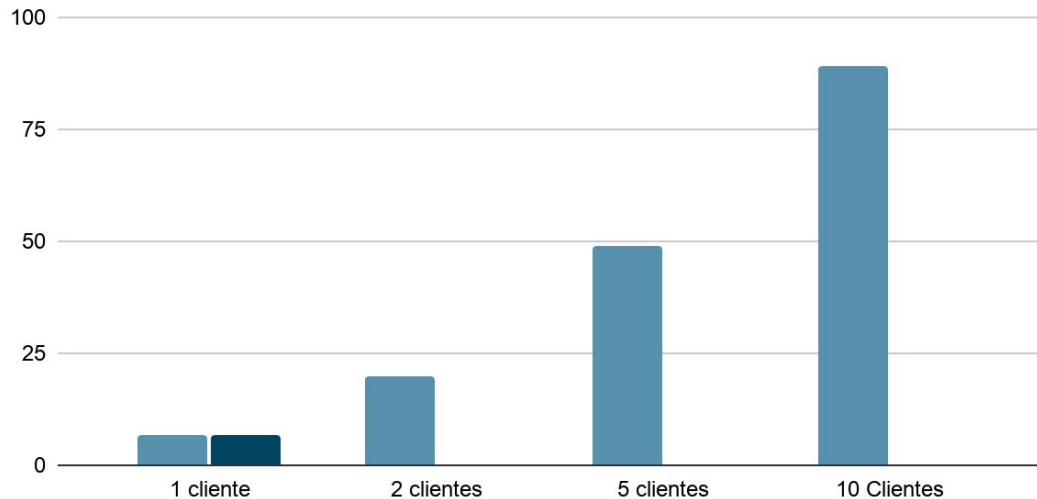
Parâmetros de sistema utilizados	Valor
Processador	3,45GHz Intel Core i5
RAM	8GB
Wifi	Ligada
Sistema Operacional	Windows 10



# Análise de resultados (Média)

Media de tempos (Milissegundos)

■ MyMiddleware ■ MyRPC





# Análise de Resultados (Desvio Padrão)

Desvio padrão (Millisegundos)

