



UNIVERSIDADE  
ESTADUAL de LONDRINA

---

VITOR HENRIQUE ROSA BATISTA DE OLIVEIRA  
VITOR PALMA ADERALDO

**SISTEMAS OPERACIONAIS: RELATÓRIO MICRO BLOGUE  
COM RMI**

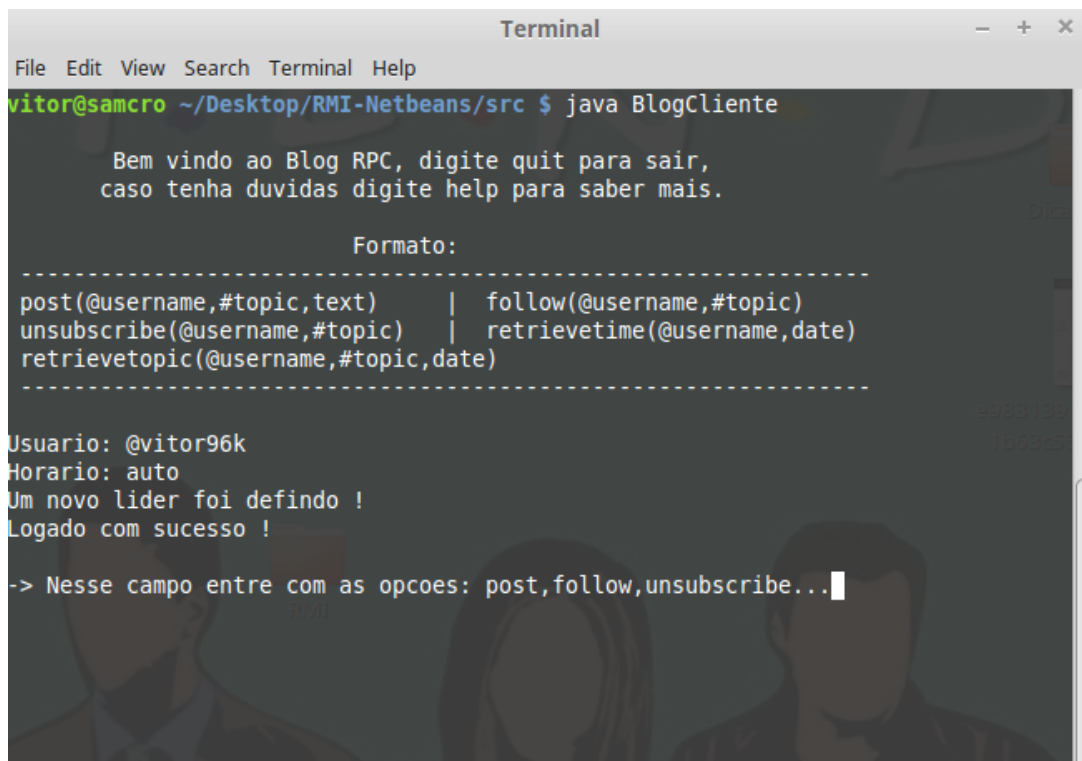
LONDRINA - PR  
2016

---

## O SISTEMA

Primeiramente o Servidor deve ser executado para o funcionamento do blog, lembrando que o Servidor se comunica com uma base de dados existente no PostgreSQL. Todas informações são armazenadas nesse banco que contém uma tabela de posts e outra de seguidores.

Ao executar o cliente, será aberto a tela de login, onde deve-se apenas digitar o seu nome de usuario e em seguida digitar o horário, para o programa funcionar corretamente entre com o valor "auto" para o campo horario, caso queira testar o programa com um horario desatualizado entre com um valor do tipo "22:52:64" (o horario é efinido manualmente em vez de ser automatico), desse modo pode-se oversavar a sincronização de relógios com grande diferença de tempo. Após entrar com o nome de usuário e horário, provavelmente o sistema irá mostrar a mensagem "Um novo líder foi definido !". Isso indica que ao se conectar foi feito uma eleição e o vencedor foi você (antes a liderança pertencia ao Servidor).



```
Terminal
File Edit View Search Terminal Help
vitor@samcro ~/Desktop/RMI-Netbeans/src $ java BlogCliente

Bem vindo ao Blog RPC, digite quit para sair,
caso tenha duvidas digite help para saber mais.

Formato:
-----
post(@username,#topic,text)      | follow(@username,#topic)
unsubscribe(@username,#topic)    | retrievetime(@username,date)
retrievetopic(@username,#topic,date)
-----

Usuario: @vitor96k
Horario: auto
Um novo lider foi definido !
Logado com sucesso !

-> Nesse campo entre com as opcoes: post,follow,unsubscribe...
```

Apos mostrar a mensagem de “Logado com sucesso !” o usuario deve inserir os comandos para operar no blog como: post, follow, unsubscribe, retrievetime, retrievetopic ou help. Lembrando que novos comandos foram inseridos nesse trabalho para gerenciar eleições, sincronização de relógio e controle de escrita/leitura.

Observação: O banco de dados utiliza duas tabelas, uma chamada posts que serve para guardar os posts dos usuários, e outra chamada follow que é usada para guardar informações a respeito dos tópicos seguido pelo usuário. Se a mensagem de boas vindas apareceu, significa que a

conexão foi realizada com sucesso, pois essa mensagem é enviada pelo servidor e não está inserida no código do Cliente.

## COMANDOS

**Post:** Usado para fazer um post. Após digitar esse comando o programa irá pedir para informar o tópico e seu respectivo texto.

**Follow:** Serve para seguir determinado tópico. Após digitar esse comando, o programa irá pedir o tópico que você deseja seguir.

**Unsubscribe:** Serve para parar de seguir determinado tópico. Após digitar esse comando o programa irá pedir o tópico que você não quer mais seguir.

**RetrieveTime:** Serve para ver todos os posts feitos que possuem como tema os tópicos que o usuário segue a partir de uma determinada data. Após digitar esse comando o programa irá pedir a data.

**RetrieveTopic** – Semelhante ao retrieveTime, porém só seleciona o tópico que o usuário forneceu. Após entrar com esse comando, o programa irá pedir a data e o tópico.

## NOVOS COMANDOS

**Info:** Exibe o id do programa e sua força, além de identificar o líder.

**Abdicar:** Faz com que o processo abdique da liderança caso seja um líder. Essa função retorna na tela para os outros participantes que foi definido um novo líder.

**SetTime:** Caso seja o líder, essa função é responsável por fazer a sincronização dos relógios.

(Cliente1 que pediu sincronização e abdicou em seguida)

```
-> info
Voce é o processo de numero 1 com força: 16
Voce é o líder
-> setTime
Seu horario: 21:47:46 Horario calculado: 20:53:53
Novo horario: 20:53:53
-> abdicar
-> 
```

(Cliente2 que recebeu a sincronização e logo após foi eleito líder devido ao Cliente1 abdicar)

```
Usuario: da
Horario: 20:00:00
Logado com sucesso !

-> Novo horario: 20:53:53
Um novo lider foi definido !

```

## **POLÍTICAS DE ELEIÇÃO**

Cada cliente recebe um identificador numérico único. Além disso, cada cliente possui um número inteiro que representa sua força. O cliente que tiver a maior força é eleito como líder. A cada cliente novo conectado é feita uma eleição, e o cliente eleito é responsável por sincronizar os relógios. Ao se abdicar da função de líder, o programa automaticamente realiza uma nova eleição.

## **POLÍTICAS DE SINCRONIZAÇÃO**

Após ser definido um líder, ele manda uma mensagem de solicitação de tempo para todos os Clientes. Os clientes respondem enviando seus respectivos horários. Após receber os horários, o líder calcula a média e envia para seus clientes a diferença de tempo entre a média e seu horário. Após receber a resposta, o cliente calcula seu novo horário com base nessa diferença.