

## T2 – Fundamentos de Processamento Paralelo e Distribuído;

Grupo 10: Brenda Billmann, Filipe Dupont  
Bandeira e João Pedro Moreira Antunes

O presente relatório descreve o desenvolvimento de um sistema distribuído que adota o modelo Cliente/Servidor e utiliza tecnologias de Comunicação Remota de Procedimentos (RPC) para simular a criação de contas de clientes e transações bancárias. O objetivo deste projeto é demonstrar a aplicação prática dos conceitos de sistemas distribuídos em um cenário real, onde a eficiência, segurança e confiabilidade são de suma importância.

Inicialmente, procedemos à customização do arquivo `banco.x`, dotando-o de uma estrutura que inclui informações essenciais das contas, tais como o ID (int) e o saldo disponível (double/float). Além disso, definimos um conjunto de funções que abrangem as operações bancárias fundamentais: criação de contas, exclusão de contas, saques, depósitos e consultas de saldo.

Após a devida configuração do arquivo `banco.x`, avançamos para a etapa de geração de código através da execução do comando `"rpcgen banco.x"`. Esse comando é responsável por produzir os arquivos cliente/servidor, que são fundamentais para a comunicação distribuída entre os diversos componentes do sistema. Esses arquivos, embora não editáveis, são necessários para garantir a integração entre os processos cliente e servidor.

Com os arquivos gerados e a estrutura do sistema devidamente estabelecida, procedemos à criação de dois processos clientes distintos, um direcionado à Agência e o outro ao Caixa Automático. Cada um desses clientes é projetado para cumprir funções específicas e oferecer uma experiência de uso otimizada para seus respectivos cenários de operação.

O cliente do Caixa Automático se destina a operações simplificadas, permitindo que os clientes executem ações como saques, depósitos e consultas de saldo. Essa abordagem focada é ideal para atender às necessidades práticas de um caixa automático, onde a simplicidade e eficiência são essenciais.

Por outro lado, o cliente da Agência é uma solução mais abrangente, habilitando mais operações, incluindo as operações presentes no cliente do Caixa Automático. Ele possui a capacidade de criar e excluir contas, o que se alinha com as responsabilidades mais amplas atribuídas a uma agência bancária.

Na interface do cliente, quando os usuários executam ações como saques, depósitos e outras operações, essas ações acionam a função correspondente no servidor. No servidor, as operações bancárias são implementadas e processadas, com base na necessidade específica de cada solicitação.

Essa abordagem de separação entre o cliente e o servidor é fundamental para garantir a organização e eficácia do sistema distribuído. Ela permite que a lógica de negócios

associada às operações bancárias seja centralizada no servidor.

Ao executar uma operação no cliente, o servidor processa a solicitação com base na ação solicitada e nos parâmetros fornecidos. A resposta resultante da operação é então transmitida de volta ao cliente. A natureza dessa resposta depende inteiramente da ação que está sendo executada e pode incluir informações relevantes, como confirmações de sucesso, atualizações de saldo, detalhes de transações ou mensagens de erro, conforme necessário.

Para operações críticas, como a criação ou exclusão de contas, bem como saques e depósitos em um ambiente bancário distribuído, é imperativo implementar uma semântica de execução *exactly-once*. Isso se deve ao fato de que, nessas operações, a precisão e a integridade dos dados são essenciais, e qualquer duplicação acidental ou perda de informações pode ter sérias consequências financeiras e de segurança.

A semântica de execução *exactly-once* garante que cada operação seja realizada uma única vez e apenas uma vez, eliminando o risco de resultados imprecisos devido a repetições não intencionais. Em outras palavras, ao aplicar essa abordagem, podemos ter a certeza de que, após a realização bem-sucedida de uma operação, não haverá tentativas posteriores de executar a mesma operação que possam comprometer a consistência dos dados.

Em um cenário bancário, a criação ou exclusão de contas representa uma ação que não pode ser duplicada ou omitida sem a devida autorização e registro adequado. Da mesma forma, em operações de saque e depósito, a garantia de que um valor específico seja debitado ou creditado apenas uma vez é crucial para evitar discrepâncias nos saldos das contas dos clientes.

Além disso, as operações não-idempotentes são especialmente relevantes para proteger contra problemas potenciais, como cobranças em duplicidade ou perda de registro de transações, que podem prejudicar a confiança dos clientes e a reputação da instituição financeira. Em resumo, a implementação de *exactly-once* nessas operações críticas é um alicerce essencial para a correção e a confiabilidade das operações bancárias em um ambiente distribuído.

Em resumo, este projeto abrangeu desde a personalização da estrutura do arquivo `banco.x` até a criação de clientes específicos para o Caixa Automático e a Agência. A diferenciação entre esses clientes é essencial para atender às necessidades específicas de seus respectivos cenários de uso.

A necessidade de aplicar a semântica de execução *exactly-once* em operações críticas, foi destacada como um importante elemento para garantir a integridade dos dados e a confiabilidade das transações.

No geral, este projeto demonstrou a aplicação prática dos conceitos de um sistema distribuído baseado no modelo Cliente/Servidor com RPC em um contexto de operações bancárias. A capacidade de oferecer funcionalidades adaptáveis, com garantia de correção dos dados, é crucial para garantir o sucesso e a segurança de instituições financeiras em um ambiente distribuído e altamente dinâmico.

## Fotos da execução:

```
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 0 4
Conta Excluida
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 1 3
Você criou a conta[1]
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 1 2
Você possui R$0.000000 na sua conta corrente
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 1 0
Insira o Valor a ser Depositado: R$1000
R$1000.00 Depositado
Você possui R$1000.00 na sua conta corrente
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 1 4
Conta Excluida
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 1 2
Você possui R$0.000000 na sua conta corrente
lipeband@ubuntu:~/Downloads/TP2_FSPD$ S
```

```
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia
Usage: ./bancoclientagencia server id(arg1) função(arg2)
deposito(0) Saque(1) Consulta(2) Criar Conta(3) Excluir Conta(4)
lipeband@ubuntu:~/Downloads/TP2_FSPD$
lipeband@ubuntu:~/Downloads/TP2_FSPD$
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 0 3
Você criou a conta[0]
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 0 0
Insira o Valor a ser Depositado: R$50
R$50.00 Depositado
Você possui R$50.00 na sua conta corrente
lipeband@ubuntu:~/Downloads/TP2_FSPD$
lipeband@ubuntu:~/Downloads/TP2_FSPD$
lipeband@ubuntu:~/Downloads/TP2_FSPD$ ./bancoclientagencia localhost 0 1
30
R$30.000000 Retirado
Você possui R$20.000000 na sua conta corrente
lipeband@ubuntu:~/Downloads/TP2_FSPD$
```