



# Computação Paralela e Distribuída – aula prática nº 9

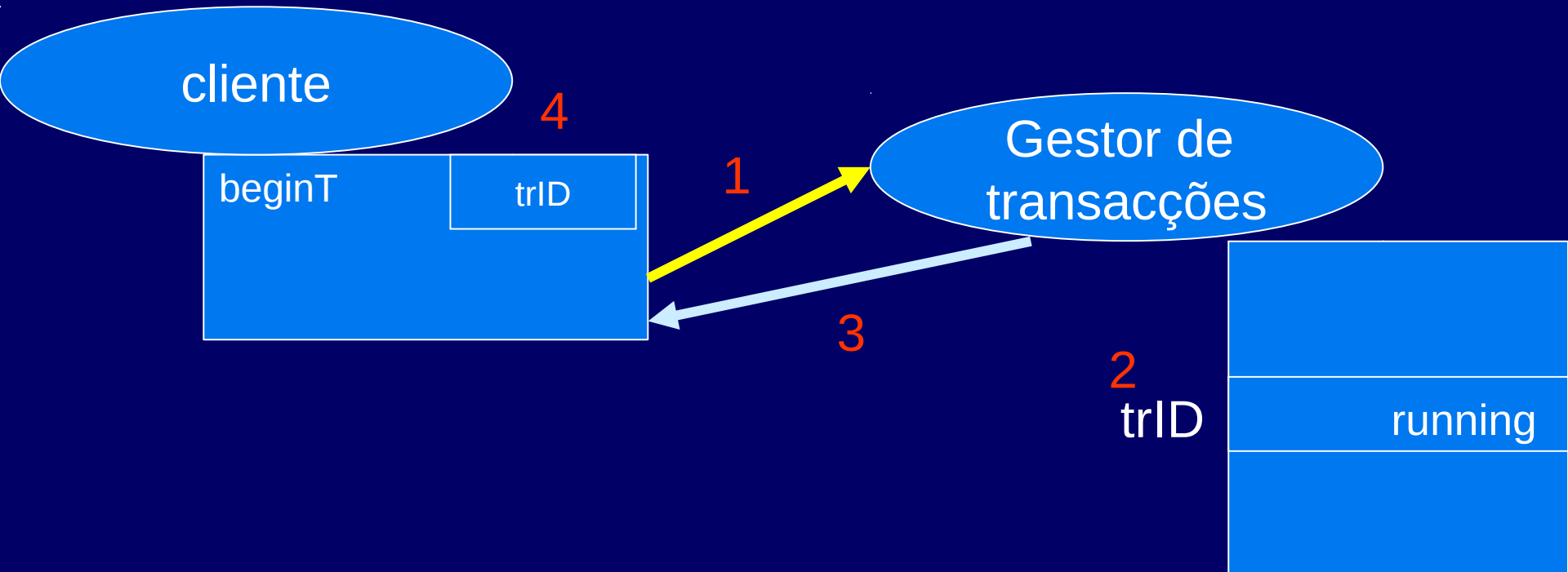
Construção de uma versão simplificada de um monitor transaccional para contas bancárias

- Gestor de transacções

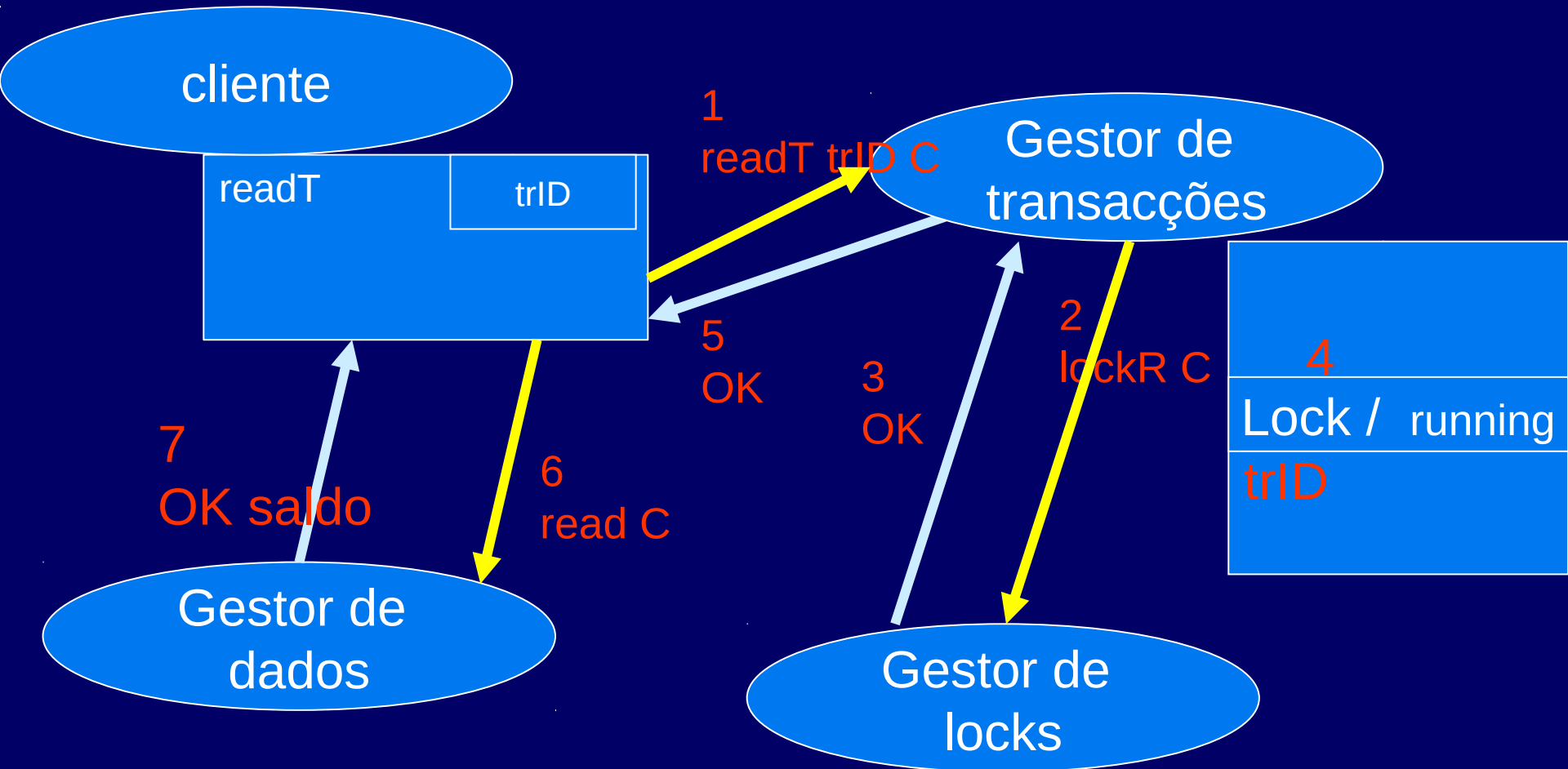
# Operações de gestão de transacções

- ♦ **int beginT** ( void) Inicia uma transacção. Devolve o identificador da transacção em caso de êxito e -1 em caso de erro.
- ♦ **int commitT** ( int tr\_id) Termina a transacção identificado por trid fazendo **commit**. Devolve 0 se a operação foi realizada com êxito ou -1 em caso de erro.
- ♦ **int abortT** ( int tr\_id) Termina a transacção identificado por trid fazendo **abort**. Devolve 0 se a operação foi realizada com êxito ou -1 em caso de erro.
- ♦ **int readT** ( int nConta) Realiza uma operação de leitura de um saldo da conta nConta. Contém a realização de um lock de leitura sobre a conta nConta. Se o lock não for conseguido por timeout a operação retorna -1
- ♦ **int writeT** ( int nConta, int novoSaldo) Realiza uma operação de escrita de um saldo da conta nConta. Contém a realização de um lock de escrita sobre a conta nConta. Se o lock não for conseguido por timeout a operação retorna -1

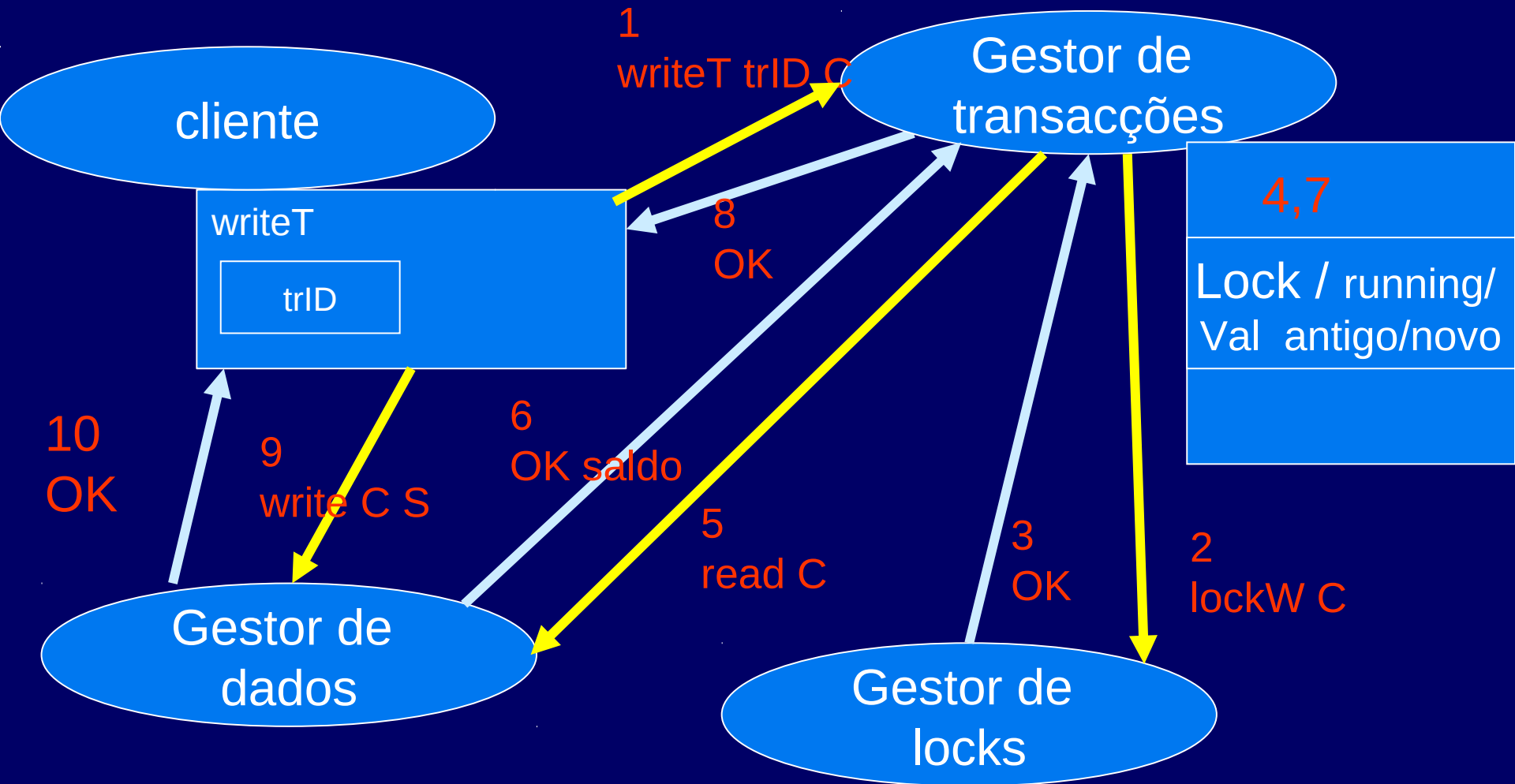
# beginT



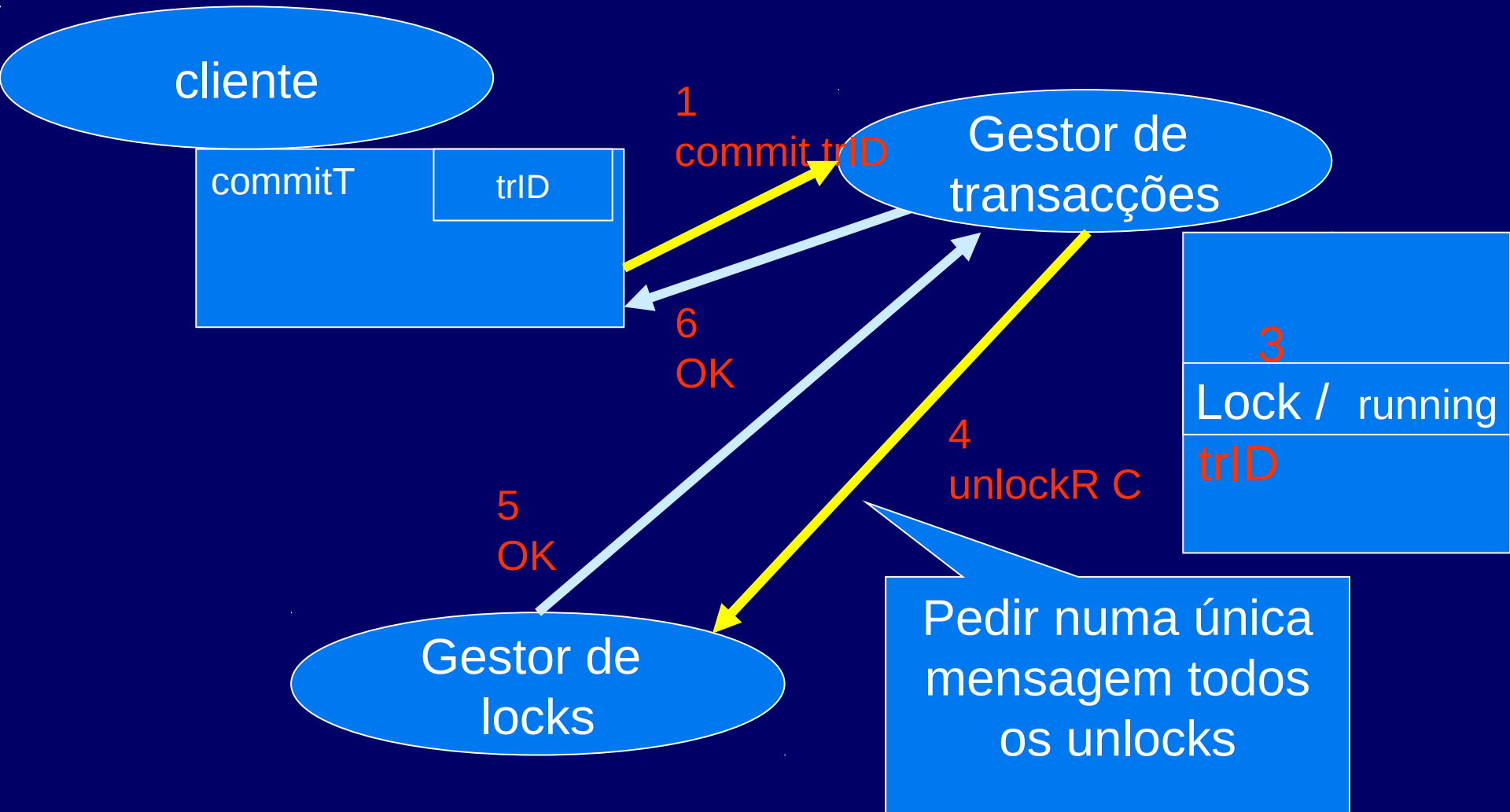
# readT



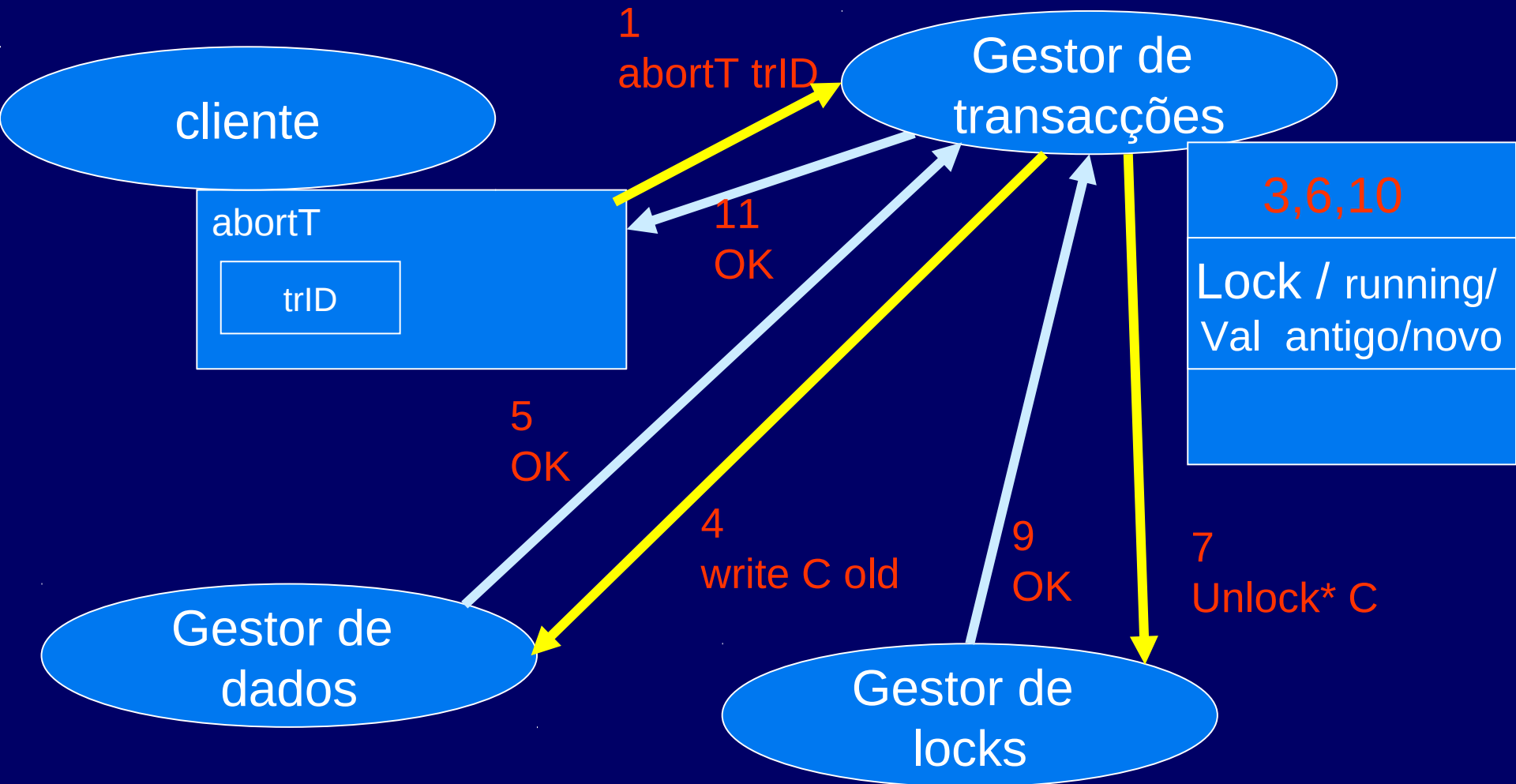
# writeT



# commitT



# abortT



# Teste do sistema

---

- ◆ Pretende-se que teste o sistema através de clientes que invocam as operações atrás definidas
- ◆ Verificar as operações de commit e abort
- ◆ Verificar que deixam de existir as anomalias
  - actualização perdida: C2 lê, C1 escreve, C2 escreve
  - leitura suja: C2 escreve, C1 Lê, C2 escreve
  - leitura irrepetível: C1 lê, C2 escreve, C1 lê