

TRATAMENTO DE FALHAS: RMI

No RMI, uma invocação remota pode falhar devido a erros de comunicação. Necessário apanhar exceção

```
String serverurl = ...;    //e.g.: "///10.123.23.54/myServer"  
IServer server = (IServer)Naming.lookup( serverurl);
```

```
try {  
    server.someop( someargs);  
    // invocação bem sucedida  
} catch( java.rmi.RemoteException e) {  
    // processar erro  
}
```

TRATAMENTO DE FALHAS: O QUE FAZER NO CLIENTE?

Tratamento das falhas de comunicação vai depender da aplicação

Aproximação simples quando existe apenas um servidor:

- Esperar algum período de tempo e tentar contactar o servidor novamente

- Ao fim de algumas tentativas, desistir

TRATAMENTO DE FALHAS: O QUE FAZER?

```
boolean executed = false;
```

```
for( int i = 0; ! executed && i < 3; i++) { // número de tentativas de contacto
    try {
        server.someop( someargs);
        executed = true;
    } catch( java.rmi.RemoteException e) {
        if( i < 2) {
            try { // esperar algum tempo
                Thread.sleep( 5000);
            } catch( InterruptedException e1) {
                // do nothing
            }
        }
    }
}
```

HEARBEAT

Um processo A envia periodicamente uma mensagem a notificar que está ativo

Um processo B que recebe esta mensagem regista o momento em que recebeu a mensagem.

O processo B considera que um processo A está inativo caso não tenha recebido um heartbeat há tempo significativo (deve considerar-se que alguma mensagens se podem perder)

KEEP-ALIVE

Um processo A envia periodicamente um pedido a um processo B a perguntar se ainda se encontra ativo. Se o processo B não responde várias vezes seguidas assume-se que está inativo.