

TRATAMENTO DE FALHAS: RMI

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

```
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: WEB SERVICES

Nos Web Services, uma invocação remota pode falhar devido a erros de comunicação. Necessário apanhar exceção

```
FileServerImplWSService service = new FileServerImplWSService();
FileServerImplWS server = service.getFileServerImplWSPort();

try {
    server.someop( someargs);
    // invocação bem sucedida
} catch( javax.xml.ws.WebServiceException e) {
    // processar erro
}
```

TRATAMENTO DE FALHAS: O QUE FAZER?

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

CONCORRÊNCIA NO SERVIDOR

Os servidores de RMI e web services tratam múltiplas invocações concorrentemente em diferentes threads

É necessário coordenar a execução dos múltiplos threads

PROTEGENDO ACESSO CONCORRENTE

```
class MyServer {  
    List<String> l1;  
  
    public synchronized int countStart( String s) {  
        Iterator<String> it = l1.iterator();  
        int n = 0;  
        while( it.hasNext() ) {  
            if( it.next().startsWith( s))  
                n++;  
        }  
        return n;  
    }  
    public synchronized void add( String s) {  
        l1.add( s);  
    }  
}
```

PROTEGENDO ACESSO CONCORRENTE

```
class MyServer {
    List<String> l1;

    public int countStart( String s) {
        int n = 0;
        synchronized( l1) {
            Iterator<String> it = l1.iterator();
            while( it.hasNext() ) {
                if( it.next().startsWith( s))
                    n++;
            }
        }
        return n;
    }

    public void add( String s) {
        synchronized( l1) {
            l1.add( s);
        }
    }
}
```