

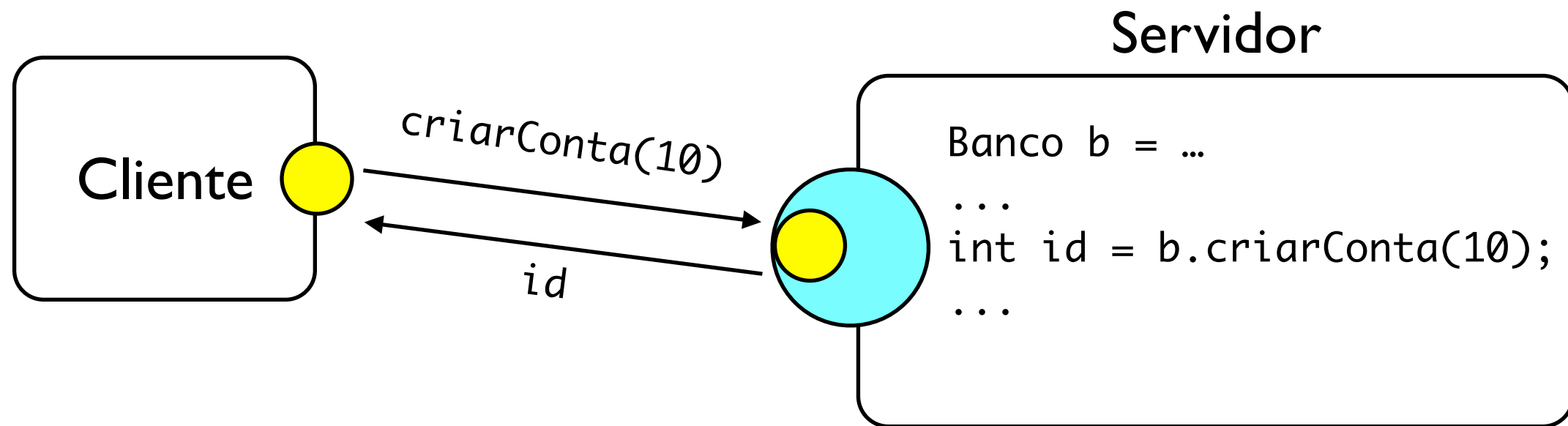
# Servidores Multithreaded com Estado Partilhado

Sistemas Distribuídos

# Exercícios

1) Implemente um servidor que ofereça a funcionalidade do “Banco”, e que aceite a conexão simultânea de múltiplos clientes.

```
public interface InterfaceBanco {  
    int criarConta(double saldo);  
    double fecharConta(int id) throws ContaInvalida;  
    double consultar(int id) throws ContaInvalida;  
    double consultarTotal(int[] contasInput);  
    void levantar(int id, double valor) throws SaldoInsuficiente, ContaInvalida;  
    void depositar(int id, double valor) throws ContaInvalida;  
    void transferir(int conta_origem, int conta_destino, double valor) throws  
ContaInvalida, SaldoInsuficiente;  
}
```



**Nota:** Para serializar variáveis e objectos de diferentes tipos (exemplo: int, double,...) no canal de comunicação de forma prática usar classes **ObjectOutputStream** e **ObjectInputStream**.

```
ObjectOutputStream out = new ObjectOutputStream (socket.getOutputStream());  
ObjectInputStream in = new ObjectInputStream (socket.getInputStream());
```

```
out.writeInt(1);  
out.writeChars("ola")  
out.flush();  
...  
in.readDouble();  
in.readInt();
```

# Exercícios

2) Implemente um servidor de conversação que aceite a conexão de múltiplos clientes. Cada mensagem enviada por um cliente é difundida por todos os clientes ligados.

**Sugestão:** permitir a cada cliente registar o seu nickname, detectando repetidos

