



```
#include "ContaI.h"
...
CORBA::ORB_var orb = CORBA::ORB::_nil();
...
int main(int argc, char* argv[])
{
    try { // exceções CORBA podem ocorrer
        orb = CORBA::ORB_init(argc, argv, "ORB");
```



```
// Ativa Root POA
CORBA::Object_ptr obj;
obj = orb->resolve_initial_references("RootPOA");
PortableServer::POA_var poa = PortableServer::POA::_narrow(obj);
PortableServer::POAManager_var ger = poa->the_POAManager();
ger->activate();
// Instancia servant
Conta_i conta_i;
// registra servant no POA + obtém ref
Conta_var conta = conta_i._this();
```



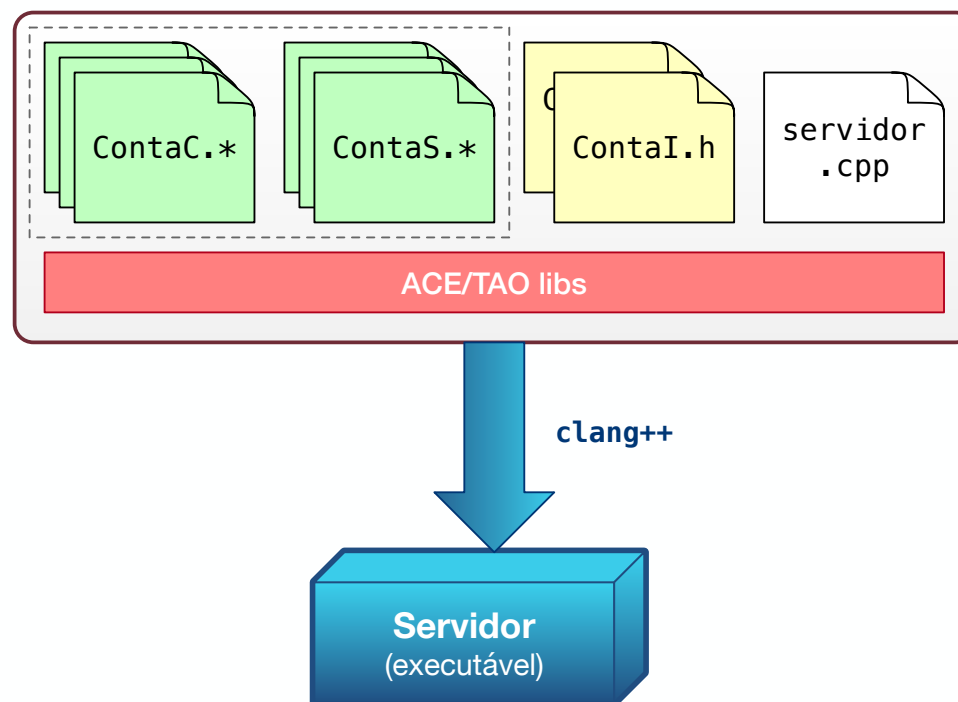
```
// Exporta IOR (salva em arquivo aqui)  
CORBA::String_var ior = orb->object_to_string(conta.in());  
ofstream arqior("conta.ior");  
arqior << ior << endl;  
arqior.close();  
// Bloqueia aguardando requisições  
orb->run();  
// Cleanup  
poa->destroy(true,true);  
orb->destroy(true);
```



```
// fim do bloco try-catch + fim do programa
} catch (CORBA::Exception& e) {
    cerr << "Exceção CORBA: " << e << endl;
}
return 0;
}
```



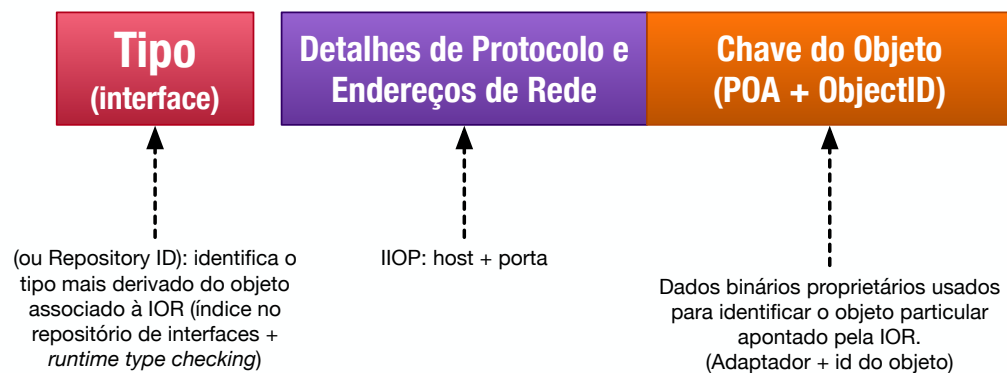
## Compilação do Servidor (Conta C++ ACE/TAO)





# Referências Transientes

Objetos Distribuídos - CORBA





## Binding de Referências Transientes

### CLIENTE

IOR

IDL:Conta:1.0

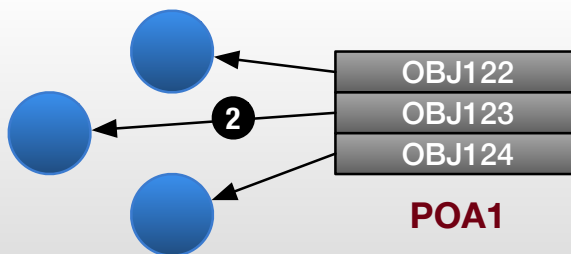
espec:1512

POA1:OBJ123

1 **op()**

SERVIDOR @ espec:1512

resposta de op()



3





## *Binding* de Referências Transientes

1. ORB cliente abre conexão com *host* + porta especificadas na IOR e envia requisição:

- Tamanho da mensagem
- Identificador da requisição (único)
- Chave do objeto (nome do POA + objeto)
- Operações + parâmetros

(o cliente aguarda resposta na conexão)



## Binding de Referências Transientes

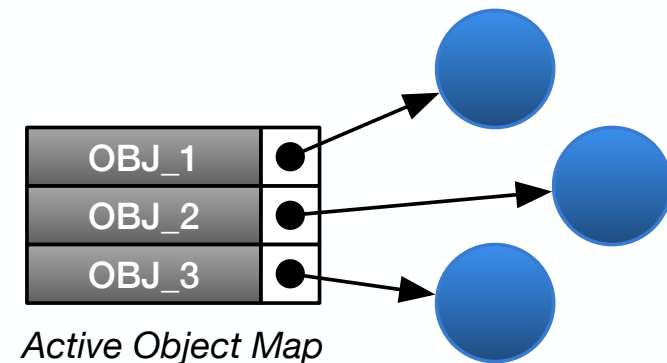
2. Servidor usa chave do objeto para localizar o *servant* adequado:

- **Active Object Map:**

- ✦ *Para cada adaptador*

- ✦ *Mapa:*

- nome  $\Leftrightarrow$  ponteiro para servant



3. Servidor envia resposta ao cliente:

- usa mesmo identificador da requisição