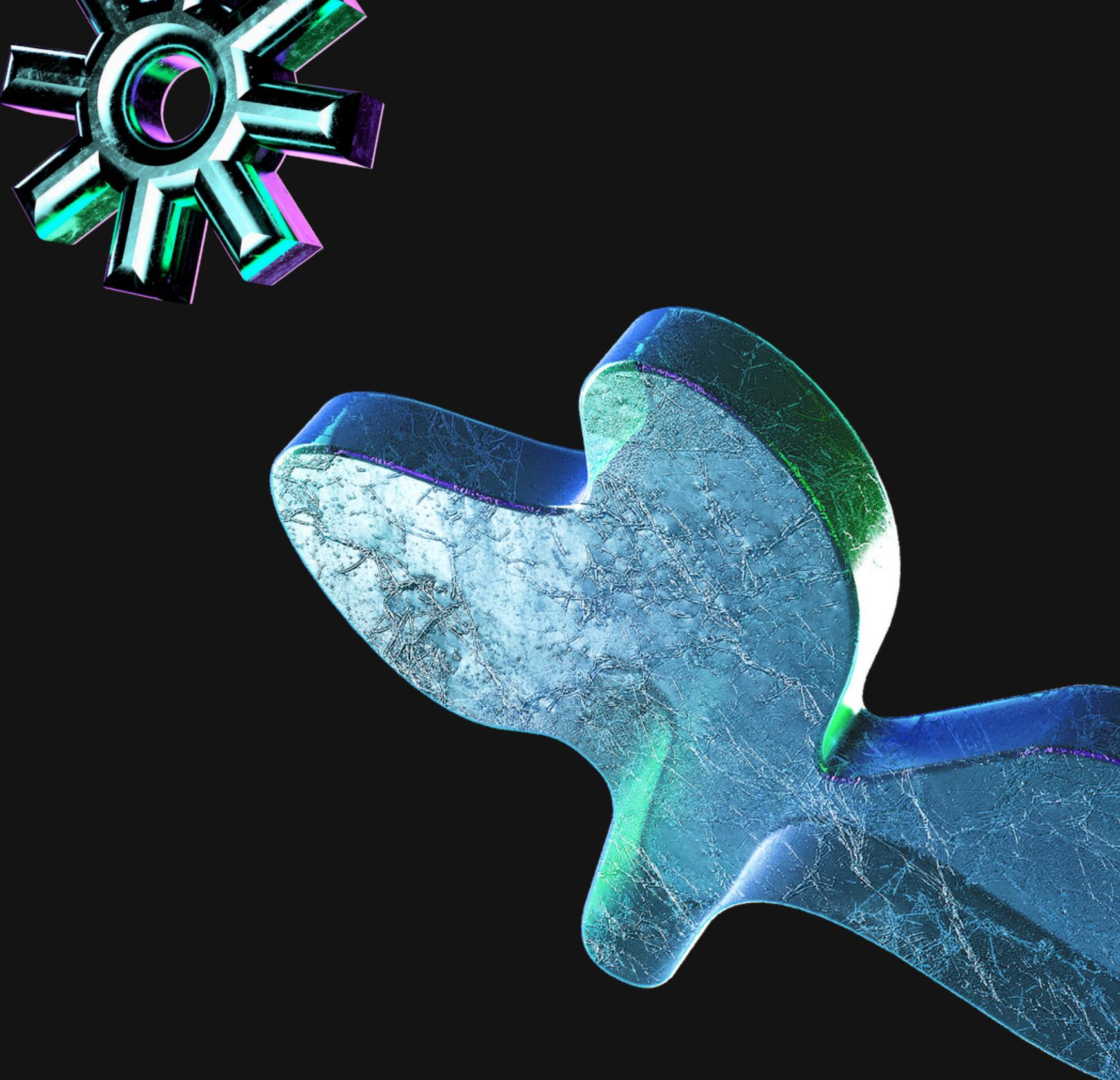
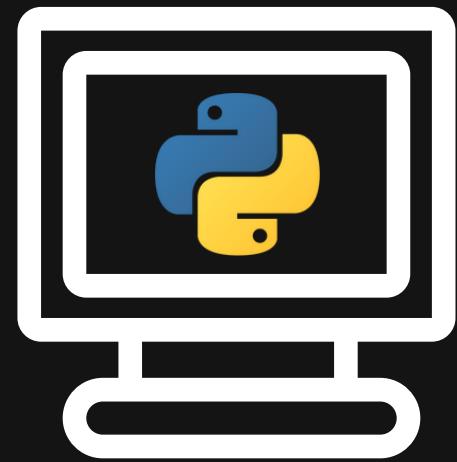
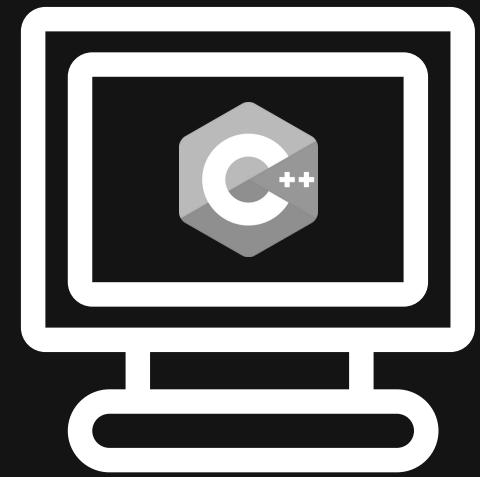


**COM242 – SISTEMAS DISTRIBUÍDOS**

# **CORBA**

Grupo 4: Vitor Hermeto, Antonio Gomes

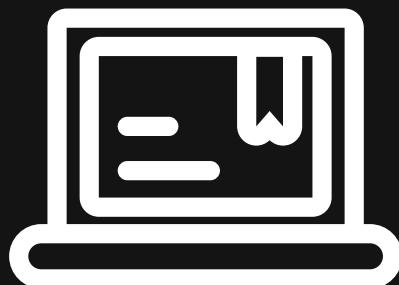




# O que é **CORBA?**



CORBA



CORBA (Common Object Request Broker Architecture) é uma tecnologia de middleware para sistemas distribuídos que permite que objetos em diferentes sistemas se comuniquem uns com os outros independentemente da linguagem de programação ou plataforma utilizada.

# Arquitetura de Middleware

O middleware é uma solução ideal para sistemas distribuídos, onde os aplicativos são executados em diferentes computadores ou sistemas operacionais. Ele fornece uma plataforma comum para comunicação, permitindo que os aplicativos possam compartilhar informações e recursos de maneira mais eficiente.

# Middleware de objetos distribuídos

---

- O encapsulamento inerente
- A propriedade relacionada da abstração de dados
- soluções mais dinâmicas e extensíveis



# Middleware baseado em componentes

---

- Dependências implícitas
- Complexidade de programação
- Falta de separação de aspectos relacionados à distribuição

# CORBA

Middleware de objetos distribuídos

INDEPENDÊNCIA DE  
LINGUAGEM E  
PLATAFORMA

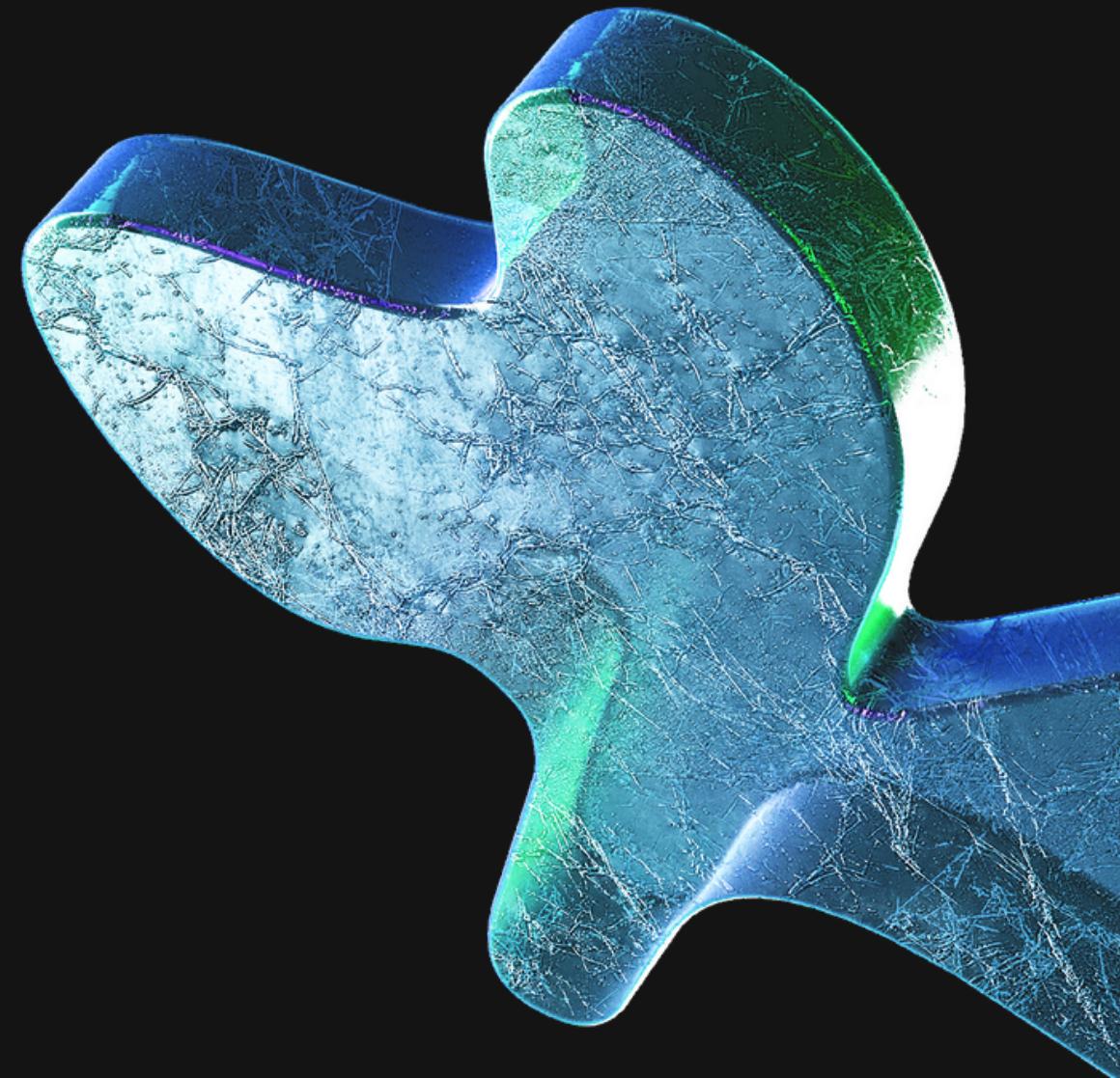
---

ORIENTAÇÃO A  
OBJETO

---

SUPORTE À EVOLUÇÃO

---



# Arquitetura

Cliente: chamada de método remoto

Stub: objeto do lado do cliente que se comunica com o ORB

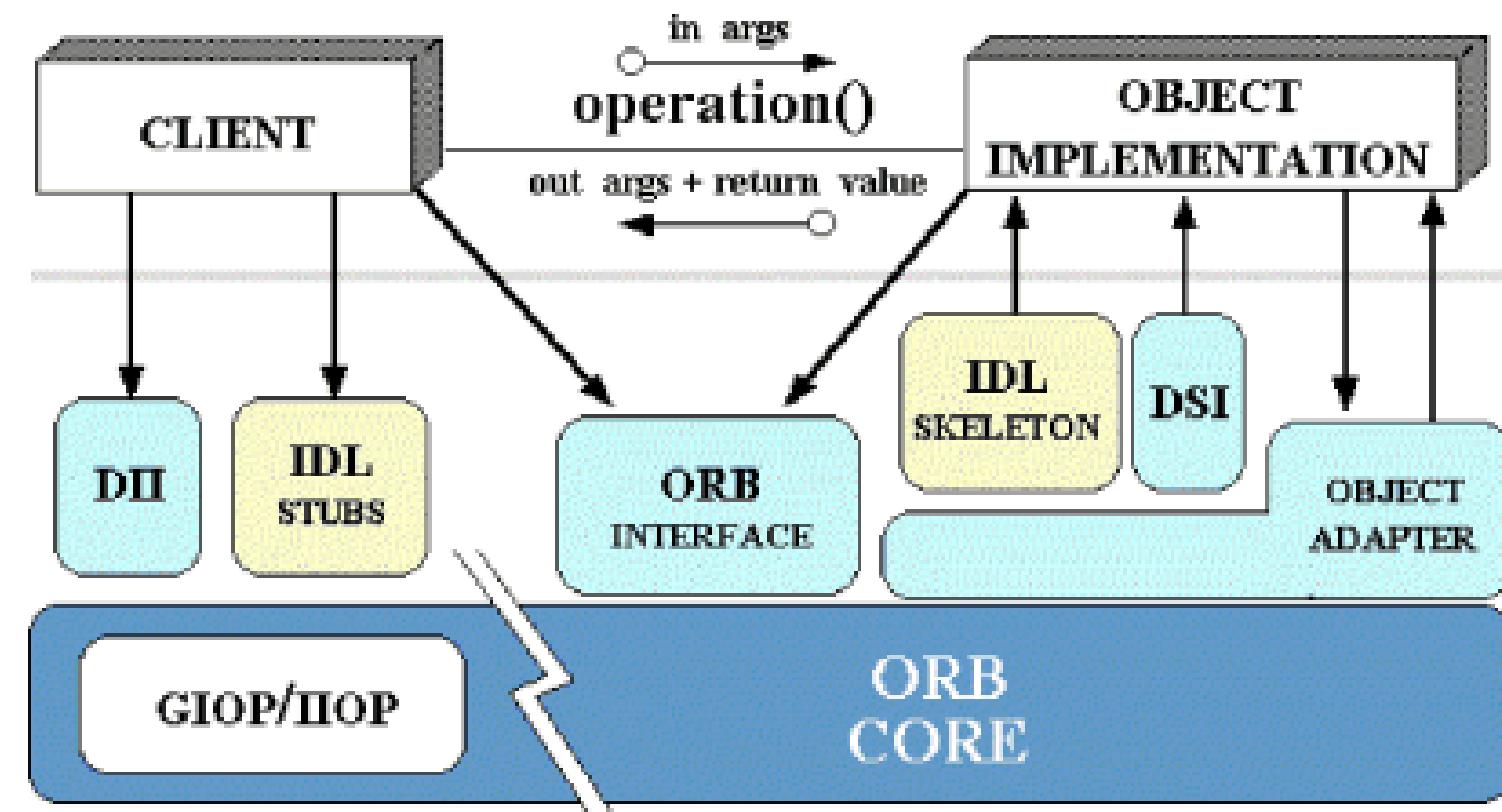
ORB: intermediário na comunicação entre o objeto remoto e o cliente

Skeleton: objeto do lado do servidor que se comunica com o ORB

Objeto Remoto: objeto remoto que o cliente deseja acessar

IDL: Interface Description Language

## CORBA Architecture



Software Architectures

OMG 1993

# Interfaces em CORBA

Fornece recursos para a definição de módulos, interfaces, tipos, atributos e assinaturas de método.

A gramática da IDL é um subconjunto da linguagem C++ ANSI, com construções adicionais para suportar assinaturas de método.

INDEPENDÊNCIA DE LINGUAGEM

INDEPENDÊNCIA DE PLATAFORMA

SUPORTE À EVOLUÇÃO DAS  
INTERFACES

SIMPLIFICAÇÃO DA COMUNICAÇÃO

# Serviços do CORBA



**conjunto de serviços padronizados e distribuídos que podem ser utilizados para gerenciar diferentes aspectos de um sistema distribuído.**

## SERVIÇO DE NOMEAÇÃO

---

fornecer uma maneira de associar nomes de objetos CORBA com os objetos reais, permitindo que os clientes localizem objetos em um sistema distribuído.

## SERVIÇO DE TRANSAÇÃO

---

permite que as transações sejam coordenadas em um sistema distribuído, garantindo que todas as operações sejam concluídas com sucesso ou revertidas em caso de falha.

## SERVIÇO DE SEGURANÇA

---

fornecer mecanismos para autenticar e autorizar usuários em um sistema distribuído, garantindo a privacidade e a segurança das informações transmitidas.

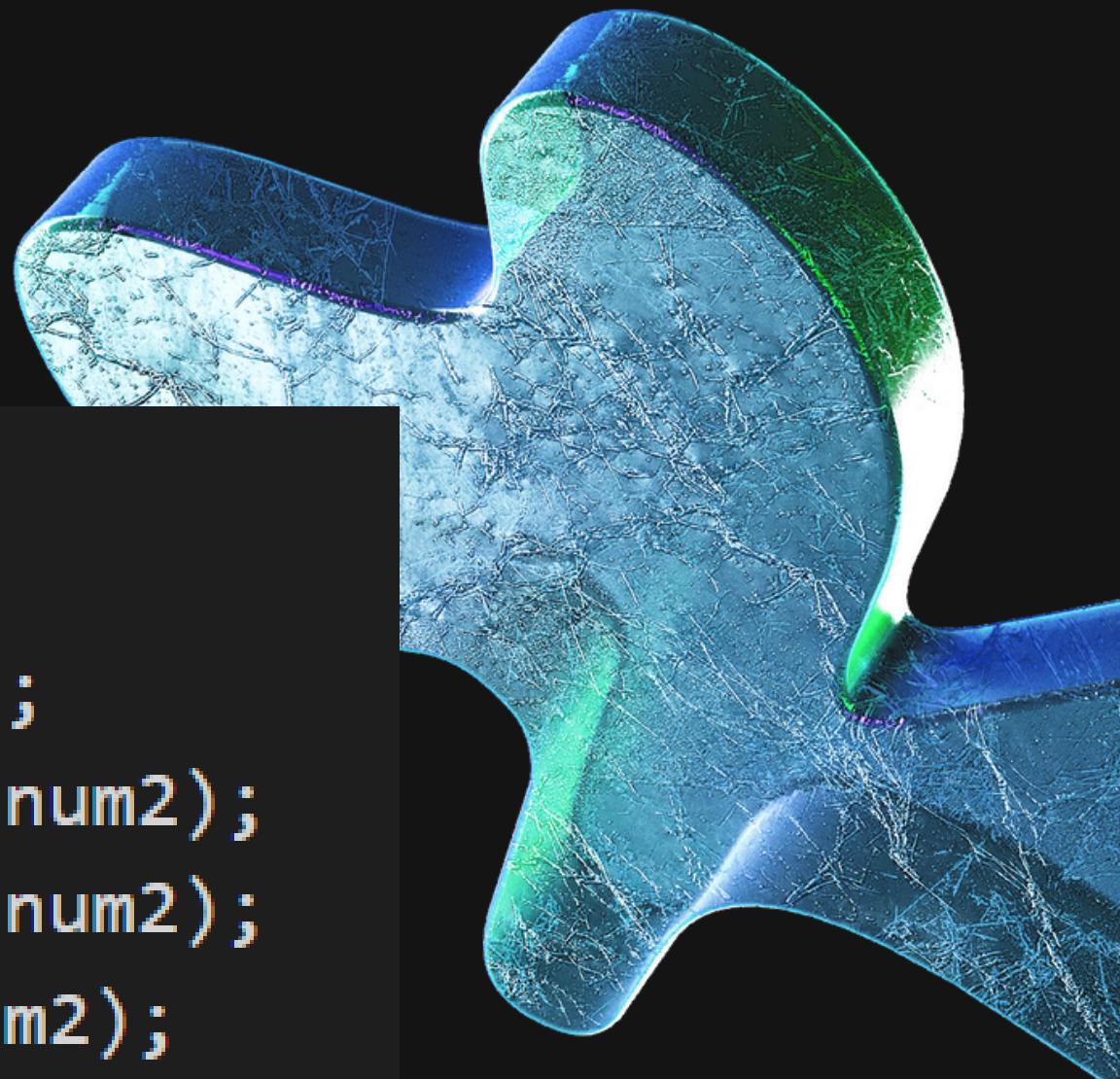
# IMPLEMENTAÇÃO

---



# Calculator.idl

```
module Calculator {  
    interface CalculatorInterface {  
        float add(in float num1, in float num2);  
        float subtract(in float num1, in float num2);  
        float multiply(in float num1, in float num2);  
        float divide(in float num1, in float num2);  
    };  
};|
```



# CalculatorImpl.java

```
public class CalculatorImpl extends CalculatorInterfacePOA {
    public float add(float num1, float num2) {
        return num1 + num2;
    }

    public float subtract(float num1, float num2) {
        return num1 - num2;
    }

    public float multiply(float num1, float num2) {
        return num1 * num2;
    }

    public float divide(float num1, float num2) {
        return num1 / num2;
    };
}
```



# CalculatorServer.java

```
public static void main(String[] args) {
    try {
        // Cria e inicializa o ORB
        ORB orb = ORB.init(args, null);

        // Cria a instância da calculadora
        CalculatorImpl calculatorImpl = new CalculatorImpl();

        // Obtém uma referência para o RootPOA e ativa o POAManager
        POA rootPOA = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
        rootPOA.the_POAManager().activate();

        // Cria o objeto CORBA e registra-o no serviço de nomes
        org.omg.CORBA.Object ref = rootPOA.servant_to_reference(calculatorImpl);
        CalculatorInterface href = CalculatorInterfaceHelper.narrow(ref);

        org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
        NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

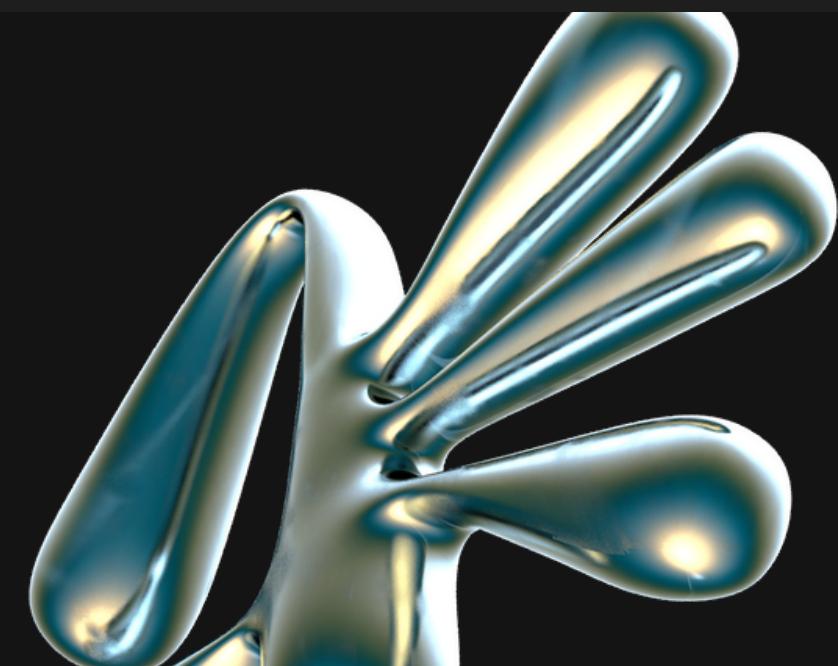
        NameComponent path[] = ncRef.to_name("Calculator");
        ncRef.rebind(path, href);
```

# CalculatorClient.java

```
public class CalculatorClient {
    public static void main(String[] args) {
        try {
            // Cria e inicializa o ORB
            ORB orb = ORB.init(args, null);

            // Obtém uma referência para o serviço de nomes
            org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
            NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

            // Obtém uma referência para a calculadora
            NameComponent path[] = ncRef.to_name("Calculator");
            CalculatorInterface calculator = CalculatorInterfaceHelper.narrow(ncRef.resolve(path));
```





# Ficou alguma dúvida?

## Referências:

George Coulouris, Jean D, Tim K e Gordon B. Sistemas Distribuídos:  
Conceitos e Projeto. 5<sup>a</sup> ed. Bookman. 2013

MCHALE, Ciaran. CORBA Explained Simply. 2007.