

1. INTRODUÇÃO

O presente relatório apresenta a atividade de Programação Guiada desenvolvida no âmbito da disciplina *Sistemas Operacionais II*. A atividade teve como objetivo demonstrar o desenvolvimento e a implementação de um serviço de Remote Procedure Call (RPC) utilizando o protocolo gRPC com a linguagem Python 3.

Este relatório documenta o processo de desenvolvimento da aplicação, incluindo capturas de tela que ilustram o funcionamento do serviço, bem como a descrição de algumas dificuldades enfrentadas ao longo do desenvolvimento..

2. RESULTADOS OBTIDOS

```
}

message MultiplyRequest {
    double a = 1;
    double b = 2;
}

message MultiplyReply {
    double result = 1;
}

message ReturnBiggerRequest {
    double a = 1;
    double b = 2;
    double c = 3;
}

message ReturnBiggerReply {
    double result = 1;
}

message QuotientRequest {
    double numerator = 1;
    double denominator = 2;
}

message QuotientReply {
    double quotient = 1;
    double rest = 2;
}

service Calculator {
    rpc Sum (SumRequest) returns (SumReply);
    rpc Multiply (MultiplyRequest) returns (MultiplyReply);
    rpc ReturnBigger (ReturnBiggerRequest) returns (ReturnBiggerReply);
    rpc Quotient (QuotientRequest) returns (QuotientReply);
}
```

Figura 1 - Captura de tela da IDL.

Fonte: elaborado pelos autores.

A Figura 1 apresenta o arquivo de Interface, contendo todas as operações solicitadas.

```
class Calculator(CalculatorService):  
    def Sum(self, request: SumRequest, context: ServicerContext) -> SumReply:  
        return SumReply(s=request.a + request.b)  
    def Multiply(self, request: MultiplyRequest, context: ServicerContext) -> MultiplyReply:  
        return MultiplyReply(result=request.a * request.b)  
    def ReturnBigger(self, request: ReturnBiggerRequest, context: ServicerContext) -> ReturnBiggerReply:  
        return ReturnBiggerReply(result=max(request.a, request.b, request.c))  
    def Quotient(self, request: QuotientRequest, context: ServicerContext) -> QuotientReply:  
        return QuotientReply(quotient=int(request.numerator / request.denominator), rest=request.numerator % request.denominator)
```

Figura 2 - Classe Calculator.
Fonte: elaborado pelos autores

Para a implementação do código, utilizamos como base a operação de soma, realizando apenas as alterações necessárias para cada operação, conforme pode ser observado no código.

```
pg2 python3 -m pytest  
===== test session starts =====  
platform linux -- Python 3.10.12, pytest-8.3.3, pluggy-1.5.0  
rootdir: /home/jpcorso/faculdade/2024-2/ap_sisop2/pg2  
collected 4 items  
  
calculator_integration_test.py .... [100%]  
  
===== 4 passed in 0.07s =====  
pg2
```

Figura 3 - Resultado dos testes para todas as operações.
Fonte: elaborado pelos autores

Conforme solicitado na atividade, realizamos testes de integração para cada uma das operações implementadas na calculadora. Todos os testes foram bem-sucedidos.

3. DIFICULDADES ENCONTRADAS

- a. Compreensão da Interface IDL.
- b. Geração de arquivos gRPC.
- c. Entendimento do funcionamento do gRPC.