



Calculadora de notas semestrais Utilizando gRPC

Elder Fernandes Maia
Jeremias Bezerra Lucas

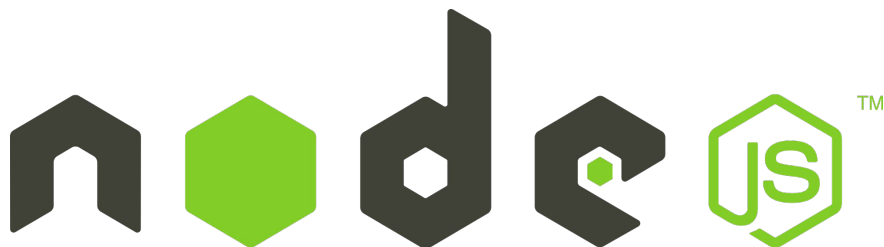


OBJETIVOS

- Calcular a média de um aluno com 2 notas (peso 2 e 3)
- Calcular a nota mínima necessária na prova final para aprovação
- Se a média for < 6 , calcular a nota necessária na prova final
- Comunicação entre cliente e servidor com gRPC
- Cliente e servidor em linguagens diferentes



Tecnologias Utilizadas



Client.js



Server.py



notas.proto

```
syntax = "proto3";

service NotaService {
    rpc CalcularNota (NotaRequest) returns (NotaResponse);
}

message NotaRequest {
    double nota1 = 1;
    double nota2 = 2;
}

message NotaResponse {
    double media = 1;
    bool aprovado = 2;
    bool reprovado = 3;
    double nota_final_necessaria = 4;
}
```



Server.py

```
from concurrent import futures
import grpc
import notas_pb2
import notas_pb2_grpc

class NotaService(notas_pb2_grpc.NotaServiceServicer):
    def CalcularNota(self, request, context):
        nota1 = request.nota1
        nota2 = request.nota2

        media = (nota1 * 2 + nota2 * 3) / 5
        aprovado = media >= 6
        nota_final_necessaria = max(0.0, 12 - media)
        reprovado = not aprovado and nota_final_necessaria > 10.0

        return notas_pb2.NotaResponse(
            media=media,
            aprovado=aprovado,
            reprovado=reprovado,
            nota_final_necessaria=nota_final_necessaria
        )
```

```
def serve():
    server = grpc.server(futures.ThreadPoolExecutor(max_workers=10))
    notas_pb2_grpc.add_NotaServiceServicer_to_server(NotaService(), server)
    server.add_insecure_port('[::]:50051')
    server.start()
    print("Servidor rodando na porta 50051...")
    server.wait_for_termination()

if __name__ == '__main__':
    serve()
```



Client.js

```
const grpc = require('@grpc/grpc-js');
const protoLoader = require('@grpc/proto-loader');
const readline = require('readline');

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

const packageDefinition = protoLoader.loadSync('../proto/notes.proto', {
  keepCase: true,
  longs: String,
  enums: String,
  defaults: true,
  oneofs: true
});

const grpcObject = grpc.loadPackageDefinition(packageDefinition);
const NotaService = grpcObject.NotaService;
const cliente = new NotaService('localhost:50051', grpc.credentials.createInsecure());

async function obterNotas() {
  return new Promise((resolve) => {
    rl.question('Digite a primeira nota (peso 2): ', (nota1) => {
      rl.question('Digite a segunda nota (peso 3): ', (nota2) => {
        resolve({
          nota1: parseFloat(nota1),
          nota2: parseFloat(nota2)
        });
      });
    });
  });
}
```

1. Importações

grpc: Biblioteca para comunicação com servidor gRPC

protoLoader: Converte arquivo .proto para JavaScript

readline: Pega input do usuário no terminal

2. Configuração Inicial

Cria interface para ler/escrever no terminal

Carrega o contrato do serviço (**notes.proto**)

3. Cria Cliente gRPC

Conecta ao servidor em localhost:50051

Client.js

```
async function main() {
  console.log('=== Calculadora de Notas IFBN ===');

  const { nota1, nota2 } = await obterNotas();

  if (isNaN(nota1) || isNaN(nota2)) {
    console.log('Por favor, informe valores numéricos válidos');
    rl.close();
    return;
  }

  console.log('\nProcessando...\n');

  cliente.CalcularNota({ nota1, nota2 }, (err, response) => {
    rl.close();

    if (err) {
      console.error("Erro:", err);
      return;
    }

    console.log("Resultado:");
    console.log("Nota 1:", nota1);
    console.log("Nota 2:", nota2);
    console.log("Média:", response.media.toFixed(2));

    const notaFinal = response.nota_final_necessaria ?? response.notaFinalNecessaria ?? 0;

    if (response.reprovado) {
      console.log("Situação: ❌ Reprovado direto");
      console.log("Motivo: Precisar de", notaFinal.toFixed(2), "na final");
    } else if (response.aprovado) {
      console.log("Situação: ✅ Aprovado");
    } else {
      console.log("Situação: ⚠️ Prova final");
      console.log("Nota necessária na final:", notaFinal.toFixed(2));
    }
  });
}

main();
```

4. Lógica Principal

Pede 2 notas ao usuário

Valida se são números

Envia para o servidor calcular

5. Processa Resposta

Mostra:

Notas digitadas

Média calculada

Situação (Aprovado/Reprovado/Prova Final)

Nota necessária na final (se aplicável)

Fluxo Simplificado:

Usuário digita notas → Cliente envia para servidor → Servidor calcula →

Cliente mostra resultado



OBRIGADO!