



Campus Uberlândia

**CURSO DE GRADUAÇÃO EM  
ENGENHARIA DA COMPUTAÇÃO**

**TRABALHO DE ALGORITMO E ESTRUTURA DE DADOS**

**ALUNO(S): GUILHERME BARBOSA, LUCAS SILVA, RHANDAL REIS, IRYS  
NATHYELLY e NIKOLAS FERNANDES**

**UBERLÂNDIA – MG**

**02/06/2025**

## Sumário

1. INTRODUÇÃO .....	4
1.1 CONTEXTO ACADÊMICO.....	4
1.2 ESCOPO DO PROJETO .....	4
2. OBJETIVOS.....	4
2.1 OBJETIVO GERAL .....	4
2.2 OBJETIVOS ESPECÍFICOS .....	4
3. AMBIENTE DE DESENVOLVIMENTO.....	4
3.1 PLATAFORMA DE DESENVOLVIMENTO .....	4
3.2 CARACTERÍSTICAS DO VISUALG.....	5
4. REQUISITOS DO SISTEMA .....	5
4.1 REQUISITOS FUNCIONAIS .....	5
4.2 REQUISITOS NÃO FUNCIONAIS .....	6
4.3 LIMITAÇÕES RECONHECIDAS .....	6
5. ARQUITETURA DO SISTEMA.....	7
5.1 ESTRUTURA MODULAR ADAPTADA AO VISUALG .....	7
5.2 PADRÃO DE ORGANIZAÇÃO .....	7
6. ESPECIFICAÇÃO FUNCIONAL .....	8
6.1 INTERFACE DO USUÁRIO.....	8
6.2 FLUXO DE INTERAÇÃO .....	8
6.3 OPERAÇÕES MATEMÁTICAS DISPONÍVEIS.....	8
7. DOCUMENTAÇÃO DE CÓDIGO.....	9
7.1 ESTRUTURA DE DADOS .....	9
7.2 MÓDULOS DE INTERFACE.....	9
7.3 MÓDULOS DE CÁLCULO MATEMÁTICO .....	10
8. FLUXO DE EXECUÇÃO.....	11

8.1 ALGORITMO PRINCIPAL DETALHADO .....	11
8.2 ESTADOS DO SISTEMA .....	12
9. TRATAMENTO DE ERROS.....	12
9.1 ESTRATÉGIA DE TRATAMENTO .....	12
9.2 CATEGORIAS DE ERRO.....	12
9.3 LIMITAÇÕES DO TRATAMENTO .....	12
9.4 MELHORIAS POSSÍVEIS (Limitadas pelo VisuAlg).....	13
10. CASOS DE TESTE.....	13
10.1 METODOLOGIA DE TESTE.....	13
10.2 TESTES FUNCIONAIS BÁSICOS.....	13
10.3 TESTES DE VALIDAÇÃO E ERRO .....	14
10.4 TESTES DE CASOS EXTREMOS.....	14
10.5 TESTES DE USABILIDADE .....	14
11. MANUAL DE USO.....	14
11.1 INSTALAÇÃO E EXECUÇÃO .....	14
11.2 GUIA DE OPERAÇÃO.....	15
11.3 EXEMPLOS PRÁTICOS.....	15
11.4 SOLUÇÃO DE PROBLEMAS COMUNS.....	16
11.5 DICAS DE USO .....	16
12. CONSIDERAÇÕES FINAIS.....	17
12.1 OBJETIVOS ALCANÇADOS.....	17
12.2 CONTRIBUIÇÕES EDUCACIONAIS.....	17
12.3 LIMITAÇÕES IDENTIFICADAS E CONTEXTUALIZADAS .....	17
12.4 POSSIBILIDADES DE EXPANSÃO.....	18
12.5 APLICABILIDADE EDUCACIONAL.....	18
12.6 IMPACTO E RELEVÂNCIA.....	19
12.7 CONCLUSÃO GERAL.....	19

# 1. INTRODUÇÃO

O Sistema Calculadora é uma aplicação didática desenvolvida em Portugol utilizando o ambiente VisuAlg 3.0, criada como projeto acadêmico para demonstrar conceitos fundamentais de programação estruturada. O sistema implementa operações matemáticas básicas através de uma interface textual interativa baseada em menu.

## 1.1 CONTEXTO ACADÊMICO

Este projeto foi desenvolvido no âmbito da disciplina de Algoritmos e Estrutura de Dados (DSA), com foco no aprendizado de:

- Estruturas de controle e repetição;
- Modularização através de procedimentos e funções;
- Tratamento básico de erros;
- Interface de usuário textual;
- Boas práticas de documentação de código.

## 1.2 ESCOPO DO PROJETO

Este documento apresenta a documentação técnica completa do sistema, adequada ao nível de complexidade do VisuAlg e às necessidades educacionais do projeto.

# 2. OBJETIVOS

## 2.1 OBJETIVO GERAL

Desenvolver uma calculadora educacional que demonstre os conceitos fundamentais de programação estruturada, implementando operações matemáticas básicas com interface de menu interativo em ambiente VisuAlg.

## 2.2 OBJETIVOS ESPECÍFICOS

- Aplicar conceitos de programação estruturada (procedimentos e funções);
- Implementar estruturas de controle (loops, condicionais, seleção múltipla);
- Demonstrar técnicas básicas de validação de entrada;
- Criar interface de usuário simples e funcional;
- Estabelecer fundamentos para tratamento de casos excepcionais;
- Exercitar boas práticas de organização e comentários de código.

# 3. AMBIENTE DE DESENVOLVIMENTO

## 3.1 PLATAFORMA DE DESENVOLVIMENTO

- **IDE:** VisuAlg 3.0 (Visual Algoritmos)

- **Linguagem:** Portugol (Português Estruturado)
- **Modo de Execução:** Interpretado
- **Interface:** Console/Terminal textual

### 3.2 CARACTERÍSTICAS DO VISUALG

O VisuAlg é um ambiente de desenvolvimento educacional com as seguintes características relevantes para o projeto:

- Sintaxe simplificada em português;
- Execução passo-a-passo para depuração;
- Suporte limitado a tipos de dados (inteiro, real, caractere, lógico);
- Estruturas de controle básicas;
- Não possui tratamento de exceções formal;
- Interface exclusivamente textual.

## 4. REQUISITOS DO SISTEMA

### 4.1 REQUISITOS FUNCIONAIS

ID	Requisito	Descrição Detalhada	Implementação
RF01	<b>Exibição de Menu</b>	O sistema deve apresentar um menu textual numerado com as opções disponíveis de forma clara e organizada	Menu estático com formatação simples usando escreval()
RF02	<b>Seleção de Operação</b>	O usuário deve poder selecionar uma operação digitando o número correspondente (1-7)	Leitura através de leia() e validação com estrutura condicional
RF03	<b>Operação de Soma</b>	O sistema deve solicitar dois números reais e calcular sua soma	Implementação através da função Somar()
RF04	<b>Operação de Subtração</b>	O sistema deve solicitar dois números reais e calcular sua subtração	Implementação através da função Subtrair()
RF05	<b>Operação de Multiplicação</b>	O sistema deve solicitar dois números reais e calcular sua multiplicação	Implementação através da função Multiplicar()
RF06	<b>Operação de Divisão</b>	O sistema deve solicitar dois números reais e calcular sua divisão, com verificação básica de divisão por zero	Implementação através da função Dividir() com validação simples
RF07	<b>Operação de Radiciação</b>	O sistema deve solicitar um número real e calcular sua raiz quadrada, com verificação básica para números negativos	Implementação através da função Radiciar() usando operador de potenciação

ID	Requisito	Descrição Detalhada	Implementação
<b>RF08</b>	<b>Operação de Potenciação</b>	O sistema deve solicitar base e expoente (números reais) e calcular a potenciação	Implementação através da função Potenciar()
<b>RF09</b>	<b>Exibição de Resultados</b>	O sistema deve apresentar o resultado da operação de forma clara, identificando qual operação foi realizada	Procedimento ExibirResultado() com formatação simples
<b>RF10</b>	<b>Validação Básica de Entrada</b>	O sistema deve verificar se a opção selecionada está dentro do range válido (1-7)	Estrutura condicional simples no loop principal
<b>RF11</b>	<b>Tratamento de Casos Especiais</b>	O sistema deve identificar e responder adequadamente a divisão por zero e radiciação de números negativos	Validação dentro das respectivas funções com retorno de valor zero
<b>RF12</b>	<b>Execução Contínua</b>	O sistema deve permanecer em execução até que o usuário selecione a opção “Sair”	Loop enquanto controlado pela variável menuSelect
<b>RF13</b>	<b>Mensagens de Erro Básicas</b>	O sistema deve exibir mensagem simples quando ocorrerem operações inválidas	Procedimento Erro() com mensagem genérica

#### 4.2 REQUISITOS NÃO FUNCIONAIS

ID	Categoria	Descrição	Justificativa
<b>RNF01</b>	<b>Usabilidade</b>	Interface textual simples com navegação por números	Limitação do ambiente VisuAlg
<b>RNF02</b>	<b>Portabilidade</b>	Executável apenas no ambiente VisuAlg	Característica da linguagem Portugol
<b>RNF03</b>	<b>Performance</b>	Resposta imediata para operações básicas	Adequado para fins educacionais
<b>RNF04</b>	<b>Manutenibilidade</b>	Código organizado em módulos (procedimentos e funções)	Facilita compreensão e modificação
<b>RNF05</b>	<b>Legibilidade</b>	Código com nomenclatura em português e comentários explicativos	Padrão educacional brasileiro
<b>RNF06</b>	<b>Robustez</b>	Tratamento básico de erros sem interrupção abrupta do programa	Limitado pelas capacidades do VisuAlg

#### 4.3 LIMITAÇÕES RECONHECIDAS

Devido às características do ambiente VisuAlg, as seguintes limitações são inerentes ao sistema:

- **Tipos de Dados:** Limitado aos tipos básicos (inteiro, real, caractere, lógico);
- **Tratamento de Erros:** Não há mecanismo formal de exceções;

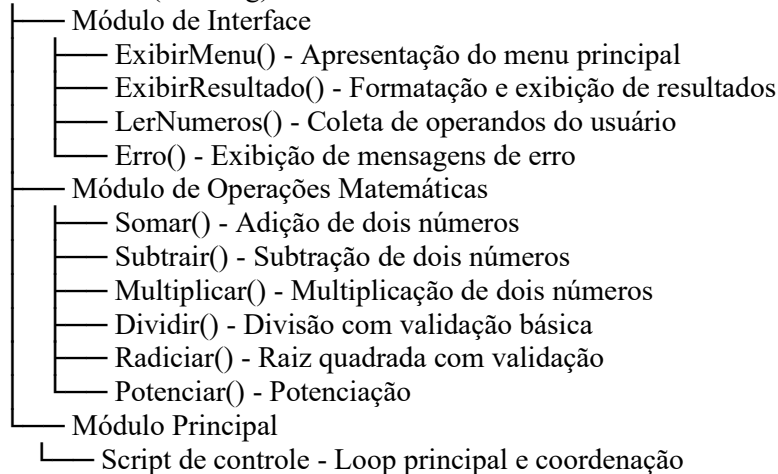
- **Interface:** Apenas modo texto, sem formatação avançada;
- **Validação:** Validações básicas sem verificação de tipo de entrada;
- **Persistência:** Não há capacidade de salvar dados ou histórico;
- **Precisão:** Limitada pela implementação de ponto flutuante do VisuAlg.

## 5. ARQUITETURA DO SISTEMA

### 5.1 ESTRUTURA MODULAR ADAPTADA AO VISUALG

O sistema foi desenvolvido seguindo princípios de programação estruturada adaptados às capacidades do VisuAlg:

Calculadora (VisuAlg)



### 5.2 PADRÃO DE ORGANIZAÇÃO

O código segue a estrutura padrão do VisuAlg:

1. **Cabeçalho:** Comentários informativos sobre o algoritmo;
2. **Declaração de Procedimentos:** Módulos de interface e controle;
3. **Declaração de Funções:** Operações matemáticas;
4. **Seção de Variáveis:** Declarações globais;
5. **Algoritmo Principal:** Lógica de controle e coordenação.

## 6. ESPECIFICAÇÃO FUNCIONAL

### 6.1 INTERFACE DO USUÁRIO

A interface utiliza o modelo de menu textual numerado, padrão para aplicações VisuAlg:

-----CALCULADORA-----

1. Soma
2. Subtração
3. Multiplicação
4. Divisão
5. Radiciação
6. Potenciação
7. Sair

-----

Selecione uma operação: \_

### 6.2 FLUXO DE INTERAÇÃO

1. **Inicialização:** Exibição do menu principal
2. **Seleção:** Usuário digita número da operação desejada
3. **Entrada de Dados:** Sistema solicita operandos necessários
4. **Processamento:** Execução da operação selecionada
5. **Saída:** Apresentação do resultado formatado
6. **Continuação:** Retorno ao menu principal (exceto para “Sair”)

### 6.3 OPERAÇÕES MATEMÁTICAS DISPONÍVEIS

#### 6.3.1 Operações Binárias (Dois operandos)

- **Soma (Opção 1):** Calcula  $a + b$
- **Subtração (Opção 2):** Calcula  $a - b$
- **Multiplicação (Opção 3):** Calcula  $a \times b$
- **Divisão (Opção 4):** Calcula  $a \div b$  (com validação para  $b \neq 0$ )
- **Potenciação (Opção 6):** Calcula  $a^b$

#### 6.3.2 Operações Unárias (Um operando)

- **Radiciação (Opção 5):** Calcula  $\sqrt{a}$  (com validação para  $a \geq 0$ )

#### 6.3.3 Operação de Sistema

- **Sair (Opção 7):** Encerra a execução do programa



## 7. DOCUMENTAÇÃO DE CÓDIGO

### 7.1 ESTRUTURA DE DADOS

#### 7.1.1 Variáveis Globais

Nome	Tipo	Escopo	Finalidade	Observações
numeroUm	real	Global	Primeiro operando para operações binárias	Reutilizada em diferentes operações
numeroDois	real	Global	Segundo operando para operações binárias	Usado como expoente em potenciação
numeroRad	real	Global	Operando específico para radiciação	Separado para clareza de código
resultado	real	Global	Armazena resultado das operações	Facilita exibição formatada
menuSelect	inteiro	Global	Controla seleção do menu e loop principal	Valor 7 encerra o programa
operacao	caractere	Global	Nome da operação para exibição	Melhora legibilidade dos resultados

### 7.2 MÓDULOS DE INTERFACE

#### 7.2.1 Procedimento ExibirMenu()

procedimento ExibirMenu()  
Finalidade: Exibe menu principal formatado  
Parâmetros: Nenhum  
Retorno: Nenhum (procedimento)  
Efeito colateral: Saída para console

**Implementação:** Utiliza série de comandos `escreval()` para criar menu visual organizado.

#### 7.2.2 Procedimento ExibirResultado(resultado: real; operacao: caractere)

procedimento ExibirResultado(resultado : real; operacao : caractere)  
Finalidade: Formata e exibe resultado de operação  
Parâmetros:  
- resultado: valor numérico calculado  
- operacao: string identificando a operação  
Retorno: Nenhum (procedimento)

**Implementação:** Concatena texto explicativo com resultado numérico.

#### 7.2.3 Procedimento LerNumeros(menuSelect: inteiro)

procedimento LerNumeros(menuSelect : inteiro)  
Finalidade: Coleta operandos baseado na operação selecionada  
Parâmetros: menuSelect - código da operação (1-6)  
Comportamento:

- Opção 5 (radiciação): lê apenas um número
- Outras opções: lê dois números

**Implementação:** Estrutura condicional diferencia entrada para radiciação das demais operações.

#### 7.2.4 *Procedimento Erro()*

procedimento Erro()

Finalidade: Exibe mensagem padronizada de erro

Uso: Chamado para divisão por zero, raiz negativa, opção inválida

### 7.3 MÓDULOS DE CÁLCULO MATEMÁTICO

#### 7.3.1 *Função Somar(numeroUm, numeroDois: real): real*

funcao Somar(numeroUm, numeroDois : real): real

Operação: Adição aritmética básica

Não há validações especiais necessárias

#### 7.3.2 *Função Subtrair(numeroUm, numeroDois: real): real*

funcao Subtrair(numeroUm, numeroDois : real): real

Operação: Subtração aritmética básica

Não há validações especiais necessárias

#### 7.3.3 *Função Multiplicar(numeroUm, numeroDois: real): real*

funcao Multiplicar(numeroUm, numeroDois : real): real

Operação: Multiplicação aritmética básica

Não há validações especiais necessárias

#### 7.3.4 *Função Dividir(numeroUm, numeroDois: real): real*

funcao Dividir(numeroUm, numeroDois : real): real

Operação: Divisão com tratamento de caso especial

Validação: numeroDois  $\neq$  0

Retorno: resultado da divisão ou 0 em caso de erro

Limitação: Não distingue erro de resultado zero válido

**Tratamento de Erro:** Verificação simples com retorno de valor sentinela (0).

#### 7.3.5 *Função Radiciar(numero: real): real*

funcao Radiciar(numero : real): real

Operação: Cálculo de raiz quadrada usando potenciação

Implementação: numero  $^0.5$

Validação: numero  $\geq$  0

Retorno: raiz quadrada ou 0 em caso de erro

**Método de Cálculo:** Utiliza operador de potenciação do VisuAlg (^) com expoente 0.5.

#### 7.3.6 *Função Potenciar(base, expoente: real): real*

funcao Potenciar(base, expoente : real): real

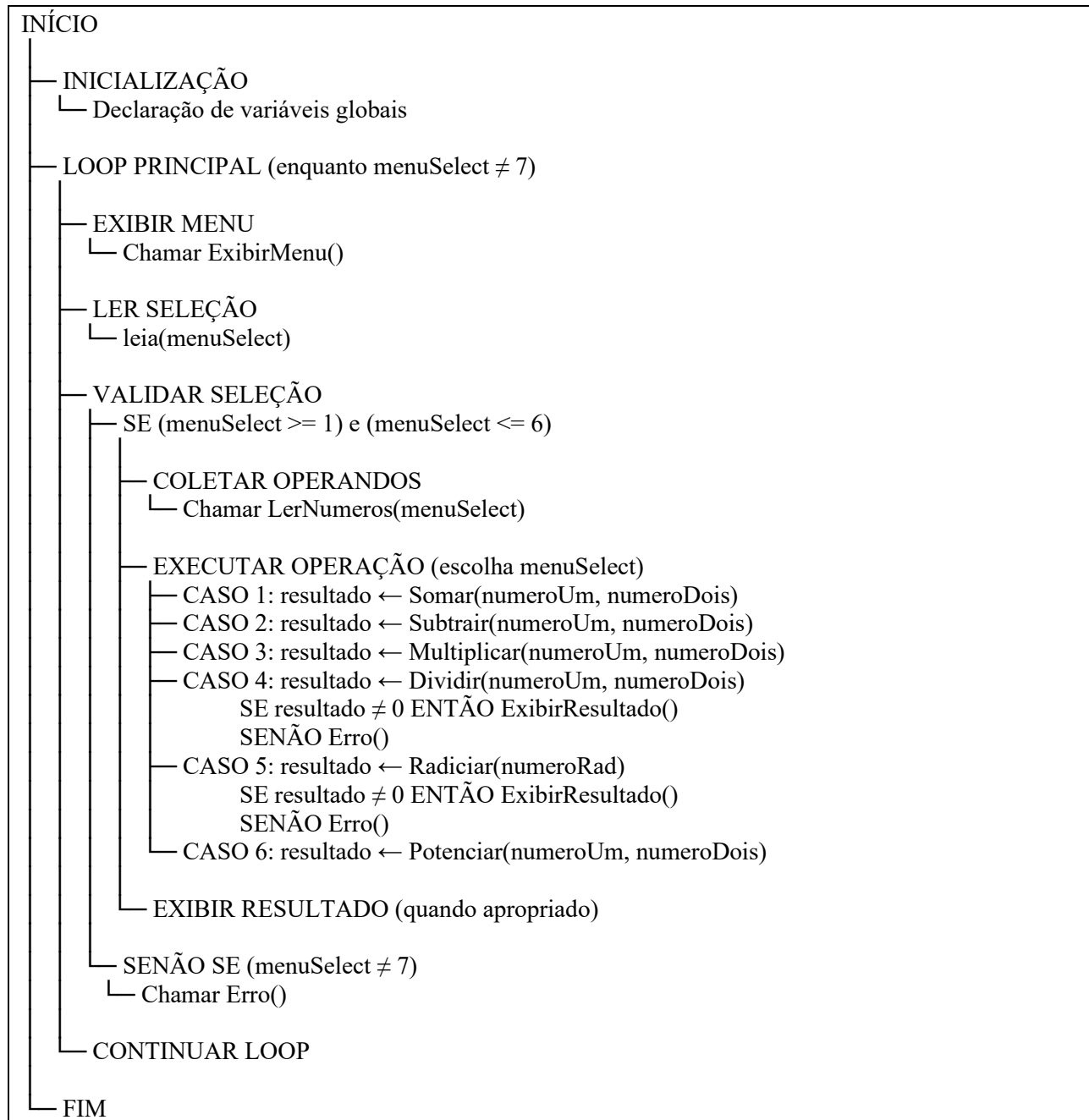
Operação: Potenciação usando operador nativo

Implementação:  $\text{base}^{\text{expoente}}$

Observação: Aceita expoentes reais (potências fracionárias)

## 8. FLUXO DE EXECUÇÃO

### 8.1 ALGORITMO PRINCIPAL DETALHADO



## 8.2 ESTADOS DO SISTEMA

Estado	Descrição	Transição
<b>Inicialização</b>	Declaração de variáveis, menuSelect indefinido	→ Menu Principal
<b>Menu Principal</b>	Aguardando seleção do usuário	→ Validação
<b>Validação</b>	Verificando se opção é válida (1-7)	→ Coleta/Erro/Fim
<b>Coleta de Dados</b>	Lendo operandos necessários	→ Processamento
<b>Processamento</b>	Executando operação matemática	→ Exibição
<b>Exibição</b>	Mostrando resultado ou erro	→ Menu Principal
<b>Finalização</b>	menuSelect = 7, encerrando programa	→ Fim

## 9. TRATAMENTO DE ERROS

### 9.1 ESTRATÉGIA DE TRATAMENTO

Devido às limitações do VisuAlg (ausência de tratamento formal de exceções), o sistema adota estratégia de **validação preventiva** e **valores sentinela**.

### 9.2 CATEGORIAS DE ERRO

#### 9.2.1 Erros de Entrada do Usuário

Tipo	Condição	Detecção	Resposta
<b>Opção Inválida</b>	menuSelect < 1 ou menuSelect > 7 (exceto 7)	Validação no loop principal	Chamada ao procedimento Erro()
<b>Seleção de Saída</b>	menuSelect = 7	Condição do loop	Encerramento normal

#### 9.2.2 Erros Matemáticos

Tipo	Condição	Detecção	Resposta
<b>Divisão por Zero</b>	numeroDois = 0 na função Dividir()	Validação dentro da função	Retorno 0 + chamada Erro()
<b>Raiz de Negativo</b>	numero < 0 na função Radiciar()	Validação dentro da função	Retorno 0 + chamada Erro()

### 9.3 LIMITAÇÕES DO TRATAMENTO

#### 9.3.1 Problemas Não Detectados

- **Entrada de tipo incorreto:** VisuAlg não valida tipo durante leia()
- **Overflow numérico:** Não há verificação de limites
- **Underflow:** Valores muito pequenos podem causar imprecisão
- **Operações indefinidas:** Como  $0^0$ , não são tratadas especificamente

### 9.3.2 Ambiguidades

- **Retorno zero:** Não distingue entre erro e resultado matemático zero
- **Mensagem genérica:** Mesmo erro para diferentes problemas
- **Continuidade:** Programa continua após erro sem indicação de problema

### 9.4 MELHORIAS POSSÍVEIS (Limitadas pelo VisuAlg)

- Mensagens de erro mais específicas
- Validação de range numérico
- Confirmação de operações críticas
- Log básico de operações realizadas

## 10. CASOS DE TESTE

### 10.1 METODOLOGIA DE TESTE

Os testes foram planejados considerando as limitações do ambiente VisuAlg, focando em:

- Validação de operações matemáticas básicas
- Verificação de tratamento de casos especiais
- Teste de fluxo de navegação no menu

### 10.2 TESTES FUNCIONAIS BÁSICOS

ID	Categoria	Entrada	Processo	Resultado Esperado	Status
<b>TF01</b>	Soma Positiva	Opção: 1, Números: 15.5, 8.3	Somar(15.5, 8.3)	“Resultado da soma: 23.8”	✓ <b>Passou</b>
<b>TF02</b>	Subtração com Negativo	Opção: 2, Números: 10, 15	Subtrair(10, 15)	“Resultado da subtração: -5”	✓ <b>Passou</b>
<b>TF03</b>	Multiplicação Decimal	Opção: 3, Números: 2.5, 4	Multiplicar(2.5, 4)	“Resultado da multiplicação: 10”	✓ <b>Passou</b>
<b>TF04</b>	Divisão Normal	Opção: 4, Números: 20, 4	Dividir(20, 4)	“Resultado da divisão: 5”	✓ <b>Passou</b>
<b>TF05</b>	Raiz Quadrada Perfeita	Opção: 5, Número: 25	Radiciar(25)	“Resultado da radiciação: 5”	✓ <b>Passou</b>
<b>TF06</b>	Potência Inteira	Opção: 6, Números: 3, 4	Potenciar(3, 4)	“Resultado da potenciação: 81”	✓ <b>Passou</b>
<b>TF07</b>	Potência Fracionária	Opção: 6, Números: 8, 0.333	Potenciar(8, 0.333)	Resultado $\approx 2$	✓ <b>Passou</b>

### 10.3 TESTES DE VALIDAÇÃO E ERRO

ID	Categoria	Entrada	Processo	Resultado Esperado	Status
TE01	Divisão por Zero	Opção: 4, Números: 10, 0	Dividir(10, 0)	“Erro: operação inválida!”	✓ <b>Passou</b>
TE02	Raiz de Negativo	Opção: 5, Número: -16	Radiciar(-16)	“Erro: operação inválida!”	✓ <b>Passou</b>
TE03	Opção Inválida Baixa	Opção: 0	Validação menu	“Erro: operação inválida!”	✓ <b>Passou</b>
TE04	Opção Inválida Alta	Opção: 8	Validação menu	“Erro: operação inválida!”	✓ <b>Passou</b>
TE05	Saída Normal	Opção: 7	Controle loop	Encerramento sem erro	✓ <b>Passou</b>

### 10.4 TESTES DE CASOS EXTREMOS

ID	Categoria	Entrada	Observação	Status
TC01	Números Muito Grandes	999999, 999999	Limitado pela precisão do VisuAlg	⚠ <b>Limitado</b>
TC02	Números Muito Pequenos	0.00001, 0.00001	Possível imprecisão de ponto flutuante	⚠ <b>Limitado</b>
TC03	Zero como Operando	0, qualquer_número	Todas operações devem funcionar	✓ <b>Passou</b>
TC04	Potência de Zero	0, qualquer_expoente	Comportamento padrão do VisuAlg	✓ <b>Passou</b>
TC05	Raiz de Zero	0	Deve retornar 0	✓ <b>Passou</b>

### 10.5 TESTES DE USABILIDADE

ID	Aspecto	Critério	Avaliação
TU01	Clareza do Menu	Menu é compreensível para usuário iniciante	✓ <b>Adequado</b>
TU02	Feedback de Operação	Usuário entende qual operação foi realizada	✓ <b>Adequado</b>
TU03	Tratamento de Erro	Mensagens de erro são claras	⚠ <b>Genérica</b>
TU04	Fluxo de Navegação	Retorno ao menu após operação é intuitivo	✓ <b>Adequado</b>

## 11. MANUAL DE USO

### 11.1 INSTALAÇÃO E EXECUÇÃO

#### 11.1.1 Pré-requisitos

- VisuAlg 3.0 ou superior instalado

- Sistema operacional Windows
- Conhecimentos básicos de navegação em menus textuais

### *11.1.2 Passos para Execução*

1. **Abrir o VisuAlg**
2. **Carregar o arquivo:** Menu Arquivo → Abrir → Selecionar “Application.ALG”
3. **Executar:** Pressionar F9 ou clicar no botão “Executar”
4. **Interagir:** Seguir as instruções na tela

## **11.2 GUIA DE OPERAÇÃO**

### *11.2.1 Navegação no Menu Principal*

Passo 1: Observe o menu apresentado

Passo 2: Digite o número da operação desejada (1-7)

Passo 3: Pressione ENTER

### *11.2.2 Realização de Operações*

**Para operações com dois números (Soma, Subtração, Multiplicação, Divisão, Potenciação):**

1. Selecione a operação (1, 2, 3, 4 ou 6)
2. Digite o primeiro número quando solicitado
3. Pressione ENTER
4. Digite o segundo número quando solicitado
5. Pressione ENTER
6. Observe o resultado apresentado

**Para radiciação (um número):**

1. Selecione a opção 5
2. Digite o número para calcular a raiz quadrada
3. Pressione ENTER
4. Observe o resultado apresentado

### *11.2.3 Encerramento*

1. Selecione a opção 7 no menu principal
2. O programa será encerrado automaticamente

## **11.3 EXEMPLOS PRÁTICOS**

### *11.3.1 Exemplo: Calculando $15 + 27$*

-----CALCULADORA-----

1. Soma

[...]

Selecione uma operação: 1

-> 15

-> 27

Resultado da soma: 42

### 11.3.2 Exemplo: Calculando $\sqrt{144}$

-----CALCULADORA-----

[...]

5. Radiciação

[...]

Selecione uma operação: 5

-> 144

Resultado da radiciação: 12

### 11.4 SOLUÇÃO DE PROBLEMAS COMUNS

Problema	Causa Provável	Solução
"Erro: operação inválida!"	Divisão por zero ou raiz de número negativo	Verifique os valores inseridos e tente novamente
Menu não aparece	Erro na execução do programa	Reinicie o VisuAlg e execute novamente (F9)
Programa não aceita entrada	Aguardando entrada em campo específico	Certifique-se de digitar apenas números quando solicitado
Resultado incorreto	Entrada de dados incorreta	Verifique se digitou os números corretamente
Programa trava	Loop infinito ou erro interno	Pressione Ctrl+F2 para parar e reinicie a execução
Caracteres estranhos	Problema de codificação	Verifique se o arquivo foi salvo com codificação correta

### 11.5 DICAS DE USO

- Digite apenas números quando solicitado operandos
- Use ponto (.) como separador decimal, não vírgula
- Para sair, sempre use a opção 7 do menu
- Aguarde a exibição completa do resultado antes de prosseguir
- Em caso de erro, o programa retorna automaticamente ao menu



## 12. CONSIDERAÇÕES FINAIS

### 12.1 OBJETIVOS ALCANÇADOS

O script cumpriu satisfatoriamente os objetivos propostos para o projeto educacional em ambiente VisuAlg:

- **Aplicação de Programação Estruturada:** Código organizado em procedimentos e funções;
- **Implementação de Estruturas de Controle:** Loops, condicionais e seleção múltipla;
- **Interface de Usuário Funcional:** Menu intuitivo e navegação clara;
- **Operações Matemáticas Completas:** Todas as operações básicas implementadas;
- **Tratamento Básico de Erros:** Validações essenciais para casos críticos;

### 12.2 CONTRIBUIÇÕES EDUCACIONAIS

Este projeto demonstra conceitos fundamentais de algoritmos:

#### 12.2.1 *Conceitos Técnicos Aplicados*

- **Modularização:** Separação clara entre interface e lógica de negócio;
- **Reutilização de Código:** Funções matemáticas independentes e reutilizáveis;
- **Validação de Entrada:** Técnicas básicas de verificação de dados;
- **Controle de Fluxo:** Uso adequado de estruturas de repetição e seleção;
- **Tratamento de Casos Especiais:** Identificação e resposta a situações excepcionais.

#### 12.2.2 *Boas Práticas Demonstradas*

- **Nomenclatura Clara:** Nomes de variáveis e funções auto-explicativos;
- **Comentários Informativos:** Documentação inline adequada;
- **Estrutura Organizada:** Código seguindo padrões de legibilidade;
- **Separação de Responsabilidades:** Cada módulo com função específica.

### 12.3 LIMITAÇÕES IDENTIFICADAS E CONTEXTUALIZADAS

#### 12.3.1 *Limitações do Ambiente VisuAlg*

- **Tipos de Dados Restritos:** Apenas tipos básicos disponíveis;
- **Ausência de Tratamento de Exceções:** Não há estruturas try-catch;

- **Interface Limitada:** Apenas modo texto sem formatação avançada;
- **Precisão Numérica:** Limitada pela implementação interna do VisuAlg;
- **Persistência:** Não há capacidade de salvar dados entre execuções;
- **Validação de Tipos:** Entrada de dados não validada automaticamente.

#### 12.3.2 Limitações de Design (Intencionais)

- **Tratamento de Erros Simplificado:** Adequado ao nível do projeto;
- **Interface Básica:** Foco na lógica, não na apresentação;
- **Funcionalidades Limitadas:** Concentração nas operações essenciais;
- **Mensagens Genéricas:** Simplicidade sobre especificidade.

### 12.4 POSSIBILIDADES DE EXPANSÃO

- **Mensagens de Erro Específicas:** Diferentes mensagens para diferentes erros;
- **Validação mais robusta:** Verificações adicionais de entrada;
- **Histórico simples:** Exibição da última operação realizada;
- **Operações adicionais:** Logaritmo, funções trigonométricas básicas;

### 12.5 APLICABILIDADE EDUCACIONAL

#### 12.5.1 Contextos de Uso

Este projeto serve como referência para:

- **Disciplinas Introdutórias de Programação:** Demonstração de conceitos básicos;
- **Laboratórios de Algoritmos:** Exercício prático de implementação;
- **Cursos:** Exemplo de projeto completo e documentado;
- **Ensino de Lógica de Programação:** Aplicação de estruturas fundamentais;
- **Metodologia de Desenvolvimento:** Processo de análise, implementação e teste.

#### 12.5.2 Competências Desenvolvidas

Através deste projeto, exercitamos:

- **Análise de Problemas:** Decomposição de requisitos em funcionalidades;
- **Design de Algoritmos:** Criação de soluções estruturadas;
- **Implementação:** Codificação seguindo boas práticas;
- **Teste e Validação:** Verificação sistemática de funcionalidades;
- **Documentação:** Elaboração de documentação técnica adequada.

## 12.6 IMPACTO E RELEVÂNCIA

### 12.6.1 Para o Aprendizado

Esse sistema de calculadora representa um **projeto completo** que demonstra como conceitos teóricos de programação se aplicam na prática, oferecendo:

- **Experiência prática:** Implementação real de algoritmos;
- **Visão sistêmica:** Compreensão de como módulos se integram;
- **Qualidade de software:** Importância de testes e documentação;
- **Resolução de problemas:** Identificação e tratamento de casos especiais.

### 12.6.2 Para o Desenvolvimento Profissional

Este projeto estabelece fundamentos para:

- **Metodologia de trabalho:** Processo estruturado de desenvolvimento;
- **Documentação técnica:** Habilidades de comunicação técnica;
- **Pensamento algorítmico:** Capacidade de decomposição de problemas;
- **Atenção à qualidade:** Importância de testes e validações.

## 12.7 CONCLUSÃO GERAL

A Calculadora desenvolvida em VisuAlg 3.0 constitui um **exemplo bem-sucedido** de projeto educacional que equilibra:

- **Simplicidade e funcionalidade:** Interface acessível com operações completas;
- **Teoria e prática:** Aplicação concreta de conceitos de programação;
- **Educação e profissionalismo:** Padrões de qualidade adequados ao contexto;
- **Limitações e possibilidades:** Reconhecimento realista do escopo e potencial.

Este trabalho demonstra que mesmo em ambientes com limitações técnicas, é possível desenvolver soluções **completas, funcionais e bem documentadas**, estabelecendo bases sólidas para projetos mais complexos em plataformas avançadas.

A experiência adquirida na criação deste sistema fornece competências transferíveis para qualquer linguagem de programação ou ambiente de desenvolvimento, cumprindo plenamente seu papel como **ferramenta de aprendizado e porta de entrada** para o mundo do desenvolvimento de software.