

ESSENCIAL C

ADILSON POLIDORO DE FRANÇA – e-mail: adilson.franca@fatec.sp.gov.br

ALESSANDRA CERQUEIRA PEREIRA – e-mail: alessandra.pereira01@fatec.sp.gov.br

DANILO MIGUEL DE ARAÚJO SALVATO – e-mail: daniilo.salvato@fatec.sp.gov.br

LUCAS CALEBE MARQUES DE ANDRADE – e-mail: lucas.andrade48@fatec.sp.gov.br

RESUMO

Este projeto foi desenvolvido com o intuito de ajudar, estudantes, profissionais e pesquisadores que tem a necessidade de fazer cálculos em seu dia a dia de estudos e trabalho de forma rápida. As contas realizadas em papéis podem facilmente ser danificadas perdendo todo o processo já feito, podendo também causar erros matemáticos durante o cálculo dos operandos. Com isso foi desenvolvido uma calculadora que realiza cálculos de somatório e outros tipos de operações utilizando linguagem de programação C, que de forma muito prática e rápida realiza cálculos complexos. Com esta calculadora é possível fazer operações variadas e foi construída baseando-se nos fundamentos da matemática sendo que com ela é possível realizar as operações de Somatório, Análise Combinatória e Lógica Proposicional. Ela possui um menu simples e amigável para que seja de fácil entendimento e de utilização ao alcance de todos, contém também menus e submenus discriminados com todas as informações necessárias para que o usuário possa compreender o que ela pode oferecer.

Palavras- Chave: Calculadora, Somatório, Programação

ABSTRACT

This project was developed with the objective to help students, professionals and researchers those have a common need of making calculations in their daily studies or work routines. The calculations performed in draft's paper may easily be damaged losing the whole process done. You can also cause little mathematical mistakes during the whole process. With that in mind we developed a calculator that performs summation calculations and other types of operations using C programming language which in a very practical and fast way performs complex calculations. With this calculator you can make a great variety of operations based on the fundamentals of mathematics. You can perform summation, combinatorial analysis and propositional logic operations. It has a simple and friendly menu and submenus which holds all the necessary information for the user to get best of everything it has to offer.

Keywords: Calculator, Sum, programming.

1. INTRODUÇÃO

A tecnologia está presente em nosso dia a dia, sempre inovando e expandindo sua amplitude de fornecer sempre uma solução para nossas rotinas, seja no trabalho, na escola, em casa ou até mesmo do outro lado do mundo.

Essa evolução é essencial para fontes de pesquisas e desenvolvimento de tecnologias para melhorar a qualidade de vida através de informações e serviços instantâneos. Sendo aplicada para uso pessoal como conversar com pessoas distantes, efetuar pagamentos, realizar comprar, entre outras atividades. No trabalho sendo utilizado para realizar reuniões, armazenar documentos, organizar sua rotina, desenvolver projetos, sempre procurando a forma mais rápida e eficaz de resolver os problemas.

Com base em pesquisas foi analisado por alunos a necessidade de uma calculadora que realize cálculos de somatória, lógica proposicional e análise combinatória. As contas realizadas em rascunhos de papéis, no qual podem facilmente serem danificadas perdendo todo o processo já feito, podendo também causar erros matemáticos durante o cálculo dos operandos.

Sendo assim foi dado início ao projeto *Essencial C* uma calculadora capaz de realizar cálculos desejado com eficiência e rapidez. Foi dado esse nome por ser um *software* essencial para quem utiliza esse tipo de operação matemática, e sua linguagem de desenvolvimento é C.

Ela possui um menu simples e amigável para que seja de fácil entendimento e utilização a todos, possuindo menus e submenus discriminados com as informações necessárias para compreender o que ela pode oferecer.

2. FUNDAMENTAÇÃO TEÓRICA

Atualmente os cálculos estão em nosso dia a dia, seja em casa, no trabalho, na escola, em algum momento de lazer, no supermercado, tudo ao nosso redor exige raciocínio de cálculos simples aos complexos. O avanço das tecnologias e todas as coisas que temos hoje, é graças aos estudos da matemática que temos desde o início da história.

A História da Matemática pode oferecer uma importante contribuição ao processo de ensino e aprendizagem dessa área do conhecimento. Ao revelar a Matemática como uma criação humana, ao mostrar necessidades e preocupações de diferentes culturas, em diferentes momentos históricos, ao estabelecer comparações entre os conceitos e processos matemáticos do passado e do presente, o professor cria condições para que aluno desenvolva atitudes e valores mais favoráveis diante desse conhecimento. Além disso, conceitos abordados em conexão com sua história constituem veículos de informação cultural, sociológica e antropológica de grande valor formativo. A História da Matemática é, nesse sentido, um instrumento de resgate da própria identidade cultural. (BRASIL, 1998).

Com tudo a matemática é vista como uma necessidade em nossas vidas, e essencial para grandes projetos futuros. Mas nem todos gostam de matemática, acham ela complexa ou desnecessária, ficar fazendo contas e contas em papéis, se perdendo em raciocínio, não é todo mundo que gosta.

Porém hoje em dia temos diversas ferramentas que auxiliam na realização de cálculos como as calculadoras físicas e digitais, *smatphones* e planilhas eletrônicas. Ultimamente também vem se desenvolvendo *softwares* para fins específicos, cálculos mais complexos e exatos.

Estudantes de desenvolvimento de sistemas que possuem matérias de algoritmos acabam tendo dificuldades com a realização de testes dos cálculos executados em seus projetos. Trabalhadores precisa de algum *software* que o auxilie na contagem de seus materiais e estoques. As pessoas precisam de algo que as auxiliem em seu dia a dia.

Para a resolução deste problema, foi realizada uma pesquisa sobre o assunto e a necessidade das pessoas, e foi comprovado a necessidade de uma calculadora que execute algumas funções para áreas específicas.

2.1 LINGUAGEM DE PROGRAMAÇÃO C E ALGUNS COMANDOS

“A linguagem C foi inventada e implementada primeiramente por Dennis Ritchie em um *DEC PDP-11* que utilizava o sistema operacional *Unix*. C é o resultado de um processo de desenvolvimento que começou com uma linguagem mais antiga, chamada *BCPL*, que ainda está em uso, em sua forma original, na Europa. *BCPL* foi desenvolvida por Martin Richards e influenciou uma linguagem chamada B, inventada por Ken Thompson. Na década de 1970, B levou ao desenvolvimento de C.” (SCHILDT, 1996, página 3)

Ela possui cinco tipos de dados sendo *char* para caracteres, *int* para dados numéricos inteiros, *float* para números reais e inteiros, o *double* também utilizados números reais e inteiros porém com uma maior precisão tendo 32 *bits* a mais que *float* e a variável *void* que permite realizar funções sem parâmetros ou que não retorna nada.

Para declaração de variáveis utilizamos o tipo da variável e dar um nome a ela, elas precisam ser declaradas antes de sua utilização, podendo ser aplicada para realizar cálculos, exibir resultados, e seus valores variam de elementos para elementos, elas podem ser utilizadas dentro e fora de uma função e em definições de parâmetros.

Podemos verificar abaixo um código básico de um cálculo de multiplicação onde declaramos as variáveis do tipo *double* sendo *numero1*, *numero2*, *resultado*. Temos também os comandos *scanf* que está inclusa na biblioteca *stdio.h* que é declarado início do código, que tem como objetivo de ler a informação que o usuário digitou e guardar em uma variável, já o *scanf* é um comando da biblioteca padrão de rotinas, que pode ser produzir dados ou texto sem um formato específico descrito.

FIGURA 1 – código em linguagem C

```

#include <stdio.h>

int main()
{
    float numero1, numero2, resultado;

    printf("Insira o número 1: ");
    scanf("%f", &numero1);

    printf("Insira o número 2: ");
    scanf("%f", &numero2);

    resultado = numero1 * numero2;

    printf("A multiplicação dos valores é: %.2f", resultado);

    return 0;
}

```

Fonte: Autores

Ela é uma linguagem que exige um baixo nível de *hardware*, pois na época de seu lançamento havia surgido os primeiros computadores pessoais que tinham um *hardware* bem fraco por seu alto custo. Ela é uma linguagem que possui diversas bibliotecas padronizadas que auxiliam em seu desenvolvimento.

2.2 SOMATÓRIO

É uma forma compacta de elaborarmos uma somatória de uma determinada regra com início e término em valores pré-definidos. A sua notação é dada por:

$$\sum_{k=a}^n regra$$

Segundo (Taneja,2010), a somatória trata cálculos com números inteiros, a regra é definida por K e N, onde K é a variável de início da repetição e N é o último valor, assim aplicamos a regra do entre as variáveis e obtemos a resposta:

$$\sum_{k=a}^n = regra = regra + regra + \dots + regra + regra$$

para $k = a$, para $k = a + 1, \dots$, para $k = n - 1$ para $k = n$

A partir daqui são adicionadas as propriedades básicas as notações de sigma ou somatória, adição subtração:

Figura 2 - Função básica de somatório

$$\sum_{k=m}^n (a_k + b_k) = \sum_{k=m}^n (a_k) + \sum_{k=m}^n (b_k)$$

$$\sum_{k=m}^n (a_k - b_k) = \sum_{k=m}^n (a_k) - \sum_{k=m}^n (b_k)$$

Fonte: Fundamentos da Matemática II (2010)

Assim além de conseguirmos determinar uma somatória podemos ver a diferença entre somatórias ou acrescentá-las umas as outras.

2.3 ANÁLISE COMBINATÓRIA

A análise combinatória é a parte da Matemática que estuda métodos que permitem resolver problemas relacionados com contagem. Muito utilizada nos estudos sobre probabilidade, ela faz análise das possibilidades e das combinações possíveis entre um conjunto de elementos. O princípio fundamental da contagem, também chamado de princípio multiplicativo, postula que:

Segundo (Taneja,2010), “Quando um evento é composto por n etapas sucessivas e independentes, de tal modo que as possibilidades da primeira etapa é X e as possibilidades da segunda etapa é Y , resulta no número total de possibilidades de o evento ocorrer, dado pelo produto $(x) \cdot (y)$ ”.

Em resumo, no princípio fundamental da contagem, multiplica-se o número de opções entre as escolhas que lhe são apresentadas, e utilizamos a o cálculo fatorial para dar sequência a multiplicação.

2.3.1 PERMUTAÇÃO SIMPLES

Uma sequência ordenada qualquer com um número N de elementos distintos, qualquer outra sequência formada pelos mesmos N elementos reordenados é chamada de permutação, (TANEJA, 2010). Assim adotamos a fórmula:

$$P_n = n!$$

2.3.2 PERMUTAÇÃO COM REPETIÇÃO

Permutação de elementos repetidos deve seguir uma forma diferente da permutação, pois elementos repetidos permutam entre si (TANEJA, 2010). Devemos separar a quantidade dos elementos que se repetem e só então resolver a equação com a seguinte formula:

$$P_n^{n_1, n_2, \dots, n_k} = \frac{n!}{n_1! \cdot n_2! \dots n_k!}$$

Em ambas as fórmulas o número de elementos é igual ao de posições. Desejamos apenas trocar os elementos de posições.

2.3.3 ARRANJO SIMPLES

Um arranjo é uma forma de organizar N elementos dentro de K posições e verificar a quantidade dos possíveis agrupamentos obtidos, sendo que a ordem em que os elementos ficam dispostos e a sua natureza fazem diferença no resultado (TANEJA, 2010).

Um arranjo simples de N elementos tomados K a K é definido por:

$$A_{n,k} = \frac{n!}{(n-k)!}$$

Sendo que não há repetição entre os N elementos. Este resultado retorna à quantidade de maneiras de organizar N elementos dentro de K posições, sendo N maior ou igual a K , sem haver elementos repetidos.

2.3.4 ARRANJO COM REPETIÇÃO

Segundo (Taneja,2010), um arranjo de n elementos tomados K a K , podendo haver repetição de elementos, é definido por:

$$AR_{n,k} = n^k$$

Este resultado retorna o número de maneiras de organizar n elementos dentro de K posições, havendo elementos repetidos. A ordem dos elementos importa.

2.3.5 COMBINAÇÃO SIMPLES

Segundo (Taneja,2010), uma combinação simples de n elementos tomados K a K é definida por:

$$Cn,k = \frac{n!}{k!. (n-k)!}$$

Sendo que não há repetição entre os n elementos. Este resultado retorna à quantidade de maneiras de organizar N elementos dentro de K posições, mantendo N maior ou igual K , com a ordem dos elementos não importando.

2.3.6 COMBINAÇÃO COM REPETIÇÃO

Segundo (Taneja,2010), uma combinação de K elementos tomados K a K , podendo haver repetição de elementos, é definido por:

$$CRn,k = \frac{(n+k-1)!}{k!. (n-1)!}$$

Vale ressaltar que a ordem dos elementos não importa.

3. MATERIAIS E MÉTODOS

Para o desenvolvimento deste projeto foram utilizados alguns materiais e métodos essenciais que ajudaram a progredir com a criação da calculadora, desde fontes de pesquisas, equipamentos, *softwares* e análises.

Portanto será apresentado todas as ferramentas físicas e lógicas utilizadas ao decorrer do processo de desenvolvimento, mostrando a evolução dos métodos aderidos para melhoria do funcionamento da calculadora e dos equipamentos e *softwares* utilizados para programação e análise.

3.1 MATERIAIS

A base essencial do projeto foi a utilização do computador pessoal, possibilitando acesso a softwares para o processo de desenvolvimento. Para análise do projeto, foi utilizado um *software* online chamado *Lucidchart*, ele possui diversas ferramentas desde plantas de casas, a fluxogramas, que foi o utilizado para estrutura de projeto facilitando em seu desenvolvimento.

O programa foi desenvolvido na *IDE Code Blocks*, uma plataforma de código aberto com arquitetura orientada a plugin. Esse *software* ajudou muito na estrutura do código, desenvolvido em linguagem C utilizando bibliotecas que auxiliaram com suas rotinas padronizadas, contendo operações como tratamento de entrada e saída de dados e cadeia de caracteres, entre operações mais complexas.

Para design e protótipos do logotipo foi utilizado o Photoshop CS6, uma ferramenta de edição de imagens, o *software* foi desenvolvido por dois irmãos iniciando com o nome de *ImagePro*. Logo após o aplicativo foi comprado e lançado como *Photoshop 1.0* exclusivo para o *Macintosh*. O programa disponibiliza diversas ferramentas para edição e criação, podendo expandir a criatividade, que possibilitou a criação do logo

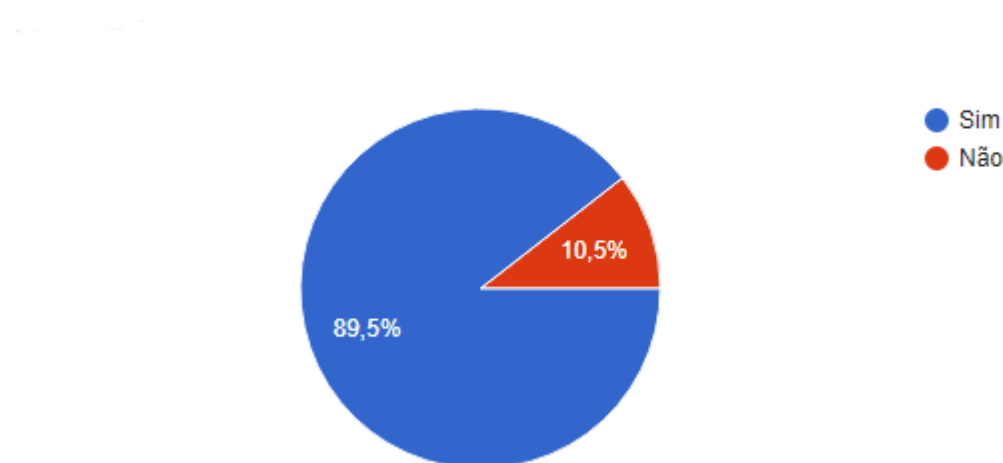
3.2 MÉTODOS

Para colocar o projeto em prática foi realizada uma reunião com a equipe para debater sobre as necessidades e a divisão de tarefas a serem executadas, para tudo foi dado a indispensabilidade de realizar uma pesquisa de campo para averiguar se a elaboração era válida.

Foi criado um formulário básico com perguntas simples, porém de grande importância no qual foi confirmado a necessidade das pessoas deste software em seu dia a dia.

FIGURA 3 – Resposta formulário 1

Você tem dificuldade para realizar cálculos de cabeça ou em rascunhos de papéis ?

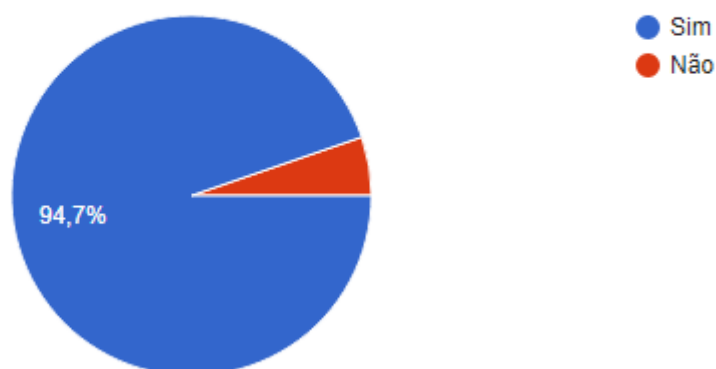


Fonte: Autores

FIGURA 4 – Resposta formulário 2

Você trabalha ou estuda em algum setor que utiliza cálculos ?

Gráfico de pizza

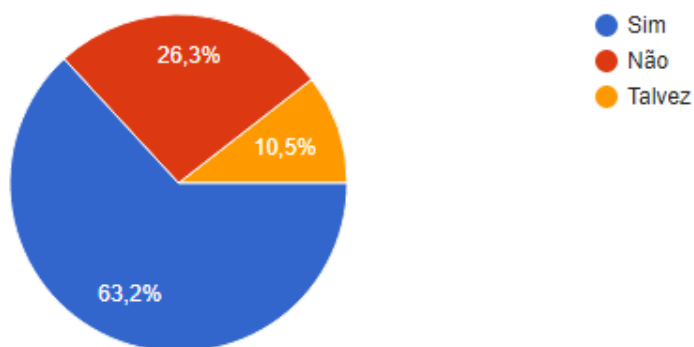


Fonte: Autores

FIGURA 5 – Resposta formulário 3

Você consideraria útil a ideia de uma calculadora que execute as operações que você tem dificuldade ?

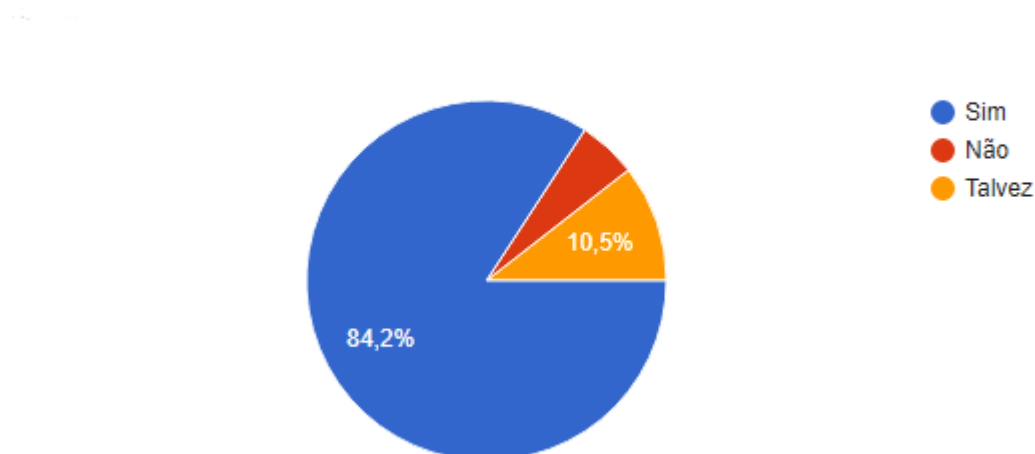
Gráfico de pizza



Fonte: Autores

FIGURA 6– Resposta formulário 4

Você utilizaria essa calculadora no seu dia a dia ?



Fonte: Autores

Como mostra os gráficos acima, maioria das pessoas possuem dificuldades com cálculos mentais, e realizá-los em rascunhos de papéis pode gerar rasuras e falhas humanas. Dado essas informações foi possível dar continuidade no projeto atribuindo as tarefas para *layout*, *design*, programação, estatísticas, cálculos, entre outros métodos administrativos para gerenciar da melhor forma o *software*.

4. DESENVOLVIMENTO

Seu desenvolvimento baseou-se diversos *softwares*, desde a criação do logo até toda estrutura do código que foi o que deu a vida ao projeto. A principal ferramenta utilizada foi a linguagem C, onde foi construído todo conteúdo, fórmulas, cálculos, funções para o funcionamento da calculadora, foram utilizados métodos que facilitou em seu desenvolvimento.

O projeto foi desenvolvido utilizando a linguagem C por obter seu requisito de baixo nível de *hardware*, e por obter uma estrutura de fácil compreensão. Os comandos da linguagem C utilizados são básicos, porém como toda linguagem exige raciocínio na hora de desenvolver a lógicas. Alguns dos seus comandos mais utilizados no projeto para realizar o processo básico de um cálculo de somatório encontra-se na imagem abaixo:

FIGURA 7 - Código da opção de um menu somatório

```

switch (valorSomatorio){
case 1: //Opcao 1 do somatório.
    printf("\n -- Somatório de J = 0 até N para a regra J --\n\n");
    printf("    Entre com o valor de N: ");
    scanf("%d", &calcular1);
    fflush(stdin); //Serve para limpar o buffer, caso o usuário entre com uma letra ao invés de número, evitando crash.
    somatorioOne(calcular1); //Chamada da função do arquivo functions.h para realizar a operação.
    printf("    1. Realizar uma nova operação.\n");
    printf("    2. Menu principal.\n\n");
    printf("    Entre com um valor listado acima: ");
    scanf("%d", &home);
    if(home < 1 || home > 2){
        fflush(stdin);
        system("cls"); //Responsável por limpar a tela.
        printf("\n    Esta não é uma opção válida.\n");
        printf("    Retornando ao menu somatório.....");
        delay(3); //Deley de 3 segundos para o usuário ler a mensagem, limpar a tela e retornar ao menu somatório.
        system("cls"); //Responsável por limpar a tela.
        goto menuSomatorio; //Retorna ao menu somatório.
    } else if (home == 1) { //Opcao 1 retorna ao menu somatório
        system("cls");
        goto menuSomatorio;
    } else {
        system("cls");
        goto menuPrincipal;
    }
break; //break case 1 (menuSomatorio)

```

Fonte: Autores

Como pode-se observar acima, esse código é para a realização de um cálculo de um somatório onde $J = 0$ até N para a regra J , utilizando-se o *Switch case* que é uma função de escolha, no qual o usuário optou pelo menu desse somatório. Então o caso escolhido foi a opção 1 do menu de somatório.

Para exibir dados e informações aos usuários utiliza-se a função *printf*, que tem como objetivo mostrar descrever ao usuário o que ele pode fazer no sistema, mostrando a ele que deve inserir a letra N para realizar o cálculo desejado.

A função *scanf* tem o objetivo de guardar na variável N o valor inserido pelo usuário, e para evitar *bugs* caso o usuário insira alguma letra, foi utilizado a função *fflush(stdin)* que limpa o *buffer*. Após ter sua variável armazenada o sistema irá realizar o processo do cálculo do somatório.

Antes de realizar o cálculo o sistema executa alguns comando para melhor apresentar ao usuário, temo a função *system(cls)* que limpa a tela deixando uma interface nova e mais limpa, e *delay(3)* onde caso a opção inserida seja invalida, o programa irá exibir uma mensagem e retornar ao menu em 3 segundos utilizando a função *goto*.

Todo esse processo é verificado dentro do comando *if* que altera o fluxo de execução com valores verdadeiros ou falsos, então se a opção escolhida pelo usuário for verdadeira ele irá executar um processo, se não executará outro.

Para o cálculo do somatório por exemplo como mostrado na imagem abaixo onde *int* declara a função *somatorioOne*. Uma variável denominada soma de valor inteiro é declarada onde irá armazenar o resultado da operação, e foi utilizada a sintaxe de laço de repetição *for*, onde realiza o cálculo do somatório selecionado e retorna o resultado com o *return* e *printf*.

FIGURA 8 - Código do somatório de J=0 até N para a regra J

```
// Inicio das funcoes para menu SOMATÓRIO.  
  
//Função designada para o Somatório de J = 0 até N para a regra J.  
int somatorioOne(int valor)  
{  
    int soma;  
  
    for (int j = 0; j <= valor; ++j)  
    {  
        soma += j;  
    }  
    return (printf("\n    O resultado do somatório é: %d\n\n", soma));  
}
```

Fonte: Autores

Todo o código foi discriminado com comentários em linha utilizando o caractere //, que facilita para os desenvolvedores a se encontrar no código e entender o que o programa faz detalhadamente, assim possibilitando a facilidade de melhorias futuras ou conserto de bugs no projeto.

4.1 IDENTIDADE VISUAL

O logo do software foi desenvolvido na ferramenta *Photoshop CS6*, foi feita análise de cores e, tamanhos, formatos entre outras características. Foram criados vários protótipos e ideias de design. Foi optado por cores de tonalidades fortes como o roxo e o amarelo que se destacaram no background escuro.

FIGURA 9 – Logo Essencial C

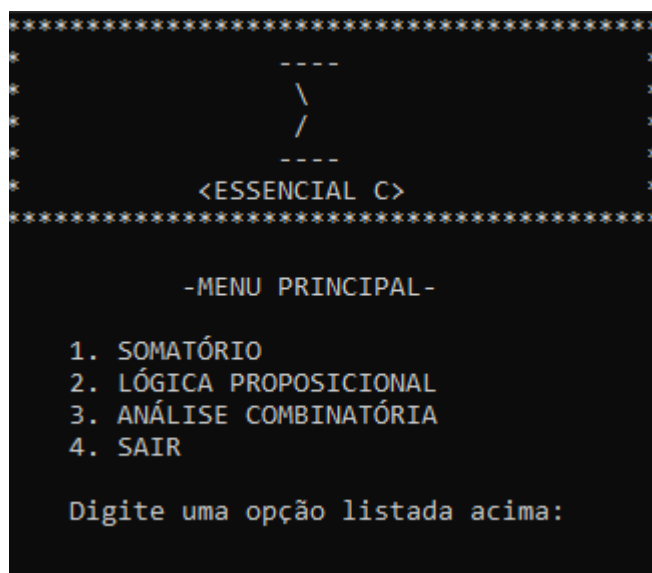


Fonte: Autores

O nome foi desenvolvido pensando na principal função da calculadora que seria somatório, porém ela possui outros cálculos. A letra C é representada por sua linguagem de programação no qual o sistema foi desenvolvido, e por ser uma Calculadora, obtendo dois sentidos na letra.

A interface do sistema foi desenvolvida pensando no usuário, obtendo uma *GUI (Graphical User Interface)* agradável e de fácil entendimento, tendo menus organizados com números ordenados e explicativos, facilitando o entendimento das funções que a calculadora oferece.

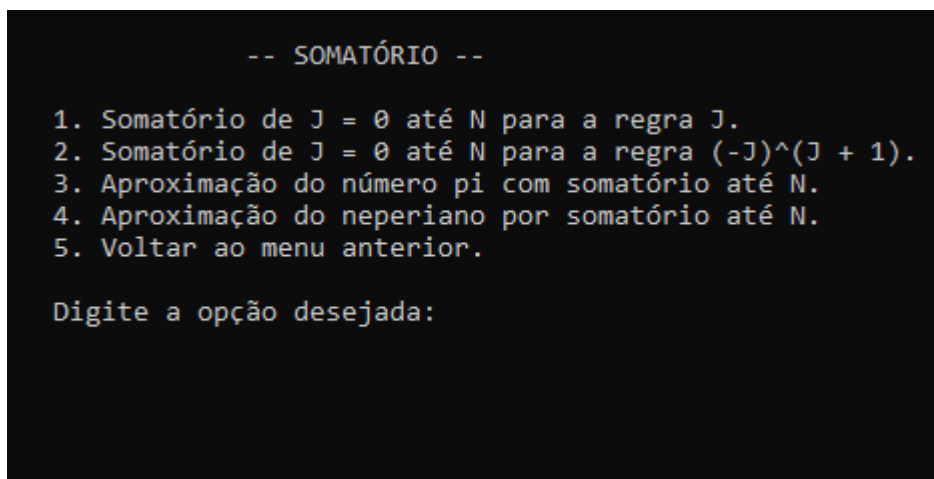
FIGURA 10 – Menu Principal



Fonte: Autores

Em cada opção existe submenus para cada operação da função que o usuário deseja executar, no qual também são organizadas por números ordenados e suas respectivas operações o que ajuda o usuário a encontrar o cálculo desejado de forma mais exata.

FIGURA 11 – Menu Somatório



Fonte: Autores

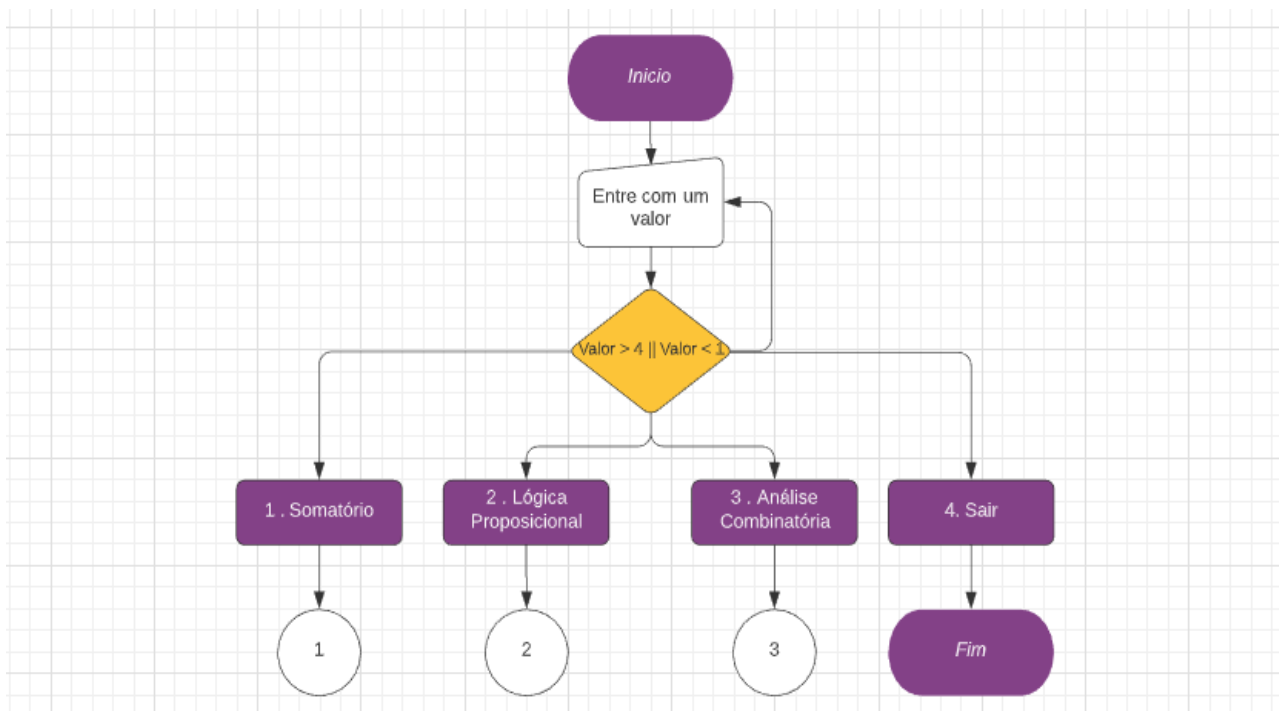
4.2 FLUXOGRAMA

Para dar início ao desenvolvimento do projeto foi preciso fazer uma análise de estrutura do que seria desenvolvido na calculadora, desde seus menus às suas funções e resultados.

Com ajuda do *software* online chamado *Lucidchart*, foi possível colocar o plano em prática começando pelo menu do sistema que possui a forma inicial e de decisão, que vai levar o usuário a decidir o cálculo que deseja executar. Os fluxogramas foram feitos separadamente por seus respectivos cálculos.

Caso a opção 1 seja selecionada, onde indica que o cálculo que o usuário deseja realizar é o de somatório, logo após escolher essa opção, ele será direcionado aos submenus onde poderá escolher qual regra deseja aplicar a seu somatório.

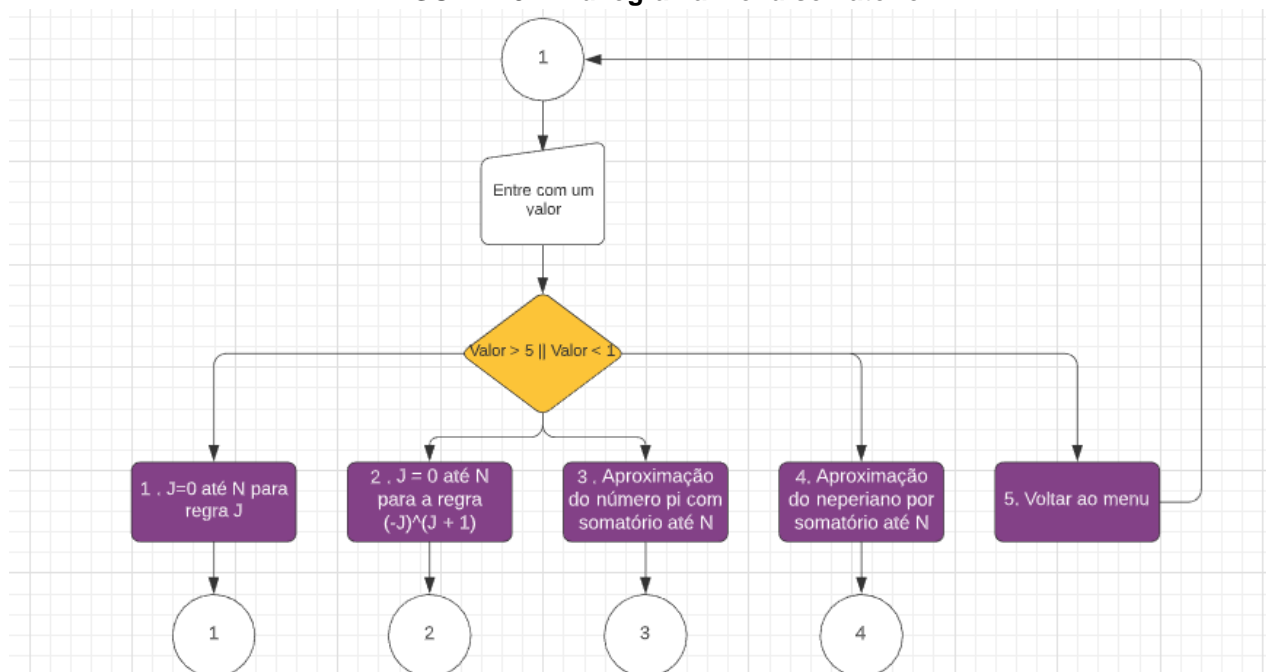
FIGURA 12 – Fluxograma menu principal



Fonte: Autores

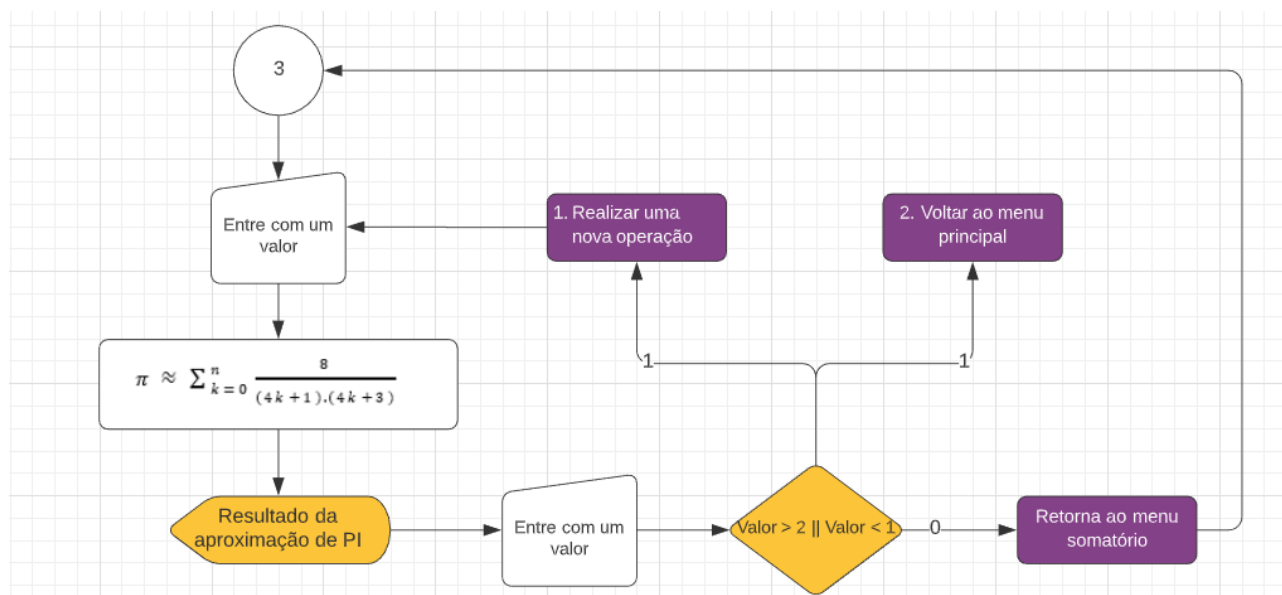
Como mostra o exemplo a abaixo, suponhamos que o usuário tenha optado pela opção número 3, onde deseja calcular a aproximação do número π (pi) com somatório até N, podemos observar que é solicitado uma entrada de dado para o usuário, onde ele insere o valor de N e o programa calcula de acordo com a fórmula exibindo o resultado, logo após o usuário pode optar por retornar ao menu principal ou realizar uma nova operação.

FIGURA 13 – Fluxograma menu somatório



Fonte: Autores

Figura 14- Fluxograma aproximação de PI com somatório até N



Fonte: Autores

5. RESULTADOS

Após realizar todas as análises e pesquisas necessárias para o desenvolvimento e colocar o projeto em prática, foi datado a necessidade de expor o projeto para que todos aqueles que necessitam dos cálculos executados nessa calculadora, estará disponível para download.

O projeto pode ser implantado em qualquer lugar e por qualquer pessoa, podendo ajudar com os problemas de cálculos encontrados no dia a dia, seja no trabalho, no colegial, na faculdade ou em casa.

6. DISCUSSÃO FINAL

O projeto obteve sua conclusão com todas as operações descritas funcionando, além de inserir funções para melhoria do código perante sua execução, podendo evitar falhas humanas. Contudo o projeto está disponível para *download*. Acompanhando também um vídeo interativo sobre uma situação em que o software poderia ser utilizado e um pouco sobre seu desenvolvimento, você pode acessá-lo através do QR Code abaixo, ou através do link:



<https://www.youtube.com/watch?v=022yn8E9k5U>

BIBLIOGRAFIA

Aitken, P. & Jones B. L. **C - Guia do Programador**. Editora Berkeley Brasil, 1994.

ARAUJO, Aldrovando L. A. **Fundamentos da matemática II**. Florianópolis: Inder Jeet Taneja, 2010. 131 p. (2ª Edição).

C for Loop. Disponível em: <https://www.programiz.com/c-programming/c-for-loop>. Acesso em: 20 mar. 2021.

GALVÃO, Gardênia de Oliveira. **Manual de Fluxogramas**: processos relacionados ao ecossistema de estímulo à inovação. Natal: Editora Ifrn, 2017.

GHEZZI, Carlo; JAZAYERI, Mehdi. **Conceitos de Linguagens de Programação**. Editora Campus, 1987.

LIPSCHUTZ, Seymour, LIPSON, Marc. **Matemática Discreta**. Porto Alegre: Bookman, 2004.

Mizrahi, V. V. **Treinamento em linguagem C** - Módulos 1 e 2. Editora McGraw-Hill, 1990.

PASSOS, Gilvan da Silva et al. **Análise Combinatória: teoria e aplicações para o ensino básico**.

2018. TANEJA, Inder Jeet; ARAUJO, Aldrovando Luiz. **Fundamentos da matemática II**. UFSC, 2006.

SOUZA, Elvis Gomes et al. **Probabilidade no Ensino Médio**: metodologia ativa como suporte. 2018.

STEWART, James. **Cálculo**. 7. ed. São Paulo: Cengage, 2013. 661 p. (1).