Texto

Descrição gerada automaticamente

**PROJETO INTERDISCIPLINAR**

**Conversor de decimais**

Conversão de decimal para as bases binário, hexadecimal e octal.

**Alunos:**

|  |  |
| --- | --- |
| **RGM** | **Nome** |
| 27422623 | Gabriel Gustavo Silva de Souza |
| 28031270 | Fabio Luiz Teixeira |

São Paulo

2023

**UNIVERSIDADE CRUZEIRO DO SUL**

**PROJETO INTERDISCIPLINAR**

**Programa de Base Decimal para outras bases**

Software de conversão

Trabalho apresentado como parte do requisito para aprovação na Disciplina de Projeto Interdisciplinar do curso de Análise e Desenvolvimento de Sistemas da Universidade Cruzeiro do Sul.

**Orientadores:** Prof. Vera Lucia Forbeck e Prof. Alvaro Andre Colombero Prado

São Paulo

2023**Sumário**

[1. Apresentação: 4](#_Toc68616856)

[1.1 Justificativa e Motivação 4](#_Toc68616857)

[1.2 Dados do Programa. 4](#_Toc68616858)

[2 Requisitos de Programação de Computadores 5-7](#_Toc68616859)

[3 Requisitos de Organização e Arquitetura de Computadores 7-9](#_Toc68616860)

[4 Consideração finais 9-10](#_Toc68616861)

[5 BIBLIOGRAFIA 11-12](#_Toc68616862)

# 1. Apresentação:

## 1.1 Justificativa e Motivação

Este tema fora escolhido para demonstrar os conhecimentos adquiridos nas aulas de Organização e Arquitetura de Computadores e Programação de Computadores por meio da construção e desenvolvimento de um software na linguagem de programação Python.

No ponto de vista acadêmico, por meio do programa desenvolvido a seguir, será apresentado o entendimento adquirido nas duas aulas citadas anteriormente com a criação de um software que em sua elaboração contém operações com Sistemas de Numeração, assim comprovando que os membros envolvidos adquiriram corretamente os conhecimentos ensinados nas duas aulas, sendo capazes de implementar os conhecimentos das duas matérias efetivamente.

## 1.2 Dados do Programa.

O programa foi criado para realizar cálculos pertinentes há conversão de bases decimal para as bases binário, octal e hexadecimal.

Quando mencionamos sistemas de numeração estamos nos referindo à utilização de um sistema para representar uma numeração, ou seja, uma quantidade. Sistematizar algo seria organizar, colocar em ordem, submeter à determinadas regras.

No dia a dia utilizamos o sistema decimal que se refere a 10 caracteres diferentes para representar os números (os dígitos de 0 a 9), mas existem várias bases numéricas e podemos converter usando regras matemáticas.

As bases numéricas que esse programa fará a conversão são, binário que se refere a 2 caracteres para representar os números (os dígitos de 0 a 1), octal que se refere a 8 caracteres diferentes para representar os números (os dígitos de 0 a 7) e o Hexadecimal que se refere a 16 caracteres utilizando também letras para representar dois dígitos (os dígitos de 0 a 9 e as letras A = 10, B = 11, C = 12, D = 13, E = 14, F = 15)

Para convertemos um número decimal para outras bases, realiza-se a divisão sucessiva pela base que será convertida, por exemplo uma base binaria a divisão será por 2, o resultado da divisão é obtido pelo último quociente e o agrupamento dos restos da divisão, na ordem do último resto obtido até o primeiro.

Esse programa tem a finalidade de automatizar os cálculos de conversão de bases decimais.

# 2 Requisitos de Programação de Computadores

Para atingirmos um programa eficiente e responsivo utilizamos os cumprimentos e requisitos referente a disciplina de programação de computadores, segue abaixo explicações referente a codificação criada. Utilizamos: entrada de dados(input), condições, operadores lógicos e expressões lógicas.

O comando de atribuição entrada de dados foi utilizado neste bloco de código onde, o usuário realiza o input de qual das opções do menu ele deseja executar, escolhendo a opção 2 o programa irá mostrar os detalhes do software, os desenvolvedores, versão etc, a opção 3 finaliza o programa e a opção 1, que é a principal, solicitará ao usuário que digite o valor que deseja realizar a conversão de decimal para as bases binário, hexadecimal e octal (Contendo validações).

Escolhendo qual a base que deseja realizar a conversão.

1 – Para Binário.

2 – Para Octal.

3 – Para Hexadecimal.

Texto

Descrição gerada automaticamente

Após utilizamos os operadores lógicos de tomada de decisão:

**IF** - Que se trata de uma estrutura de condição que permite avaliar uma expressão e, de acordo com seu resultado, executar uma determinada ação.

**ELSE** - Que se trata de uma estrutura de condição utilizada como SENÃO e com esta, definiremos o bloco de instrução a ser executado todas as vezes que a expressão definida for igual a falso.

**ELIF** – Que se trata de uma estrutura de condição queserve para verificar outra condição caso a condição do IF seja falsa.

**WHILE** – Que se trata de uma estrutura de condição que faz com que um conjunto de instruções seja executado enquanto uma condição é atendida. Quando o resultado dessa condição passa a ser falso, a execução do loop é interrompida.

Utilizamos as seguintes expressões lógicas na codificação:

|  |  |
| --- | --- |
| **Operador** | **Operação** |
| **==** | *Igual* |
| **% ou mod** | *Resto de uma divisão inteira* |
| **/** | *Divisão* |
| **+** | *Soma* |
| **>** | *Maior* |

Após o input do usuário e a escolha da base que ele deseja realizar a conversão, os blocos de códigos trabalham da seguinte maneira:

**Binário**

Se caso a base escolhida for a 1 – Binário, ele entrará no If de base ==1 e realizará a seguinte bloco de código:

Texto

Descrição gerada automaticamente

**Octal.**

Se caso a base escolhida for a 2 – Octal ele entrará no elif de base ==2 e realizará a seguinte bloco de código:

Texto

Descrição gerada automaticamente

**Hexadecimal.**

Se caso a base escolhida for a 3 – Hexadecimal ele entrará no elif de base ==3 e realizará a seguinte bloco de código:

Texto

Descrição gerada automaticamente

# 3 Requisitos de Organização e Arquitetura de Computadores

# Conversão Decimal Para Binário

A conversão de decimal para binário (ou seja da base 10 para a base 2), consiste em dividir progressivamente o valor decimal por 2, obtendo-se um resultado e um resto. De referir que o resultado em cada iteração terá sempre o valor de 0 ou 1. Deve-se dividir o número até que o quociente da divisão seja igual a 0 (zero).

Diagrama

Descrição gerada automaticamente

Resultado: 22 (10) > 10110 (2)

**Exemplo:**

(2461)d

2461/2 = 1230, resto = 1

1230/2 = 615, resto = 0

615/2 = 307, resto = 1

307/2 = 153, resto = 1

153/2 = 76, resto = 1

76/2 = 38, resto = 0

38/2 = 19, resto = 0

19/2 = 9, resto = 1

9/2 = 4, resto = 1

4/2 = 2, resto = 0

2/2 = 1, resto = 0

1/2 = 0, resto = 1

100110011101. Este é o equivalente binário ao número decimal 2461

No programa python ficará dessa forma, o mesmo exemplo:

Texto

Descrição gerada automaticamente

# Conversão Decimal Para Octal

A conversão de decimal para octal (ou seja da base 10 para a base 8), consiste em dividir progressivamente o valor decimal por 8, obtendo-se um resultado e um resto. De referir que o resultado em cada iteração terá sempre um valor menor que 7.

Diagrama

Descrição gerada automaticamente

Resultado: 407(10) > 627(8)

Exemplo:

(325)d

325/8 = 40, resto 5

40/8 = 5, resto 0

5/8 = 0, resto 5

**(505)o** Este é o octal equivalente ao decimal 325

No programa python ficará dessa forma, o mesmo exemplo:  
Texto

Descrição gerada automaticamente

# Conversão Decimal PARA Hexadecimal

A conversão de decimal para hexadecimal(ou seja da base 10 para a base 16), consiste em dividir progressivamente o valor decimal por 16, obtendo-se um resultado e um resto. Não esquecer que o sistema hexadecimal utiliza os símbolos: 0,1,2,3,4,5,6,7,8,9 do sistema decimal e as letras A,B,C,D,E,F.

(Equivalências: A=10,B=11,C=12,D=13,E=14,F=15)

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Resultado: 53120(10) > CF80(16)

Exemplo

(20034)d

20034/16 = 1252, resto 2

1252/16 = 78, resto 4

78/16 = 4, resto 14

4/16 = 0, resto 4

(4E42)h Este é o hexadecimal equivalente ao número decimal 20034

No programa python ficará dessa forma, o mesmo exemplo:

Texto

Descrição gerada automaticamente

# 4 Consideração finais

**Facilidade:** A Facilidade que tivemos na construção dos códigos, foi por conta de já ter visto alguns exercícios desse tipo na sala de aula, como por exemplo: como construir uma sequência de **IF, ELIF** ou **ELSE** assim como validação dos próprios.

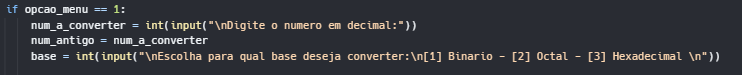
**Exemplo de sequência de IF, ELIF e ELSE:**

Texto

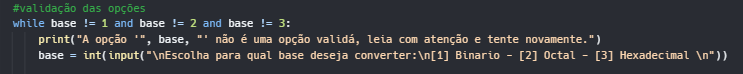
Descrição gerada automaticamente

**Dificuldade:** A Dificuldade que tivemos, é em como estruturar o código, com a quantidade de variáveis que tinha, em como organizar elas no caso.

Exemplo:



Assim também como utilizar o comando “**While”**, aonde aprendemos pouco dele em sala de aula, mas ao utilizar não achamos tanta dificuldade.



A maior dificuldade foi em montar a base Hexadecimal, no código tem muita validação de **IF,** a forma que encontramos foi essa, especificar individualmente cada numeral em sua respectiva letra.

Texto

Descrição gerada automaticamente

Assim também em montar outras variáveis dentro do Hexadecimal, para então ele fazer o cálculo e a substituição dos numerais.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

# 5 BIBLIOGRAFIA

**Arquitetura e Organização de Computadores:**

# MONTEIRO, M. A. Introdução a Organização de Computadores. 5. ed. Brasil: Ltc-Livros Técnicos e Cientificos, 2007.

STALLINGS, W. S. Arquitetura e Organização de Computadores: Projeto Para o Desempenho. 8. ed. Sao Paulo: Prentice Hall, 2010.

TANENBAUM, A. S. Organização Estruturada de Computadores. 5. ed. Brasil: Prentice Hall Brasil, 2007.

BAER, J. L. Arquitetura de Microprocessadores: do simples pipeline ao multiprocessador em chip. Rio de Janeiro: LTC, 2013. (e-book).

HENNESSY, J. L.; PATTERSON, D. A. Computer organization and design: the hardware/software interface. Rio de Janeiro: Elsevier, 2012. Disponível em: https://ict.iitk.ac.in/wpcontent/uploads/CS422-Computer-ArchitectureComputerOrganizationAndDesign5thEdition2014.pdf Acesso em: 26 nov.2020.

MARTINS, C. A. P. S. Arquitetura de computadores: educação, ensino e Aprendizado. SBC, 2012. Disponível em: http://siaibib01.univali.br/pdf/Ebook%20Arquitetura%20de%20computadores- %20educa%C3%A7%C3%A3o,%20ensino%20e%20aprendizado.pdf. Acesso em: 26 nov. 2020.

PAIXÃO, R. R. Arquitetura de computadores: PCs. São Paulo: Erica, 2014.(e-book).

WEBER, R. F. Fundamentos de arquitetura de computadores. 4. Porto Alegre Bookman 2012 (ebook).

**Programação de Computadores:**

FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de Programacao: A Construção de Algoritmos e Estrutura de Dados. 3. ed. São Paulo: Makron Books do Brasil, 2005.

MANZANO, J. A. N. G. Algoritmos: Lógica Para Desenvolvimento de Programação. 20. ed. Sao Paulo: Érica, 2007.

VILARIM, G. O. Algoritmos: Programação para Iniciantes. Rio de Janeiro: Ciência Moderna, 2004.

ANA FERNANDA GOMES ASCENCIO; EDILENE

APARECIDA VENERUCHI DE CAMPOS Fundamentos da Programação de Computadores São Paulo: Pearson, 2012

(e-book).

ANA FERNANDA GOMES ASCENCIO; EDILENE

APARECIDA VENERUCHI DE CAMPOS Fundamentos da Programação de Computadores São Paulo: Pearson, 2012

(e-book)