

PROJETO INTERDISCIPLINAR

CONVERSOR DE BASES NUMÉRICAS

Python Para Conversão de Bases

RGM	Nome
38989158	Abraão Vieri Souza de Oliveira
38060558	Gustavo Ícaro Camargos Oliveira
38746387	Guilherme Augusto Caldeira Passos
38649501	Guilherme Soares
36969966	Vitor Douglas da Silva

São Paulo / São Paulo

2024

UNIVERSIDADE CRUZEIRO DO SUL

PROJETO INTERDISCIPLINAR

CONVERSOR DE BASES NUMÉRICAS

Python Para Conversão de Bases

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

Orientadores:

Prof. Giulio Guiyti Rossignolo Suzumuraz

Prof. Antonio Eduardo Marques Da Silva

São Paulo / São Paulo

2024

Sumário

1. APRESENTAÇÃO:	4
1.1 Justificativa e Motivação	4
1.2 Dados do Sistema.	4
2 ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES	6
3 PROGRAMAÇÃO DE COMPUTADORES	8
4 CONSIDERAÇÕES FINAIS	9
5 BIBLIOGRAFIA	10

1. APRESENTAÇÃO:

1.1 Justificativa e Motivação

A escolha do sistema de conversão de bases numéricas foi motivada pela importância prática dessa ferramenta em áreas como matemática, ciência da computação e engenharia.

Além disso, a oportunidade de explorar o potencial do framework Flask em Python para desenvolver uma aplicação acessível e fácil de usar foi um incentivo adicional. O objetivo foi criar uma solução simples e eficaz para um problema comum, ao mesmo tempo em que se contribui para a comunidade de desenvolvedores por meio do compartilhamento de código-fonte aberto.

Os projetos interdisciplinares são como um espelho da vida profissional, onde os desafios exigem abordagens que vão além das disciplinas acadêmicas. Participar desses projetos ajuda os estudantes a desenvolver habilidades-chave para o mercado de trabalho atual, como colaboração eficaz, comunicação clara e resolução criativa de problemas.

Eles também incentivam a aplicação prática do conhecimento teórico, fortalecendo a compreensão dos conceitos acadêmicos e preparando os alunos para os desafios futuros. Além disso, promovem uma visão integrada dos problemas, essenciais em um mundo onde a resolução de desafios exige uma abordagem multifacetada.

A participação nesses projetos proporciona uma valiosa experiência de trabalho em equipe, valorizada pelos empregadores. Em suma, destacar a importância dos projetos interdisciplinares na formação profissional é crucial para preparar os alunos para os desafios do mundo real, desenvolvendo habilidades essenciais e promovendo uma cultura de colaboração e inovação.

1.2 Dados do Sistema.

O sistema desenvolvido é uma aplicação web utilizando o poderoso framework Flask em Python. Ele oferece uma funcionalidade crucial para o pedido: a capacidade de converter números entre diferentes bases numéricas, como binário, octal, decimal e hexadecimal.

Quando um usuário acessa a página principal do sistema, ele é recebido por um formulário intuitivo. Neste formulário, o usuário pode inserir o número a ser convertido, bem como especificar a base inicial e a base final desejadas. Isso proporciona uma experiência amigável e direta, mesmo para aqueles que não possuem conhecimentos avançados em programação ou matemática.

A função converter é a principal de toda operação. Ela recebe como entrada o número a ser convertido, a base inicial e a base final. Primeiro, converte o número para a base decimal, que serve como intermediária. Em seguida, realiza a conversão da base decimal para a base final desejada. Isso é feito através de uma série de instruções condicionais, dependendo da base final escolhida pelo usuário. Se a base final for binária, o número é convertido para binário; se for octal, para octal; e assim por diante.

É importante destacar que o sistema é robusto e seguro. Ele inclui tratamento de erros, como a verificação se o número e as bases fornecidas são válidos. Isso garante que o sistema funcione de forma confiável, mesmo diante de entradas incorretas.

Estudantes, profissionais e entusiastas encontram nesse sistema uma ferramenta valiosa para realizar conversões entre bases numéricas com facilidade e precisão. Elimina a necessidade de cálculos manuais, economizando tempo e reduzindo erros. Além disso, a flexibilidade do Flask permite que o sistema seja facilmente personalizado e expandido para atender às necessidades específicas dos usuários.

Em resumo, este sistema representa uma síntese entre funcionalidade poderosa e acessibilidade. Ao oferecer uma solução eficaz para uma tarefa matemática complexa, ele demonstra o potencial transformador da tecnologia na simplificação de processos e na melhoria da experiência do usuário.

2. ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

As bases numéricas desempenham um papel fundamental na organização e arquitetura da computação. Elas são essenciais para representar e manipular informações de forma eficiente nos sistemas computacionais. As bases binária, octal, decimal e hexadecimal são usadas para representar dados e são importantes na lógica digital. A compreensão das bases numéricas é crucial para a conversão entre sistemas de numeração, manipulação eficiente de dados, transmissão correta de dados em redes de computadores, e funcionamento de processadores e dispositivos de armazenamento. Em resumo, as bases numéricas são essenciais para a representação, manipulação e processamento eficiente de dados em sistemas computacionais. A conversão de bases numéricas é um conceito importante na arquitetura da computação, pois nos ajuda a entender como os computadores lidam com diferentes tipos de números.

Vamos começar pelo sistema decimal, que é o que estamos mais acostumados, usando os dígitos de 0 a 9. Por exemplo, o número 235 em decimal representa "2 vezes 100, somado a 3 vezes 10, somado a 5 vezes 1".

Já o sistema binário, muito usado em computação, usa apenas os dígitos 0 e 1. Por exemplo, o número binário 1010 representa "1 vezes 8, somado a 0 vezes 4, somado a 1 vez 2, somado a 0 vezes 1" em decimal.

O sistema octal usa os dígitos de 0 a 7, e cada dígito octal pode ser representado por uma sequência de três dígitos binários. Por exemplo, o número octal 52 representa "5 vezes 8, somado a 2 vezes 1" em decimal.

Por fim, o sistema hexadecimal usa os dígitos de 0 a 9 e as letras de A a F. Ele é muito útil na programação por ser uma forma concisa de representar valores binários. Por exemplo, o número hexadecimal AE representa "10 vezes 16, somado a 14 vezes 1" em decimal.

Entender como converter entre essas bases numéricas nos ajuda a compreender como os dados são armazenados e processados dentro dos computadores. Além disso, essa habilidade é útil ao lidar com programação de baixo nível e comunicação com dispositivos eletrônicos.

e) Binário para Hexadecimal

$$10110010011110_2 \rightarrow X_{16}$$

0010	1100	1001	1110
2	12	9	14
2	C	9	E

$$\therefore 2C9E_{16}$$

c) Decimal para Binário

$$224_{10} \rightarrow X_2$$

$$\begin{array}{r}
 224 \div 2 \\
 \hline
 112 \div 2 \\
 \hline
 56 \div 2 \\
 \hline
 28 \div 2 \\
 \hline
 14 \div 2 \\
 \hline
 7 \div 2 \\
 \hline
 3 \div 2 \\
 \hline
 1 \div 2 \\
 \hline
 1 \div 2 \\
 \hline
 0
 \end{array}$$

LSD \rightarrow 0 MSD \rightarrow 1 1

$$224_{10} = 11100000_2$$

b) Hexadecimal para Decimal

$$2DBF_{16} = X_{10}$$

16^3	16^2	16^1	16^0
2	D	B	F

$$\begin{aligned}
 & 2 \times 16^3 + D \times 16^2 + B \times 16^1 + F \times 16^0 = \\
 & 2 \times 16^3 + 13 \times 16^2 + 11 \times 16^1 + 15 \times 16^0 = \\
 & 2 \times 4096 + 13 \times 256 + 11 \times 16 + 15 \times 1 = \\
 & 8192 + 3328 + 176 + 15 = 11711_{10}
 \end{aligned}$$

d) Decimal p/ Hexadecimal

$$3876_{10} \rightarrow X_{16}$$

$$\begin{array}{r}
 3876 \div 16 \\
 \hline
 242 \div 16 \\
 \hline
 15
 \end{array}$$

$$F24_{16}$$

e) Binário para Hexadecimal

$$10110010011110_2 \rightarrow X_{16}$$

0010	1100	1001	1110
2	12	9	14
2	C	9	E

$$\therefore 2C9E_{16}$$

3. PROGRAMAÇÃO DE COMPUTADORES

@app.route('/'): Este é um decorador em Flask que é usado para vincular a função `index()` a uma URL específica, neste caso, a URL principal.

def: É usada para definir uma função. Uma função é um bloco de código que executa uma tarefa específica quando chamado. Ao usar `def`, você está criando uma função com um nome específico que pode aceitar argumentos e executar instruções quando é chamada.

if: É executado se a condição especificada for avaliada como verdadeira.

else: é executado se a condição especificada no `if` for avaliada como falsa.

elif: A palavra-chave `elif` é uma abreviação de "else if". Ela permite verificar múltiplas condições em sequência

```
7
8 # Criando uma função para converter as bases numéricas
9 def converter(numero, base_inicial, base_final):
10     try:
11         # Convertendo para decimal primeiro
12         decimal = int(numero, base_inicial)
13
14         # Convertendo de decimal para a base final
15         if base_final == 2:
16             resultado = bin(decimal)
17         elif base_final == 8:
18             resultado = oct(decimal)
19         elif base_final == 10:
20             resultado = decimal
21         elif base_final == 16:
22             resultado = hex(decimal)
23
24         # Retorna o valor da conversão se estiver correta, se não, mostra um aviso de erro
25         return resultado
26     except ValueError:
27         return "Erro: Certifique-se de que o número e as bases estão corretos."
```


4. CONSIDERAÇÕES FINAIS

No projeto interdisciplinar, foi encontrada uma maior facilidade de utilizar HTML para criar a estrutura de uma página web. No entanto, ao explorar o Flask, um framework em Python, um desafio maior surgiu. Foi preciso compreender conceitos como rotas e funções de visualização para processar as solicitações dos usuários. Isso exigiu prática e paciência, semelhante a aprender a tocar um novo instrumento musical.

Dentro do Flask, conceitos como loops "def" e "while" adicionaram outra camada de complexidade. Embora sejam usados para controlar o fluxo de execução do código, demandam tempo e dedicação para serem dominados, como aprender uma nova língua e compreender sua gramática.

5. BIBLIOGRAFIA

CRUZ, Felipe. Python:Escreva seus primeiros programas. Casa do Código: São Paulo, 2015.

MACHADO, Francis Berenger; MAIA, Luiz Paulo. Arquitetura de Sistemas Operacionais. 4. ed. Rio de Janeiro: LTC, 2007.

MENEZES, Nilo Ney Coutinho. Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes. 2. Ed. São Paulo: Novatec, 2014.

SILVA, João. Aprendendo sobre a história brasileira: a República Velha. YouTube, 10 de janeiro de 2020. 15min30s. Disponível em: <<https://youtu.be/K2ej4z8Mbg?si=cD15-u8uwuqtFOx2>>

GEEKSFORGEEKS. Program for Shortest Job First (or SJF) CPU Scheduling | Set 1 (Non-preemptive). Disponível em: <https://www.geeksforgeeks.org/program-for-shortest-job-first-or-sjf-cpu-scheduling-set-1-non-preemptive/>. Acessado em: 17 jul. 2019.

GEEKSFORGEEKS. Program for Best Fit algorithm in Memory Management. Disponível em: <https://www.geeksforgeeks.org/program-best-fit-algorithm-memory-management/>. Acessado em: 24 jul. 2019. SHAH, Ketan; MITRA, Anirban; MATANI, Dhruv. An O (1) algorithm for implementing the LFU cache eviction scheme. v. 1, p. 1-8, 2010.