

	Unifeso: Centro Universitário Serra dos Órgãos	
	Disciplina: Estrutura de Dados e Paradigmas	Código: 1305100006
	Professor(a): Alexandra Raibolt	
	Semestre: 2024.1	
	Curso: Bacharelado em Ciência da Computação	

Trabalho 1 — Avaliação 2

Leia as Instruções:

- Este trabalho deve ser feito em grupo de 5 (cinco) integrantes.
- Compacte (.zip ou .rar) as respostas das questões.
- Realize o envio do arquivo compactado composto pelos nomes dos(as) alunos(as) e matrícula até 25/06/2024 às 23:59 no Ambiente Virtual do Unifeso (CANVAS).
- Apenas 1 (um) integrante de cada dupla precisa realizar a submissão no Ambiente Virtual do Unifeso (CANVAS).

Questão 1 Crie um algoritmo em Python que implemente uma Tabela Hash para armazenar nomes, utilizando o Paradigma de Programação Orientada a Objetos. **(10.0 pontos)**

Requisitos:

1. Implemente uma classe chamada TabelaHash que será responsável por armazenar os nomes na tabela hash.
2. Dentro da classe TabelaHash, implemente os métodos necessários para adicionar, buscar e remover nomes na tabela.
3. Crie uma função chamada hash_divisao que recebe um nome como parâmetro e retorna o índice da tabela hash utilizando o Método da Divisão.
4. Crie uma função chamada hash_dobra que recebe um nome como parâmetro e retorna o índice da tabela hash utilizando o Método da Dobra.
5. Crie uma função chamada hash_multiplicacao que recebe um nome como parâmetro e retorna o índice da tabela hash utilizando o Método da Multiplicação.
6. Implemente as classes necessárias para o tratamento de colisões escolhido pelo usuário: Encadeamento Exterio, Encadeamento Interior ou Endereçamento Aberto (Tentativa Sequencial).
7. No programa principal, solicite ao usuário qual tipo de função hash e tratamento de colisões ele deseja utilizar.
8. Crie um loop que permite ao usuário adicionar, buscar e remover nomes na tabela hash até que ele decida sair do programa.
9. Exiba a tabela hash atualizada após cada operação.

Observações:

- Certifique-se de tratar possíveis erros de entrada do usuário.
- Utilize boas práticas de programação, incluindo comentários explicativos no código.
- Teste seu algoritmo com diferentes entradas e verifique se ele está funcionando corretamente.