

A solução desenvolvida implementa um sistema modular para geração, armazenamento, leitura e análise de registros de alunos utilizando dois modelos distintos de organização de arquivos: registros de tamanho fixo e registros de tamanho variável, com suporte adicional à simulação de espalhamento (fragmentação). O objetivo é comparar as características, vantagens e limitações de cada abordagem, considerando também um limite de tamanho máximo de bloco, configurado pelo usuário em tempo de execução.

O projeto é dividido em módulos independentes, facilitando a manutenção, testes individuais e reutilização do código:

`gerar_dados.py`: gera automaticamente um conjunto fictício de alunos contendo os campos matrícula, nome, CPF, curso, nome da mãe, nome do pai, ano de ingresso e CA.

`gravar_fixo.py`: grava cada registro com comprimento padronizado de 170 bytes, completando com caracteres de padding (#). Esse formato simula sistemas de armazenamento de acesso direto.

`gravar_variavel.py`: grava cada registro como uma linha de tamanho variável, utilizando | como delimitador e permitindo economia de espaço.

`leitura.py`: contém rotinas para ler tanto registros fixos (via leitura posicional) quanto variáveis (via parsing), reconstruindo os campos originais.

`analise.py`: calcula tamanho total do arquivo, quantidade de registros e tamanho médio por registro, permitindo comparar a eficiência entre os dois modelos.

`main.py`: gerencia a interação com o usuário, permitindo escolher o modelo de armazenamento, informar o tamanho máximo do bloco e ativar ou não o espalhamento.

Ao iniciar a aplicação, o usuário informa o tamanho máximo do bloco em bytes e seleciona o formato desejado. No caso de registros variáveis, é possível ativar o modo de espalhamento, que simula um cenário em que registros podem ultrapassar o limite do bloco, fragmentando-se em múltiplas áreas de armazenamento. Esse comportamento permite observar como sistemas reais lidam com registros que não cabem inteiramente em um único bloco.

Após a gravação dos dados, o programa realiza a leitura dos arquivos gerados e exibe uma prévia dos registros reconstruídos. Em seguida, executa uma análise quantitativa do arquivo, permitindo comparar espaço ocupado e eficiência de acesso.

A solução final atende aos requisitos da atividade por:

Implementar ambos os modelos de organização fixo e variável.

Considerar limites de armazenamento via tamanho do bloco.

Permitir simulação de espalhamento de registros.

Oferecer leitura e reconstrução correta dos dados armazenados.

Oferecer análise objetiva do espaço consumido e eficiência.

Seguir uma abordagem modular, favorecendo organização, entendimento e testes.

Com isso, o trabalho demonstra na prática como diferentes técnicas de organização de arquivos afetam a forma como dados são armazenados, acessados e manipulados em sistemas reais.