

Tema: Biblioteca

Exercícios de Programação em Linguagem C Simulando BDs em Vetores Tema: Biblioteca

Este documento apresenta um conjunto de estruturas de structs e vetores de structs para simular um banco de dados de uma biblioteca. O objetivo é treinar o uso de múltiplos vetores em C, explorando relações entre diferentes entidades e resolvendo exercícios que exigem raciocínio lógico e cruzamento de dados.

1. Declaração das Structs

Autor

Representa um autor cadastrado na biblioteca. Cada autor tem um código de nacionalidade.

- idAutor: identificador único do autor (unsigned int)
- codNacionalidade: código numérico representando a nacionalidade (unsigned int)
- anoNascimento: ano de nascimento do autor (unsigned int)

Editora

Representa editoras. Usa-se código de país e ano de fundação, tudo numérico.

- idEditora: identificador único da editora
- codPais: código do país de origem (unsigned int)
- anoFundacao: ano de fundação (unsigned int)

Categoria

Categorias de livros representadas por códigos numéricos.

- idCategoria: identificador único da categoria
- codArea: código da área do conhecimento (unsigned int)

Livro

Representa obras catalogadas. Relações por ids para autor, editora e categoria.

- idLivro: identificador único do livro
- idAutor: id do autor (chave estrangeira para Autor)
- idEditora: id da editora (chave estrangeira)
- idCategoria: id da categoria (chave estrangeira)
- anoPublicacao: ano de publicação (unsigned int)
- numPaginas: número de páginas de páginas (unsigned int)

Exemplar

Representa cópias físicas de livros. Estado de conservação codificado numericamente.

- idExemplar: identificador único do exemplar
- idLivro: id do livro ao qual o exemplar pertence
- codEstado: código do estado de conservação
(1=novo,2=bom,3=regular,4=ruim)

PessoaAluno

Usuários do tipo aluno.

- idAluno: identificador único do aluno
- codCurso: código do curso em que está matriculado
- anoIngresso: ano de ingresso no curso

PessoaProfessor

Usuários do tipo professor.

- idProfessor: identificador único do professor
- codDepartamento: código do departamento
- anoContratacao: ano de contratação

Funcionario

Funcionários da biblioteca.

- idFuncionario: identificador único
- codCargo: código do cargo (unsigned int)
- anoAdmissao: ano de admissão (unsigned int)

Emprestimo

Registro de empréstimo de exemplar para pessoa (aluno ou professor).

- idEmprestimo: identificador do empréstimo
- idExemplar: exemplar emprestado
- idPessoa: id da pessoa que pegou (aluno ou professor)
- tipoPessoa: 0=aluno, 1=professor
- anoEmprestimo: ano do empréstimo (unsigned int)
- anoDevolucao: ano da devolução; 0 se ainda não devolvido (unsigned int)

Reserva

Registro de reserva de livro (por pessoa).

- idReserva: identificador da reserva
- idLivro: livro reservado
- idPessoa: pessoa que reservou
- tipoPessoa: 0=aluno, 1=professor
- anoReserva: ano da reserva

SalaEstudo

Salas de estudo disponíveis.

- idSala: identificador da sala
- capacidade: capacidade máxima (unsigned int)
- codLocalizacao: código da localização/setor (unsigned int)

ReservaSala

Reserva de sala por pessoa.

- idReservaSala: identificador
- idSala: sala reservada
- idPessoa: pessoa que reservou
- tipoPessoa: 0=aluno, 1=professor
- anoReserva: ano da reserva
- qtdPessoas: quantidade de pessoas na reserva

Multa

Multas aplicadas por atraso.

- idMulta: identificador da multa
- idEmprestimo: empréstimo associado
- valor: valor da multa (float)
- pago: 0=não, 1=sim

Evento

Eventos organizados pela biblioteca; sem descrição textual.

- idEvento: identificador do evento
- codTipoEvento: código do tipo de evento (unsigned int)
- anoEvento: ano em que ocorreu

ParticipacaoEvento

Participação de pessoa em evento.

- idParticipacao: identificador

- idEvento: evento associado
- idPessoa: pessoa participante
- tipoPessoa: 0=aluno, 1=professor

Doacao

Registro de doações de livros (quem doou é representado por código de pessoa ou código externo).

- idDoacao: identificador
- idLivro: livro doado
- codDoador: código do doador (0=externo, >0 = idPessoa)
- anoDoacao: ano da doação

RevisaoExemplar

Revisões técnicas feitas em exemplares por funcionários.

- idRevisao: identificador
- idExemplar: exemplar revisado
- idFuncionario: funcionário responsável
- anoRevisao: ano da revisão

SugestaoCompra

Sugestões de compra; títulos.

- idSugestao: identificador
- idPessoa: quem sugeriu
- tipoPessoa: 0=aluno, 1=professor
- codTemaLivro: código do tema do livro sugerido
- anoSugestao: ano da sugestão

RecomendacaoLeitura

Recomendações feitas por professores para livros existentes.

- idRecomendacao: identificador
- idProfessor: professor que recomenda

- idLivro: livro recomendado
- anoRecomendacao: ano

2. Observações de Semântica e Integridade

- IDs referenciam registros em outras tabelas (foreign keys). Sempre trate comparações por igualdade de IDs.
- Campos de ano com valor 0 indicam ausência (por exemplo, anoDevolucao==0 significa NÃO devolvido).
- Tipos de pessoa (tipoPessoa) sempre 0 para aluno e 1 para professor nas tabelas onde aplicável.
- Estados e códigos (codEstado, codTipoEvento, codTemaLivro, codLocalizacao, etc.) são inteiros que representam categorias.
- Não há campos textuais: qualquer dado que seria texto foi substituído por um código inteiro correspondente.

3. Exercícios

Todos os exercícios recebem as quantidades de structs presentes em cada vetor passado como parâmetro.

Exercício 1: Alunos sem empréstimos no ano

Nome da função: alunosSemEmprestimosAno

Parâmetros: vetor PessoaAluno, vetor Emprestimo, unsigned int ano.

Retorno: inteiro quantidade de alunos que NÃO fizeram empréstimos no ano indicado.

Descrição: considerar apenas empréstimos cujo anoEmprestimo == ano. Use quantificador universal para 'nenhum empréstimo'.

Exercício 2: Livros sem exemplares em bom estado

Nome da função: livrosSemExemplarBom

Parâmetros: vetor Livro, vetor Exemplar, vetor Categoria (para filtrar por categoria id opcional).

Retorno: inteiro quantidade de livros de uma dada categoria que não possuem EXEMPLARES com codEstado==1 (novo) ou 2 (bom).

Descrição: se categoria não for especificada (idCategoria==0), considerar todos os livros.

Exercício 3: Professores que foram a todos os eventos

Nome da função: professoresEmTodosEventos

Parâmetros: vetor PessoaProfessor, vetor Evento, vetor ParticipacaoEvento, unsigned int ano.

Retorno: inteiro quantidade de professores que participaram de TODOS os eventos daquele ano.

Descrição: para cada professor verificar participação em cada evento do ano (quantificador universal sobre eventos).

Exercício 4: Reservas não atendidas por falta de exemplar

Nome da função: reservasNaoAtendidas

Parâmetros: vetor Reserva, vetor Livro, vetor Exemplar, vetor Emprestimo.

Retorno: inteiro quantidade de reservas que, no ano da reserva, não puderam ser atendidas porque todos os exemplares do livro estavam emprestados.

Descrição: para cada reserva contar se existia pelo menos um exemplar disponível no ano (i.e., exemplar do livro com nenhum empréstimo ativo no ano).

Exercício 5: Salas sempre lotadas por ano

Nome da função: salasSempreLotadasAno

Parâmetros: vetor SalaEstudo, vetor ReservaSala, unsigned int ano, vetor PessoaAluno, vetor PessoaProfessor.

Retorno: inteiro quantidade de salas cuja todas as reservas daquele ano tiveram qtdPessoas == capacidade da sala.

Descrição: considerar reservas de alunos e professores; use quantificador 'para todas as reservas'.

Exercício 6: Exemplares revisados por múltiplos funcionários

Nome da função: exemplaresRevisadosPorVarios

Parâmetros: vetor RevisaoExemplar, vetor Exemplar, vetor Funcionario, vetor

Livro, vetor Categoria.

Retorno: inteiro quantidade de exemplares que tiveram revisões realizadas por pelo menos 2 funcionários diferentes.

Descrição: cruzar revisões e exemplares; categoria permite filtrar por área se desejado (0 = sem filtro).

Exercício 7: Usuários sem multas e com empréstimos devolvidos

Nome da função: usuariosSemMultasComDevolucao

Parâmetros: vetor PessoaAluno, vetor PessoaProfessor, vetor Empréstimo, vetor Multa, vetor Doacao, vetor Pagamento (opcional).

Retorno: inteiro quantidade de pessoas (alunos e professores) que NÃO possuem multas associadas e que devolveram TODOS os exemplares emprestados no ano indicado.

Descrição: aplicar quantificador universal sobre empréstimos do usuário; usar Multa para excluir quem possui multas (pago==0).

Exercício 8: Sugestões atendidas por doação/compra

Nome da função: sugestoesAtendidasPorAquisição

Parâmetros: vetor SugestaoCompra, vetor Livro, vetor Doacao, vetor CompraSimulada (pode ser representada por Pedido+ItemPedido), vetor Editora, vetor Autor, vetor Categoria.

Retorno: inteiro quantidade de sugestões cuja obra sugerida foi posteriormente adquirida pela biblioteca (via doação ou compra) e que tenha correspondência por autor e categoria.

Descrição: correspondência aproximada via códigos: codTemaLivro deve corresponder a categoria ou autor codificado; considere ano como referência temporal (sugestao.anoSugestao < anoAquisição).

Exercício 9: Professores que recomendaram apenas livros de sua editora preferida

Nome da função: professoresRecomendamEditora

Parâmetros: vetor Professor, vetor RecomendacaoLeitura, vetor Livro, vetor Editora, vetor Autor, vetor ParticipacaoEvento, vetor Doacao, vetor CompraSimulada.

Retorno: inteiro quantidade de professores que recomendaram exclusivamente livros publicados por uma mesma editora (i.e., todas as suas recomendações apontam para livros de um único idEditora).

Descrição: verificar por professor o conjunto de editora dos livros recomendados ser singleton.

Exercício 10: Relatório complexo de acervo

Nome da função: `relatorioAcervoCompleto`

Parâmetros: todos os vetores do BD (Autor, Editora, Categoria, Livro, Exemplar, PessoaAluno, PessoaProfessor, Funcionario, Emprestimo, Reserva, SalaEstudo, ReservaSala, Multa, Evento, ParticipacaoEvento, Doacao, RevisaoExemplar, SugestaoCompra, RecomendacaoLeitura, ...).

Retorno: estrutura sumarizada (p.ex., counts por categoria, percentuais de exemplares em bom estado, média de revisões por exemplar).

Descrição: é um exercício aberto para combinar/agregar diversas métricas; espere usar quantificadores universais e existenciais simultaneamente.

4. Conjunto de Dados Hardcoded:

Abaixo um conjunto de vetores com valores numéricos para teste. Você pode copiar/colar no código C (main) e usar para validar suas funções.

// Exemplos reduzidos (valores totalmente numéricos; campos textuais foram convertidos em códigos)

```
Autor autores[3] = {  
    {1, 10, 1950}, {2, 20, 1910}, {3, 10, 1940}  
};
```

```
Editora editoras[2] = {  
    {1, 1, 1955}, {2, 2, 1920}  
};
```

```
Categoria categorias[2] = {  
    {1, 100}, {2, 200}  
};
```

```
Livro livros[4] = {  
    {1, 1, 1, 1, 1899, 300},  
    {2, 2, 2, 2, 1949, 250},
```

```
{3, 3, 1, 1, 1977, 150},  
{4, 1, 2, 1, 2005, 400}  
};
```

```
Exemplar exemplares[5] = {  
    {1,1,1}, {2,1,2}, {3,2,2}, {4,3,1}, {5,4,1}  
};
```

```
PessoaAluno alunos[3] = {  
    {1, 101, 2020}, {2, 102, 2021}, {3, 103, 2022}  
};
```

```
PessoaProfessor professores[2] = {  
    {1, 301, 2010}, {2, 302, 2015}  
};
```

```
Funcionario funcionarios[2] = {  
    {1, 401, 2000}, {2, 402, 2005}  
};
```

```
Emprestimo emprestimos[5] = {  
    {1,1,1,0,2024,2024},  
    {2,2,2,0,2024,0},  
    {3,3,3,0,2023,2023},  
    {4,4,1,1,2024,2024},  
    {5,5,4,0,2024,0}  
};
```

```
Reserva reservas[3] = {  
    {1,1,1,0,2024}, {2,2,3,0,2024}, {3,4,2,1,2024}  
};
```

```
SalaEstudo salas[2] = {  
    {1,4,1}, {2,6,2}  
};
```

```
ReservaSala reservasSala[2] = {  
    {1,1,1,0,2024,4}, {2,2,2,1,2024,6}  
};
```

```
Multa multas[2] = {  
    {1,1,5.0,0}, {2,3,10.0,1}  
};
```

```
};
```

```
Evento eventos[2] = {  
    {1, 1, 2024}, {2, 2, 2024}  
};
```

```
ParticipacaoEvento participacoes[3] = {  
    {1,1,1,1}, {2,1,2,1}, {3,2,1,0}  
};
```

```
Doacao doacoes[2] = {  
    {1,2,0,2023}, {2,3,0,2024}  
};
```

```
RevisaoExemplar revisoes[3] = {  
    {1,1,1,2023}, {2,1,2,2024}, {3,4,1,2024}  
};
```

```
SugestaoCompra sugestoes[2] = {  
    {1,1,0,10,2023}, {2,2,1,20,2024}  
};
```

```
RecomendacaoLeitura recomendacoes[2] = {  
    {1,1,1,2024}, {2,2,3,2024}  
};
```

5. Instruções para Uso

- Copie as declarações das structs e os vetores de dados para um arquivo .c com um main simples.
- NÃO use strings em suas soluções; utilize apenas comparações e operações numéricas.
- Todas as funções pedidas devem retornar valores (int, float ou estruturas agregadas) e NÃO devem imprimir diretamente.