nUSP	RunCodes	Observação	Código	Nota
13717786	0	Manpulação dos índices direto no arquivo. Índice secundário fortemente ligado. Busca por id sequencial. Remoção no índice secundário na operação de deleção por id somente reescreve o índice secundário no arquivo (o contrário também) (?).	5,5	2,75
12544570	10	Índice secundário fortemente ligado. Não faz deleção lógica no arquivo de dados (?).	8	9
13692334	10	Sugestão: Vetores de índices poderiam ser dinâmicos (realloc).	10	10
13692417	10	Índice secundário fortemente ligado. Não utilizou busca binária. Não faz deleção lógica no arquivo de dados. Sugestões: operações de busca são recorrentes (prórpria busca, verificar id duplicado, localizar para remoção), poderia utilizar uma função para fazer. Variáveis globais normalmente não são consideradas boas práticas. Duas formas diferentes de ler um livro do arquivo de dados, poderia refatorar para uma única função.	7	8,5
13692309	10	Sugestão: nomes poderiam ser mais explícitos (ex: Is => indiceSecundario).	10	10
13733579	10	Sugestões: Vetores de índices poderiam ser dinâmicos (realloc). Poderia ter separado as funcionalidades em arquivos (ex; livro. c indicePrimario.c, etc.).	10	10
11858651	6,67	Índice secundário fortemente ligado. Dados (livros) mantidos em memória principal. . Índice primário possui byteoffset e índice secundário rrn.	4	5,34
13672922	10	Dados (livros) mantidos em memória principal. Estruturas estão misturadas, único registro mantém os dados e o rrn (logo não há índice secundário fracamente ligado).	3	6,5
13838346	10	Sugestão: Variáveis globais normalmente não são consideradas boas práticas.	10	10
12682435	5	Índices são maniuplados em memória secundária. Não entendi o propósito da lista encadeada (em cada inserção ela começa vazia (head = NULL) (?)). Código muito complexo. Entradas no índice secundário não são mantidas ordenadas.	5,5	5,25
13861176	10	Índice secundário fortemente ligado. Índices manipulados em memória secundária. Busca por id sequencial. Buscas não utilizam byteoffset, é feita a busca no índice e no arquivo de dados.	2	6
13750791	10	Entradas no índice secundário não são mantidas ordenadas. Sugestão: poderia reutilizar funções de busca nas remoções.	9	9,5
13828592	10	Sugestão: muito código repetido poderia ser refatorado em funções.	10	10
13727485	10	Entradas no índice secundário não são mantidas ordenadas. Sugestão: a busca binária poderia ser mais abstrata, retornando a posição no array e não o prório IndexEntry, dessa forma ela poderia ser reutilizada na remoção.	9,5	9,75
10727952	10	Entradas no índice secundário não são mantidas ordenadas. Não faz deleção lógica no arquivo de dados.	8	9
11838777	10	Índices são maniuplados em memória secundária. Busca por autor recupera id e busca ele direto no arquivo de dados.	5,5	7,75
13673242	-		-	-
13732352	10	Sugestão: poderia reutilizar a busca binária para achar a posição do índice removido.	10	10
11871181	8,33	Sugestão: poderia utilizar mais funções, principalmente dentro dos ifs do loop de operações.	10	9,17
11833236	10	Não faz deleção lógica no arquivo de dados.	9	9,5
11781587	10	Índice secundário fortemente ligado. Não utiliza busca binária.	8	9
11295810	10	Não faz busca binária. Manipulação é feita diretamente no arquivo de dados. Índice só possui o id.	0	5
13696641	10	Índice secundário fortemente ligado. Sugestão: muita lógica na main, diversos blocos poderiam ser refatorados em funções.	9	9,5
13692362	5	Índice secundário fortemente ligado. Não faz busca binária. Índices manipulados diretamente nos arquivos.	5,5	5,25
9793502	10	Operações nos índices (lista, ordenação, busca binária, etc.) não feitas, (usa dicionário).	6	8