



## Trabalho 02

**Disciplina:** Organização de Arquivos - turma A

**Semestre:** 2014/2

**Entrega:** 03/12/2014 - 23h50.

### Correção:

- Entregar arquivos fontes via website do curso ([aprender.unb.br](http://aprender.unb.br)) em único arquivo zipado.
- Excepcionalmente para o Trabalho 2, não será exigido um relatório. Deverá ser entregue apenas código e comentários.
- Incluir um comentário com os nomes e matriculas dos alunos.
- Os trabalhos devem seguir boas práticas de programação (clareza, organização, comentários, etc.)
- Trabalho identificado como cópia (qualquer uma das partes) receberá nota zero.

**Grupos:** Neste projeto está permitido a formação de grupos de no máximo dois alunos. Somente um dos membros do grupo deve entregar o trabalho onde conste nome e matrícula de todos os participantes.

### Especificações:

- O arquivo *lista.txt* disponibilizado no moodle contém campos de tamanhos fixo com informação sobre MATRICULA, NOME, OPÇÃO, CURSO e TURMA nessa ordem. A partir desse arquivo, elabore um programa para criação de um arquivo de índices estruturado em árvore B, de ordem 5, para armazenagem em memória secundária, com as seguintes características:
- **Índice primário** - Utilize para criação da chave primária os campos de MATRICULA e NOME concatenados num total de 30 caracteres.
- **Terminal de saída** - Crie uma rotina para gravar em disco o arquivo de índices primários utilizando árvores B (*indicelista.bt*) e uma rotina para visualização do mesmo em monitor. Essa rotina deverá ser empregada após qualquer modificação sobre o arquivo. Modificações sobre o arquivo de dados também deverão ser gravadas em disco.
- **Busca de registros** - Crie uma rotina para buscar registros através de chave primária. A busca deve carregar em memória principal somente um nó da árvore B por vez. A rotina deve devolver o conteúdo do registro e o número de buscas (seeks) envolvidos.
- **Inclusão de registros** - Crie uma rotina que permita a inclusão de novo registro no arquivo de dados e no arquivo de índices em árvore B. Utilizar técnicas de divisão e promoção para preservar a estrutura de árvore B. Não é necessária a re-utilização de espaço disponível (uso



de LED) no arquivo de dados, a inclusão poderá ser feita ao final mesmo existindo espaço disponível.

- O programa deverá ser implementado em linguagem C ou C++. Em linguagem C, o programa deverá seguir o padrão ANSI. Em linguagem C++ o programa deverá ser compilável no sistema operacional UNIX.