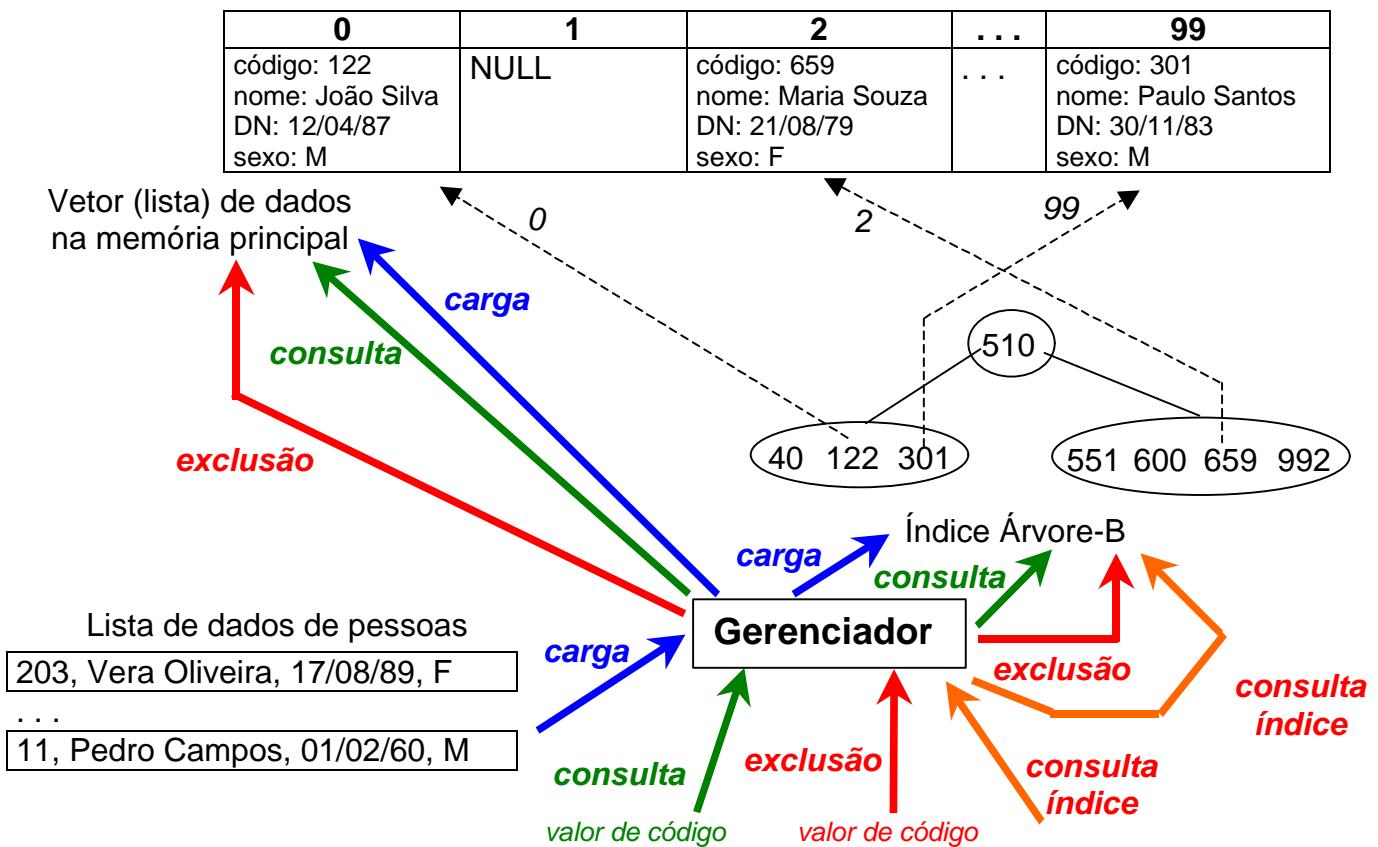


## Trabalho Prático II – Gerenciador de Índices em Árvore-B

O trabalho prático II é um trabalho de implementação que visa praticar os conhecimentos adquiridos sobre estruturas de dados em árvore e métodos de pesquisa de dados. Este trabalho pode ser implementado em **qualquer linguagem de programação orientada a objetos**, em **grupos de até três pessoas**. O trabalho deve ser apresentado ao professor por **todos** os integrantes do grupo. Uma **documentação de código** (classes/métodos implementados e suas finalidades) e o **próprio código fonte** devem ser entregues.

O trabalho a ser implementado é um **gerenciador de índices do tipo árvore-B**. A figura a seguir dá uma idéia das operações a serem controladas por este gerenciador.



As estruturas de dados manipuladas pelo gerenciador são:

- um **vetor em memória (lista de tamanho 100)** com dados de pessoas. Estes dados são: código (inteiro  $> 0$  e  $< 1000$ ), nome, data de nascimento e sexo (M ou F);
- um **índice do tipo árvore-B**, com  $n = 10$ . Cada nodo mantém um vetor com até 9 objetos. Cada objeto possui uma chave (código da pessoa) e um endereço (posição no vetor em onde estão os dados da referida pessoa);
- uma **lista de entrada (lista encadeada)** com dados de pessoas.

As operações a serem realizadas pelo gerenciador, conforme mostra a figura, são:

1. **Carga:** realiza a inclusão de um ou mais dados de pessoas no vetor e no índice. Recebe uma lista de entrada (esta lista pode ser imaginada como o resultado de uma consulta a um BD) com dados de pessoas. Para cada pessoa desta lista é feita a inserção do par de dados (chave, endereço) na árvore B e dos seus dados no vetor em memória, na primeira posição livre encontrada no vetor. Caso um código de pessoa da lista já exista no índice ou o código seja inválido ( $\leq 0$  ou  $\geq 1000$ ), esta pessoa não é inserida e uma mensagem é mostrada ao usuário para informá-lo. Caso o vetor se torne cheio durante a carga, o processo é interrompido e uma mensagem respectiva é também gerada. Não há problema se apenas alguns dados forem inseridos;
2. **Consulta:** realiza a consulta a dados de uma pessoa com base no seu código. Recebe um valor de código e consulta o índice para determinar o endereço (posição) onde se encontram os dados da respectiva pessoa. Em seguida, acessa o vetor no respectivo endereço e exibe os dados da pessoa. Caso o valor do código seja inválido ou não exista no índice, mensagens de erro devem ser geradas;
3. **Exclusão:** realiza a exclusão dos dados de uma pessoa do vetor e do índice. Recebe um valor de código e consulta o índice para determinar o endereço (posição) onde se encontram os dados da respectiva pessoa. Em seguida, acessa o vetor e seta NULL na respectiva posição. Caso o valor do código seja inválido ou não exista no índice, mensagens de erro devem ser geradas;
4. **Consulta ao índice:** exibe o conteúdo da árvore B. Sugestão:

raiz: a1 510 a2  
a1: ^ 40 ^ 122 ^ 301 ^  
a2: ^ 551 ^ 600 ^ 659 ^ 992 ^

### Observações:

- A definição das classes com seus respectivos métodos e exceções fica a cargo de cada grupo;
- Não é necessário implementar uma interface gráfica sofisticada para o trabalho;
- A **apresentação do trabalho** deve ser feita até o dia **26/02**, em um horário extra-classe a ser marcado com o professor;
- Apesar do trabalho ser em grupo, a avaliação é individual. Por isso, todos os integrantes do grupo devem comparecer à apresentação;
- Caso ocorrer **cópia ou reuso evidente de código** entre dois ou mais grupos, a nota destes grupos no trabalho será **zero**.