

Estruturas de dados

Pesquisa de dados em tabelas

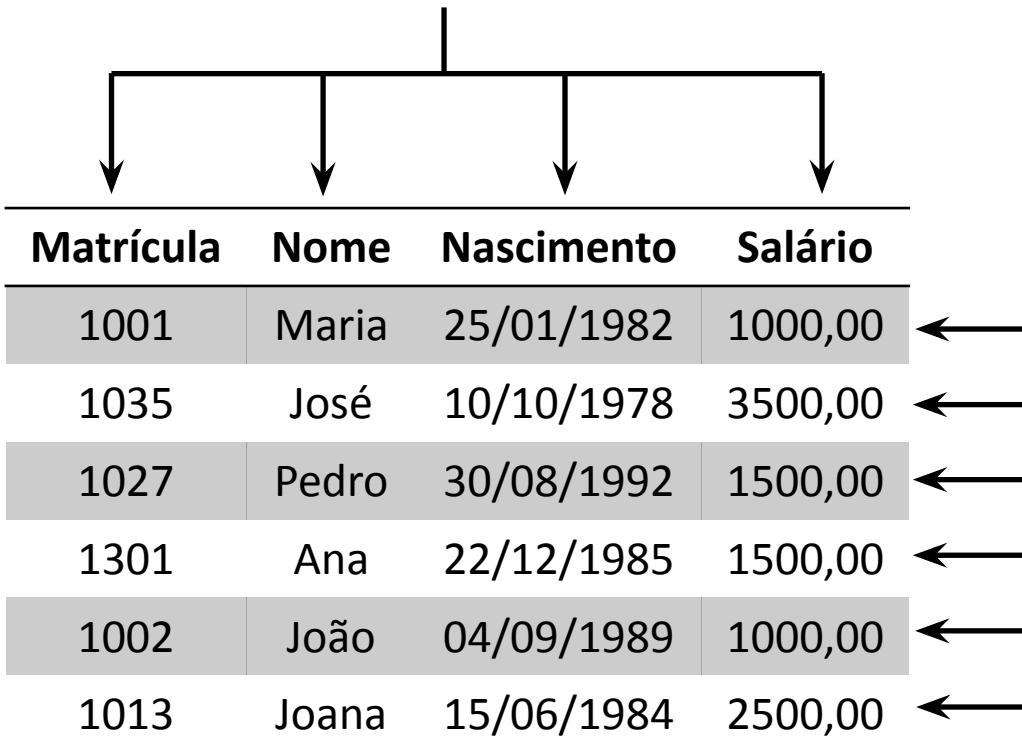
Contextualização

- Muitas vezes, o processamento de aplicações envolve a manipulação de conjunto de informações organizados em tabelas.
 - Estas tabelas possuem dados que podem ser alterados ou consultados.
- Uma tabela é formada por um conjunto de entradas (registros).
 - Cada entrada possui um conjunto de campos;
 - Normalmente armazena informações sobre objetos do mesmo tipo (entidade).
- Exemplo: dados cadastrais de funcionários de uma empresa.
 - Quais dados podem ser importantes? Matrícula (única), Nome, Data de nascimento e salário.

Tabelas

- Entidade Funcionário:
 - Formada por quatro campos;
 - Atualmente possui seis registros.

campos



Funcionário

Matrícula	Nome	Nascimento	Salário
1001	Maria	25/01/1982	1000,00
1035	José	10/10/1978	3500,00
1027	Pedro	30/08/1992	1500,00
1301	Ana	22/12/1985	1500,00
1002	João	04/09/1989	1000,00
1013	Joana	15/06/1984	2500,00

Registros

Chave primária

- Refere-se a um campo, cujos valores não podem se repetir;
- Toda tabela deve possuir um campo *chave primária*, que irá armazenar, **obrigatoriamente**, valores diferentes para cada entrada.
- Qual é a chave primária da tabela funcionário?
 - Matrícula.

Funcionário			
Matrícula	Nome	Nascimento	Salário
1001	Maria	25/01/1982	1000,00
1035	José	10/10/1978	3500,00
1027	Pedro	30/08/1992	1500,00
1301	Ana	22/12/1985	1500,00
1002	João	04/09/1989	1000,00
1013	Joana	15/06/1984	2500,00

Tabelas

- Como representar tabelas em C?

Funcionário			
Matrícula	Nome	Nascimento	Salário
1001	Maria	25/01/1982	1000,00
1035	José	10/10/1978	3500,00
1027	Pedro	30/08/1992	1500,00
1301	Ana	22/12/1985	1500,00
1002	João	04/09/1989	1000,00
1013	Joana	15/06/1984	2500,00

Tabelas

- Como representar tabelas em C?
 - Através das structs!

```

struct tabela {
    int matricula;
    char nome[20];
    char nasc[10];
    float salario;
};

typedef struct tabela Empresa;

```

```

int main () {
    Empresa funcionario[10]; // ou com alocação dinâmica.
    funcionario[0].matricula = 1001;
    strcpy (funcionario[0].nome, "Maria");
    strcpy (funcionario[0].nasc, "25/01/1982");
    funcionario[0].salario = 1000.00;
}

```

Funcionário			
Matrícula	Nome	Nascimento	Salário
1001	Maria	25/01/1982	1000,00
1035	José	10/10/1978	3500,00
1027	Pedro	30/08/1992	1500,00
1301	Ana	22/12/1985	1500,00
1002	João	04/09/1989	1000,00
1013	Joana	15/06/1984	2500,00

Pesquisa em tabelas

- Uma das operações mais frequentes no processamento de informações é a de busca (consulta) por dados armazenados em tabelas.
 - Visa acessar informações específicas sobre um ou mais registros da tabela;
 - A busca por registros normalmente é realizada a partir da chave primária, que deve ser fornecida pelo usuário e usada na comparação com cada uma das entradas da tabela.
 - Principais métodos de pesquisa:
 - Pesquisa sequencial;
 - Pesquisa binária.

Pesquisa sequencial

- Método mais simples de pesquisa e com performance mais modesta;
- Consiste na realização de uma varredura por todas as entradas da tabela, iniciando no primeiro registro, passando pelo segundo, terceiro, e repetir esse processo até encerrar;
- A chave primária de cada entrada é comparada com o **argumento de busca** (informado pelo usuário);
- A pesquisa termina em duas situações:
 - Se a chave primária de um registro é igual ao argumento de busca;
 - Se a pesquisa atingir o final da tabela (registro não encontrado).

Pesquisa sequencial

- Exemplo: argumento de busca = 1301;
- Comparar argumento de busca com a matrícula de cada funcionário, iniciando pelo primeiro registro.
 - Compara 1301 com 1001;
 - Compara 1301 com 1035;
 - Compara 1301 com 1027;
 - Compara 1301 com 1301;

Funcionário			
Matrícula	Nome	Nascimento	Salário
1001	Maria	25/01/1982	1000,00
1035	José	10/10/1978	3500,00
1027	Pedro	30/08/1992	1500,00
1301	Ana	22/12/1985	1500,00
1002	João	04/09/1989	1000,00
1013	Joana	15/06/1984	2500,00



Pesquisa binária

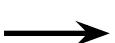
- Método que pode ser aplicado em tabelas cujos **registros estão ordenados pela chave primária**.
- Consiste em:
 1. Comparar o argumento de busca com a chave do registro localizado no **meio do conjunto de dados** (ordem crescente);
 2. Se o argumento for igual a chave, a pesquisa termina com sucesso;
 3. Se o **argumento for menor** que a chave, o processo é repetido com a metade superior da tabela;
 4. Se o **argumento for maior** que a chave, o processo é repetido com a metade inferior da tabela;
 5. A cada comparação, o número de elementos a serem pesquisados cai pela metade.

Pesquisa binária

- Exemplo: argumento de busca = 1350;
- Compara argumento de busca com a matrícula de cada funcionário, iniciando pelo registro central da tabela.
 - Compara 1350 com 1027;
 - Compara 1350 com 1301;
 - Compara 1350 com 1350.

Perceba que não houve necessidade de comparar o argumento com as matrículas 1001, 1002, 1013 e 1035.

Funcionário			
Matrícula	Nome	Nascimento	Salário
1001	Maria	25/01/1982	1000,00
1002	João	04/09/1989	1000,00
1013	Joana	15/06/1984	2500,00
1027	Pedro	30/08/1992	1500,00
1035	José	10/10/1978	3500,00
1301	Ana	22/12/1985	1500,00
1350	José	11/07/1990	1000,00



Exercício de aprendizagem 1



Sistemas para Internet
UFSM

- Utilizando o material disponibilizado no moodle, referente a **Pesquisa Sequencial**, implemente um algoritmo com uma função RECURSIVA que realize uma busca a partir da matrícula de um aluno e apresente suas informações;
- Caso o aluno não seja encontrado, apresente uma mensagem adequada;
- Existe diferença na busca caso os registros estejam ordenados pela chave primária? Qual?

Exercício de aprendizagem 2

- Utilizando o material disponibilizado no moodle, referente a **Pesquisa Binária**, implemente um algoritmo com uma função RECURSIVA que realize uma busca a partir da matrícula de um aluno e apresente suas informações ;
- Caso o aluno não seja encontrado, apresente uma mensagem adequada.