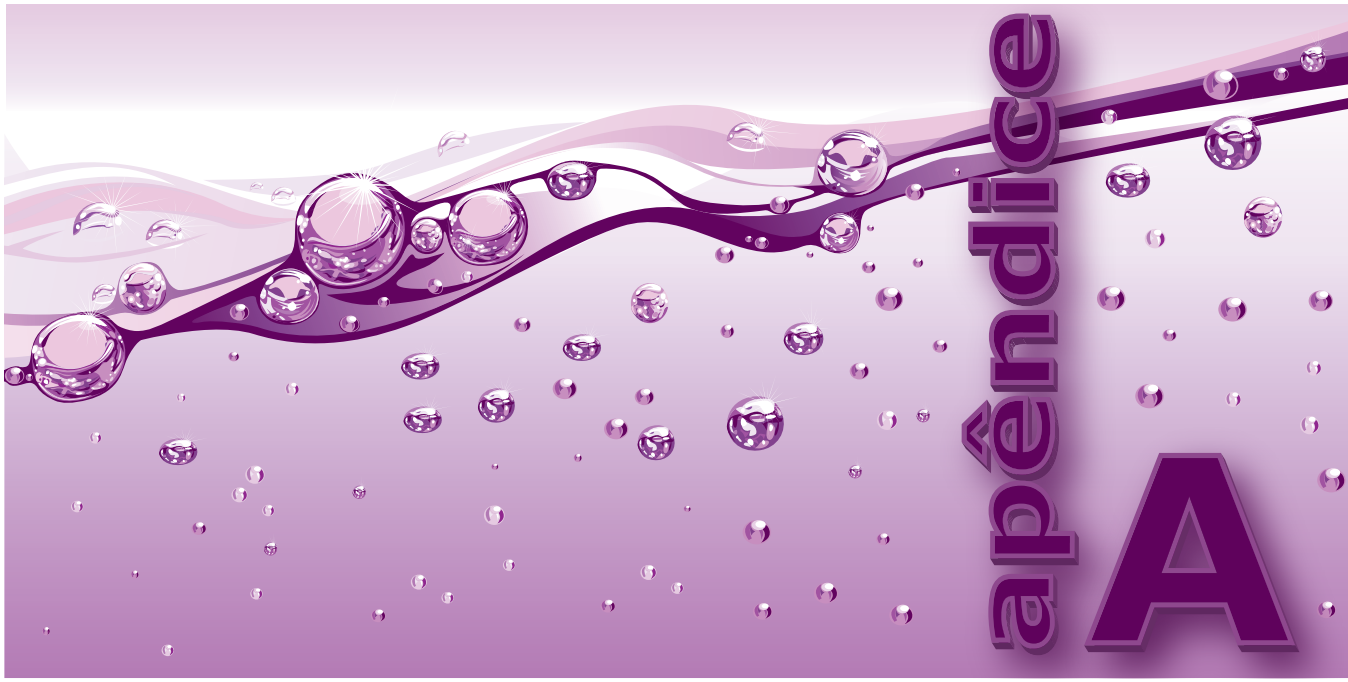

Arquivos de Dados

NESTE LIVRO são utilizados seis arquivos de dados para tornar mais concretos os exemplos dos diversos tipos de busca e ordenação. As interpretações dos conteúdos desses arquivos são menos importantes do que seus tamanhos, pois o que determina se um certo arquivo é adequado para um determinado tipo de busca ou ordenação é sua dimensão. Esses arquivos e os capítulos nos quais eles são usados são apresentados na **Tabela A–1**.

Nome do Arquivo	Tamanho	Tipo	Capítulos
`Tudor.txt`	*197 Bytes*	Texto	**2**
`Tudor.bin`	*336 bytes*	Binário	**2**, **11** e **12**
`CEPs.bin`	*151 MiB*	Binário	**3**, **4** e **7**
`CensoMEC.bin`	*4,29 GiB*	Binário	**3** e **6**
`Machado.txt`	*210 KiB*	Texto	**9**
`DNA.txt`	*53.6 MiB*	Texto	**9**

Tabela A–1: Arquivos Usados em Exemplos

Esses arquivos de dados, que serão descritos neste apêndice, podem ser encontrados no site dedicado ao livro na internet (v. **Prefácio**).

A.1 Tudor

`Tudor.txt` é um arquivo de texto no qual cada linha (**registro**) é dividida em quatro partes (**campos**) separadas por caracteres de tabulação. Os dados constituem uma turma escolar surreal formada pelo rei Henrique VIII e suas seis esposas. O conteúdo desse minúsculo arquivo é mostrado na **Tabela A–2**.

Henrique VIII	1029	9.5	9.0
Catarina Aragon	1014	5.5	6.5
Ana Bolena	1012	7.8	8.0
Joana Seymour	1017	7.7	8.7
Ana de Cleves	1022	4.5	6.0
Catarina Howard	1340	6.0	7.7
Catarina Parr	1440	4.0	6.0

Tabela A–2: Conteúdo do Arquivo Tudor.txt

O arquivo `Tudor.bin` é a versão binária do arquivo `Tudor.txt` e possui os campos descritos na **Tabela A–3**.

Nome do Campo	Interpretação	Tipo de Dado
nome	Nome do aluno	char[21]
matr	Matrícula do aluno	char[5]
n1	Primeira nota	double
n2	Segunda nota	double

Tabela A–3: Campos do Arquivo Tudor.bin

Os tipos de dados que representam os registros do arquivo `Tudor.bin` são definidos em C como:

```
   /* Tipo do nome */
typedef char tNome[MAX_NOME + 1];

   /* Tipo da matrícula */
typedef char tMatricula[TAM_MATR + 1];

typedef struct {
           tNome      nome;   /* Nome do aluno */
           tMatricula matr;   /* Sua matrícula */
           double     n1, n2; /* Suas notas    */
       } tAluno;
```

Nessas definições de tipos, `MAX_NOME` e `TAM_MATR` são constantes simbólicas previamente definidas com os valores `20` e `4`, respectivamente.

O arquivo `Tudor.bin` é usado nos **Exemplos de Programação** (**Seção 2.15**) do **Capítulo 2**, no exemplo de ordenação de ponteiros (v. **Seção 11.8.2**) e no exemplo de ordenação de arquivos por indexação (v. **Seção 12.7.1**).

A.2 CEPs

O arquivo `CEPs.bin` possui *673580* registros e seu tamanho é cerca de *151* MiB. Esse arquivo é usado em todos os exemplos de tabelas de busca dos **Capítulos 3**, **4** e **7**. Como esse arquivo é relativamente grande para ser completamente carregado em memória, as tabelas de busca que o utilizam são mantidas em memória principal e armazenam apenas as chaves dos registros com as respectivas posições desses registros no arquivo (i.e., elas são tabelas de busca interna com chaves externas).

O arquivo `CEPs.bin` é composto por registros que contêm os campos apresentados na **Tabela A–4**.

Campo	Interpretação	Tipo
numero	Número do registro na lista	int
UF	Estado do CEP	char[2]
localidadeNumero	Número identificador da localidade	int
nomeAbr	Nome abreviado da localidade	char[40]
nome	Nome completo da localidade	char[40]
bairroInicio	Bairro onde se inicia a contagem dos CEPs da localidade	int
bairroFim	Bairro que delimita o fim da contagem dos CEPs	int
CEP	Número do CEP	char[8]
complemento	Algum complemento da localidade	char[60]
tipoLogradouro	Rua, praça, avenida etc.	char[10]
statusLogradouro	Status da atividade postal do logradouro	char
nomeSemAcento	Nome sem acentuação do logradouro	char[40]
chaveDNE	Chave do registro da localidade no banco de dados dos correios (DNE)	char[16]

Tabela A–4: Campos do Arquivo CEPs.bin

O tipo de dado que representa os registros do arquivo `CEPs.bin` é definido em C como:

```
typedef struct {
           long numero;
           char UF[TAM_UF];
           int  localidadeNumero;
           char nomeAbr[MAX_NOME];
           char nome[MAX_NOME];
           int  bairroInicio;
           int  bairroFim;
           char CEP[TAM_CEP];
           char complemento[MAX_COMP];
           char tipoLogradouro[MAX_TIPO_LOG];
           char statusLogradouro;
           char nomeSemAcento[MAX_NOME];
           char chaveDNE[TAM_DNE];
         } tRegistroCEP;
```

As constantes simbólicas (`TAM_UF`, `MAX_NOME` etc) são previamente definidas e seus valores não contribuem para o entendimento dos exemplos que usam o arquivo `CEPs.bin`, de modo que você não precisa preocupar-se com eles. As definições dessas constantes são as seguintes:

```
#define MAX_NOME  40 /* Número máximo de caracteres num nome */
#define TAM_UF     2 /* Número de dígitos num UF */
#define TAM_CEP    8 /* Número de dígitos num CEP */
#define MAX_COMP  60 /* Número máximo de caracteres num complemento */
#define MAX_TIPO_LOG 10 /* Número máximo de caracteres num tipo de logradouro */
#define TAM_DNE   16 /* Número de dígitos numa chave DNE */
```

Nos exemplos de tabelas de busca que usam chave interna o conteúdo efetivo de cada elemento (ou nó) da tabela é definido usando os seguintes tipos:

```
typedef char tCEP[TAM_CEP + 1]; /* Tipo de chave */

   /* Tipo de conteúdo de um elemento */
typedef struct {
           tCEP chave; /* CEP */
           int  valor; /* Índice do CEP no arquivo de registos de CEPs */
         } tCEP_Ind;
```

Observação: É importante notar que o conteúdo do arquivo `CEPs.bin` é muito antigo e foi obtido via internet numa época em que ele era acessível livremente sem nenhum custo. Atualmente, um banco de dados completo de CEPs contendo cerca de *900.000* registros é vendido pelos Correios pela bagatela de *R$2.500*. Portanto esse arquivo está totalmente desatualizado e não serve para nenhum propósito prático. Como foi dito acima, sua importância nesse livro deve-se ao seu tamanho, e não à interpretação de seu conteúdo.

A.3 Censo Escolar

O tamanho do arquivo `CensoMEC.bin` é cerca de *4,29 GiB* e seu número de registros é *9.565.483*. Esse arquivo é derivado do **Censo da Educação Superior 2012** realizado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) vinculado ao Ministério da Educação (MEC). Em seu formato original CSV[1], cada registro desse banco de dados possui *98* campos e o tamanho do arquivo é uma aberração: cerca de *16 GiB* (a maior parte dele inútil!). Para facilitar o manuseio desse arquivo, o número de campos foi reduzido para apenas *8*, que são descritos na **Tabela A–5**.

Campo	Interpretação	Tipo
`codigoAluno`	Código de identificação atribuído ao aluno	`long`
`codAlunoCurso`	Combinação do código do aluno com o curso que ele frequenta (chave primária)	`int`
`nomeIES`	Nome da instituição à qual o aluno pertence	`char[200]`
`nomeCurso`	Nome do curso que o aluno frequenta	`char[200]`
`sexoAluno`	Sexo do aluno	`char`
`idadeAluno`	Idade do aluno	`int`
`UFNascimento`	UF na qual o aluno nasceu[†]	`char[30]`
`anoIngresso`	Ano de ingresso do aluno no curso	`int`

Tabela A–5: Campos do Arquivo CensoMEC.bin

[†] Se o leitor ficar surpreso com o fato de UF ser representada com 30 caracteres, o autor lhe é solidário...

O tipo de dado que representa os registros do arquivo `CensoMEC.bin` é definido em C como:

```
typedef struct {
           long codigoAluno;
           int  codAlunoCurso; /* chave */
           char nomeIES[MAX_NOME_IES];
```

[1] **CSV** é um tipo de arquivo de texto bastante utilizado em programas de bancos de dados e planilhas eletrônicas no qual cada dois campos consecutivos são separados por ponto e vírgula.

```
          char nomeCurso[MAX_NOME_CURSO];
          char sexoAluno;
          int  idadeAluno;
          char UFNascimento[MAX_UF_NASCIMENTO];
          int  anoIngresso;
        } tRegistroMEC;
```

As definições das constantes simbólicas que aparecem nessa última definição de tipo são as seguintes:

```
#define TAM_CODIGO_ALUNO  12 /* Número de dígitos num código de aluno */
#define TAM_ALUNO_CURSO    8  /* Número de dígitos num código de aluno/curso */
#define MAX_NOME_IES     200 /* Tamanho máximo de um nome de IES   */
#define MAX_NOME_CURSO   200 /* Tamanho máximo de um nome de curso */
#define MAX_UF_NASCIMENTO 30 /* Tamanho máximo de um nome de UF    */
```

Nos exemplos de tabelas de busca que usam chaves externas o conteúdo efetivo de cada elemento (ou nó) da tabela é definido usando os seguintes tipos:

```
typedef int tChave; /* Tipo da chave */

   /* Tipo de conteúdo de um elemento */
typedef struct  {
          tChave chave;  /* Chave do registro */
          int    indice; /* Posição do registro no arquivo */
} tChaveIndice;
```

O arquivo `CensoMEC.bin` é usado apenas em buscas (v. capítulos **6** e **8**) e ordenações (v. **Capítulo 11**) em memória secundária.

A.4 Machado

O arquivo `Machado.txt` é uma transcrição do romance *Ressurreição* de Machado de Assis, publicado em 1872 (v. **Bibliografia**), convertido pelo autor desta obra em texto puro. Esse arquivo é usado na **Seção 9.10.2** e na **Seção 9.10.6**.

A.5 DNA

O arquivo `DNA.txt` é um arquivo de texto em formato **FASTA** contendo parte do genoma humano **HG-19**[2]. Esse arquivo pode ser facilmente obtido em formato comprimido via internet e é usado no exemplo apresentado na **Seção 9.10.7**. Ele também está disponível no site dedicado a este livro.

[2] Arquivo do tipo FASTA representa um padrão internacional para arquivos de texto representando nucleotídeos e peptídeos. HG-19 é o mapa cromossômico mitocondrial de número 19 do genoma humano realizado em fevereiro de 2009. Se o leitor não entende nada disso, está quite com o autor... O importante, neste caso, é o formato e o tamanho do arquivo e não sua interpretação.