

**Universidade de São Paulo**  
**Instituto de Ciências Matemáticas e de Computação**  
**Departamento de Ciências de Computação**  
**Disciplina de Organização de Arquivos**  
**Profa. Dra. Cristina Dutra de Aguiar Ciferri**  
**PAE Fábio Felix Dias (Turma A)**  
**PAE Anderson Chaves Carniel (Turma B)**

**Primeira Parte do Trabalho Prático (Parte I)**  
**Valor: 30%**

*O trabalho deve ser feito em grupo de 4 alunos. A solução deve ser proposta exclusivamente pelo grupo com base nos conhecimentos adquiridos ao longo das aulas. Consulte as notas de aula e o livro texto quando necessário.*

---

**Descrição do arquivo de dados**

---

Cada registro do arquivo de dados deve conter dados relacionados a:

**Turma A.** Dados dos domínios governamentais registrados no Registro.br (coletado em <http://dados.gov.br/dataset/dominios-gov-br>), de forma que cada registro indique um domínio governamental.

- Campos de tamanho fixo: *ticket* (número do ticket vinculado ao registro da URL no cadastro do Registro.br), *documento* (número do documento informado ao Registro.br no cadastro da URL), *dataHoraCadastro* (data e hora de quando a URL foi cadastrada), *dataHoraAtualiza* (data e hora de quando a URL foi modificada).
- Campos de tamanho variável: *dominio* (URL cadastrada no Registro.br), *nome* (nome do Órgão ou Entidade responsável pela URL), *cidade* (cidade cadastrada no Registro.br), *uf* (nome do estado por extenso).

**Turma B.** Dados cadastrais de companhias abertas no Brasil (coletado em <http://dados.gov.br/dataset/cias-abertas>), de forma que cada registro indique uma companhia aberta.

- Campos de tamanho fixo: *CNPJ* (CNPJ de cadastro da companhia), *dataRegistro* (data em que a companhia foi registrada sob o CNPJ), *dataCancelamento* (data em que a companhia foi fechada), *CNPJauditor* (CNPJ da empresa que presta auditoria para a companhia).
- Campos de tamanho variável: *nomeSocial* (nome no qual foi registrado o CNPJ), *nomeFantasia* (nome fantasia da companhia), *motivoCancelamento* (razão pela qual a companhia foi fechada), *nomeEmpresa* (nome da empresa que presta auditoria).

---

### Organização do arquivo de dados

---

Deve ser considerada a *organização híbrida de campos e registros*, da seguinte forma:

#### **Turma A.**

- Campos de tamanho fixo e campos de tamanho variável. Para os campos de tamanho variável, deve-se usar o método *indicador de tamanho*.
- Registros de tamanho variável. Para os registros de tamanho variável, deve-se usar os seguintes três métodos: (i) *indicador de tamanho*; (ii) *delimitadores entre registros*; e (iii) *número fixo de campos*.

#### **Turma B.**

- Campos de tamanho fixo e campos de tamanho variável. Para os campos de tamanho variável, deve-se usar o método *delimitadores entre campos*.
- Registros de tamanho variável. Para os registros de tamanho variável, deve-se usar os seguintes três métodos: (i) *indicador de tamanho*; (ii) *delimitadores entre registros*; e (iii) *número fixo de campos*.

### Importante.

Não é permitido misturar os métodos para organização em registros. Por exemplo, não é permitido misturar o método *número fixo de campos* com o método *delimitadores entre registros*. Ou seja, cada método deve ser implementado de forma separada, conforme solicitado. A mistura dos métodos será avaliada como errada.

---

## Programa

---

Implemente um programa em C que ofereça as seguintes funcionalidades.

[1] Ofereça uma interface por meio da qual o usuário possa escolher a funcionalidade a ser realizada.

[2] Permita a leitura de vários registros obtidos a partir de um arquivo de entrada e a gravação desses registros em um arquivo de dados de saída. O arquivo de entrada será fornecido juntamente com a especificação do projeto, enquanto que o arquivo de dados de saída deverá ser gerado como parte deste trabalho prático.

[3] Permita a recuperação dos dados, de todos os registros, armazenados no arquivo de dados, mostrando os dados de forma organizada na saída padrão para permitir a distinção dos campos e registros. Deve-se permitir a visualização dos registros, um por vez.

[4] Permita a recuperação dos dados de todos os registros que satisfaçam um critério de busca determinado pelo usuário. Por exemplo, para a Turma A, o usuário pode solicitar a exibição de todos os registros de um determinado *domínio*, enquanto que para a Turma B, o usuário pode solicitar a exibição de todos os registros que possuem um determinado *nomeFantasia*. Os dados solicitados devem ser mostrados de forma organizada na saída padrão para permitir a distinção dos campos e registros. Deve-se permitir a visualização dos registros, um por vez.

[5] Permita a recuperação dos dados de um registro, a partir da identificação do registro desejado pelo usuário. Por exemplo, o usuário pode solicitar a recuperação dos dados do terceiro registro ou do quinto registro. Os dados solicitados devem ser mostrados de forma organizada na saída padrão para permitir a distinção dos campos.

[6] Permita a recuperação de um campo específico de um registro de dados, a partir da identificação do campo e do registro desejado pelo usuário. Por exemplo, o usuário pode solicitar a recuperação do terceiro campo do quinto registro. O dados solicitado deve ser mostrado de forma organizada na saída padrão.

---

### Restrições

---

As seguintes restrições têm que ser garantidas no desenvolvimento do trabalho.

[1] O arquivo de dados deve ser gravado em disco no **modo binário**. O modo texto não deve ser usado.

[2] Os dados do registro descrevem os nomes dos campos, os quais não podem ser alterados. Ademais, todos os campos devem estar presentes na implementação, e nenhum campo adicional pode ser incluído. O tamanho de cada campo deve ser determinado pelo grupo. Essa escolha deve ser explicada na **documentação externa**.

[3] Devem ser exibidos avisos ou mensagens de erro sempre que apropriado.

[4] Os dados devem ser escritos e lidos campo a campo. Pode-se usar também a serialização (memcpy).

[5] Os integrantes do grupo devem constar como comentário no início do código (i.e. NUSP e nome de cada integrante do grupo). Não será atribuída nota ao aluno cujos dados não constarem no código fonte.

[6] Todo código fonte deve ser documentado. A **documentação interna** inclui, dentre outros, a documentação de procedimentos, de funções, de variáveis, de partes do código fonte que realizam tarefas específicas. Ou seja, o código fonte deve ser documentado tanto em nível de rotinas quanto em nível de variáveis e blocos funcionais.

[7] A interface pode ser feita em modo texto (terminal) ou modo gráfico e deve ser funcional.

[8] A implementação deve ser realizada usando a linguagem de programação C. As funções das bibliotecas <stdio.h> devem ser utilizadas para operações relacionadas à escrita e leitura dos arquivos. A implementação não deve ser feita em qualquer outra linguagem de programação. O programa deverá compilar no GCC versão 4.8.2 ou superior para Linux.

[9] O programa deve ser acompanhado de uma **documentação externa**. A documentação externa deve conter a descrição dos principais conceitos usados no trabalho prático, as decisões de projeto e as suas justificativas (ex.: estruturas de dados e algoritmos usados), assim como qualquer outra consideração adicional assumida no desenvolvimento do trabalho prático. Em detalhes, a documentação externa deve possuir:

- CAPA, com as seguintes informações: o nome da instituição, o nome do curso, o nome da disciplina, o nome do professor responsável, o nome do trabalho prático, o nome dos participantes e os respectivos números USP, e a data de entrega do trabalho prático.
- ÍNDICE, listando os nomes das seções que compõem o trabalho prático e as suas respectivas páginas de início.
- SEÇÕES 1 a N: Quaisquer decisões de projeto. Em detalhes, a documentação referente a essas seções deve conter a descrição dos principais conceitos usados no trabalho prático, incluindo desenhos que facilitem a compreensão das estruturas de dados, as decisões de projeto e as suas justificativas, assim como qualquer outra consideração adicional assumida no desenvolvimento do trabalho prático. Todas as funcionalidades do programa devem ser descritas em

detalhes. Por exemplo, na Parte I do trabalho, deve ser feita a descrição dos campos dos registros do arquivo de dados, contendo os nomes dos campos, os tamanhos dos campos e um desenho que mostra visualmente a estrutura dos registros. A escolha dos tamanhos dos campos deve ser justificada. A escolha dos tamanhos dos campos, bem como dos campos propriamente ditos, deve ser justificada. Coloque aqui também os sistemas operacionais que foram usados e como o programa deve ser compilado e executado.

- SEÇÃO N+1: Cópias de telas da interface, por meio das quais é possível entender o funcionamento do programa.
- SEÇÃO N+2: Baterias de testes, as quais devem gerar informações que permitam acompanhar a execução do programa.
- REFERÊNCIAS BIBLIOGRÁFICAS, caso necessário.

---

### Fundamentação Teórica

---

Conceitos e características dos diversos métodos para representar os conceitos de campo e de registro em um arquivo de dados podem ser encontrados nas transparências de sala de aula e também nas páginas 96 a 107 do livro *File Structures (second edition)*, de Michael J. Folk e Bill Zoellick.

---

### Material para Entregar

---

**Arquivo compactado.** Deve ser preparado um arquivo .zip contendo:

- Código fonte do programa devidamente documentado.
- Makefile para a compilação do programa.
- Bibliotecas necessárias para a execução do programa.
- Documentação externa em formato .pdf.

### Instruções de entrega.

[1] Enviar o arquivo compactado para o e-mail [labbdciferri@gmail.com](mailto:labbdciferri@gmail.com), com o seguinte assunto: [Organização de Arquivos] Trabalho Prático 2017 – Turma X – Parte Y. Deve constar no corpo da mensagem o NUSP e nome de cada integrante do grupo. Não será atribuída nota ao aluno cujos dados não constarem no corpo da mensagem.

[2] Entregar pessoalmente a documentação externa impressa em horário e local definido na página da disciplina.

---

### Critério de Correção

---

**Critério de avaliação do trabalho.** Na correção do trabalho, serão ponderados os seguintes aspectos.

- Qualidade da documentação (interna e externa) entregue. MAIOR PESO
- Corretude da execução do programa. MAIOR PESO
- Qualidade da interface.

### Restrições adicionais sobre o critério de correção.

- A não execução de um programa devido a erros de compilação implica que a nota final da parte do trabalho será igual a zero (0).
- A ausência da documentação interna implica que haverá uma diminuição expressiva na nota do trabalho.
- A ausência da documentação externa implica que haverá uma diminuição expressiva na nota do trabalho.
- A inserção de palavras ofensivas nos arquivos e em qualquer outro material entregue implica que a nota final da parte do trabalho será igual a zero (0).
- Em caso de cola, as notas dos trabalhos envolvidos será igual a zero (0).

**Critério de avaliação dos integrantes.** Podem ser incluídas uma ou mais perguntas a respeito do trabalho na prova.

---

**Data de Entrega do Trabalho**

---

Na data especificada na página da disciplina.

**Bom Trabalho !**