Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
Disciplina de Algoritmos e Estruturas de Dados II (SCC0603)
Profa. Dra. Cristina Dutra de Aguiar Ciferri
PAE Rayner de Melo

Trabalho Prático

Este trabalho tem como objetivos: (i) realizar o armazenamento e a recuperação de registros em arquivos; e (ii) utilizar conceitos relacionados à teoria de grafos para gerar grafos de derivação que simulem os níveis de agregação de um *data warehouse*.

> *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. Material adicional sobre ambientes de data warehousing podem ser encontrados na página da disciplina.*

FUNDAMENTAÇÃO TEÓRICA
- ORGANIZAÇÃO DE ARQUIVOS -

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.

FUNDAMENTAÇÃO TEÓRICA
- DATA WAREHOUSING -

1. Ambientes de Data Warehousing

Um ambiente de *data warehousing* provê a base para o ambiente informacional de uma empresa, proporcionando eficiência e flexibilidade na obtenção de informações estratégicas sobre o negócio, por meio da recuperação de informação sumarizada de qualidade, própria para a tomada de decisão [Chaudhuri and Dayal, 1997]. O acesso à informação dos provedores autônomos, heterogêneos e distribuídos que compõem o ambiente é realizado, geralmente, em duas etapas. Na primeira delas, conhecida como ETL (*extract, transform, load*), os dados de cada provedor são extraídos, filtrados, integrados aos dados relevantes de outros provedores e armazenados no principal componente do ambiente, o *data warehouse*. Na segunda etapa, conhecida como etapa de análise e consulta, consultas analíticas, denominadas consultas OLAP (*on-line analytical processing*), são executadas diretamente no *data warehouse*, sem acessar os provedores de informação originais.

Portanto, a informação integrada torna-se disponível para consulta ou análise imediata de usuários de sistemas de suporte à decisão (SSD) [Chaudhuri et al., 2011, Ciferri et al., 2013]. Por exemplo, uma aplicação de *data warehousing* de uma cadeia de supermercados poderia integrar dados relativos à venda de produtos em filiais ao longo do tempo. Usuários desta aplicação poderiam realizar, utilizando estas informações integradas, análises de tendências simples (quais as vendas mensais de um determinado produto no ano de 1998?), análises comparativas (quais as vendas mensais dos produtos de uma determinada marca nos últimos três anos?) e análises de tendência múltiplas (quais as vendas mensais dos produtos de uma determinada marca nos últimos três anos, de acordo com as promoções realizadas no período do Dia dos Namorados e do Dia das Mães?).

2. Data Warehouse

O *data warehouse* consiste em um dos principais componentes de um ambiente de *data warehousing*. Ele é um banco de dados especialmente organizado para armazenar dados integrados, orientados a assunto, não voláteis e históricos. Os dados integrados são gerados pela etapa de ETL. Eles são orientados a assunto, ou seja, relativos aos temas de negócio de maior interesse da corporação, tais como clientes, produtos, promoções, contas e vendas. A característica de não-volatilidade está relacionada ao fato de que o conteúdo do *data warehouse* permanece estável por longos períodos de tempo. Por fim, os dados do *data warehouse* são históricos, ou seja, relevantes a algum período de tempo (usualmente 5 a 10 anos).

Outra característica do *data warehouse* é que seus dados são usualmente modelados multidimensionalmente [Wu and Buchmann, 1997], em função das análises efetuadas pelos usuários de SSD, as quais têm por objetivo a visualização dos dados segundo diferentes perspectivas (ou seja, dimensões). Uma visão multidimensional (ou agregação) inclui um conjunto de medidas numéricas de interesse, que são os objetos de análise relevantes ao negócio, e um conjunto de dimensões, as quais contêm atributos que determinam o contexto para as medidas numéricas. Uma dimensão é descrita por atributos, sendo que cada atributo pode ter um relacionamento com outros atributos da dimensão por meio de hierarquias, as quais especificam diferentes níveis de granularidade e agregação dos dados. Por exemplo, na aplicação de *data warehousing* da cadeia de supermercados que integra dados relativos a *vendas*, *produtos* e *filiais* ao longo do *tempo*, a medida numérica vendas é determinada pelas dimensões *produto*, *filial* e *tempo*. Ademais, a dimensão *tempo* pode ser pode ser expressa pela hierarquia de atributos (*ano*) $\preceq$ (*semestre*) $\preceq$ (*trimestre*) $\preceq$ (*mês*) $\preceq$ (*dia*), sendo que *dia* é o atributo de menor granularidade, enquanto que *ano* é o atributo de maior granularidade. O operador $\preceq$ impõe uma ordem parcial, especificando que uma agregação de maior granularidade pode ser determinada usando outra agregação de menor granularidade. Por exemplo, (*trimestre*) pode ser determinado a partir de (*mês*). A existência de mais do que uma hierarquia de atributos para uma mesma dimensão é totalmente plausível. Outra hierarquia para a dimensão tempo é (*ano*) $\preceq$ (*quadrimestre*) $\preceq$ (*mês*) $\preceq$ (*dia*).

Além de serem modelados multidimensionalmente, os dados do *data warehouse* também são organizados em diferentes níveis de agregação, desde um nível inferior que possui dados detalhados obtidos do ambiente operacional da corporação, até um nível superior que possui dados muito resumidos. Também podem existir vários níveis intermediários entre estes dois níveis, representando graus de agregação crescentes [Harinarayan et al., 1996]. A organização dos dados em níveis de agregação permite que as medidas numéricas sejam analisadas segundo diferentes perspectivas. No exemplo corrente, pode-se analisar: (i) *vendas* por *produto* por *filial* por *dia* (nível inferior); (ii) *vendas* por *produto* por *mês* e *vendas* por *filial* por *ano* (diferentes níveis intermediários); e (iii) total de *vendas* (nível superior).

3. Grafo de Derivação

Um grafo orientado é um par (V,E) de conjuntos disjuntos de vértices V e arestas E, juntamente com dois mapeamentos: inic: E $\rightarrow$ V e term: E $\rightarrow$ V. Estes mapeamentos associam a cada aresta *e* um vértice inicial inic(*e*) e um vértice terminal term(*e*), de forma que e é direcionada de inic(*e*) para term(*e*). Um grafo orientado não possui ciclos, ou seja, inic(*e*) $\neq$ term(*e*). Adicionalmente, um grafo orientado não possui arestas múltiplas. Isto significa que existe somente uma única aresta entre os mesmos dois vértices do grafo [Diestel, 1997].

Um *lattice* de visões (ou *lattice* de visões agregadas ou, resumidamente, *lattice* de agregações) é definido pelas seguintes propriedades [Harinarayan et al., 1996]:

- Existe uma ordenação parcial $\preceq$ entre as agregações no *lattice*. Para as visões agregadas *u* e *v*, $v \preceq u$ se e somente se *v* pode ser determinada usando somente os resultados de *u*. Em outras palavras, a agregação *v* é dependente da agregação *u*;
- Existe uma visão topo, da qual cada outra visão agregada é dependente. Mais especificamente, esta visão é utilizada para derivar qualquer outra visão no *lattice*; e
- Pode existir uma visão completamente agregada que pode ser calculada a partir de qualquer outra visão no *lattice*. A visão *all* representa a visão completamente agregada.

Adicionalmente, algumas funções podem ser definidas para os elementos do *lattice*. Dado três agregações *v*, *w* e *u*, ancestrais, descendentes, ancestrais diretos e descendentes diretos de *v* são definidos como:

$$\text{ancestrais}(v) = \{ w \mid v \preceq w \} . \quad (1)$$

$$\text{descendentes}(v) = \{ w \mid w \preceq v \} . \quad (2)$$

$$\text{ancestrais\_diretos}(v) = \text{pais}(v) = \{ w \mid v \prec w, \nexists u, v \prec u, u \prec w \} . \quad (3)$$

$$\text{descendentes\_diretos}(v) = \text{filhos}(v) = \{ w \mid w \prec v, \nexists u, w \prec u, u \prec v \} . \quad (4)$$

Nas Equações 3 e 4, a relação existente entre $\prec$ e $\preceq$ é representada por:

$$v \prec w \Rightarrow v \preceq w \wedge v \neq w . \quad (5)$$

Uma vez definidos os conceitos de grafo orientado e de *lattice* de visões, o conceito de grafo de derivação pode ser definido. Um grafo de derivação G consiste em um grafo orientado no qual V(G) representa um conjunto de visões agregadas (medidas numéricas agregadas), ao passo que E(G) representa um conjunto de relações de dependência $\preceq$ entre essas visões agregadas. Assim, se a aresta $e \in$ G representa uma relação de dependência entre dois vértices v e $u \in$ G de forma que $v \preceq u$, então inic(e) = u e term(e) = v.

5. Exemplos de Grafo de Derivação

Visando exemplificar os conceitos discutidos anteriormente, a Figura 1 ilustra dois grafos de derivação. O primeiro deles, exibido na Figura 1a, considera apenas as dimensões *produto* (p), *filial* (f) e *tempo* (t). Já a Figura 1b exibe o grafo de derivação em termos das dimensões e das suas respectivas hierarquias de atributos. Nessa figura, são consideradas: (i) as dimensões *produto* (p) e *filial* (f); e (ii) as hierarquias de atributos (*all*) $\preceq$ *marca* (*m*) $\preceq$ *produto* (*p*) e (*all*) $\preceq$ *estado* (*e*) $\preceq$ *cidade* (*c*) $\preceq$ *filial* (*f*).

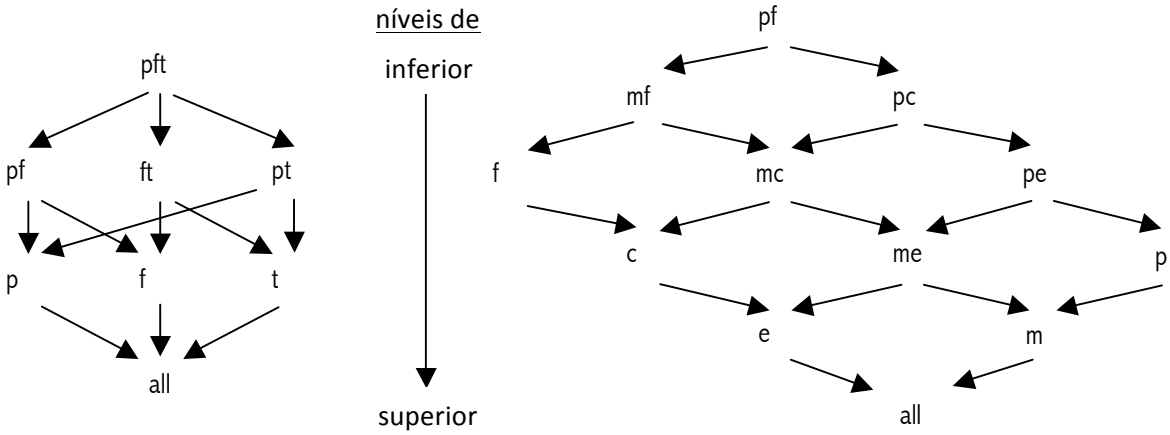

Figura 1a Dimensões produto (p), filial (f) e tempo (t).

Figura 1b Dimensões produto (p) e filial (f), e marca (m) $\prec$ produto (p) e estado (e) $\prec$ cidade (c) $\prec$ filial (f)

Figura 1 Exemplos de grafos de derivação.

Cada agregação (ou seja, cada vértice do grafo de derivação) agrega medidas numéricas sobre as dimensões presentes naquele vértice, e é nomeada de acordo com essas dimensões. Quando são considerados os atributos das dimensões, então os vértices também são nomeados de acordo com essas dimensões, considerando no entanto a granularidade de seus atributos. Considerando a aplicação corrente de *data warehousing* de uma cadeia de supermercados, a qual analisa a medida numérica *vendas* de acordo com as dimensões presentes, o vértice (*pf*) em ambas as figuras representa a visão multidimensional *vendas* por *produto* por *filial*. O vértice (*me*) na Figura 1b representa a visão multidimensional *vendas* por *marca* por *estado*.

Já as dependências existentes entre agregações adjacentes são representadas por meio de arestas direcionadas. O grafo de derivação apenas ilustra as dependências entre as agregações adjacentes, e não as suas cláusulas transitivas. Na Figura 1a, o vértice (*p*) pode ser computado a partir do vértice (*pf*), ou seja, *inic(pf_p)* = *pf* e *term(pf_p)* = *p*. Mais detalhadamente, para o vértice (*p*), as funções definidas nas Equações 1 a 4 podem ser instanciadas respectivamente por:

- ancestrais(p) = {(p), (pf), (pt), (pft)};
- descendentes(p) = {(p), (all)};
- ancestrais_diretos(p) = {(pf), (pt)}; e
- descendentes_diretos(p) = {(all)}.

Na Figura 1 também é exemplificada a relação existente entre visões agregadas e níveis de agregação. Dentro deste contexto, a visão (*pft*) na Figura 1a corresponde ao nível inferior da hierarquia de agregação. A mesma observação é válida para a visão (*pf*) da Figura 1b. Por outro lado, a visão completamente agregada (*all*) corresponde ao nível mais superior da hierarquia de agregação.

Referências bibliográficas

Chaudhuri, S., Dayal, U. (1997). An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1):65-74.

Chaudhuri, S., Dayal, U., Narasayya, V.R. (2011) An overview of business intelligence technology. *Communications of the ACM*, 54(8):88-98.

Ciferri, C., Ciferri, R., Gómez, L., Schneider, M., Vaisman, A., Zimányi, E. (2013). Cube algebra: A generic user-centric model and query language for OLAP cubes. *Journal of Data Warehousing and Mining*, 9(2): 39-65.

Diestel, R. (1997). *Graph Theory*. Springer-Verlag New York, Inc., USA.

Harinarayan, V.; Rajaraman, A.; Ullman, J. D. (1996). Implementing data cubes efficiently. *SIGMOD Record*, 25:205-216.

Wu, M.-C., Buchmann, A.P. (1997). Research Issues in Data Warehousing. In *Proceedings of The German Database Conference*, pages 61-82.

TRABALHO PRÁTICO

Programa. Implemente um programa 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 obtenção por meio da entrada padrão (teclado) das informações que serão usadas para gerar o grafo de derivação. Ou seja, devem ser obtidos:

- Os nomes das dimensões.
- Quando as dimensões forem descritas por atributos, os nomes dos atributos de cada dimensão.
- Quando os atributos de uma dimensão forem relacionados entre si por meio de hierarquias de atributos, a hierarquia propriamente dita.

IMPORTANTE. Todas as informações obtidas devem ser gravadas em disco por meio de um arquivo de dados no modo binário. O modo texto não deve ser usado. A forma de organização do arquivo de dados em campos e registros deve ser definida pelo grupo e deve ser explicada na documentação externa. Por exemplo, considere um determinado campo. Para esse campo deve ser explicado, na documentação externa, seu significado, o porque da escolha de seu tamanho, se ele for de tamanho fixo, ou o porque da escolha dele ser de tamanho variável.

[3] Realize a geração automática de siglas que identifiquem cada dimensão ou atributo que irá representar um vértice do grafo de derivação. Para a dimensão *produto*, exemplos de siglas incluem *p*, *P*, *Pr*.

IMPORTANTE. As siglas também devem ser armazenadas no arquivo de dados usado na funcionalidade [2]. Adicionalmente, deve constar da documentação externa a forma na qual a geração de siglas foi feita.

[4] 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.

[5] Permita a visualização gráfica do grafo de derivação gerado.
IMPORTANTE. Para oferecer essa funcionalidade, é possível usar bibliotecas ou ferramentas apropriadas.

[6] Permita a visualização dos ancestrais, descendentes, ancestrais diretos e descendentes diretos de um vértice *v* do grafo de derivação. O vértice *v* em questão deve ser obtido por meio da entrada padrão (teclado).

RESTRIÇÕES

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

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

[2] 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.

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

[4] 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. Com exceção da funcionalidade [5] (visualização gráfica do grafo de derivação gerado), a implementação não deve ser feita em qualquer outra linguagem de programação. O programa deverá compilar no GCC versão 4.8.2 ou superior.

[5] Conforme já especificado anteriormente, para o oferecimento da funcionalidade [5] (visualização gráfica do grafo de derivação gerado), é possível usar qualquer biblioteca ou ferramenta apropriada. Um exemplo de ferramenta voltada à visualização de grafos pode ser encontrada em: http://www.graphviz.org/

[6] As dimensões, seus atributos e suas respectivas hierarquias de atributos devem ser as mais próximas da realidade. A seguir são descritos alguns exemplos.

- Em uma aplicação de *data warehousing* para um supermercado que analisa seus lucros, pode-se pensar nas medidas numéricas *unidades-vendidas*, *lucro-venda* e nas dimensões *produto*, *filial*, *data*, *promoção*. Quanto às hierarquias de atributos, pode-se pensar em:
  - para a dimensão produto: (*all*) $\preceq$ (*marca*) $\preceq$ (*produto*)
  - para a dimensão filial: (*all*) $\preceq$ (*país*) $\preceq$ (*região*) $\preceq$ (*estado*) $\preceq$ (*cidade*) $\preceq$ (*filial*)
  - para a dimensão data: (*ano*) $\preceq$ (*semestre*) $\preceq$ (*trimestre*) $\preceq$ (*mês*) $\preceq$ (*dia*)
  - para a dimensão promoção: (*all*) $\preceq$ (*tipoPromoção*) $\preceq$ (*promoção*)
- Em uma aplicação de *data warehousing* para um hospital que analisa gastos de produtos, pode-se pensar nas medidas numéricas *unidades-gastas* e nas dimensões *produto*, *setor*, *data*. Quanto às hierarquias de atributos:
  - para a dimensão produto: (*all*) $\preceq$ (*marca*) $\preceq$ (*produto*)
  - para a dimensão data: (*all*) $\preceq$ (*ano*) $\preceq$ (*semestre*) $\preceq$ (*trimestre*) $\preceq$ (*mês*) $\preceq$ (*dia*)
  - para a dimensão setor: (*all*) $\preceq$ (*setor*)
- Em uma aplicação de *data warehousing* para uma rede de hospitais que analisa a taxa de ocupação de instalações de cada hospital, pode-se pensar nas medidas

numéricas *contagem-utilização* e nas dimensões *instalação*, *statusUtilização*, *médico*, *setor*, *hospital*, *data*. Quanto às hierarquias de atributos:

- para a dimensão instalação: (*all*) $\preceq$ (*tipoInstalação*)
- para a dimensão statusUtilização: (*all*) $\preceq$ (*statusUtilização*)
- para a dimensão médico: (*all*) $\preceq$ (*médico*)
- para a dimensão setor: (*all*) $\preceq$ (*setor*)
- para a dimensão hospital: (*all*) $\preceq$ (*país*) $\preceq$ (*região*) $\preceq$ (*estado*) $\preceq$ (*cidade*) $\preceq$ (*hospital*)
- para a dimensão data: (*ano*) $\preceq$ (*semestre*) $\preceq$ (*trimestre*) $\preceq$ (*mês*) $\preceq$ (*dia*)

- Em uma aplicação de *data warehousing* para uma rede de hospitais que realiza o controle de gestantes, pode-se pensar na medida numérica *total-gestantes*, e nas dimensões *dadosDemográficos*, *localExamePreNatal*, *dadosEstatísticosPreNatal*, *hospitalNascimentoBebe*, *data*. Quanto às hierarquias de atributos:
  - para a dimensão dadosDemográficos: (*all*) $\preceq$ (*faixaEtaria*)
  - para a dimensão localExamePreNatal: (*all*) $\preceq$ (*pais*) $\preceq$ (*região*) $\preceq$ (*estado*) $\preceq$ (*cidade*) $\preceq$ (hospital)
  - para a dimensão dadosEstatísticosPreNatal: (*all*) $\preceq$ (*nroExames*)
  - para a dimensão hospitalNascimentoBebe: (*all*) $\preceq$ (*país*) $\preceq$ (*região*) $\preceq$ (*estado*) $\preceq$ (*cidade*) $\preceq$ (hospital)
  - para a dimensão data: (*all*) $\preceq$ (*ano*) $\preceq$ (*semestre*) $\preceq$ (*trimestre*) $\preceq$ (*mês*) $\preceq$ (*dia*)

- Em uma aplicação de *data warehousing* para um departamento que controla os projetos desenvolvidos pelos seus docentes, pode-se pensar nas medidas numéricas *valorTotalConcedido, valorLiberadoCusteio/Capital, Reserva Técnica,* e nas dimensões *projeto, coordenador, órgãoFinanciador, dataInício, dataTérmino e situação*. Quanto às hierarquias de atributos:
  - para a dimensão projeto: (*all*) $\preceq$ (*areaPesquisa*) $\preceq$ (*projeto*)
  - para a dimensão coordenador: (*all*) $\preceq$ (*areaPesquisa*) $\preceq$ (*coordenador*)
  - para a dimensão órgãoFinanciador: (*all*) $\preceq$ (*país*) $\preceq$ (*região*) $\preceq$ (*estado*)

- para a dimensão dataInício: (*all*) $\preceq$ (*ano*) $\preceq$ (*semestre*) $\preceq$ (*trimestre*) $\preceq$ (*mês*) $\preceq$ (*dia*)
- para a dimensão dataTérmino: (*all*) $\preceq$ (*ano*) $\preceq$ (*semestre*) $\preceq$ (*trimestre*) $\preceq$ (*mês*) $\preceq$ (*dia*)
- para a dimensão situação: (*all*) $\preceq$ (*situação*)

- Em uma aplicação de *data warehousing* para um instituto de pesquisa que controla os projetos desenvolvidos pelos seus pesquisadores, pode-se pensar na medida numérica *quantidadePublicação,* e nas dimensões *pesquisador, localPublicação, tipoPublicação, classificaçãoPublicação, data*. Quanto às hierarquias de atributos:
  - para a dimensão pesquisador: (*all*) $\preceq$ (*departamento*) $\preceq$ (*pesquisador*)
  - para a dimensão localPublicação: (*all*) $\preceq$ (localPublicação)
  - para a dimensão tipoPublicação: (*all*) $\preceq$ (*tipoPublicação*)
  - para a dimensão classificaçãoPublicação: (*all*) $\preceq$ (*classificaçãoAgregada*) $\preceq$ (*classificaçãoDetalhada*)
  - para a dimensão data: (*all*) $\preceq$ (*ano*)

[7] 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, 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. 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.

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 executar o programa.
- Documentação externa em formato .pdf.

Instruções de entrega.

[1] Enviar o arquivo compactado para o e-mail labbdciferri@gmail.com, com o seguinte assunto: [AEDII] Trabalho Prático 2016 – Turma X. 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. MAIOR PESO
- Qualidade da documentação externa. 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 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).
- Devem ser exibidos avisos ou mensagens de erro quando apropriado.

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

Bom Trabalho !