

Departamento de Ciência da Computação – Universidade de Brasília  
Programação Sistemática – 1º/2015  
Data de entrega: 24/04/2015, 23:59h.

## **Trabalho Prático – Parte 1**

### **Introdução:**

---

O Sucupira é uma nova e importante ferramenta para coletar informações, realizar análises e avaliações e ser a base de referência do Sistema Nacional de Pós-Graduação (SNPG). A Plataforma Sucupira deve disponibilizar em tempo real e com muito mais transparência para toda a comunidade acadêmica as informações, processos e procedimentos que a CAPES realiza no SNPG. Igualmente, a Plataforma Sucupira propiciará a parte gerencial-operacional de todos os processos e permitirá maior participação das pró-reitorias e coordenadores de programas de pós-graduação. A escolha do nome é uma homenagem ao professor Newton Sucupira, autor do Parecer nº 977 de 1965. O documento conceituou, formatou e institucionalizou a pós-graduação brasileira nos moldes como é até os dias de hoje.

A especificação para o trabalho prático da disciplina de PS em 1/2015 consiste em implementar um sistema para auxiliar na coleta das informações necessárias para o Sucupira a partir dos currículos Lattes dos professores da Pós Graduação em Informática (PPGI) do CIC. O programa deve ser implementado em dois componentes: um que filtra as informações dos currículos Lattes e outro que processa essas informações em relatórios relativas a projetos e publicações dos professores em questão.

### **Parte 1 – Componente que processa os currículos Lattes**

---

#### **Funcionamento do componente**

Essa parte do sistema deverá receber como entrada os currículos lattes de todos os professores do PPGI em XML. Usando a biblioteca JDOM (em JDOM.org) deve-se extrair algumas informações necessárias para criar o arquivo de saída em formato txt contendo as informações relativas a todos os projetos de cada professor. Nesse arquivo deverá conter as informações de acordo com o exemplo a baixo de todos os professores do departamento.

A primeira parte a ser processada por esse componente será a informação relativa ao nome completo e aos diferentes nomes para citação de cada professor conforme a seguinte tag exemplo:

**NOME COMPLETO**="GENAINA NUNES RODRIGUES"

**NOME-PARA-CITACAO** ="RODRIGUES, G. N.;Rodrigues, Genaina Nunes;Rodrigues, Genáina Nunes;RODRIGUES, GENAÍNA N.;RODRIGUES,

A segunda parte a ser extraída do arquivo XML será a informação relativa ao grupo de projetos de cada professor conforme descrito logo abaixo

**<PARTICIPACAO-EM-PROJETO** SEQUENCIA-FUNCAO-ATIVIDADE="4" FLAG-PERIODO="ATUAL" MES-INICIO="" ANO-INICIO="2012" MES-FIM="" ANO-FIM="" ....>

Produzindo como resultado as seguintes informações no arquivo em formato txt:

**ANO-INICIO**="2012" **MES-FIM**="" **ANO-FIM**=""

**NOME-DO-PROJETO**=Uso de Cenarios Implicitos no Suporte a Analise de Dependabilidade em Sistemas a Computacionais

**SITUACAO**=EM\_ANDAMENTO

**NATUREZA**="PESQUISA"

**DESCRICAO-DO-PROJETO**="Cenarios são uma forma de representar diferentes visões e abstrações do software e descrevem dois aspectos de uma sistema: ...

**INTEGRANTES-DO-PROJETO:**

**1.NOME-COMPLETO**="Genaina Nunes Rodrigues" **RESPONSAVEL**="SIM"

**2.NOME-COMPLETO**="ALBA CRISTINA MAGALHAES ALVES DE MELO" **RESPONSAVEL**="NAO"

**3.NOME-COMPLETO**="Rami Bahsoon" **RESPONSAVEL**="NAO"

## Modularização

Para cada um dos módulos desse componente, deverá ser criado um módulo de declaração e um módulo de implementação, Além de serem aplicadas as devidas regras de encapsulamento do código. Lembrando que cada módulo deverá ser possível a compilação independente.

### Módulos obrigatórios do programa:

#### . Módulo de leitura dos XML's .

Esse módulo receberá uma lista de nome e retornara uma lista de documentos XML usando a biblioteca JDOM.

Entrada: Arquivo com os nomes dos professores e XMLs dos currículos.

Saída: lista de documentos que são os currículos dos professores.

#### **. Módulo de tratamento dos XML .**

Esse módulo receberá um conjunto de documentos XML e irá extrair as informações necessárias dentro de cada documento usando a biblioteca JDOM e terá como saída objetos de uma classe criada pelo desenvolvedor para auxiliar na escrita dos arquivos

Entrada: XML dos currículos dos professores

Saída: Lista de uma classe professores que contém as informações dos professores e seus respectivos projetos. (Uma instância da classe deverá conter todos os projetos de cada professor).

#### **. Módulo de persistência dos dados .**

Esse modulo irá receber uma lista dos objetos e irá escrever nos arquivos txt de saída, conforme exemplo supracitado.

Entrada: Lista da classe professores.

Saída: Arquivos txt com as informações extraídas relativas aos campos: *nome completo, nomes para citação e projetos* (com seus respectivos atributos).

### **Controle de qualidade das funcionalidades**

Deve-se criar um módulo controlador de teste (disciplinado) usando o JUnit para testar se as principais funcionalidades e restrições dos módulos de armazenamento, tratamento e persistência dos dados atendem a especificação. O teste disciplinado deve seguir os seguintes passos:

- 1) Antes de testar: produzir um roteiro de teste
- 2) Antes de iniciar o teste: estabelecer o cenário do teste
- 3) Criar um módulo controlador de teste, usando a ferramenta JUnit para testar as principais funcionalidades de cada módulo.
- 4) Ao testar: produzir um laudo em que todas as discrepâncias encontradas são registradas. Esse laudo pode ser uma saída da execução do JUnit. Somente termine o teste antes de completar o roteiro, caso observe que não vale mais a pena continuar executando o roteiro, uma vez que o contexto para o resto está danificado

Após a correção: repetir o teste a partir de 2 até o roteiro passar sem encontrar falhas.

---

## Parte 2 Componente que processa as informações sobre os projetos.

---

### Funcionamento do componente

Esse componente deverá receber como entrada o arquivo texto gerado pelo componente que processa os currículos Lattes e o arquivo de discentes do PPGL .

Deverá ser feito o tratamento dos dados do arquivo de texto e feita a verificação de quais professores do departamento do CIC estão coordenando ou estão fazendo parte de algum projeto na plataforma lattes. Lembrando que o nome constado no arquivo pode ser qualquer um dos nomes presentes na lista de *nomes para citação* de cada professor. Deve-se gerar (em memória) duas listas: uma contendo os docentes do PPGL e seus respectivos nomes para citação e outra lista com o nome dos discentes do PPGL. O nome dos discentes deve ser processado a partir de um arquivo txt fornecido.

Em particular, o campo integrantes de cada projeto deve identificar se os integrantes são docentes, discentes do PPGL ou participante externo. Caso o nome do participante do projeto não conste na lista de docentes ou discentes do PPGL, esse participante deve ser classificado como participante externo. Por fim, deve-se construir um grafo de relacionamento entre docentes a partir dos integrantes de cada projeto.

Logo abaixo segue um exemplo do arquivo de saída:

**NOME COMPLETO**="GENAINA NUNES RODRIGUES"

#### Projetos

##### Projeto 1:

**NOME**=Uso de Cenários Implícitos no Suporte a Análise de Dependabilidade em Sistemas Computacionais

**ANO INICIO**="2012"

**ANO FIM**=""

**SITUACAO**=EM\_ANDAMENTO

**NATUREZA**="PESQUISA"

**DESCRICAO** ="Cenários são uma forma de representar diferentes visões e abstrações do software e descrevem dois aspectos de um sistema: ..."

#### INTEGRANTES

1 Genaina Nunes Rodrigues (**Responsável**)

2 ALBA CRISTINA MAGALHAES ALVES DE MELO (**Docente**)

3 Danilo Filgueira Mendonça (**Discente**)

## **Projeto 2:**

...

### **Grafo de relacionamentos:**

GENAINA NUNES RODRIGUES: A. ZAGHETTO, C. RALHA, V. ALVES

ALEXANDRE ZAGHETTO: B. MACHIAVELLO, C. RALHA.

...

## **Modularização**

Para cada um dos módulos a seguir, deverá ser criado um módulo de declaração e um módulo de implementação, Criação do makefile contendo as diretivas de compilação e geração dos executáveis de todos os módulos desse componente.

### **Módulos obrigatórios do programa:**

#### **. Módulo de leitura dos dados dos arquivos de entrada.**

Esse módulo extrai os dados dos arquivos de entrada e os armazena em uma estrutura de dados à escolha do grupo.

Entrada: Arquivos textos gerados pelo componente em Java e lista dos discentes do PPGI.

Saída: Uma estrutura de dados com as informações dos projetos, outra estrutura relativa a professores e seus nomes para citação e outra estrutura para o nome dos discentes.

#### **. Módulo de tratamento da informação**

Esse módulo trata as informações das estruturas de dados para garantir consistência das informações relativas aos participantes do projeto: docente, discentes ou participantes externos. Além disso ele deve colocar essas informações numa estrutura de dado final que será usada para gravar as informações no arquivo.

Entrada: Uma estrutura de dados com as informações dos projetos, dos docentes e dos discentes.

Saída: Uma estrutura de dados dos projetos com as informações corretas sobre seus integrantes.

#### **. Módulo de criação do grafo**

Esse módulo realiza os cruzamentos dos dados armazenados anteriormente e os organiza em uma estrutura que informa com quais professores do departamento um professor colaborou (ou colabora) em projetos.

Entrada: Uma estrutura de dados com as informações dos projetos e seus integrantes devidamente classificados em docente, discente ou participante externo.

Saída: Uma estrutura preenchida com o grafo de relacionamento.

#### **. Módulo de persistência dos dados**

Esse módulo armazena um arquivo de saída com nome “ProjetosPPGI.txt” o resultado final dos dados processados.

Entrada: Uma estrutura de dados com o projeto e outra relativa ao grafo de relacionamentos.

Saída: Um arquivo texto final com as informações de todos os professores, seus projetos outro arquivo contendo as informações relativas ao grafo de relacionamento.

### **Controle de qualidade das funcionalidades**

Depois de pronto, deve-se criar um módulo controlador de teste (disciplinado) usando o CUnit para testar se as principais funcionalidades e restrições dos módulos de armazenamento, tratamento e persistência dos dados atendem a especificação. O teste disciplinado deve seguir os seguintes passos:

- 1) Antes de testar: produzir um roteiro de teste
- 2) Antes de iniciar o teste: estabelecer o cenário do teste
- 3) Criar um módulo controlador de teste, usando a ferramenta CUnit para testar as principais funcionalidades de cada módulo.
- 4) Ao testar: produzir um laudo em que todas as discrepâncias encontradas são registradas. Esse laudo pode ser uma saída da execução do CUnit. Somente termine o teste antes de completar o roteiro, caso observe que não vale mais a pena continuar executando o roteiro, uma vez que o contexto para o resto está danificado

Após a correção: repetir o teste a partir de 2 até o roteiro passar sem encontrar falhas.

## Parte 2 - Escrita

---

Deverão ser entregues em conjunto com os códigos fonte:

- Um arquivo **LEIAME.TXT** contendo a explicação de como utilizar o(s) programa(s).
- Tantos arquivos **RELATORIO-nome.TXT** quantos forem os membros do grupo. O tema **nome** deve identificar o membro do grupo ao qual se refere o relatório. Estes arquivos devem conter uma tabela de registro de trabalho organizada como a seguir:

### **Data | Horas Trabalhadas | Tipo Tarefa | Descrição da Tarefa Realizada**

Na descrição da tarefa redija uma explicação breve sobre o que o componente do grupo fez. Esta descrição deve estar de acordo com o Tipo Tarefa. Cada Tipo Tarefa identifica uma natureza de atividade que deverá ser discriminada explicitamente, mesmo que, durante uma mesma sessão de trabalho tenham sido realizadas diversas tarefas. Os tipos de tarefa são:

- Estudar
- Especificar os módulos
- Especificar as funções
- Revisar especificações
- Projetar
- Revisar projetos
- Codificar módulo
- Revisar código do módulo
- Redigir casos de teste
- Revisar casos de teste
- Realizar os testes
- Diagnosticar e corrigir os problemas encontrados

### **Observações:**

Dica: Preencha esta tabela de atividades ao longo do processo. **NÃO DEIXE PARA ÚLTIMA HORA, POIS VOCÊ NÃO SE LEMBRARÁ DO QUE FEZ TAL DIA, TAL HORA.** Com relatórios similares a esse você aprende a planejar o seu trabalho.