

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL**  
**INSTITUTO DE INFORMÁTICA**  
**DEPARTAMENTO DE INFORMÁTICA APLICADA**  
**INF01120 - TÉCNICAS DE CONSTRUÇÃO DE PROGRAMAS**  
**PROFESSOR: Marcelo Soares Pimenta (*mpimenta@inf.ufrgs.br*)**  
**Semestre 2009/2**  
**Enunciado do Trabalho Prático**

### **Introdução**

Formar grupos de alunos (sugestão: 2 a 3 membros) e entregar lista de membros ao professor, pessoalmente ou via e-mail (preferencialmente). Trabalhos individuais podem ser aceitos mas a sua aceitação deverá ser analisada caso a caso (explique via email ao professor as suas razões).

### **Atividades a realizar:**

O objetivo do trabalho é a definição, a IMPLEMENTAÇÃO, teste e depuração de um ARPEJADOR, um software que recebe uma CIFRA como entrada e gera (informalmente, 'toca' via acionamento de funções de som) um conjunto (na verdade um conjunto dentre os possíveis, no caso de algumas cifras) de notas que compõem o acorde representado pela cifra.

A entrada da cifra pode ser via teclado ou via leitura em um arquivo tipo TXT (em ambos os casos usando a mesma notação) ou ambos ( ; - ) ;

Para a saída sonora, todos podem usar alguma API ou biblioteca de funções de som. Um exemplo (recomendado mas não obrigatório) é a JavaSound. Duas formas de apresentar o acorde é a sequência de notas tocadas uma após a outra com pequeno intervalo de tempo (formato ARPEJO) ou todas notas simultaneamente (formato ACORDE). O arpejador desenvolvido deve obrigatoriamente apresentar na primeira forma (arpejo) e opcionalmente a segunda (acorde) ou ambas ( ; - ). O nome do acorde - em formato texto descrevendo a Cifra pode ser opcionalmente mostrado (p.ex. A#4/7 corresponde a '**Lá sustenido com quarta com sétima**') - uma demonstração pode ser achada no software exemplado item c) abaixo.

A bibliografia básica inclui:

a) Sobre CIFRAS e Acorde:

[http://pt.wikipedia.org/wiki/Cifra\\_\(música\)](http://pt.wikipedia.org/wiki/Cifra_(música))

<http://pt.wikipedia.org/wiki/Acorde>

b) Sobre JavaSound:

<http://java.sun.com/products/java-media/sound/>

<http://java.sun.com/products/java-media/sound/samples/JavaSoundDemo/>

c) Exemplo de arpejador :

<http://www.cifras.com.br/dicionario/>

(Note que a partir da cifra escolhida, esta ferramenta NAO toca o som, APENAS mostra as teclas (em um teclado) ou cordas (p.ex. em um violão ou cavaquinho) que precisam ser tocadas para gerar o acorde representado pela cifra !!

O trabalho pode ser dividido em partes:

#### **1ª Parte:**

Elaborar uma lista (informal, em português) dos requisitos que o sistema deve possuir no ponto de vista dos autores do projeto. Cada requisito é um item da lista e pode ser do tipo funcional (relativo a funcionalidade do sistema) ou não-funcional (relativo a características de performance, usabilidade, cronograma, orçamento, limitações tecnológicas, entre outras). Sugere-se que cada requisito seja explicitamente rotulado como FUNCIONAL ou NAO-FUNCIONAL.

Neste trabalho em particular, o grupo deve elaborar uma lista dos elementos que fazem parte do modelo e uma lista de quais operações o sistema permitirá que sejam realizadas sobre quais elementos, deixando claro quais as consequências (alteração de aparência, mensagens ao usuário, etc).

### **2ª Parte:**

Nesta parte, o grupo deve PROJETAR a parte funcional do sistema, preferencialmente definindo Classes correspondentes a Tipos Abstratos de Dados que fazem parte do programa. Se for utilizada alguma abordagem não vista em aula, um resumo desta deve ser entregue ao professor junto com a documentação do trabalho, assim como a lista de referências bibliográficas consultadas sobre esta abordagem.

### **3ª Parte:**

Nesta parte, o grupo deve PROJETAR a interface com o usuário do sistema. Para isto, deverá no mínimo estabelecer o perfil básico dos usuários do sistema e as tarefas principais que serão suportadas. Além disto, a documentação deverá incluir uma impressão de cada tela/unidade de apresentação por página, seguida das explicações/argumentações que justifiquem seu *layout* e seu comportamento.

### **4ª Parte:**

Nesta parte, o grupo deve IMPLEMENTAR um protótipo do sistema projetado nas partes anteriores. É definida *a priori* a tecnologia Java para ser usada mas o grupo pode alternativamente escolher a linguagem e a plataforma mais adequados para seu desenvolvimento, obviamente compatível com os elementos e restrições definidos pelo professor, mas a sua aceitação deverá ser analisada caso a caso (explique via email ao professor as suas razões). Reuso de algum código existente é tolerado, mas deve ser sinalizado na documentação. Implementar o protótipo com o máximo possível de funções que foram projetadas. Não esquecer de TESTAR e DEPURAR o que foi implementado ANTES de demonstrá-lo ao professor.

### **5ª Parte:**

Nesta Parte, o grupo deve APRESENTAR as características principais de seu projeto e DEMONSTRAR o sistema para o professor. No final do semestre, cada grupo poderá ser convidado a apresentar as idéias principais de seu trabalho para a turma (20 min no máximo por apresentação). Além disto, a versão implementada do sistema deverá ser apresentada ao professor em uma sessão especial de demonstração em data a ser definida. Assume-se que a documentação final do trabalho será entregue ao professor no máximo até o dia da demonstração. O professor não apenas fará uso da documentação para utilizar o sistema e avaliá-lo mas também poderá questionar oralmente os componentes do grupo a respeito de todas as partes do trabalho. Deve ficar clara nas respostas a participação e o engajamento de todos os indivíduos no trabalho do grupo.

### **Observações:**

O trabalho a ser entregue inclui a documentação associada a TODAS as partes e deve conter também a identificação do grupo (número do grupo, nomes de todos componentes) e todas as suposições feitas durante a realização do trabalho. Definições mais precisas que porventura sejam necessárias serão acrescentadas no decorrer do semestre. Em caso de dúvidas, consulte primeiro a bibliografia disponível (preferencialmente), depois pesquise em outras bibliografias (use a biblioteca e a Internet) e, em caso de necessidade, consulte o professor pessoalmente ou via e-mail. A disciplina prevê algumas aulas de reuniões com o professor para acompanhamento dos projetos e esclarecimento das dúvidas surgidas. Em caso de reunião fora do período de aula, lembre-se de agendar com o professor (via email) para garantir que será atendido. Sem agendamento prévio, a prioridade do professor será sempre atender aos compromissos agendados previamente. Bom trabalho!!

### **Datas importantes:**

- Entrega da Documentação: Documentos e código fonte: até **16/novembro** – 2ª f
- Demonstração– a partir de 4ª f 18/novembro (na prática, a data de demonstração deve ser agendada com o professor)