



UNIVERSIDADE ESTADUAL DE GOIÁS  
UNUCET  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO  
DISCIPLINA ESTRUTURA DE DADOS



# ATIVIDADE PRÁTICA EM ESTRUTURA DE DADOS – ATIVIDADE PARCIAL -1ºVA

Esta atividade é o componente de primeira VA. É o componente para a avaliação e desenvolvimento dos conhecimentos envolvidos na disciplina Estrutura de Dados. O valor dessa atividade é 6,0 (seis) pontos.

Prof. Márcio Giovane Cunha Fernandes, Me.

Goiânia, 23 de abril de 2015.



UNIVERSIDADE ESTADUAL DE GOIÁS  
UNUCET  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO  
DISCIPLINA ESTRUTURA DE DADOS



## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
<b>2. REGRAS DA ATIVIDADE PRÁTICA .....</b>	<b>1</b>
2.1. DESENVOLVIMENTO DO TRABALHO .....	1
2.2. REQUISITOS FUNCIONAIS .....	1
2.3. REQUISITOS NÃO-FUNCIONAIS .....	2
2.4. REQUISITOS TÉCNICOS .....	2
<b>3. FORMA DE AVALIAÇÃO .....</b>	<b>2</b>
<b>4. DATAS .....</b>	<b>3</b>

## 1. INTRODUÇÃO

O trabalho introduzido nesse documento busca a realização de prática em estrutura de dados e catalisar a aplicação dos conceitos de programação de computadores e paradigma de programação orientado a objeto (POO).

O processo de aprendizado exige, além da literatura e da exposição de conteúdo, uma prática empírica cuja dimensão não pode ser abordada apenas durante a realização da disciplina.

Enfrentar-se-á essa realidade na referida disciplina através de um ensaio. Para tal, é proposto a construção de um sistema computacional modelar um pequeno conjunto de operações desempenhadas em atividades bancárias.

Na realização dessa atividade encerrar-se-á os conceitos mínimos necessários à obtenção da nota parcial para a 1ª VA na disciplina de Estrutura de Dados, assunto que esse documento presta-se a esclarecer e definir.

## 2. REGRAS DA ATIVIDADE PRÁTICA

### 2.1. DESENVOLVIMENTO DO TRABALHO

1. Elaborar um software para representar vários bancos financeiros, suas contas correntes e contas poupanças e algumas operações.

### 2.2. REQUISITOS FUNCIONAIS

1. Criar bancos financeiros informando o nome do banco, o número da agência que esse banco representa, o endereço e a bandeira (itaú, bradesco, caixa econômica, etc...);
2. Criar conta poupança informando dados do cliente (nome, CPF, endereço), número da conta, número da agência e taxa de rendimento ao mês;
3. A conta poupança deve permitir o cálculo de rendimentos;
4. Criar conta corrente informando dados do cliente (nome, CPF, endereço), número da conta, número da agência;
5. Excluir uma conta poupança ou conta corrente específica;
6. Excluir todas as contas de um banco;
7. Realizar depósitos e saques nas contas poupança e conta corrente;
8. Pesquisar contas por dados do cliente (nome, CPF);
9. Pesquisar contas pelo número da conta;
10. Transferir dinheiro entre contas de um mesmo banco;

11. Transferir dinheiro entre contas de bancos diferentes;
12. Listar todas as contas de um banco;
13. Listar uma conta de um banco, que será escolhida por nome, ou por cpf, ou por número da conta;
14. Pesquisar um banco a partir do nome, bandeira ou número de agência;
15. Listar todos os bancos criados.

### **2.3. REQUISITOS NÃO-FUNCIONAIS**

1. O banco não pode ser criado sem os dados apontados pelo item 2.2.1;
2. O sistema deve gerar automaticamente o número das contas de um banco;
3. Durante a criação de uma conta corrente ou poupança, os dados número da conta e número da agência devem ser obtidos do banco responsável por tal conta;
4. A conta poupança recebe a quantidade de meses em que o dinheiro ficará aplicado e informa o lucro obtido para tal quantidade;
5. Depósitos não podem receber valor nulo (zero) ou negativo;
6. Saques não podem exceder o saldo de uma conta;
7. As transferências de dinheiro entre contas no mesmo banco e entre bancos diferentes devem ser protegidas de eventuais erros (exemplo: erro de número de conta, valores inapropriados etc.);
8. Todas as interações usuário/sistema definidos no item 2.2 devem estar disponíveis em um menu para a intervenção do usuário;
9. A interação com o usuário deve ser fácil e de boa aparência;
10. Um banco não pode ser excluído se possuir contas ativas.

### **2.4. REQUISITOS TÉCNICOS**

1. Uso da linguagem java, C++ ou C;
2. Uso da estrutura de dados Lista para representar todas as coleções caracterizadas no problema proposto (lista de bancos, lista de contas etc.);
3. Uso do paradigma procedural ou paradigma orientado a objeto;
4. Sistema funcionando completamente.

## **3. FORMA DE AVALIAÇÃO**

1. Apresentação à sala;
2. Realização em grupos com quantidade de alunos entre 4 e 5 pessoas;
3. A presença na apresentação dos outros grupos é obrigatória;
4. O professor fará perguntas durante a apresentação.
5. Dia 24/04/2015 acontecerá a formação dos grupos e o sorteio da data e sequência de



apresentação;

6. Apenas um aluno apresentará o trabalho;
7. O aluno a apresentar o trabalho será escolhido de duas formas: vontade do professor ou sorteio (forma definida no dia da apresentação);
8. Os itens deste projeto que serão avaliados constam nos itens 2.2 e 2.3 com o valor de 4,0 (pontos); os demais 2,0 pontos serão de conceito do professor considerando o cuidado, a organização, clareza e qualidade da apresentação do trabalho;
9. Questões não arroladas nesse documento serão resolvidas em debate entre os reclamantes e o professor.

#### **4. DATAS**

- Apresentação dias 15 e 20 de abril de 2015.