



## PLANO DE APRENDIZAGEM

<b>1. DADOS DE IDENTIFICAÇÃO</b>				
<b>Curso:</b> Bacharelado em Sistemas de Informação				
<b>Disciplina:</b> Linguagem de Programação II			<b>Código:</b> SIF4N182	
<b>Professor:</b> Me. Igor Peterson Oliveira Santos			<b>e-mail:</b> igor.santos@fasete.edu.br	
<b>CH Teórica:</b> 40h	<b>CH Prática:</b> 20h	<b>CH Híbrida:</b> 20h	<b>CH Total:</b> 80h	<b>Créditos:</b> 04
<b>Pré-requisito:</b> Fundamentos de Programação				
<b>Período:</b> IV		<b>Ano:</b> 2019.2		

### 2. EMENTA:

Fundamentos de Orientação a Objetos: termos básicos, objetos, atributos e métodos, encapsulamento, mensagens, classes, herança, hierarquia de classes, objetos compostos, polimorfismo.

Uso da orientação a objetos na Linguagem Java: definição de classes, visibilidade dos membros de uma classe, redefinição de métodos, compatibilidade entre objetos, construtores e destrutores, operadores de classe. Tratamento de exceções.

Introdução ao Netbeans: ambiente de programação, eventos e procedimentos, visão geral dos componentes. Desenvolvimento de *software* utilizando arquitetura em camadas. Biblioteca de Classes para Construção de Janelas: swing.

### 3. COMPETÊNCIAS E HABILIDADES DA DISCIPLINA:

- Realizar a identificação de problemas que possuam solução algorítmica. As soluções se darão por meio da modelagem das classes do sistema e como este se comportará para resolver problemas reais.
- Resolver problemas usando ambientes de programação. Para alcance disto, será essencial o conceito de Orientação a Objetos (OO), na linguagem de programação Java, com a aplicação no projeto Desktop.

### 4. OBJETIVO GERAL DA APRENDIZAGEM:

Desenvolver no aluno a capacidade de interpretar problemas de pequeno e médio porte, além de capacitá-los para planejar e codificar *softwares* baseado no paradigma orientado a objetos.

### 5. CONTEÚDOS:

#### 5.1: 1ª ETAPA

##### 5.1.1 CONTEÚDOS PRESENCIAIS (30 horas)

1. Apresentação da Disciplina (01h)
  - i. Introdução a OO e Java
2. Estrutura Básica da linguagem Java (04h)
  - i. Tipos de Dados
  - ii. Variáveis e Métodos



- iii. Operadores
- iv. Controle de Fluxo
- 3. Orientação a Objetos (OO) (06h)
  - i. Conceitos de OO
  - ii. Definição de Objetos e Classes
- 4. Listas (03h)
- 5. Construtores (03h)
- 6. Encapsulamento (03h)
- 7. Herança (04h)
- 8. Polimorfismo (03h)
- 9. Classes Abstratas (02h)
- 10. Interfaces (01h)

#### **5.1.2 CONTEÚDOS NO AMBIENTE VIRTUAL DE APRENDIZAGEM (10 Horas)**

- 1. Lista01 - Introdução à Java (Controles de Fluxo) (2,5h)
- 2. Lista02 - Objetos, Classes, Encapsulamento e Construtores (2,5h)
- 3. Lista03 - Polimorfismo e Herança (2,5h)
- 4. Lista04 - Classes Abstratas e Interfaces (2,5h)

### **5.2: 2ª ETAPA**

#### **5.2.1 CONTEÚDOS PRESENCIAIS (30 horas)**

- 1. Erros e Exceções (03h)
- 2. Arquitetura MVC (03h)
- 3. Testes de Unidade (06h)
- 4. Acesso a Banco de Dados (09h)
- 5. Programação Interface Gráfica (09h)
  - i. AWT
  - ii. Modelo de Eventos
  - iii. Swing

#### **5.2.2 CONTEÚDOS NO AMBIENTE VIRTUAL DE APRENDIZAGEM (10 Horas)**

O quadro abaixo corresponde às fases de entrega do projeto. Será considerado a entregas incrementais, de modo que o professor consiga acompanhar e ver o desempenho no processo de desenvolvimento do *software* das equipes.

O cronograma com as datas de entrega ficará a cargo do professor e estará disponível para as equipes no início da 2ª etapa. Caso o aluno não entregue a atividade no prazo estabelecido, receberá nota zero, sem exceção. Na fase de entrega da apresentação, se o *software* não executar será atribuída a nota zero para o projeto independente das notas parciais. A atribuição das notas e atividades, pelo professor, seguirá o cronograma de fases como segue:

<b>Fases</b>	<b>Atividade</b>	<b>Valor</b>
Fase 1	Início do Projeto e distribuição dos temas	-
Fase 2	Criação do Projeto com seus pacotes MVC e entidades	0,5
Fase 3	Implementação das controladoras e DAOS	1,5
Fase 4	Criação das telas do sistema	1,0
Fase 5	Implementação das funcionalidades parciais do sistema	1,0
Fase 6	Apresentação do Projeto Completo + entrega do documento de <i>software</i>	6,0
<b>Total</b>		<b>10,0</b>



## **6. METODOLOGIA DO TRABALHO:**

Os procedimentos a serem adotados serão os seguintes: aulas expositivas intercaladas com aulas práticas onde serão feitos pequenos protótipos de programas utilizando os conceitos apresentados; projeto de desenvolvimento de um *software* completo.

### **1ª ETAPA:**

#### 1) Práticas de Programação (10,0 pontos):

Serão propostos problemas e/ou maratona de programação a serem resolvidos individualmente ou em duplas sobre um determinado conteúdo abordado em sala sobre Programação Orientado a Objetos, são eles: Classes, Objetos, Herança, Polimorfismo, Encapsulamento e outros. O aluno poderá realizar estudos específicos e apresentar as soluções para os problemas indicados pelo professor.

#### 2) Avaliação Processual (10,0 pontos)

A avaliação escrita será composta por questões subjetivas, versando os temas discutidos até o momento apresentado na 1ª Etapa da disciplina. A avaliação será individual, e aplicada uma semana antes da Avaliação Institucional. Caso ocorram alterações no Calendário Acadêmico de 2019, esta data poderá ser alterada, e devidamente comunicada aos estudantes.

#### 3) Avaliação Institucional (10,0 pontos)

A avaliação escrita será composta por questões objetivas e subjetivas, versando sobre todos os temas discutidos na 1ª Etapa da disciplina. A avaliação será concebida no mesmo modelo estrutural de elaboração de questões utilizados nas provas do ENADE, suas regras e orientações estarão expostas no campo de instruções. A avaliação será individual e aplicada dentro do calendário acadêmico. Caso ocorram alterações no Calendário Acadêmico de 2019, esta data poderá ser alterada e devidamente comunicada aos discentes.

### **2ª ETAPA**

#### 4) Aprendizagem Baseada em Problemas (ABP) (10,0 pontos)

Para a resolução do projeto final e gráfico da disciplina serão propostos problemas e situações reais de como os alunos poderão criar relatórios, telas e gerenciar as regras de negócios do sistema. Isso será fundamental para um aprendizado amplo e disseminado para a turma, uma vez que todos deverão expor as propostas de resolução identificados dos problemas. O arquivo detalhado que corresponde ao projeto da disciplina encontra-se em Apêndice I.

#### 5) Avaliação Processual (10,0 pontos)

Esta avaliação segue sob as mesmas diretrizes do item 2) Avaliação Processual.

#### 6) Avaliação Institucional (10,0 pontos)

Esta avaliação segue sob as mesmas diretrizes do item 3) Avaliação Institucional. A avaliação será individual, e aplicada dentro do calendário acadêmico. Caso ocorram alterações no Calendário Acadêmico de 2019, esta data poderá ser alterada, e devidamente comunicada aos discentes.



### **7. RECURSOS:**

<input type="checkbox"/> Sala (comum)	<input type="checkbox"/> Sala Configuração Flexível	<input checked="" type="checkbox"/> Laboratório(s) - agendar
<input type="checkbox"/> Práticas em Campo	<input checked="" type="checkbox"/> Kit multimídia	<input type="checkbox"/> Outros (informar)

### **8. SISTEMA DE AVALIAÇÃO:**

#### **1ª ETAPA**

- Práticas de Programação (10,0 pontos)
- Avaliação Processual (10,0 pontos)
- Avaliação Institucional (10,0 pontos)

#### **2ª ETAPA**

- Aprendizagem Baseada em Problemas (ABP) (10,0 pontos)
- Avaliação Processual (10,0 pontos)
- Avaliação Institucional (10,0 pontos)

#### **Observações:**

- **Média das Etapas:** será igual a média aritmética das avaliações nas respectivas etapas.
- **Detalhes das atividades no item 6. METODOLOGIA DO TRABALHO**

### **9. ATENDIMENTO EXTRA CLASSE:**

Em caráter complementar, o professor oferece atendimento, diariamente, através do endereço eletrônico: [igor.santos@fasete.edu.br](mailto:igor.santos@fasete.edu.br)

### **10. BIBLIOGRAFIA BÁSICA:**

ANSELMO, Fernando. **Aplicando Lógica Orientada a Objetos em Java**. Florianópolis: Visual Books, 2005.

CORNELL, Gary; HORSTMANN, Cay S. **Core Java 2: Fundamentos**. v. 1. São Paulo: Pearson, 2004.

SIERRA, Kathy; BATES, Bert. **SCJP: Certificação Sun para Programador Java 5: Guia de Estudo**. Rio de Janeiro: Alta Books, 2007.

SIERRA, Kathy; BATES, Bert. **Use a Cabeça! Java**. Rio de Janeiro: Alta Books, 2007.

### **11. BIBLIOGRAFIA COMPLEMENTAR:**

CANTÚ, Marco. **Dominando o delphi 7: a bíblia**. São Paulo: Pearson education,

COAD, Peter e YOURDON, Edward. **Object-Oriented Analysis**. Prentice-Hall, 2000.

DEITEL, H. M. **Java: Como Programar**. Porto Alegre: Bookman, 2003.



JOSUTTIS, Nicolai M. **Object-oriented programming in C++**. Wiley Computer Publishing, 2002.

## **12. LEITURA COMPLEMENTAR:**

## **13. INFORMAÇÕES COMPLEMENTARES:**

O aluno que faltar a prova escrita da 1ª chamada terá o direito de fazer uma prova de Segunda Chamada, cujo conteúdo abrangerá todo o programa. O aluno não terá direito de fazer Segunda Chamada para substituir as notas das listas de exercícios, atividades em sala ou do projeto.

- A verificação do rendimento escolar compreende: frequência e aproveitamento nos estudos, os quais devem ser atingidos conjuntamente;
- É exigida a frequência mínima (FS) 75% (setenta e cinco por cento) das atividades;
- O aluno com frequência suficiente (FS), e média das notas de avaliações do semestre entre 3,0 (três) e 7,0 (sete vírgula zero) terá direito à prova final no semestre;
- Será atribuída nota 0 (zero) ao aluno que não comparecer às atividades ou não apresentar trabalhos no prazo estabelecido;
- Todas as avaliações serão expressas em notas graduadas de 0 (zero) a 10 (dez); e,
- A nota mínima de aprovação na disciplina é 5,0 (cinco vírgula zero). A AVALIAÇÃO FINAL, abrangerá todo o conteúdo do programa.

## **14. APROVAÇÃO:**

Aprovado em \_\_\_\_/\_\_\_\_/\_\_\_\_

Homologado em \_\_\_\_/\_\_\_\_/\_\_\_\_

**COORDENADOR**

**GERÊNCIA ACADÊMICA**

OBS: As datas das avaliações poderão sofrer alterações de acordo com o disciplinado pela secretaria acadêmica da FASETE.



## APÊNDICE I

### **PROJETO DA DISCIPLINA LINGUAGEM DE PROGRAMAÇÃO II**

#### **1- INTRODUÇÃO**

O Desenvolvimento para a área de trabalho (*Desktop* em inglês) é um termo amplo que descreve o processo de criar *software* que será executado em computadores padrão, incluindo seu *desktop*, *laptop* ou computador de uso geral. O *software* em desenvolvimento pode ser um *software* de sistema destinado a ajudar seu sistema operacional ou *software* aplicativo. O *software* aplicativo é criado para executar uma única tarefa ou um conjunto de tarefas relacionadas, incluindo jogos, processadores de texto e aplicativos empresariais personalizados.

Sendo assim, a disciplina de Linguagem de Programação II (LP2), do 4º período do curso de Sistemas de Informação da Faculdade Sete de Setembro (Fasete), busca propagar conhecimento através de um projeto de desenvolvimento de *Software Desktop* em parceria com as disciplinas de Engenharia de *Software 2* e Banco de Dados, concebendo assim o projeto interdisciplinar. Na disciplina de LP2 o professor responsável tem a preocupação de guiar os alunos com o uso da linguagem de programação Java e os demais conhecimentos necessários para o sucesso da construção do programa.

#### **2- PROFESSOR MENTOR**

O professor mentor responsável será Me. Igor Peterson Oliveira Santos, cujo far-se-á presente para guiar os alunos nas soluções dos problemas de *software* e tecnologias adotadas para o desenvolvimento dos programas de *software*.

#### **3- EQUIPE DE ALUNOS**

Para os projetos a serem desenvolvidos, poderão ser formadas equipes de até dois alunos. Estes deverão estar regularmente matriculados na disciplina de LP2, sem exceção.

#### **4- PERÍODO**

O período de duração do projeto é de aproximadamente 4 meses. O tempo é referente à carga horária da disciplina de LP2, distribuída em média nessa quantidade de meses.

#### **5- OBJETIVO**

O objetivo principal deste projeto é a construção de um programa de *software desktop* desenvolvido em Java (2019) usando a IDE de desenvolvimento Netbeans (2019) para resolução de um problema. O projeto será construído baseado na proposta de Aprendizagem Baseada em Problema (ABP), logo os alunos terão a responsabilidade de resolver problemas do cotidiano detectados no parâmetro socioeconômico de Paulo Afonso e região.



## 6- JUSTIFICATIVA

A ABP representa um método de aprendizagem que tem por base a utilização de problemas como ponto de partida para a aquisição e integração de novos conhecimentos. Em essência, promove uma aprendizagem centrada no aluno, sendo os professores meros facilitadores do processo de produção do conhecimento. Nesse processo, os problemas são um estímulo para a aprendizagem e para o desenvolvimento das habilidades de resolução (Barrows, 1986 apud Dourado e Souza, 2015).

Durante o processo de construção serão propostos problemas e situações reais de como os alunos poderão criar relatórios, telas e gerenciar as regras de negócios do sistema. Isso será fundamental para um aprendizado amplo e disseminado para a turma, uma vez que todos deverão expor as propostas de resolução identificados dos problemas.

Acredita-se que ao término do projeto o aluno estará mais apto para resolver problemas através de *softwares*, além de compreender melhor a integração de todo o conhecimento adquirido até o atual período cursado, disseminando o conhecimento das disciplinas e preparando inicialmente o aluno para estar apto para o mercado de trabalho.

## 7- FASES E DISTRIBUIÇÃO DAS NOTAS

O quadro abaixo corresponde às fases de entrega do projeto. Será considerado a entregas incrementais, de modo que o professor mentor consiga acompanhar e ver o desempenho no processo de desenvolvimento do *software* das equipes.

O cronograma com as datas de entrega ficará a cargo do professor e estará disponível para as equipes no início das aulas da disciplina. Caso o aluno não entregue a atividade no prazo estabelecido, receberá nota zero, sem exceção. Na fase de entrega da apresentação, se o *software* não executar será atribuída a nota zero para o projeto independente das notas parciais.

A atribuição das notas e atividades, pelo professor, seguirá o cronograma de fases como segue:

<b>Fases</b>	<b>Atividade</b>	<b>Valor</b>
Fase 1	Início do Projeto e distribuição dos temas	-
Fase 2	Primeira versão do documento contendo o minimundo	<b>0,5</b>
Fase 3	Modelo Conceitual e Lógico do Banco	<b>1,0</b>
Fase 4	Diagrama de: Casos de Usos e Classes	<b>0,5</b>
Fase 5	Implementação das: Classes básicas, scripts de criação de tabelas	<b>0,5</b>
Fase 6	Implementação das coleções de dados	<b>1,0</b>
Fase 7	Implementação da Arquitetura de Camadas	<b>0,5</b>
Fase 8	Implementação dos Cadastros e fachada	<b>0,5</b>
Fase 9	Implementação da GUI dos demais Casos de Uso	<b>1,0</b>
Fase 10	Implementação das Regras de Negócio e Tratamento de Exceções	<b>0,5</b>
Fase 11	Apresentação dos Projetos + entrega do documento de <i>software</i>	<b>4,0</b>
<b>Total</b>		<b>10,0</b>



## 8- RECOMENDAÇÕES FINAIS

Na apresentação os alunos deverão entregar a documentação do *software* por meio de uma cópia impressas de acordo com o cronograma. A parte referente a implementação deverá ser enviada para o e-mail do professor da disciplina de LP2 pelo e-mail <igor.santos@fasete.edu.br>, não sendo necessário entregar cópias físicas do código.

## REFERÊNCIAS

JAVA. **Linguagem de Programação Orientada a Objetos Java**. Disponível em <[http://www.java.com/pt\\_BR/](http://www.java.com/pt_BR/)>. Acesso em janeiro de 2019.

NETBEANS IDE. **Ambiente de Desenvolvimento Integrado Java**. Disponível em <<http://netbeans.org/>>. Acesso em janeiro de 2019.

SOUZA, S. C.; DOURADO, L. **Aprendizagem Baseada Em Problemas (Abp): Um Método De Aprendizagem Inovador Para O Ensino Educativo**. HOLOS, v5, 2015.



