



Plano de Ensino

1. Dados de Identificação

Curso: **Ciência da Computação**

Componente Curricular: **Engenharia de Software II**

Fase: **6º**

Ano/Semestre: **2015/2**

Número da turma: **11581**

Número de Créditos: **4**

Carga horária - Hora Aula: **72**

Carga horária - Hora Relógio: **60**

Professora: **Graziela Simone Tonin**

Atendimento ao aluno: **sexta-feira das 19 às 21hs**

2. Objetivo Geral do Curso

O curso tem por objetivo a formação integral de novos cientistas e profissionais da computação, os quais deverão possuir conhecimentos técnicos e científicos e serem capazes de aplicar estes conhecimentos, de forma inovadora e transformadora, nas diferentes áreas de conhecimento da Computação. Adicionalmente, os egressos do curso deverão ser capazes de adaptar-se às constantes mudanças tecnológicas e sociais, e ter uma formação ao mesmo tempo cidadã, interdisciplinar e profissional.

3. Ementa

Análise de Requisitos. Técnicas existentes para a extração de requisitos. Modelos de Processos existentes para o desenvolvimento de aplicações orientadas a objetos. O paradigma orientado a objetos. Técnicas de análise orientada a objetos. Utilização de uma ferramenta de apoio para a elaboração dos modelos de análise.

5. Objetivo

5.1 Geral

- Compreender as diversas técnicas de levantamento de requisitos e métodos de análise. Analisar e projetar sistemas informatizados utilizando uma metodologia orientada a objetos.

5.2 Específicos

- Compreender as diferentes técnicas existentes para extração de requisitos;
- Conhecer os modelos de processos existentes para o desenvolvimento de aplicações orientadas a objetos;
- Conhecer o método de análise e projeto orientado a objetivos utilizando UML;
- Utilizar uma ferramenta de apoio para a elaboração dos modelos da UML;
- Projetar e desenvolver um software, onde o aluno possa vivenciar a prática necessária para entender os conteúdos estudados.

6. Cronograma e Conteúdo Programático

Semana	Assunto
06/08/2015	Apresentação da disciplina. Levantamento de Requisitos Processo de engenharia de requisitos. Ferramentas para gerir requisitos. Validação de requisitos.
13/08/2015	Apresentação de trabalhos sobre as ferramentas. Introdução a UML. Formação dos times e definição de que software será desenvolvido. Apresentação da organização da equipe. Configuração de ambiente.
20/08/2015	Apresentação da Evolução do Projeto do Software. (Primeira reunião com cliente) Casos de Uso Especificação/documento de requisitos através de descrição dos casos de uso. Desenvolvimento do projeto
27/08/2015	Ferramenta de modelagem UML Astah Community. Aplicação dos conceitos aprendidos no projeto. Apresentação da Evolução do Projeto do Software. (Definição de requisitos, tecnologias, ambiente e resultado da segunda reunião com cliente).
03/09/2015	O paradigma OO e a UML. RUP – Rational Unified Process. Requisitos do projeto definido. Configuração do ambiente e início do desenvolvimento. Casos de uso
10/09/2015	Diagrama de classes: classes, atributos, métodos, tipos de relacionamentos e associações. Aplicação dos conceitos no projeto.
17/09/2015	Diagrama de objetos. Primeira parcial de apresentação e avaliação do andamento do projeto.
24/09/2015	Modelo conceitual X modelo de domínio. Diagrama de pacotes. Desenvolvimento do Projeto.
01/10/2015	Diagrama de sequência. Diagrama de comunicação. Desenvolvimento do Projeto.
08/10/2015	Diagrama de atividades. Desenvolvimento do Projeto.
15/10/2015	Diagrama de máquina de estados. Desenvolvimento do Projeto.
22/10/2015	Visão geral dos outros diagramas da UML. Desenvolvimento do Projeto.
29/10/2015	Padrões de projeto. Projeto navegacional. Noções de usabilidade e critérios ergonômicos.
05/11/2015	Usabilidade. E boas práticas de IHC. Desenvolvimento do Projeto. Apresentação final do trabalho.
12/11/2015	Prova de recuperação NP2.

<p>Total: 72 h/aula</p>

7. Procedimentos Metodológicos (estratégias de ensino, equipamentos, entre outros)

Conduzir a disciplina com aulas expositivas/dialogadas enquanto discutidos os itens de cunho teórico, evoluindo em tópicos específicos para exercícios práticos, demonstrações, contextualização baseada em publicações atualizadas. Uso de atividades em laboratórios com o objetivo de apresentar/exercitar os conceitos estudados.

O plágio e a cola serão tratados de forma rígida (nota 0 para os envolvidos na atividade em questão).

8. Avaliação do Processo Ensino-Aprendizagem

Uso de abordagens tais como: provas teóricas, avaliação escrita em aula, exercícios extra-classe, desenvolvimento de um projeto real, entre outros.

Cinquenta por cento da nota é relativa a aplicação das técnicas de engenharia de software no desenvolvimento de um software e os outros cinquenta por cento será aplicado uma prova. Sendo a nota final relativo à média simples da soma das duas notas.

Será ofertada prova aos estudantes que não obtiveram média maior ou igual a 6,0 .

A reposição será das avaliações e substitutiva. Os trabalhos não poderão ser recuperados.

8.1 Recuperação: Novas Oportunidades de Aprendizagem e Avaliação.

O aluno terá direito a uma prova de recuperação caso não tenha obtido a nota mínima para aprovação.

9. Referências

9.1 Básicas

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML Guia do Usuário**. Rio de Janeiro: Campus, 2006.

WAZLAWICK, R. S. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. Rio de Janeiro: Campus, 2004.

GUEDES, G. T. A. **UML 2 – Uma Abordagem Prática**. São Paulo: Novatec, 2009.

RUMBAUGH, J.; BLAHA, M. **Modelagem e Projetos Baseados em Objetos**. Rio de Janeiro: Campus, 2004.

9.2 Complementares

SOMMERVILLE, I. **Engenharia de Software**. 8. ed. São Paulo: Addison - Wesley, 2008.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. São Paulo: Bookman Companhia Ed., 2006.

PFFLEGER, S. L. **Engenharia de Software**. 2. ed. São Paulo: Prentice-Hall, 2004.

LARMAN, C. **Utilizando UML e Padrões: uma Introdução à Análise e ao Projeto Orientados a Objetos**. 3. ed. São Paulo: Bookman Companhia, 2007.