

Contextualização

O desenvolvimento de software no mercado requer cada vez mais o conhecimento dos processos de negócio e as informações que são produzidas. O valor agregado da tecnologia nas empresas está centrado no potencial dos sistemas em extrair conhecimento e colaborar para as estratégias e tomadas de decisão.

Neste contexto a modelagem dos sistemas tem uma importância fundamental, na medida em que oferece suporte para investigação, análise e validação dos procedimentos apreendidos durante as etapas do processo de requisitos. Observa-se que quanto maior a aderência à realidade do usuário, maior será o sucesso nos resultados obtidos.

Construímos modelos para comunicar a estrutura e o comportamento desejado do sistema. Construímos modelos para visualizar e controlar a arquitetura do sistema. Construímos modelos para compreender melhor o sistema que estamos elaborando, muitas vezes expondo oportunidades de simplificação e reaproveitamento. Construímos modelos para gerenciar os riscos.

A UML (Unified Modelling Language), linguagem de representação utilizada nessa disciplina, oferece uma diversidade de modelos para representação das partes físicas e lógicas do sistema em desenvolvimento. Os modelos são integrados e, a todo o momento, poderá ser necessário retornar aos modelos anteriormente construídos e realizar algum tipo de correção. Os modelos fornecem múltiplas visões do sistema a ser modelado, analisando-o e modelando-o sob diversos aspectos, procurando-se assim atingir a completude da modelagem.

A capacidade de representação do negócio por meio de modelos da UML e ter visibilidade para a construção do sistema são competências que devem ser desenvolvidas no aluno nesta disciplina.

Ementa

Conceitos Básicos de Modelagem; A Linguagem UML; O Ciclo de Vida Iterativo e Incremental; Utilizando UML no Ciclo de Vida ? Concepção, Elaboração, Construção e Transição; Diagramas UML 2.0 no ciclo de vida do desenvolvimento de software.

Objetivos Gerais

Solucionar problemas do mundo real, fazendo uso da linguagem UML na representação de modelos.

Objetivos Específicos

- Conhecer o ciclo de vida iterativo e incremental, utilizado no desenvolvimento de software orientado a objetos;
- Conhecer os princípios e práticas do Processo Unificado - RUP;
- Definir a ordem das iterações do desenvolvimento de software;
- Identificar requisitos funcionais e não-funcionais para representação em modelos de software;
- Utilizar os diagramas propostos pela UML;
- Construir modelos de software baseados na UML;
- Analisar a melhor forma de representação do negócio;
- Empregar as técnicas de acordo com a natureza do modelo a ser desenvolvido.

Conteúdos

UNIDADE 1 - Conceitos Básicos de Modelagem

- 1.1. A Importância da Modelagem
- 1.2. Princípios de Modelagem
- 1.3. Análise e Projeto Orientados a Objeto

UNIDADE 2 - A Linguagem UML (Unified Modeling Language)

- 2.1 Introdução a UML
- 2.2 Visões da UML
- 2.3 Síntese Geral dos Diagramas UML
- 2.4 Ferramentas CASE (Computer-Aided Software Engineering) baseadas na UML

UNIDADE 3 - O Ciclo de Vida Iterativo e Incremental

- 3.1 Apresentação
- 3.2 Etapas e Disciplinas
- 3.3 Técnicas e modelos aplicados
- 3.4 Definição das iterações

UNIDADE 4 - Utilizando UML no Ciclo de Vida

4.1 Fase - Concepção (ênfase no escopo do sistema)

4.1.1 Diagrama de Caso de Uso

- Apresentação
- Notação
- Aplicação
- Descrição de Caso de Uso

4.1.2 Diagrama de Classe - Modelo de domínio

- Apresentação
- Notação
- Aplicação

4.1.3 Diagrama de Objetos

- Apresentação
- Notação
- Aplicação

4.1.2 Diagrama de Pacotes

- Apresentação
- Notação
- Aplicação

4.2 Fase - Elaboração (ênfase na arquitetura do projeto)

4.2.1 Modelo de Classes de Projeto

- Definição da Visibilidade entre Objetos
- Adição de Operações
- Adição de Interfaces
- Relacionamentos de Dependência
- Adição de Classes Utilitárias e de Coleções

4.2.2 Diagramas de Interação

- Diagrama de Sequencia
- Diagrama de Comunicação
- Diagrama de Visão Geral da Interação
- Diagrama de Temporização

4.2.3 Diagrama de Estado

- Apresentação
- Notação
- Aplicação

4.2.4 Diagrama de Atividades

- Apresentação
- Notação
- Aplicação

4.3 Fase Construção - ênfase no desenvolvimento

4.3.1 Diagrama de Componentes

- Apresentação
- Notação
- Aplicação

4.4 Fase Transição - ênfase na implantação

4.4.1 Diagrama de Implantação

- Apresentação

Procedimentos de Avaliação

O processo de avaliação oficial será composto de três etapas: Avaliação 1 (AV1), Avaliação 2 (AV2) e Avaliação 3 (AV3). Sendo AV2 e AV3 unificadas a partir de um banco de questões proposto pelos professores da Estácio de todo o Brasil;

As avaliações poderão ser realizadas através de provas teóricas, provas práticas e realização de projetos ou outros trabalhos, representando atividades acadêmicas de ensino de acordo com as especificidades de cada disciplina. A soma de todas as atividades que possam vir a compor o grau final de cada avaliação não poderá ultrapassar o grau máximo de dez, permitindo-se atribuir valor decimal às avaliações. Caso a disciplina, atendendo ao projeto pedagógico de cada Curso, além de provas teóricas e/ou práticas, contemple outras atividades acadêmicas de ensino, estas não poderão ultrapassar 20% da composição do grau final;

A AV1 contemplará o conteúdo da disciplina até a sua realização, incluindo o das atividades estruturadas;

As AV2 e AV3 abrangerão todo o conteúdo da disciplina, incluindo o das atividades estruturadas.

Para aprovação na disciplina, o aluno deverá:

1. Atingir resultado igual ou superior a seis, calculado a partir da média aritmética entre os graus das avaliações, considerando-se apenas as duas maiores notas obtidas dentre as três etapas de avaliação (AV1, AV2 e AV3) - a média aritmética obtida será o grau final do aluno na disciplina;
2. Obter grau igual ou superior a quatro em, pelo menos, duas das três avaliações;
3. Frequentar, no mínimo, 75% das aulas ministradas.

As disciplinas oferecidas na modalidade de Educação a Distância (EaD) seguirão o mesmo critério de avaliação das disciplinas presenciais.

Para a avaliação do Trabalho de Conclusão de Curso (TCC), ou trabalhos de mesma natureza, será atribuído grau único para a disciplina que, para aprovação do aluno, deverá ser igual ou maior do que seis.

Mais detalhes: Portaria D.E. n.º 02, de 18 de novembro de 2009.

A AV1 contemplará o conteúdo da disciplina até a sua realização, incluindo o das atividades estruturadas;

As AV2 e AV3 abrangerão todo o conteúdo da disciplina, incluindo o das atividades estruturadas.

Para aprovação na disciplina, o aluno deverá:

1. Atingir resultado igual ou superior a seis, calculado a partir da média aritmética entre os graus das avaliações, considerando-se apenas as duas maiores notas obtidas dentre as três etapas de avaliação (AV1, AV2 e AV3) - a média aritmética obtida será o grau final do aluno na disciplina;
2. Obter grau igual ou superior a quatro em, pelo menos, duas das três avaliações;
3. Frequentar, no mínimo, 75% das aulas ministradas.

As disciplinas oferecidas na modalidade de Educação a Distância (EaD) seguirão o mesmo critério de avaliação das disciplinas presenciais.

Para a avaliação do Trabalho de Conclusão de Curso (TCC), ou trabalhos de mesma natureza, será atribuído grau único para a disciplina que, para aprovação do aluno, deverá ser igual ou maior do que seis.

Mais detalhes: Portaria D.E. n.º 02, de 18 de novembro de 2009.

Bibliografia Básica

CASATI, João Paulo. **Modelagem de sistemas[BV:RE]**. 1. ed. Rio de Janeiro: SESES, 2016.
Disponível em: <http://repositorio.savaestacio.com.br/site/index.html#/objeto/detalhes/126ADA15-D1D6-414E-BBF4-30A348881E90>

LUIZ ANTONIO LEÃO LISBOA JUNIOR. **PADRÕES DE PROJETO DE SOFTWARE[BV:RE]**. 1. ed.. Rio de Janeiro: SESES, 2018.
Disponível em: <http://repositorio.savaestacio.com.br/site/index.html#/objeto/detalhes/F35AEE02-F401-DE4B-9F91-4BB1245FD3E6>

MEDEIROS, Ernani S. **Desenvolvendo Software com UML 2.0: definitivo[BV:PE]**. 1. ed. São Paulo: Pearson, 2012.
Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/2921>

Bibliografia Complementar

FOWLER, Martin. **UML Essencial - Um Breve Guia Para a Linguagem-Padrão[BV:MB]**. 3. ed. Porto Alegre: Artmed, 2005.
Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788560031382/cfi/6/2!/4/2@0:0.00>

GAMMA, ERICH ; HELM, Richard ; JOHSON, Ralph ; VILSSIDES, John. **Padrões de Projetos: Soluções Reutilizáveis de Software Orientado a Objetos[BV:MB]**. 1. ed.. Porto Alegre: Bookman, 2011.
Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577800469>

GAMMA, ERICH ; HELM, Richard ; JOHSON, Ralph ; VILSSIDES, John. **Padrões de Projetos: Soluções Reutilizáveis de Software Orientado a Objetos[BV:MB]**. 1. ed.. Porto Alegre: Bookman, 2011.
Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577800469>

LARMAN, Craig. **Utilizando UML e Padrões [BV:MB]**. 3ed. Porto Alegre: Bookman, 2007.
Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577800476/cfi/0!/4/4@0:0.00:2.37>

Lee, Richard C.; Tepfenhart, William M. **UML e C++: guia prático de desenvolvimento orientado a objeto[BV:PE]**. 1. ed.. São Paulo: Pearson, 2013.
Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/40>

PAGE-JONES, Meilir. **Fundamentos do desenho orientado a objeto com UML [BV:PE]**. 1. ed. São Paulo: Pearson, 2001.
Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/33>

Outras Informações