

EMENTA

Análise, projeto e modelagem de projetos orientado a objetos. Análise de padrões de projeto, gerenciamento de configuração, incluindo gerenciamento de versões e release, qualidade de processo com seus modelos e engenharia de software orientada a serviços.

COMPETÊNCIAS**I - ANALISAR E RESOLVER PROBLEMAS**

IX. Pensamento lógico - Pensar e usar a lógica formal estabelecendo relações, comparações e distinções em diferentes situações.

X. Representação espacial e modelagem - Representar graficamente desenhos manuais e modelos, através das técnicas apropriadas.

XI. Espírito de pesquisa - Pesquisar e realizar experimentos com rigor científico para solucionar problemas, buscando a inovação.

XII. Ética e responsabilidade socioambiental - Atuar com ética e em conformidade com os aspectos socioambientais, com ações pautadas na sustentabilidade e na adoção de tecnologias limpas.

XIV Domínio da tecnologia da informação e comunicação - Utilizar sistemas informatizados requeridos para a operacionalização da profissão.

XV. Visão estratégica - Planejar ações a curto, médio e longo prazo para atingir metas, antecipando tendências e novas oportunidades.

XXI – Usabilidade para internet - Analisar, projetar e avaliar a usabilidade de sistemas para Internet. (Sistemas para Internet)

XX - Gestão de projetos em TI - Aplicar conceitos, métodos, técnicas e ferramentas de gerenciamento de projetos em sua área de atuação. (CST em Gestão da Tecnologia da Informação)

XXI - Tecnologia da informação - Identificar oportunidades de mudanças e projetar soluções usando tecnologias da informação nas organizações. (CST em Gestão da Tecnologia da Informação)

XXII - Programação - Desenvolver aplicativos em mais de uma linguagem que estejam ou não ligados a banco de dados. (CST em Gestão da Tecnologia da Informação)

XVII - Criação e desenvolvimento de sistemas computacionais - Especificar, projetar, desenvolver, implementar, documentar, manter e avaliar sistemas de computação, empregando metodologias, tecnologias e ferramentas de engenharia de software, linguagens de programação e banco de dados. (Análise E Desenvolvimento De Sistemas, Ciência da Computação)

XVIII – Qualidade de software - Avaliar, selecionar e utilizar ferramentas, metodologias e tecnologias adequadas ao problema e ao contexto para a produção de sistemas computacionais. (Análise E Desenvolvimento De Sistemas)

XIX - Requisitos de software e Projeto de interface - Especificar e gerenciar requisitos de software, gerenciar configurações de projeto de software e o projeto de interfaces. (Análise E Desenvolvimento De Sistemas)

XX - Manutenção de sistemas - Realizar rotinas demandadas e programadas de verificação nos sistemas de informação implementados. (Análise E Desenvolvimento De Sistemas)

XIX - Solução de problemas computacionais - Solucionar problemas computacionais a partir de preceitos matemáticos e de ferramentas de desenvolvimento. (Ciência da Computação)

XXII – Gestão de Projetos - Gerenciar projetos de desenvolvimento de sistemas computacionais. (Ciência da Computação)

XVII - Sistemas de apoio a decisão - Identificar e projetar soluções de alto nível e opções de fornecimento de serviços, realizando estudos de viabilidade com múltiplos critérios e decisão. (Sistemas de Informação)

XX - Interação Humano Computador - Aplicar os princípios de interação humano computador para avaliar e construir produtos incluindo interface do usuário, páginas web, sistemas multimídia e sistemas móveis. (Sistemas de Informação)

OBJETIVOS DE APRENDIZAGEM

- Aplicar e desenvolver o RUP em conjunto com a UML
- Elaborar e desenvolver a documentação de um projeto orientado a objetos com UML.
- Aplicar os principais padrões de projeto, aprofundando a compreensão sobre análise e projeto orientado a objetos.
- Aplicar os conceitos básicos de boas práticas de Desenvolvimento de Software
- Aplicar e Desenvolver os conceitos básicos de Testes Ágeis de Software
- Aplicar os princípios básicos da Engenharia de Software Orientada a Serviços.
- Elaborar os diagramas da UML para representar o problema. Refinar os diagramas incluindo particularidades tecnológicas para detalhar a documentação do sistema a ser codificado;
- Aplicar os conhecimentos acerca de Gerenciamento de Configuração de Software e seus desdobramentos.
- Elaborar, planejar e desenvolver documentação de Testes de Software (plano de testes, cenário de testes e casos de teste)
- Construir diagramas que representem a arquitetura de implantação e implementação dos componentes do software.

ATIVIDADE PRÁTICA SUPERVISIONADA

5. Analisar princípios e boas práticas no desenvolvimento de software e quais técnicas são aplicáveis

CRONOGRAMA DE AULAS

1. PROJETO DE SOFTWARE ORIENTADO A OBJETOS COM UML RUP (Processo Unificado – Rational) e UML Estudo de Viabilidade e Análise de Requisitos Diagrama de Caso de Uso	Objetivos de Aprendizagem		Competências Relacionadas
	<ol style="list-style-type: none"> 1. Elaborar e desenvolver a documentação de um projeto orientado a objetos utilizando o RUP com a UML e seu conjunto de artefatos 2. Análise de Viabilidade de requisitos de projetos 3. Revisar e reforçar os conceitos de requisitos de software 4. Documentar requisitos de software 5. Apresentar um modelo de documentação de Software baseado no documento de Visão do RUP 6. Utilizar casos de uso e o diagrama de caso de uso como ferramenta de projeto 		IX XIII XIV XVI XVIII XXI
	Estratégias de Ensino Aula expositiva dialogada / Apresentação e Desenvolvimento do Projeto da Disciplina Sequencia sugerida: <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica • Aula expositiva dialogada: Apresentação dos conceitos da modelagem orientada a objetos com RUP e UML. • Template de documento de visão do RUP • Análise e Viabilidade de Requisitos • Requisitos de Software • Demonstração do uso dos diagramas de caso de uso. • Atividade prática: Alunos formam grupos para planejar o trabalho da disciplina. • Alunos realizam atividades sobre Requisitos, Caso de Uso e Diagrama de Caso de Uso • Atividade de fixação individual extraclasse 	Avaliação Formativa Alunos trabalham em grupo planejando o projeto da disciplina; Alunos realizam atividades sobre requisitos e caso de uso baseado em estudo de caso.	Recursos LARMAN, Craig. Utilizando UML e Padrões. Bookman, 08/2011. FOWLER, Martin. UML Essencial. Bookman, 08/2011. Booch G. UML. Guia do Usuário Campus Elsevier - 2006 Medeiros E. Desenvolvendo Software Com Uml 2.0 - Definitivo - 2004 Bezerra, E. Princípio de Análise e Projetos de Sistemas com Uml 3ª Ed. Campus 2015
2. PROJETO DE SOFTWARE ORIENTADO A OBJETOS COM UML - Casos de Uso, diagrama de caso de uso Matriz de Rastreabilidade de Requisito Descrição detalhada de Caso de Uso Modelo de Caso de Uso	Objetivos de Aprendizagem		Competências Relacionadas
	<ol style="list-style-type: none"> 1. Apresentar o Modelo de Caso de Uso. 2. Identificar os atores, casos de uso e associações (generalização/especialização, dependências: include e extends e comunicação) 3. Elaborar um exemplo de matriz de rastreabilidade de requisitos 4. Descrever um caso de uso detalhadamente (caso de uso textual) com fluxo normal, alternativo e de exceção. 		IX XIII XIV XVI XVIII XXI
	Estratégias de Ensino Aula expositiva dialogada alternada com a prática / Desenvolvimento do Projeto da Disciplina Sequência sugerida: <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica. • Aula expositiva dialogada, alternada com a prática: Apresentação do projeto arquitetural com UML. • Modelos de caso de uso, identificação dos atores e casos de uso • Representação das associações do diagrama de caso de uso, associações ator-ator, ator-caso de uso e caso de uso – caso de uso 	Avaliação Formativa Alunos modelam diagrama de caso de Uso utilizando as ferramentas de UML	Recursos LARMAN, Craig. Utilizando UML e Padrões. Bookman, 08/2011. FOWLER, Martin. UML Essencial. Bookman, 08/2011. Booch G. UML. Guia do Usuário Campus Elsevier - 2006 Medeiros E. Desenvolvendo Software Com Uml 2.0 - Definitivo - 2004 Bezerra, E. Princípio de Análise e Projetos de Sistemas com Uml 3ª Ed. Campus 2015

3. PROJETO DE SOFTWARE ORIENTADO A OBJETOS COM UML Diagrama de Atividades Classes e Objetos Diagrama de Classes Conceitual	<ul style="list-style-type: none"> Elaborar um exemplo de matriz de rastreabilidade de requisitos (RFxUC ou RFxRNF) Detalhamento de Caso de Uso (Fluxo Normal, Alternativo e de Exceção) Apresentação de Template para detalhamento de caso de uso Atividade prática: Documentar os artefatos vistos na aula. Aplicação de QUIZ Atividade de fixação individual extraclasse 		
	Objetivos de Aprendizagem		Competências Relacionadas
	<ol style="list-style-type: none"> 1. Identificar as classes num projeto 2. Elaborar os diagramas de atividades 3. Elaborar o diagrama de classes. 4. Analisar os principais conceitos de análise Orientada a Objetos 		IX XIII XIV XVI XVIII XXI
	Estratégias de Ensino Aula prática em laboratório de informática / exposição de slides com o assunto da aula / Desenvolvimento do Projeto da disciplina/ Socrative ou Kahoot. Sequência sugerida: <ul style="list-style-type: none"> Apresentação dos objetivos de aprendizagem Levantamento dos conhecimentos prévios: pesquisa diagnóstica, breve revisão da aula anterior Aula expositiva dialogada alternada com atividades práticas na ferramenta de UML: Diagrama de Atividades, conceito de classes, objetos, orientação a objetos e diagrama de classes conceitual. Tipos de associações e multiplicidade entre classes Demonstração da criação de classes com objetos identificados com o método apresentado. Atividade prática Aplicação de QUIZ Atividade de fixação individual extraclasse 	Avaliação Formativa Os alunos formam duplas para elaborar os diagramas de atividades, identificar as classes e elaborar os diagramas de classes. Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.	Recursos LARMAN, Craig. Utilizando UML e Padrões. Bookman, 08/2011. FOWLER, Martin. UML Essencial. Bookman, 08/2011. Booch G. UML. Guia do Usuário Campus Elsevier - 2006 Medeiros E. Desenvolvendo Software Com Uml 2.0 - Definitivo - 2004 Bezerra, E. Princípio de Análise e Projetos de Sistemas com Uml 3ª Ed. Campus 2015
4. PROJETO DE SOFTWARE ORIENTADO A OBJETOS COM UML - Modelos estruturais de projeto: diagrama de classes. Classes de Projeto Arquitetura em Camadas Camadas BCE (Boundary, Control, Entity)	Objetivos de Aprendizagem		Competências Relacionadas
	<ol style="list-style-type: none"> 1. Apontar a importância do diagrama de classes no contexto do projeto orientado a objetos. 2. Elaborar diagramas de classe como parte de um projeto de software orientado a objetos. 3. Identificar padrões da arquitetura em camadas do modelo BCE (Boundary, Control e Entity) 4. Elaborar modelos de classes de projetos baseando-se na arquitetura de implementação do sistema. 		IX XIII XIV XVI XVIII XXI
	Estratégias de Ensino Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot. Sequência sugerida: <ul style="list-style-type: none"> Apresentação dos objetivos de aprendizagem Levantamento dos conhecimentos prévios: pesquisa diagnóstica, revisão da aula anterior. Aula expositiva dialogada, alternada com atividades práticas (desenvolvimento do projeto do Diagrama): Apresentação do 	Avaliação Formativa Os alunos devem se agrupar em, no máximo, grupo para apresentar o diagrama de classes para um software escolhido pelo professor. Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.	Recursos LARMAN, Craig. Utilizando UML e Padrões. Bookman, 2011. FOWLER, Martin. UML Essencial. Bookman, 2011. Booch G. UML. Guia do Usuário Campus Elsevier - 2006 Medeiros E. Desenvolvendo Software Com Uml 2.0 - Definitivo - 2004

	<p>diagrama de classes de projeto. Arquitetura BCE (Boundary, Control, Entity)</p> <ul style="list-style-type: none">• Demonstração da aplicação do diagrama de classes em um estudo de caso exposto pelo professor.• Atividade prática• Correção da atividade prática• Aplicação de QUIZ• Atividade de fixação individual extraclasse	Bezerra, E. Princípio de Análise e Projetos de Sistemas com Uml – Capítulo 8, 3ª Ed. Campus 2015
5. PROJETO DE SOFTWARE ORIENTADO A OBJETOS COM UML - diagrama de sequência. - Classes de Projeto e diagrama de Sequência.	Objetivos de Aprendizagem	
	1. Utilizar os diagramas de sequência, estados e atividades em um projeto orientado a objetos com UML. 2. Elaborar diagramas de sequência, estados e atividade em um projeto de software. 3. Realizar a especificação de interfaces de software em um projeto orientado a objetos. 4. Utilizar UML para especificar interfaces de software em um projeto orientado a objetos	
	IX XIII XIV XVI XVIII XXI	
5. PROJETO DE SOFTWARE ORIENTADO A OBJETOS COM UML - diagrama de sequência. - Classes de Projeto e diagrama de Sequência.	Estratégias de Ensino Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot. Sequência sugerida: <ul style="list-style-type: none">• Apresentação dos objetivos de aprendizagem• Levantamento dos conhecimentos prévios: pesquisa diagnóstica• Aula expositiva dialogada: Apresentação dos diagramas de sequência, diagramas de estados, diagramas de atividades e especificação.• Diagrama de Sequência: Troca de mensagens, linha de vida, objetos e passagem de parâmetros.• Demonstração da aplicação dos diagramas vistos e da especificação.• Atividade prática• Correção da atividade prática• Aplicação de QUIZ• Atividade de fixação individual extraclasse	Avaliação Formativa Os alunos devem se agrupar em, no máximo, grupo para apresentar os diagramas de sequência, estado e atividades para um software escolhido pelo professor. Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.
	Recursos SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 7. Pearson Addison Wesley. 2011.	
6. PADRÕES DE PROJETO - Elementos essenciais dos padrões de projeto. Padrões e Arquiteturas de Projetos.	Objetivos de Aprendizagem	
	1. Conceituar padrões de projeto. 2. Identificar os elementos essenciais dos padrões de projetos. 3. Descrever a utilização de padrões de projetos em problemas de projeto de software.	
6. PADRÕES DE PROJETO - Elementos essenciais dos padrões de projeto. Padrões e Arquiteturas de Projetos.	Competências Relacionadas	
	I VIII IX XI XIII XXI	

7. PROJETO DE SOFTWARE ORIENTADO A OBJETOS COM UML Componentes, Arquitetura e Implantação Diagrama de Componentes Diagrama de Arquitetura e Implantação	<p>Estratégias de Ensino</p> <p>Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Aula expositiva dialogada: Apresentação dos principais padrões de projeto de software (GOF, padrões de criação, padrões estruturais, padrões comportamentais) • Demonstração do uso dos padrões de projeto em exemplos práticos de softwares. • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	<p>Avaliação Formativa</p> <p>Os alunos devem se agrupar para apresentar uma classificação dos padrões de projeto existentes e demonstrar exemplos.</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>Recursos</p> <p>SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 7. Pearson Addison Wesley. 2011.</p> <p>PRESSMAN, R. , MAXIM, B. Engenharia de Software, Capítulo 16, 8th edição. AMGH, 01/2016</p>
	<p>Objetivos de Aprendizagem</p> <ol style="list-style-type: none"> 1. Aprender todos os componentes de software e representá-los no diagrama. 2. Analisar a arquitetura e implantação dos sistemas segundo a linguagem de modelagem Unificada (UML) 3. Elaborar diagramas de arquitetura e implantação. 		<p>Competências Relacionadas</p> <p>IX XIII XIV XVI XVIII XXI</p>
	<p>Estratégias de Ensino</p> <p>Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica, revisão da aula anterior. • Aula expositiva dialogada, alternada com atividades práticas (conclusão do projeto da disciplina): Diagrama de componentes e seus símbolos (interface fornecida e interface requerida) • Demonstração da aplicação do diagrama de componentes e o diagrama de arquitetura e implantação num estudo de caso exposto pelo professor. • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	<p>Avaliação Formativa</p> <p>Os alunos devem se agrupar em, no máximo, grupo para utilizar uma ferramenta de gerenciamento de versões.</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>Recursos</p> <p>LARMAN, Craig. Utilizando UML e Padrões. Bookman, 08/2011.</p> <p>FOWLER, Martin. UML Essencial. Bookman, 08/2011.</p> <p>Booch G. UML. Guia do Usuário Campus Elsevier - 2006</p> <p>Medeiros E. Desenvolvendo Software Com Uml 2.0 - Definitivo - 2004</p> <p>Bezerra, E. Princípio de Análise e Projetos de Sistemas com Uml 3ª Ed. Campus 2015</p>
	<p>Objetivos de Aprendizagem</p> <ol style="list-style-type: none"> 1. Aprender todos os componentes de software e representá-los no diagrama. 2. Analisar a arquitetura e implantação dos sistemas segundo a linguagem de modelagem Unificada (UML) 3. Elaborar diagramas de arquitetura e implantação. 		<p>Competências Relacionadas</p> <p>IX XIII XIV XVI XVIII XXI</p>
8. PROJETO DE SOFTWARE ORIENTADO A OBJETOS COM UML Componentes, Arquitetura e	<p>Objetivos de Aprendizagem</p> <ol style="list-style-type: none"> 1. Aprender todos os componentes de software e representá-los no diagrama. 2. Analisar a arquitetura e implantação dos sistemas segundo a linguagem de modelagem Unificada (UML) 3. Elaborar diagramas de arquitetura e implantação. 		<p>Competências Relacionadas</p> <p>IX XIII XIV XVI XVIII XXI</p>

<p>9. GERENCIAMENTO DE CONFIGURAÇÃO</p> <p>- Gerenciamento de versões.</p> <p>Gerência e Configuração</p> <p>- Integração de sistemas.</p> <p>Rastreamento de Problemas</p>	<p>Estratégias de Ensino</p> <p>Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica, revisão da aula anterior. • Aula expositiva dialogada, alternada com atividades práticas (conclusão do projeto da disciplina): Diagrama de componentes e seus símbolos (interface fornecida e interface requerida) • Demonstração da aplicação do diagrama de componentes e o diagrama de arquitetura e implantação num estudo de caso exposto pelo professor. • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	<p>Avaliação Formativa</p> <p>Os alunos deverão se agrupar para utilizar uma ferramenta de gerenciamento de versões e configurar os arquivos e códigos do projeto da disciplina num repositório de arquivos.</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>Recursos</p> <p>PRESSMAN, R. , MAXIM, B. Engenharia de Software, Capítulo 29, 8th edição. AMGH, 01/2016</p> <p>SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 25. Pearson Addison Wesley. 2011.</p>
	<p>Objetivos de Aprendizagem</p> <ol style="list-style-type: none"> 1. Descrever os conceitos utilizados em gerenciamento de versões. 2. Utilizar um sistema de gerenciamento de versões. 3. Aplicar o gerenciamento de versões em um projeto de software. 		<p>Competências Relacionadas</p> <p>I VIII XVI XVIII XXI</p>
	<p>Estratégias de Ensino</p> <p>Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo / Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Aula expositiva dialogada: Apresentação do funcionamento do gerenciamento de versões. • Demonstração do uso de gerenciamento de versões com ferramentas online como github ou bitbucket. • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	<p>Avaliação Formativa</p> <p>Os alunos deverão se agrupar para utilizar uma ferramenta de gerenciamento de versões e configurar os arquivos e códigos do projeto da disciplina num repositório de arquivos.</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>Recursos</p> <p>SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 24. Pearson Addison Wesley. 2011.</p> <p>PRESSMAN, R. MAXIM, B. Engenharia de Software, Capítulo 19, 20 e 21, 8th edição. AMGH, 01/2016</p>
	<p>Objetivos de Aprendizagem</p> <ol style="list-style-type: none"> 1. Descrever os conceitos utilizados em gerenciamento de versões. 2. Utilizar um sistema de gerenciamento de versões. 3. Aplicar o gerenciamento de versões em um projeto de software. 		<p>Competências Relacionadas</p> <p>I VIII XVI XVIII XXI</p>
<p>10. GERENCIAMENTO DE CONFIGURAÇÃO</p> <p>- Gerenciamento de versões.</p> <p>Gerência e Configuração</p> <p>- Integração de sistemas.</p>	<p>Objetivos de Aprendizagem</p> <ol style="list-style-type: none"> 1. Descrever os conceitos utilizados em gerenciamento de versões. 2. Utilizar um sistema de gerenciamento de versões. 3. Aplicar o gerenciamento de versões em um projeto de software. 		<p>Competências Relacionadas</p> <p>I VIII XVI XVIII XXI</p>

11. TESTES DE SOFTWARE Ferramentas de Testes Plano de Testes Cenários de Testes Casos de Testes	Estratégias de Ensino Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo / Socrative ou Kahoot. Sequência sugerida: <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Aula expositiva dialogada: Apresentação do funcionamento do gerenciamento de versões. • Demonstração do uso de gerenciamento de versões com ferramentas online como github ou bitbucket. • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	Avaliação Formativa Os alunos deverão se agrupar para utilizar uma ferramenta de gerenciamento de versões e configurar os arquivos e códigos do projeto da disciplina num repositório de arquivos. Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.	Recursos SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 26. Pearson Addison Wesley. 2011. PRESSMAN, R. MAXIM, B. Engenharia de Software, Capítulo 19, 20 e 21, 8th edição. AMGH, 01/2016
	Objetivos de Aprendizagem 1. Compreender a importância do Testes como atributo de qualidade no desenvolvimento de Software 2. Elaborar um plano de testes de Software 3. Elabora e planejar cenários de testes com seus respectivos casos de teste		Competências Relacionadas VIII XVI XVIII
	Estratégias de Ensino Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot. Sequência sugerida: <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica • Aula expositiva dialogada alternada com atividades práticas em laboratório: os tipos de testes, ferramentas de Testes automatizados. Plano de Testes, Cenários de Testes e Casos de Testes. Elaboração de um Plano de Teste com cenários e casos de teste. • Demonstração de um documento completo de Testes de Software • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	Avaliação Formativa Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.	Recursos SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 26. Pearson Addison Wesley. 2011.
	Objetivos de Aprendizagem 1. Conhecer as principais ferramentas de Testes automatizados 2. Realizar Testes Automatizados de Software 3. Conhecer as práticas ágeis de testes: TDD, BDD e ATDD 4. Planejar cenários de testes ágeis aplicando a semântica do BDD 5. Aplicar os testes e as práticas ágeis do TDD, BDD e ATDD		Competências Relacionadas VIII XVI XVIII
12 TESTES E PRÁTICAS ÁGEIS ATDD, STDD,DDD,FDD, TDD e BDD			

13. DESENVOLVIMENTO ÁGIL DE SOFTWARE - AGILE HÍBRIDO - eXtreme Programming (XP). SCRUM	Estratégias de Ensino	Avaliação Formativa	Recursos
	<p>Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica • Aula expositiva dialogada alternada com prática em laboratório: Apresentação das ferramentas de teste automatizado; • Práticas Ágeis: TDD, BDD, TDD e ATDD. • Escrita de Histórias de Testes – STDD • Escrita de Testes utilizando a semântica do BDD • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extra classe. 	<p>Atividade prática aplicando os conceitos e Semântica dos testes e práticas ágeis</p> <p>Exercícios e Atividades de fixação.</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 26. Pearson Addison Wesley. 2011.</p> <p>http://tdd.caelum.com.br/</p> <p>https://www.linkedin.com/pulse/agile-development-difference-between-tddatddbdd-komal-sureka/</p> <p>https://agilecoachninja.wordpress.com/2016/05/02/tdd-stdd-ddd-fdd-bdd-atdd/</p>
	Objetivos de Aprendizagem	Competências Relacionadas	
	<ol style="list-style-type: none"> 1. Aplicar os conceitos básicos sobre Desenvolvimento ágil de software. 2. Analisar a transição e mudança de cultura entre modelos tradicionais e modelos ágeis 3. Descrever os conceitos envolvidos no SCRUM, XP e Agile híbrido (misto). 4. Aplicar os conceitos de Agile Híbrido como modelo de transição de cultura 5. Aplicar o XP num projeto de desenvolvimento de software 6. Identificar o escopo de aplicação de XP em equipes de desenvolvimento de software. 7. Aplicar XP em um projeto de desenvolvimento de software. 	<p>I</p> <p>VIII</p> <p>IX</p> <p>XVI</p> <p>XVIII</p> <p>XXI</p>	
14. DESENVOLVIMENTO ÁGIL DE SOFTWARE - Scrum. - Scrum em larga escala	Estratégias de Ensino	Avaliação Formativa	Recursos
	<p>Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica • Aula expositiva dialogada alternada com Atividades práticas em laboratório: Apresentação dos principais modelos AGILE Híbrido. Apresentação do desenvolvimento ágil com SCRUM e XP. • Demonstração da aplicação do modelo híbrido em projetos. • Demonstração da aplicação de XP e seus artefatos em um projeto de software. • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	<p>Os alunos devem se agrupar em grupo para escrever um relatório sobre modelos híbridos e XP.</p> <p>Produção dos artefatos de software propostos em cada modelo.</p> <p>Propor desenvolvimento e/ou simulação de um projeto de desenvolvimento que utilize o AGILE HÍBRIDO (Scrum misto)</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>Gomes, A. F. Agile Desenvolvimento de software com entregas frequentes e foco no valor de negócio – 2013 - CASA DO CODIGO - SP</p> <p>Teles, M. V. Extreme Programming - 2014 - NOVATEC</p> <p>Beck K. Programação Extrema XP Explicada - BOOKMAN- 2004</p> <p>http://projectmanagementhistory.com/ExtremeProgramming(XP).html</p> <p>https://projetoseti.com.br/agil-x-tradicional-projetos-hibridos/</p> <p>https://www.martinfowler.com/bliki/ExtremeProgramming.html</p> <p>https://clarify.com.br/gestao-projetos/abordagem-hibrida-gestao-vantagens/</p> <p>https://www.desenvolvimentoagil.com.br/scrum</p> <p>http://www.desenvolvimentoagil.com.br/xp/</p>
	Objetivos de Aprendizagem	Competências Relacionadas	
	<ol style="list-style-type: none"> 1. Descrever como funciona Scrum no desenvolvimento de software. 2. Descrever características de escalabilidade no SCRUM 3. Identificar os principais papéis no Scrum. 4. Utilizar os artefatos de Scrum. 5. Analisar exemplos de projetos com SCRUM em larga escala 6. Aplicar Scrum no desenvolvimento de um software. 	<p>I</p> <p>VIII</p> <p>IX</p> <p>XVI</p> <p>XVIII</p> <p>XXI</p>	

15. SOA – ENGENHARIA DE SOFTWARE ORIENTADA A SERVIÇOS - Definição - Princípios Básicos	<p>Estratégias de Ensino</p> <p>Aula prática em laboratório de informática/ exposição de slides com o assunto da aula/ trabalho em grupo/ Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica e atividade de desenvolvimento de projeto • Aula expositiva dialogada alternada com prática: Apresentação do Scrum como processo de desenvolvimento de software. Escalabilidade do SCRUM em Projetos • Demonstração da aplicação do Scrum em um projeto de software. • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	<p>Avaliação Formativa</p> <p>Os alunos devem se agrupar em, no máximo, grupo para escrever um relatório sobre Scrum Escalável.</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>Recursos</p> <p>Guia do SCRUM em larga escala https://www.scrumatscale.com/scrum-at-scale-guide/</p> <p>Sutherland J. Scrum. A Arte de Fazer o Dobro do Trabalho na Metade do Tempo – LEYA</p> <p>GUIA DO SCRUM BR https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf</p> <p>CONHECIMENTO EM SCRUM (SBOK) https://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016-Portuguese.pdf</p>
	<p>Objetivos de Aprendizagem</p> <ol style="list-style-type: none"> 1. Aplicar os princípios básicos da Engenharia de Software Orientada a Serviços. 2. Descrever como um software pode ser desenvolvido usando SOA. 3. Utilizar SOA no desenvolvimento de um software. 		<p>Competências Relacionadas</p> <p>XVI XVIII XXI</p>
	<p>Estratégias de Ensino</p> <p>Aula prática em laboratório de informática / exposição de slides com o assunto da aula / trabalho em grupo/ Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica (forms) • Aula expositiva dialogada: Apresentação do funcionamento da arquitetura orientada a serviços. • Demonstração da aplicação de SOA em um projeto de software apresentado pelo professor. • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	<p>Avaliação Formativa</p> <p>Os alunos devem se agrupar em, no máximo, grupo para escrever um relatório sobre SOA.</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>Recursos</p> <p>SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 19. Pearson Addison Wesley. 2011.</p> <p>ERL, T. SOA - Princípio de Design de Serviços. Pearson Education do Brasil. 2009</p>
16. ENGENHARIA DE SOFTWARE ORIENTADA A SERVIÇOS - Desenvolvimento de software com serviços.	<p>Objetivos de Aprendizagem</p> <ol style="list-style-type: none"> 1. Aplicar os princípios básicos da Engenharia de Software Orientada a Serviços 2. Descrever como um software pode ser desenvolvido com engenharia de software orientada a serviços. 3. Aplicar engenharia de software orientada a serviços no desenvolvimento de um software. 		<p>Competências Relacionadas</p> <p>XVI XVIII XXI</p>

	Estratégias de Ensino	Avaliação Formativa	Recursos
	<p>Aula prática em laboratório de informática / exposição de slides com o assunto da aula / trabalho em grupo/ Socrative ou Kahoot.</p> <p>Sequência sugerida:</p> <ul style="list-style-type: none"> • Apresentação dos objetivos de aprendizagem • Levantamento dos conhecimentos prévios: pesquisa diagnóstica (forms) • Aula expositiva dialogada: Apresentação sobre desenvolvimento de software com serviços. • Demonstração da criação software com serviços. • Atividade prática • Correção da atividade prática • Aplicação de QUIZ • Atividade de fixação individual extraclasse 	<p>Os alunos devem se agrupar em, no máximo, grupo para escrever um relatório sobre um software desenvolvido com serviços.</p> <p>Questionário sobre o assunto no final da aula usando os últimos slides, Kahoot ou Socrative.</p>	<p>SOMMERVILLE, I. Engenharia de Software. 9a edição. Capítulo 19. Pearson Addison Wesley. 2011.</p> <p>ERL, T. SOA - Princípio de Design de Serviços. Pearson Education do Brasil. 2009</p>

17	
18	Essas unidades estão disponíveis para acomodar, quando presente, e não necessariamente nesta ordem:
19	- Aplicação de avaliações
20	- Revisão ou reforço de conteúdos mediante avaliação da performance da turma
21	- Feriados e eventos fortuitos
22	

AValiação (Práticas e Projetos)

A Média Final (MF) da disciplina considera os seguintes elementos e valores:

N1	N2
<p>A1 – Avaliação(ões) a ser(em) definida(s) de acordo com os objetivos de aprendizagem</p> <p>[nota de 0 a 10]</p>	<p>A2 – Avaliação(ões) a ser(em) definida(s) de acordo com os objetivos de aprendizagem</p> <p>[nota de 0 a 10]</p>

A Média Final (MF) é calculada por meio da média ponderada das duas notas, N1 e N2, com peso, respectivamente de 40% e 60%, resultante da seguinte equação:

$$MF = (N1 \cdot 0,4) + (N2 \cdot 0,6)$$

Para aprovação, a Média Final deverá ser igual ou superior a 6,0 (seis), além da necessária frequência mínima de 75% nas aulas.

Não haverá Avaliação Substitutiva.

BIBLIOGRAFIA BÁSICA

- LARMAN, Craig. Utilizando UML e Padrões. Bookman, 2011. Disponível em: Minha Biblioteca. Acesso em: 16 set. 2019.
- Bezerra, E. Princípio de Análise e Projetos de Sistemas com Uml, 3ª Ed. Campus 2015.
- SOMMERVILLE, I. Engenharia de Software. 9a edição. Pearson Addison Wesley. 2011. Disponível em: Biblioteca Virtual. Acesso em: 16 set. 2019.

BIBLIOGRAFIA COMPLEMENTAR

- ERL, T. SOA - Princípio de Design de Serviços. Pearson Education do Brasil. 2009. Disponível em: Biblioteca Virtual. Acesso em: 16 set. 2019.
- MEDEIROS, E. Desenvolvendo Software com UML 2.0 Definitivo. Pearson Makron Books. 2004. Disponível em: Biblioteca Virtual. Acesso em: 16 set. 2019.
- LEE, R. C. TEPFENHART, W. M. UML e C++ - Guia Prática de Desenvolvimento Orientado a Objeto. Makron Books. 2001. Disponível em: Biblioteca Virtual. Acesso em: 16 set. 2019.
- GALLOTTI, G. M. A. Qualidade de Software. Bibliografia Universitária Pearson. Pearson. 2017. Disponível em: Biblioteca Virtual. Acesso em: 16 set. 2019.
- PAGE-JONES, M. Fundamentos do Desenho Orientado a Objeto com UML. Pearson Makron Books. 2000. Disponível em: Biblioteca Virtual. Acesso em: 16 set. 2019.

<i>Desenvolvido por</i>	<i>Othon Marcelo Nunes Batista</i>
<i>Data</i>	<i>02 de Outubro de 2017</i>
<i>1ª Atualização</i>	<i>Elaine Barbosa de Figueiredo 20/06/2018</i>
<i>2ª Atualização</i>	<i>Elaine Barbosa de Figueiredo 25/11/2019</i>