



Plano de Atividades Domiciliares ADE

Unidade Curricular: Engenharia de Software

Professor(es):
Luiz Eduardo Galvão Martins

Contato: *legmartins@unifesp.br*

Ano Letivo: 2021

Semestre: 2º

Carga horária total: 36P + 36T

Turmas: Integral

Plataforma de acesso ao curso: *Moodle* UNIFESP

Objetivos (remoto): Introduzir os principais aspectos da Engenharia de Software. Capacitar os alunos nas principais práticas da Engenharia de Software, envolvendo as atividades de engenharia de requisitos, projeto e modelagem de software, desenvolvimento, teste, verificação e validação de software.

Conteúdo Programático e Cronograma

Semana 1

Atividades síncronas (1 hora): Apresentação da disciplina e critérios de avaliação. Introdução à Engenharia de Software (importância e motivação).

Atividades assíncronas (3 horas): Leitura e análise de artigos sobre Engenharia de Software.

Semana 2

Atividades síncronas (1 hora): Modelos de processo de software. Ciclo de vida de desenvolvimento de software. Paradigmas de desenvolvimento de software

Atividades assíncronas (3 horas): Leitura e estudo sobre modelos de processo de software e paradigmas de desenvolvimento.

Semana 3

Atividades síncronas (1 hora): Introdução à engenharia de requisitos. Elicitação, análise, modelagem, especificação e validação de requisitos de software.

Atividades assíncronas (3 horas): Leitura, análise de vídeos, e estudo sobre engenharia de requisitos.

Semana 4

Atividades síncronas (1 hora): Modelos arquiteturais de software. Introdução à UML. Diagrama de casos de uso.

Atividades assíncronas (3 horas): Leitura, análise de vídeos, e estudo sobre UML. Início do desenvolvimento do projeto – especificação de requisitos.



Semana 5

Atividades síncronas (1 hora): Modelagem de software usando UML: diagrama de atividades, diagrama de classes.

Atividades assíncronas (3 horas): Leitura, análise de vídeos, e estudo sobre UML. Desenvolvimento do projeto – especificação de requisitos.

Semana 6

Atividades síncronas (1 hora): Modelagem de software usando UML: diagrama de estados, diagrama de sequência.

Atividades assíncronas (3 horas): Leitura, análise de vídeos, e estudo sobre UML. Desenvolvimento do projeto – especificação da arquitetura do software. Modelagem do software.

Semana 7

Atividades síncronas (1 hora): Boas práticas de codificação.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – modelagem do software e codificação.

Semana 8

Atividades síncronas (1 hora): Teste de software. Verificação e validação de software.

Atividades assíncronas (3 horas): Leitura, análise de vídeos, e estudo sobre teste de software. Desenvolvimento do projeto – modelagem do software e codificação.

Semana 9

Atividades síncronas (1 hora): Manutenção de software. Tipos de manutenção de software.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – modelagem do software, codificação e testes.

Semana 10

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – modelagem do software, codificação e testes.

Semana 11

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – modelagem do software, codificação e testes.

Semana 12

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – codificação e testes.

Semana 13

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – codificação e testes. Entregável 1.

Semana 14

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – codificação e testes.



Semana 15

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – codificação e testes. Entregável 2.

Semana 16

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – codificação e testes.

Semana 17

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – codificação e testes. Entregável 3.

Semana 18

Atividades síncronas (1 hora): Inspeção dos artefatos de software produzidos pelos grupos.

Atividades assíncronas (3 horas): Desenvolvimento do projeto – codificação e testes. Entrega do projeto final. Lições aprendidas com o desenvolvimento do projeto.

Metodologia de Ensino Utilizada: Aulas expositivas online ao vivo, videoaulas gravadas, atividades assíncronas, desenvolvimento de um projeto. Inspeção dos projetos. Discussões sobre lições aprendidas.

Metodologia de Avaliação (estratégias para atingir conceitos “cumprido” ou “não cumprido”):

Os alunos entregarão um projeto, de forma modular, ao longo semestre. Este projeto envolverá:

- Elicitação e especificação dos requisitos de software
- Especificação da arquitetura de software
- Modelagem de software
- Codificação e testes de software

Bibliografia básica e complementar para uso remoto

- SOMMERVILLE, Ian. Engenharia de software, 10ª ed. Editora Pearson 768 ISBN 9788543024974. *Acesso ao ebook pela coleção Pearson.*
- MEDEIROS, Ernani. Desenvolvimento de Software com UML 2.0 Definitivo. Editora Pearson 288 ISBN 9788534615297. *Acesso ao ebook pela coleção Pearson.*
- Artigos e tutoriais de acesso livre sobre engenharia de software (arquivos pdf e html).