

[Informações](#)[Ementa](#)[Bibliografia](#)[Alunos - Solicitações](#)[Ocupação](#)[Alunos - Matriculados](#)[Encontros](#)[Documentos](#)[Ficha 2](#)[Extensão](#)

## Ficha 2 - HENRI FREDERICO EBERSPACHER

### Programa

#### 1. Processo de Desenvolvimento de software:

- Modelos de processos de desenvolvimento de software (ciclo de vida);
- Atividades de Processos;
- Disciplinas de desenvolvimento de software;
- Engenharia de Software - uma abordagem em camadas;
- Um arcabouço de processos.

#### 2. Métodos e Ferramentas de Desenvolvimento de software:

- Métodos e ferramentas orientadas a funções e dados;
- Métodos e ferramentas orientadas a objetos.

#### 3. Processo unificado de desenvolvimento de software:

- Conceitos, Fases do ciclo de vida;
- Requisitos, análise e projeto, implementação, testes, manutenção de software;
- UML;
- Planejamento e execução de projetos utilizando o Processo Unificado.

#### 4. Métodos ágeis de desenvolvimento de software:

- Método SCRUM de Gerenciamento de projetos;
- Extreme Programming (XP).

#### 5. Prática de Engenharia de Software:

- Essência da prática; Práticas de comunicação, Planejamento e Modelagem, Construção e Implementação.

### Objetivo geral

Desenvolver as capacidades técnicas, organizativas e metodológicas para compreender e utilizar as ferramentas envolvidas no processo de desenvolvimento de softwares, levando em consideração as normas de qualidade e usabilidade.

### Objetivos específicos

- Compreender as atividades envolvidas no processo de desenvolvimento de software.

- Realizar análise e projeto de software aplicando metodologias de desenvolvimento tradicionais ou ágeis e padrões de desenvolvimento, normas técnicas e de qualidade.
- Especificar software usando modelagem UML.

## Procedimentos didáticos

A disciplina será desenvolvida mediante aulas expositivas e aplicação de metodologias ativas de aprendizagem, sobretudo PjBL (*Project-Based Learning*). As aulas expositivas-dialogadas serão empregadas para explicação dos conteúdos curriculares teóricos, da articulação teórico-prática e para o esclarecimento de dúvidas de conteúdo, de metodologia, de avaliação e para o feedback do processo avaliativo.

Atividades práticas serão propostas, envolvendo a discussão de situações problemas em cenários de desenvolvimento de software, bem como o desenvolvimento de um seminário temático com discussão dos temas em evidência para o contexto real de aplicação das competências da disciplina.

A plataforma Microsoft TEAMS, disponível gratuitamente para todos os estudantes com registro ativo na UFPR, será utilizada para todas as comunicações oficiais, agenda, datas importantes, disponibilização de material adicional: artigos, slides, pdf etc. e para interação/comunicação durante o semestre, incluindo a entrega dos trabalhos.

## Formas de avaliação

Serão realizadas:

- Atividades individuais de exercício, resolução e modelagem, correspondendo a 30% da avaliação;
- Atividade de seminário temático correspondendo a 30% da avaliação;
- Avaliação somativa individual, duas provas P1 e P2, correspondendo a 40% da avaliação (20% cada uma das avaliações).

A Média semestral será calculada mediante uma média ponderada das atividades realizadas conforme pesos apresentados acima.

As provas individuais (duas bimestrais) serão agendadas em acordo com a turma, com previsão para:

- 1a. quinzena de dezembro/2022 e

- 2a. quinzena de fevereiro/2023.

A aprovação segue o padrão da universidade, ou seja:

- Média  $\geq 7.0$ , aprovado sem exame;
- $4.0 \leq$  Média  $< 7.0$ , em exame final;
- Média  $< 4.0$ , reprovado sem direito a exame final.

O exame final será realizado mediante prova de recuperação de nota a ser realizada em 02/março/2023.

A frequência mínima para aprovação deve ser maior ou igual a 75% (a postagem das atividades propostas será computada na frequência do aluno).

## Bibliografia básica

1. SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011; reimpressão 2014. 529 p., il. inclui referencias ISBN 9788579361081.
2. PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. 7. ed. Porto Alegre RS: AMGH, 2011. 780 p., il. Inclui bibliografia e índice, ISBN 9788563308337 (broch.).
3. LARMAN, Craig. Utilizando UML e padrões: uma introdução a análise e ao projeto orientado a objetos e ao desenvolvimento iterativo. 3 ed. Porto Alegre: Bookman, 2007. 695 p., il. inclui bibliografia e índice, ISBN 9788560031528.

## Bibliografia complementar

1. COHN, Mike. Desenvolvimento de software com Scrum: aplicando métodos ágeis com sucesso. Porto Alegre: Bookman, 2011. 496 p., il. Inclui referencias e índice. ISBN 9788577808076 (broch.).
2. PETERS, James F; PEDRYCZ, Witold. Engenharia de software: teoria e prática. Rio de Janeiro: Campus, 2001. XVII, 602. Inclui bibliografia e índice. ISBN 8535207465 (broch.).
3. JINO, Mario; MALDONADO, José Carlos; DELAMARO, Márico Eduardo. Introdução ao teste de software. Rio de Janeiro: Elsevier: Campus, 2007. 394 p., il. (Série Campus SBC). Inclui bibliografia e índice. ISBN 9788535226348 (broch.).
4. MOLINARI, Leonardo. Inovação e automação de testes de software. São Paulo: Erica, 2010 (reimpressão 2014). 140 p., il. Inclui bibliografia e índice. ISBN 9788536502694 (broch.).
5. PAULA FILHO, Wilson de Pádua. Engenharia de Software: fundamentos, métodos e padrões. 2. ed. Rio de Janeiro: LTC, c2003. 602 p., il. Inclui índice, bibliografia, glossário e apêndices. ISBN 8521613393 (broch.).