



Universidade Estadual de Campinas Instituto de Computação

Prof. Dr. Bruno Barbieri de Pontes Cafeo
cafeo@ic.unicamp.br
<https://ic.unicamp.br/~cafeo/>



MC656 - Engenharia de Software (2024s1)

1 Objetivos da Disciplina

Ao final do curso, o aluno deve ser capaz de:

- Aprofundar a compreensão das necessidades inerentes à Engenharia de Software em larga escala, explorando métodos e práticas essenciais para o desenvolvimento de sistemas complexos.
- Refinar a capacidade de concepção e implementação de sistemas de software por meio da assimilação de um ciclo de vida abrangente, abordando integralmente desde a elicitação de requisitos até a fase de testes, mediante a aplicação de métodos, práticas e ferramentas especializadas.
- Desenvolver proficiência na adoção de metodologias avançadas, boas práticas e ferramentas contemporâneas para condução das atividades do ciclo de vida de um software, visando a entrega de soluções tecnológicas resilientes e eficazes.

2 Horário de Aulas

As aulas ocorrerão nos seguintes dias e horários:

Dia da Semana	Horário	Local	Tipo
Terça-feira	21h00 - 23h00	CB17	Teórica
Quinta-feira	19h00 - 21h00	CB17	Teórica

3 Atendimento

Atendimento presencial ou via Google Meet agendado com, pelo menos, 24h de antecedência pelo link do Google Calendar: <http://tinyurl.com/duvidas-mc646-2024s1>.

4 Programa da Disciplina

Os tópicos a serem apresentados no curso incluem:

- U_0 : Introdução à Engenharia de Software e Visão Sociotécnica.
- U_1 : Processos de Software.
- U_2 : Gerência de Configuração de Software.
- U_3 : Engenharia de Requisitos.
- U_4 : Análise e Projeto de Sistemas de Software.
- U_5 : Teste de Software e Liberação de Software.

5 Conhecimento esperado

- **Compreensão de Abstrações em Tipos Abstratos de Dados (TAD) e Objetos:**
 - Familiaridade com a capacidade de abstrair elementos do mundo real para construção de Tipos Abstratos de Dados.
 - Conhecimento sobre a utilização de objetos para modelar e representar entidades e conceitos no contexto do desenvolvimento de software.
- **Familiaridade com Programação Modular:**
 - Conhecimento em programação modular, utilizando linguagens de programação que oferecem suporte à definição de interfaces e componentes (módulos).
 - Habilidade para dividir o código em módulos independentes, promovendo uma estrutura organizada e reduzindo interdependências entre as diversas partes do sistema.
- **Conhecimento em Desenvolvimento Orientado a Objetos:**
 - Familiaridade com os princípios da programação orientada a objetos, incluindo encapsulamento, herança e polimorfismo, para desenvolvimento eficaz e reutilização de código.

6 Página do Curso

O material da disciplina ficará disponível na sala do Google Classroom. O código de inscrição será enviado pelo professor por e-mail, caso necessário.

7 Avaliações e Composição da Nota

A média da disciplina M será calculada da seguinte forma:

$$M = (0.8 * (A_1 + A_2 + A_3 + A_4 + A_5)) + (0.2 * PF)$$

onde A_x são as avaliações das unidades de ensino, e PF é a nota do Projeto Final. Todas as atividades e projeto final serão realizados em grupos de 3 ou 4 alunos.

Um aluno poderá fazer o exame SOMENTE se a média M estiver entre 2.5 (inclusive) e 4.9 (inclusive). O exame consiste em refazer as atividades avaliativas já corrigidas e que o aluno não obteve sucesso. Então, é sugerido que o aluno refaça apenas atividades em que a nota foi inferior a 5,0. Após entregar as reavaliações, a média será recalculada conforme a equação definida anteriormente e, se o aluno atingir pelo menos 5,0 na média final, este será aprovado e terá 5,0 como teto da nota final. Assim, não há possibilidade de um aluno de exame ter uma nota final maior que 5,0 quando aprovado.

8 Datas das Avaliações

A seguir estão as datas importantes para o curso:

- A_1 : 18/03/2024
- A_2 : 15/04/2024
- A_3 : 30/04/2024
- A_4 : 03/06/2024
- A_5 : 24/06/2024
- PF : 27/06/2024
- Exame Final: 11/07/2024

9 Referências

Algumas das referências consideradas importantes para o cumprimento do conteúdo proposto encontram-se listadas a seguir. O material complementar a ser utilizado será indicado na página da disciplina.

- Livro texto:
 - Marco Tulio Valente. *Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade*. Leanpub, 2020. Edição online gratuita: <https://engsoftmoderna.info/>
- Adicionais:
 - Sommerville, I. (2016). *Software Engineering, 10th edition*. Pearson.
 - Prikladnicki, Rafael, Renato Willi, and Fabiano Milani. *Métodos ágeis para desenvolvimento de software*. Bookman Editora, 2014.
 - Fowler, Martin. *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 2018.
 - Suryanarayana, Girish, Ganesh Samarthiyam, and Tushar Sharma. *Refactoring for software design smells: managing technical debt*. Morgan Kaufmann, 2014.
 - Delamaro, M., Jino, M., & Maldonado, J. (2017). *Introdução ao teste de software*. Elsevier Brasil.
 - Humble, Jez, and David Farley. *Continuous delivery: reliable software releases through build, test, and deployment automation*. Pearson Education, 2010.

Observações

- Não haverá provas ou trabalhos substitutivos que não constem nesse PDD.
- Qualquer tentativa de fraude nas atividades ou nos trabalhos/projetos implicará em nota final $M = 0$ (zero) para todos os envolvidos.
- As datas estipuladas para a entrega das avaliações encontram-se detalhadas no cronograma da disciplina (Slide 1). **O prazo limite para submissão é invariavelmente até as 23:59 do dia estipulado.**