

MO620 – Engenharia de Software II

MC976 – Tópicos em Engenharia de Software II

Horário: terças e quintas 16:00 – 18:00 --- 2S 2021

Professor: Cecília Mary Fischer Rubira
cmrubira@ic.unicamp.br -- IC-Unicamp

Disciplina Remota: metodologia de ensino a distância baseada em aulas gravadas, aulas online (síncronas), aulas de dúvidas online, entrega de exercícios para avaliação.

Ferramentas utilizadas: Plataforma Moodle, Google Meet para aulas síncronas, Diário de Bordo organizado por semanas.

Ementa:

- Conceitos da análise OO e projeto OO com foco na engenharia de modelagem de sistemas de software.
- Modelagem estática: construção de modelos de classes.
- Modelagem dinâmica: construção de modelos dinâmicos: modelos de sequência, de colaboração e de atividades.
- Conceitos Básicos: identificar objetos e classificá-los em classes, especificar atributos e operações das interfaces públicas (TAD e encapsulamento), herança e polimorfismo, e tratar relacionamentos de generalização/especialização, agregação e associação entre as classes.

Conceitos Avançados: herança múltipla, interfaces, pacotes, metaclasses, delegação.

- Ênfase em Padrões de projeto: model-view-controller (MVC), composite, observer, singleton, abstract factory, etc
- Conceitos básicos de arquitetura de software, definição de componentes, conectores e configurações arquiteturais.
- Projeto arquitetural e padrões arquiteturais (layer (camadas), centrado em dados, máquina virtual, etc)
- Visão geral de Atributos de qualidade (segurança, disponibilidade, etc) associados ao modelo arquitetural.
- Métodos de desenvolvimento de software centrados na arquitetura.
- Modelos para análise de requisitos de *safety* para sistemas computacionais, modelos probabilísticos de decisões.

Objetivo:

Ao final da disciplina o aluno deve ser capaz de criar modelos usando os conceitos do modelo de objetos e deve ser capaz de, partindo da especificação de um problema, criar modelos (estáticos e dinâmicos) que representem uma solução para o problema. Também o aluno deve ser capaz de definir um modelo inicial da arquitetura de software do sistema alvo, de acordo com os requisitos de qualidade priorizados durante o projeto arquitetural.

Critério de Avaliação: tarefas com entregas semanais e trabalho de pesquisa ao final do curso.

Cronograma Detalhado: a ser entregue.

Bibliografia:

I. Sommerville. Software Engineering, Addison-Wesley, 10th edition, 2015.

G. Booch, J. Rumbaugh and I. Jacobson. The Unified Modeling Language User Guide, second edition, Addison-Wesley, 2005.