#### **Construção de Software**

Carga horária

Teórica: 48 horas

Prática: 80 horas

Total: 128 horas

Unidade responsável: Instituto de Informática (INF)

Natureza: Núcleo Específico (NE) / Obrigatória

Pré-requisito: Não possui

Ementa:

1. Visão geral (4h): construção (minimizar complexidade, antecipação de mudança, verificação, padrões), projeto de software (software design), qualidade de produto.
2. Planejamento (8h): linguagens de programação e processos de construção.
3. Gerência de construção (16h): controle de versão, inspeção e revisão de código.
4. Fundamentos de codificação (32h): estratégias recomendadas para criar código, variáveis, classes, interfaces, polimorfismo, rotinas, recursão, condições, laços, tratamento de exceção, reflexão, programação defensiva, padrão de codificação (leiaute e estilo), documentação, ferramentas de programação.
5. Projeto (design) detalhado e codificação (32h): noções de projeto detalhado, especificação de projeto, análise sintática (*parsing*), expressões regulares, parametrização (*generics*), *closure*, *logging*, configuração de software em tempo de execução. Internacionalização. Técnicas de construção baseadas em estado e tabelas.
6. Refatoração (8h).
7. Testes de unidade (16h).
8. Detecção e remoção de defeitos (debugging) (8h).
9. Integração (4h): integração contínua.

#### *Condições mínimas (estar apto a):*

* (Segue instruções). Explicar a construção de software, a relação dessa área de conhecimento com as demais da engenharia de software e a relação com o conceito de sistema.
* (Faz com orientação). Usar e configurar ferramentas para edição, documentação, compilação, depuração (*debugging*), *build*, teste, controle de versão, coleta de medidas (desempenho, consumo de memória, análise estática e cobertura), integração contínua, e aplicativos fundamentais de linha de comandos.
* (Faz com orientação). Criar código orientado a objetos em conformidade com o projeto (*design*) detalhado seguindo estratégias recomendadas.
* (Segue instruções). Usar processos para a construção de software (criação de código, controle de versão, inspeção e integração).
* (Segue instruções). Detalhar projeto (*design*) em conformidade com requisitos de software.
* (Segue instruções). Depurar (*debugging*) código.
* (Segue instruções). Criar testes de unidade básicos empregando estratégias recomendadas.
* (Segue instruções). Localizar e reutilizar código (bibliotecas e *frameworks*).
* (Segue instruções). Colaborar com a construção de código em equipe.
* (Segue instruções). Avaliar a qualidade interna de código e, quando apropriado, aplicar estratégias recomendadas de refatoração.

Bibliografia básica:

* MCCONNELL, S. Code Complete: um guia prático para a construção de software 2.a edição. Porto Alegre, RS: Bookman, 2005. ISBN 8536305045.
* GOODLIFFE, P. Como ser um programador melhor. Novatec, 2015. ISBN 978-85-7522-415-1.
* ORAM, A.; WILSON, G. Beautiful code. O'Reilly, 2007. ISBN 9780596510046.

Bibliografia complementar:

* FOWLER, M. Refatoração: aperfeiçoando o projeto de código existente Porto Alegre: Bookman, 2004, ISBN 8536303956.
* IRVINE, K. R. C++ and object-oriented programming Upper Saddle River: Prentice-Hall, 1997. 526 p. Bibliografia e indice ISBN 0023598522 (broch.)
* BECK, K. Implementation patterns Upper Saddle River, NJ: Addison-Wesley, 2008. ISBN 0321413091.
* MARTIN, R. C. Clean code: a handbook of agile software craftsmanship. Prentice Hall, 2009. ISBN 0132350882.
* AGANS, D. J. Debugging the nine indispensable rules for finding even the most elusive software and hardware problems. AMACOM, 2002. ISBN 0814471684.

Informações: