#### **Software Concorrente e Distribuído**

Carga horária

Teórica: 32 horas

Prática: 32 horas

Total: 64 horas

Unidade responsável: Instituto de Informática (INF)

Natureza: Núcleo Específico (NE) / Obrigatória

Pré-requisito: Não possui

Ementa:

1. Qualidade de produto (4h): escalabilidade, tolerância a falhas, disponibilidade, desempenho.
2. Conceitos (4h): middleware, distribuição, paralelismo, concorrência, RESTful, web, internet das coisas, contêineres, troca de mensagens, micro-serviços, serviços web.
3. Métodos para construção de software distribuído e para construção de software concorrente (8h).
4. Concorrência (16h): paralelismo, aplicações IO-intensive e CPU-intensive, thread safe, memória compartilhada, Software Transactional Memory, atores, abordagens para explorar concorrência.
5. Distribuição (24h): projeto, implementação e uso de interface de programação, RESTful, serviços web e micro-serviços (microservices).
6. Troca de mensagens (*messaging*) (8h).

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

* (Faz com orientação). Explicar a construção de software, concorrência/distribuição e a relação desses com a engenharia de software e 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). 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.
* (Segue instruções). Explicar conceitos de concorrência e distribuição.
* (Segue instruções). Criar código que faz uso de concorrência, distribuição e ferramentas correlatas.

Bibliografia básica:

* BUTCHER, P. Seven Concurrent Models in Seven Weeks. Pragmatic Bookshelf, 2014. ISBN-10: 1937785653
* DAIGNEAU, R. Service Design Pattern. Addison-Wesley, 2011. ISBN 032154420X.
* FOSTER, I. Designing and building parallel programsconcepts and tools for parallel software engineering Reading, Mass.: Addison-Wesley, 1995. ISBN 0201575949.

Bibliografia complementar:

* NEWMAN, S. Building Microservices. O’Reilly, 2015.
* BARNES, L. L. Client/server & beyound strategies for 21st century. Prentice-Hall, 1997. ISBN 0135325161.
* SHATZ, S. M. Development of distributed software concepts and tools. Macmillan, 1993. ISBN 0024096113.
* PAGE-JONES, M. Fundamentals of object-oriented design in UML.  Addison-Wesley, 2003. ISBN 020169946X.
* RODRIGUES, L.; GUERRAOUI, R. Introduction to reliable distributed programming. Springer, 2006. ISBN 9783540288459.

Informações: