#### **Software para Sistemas Ubíquos**

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. Sistemas de informação que fazem uso de dispositivos (ubíquos) (16h): smartphones, sensores, internet das coisas (IoT), *stream analytics* e aspectos de segurança (vulnerabilidades, criptografia, certificados digitais).
2. Definição de arquiteturas para soluções móveis (16): conectar serviços, possivelmente de grande volume, fluxo e em tempo real, com a necessidade de analisá-los.
3. Desenvolvimento de código para smartphone, sensor ou outro dispositivo capaz de alimentar/receber informações de sistema de informação (32h).

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

* (Faz com orientação). Explicar a construção de software, dispositivos e a relação desses com a engenharia de software e de sistema.
* (Faz sem orientação). Explicar a construção de software e a relação dessa área de conhecimento com as demais da engenharia de software.
* (Faz com orientação). Usar e configurar ferramentas para edição, compilação, depuração (debugging), build, teste, controle de versão, coleta de medidas (desempenho, consumo de memória, análise estática e cobertura), integração contínua, e aplicativos de linha de comandos para busca e encadear comandos.
* (Faz com orientação). Criar código orientado a objetos em conformidade com o projeto (design) detalhado seguindo estratégias recomendadas.
* (Faz com orientação). Usar processos para a construção de software (criação de código, controle de versão, inspeção e integração).
* (Faz com orientação). Detalhar projeto (design) em conformidade com requisitos de software.
* (Faz com orientação). Detalhar projeto de interação (design) em conformidade com requisitos de software.
* (Faz com orientação). Localizar e reutilizar código (bibliotecas e frameworks).
* (Faz com orientação). Colaborar com a construção de código em equipe.
* (Faz com orientação). Explicar a construção de software, segurança e a relação desses com a engenharia de software e de sistema.
* (Segue instruções). Auxiliar na criação de modelo de ameaça.
* (Segue instruções). Criar código que faz uso de recursos de segurança (criptografia de dados, assinatura e verificação de assinatura digital).

Bibliografia básica:

* MCGRAW, G. Software Security: Build Security In. Addison-Wesley, 2006.
* GREENGARD, S. The Internet of Things. The MIT Press, 2015.
* WILMSHURST, T. An introduction to the design of small-scale embedded systems. Palgrave, 2001. ISBN 0333929942.

Bibliografia complementar:

* MANICO, J.; DETLEFESEN, A. Iron-Clad Java: Building Secure Web Applications. McGraw-Hill, 2014.
* HOWARD, M. 19 deadly sins of software securityprogramming flaws and how to fix them. McGraw-Hill/Osborne, 2005. ISBN 0072260858.
* MARK, D. Dominando o desenvolvimento no iPhone: explorando o SDK do iPhone. Alta Books, 2009. ISBN 97885760833757.
* VAHID, F. Embedded system design: a unified hardware/software introduction. John Wiley & Sons, 2002. ISBN 0471386782.
* KERN, C.; KESAVAN, A.; DASWANI, N. Foundations of security: what every programmer needs to know. Apress, 2007. ISBN 9781590597842.

Informações: