UNIVERSIDADE FEDERAL DO ABC – UFABC

CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PLANO DE ENSINO

ANO LETIVO	QUADRIMESTRE	TURNO	CAMPUS
2018	Q3	Noturno	Santo André

CÓDIGO	NOME	TPI
MCZA034-14	Programação Segura	2-2-4
TURMA	**RECOMENDAÇÕES**	
NA2MCZA034-17SA	Algoritmos e Estruturas de Dados I	

EMENTA

Segurança no processo de desenvolvimento de software; vulnerabilidades: descrição, tecnologias (linguagens, sistemas operacionais) envolvidas,
prevenção e correção; ferramentas para prevenção de vulnerabilidade; Características relevantes de linguagens de programação: sistemas de exceções, sistema de tipos, código, nativo versus bytecode, outras características. prática: busca por vulnerabilidades em produtos reais.

OBJETIVOS

(i) Conhecer fundamentos de segurança no processo de desenvolvimento de software; (ii) Conhecer classes de vulnerabilidade de software associadas a linguagens de programação, sistemas operacionais e sistemas de comunicação; (iii) Conhecer técnicas de detecção e prevenção de vulnerabilidade em software; (iv) Realizar práticas de análise de código de software, por inspeção manual e com apoio de ferramentas de análise, para identificação de código vulnerável; (v) Conhecer boas práticas de programação que auxiliam na proteção de dados e sistemas contra o mau uso de programas.

PLANEJAMENTO PRELIMINAR DE AULAS

Semana 1: Apresentação da disciplina; fundamentos; processo de desenvolvimento de software seguro. Exemplo com vulnerabilidades de validação da entrada cross-site scripting (xxs).

Semana 2: Vulnerabilidades de código; buffer overflow; shellcode.

Semana 3: Validação da entrada e vulnerabilidades associadas; strings de formatação; injeção de código.

Semana 4: Técnicas de detecção: estáticas e dinâmicas; inspeção manual e classes de ferramentas.

Semana 5: Vulnerabilidades de tempo e estado (race conditions); tratamento de exceções; mais vulnerabilidades de aplicações Web e BD.

Semana 6: Prova 1 e projeto de análise de código.

Semana 7: Mal uso de bibliotecas de programação e de segurança da informação (criptografia enfraquecida, erros em gerenciamento de acesso); aspectos em aplicações móveis.

Semana 8: Vulnerabilidades de projeto; qualidade de código; encapsulamento; ambiente; normas de segurança e de qualidade de software; aspectos em ambientes de computação distribuída.

Semana 9: Prática com ferramentas de análise de código.

Semana 10: Prova 2 e apresentações de projetos.

Semana 11: Prova Substitutiva e apresentações de projetos.

Semana 12: Mecanismo de Recuperação.

AVALIAÇÕES

Avaliações do Período Letivo Regular:

Composição: 2 provas e atividades durante o quadrimestre

- 30% prova 1: semana 6 (25/10/2018)
- 30% prova 2: semana 10 (22/11/2018)
- 25% projeto de análise de código
- 15% atividades de laboratório e exercícios

Avaliação Substitutiva:

Estarão habilitados para a avaliação substitutiva os alunos que se ausentarem a uma das avaliações do período regular e contemplados pelo benefício de acordo com a Resolução CONSEPE no. 181, de 23 de outubro de 2014.

Data da prova sub: semana 11 (29/11/2018)

Caso o aluno se ausente de mais de uma avaliação do período regular, o conceito da avaliação substitutiva será concedido para UMA ÚNICA avaliação não realizada, privilegiando a de maior peso ponderado.

Alunos que fizeram todas as avaliações NÃO TERÃO DIREITO à avaliação substitutiva.

Avaliação de Recuperação:

Estarão habilitados para a avaliação de recuperação os alunos que obtiverem conceito final **D** ou **F** na conclusão de todas as atividades e avaliações aplicadas no período letivo regular, obedecendo as regras indicadas na Resolução CONSEPE no. 182, de 23 de outubro de 2014.

Data da prova de recuperação: semana 12 (06/12/2018)

ATIVIDADES DE APOIO

Esta disciplina prevê um horário de atendimento extraclasse para atividades de apoio aos estudantes regulares desta turma, conforme disposto na Resolução CONSUNI 183, de 31 de outubro de 2017.

Os horários de atendimento semanal terão carga horária total de 2 horas, sendo realizadas nos seguintes dias, locais e horários:

- Quintas-feiras, das 19:00h às 21:00h, sala 533-2.

BIBLIOGRAFIA RECOMENDADA

Bibliografia Básica

- CORREIA, M.P.; SOUSA, P.J.: "Segurança no Software" 2a ed., FCA, 2017.
- HOWARD, M.; LEBLANC, D.; VIEGA, J.: "24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them". McGraw-Hill Ed, 2009.

Bibliografia Complementar

- THOMPSON, H.; CHASE, S. G. The Software Vulnerability Guide. Hingham, USA: Charles River Media, 2005.
- DOWD, M.; MCDONALD, J.; SCHUH, J. The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities. Upper Saddle River, NJ: Pearson, 2006.
- CHESS, B.; WEST, J. Secure Programming with Static Analysis. Boston, USA: Addison-Wesley Professional, 2007.
- GRAFF, M. G.; VAN WYK, K. R. Secure coding: principles and practices. Sebastopol, USA: O'Reilly, 2003.
- HOWARD, M.; LEBLANC, D. Writing secure code. 2 a edição. Redmond, USA: Microsoft Press, 2003.
- SEBESTA, R. Conceitos de linguagens de programação. 9a edição. Porto Alegre, RS:

Bookman, 2011.
- HARBISON, S.; STEELE JR, G. L. C: manual de referência. São Paulo, SP: Prentice Hall/Ciência Moderna, 2002.
- ROCHKIND, M. Advanced UNIX Programming, 2 a edição. Boston, USA: Addison-Wesley, 2004.
- STEVENS, W. R.; RAGO, S. Advanced Programming in the UNIX Environment. 2a edição. Boston, USA: Addison-Wesley, 2008.

PROFESSOR(ES) RESPONSÁVEL(IS)
Profa. Dra. Denise Hideko Goya Prof. Dr. Carlos da Silva dos Santos