

MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA
DIRETORIA DE ENSINO (DIREN)
DEPARTAMENTO DE ENSINO SUPERIOR (DEPES)
DEPARTAMENTO DE INFORMÁTICA (DEPIN)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO (BCC)

DEPARTAMENTO	PLANO DE CURSO DA DISCIPLINA
DEPIN - Departamento Acadêmico de Informática	ARQUITETURA E PADRÕES DE SOFTWARE

CÓDIGO	PERÍODO	ANO	SEMESTRE	PRÉ-REQUISITOS
GCC1520	5º	2012	2	GCC 1415 Programação de Software para WEB GCC 1416 Análise e Projeto de Sistemas
CRÉDITOS	AULAS/SEMANA			TOTAL DE AULAS NO SEMESTRE
4	TEÓRICA	PRÁTICA	ESTÁGIO	
	4	0	0	72

EMENTA
Introdução aos conceitos de arquitetura de software. Estilos e padrões arquiteturais. Táticas para atender atributos de qualidade na arquitetura. Padrões de projeto (<i>design patterns</i>): GoF.

BIBLIOGRAFIA
Bibliografia básica 1. BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. Software architecture in practice. Addison-Wesley Professional, 2003. 2. ROZANSKI, Nick; WOODS, Eóin. Software systems architecture: working with stakeholders using viewpoints and perspectives. Addison-Wesley, 2011. 3. GAMMA, Erich, HELM, Richard, JOHNSON, Ralph & VLISSIDES, John, Padrões de Projeto – Soluções Reutilizáveis de Software Orientado a Objetos, 2ª edição, Porto Alegre: Bookman, 2000. ISBN 9788573076103. Bibliografia complementar 1. FAIRBANKS, George. Just enough software architecture: a risk-driven approach. Marshall & Brainerd, 2010. 2. Richard N. Taylor, Nenad Medvidovic, Eric Dashofy, Software Architecture: Foundations, Theory, and Practice. ISBN: 978-0-470-16774-8. John Wiley & Sons, ©2009. 3. CERVANTES, Humberto; KAZMAN, Rick. Designing software architectures: a practical approach. Addison-Wesley Professional, 2016. 4. SHAW, Mary; GARLAN, David. Software architecture. Englewood Cliffs: Prentice Hall, 1996. 5. CLEMENTS, Paul et al. Documenting software architectures: views and beyond. Pearson Education, 2002.

OBJETIVO GERAL

Familiarizar os alunos com conceitos avançados de desenvolvimento de software com o uso de padrões de software. Desenvolver a capacidade de identificar problemas em sistemas de software, analisá-los, e propor soluções baseadas no uso de arquiteturas e padrões de software.

METODOLOGIA

- Aulas expositivas com recursos audiovisuais
- Desenvolvimento de estudos de casos em laboratório, com uso de desenvolvimento dirigido por testes.
- Atendimento e orientação do professor visando o desenvolvimento do projeto de um sistema de software que envolva padrões estudados na disciplina.

CRITÉRIO DE AVALIAÇÃO

A avaliação semestral da envolve duas provas escritas (P1 e P2). As datas das provas estão no calendário do CEFET. Haverá também trabalhos individuais envolvendo os assuntos da disciplina. Os pesos de cada componente são os seguintes: as provas valem 80% da nota e os trabalhos valem os restantes 20%.

A média semestral (MS) será calculada pela fórmula $MS = [(0,8 \times P1 + 0,2 \times T1) + (0,8 \times P2 + 0,2 \times T2)]/2$

Na expressão acima, T1 e T2 são as médias dos trabalhos passados na primeira e segunda fase do curso, respectivamente.

O aluno que faltar a uma das duas provas (P1 ou P2) terá direito a uma avaliação alternativa, denominada segunda chamada, versando sobre todos os tópicos abordados no curso, e cuja data também é agendada entre docente e discentes. A nota obtida nessa 2ª chamada substituirá a da avaliação P1 ou P2 onde o aluno não esteve presente. Caso ele falte às duas avaliações, terá atribuído o grau ZERO em uma delas.

Segundo o regimento do CEFET-RJ, caso o aluno obtenha média parcial inferior a 3,0 (três e zero) estará reprovado diretamente. Graus MP maiores ou iguais a 7,0 (sete e zero) aprovam diretamente o aluno. Em situações onde o aluno tenha grau MP entre 3,0 inclusive e 7,0 exclusive, terá direito a uma prova final (PF), que, juntamente com a média parcial gerará uma nova média, denominada média final (MF). Essa média é calculada da seguinte forma:

$$MF = (MP + PF) / 2$$

Para ser aprovado, o aluno deve alcançar uma MF maior ou igual a 5,0 (cinco e zero). Caso contrário, estará reprovado, devendo repetir a componente curricular.

CHEFE DO DEPARTAMENTO

NOME	ASSINATURA

PROFESSOR RESPONSÁVEL PELA DISCIPLINA

NOME	ASSINATURA

PROGRAMA

1. Introdução aos conceitos de arquitetura

- 1.1. Definições de arquitetura de software
- 1.2. Descrição de Arquitetura
 - 1.2.1. Tradicional (Visão, ponto de vista, perspectiva)
 - 1.2.2. Ágil (cenários)
- 1.3. Atributos de qualidade e sua relação com arquitetura
- 1.4. Processo de definição de arquitetura (ADD - Attribute-Driven Design)
- 1.5. Processo de avaliação de arquitetura (ATAM – Architecture Tradeoff Analysis Method)
- 1.6. Prova de Conceito
2. Estilos e padrões arquiteturais
 - 2.1 Layers
 - 2.2 MVC
 - 2.3 Plugin Architecture
 - 2.4 Pipe-and-Filter
 - 2.5 Client-Server
 - 2.6 Peer-to-peer
 - 2.7 Shared-Data
 - 2.8 Broker
 - 2.9 Publish-Subscribe
 - 2.10 SOA
 - 2.11 REST
 - 2.12 Microservices
 - 2.13 Map-Reduce
3. Táticas para atender atributos de qualidade na arquitetura
 - 3.1 Disponibilidade
 - 3.2 Interoperabilidade
 - 3.3 Modificabilidade
 - 3.4 Performance
 - 3.5 Segurança
 - 3.6 Testabilidade
 - 3.7 Usabilidade
4. Padrões de projeto
 - 4.1. Padrões GoF: visão geral, motivação, classificação
 - 4.2. Padrões GoF, parte I: Adapter, Aggregate, Singleton, Iterator, Template Method, Decorator
 - 4.3. Padrões GoF, parte II: Factory Method, Abstract Factory, Façade
 - 4.4. Padrões GoF, parte III: Strategy, State, Chain of Responsibility, Observer, Mediator