

## PLANO DE ENSINO

### Identificação

**Componente curricular:** IMD0103 – Arquitetura e Projeto de Software (30h) **Período letivo:** 2018.2

**Professor:** Dr. Everton Ranielly de Sousa Cavalcante (2316877)

**Local e horário de aulas:** 4M56 4T1 / CIVT-A309

### Objetivos

- Introduzir conceitos relacionados a arquiteturas de *software*, seu projeto, documentação e análise
- Apresentar e discutir problemas decorrentes da adoção de más práticas de projeto e implementação de sistemas de *software* que podem impactar em suas arquiteturas e, conseqüentemente, em atributos de qualidade

### Competências e habilidades

Através deste componente curricular, o estudante deverá ser capaz de

- Compreender os principais conceitos relacionados a arquiteturas de *software*
- Reconhecer o papel central que arquiteturas podem exercer no desenvolvimento de sistemas de *software* e sua influência sobre atributos de qualidade
- Identificar problemas no contexto do projeto e desenvolvimento de sistemas de *software* que podem refletir-se a nível arquitetural e como trata-los

### Conteúdo

- Conceitos em Arquitetura de Software
- Padrões e estilos arquiteturais
- Visões arquiteturais
- Modelagem de *software* e documentação de arquiteturas
- Atributos de qualidade
- Avaliação arquitetural
- Dívida técnica
- Anomalias de código e arquiteturais
- Evolução, degradação e refatoração de *software*

### Metodologia

A metodologia de ensino a ser adotada na condução deste componente curricular estrutura-se fundamentalmente em:

- aulas teóricas expositivas, com o objetivo de apresentar as principais temáticas relacionadas à área de Arquitetura de Software;
- trabalhos práticos, no intuito de capacitar o estudante para o projeto e análise adequados de arquiteturas de *software*;

- (iii) estudos dirigidos, com o objetivo de incentivar uma atividade intelectual e autodidata para promover maior capacidade de raciocínio e compreensão dos conteúdos.

### Recursos didáticos

- Computador
- Projetor multimídia
- Quadro branco
- Palestras e/ou aulas em vídeo

### Avaliação

**Sobre os instrumentos para avaliação da aprendizagem.** Os principais instrumentos para avaliação da aprendizagem serão (i) uma avaliação escrita individual e presencial e (ii) trabalhos práticos. Ao longo da condução do componente curricular, poderão também ser realizados outros tipos de atividades, tais como estudos dirigidos e outros tipos de exercício.

**Sobre os trabalhos.** Os trabalhos práticos versarão sobre o projeto e análise de arquiteturas de *software* de sistemas existentes, em desenvolvimento ou já em operação, utilizando os conceitos e técnicas vistos ao longo da execução do componente curricular. Na data estabelecida para entrega do trabalho, além do envio dos artefatos solicitados, deverá ser feita uma apresentação oral de até dez minutos à turma, expondo e discutindo o trabalho desenvolvido. O rendimento desse trabalho será dado pela avaliação do relatório a ser entregue e da apresentação oral.

**Sobre a composição do rendimento acadêmico.** O rendimento acadêmico do componente curricular será calculado a partir da média ponderada dos rendimentos de cada atividade avaliativa desenvolvida (avaliação escrita, trabalhos práticos, estudos dirigidos etc.), expressos em valores numéricos entre 0,0 e 10,0 pontos. Ao estudante que não comparecer à avaliação escrita ou não entregar alguma das atividades será atribuída nota zero.

Considerando o disposto no Art. 74 da Resolução nº 197/2013-CONSEPE, de 10 de dezembro de 2013, que sobre dispõe sobre normas dos programas e cursos de Pós-graduação da UFRN, a avaliação de desempenho do estudante no componente curricular materializa-se na forma dos conceitos *A* (muito bom), *B* (bom), *C* (regular), *D* (insuficiente) e *E* (reprovado por faltas). Para fins de conversão entre tal escala de conceitos e valores numéricos utilizados para computação do desempenho do estudante nas atividades avaliativas, serão utilizadas as seguintes relações de correspondência:

Escala numérica	Conceito correspondente
9,0 – 10,0	A
8,0 – 8,9	B
7,0 – 7,9	C
0,0 – 6,9	D

Conforme Art. 75 da Resolução nº 197/2013-CONSEPE, a aprovação no componente curricular está condicionada (i) à obtenção de conceito igual ou superior a *C* e (ii) à presença mínima de 75% das aulas presenciais ministradas, consoante o Art. 7º da Resolução nº 01/2007-CNE/CES, de 8 de junho de 2007, que estabelece as normas para cursos de Pós-graduação *lato sensu* no Brasil. O estudante que não atinge tais critérios de aprovação é considerado reprovado.

### Cronograma de aulas

Data	Dia da semana	Conteúdo
04/07/2018	Quarta-feira	Apresentação do plano de ensino
11/07/2018	Quarta-feira	Arquitetura de Software: Conceituação
18/07/2018	Quarta-feira	Padrões, estilos e visões arquiteturais
25/07/2018	Quarta-feira	Modelagem de <i>software</i> e documentação de arquiteturas

01/08/2018	Quarta-feira	Avaliação arquitetural
08/08/2018	Quarta-feira	Anomalias e refatoração
15/08/2018	Quarta-feira	Atributos de qualidade
22/08/2018	Quarta-feira	Avaliação escrita

### Bibliografia básica

- BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. *Software Architecture in practice*. 3. ed. USA: Addison-Wesley/Pearson Education, Inc., 2013 (The SEI Series in Software Engineering).
- BUSCHMANN, Frank; MEUNIER, Regine; ROHNERT, Hans; SOMMERLAD, Peter; STAL, Michael. *Pattern-Oriented Software Architecture – Volume 1: A system of patterns*. United Kingdom: John Wiley & Sons Ltd., 1996 (Wiley Series in Software Design Patterns).
- CLEMENTS, Paul *et al.* *Documenting software architectures: Views and beyond*. USA: Addison-Wesley/Pearson Education, Inc., 2010 (The SEI Series in Software Engineering).
- CLEMENTS, Paul; KAZMAN, Rick; KLEIN, Mark. *Evaluating software architectures: Methods and case studies*. USA: Addison-Wesley/Pearson Education, Inc., 2002 (The SEI Series in Software Engineering).
- FOWLER, Martin; BECK, Kent; BRANT, John; OPDYKE, William; ROBERTS, Don. *Refactoring: Improving design of existing code*. USA: Addison-Wesley Longman, Inc., 1999.
- TAYLOR, Richard N.; MEDVIDOVIC, Nenad; DASHOFY, Eric M. *Software Architecture: Foundations, theory, and practice*. USA: John Wiley & Sons, Inc., 2010.

### Bibliografia complementar

- KNODEL, Jens; NAAB, Matthias. *Pragmatic evaluation of software architectures*. Switzerland: Springer International Publishing AG, 2016 (The Fraunhofer IESE Series on Software and Systems Engineering).
- OQUENDO, Flavio; LEITE, Jair; BATISTA, Thaís. *Software Architecture in action: Designing and executing architectural models with SysADL grounded on the OMG SysML Standard*. Switzerland: Springer International Publishing AG, 2016 (Undergraduate Topics in Computer Science).
- SURYANARAYANA, Girish; SAMARTHYAM, Ganesh; SHARMA, Tushar. *Refactoring for software design smells: Managing technical debt*. USA: Morgan-Kaufman/Elsevier, Inc., 2015.
- VOGEL, Oliver; ARNOLD, Ingo; CHUGTAI, Arif; KEHRER, Timo. *Software Architecture: A comprehensive framework and guide for practitioners*. Germany: Springer-Verlag Berlin Heidelberg, 2011.

### Observações

**Sobre o atendimento extraclasse.** O atendimento extraclasse será feito entrando-se em contato previamente com o professor via *e-mail*, através do endereço [everton@dimap.ufrn.br](mailto:everton@dimap.ufrn.br).

**Sobre o controle de presença às aulas presenciais.** A frequência a aulas presenciais será rigorosamente registrada via sistema SIGAA e/ou lista de presença.

**Sobre a realização de aulas extra e/ou de reposição.** Conforme a necessidade, poderão ser realizadas aulas extras e/ou de reposição, as quais contabilizarão como carga horária ministrada e terão frequência registrada normalmente. O dia da semana, horário e local de realização de tais aulas será estabelecido em comum acordo entre o professor e a maioria absoluta dos estudantes da turma, ouvidas a Coordenação do Programa de Residência em Tecnologia da Informação Aplicada à Área Jurídica e a gestão do Órgão do Poder Judiciário associado.