



2018

Universidade de Brasília-FGA

Plano de Ensino - Desenho de Software

Plano de ensino da disciplina de Desenho de Software, ministrada pela professora Milene Serrano.

UNIVERSIDADE DE BRASÍLIA

FACULDADE DO GAMA

CURSO:	ENGENHARIAS	SEMESTRE/ANO:	02/2018
DISCIPLINA:	Arquitetura e Desenho de Software	CÓDIGO:	203882
CARGA HORÁRIA:	60 hrs	CRÉDITOS:	04
PROFESSOR:	Dra. Milene Serrano Segunda e Sexta 08:00-10:00 ↳ <i>Local Lab. MOCAP (segunda-feira e sexta-feira)</i>		

PLANO DE ENSINO

1. EMENTA

1. Contextualização para Desenho de Software.
2. Projeto/Desenho de Software Orientado a Objetos.
3. Padrões de Projeto (GRASP e GoF).
4. Contextualização para Arquitetura de Software.
5. Estilos Arquiteturais.
6. Padrões Arquiteturais.
7. Documentação de Arquitetura de Software.
9. *Framework*.
8. *Model Driven Architecture* (MDA) <<em aulas desafio, provavelmente>>.

2. OBJETIVOS DA DISCIPLINA

A disciplina de Arquitetura e Desenho de Software visa apresentar ao aluno uma série de técnicas, modelos e conceitos relacionados ao projeto de software de modo que se espera, ao final do curso, que o aluno seja capaz de usá-los para a solução de algum problema através de um software.

Espera-se do aluno a capacidade de transformar requisitos de software (funcionais e não funcionais) em um desenho (projeto) de software. Tal projeto deve contemplar / representar diversas características do software, tais como: estruturas, comportamentos, implantação e arquiteturas. Conceitos de projeto (ex. abstração, coesão, acoplamento, decomposição, modularização, encapsulamento, dentre outros) bem como boas práticas, como os padrões de projeto, também serão vistos.

Como plano de fundo da disciplina serão cobertas abordagens tradicionais, ágeis e emergentes.

Conteúdos de Arquitetura de Software (ex. conceitos básicos, estilos, padrões e visões arquiteturais) serão também abordados ao longo do curso. Adicionalmente, tem-se o estudo de reutilização de software.

3. CONTEÚDO PROGRAMÁTICO (síntese)

- Tópico 1: Introdução
 - Plano de Ensino
 - Introdução à Disciplina
 - Revisão de Engenharia de Requisitos

➤ Tópico 2: Desenho de Software

- 1. Projeto/Desenho de Software
 - 1.1. Projeto/Desenho de Software Não Orientado a Abordagens Específicas
 - 1.2. Projeto/Desenho de Software Orientado a Abordagens Tradicionais
 - 1.3. Projeto/Desenho de Software Orientado a Abordagens Ágeis
 - 1.4. Projeto/Desenho de Software Orientado a Abordagens Alternativas
 - 2. Projeto/Desenho de Software Orientado a Objetos
- 2. Processos, Metodologias e Abordagens
 - 2.1. OpenUp, PUÁgil, Scrum, SAFE, Lean, XP e Similares
- 3. Paradigmas e Boas Práticas
 - 3.1. Orientação a Objetos
- 4. Padrões de Projeto
 - 4.1. Padrões GRASP
 - 4.1.1. Criador
 - 4.1.2. Especialista
 - 4.1.3. Alta Coesão
 - 4.1.4. Baixo Acomplamento
 - 4.1.5. *Controller*
 - 4.1.6. Polimorfismo
 - 4.1.7. Indireção
 - 4.1.8. Invenção Pura
 - 4.1.9. Variações Protegidas
 - 4.2. Padrões GoFs
 - 4.2.1. GoFs Criacionais
 - 4.2.2. GoFs Estruturais
 - 4.2.3. GoFs Comportamentais

➤ Tópico 3: Arquitetura de Software

- 5. Contextualização para Arquitetura de Software
 - 5.1. Definições
- 6. Estilos Arquiteturais
 - 6.1. Visão Geral
 - 6.2. *Stand-alone*
 - 6.3. Cliente-servidor
 - 6.4. *Peer to Peer*

- 6.5. Objetos Distribuídos
- 6.6. N-camadas
- 6.7. Dutos e Filtros (*Pipes and Filters*)
- 6.8. Repositório
- 6.9. Orientado a Eventos (*Publisher/Subscriber*) <<a depender da disponibilidade>>
- 6.10. Orientado a Serviços (SOA) <<a depender da disponibilidade>>
- 7. Padrões Arquiteturais
 - 7.1. Visão Geral
 - 7.2. Padrão *Model-View-Controller* (MVC), e
 - 7.3. Padrão BROKER e Padrões Emergentes <<a depender da disponibilidade>>
- 8. Documentação de Arquitetura de Software
 - 8.1 Trabalhando com Múltiplas Visões (notação UML)
- 9. Framework
 - 9.1 Noções Básicas sobre Framework e Reutilização de Software
- 10. *Model Driven Architecture* (MDA) <<apresentado via aula desafio, diante da disponibilidade>>
 - 10.1 Noções Básicas
- Tópico 4: Seminários
 - 4.1. Exposição dos trabalhos via Seminários
- Tópico 5: Fechamento
 - 5.1. Apresentação e Revisão de Menções
 - 5.2. *Fishbowl* e outras atividades para debatermos sobre o conteúdo do curso
 - 5.3. Aulas desafio <<a depender da disponibilidade>>

OBS_01: Conteúdos “desafio” também serão explorados, tendo disponibilidade de aulas, mais ao final do curso. Linha de Produto e MDA são exemplos de tópicos a serem considerados nesse caso.

OBS_02: Os conteúdos dos tópicos mencionados serão ministrados em diferentes módulos, conforme cronograma.

4. MÉTODO DE ENSINO

As aulas são expositivas, realizadas em sala de aula e laboratório, com conteúdo teórico e atividades de modelagem e implementação. A fim de fortalecer a aprendizagem na disciplina, a avaliação estilo prova (individual) é complementada com propostas de avaliações baseadas em atividades intra (via dinâmicas) e extraclasse (as quais variam de semestre para semestre) bem como em trabalhos de desenvolvimento e programação baseados no conteúdo abordado ao longo da disciplina.

A plataforma Moodle é utilizada como mecanismo oficial de comunicação entre professora, alunas(os) e monitoras(es). O Plano de Ensino e o Cronograma de Atividades da disciplina serão disponibilizados às(aos) alunas(os) para

consulta e *download* no ambiente Moodle da disciplina. Os tópicos serão estudados em diferentes livros, os quais são listados no tópico "Bibliografia".

5. AVALIAÇÃO

A menção final será constituída das seguintes avaliações:

- **Nota 1** (70% da Nota Final), constituída por:
 - Prova (individual e sem consulta).
- **Nota 2** (30% da Nota Final), constituída por:
 - Trabalho de Desenvolvimento de Software (a ser definido em detalhes em breve, sendo focado nos conteúdos programáticos da disciplina), e rendimento controlado via dinâmicas presenciais (vide previsão no cronograma – faltas nas dinâmicas comprometem essa nota). Avaliações de artefatos e implementações individualizadas.
- Atividades Intra e Extraclasse (adicionais) também poderão ocorrer ao longo do semestre, a depender do período letivo. Ocorrendo essas avaliações adicionais, suas pontuações variam de semestre para semestre, e de atividade para atividade. Portanto, participem! :)
- Em caso de falta na prova, o(a) aluno(a) poderá realizar a prova substitutiva (dia previsto, conforme cronograma). Para tanto, a falta precisa ser devidamente justificada. Prova SUB, individual, conteúdo total, e capaz de substituir apenas a nota da prova faltante (no caso, apenas a prova que o(a) aluno(a) faltar e justificar).

A Nota Final será dada pela Média Final (MF), calculada por:

$$\text{Nota Final} = \text{MF} = (\text{Nota 1} * 70\%) + (\text{Nota 2} * 30\%)$$

O aluno será aprovado se **MF** \geq **5** e **Frequência** \geq **75%**.

- Exemplo: Nota 1 (8,0), Nota 2 (7,0) e Frequência \geq 75%, temos:
- $\text{MF} = (8,0 * 70\%) + (7,0 * 30\%) =$
 $5.6 + 2.1 = 7,7$
- Portanto, **aprovado** com **MS**.
- Não está considerando a nota extra das atividades intra e extraclasse.

Datas importantes (sujeito a alteração): Confira o Cronograma.

AVISOS IMPORTANTES:

- Os trabalhos devem ser acompanhados de documentação (completa, com toda a especificação do projeto com base nos conteúdos programáticos ministrados ao longo da disciplina – i.e. Engenharia de Requisitos, Artefatos de Projeto, Detalhamento Metodológico, Modelos, Padrões de Projeto (GRASPs e GoFs), Arquitetura (Estilos, Padrões e Visões) e Reutilização), manual de instalação (caso explore suportes tecnológicos diferenciados), e código fonte na linguagem de programação do projeto. Será exigido o uso da Wiki do Git para organizar a documentação, facilitando a exposição tanto nas dinâmicas, quanto no seminário final. Encaminhar, via Moodle, quando solicitado, os dados da entrega.
- Será cobrado o uso do Git. Usem “nomes” identificáveis, permitindo identificar cada membro do grupo.
- Qualquer atividade, trabalho e/ou questão de prova que seja identificada como cópia receberá nota zero (todos os envolvidos).
- Presença em sala de aula será um critério considerado. Portanto, sejam pontuais e procurem não faltar! Tolerância: 10 minutos (não repetidas vezes). :)

6. BIBLIOGRAFIA

BÁSICA:

1. (BIBLIOTECA FGA – 15 unidades) Pfleeger, Shari Lawrence. Engenharia de Software: Teoria e Prática. 2ª. Edição. São Paulo: Prentice Hall, c2004. xix, 535 p. ISBN 9788587918314. [OPEN ACCESS] Slides Prentice Hall Education http://wps.prenhall.com/br_pfleeger_engsoftware_2/ (último acesso: 2017)
2. (BIBLIOTECA FGA – 16 unidades) Larman, Craig. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos e ao Desenvolvimento Iterativo. 3ª. Edição. Porto Alegre, RS: Bookman, 2007. xiv, 695 p. ISBN 9788560031528 OU Larman, Craig. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos. Porto Alegre: Bookman, 2000. 492p. ISBN 85-7307-651-8 OU [OPEN ACCESS] Larman, Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Unified Process. 2ª Edition. Free Book for Everyone.
3. (EBRARY) Zhu, H. Software Design Methodology: From Principles to Architectural Styles. Butterworth-Heinemann. 2005.

COMPLEMENTAR:

1. (BIBLIOTECA FGA – 11 unidades) Pressman, Roger S. Engenharia de Software: Uma Abordagem Profissional. 7ª. Edição. Porto Alegre: AMGH, 2011. 780 p. ISBN 9788563308337 OU Pressman, Roger S. Engenharia de Software: Uma Abordagem Profissional. 5ª. Edição. Rio de Janeiro: McGraw-Hill, c2002. ISBN 8586804258.
2. (BIBLIOTECA FGA – 7 unidades) Page-Jones, Meilir. Fundamentals of Object-Oriented Design in UML. New York: Dorset, The Addison-Wesley object technology series. c2000. xxi, 458 p. ISBN 020169946X.
3. (BIBLIOTECA FGA – 4 unidades) Sommerville, Ian. Engenharia de Software. 8ª. Edição. São Paulo: Pearson Addison Wesley. 2007. 552 p. ISBN 978-85-88639-28-7. [OPEN ACCESS] Slides Pearson Education

4. (EBRARY) Coplien, James and Bjørnvig, Gertrud. Learn Architecture: For Agile Software Development. John Wiley & Sons. 2010.
5. (EBRARY) Duggan, Dominic. Quantitative Software Engineering Series : Enterprise Software Architecture and Design : Entities, Services, and Resources (1). Hoboken, US: Wiley-IEEE Computer Society Pr, 2012.
6. (EBRARY) Klimczak, Erik. Design for Software: A Playbook for Developers (1). Somerset, GB: Wiley, 2013.
7. (Open Access) Portal sobre Arquitetura de Software. Disponível em: www.softwarearchitectureportal.org/ (último acesso: 2017)
8. (Open Access) Portal sobre MDA. Disponível em: <http://www.omg.org/mda/> (último acesso: 2017)

7. PROGRAMAÇÃO PREVISTA (Sujeita a Alterações)

Dia	Aula	Principais Tópicos
13/08/2018	Aula 01 - Introdutória	<p>Tópico 1: Introdução/Revisão</p> <p>Plano de Ensino</p> <p>Introdução à Disciplina</p> <p>Divisão dos Grupos</p> <p>Requisitos (Revisão)</p>
17/08/2018	Aula 02	<p>Tópico 2: Desenho de Software</p> <p>Módulo Desenho/Projeto Não Orientado à Abordagens Específicas</p> <p>Projeto/Desenho</p> <p>Planos de Riscos, Prazos e Custos</p> <p>5W2H</p> <p>Causa-Efeito</p> <p>Mapa Mental</p> <p>RichPicture</p> <p>Léxicos/Glossário</p>
20/08/2018	Aula 03	<p>Dinâmica I</p> <p>Elaboração de Artefatos Genéricos (independentes da metodologia a ser utilizada)</p>
24/08/2018	Aula 04	<p>Tópico 2: Desenho de Software</p> <p>Módulo Processos/Metodologias/Abordagens</p> <p>Apresentações de Processos, Metodologias e/ou Abordagens (+/- 10min/grupo)</p> <p><<a ser definido, por grupo>></p> <p>Exemplos:</p> <p>RUP</p> <p>OpenUp</p> <p>XP</p> <p>Scrum</p> <p>SAFE</p> <p>Lean</p> <p>Kanban</p> <p>Outros</p>
27/08/2018	Aula 05	<p>Dinâmica II</p> <p>Desenho da Metodologia Específica para o Projeto (Bizagi ou outra)</p>
31/08/2018	Aula 06	<p>Tópico 2: Desenho de Software</p> <p>Módulo Paradigmas e Boas Práticas</p> <p>Orientação a Objetos (princípios)</p>

3/9/2018	Aula 07	<p>Tópico 2: Desenho de Software</p> <p>Módulo Desenho/Projeto Orientado à Abordagens Tradicionais</p> <p>Modelagem Tradicional (notação UML)</p> <p>Modelo de Classes</p> <p>Modelos de Sequência</p> <p>Modelo de Colaboração <<a depender da disponibilidade de tempo>></p>
7/9/2018	Feriado	Independência
10/9/2018	Aula 08	<p>Tópico 2: Desenho de Software</p> <p>Módulo Desenho/Projeto Orientado à Abordagens Alternativas</p> <p>Notação SADT</p> <p>Modelagem NFR</p> <p>Modelagem i* <<a depender da disponibilidade de tempo>></p>
14/9/2018	Aula 09	<p>Dinâmica III</p> <p>Modelagem em aderência à metodologia estabelecida para cada projeto</p>
17/9/2018	Aula 10	<p>Tópico 2: Desenho de Software</p> <p>Módulo Padrões de Projeto - GRASP</p> <p>Criador</p> <p>Especialista</p> <p>Alta Coesão</p> <p>Baixo Acomplamento</p> <p>Controller</p> <p>Polimorfismo</p> <p>Indireção</p> <p>Invenção Pura</p> <p>Variações Protegidas</p>
21/9/2018	Aula 11	<p>Tópico 2: Desenho de Software</p> <p>Módulo Padrões de Projeto - GoF</p> <p>GoFs Criacionais</p>
24/9/2018	Aula 12	Semana Universitária <<aula não presencial>>
28/9/2018	Aula 13	Semana Universitária <<aula não presencial>>

1/10/2018	Aula 14	Tópico 2: Desenho de Software Módulo Padrões de Projeto - GoF GoFs Estruturais
5/10/2018	Aula 15	Tópico 2: Desenho de Software Módulo Padrões de Projeto - GoF GoFs Comportamentais
8/10/2018	Aula 16	Tópico 2: Desenho de Software Módulo Padrões de Projeto - GoF Apresentações de GoFs (+/- 10min/grupo) <<a ser definido, por grupo>>
12/10/2018	Feriado	Nossa Sra. Aparecida
15/10/2018	Aula 17	Dinâmica IV Modelagem e implementação de padrões. OBS: Na monitoria, haverá uma aula de revisão para a prova, em torno desse dia e antecedendo o dia da prova (a combinar).
19/10/2018	Aula 18 - Prova	Individual, Sem Consulta & Escrita à Mão (Parte Conceitual e Parte Programação - Conteúdo: TODO O CONTEÚDO MINISTRADO ATÉ O MOMENTO). FALTAS APENAS JUSTIFICADAS - Necessário para realização da SUB.
22/10/2018	Aula 19 - SUB	Individual, Sem Consulta & Escrita à Mão (Parte Conceitual e Parte Programação - Conteúdo: TODO O CONTEÚDO MINISTRADO ATÉ O MOMENTO) OBS: Apenas para aqueles que deixaram de realizar a prova, substituindo exclusivamente essa nota faltante.
26/10/2018	Aula 20	Tópico 3: Arquitetura de Software Módulo Visão Geral sobre Arquitetura de Software <<Introdução>>
29/10/2018	Aula 21	Tópico 3: Arquitetura de Software Módulo Estilos e Padrões Arquiteturais (Parte I) Provavelmente: Stand-alone/Cliente-Servidor/P2P/Objetos Distribuídos

2/11/2018	Feriado	Finados
5/11/2018	Aula 22	<p>Tópico 3: Arquitetura de Software</p> <p>Módulo Estilos e Padrões Arquiteturais (Parte II)</p> <p>Provavelmente: N-Camadas (adicionalmente, Padrão MVC)/Repositório/<i>Pipes and Filters</i> /Orientado à Evento <<a depender da disponibilidade de tempo>></p>
9/11/2018	Aula 23	<p>Tópico 3: Arquitetura de Software</p> <p>Módulo Visões Arquiteturais</p> <p>Documento de Arquitetura de Software (DAS)</p>
12/11/2018	Aula 24	<p>Tópico 3: Arquitetura de Software</p> <p>Módulo Reutilização de Software</p> <p>Framework e Outras Soluções Reutilizáveis</p>
16/11/2018	Aula 25 - Ponto Facultativo	Proximidade ao Feriado da Proclamação da República <<aula não presencial>>
19/11/2018	Aula 26	<p>Dinâmica V</p> <p>Arquitetura & Reutilização de Software</p>
23/11/2018	Aula 27	<p>Tópico 4: Seminários</p> <p>Apresentação/Seminário (Primeiro Dia) - 15 min/grupo + tempo para perguntas</p>
26/11/2018	Aula 28	<p>Tópico 4: Seminários</p> <p>Apresentação/Seminário (Segundo Dia) - 15 min/grupo + tempo para perguntas</p>
30/11/2018	Aula 29	<p>Tópico 5: Fechamento</p> <p>Revisão e Menções Finais (possivelmente)</p>

3/12/2018	Aula 30	Tópico 5: Fechamento Debate sobre Resultados da Disciplina (Modelo: <i>Fishbowl</i> - se possível)
7/12/2018	Aula 31	Tópico 5: Fechamento Aula Desafio <<tema: em definição e a depender da disponibilidade>>
* Mudanças no cronograma poderão ocorrer ao longo do semestre.		
* Exercícios de revisão para as provas serão disponibilizados, provavelmente, via monitoria - antes da prova.		
* Se possível, serão desenhadas aulas de revisão em horário alternativo às aulas, junto aos monitores.		
<p>Nas dinâmicas.... <<algumas adaptações poderão ocorrer, dadas necessidades específicas do semestre>></p> <p>Fase de Planejamento:</p> <p>Os 15 minutos iniciais da dinâmica serão consumidos para planejamento dos artefatos por parte das equipes.</p> <p>Nesse período, podem solicitar orientação à professora e/ou aos monitores presentes.</p> <p>Aqui, cabe debate entre os membros sobre os artefatos que serão realizados bem como o escopo de atuação de cada membro ou de cada dupla.</p> <p>Portanto, podem usar "pair design", ou seja, artefatos sendo modelados dois a dois membros...</p> <p>* se envolver implementação na dinâmica, podem usar <i>pair programming</i> . :)</p> <p>Fase de Execução:</p> <p>Serão dedicados 25 minutos da aula para elaboração/modelagem do(s) artefato(s), conforme planejado pelas equipes.</p> <p>* se envolver implementação na dinâmica, caberá desenvolvimento também (pelo menos um esboço da solução). :)</p> <p>Fase de Avaliação:</p> <p>Essa avaliação será realizada a partir de 40 minutos de aula, ou seja, nos 70 minutos finais da aula.</p> <p>Serão 10 minutos, no máximo, por grupo, para mostrar, bem brevemente, o que fizeram...</p> <p>Demais grupos que não estiverem, naquele momento, sendo avaliados, poderão continuar elaborando/modelando o(s) artefatos.</p> <p>Lembrando que essas avaliações são tanto por equipe, quanto individuais...</p>		